ComicVerse: Expanding the Frontiers of AI in Comic Books with Holistic Understanding

by

Gürkan Soykan

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

September 7, 2023

ComicVerse: Expanding the Frontiers of AI in Comic Books with Holistic Understanding

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Gürkan Soykan

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Deniz Yuret (Advisor)

_____

Assoc. Prof. Mehmet Erkut Erdem

_____

Prof. Yücel Yemez

Date: _____

Dedicated to my family, to everyone with whom I've established a telepathic connection, and to those who have always supported me without any doubts.

ABSTRACT

ComicVerse: Expanding the Frontiers of AI in Comic Books with Holistic Understanding
Gürkan Soykan
Master of Science in Computer Science and Engineering
September 7, 2023

Comics are a unique and multimodal medium that conveys stories and ideas through sequential imagery often accompanied by text for dialogue and narration. Comics' elaborate visual language exhibits variations from different authors, cultures, periods, technologies, and artistic styles. Consequently, the computational analysis of comic books requires addressing fundamental challenges in computer vision and natural language processing. In this thesis, I aim to enhance neural comic book understanding by making use of comics' unique multimodal nature and processing comics in a character-centric approach. The primary data source for this thesis is the Golden Age of American Comics due to its public accessibility and abundance of comic series. However, the availability of annotated data is limited. Thus, to achieve my goal, I have adopted a holistic approach composed of four main steps ranging from curating datasets to proposing novel tasks and architectures for comics. The first three steps aim to create a machine-readable comics database by locating comic book panels, identifying their components, and transforming character identities into a dialogue-like structure and the final step uses this database to train a transformer-based model. The first step involves extracting high-quality text data from speech bubbles and narrative box images using OCR models. I decompose comic pages into their constituent components in the second step through detection, segmentation, and association tasks with a refined Multi-Task Learning (MTL) model. Detection involves identifying panels, speech bubbles, narrative boxes, character faces, and bodies. Segmentation focuses on isolating speech bubbles and panels, while the association task involves linking speech bubbles with character faces and bodies. In the third step, I utilize the paired character faces and bodies obtained from the previous stage to create character instances and, subsequently, reidentify and track these instances across sequential panels. In the final step of my thesis, I propose a

multimodal framework by introducing the *ComicBERT* model, which exploits the abovementioned structure. Cloze-style tasks were used to evaluate ComicBERT's contextual understanding capabilities. Furthermore, I propose a new task called *Scene-Cloze*, which predicts the next panel given n previous panels as context. As a result, my approach achieves a new state-of-the-art performance in Text-Cloze and Visual-Cloze tasks with accuracies of 69.5% and 77.1%, respectively, thus getting closer to the human baseline. Overall, the highlights of my contributions are as follows:

1. I curated and shared *COMICS Text+ Dataset* with over two million transcriptions of textboxes from the golden age of comics. In addition, I open-sourced the text detection and recognition models that are fine-tuned for the task and datasets used in their training.

2. I refined a *MTL framework* for detection, segmentation, and association tasks and achieved SOTA results in comic character face and body-to-speech bubble association tasks.

3. I proposed a novel *Identity-Aware Semi-Supervised Learning for Comic Character Re-Identification* framework to generate unified and identity-aligned comic character embeddings and identity representations. Furthermore, I generated two new datasets: the *Comic Character Instances Dataset*, encompassing over a million character instances used in the self-supervision phase, and the *Comic Sequence Identity Dataset*, containing annotations of identities within sets of four consecutive comic panels used in semi-supervision phase.

4. I introduced the multimodal *Comicsformer*, a transformer-encoder architecture capable of processing sequential panels and their constituents. It serves as the backbone for the *Masked Comic Modeling (MCM)* task, a novel self-supervised pre-training strategy for comics, resulting in *ComicBERT*, a potential foundation model for golden age comics. ComicBERT achieves SOTA performance in cloze-style tasks, particularly in text-cloze and visual-cloze tasks, approaching human-level comprehension.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Gürkan Soykan
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans
7 Eylül 2023

Çizgi romanlar, hikayeleri ve fikirleri sıralı görseller aracılığıyla ileten benzersiz ve çok-kipli bir görsel iletişim aracıdır. Görsellere genellikle diyalog ve anlatı metinleri eşlik eder. Çizgi romanların karmaşık görsel dilinde, farklı yazarlar, kültürler, zaman dönemleri, teknolojiler ve sanatsal tarzlar arasında değişiklikler görülür. Bu nedenle çizgi romanların hesaplamalı analizi, temel bilgisayarlı görü ve doğal dil işleme yöntemlerinin kullanılmasını gerektirir. Bu tezde, çizgi romanların benzersiz çoklu modalitesini kullanarak nöral çizgi roman anlayışını artırmayı amaçlıyorum ve bunu yaparken çizgi romanları karakter merkezli bir yaklaşım ile işlemeyi hedefliyorum. Kamuya açık olan *Amerikan Çizgi Romanlarının Altın Çağı* dönemine ait çizgi romanlar ile çalışmak hem döneme ait çizgi roman sayısının fazlalığından hem de telif hakları uygunluğundan dolayı seçilmiştir. Ancak, etiketli veri bu alanda oldukça sınırlıdır. Bu nedenle amacıma ulaşmak için, veri kümesi oluşturma, çizgi romanlar için yeni görevler ve mimariler önerme gibi çalışmalar içeren dört temel adımdan oluşan bütünsel bir yaklaşım benimsedim. İlk adım, *Optik Karakter Tanıma (OCR)* modelleri kullanarak konuşma balonları ve anlatı kutusu görüntülerinden yüksek kaliteli metin verisi çıkarmayı içerir. İkinci adımda, *Çok Görevli Öğrenme (MTL)* modelini iyileştirme yoluyla çizgi roman sayfalarını bileşenlerine ayırmayı başardım. Bu bileşenler şunlardır: panel, konuşma balonu, anlatı kutusu, karakter yüzü ve vücudunu tespit etmek, konuşma balonu ve panel segmente etmek. Ayrıca MTL modelinin görevlerinden birisi de konuşma balonlarını karakterlerin yüz ve vücutlarıyla ilişkilendirmektir. Üçüncü adımda ise, önceki aşamadan elde edilen karakter yüzlerini ve vücutlarını birleştirerek tekil karakterler bulunur ve karakterleri sıralı paneller arasında yeniden tanımlamanır ve takip edilir. Bu üç adım, çizgi roman panelini bulma, bileşenlerini tanımlama ve karakter kimliklerini diyalog benzeri bir yapıya dönüştürme imkanı sağlamaktadır. Dolayısıyla tezin son adımında ise, önceki yapının faydalarından yararlanmak için multimodal *ComicBERT* modelini geliştirdim.

ComicBERT'in içerik anlama yeteneklerini değerlendirmek için cloze tarzı görevleri kullandım. Ayrıca, *Scene-Cloze* adını verdiğim yeni bir görev öneriyorum. Sonuç olarak, yaklaşımım, özellikle metin ve visual cloze görevlerinde %69.5 ve %77.1'lik doğruluklar elde ederek insan seviyesine yaklaşıyor. Genel olarak, katkılarım şunlardır:

1. *COMICS Text+ Dataset* adını taşıyan iki milyondan fazla konuşma balonu ve anlatı kutusunun transkriptini içeren bir veri kümesi oluşturuldu ve paylaşıldı. Ayrıca, metin algılama ve tanıma modellerini açık kaynak olarak eğitimlerinde kullandığım etiketli veri setleriyle birlikte paylaşıldı.

2. Algılama, segmentasyon ve ilişkilendirme görevleri için bir *MTL model'ini* her yönüyle daha iyi hale getirerek, çizgi roman karakter yüzü ve vücudu ile konuşma balonu ilişkilendirme görevinde SOTA sonuçlar elde edildi.

3. Birleşik ve kimlik uyumlu çizgi roman karakter özellik vektörleri ve kimlik temsilleri üretmek için *Çizgi Roman Karakterinin Yeniden Tanımlanması için Kimliğe Duyarlı Yarı Denetimli Öğrenme* yapısını öne sürüldü. Ayrıca, öz denetim aşamasında kullanılan veri kümesini, *Comic Character Instances Dataset*, oluşturdum ve yarı izleme aşamasında kullanılan dörtlü ardışık çizgi roman panellerinin içindeki kimlik etiketlerini içeren *Comic Sequence Identity Dataset*'i derlendi.

4. Sıralı panelleri ve bileşenlerini işleyebilen bir transformer-kodlayıcı mimarisi olan multimodal *Comicsformer* tanıtıldı. Çizgi romanlar için yeni, kendi kendini denetleyen bir ön eğitim stratejisi olan *Masked Comic Modeling (MCM)* görevi için omurga görevi görür ve sonuçta çizgi romanlar için potansiyel bir *Foundation* model olan *ComicBERT* ortaya çıkar. ComicBERT, cloze tarzı görevlerde, özellikle metin cloze ve görsel cloze görevlerinde, insan düzeyinde kavramaya yaklaşan SOTA performansına ulaşmıştır.

ACKNOWLEDGMENTS

It has been exactly three years since I started my master's journey. Each phase of this journey can be considered as its own saga. From universal pandemics to national challenges, I have unfortunately experienced many things. However, I have had the opportunity to develop myself in the domain of modern artificial intelligence to the fullest extent, thanks to my professors, mentors, colleagues, and the KUACC HPC Cluster. Hence, this section will be a bit lengthy :)

First and foremost, I would like to express my endless gratitude to the Intelligent User Interfaces (IUI) Lab members. Special thanks go to my incredible teammates, Barış Batuhan Topal and Alara Zindancıoğlu. Our discussions were always highly creative and productive. Additionally, the time spent with my peers from the KUIS AI and Inzva communities and the NLP reading group was truly enlightening. I also extend my appreciation to the physical manifestation of the IUI Lab, ENG 266, which always felt like a second home.

Beyond academia, there are some individuals whose unwavering support and presence have been instrumental. In no particular order, I want to acknowledge Berke Ataç, Emre Ekinci, Sertaç Aksoy, Sabri Ayeş, Yiğit Tekşen, Emin Ayar, and the entire Naylalabs team, Oğuzhan Erarslan, İpek Cerrahoğlu, and Sarp Başaraner, your closeness has meant the world to me, and I offer my heartfelt gratitude. I must also take a moment to express my gratitude to my family, my partner Gabrielle Reeves, and my beloved daughter Lolita Hanım. Without them, I would not be writing these words today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SOTA	state-of-the-art
GT	Ground Truth
AP	Average Precision
MSE	Mean Squared Error
FP & FN	False Positive & False Negative
UI	User Interface
SGD	Stochastic Gradient Descend
M109	Manga 109 Dataset
DCM	DCM 772 Dataset
eBD	eBDtheque Dataset
DB	Differentiable Binarization
DCN	Deformable Convolutional Networks
PAN	Pixel Aggregation Network
PSENet	Progressive Scale Expansion Network
CRNN	Convolutional Recurrent Neural Network
SAR	Show-Attend-and-Read
MAP	Mean Average Precision
MAP@R	Mean Average Precision at R
MRR	Mean Reciprocal Rank
P@1	Precision at 1
R-P	R-Precision
SSL	Self-supervised learning

Chapter 1

INTRODUCTION

Comics is a multimodal structure and medium that uses writing and images in a spectrum, constituting only images or writings, mostly sequentially, to convey a story or an idea. Comics is at the intersection of education [Herbst et al., 2011], sociocultural studies [Brienza, 2010], linguistics and cognitive sciences [Laubrock and Dunst, 2020, Augereau et al., 2017]. Understanding comics enables progress in all those areas. Moreover, methods used for processing and understanding can be translated and used on forms that are propagated with *Visual Language* including but not limited to cave paintings, mangas, graphic novels, Franco-Belgian comics called "Bande dessinée" and so on [Cohn, 2013].

The unique structure of comics can be analyzed at low-level and high-level. On a low-level, natural language processing techniques can be utilized to extract texts of dialogues, narratives, and onomatopeias. Computer vision methods can be used to detect and segment motion lines, characters, and location of objects and components [Nguyen et al., 2019, Rigaud et al., 2015a, Hartel and Dunst, 2021, Nguyen et al., 2018]. As for high- level, relations between characters, storytelling, narrative understanding, inter-panel and intra-panel events must be studied. Although high-level features of natural images and scenes are widely studied [Minaee et al., 2021], the same cannot be pointed out for comics. That is due to the great variety of comics across time, location, different styles, and lack of annotated data (see Table 2.1 for an overview of datasets). Limited datasets on the domain can also be explained by the challenges of copyrighting and access rights.

In this thesis, a trajectory from low-level to high-level analysis will be discerned. The thesis is primarily composed of four interconnected steps. Each step constitutes

a coherent study on its own while collectively forming a meaningful whole. The primary data source for this thesis consists of comic books from the golden age of comics. This choice is made due to the availability of around 4000 series and their pages from the COMICS dataset [Iyyer et al., 2017], which is free from copyright and related issues and it is the largest comics corpus.

The first part of the thesis focuses on text extraction. As mentioned earlier, there is a lack of a comprehensive and accurate dataset. Although COMICS provides text data, the evaluation of extracted text quality was not shared. Hence, I aimed to define a gold standard for text extraction in comic books. Within this scope, I conducted benchmarking with various text detection and recognition models. During this process, utilizing speech bubbles and narrative box images, I developed an end-to-end OCR pipeline encompassing text detection and text recognition. This endeavor followed the principles of the machine learning life cycle, including data collection and continuous model training and evaluation steps. Quantitatively, I curated the most accurate and comprehensive dataset for the golden age of comics, named *Comics Text+* in terms of speech bubble and narrative texts. Along this trajectory, I also shared the datasets collected for text detection and recognition, as well as the models trained with this data. Moreover, the models derived from this pipeline were employed in the subsequent phases of the thesis.

In the second phase of the thesis, a transition from low-level tasks to high-level tasks transition happened with the speech bubble to character association task. In this context, the *COMIC MTL* [Nguyen et al., 2019] framework served as the foundation. As implied by its name, COMIC MTL is a multi-task learning framework that simultaneously optimizes three core tasks: detection, segmentation, and association. These encompass the detection of panels, speech bubbles, narrative boxes, character faces, and bodies; the segmentation of speech bubbles and panels; and the association between character faces and bodies with speech bubbles. In this stage of the thesis, I built upon the COMIC MTL structure and introduced improvements at the architectural level, employed new sampling methods, optimized the pair-pool process, proposed a novel masking strategy for the relation branch, and refined post-processing methods. As a result, I achieved SOTA results in comic analysis,

particularly for tasks involving the association of character faces and bodies with speech bubbles. This phase of the thesis can be perceived as a foundational element or glue that binds the entirety of the thesis together. The speech bubble segmentations obtained here play a vital role in assisting text extraction in conjunction with the OCR models developed in the first phase. Furthermore, all the detected components contribute significantly to the sequential processing of comic panels and other components, enhancing the thesis's overall coherence and impact.

In the preceding two sections, I establish the following capabilities: the ability to detect character faces and bodies, the ability to identify character instances from pairs of faces and bodies using rule-based methods, speech bubble segmentation, text extraction from speech bubbles, and the capacity to associate character instances with speech bubbles. In the third part of the thesis, I continue along this line of reasoning, directing my focus towards determining the identities of characters. Within this scope, I address the comic character reidentification task. Given the challenge of a lack of annotated data, I approach this task in the form of semi-supervised learning. In doing so, I propose a novel *Identity-Aware Self-Supervision Framework*, which simultaneously processes both characters' faces and bodies, utilizing information about face-body correspondence and formulating it as *Identity-Awareness Loss*. This approach enables the encoding of identity information on both face and body. This is important since not every comic character may possess a detectable face and body due to occlusion, pose variation, or artistic factors such as close-up drawings. Following this by employing the self-supervised model as a backbone and fine-tuning it with a linear projector, semi-supervision was completed. The data used during fine-tuning is shared as the *Comic Sequence Identity Dataset*, along with the data collection tool. While deriving unified and identity-aligned character embeddings from the self-supervised backbone, Obtaining identity representations for character instances from the fine-tuned model was made possible. I enable comic character reidentification across panels by utilizing clustering techniques on these identity representations of characters of sequential comic panels.

For the final part of the thesis, I introduce, for the first time, a multimodal transformer-based foundational model for comics that possesses the ability to process

sequential panels alongside their constituents: characters, associated text, narrative text, and panel images. This transformer architecture, the *Comicsformer*, stands as a vanilla transformer encoder distinguished by its input and input encoding processes making use of all the preceding components.

Utilizing this architecture, I present a novel self-supervised pre-training strategy termed as *Masked Comic Modeling (MCM)*. Within this strategy, the objective is to maximize agreement between the masked outputs of the Comicsformer and its corresponding panel units' input embedding. The application of MCM-driven pre-training to the Comicsformer architecture creates *ComicBERT*, inspired by BERT [Devlin et al., 2019]. It is important to note that ComicBERT is the aforementioned foundation model for the golden age of comics. Also, it can potentially be the foundation model for all comics and similar mediums when pre-trained with the appropriate data. The efficacy of ComicBERT was measured by its performance in downstream tasks using cloze-style tasks. I achieved SOTA results, particularly for text-cloze and visual-cloze tasks [Iyyer et al., 2017], predicting a panel's speech text or image given previous n panels as context, thus approaching human-baseline performance. This demonstrates the model's ability to comprehend comic contexts, paving the way for future research in this domain.

In addition to the four sections, there is also a chapter in which I evaluate the recent and highly popular large language models (LLMs), which have revolutionized numerous NLP tasks, specifically focusing on the text-cloze task.

Chapter 2

A GOLD STANDARD AND BENCHMARK FOR COMICS OCR

2.1 Introduction

The amount of annotated data creates a bottleneck for studies on comics by deep learning methods. Although there have been works including panel, speech bubble detection, and segmentation [Nguyen et al., 2019, Rigaud et al., 2015a, Hartel and Dunst, 2021, Nguyen et al., 2018], onomatopeia detection and recognition [Baek et al., 2022], a single extensive dataset with OCR extracted comics texts, called COMICS [Iyyer et al., 2017] exists. When the text quality of the COMICS dataset is measured over a sample, it can be seen as unreliable (see Subsection 2.3.1). There is limited study on the OCR of dialogue and narrative texts to produce high-quality text data. In [Hartel and Dunst, 2021], such a pipeline is presented. However, they cannot share their data and models due to copyright issues.

This study presents a complete pipeline and holistic method for OCR of comics to extract text from speech bubbles of panels. While doing this, the most extensive dataset available is used, COMICS [Iyyer et al., 2017], to curate a reliable and comprehensive dataset of American golden age comics as they are uniform, and work that has been done can be improved with more accurate data. I first observe the reported and unreported problems in the COMICS dataset [Iyyer et al., 2017]. Separate solutions for text detection and text recognition for comics and created an end-to-end system using the MMOCR toolbox[Kuang et al., 2021] is then developed. In this process, I continuously annotate panel and speech bubbles from the COMICS dataset, using the annotation tool that is extended on LabelMe [Wada, 2018], train the models, and evaluate the ground-truth data that I create and share. When the model performance is improved and saturated enough, the best end-to-end pair is

chosen to process the COMICS dataset's speech bubbles and extract the OCR texts. As a final step, we post-process the OCR data to finalize the *COMICS TEXT+* dataset. My contributions can be highlighted as follows:

- I released a substantially refined version of the COMICS dataset [Iyyer et al., 2017] OCR results called *COMICS Text+* that outperformed the text quality of its predecessor by high margins. It is the best quality and the most comprehensive dataset, with over two million transcriptions of textboxes publicly shared.

- I released publicly available text detection and text recognition datasets created from Golden Age comics called respectively, *COMICS Text+: Text Detection Dataset* and *COMICS Text+: Text Recognition Dataset* along with ground truth data to validate end-to-end OCR pipelines.

- I trained and benchmarked the performance of 14 text detection and ten text recognition models provided by MMOCR toolbox [Kuang et al., 2021] on COMICS Text+: text detection & text Recognition datasets. An analysis of the effects of training size on performance was done. It was shown that the bottleneck for the OCR studies for the comics is text detection models. However, it was also shown that after 200 textbox annotations, their performance dramatically improved.

- I provided publicly available and fine-tuned text detection, FCENet [Zhu et al., 2021] based, and text recognition, MASTER [Lu et al., 2021] based, models trained on COMICS Text+: text detection & text recognition datasets. There were no such specific models and datasets, so I hoped this work would be the baseline for future studies and lower the entry barrier to the domain.

- I shared an annotation tool forked from LabelMe [Wada, 2018] that enabled the use of text detection and text recognition models to aid and speed up the annotation process and converted annotations into text detection and recogni-

tion datasets that were used to create COMICS Text+: text detection & text Recognition datasets.

- The comics processing backbone proposed with the COMICS dataset was reproduced. The reproduced model was trained with both COMICS and *COMICS TEXT+*, and their test results were compared. The model trained with *COMICS TEXT+* outperformed the other and achieved SOTA results on most cloze-style tasks, proving the increased quality of the text data.

2.2 Related Work

2.2.1 Text Detection

Deep learning has recently taken over the scene text detection area. According to the granularity of prediction results, scene text detection techniques with deep learning can be divided into three types: regression-based, part-based, and segmentation-based approaches [Liao et al., 2022].

Regression-based approaches are a set of models that directly predict the bounding box coordinates of each instance. These approaches typically use relatively simple post-processing techniques (e.g., non-maximum suppression) to generate final results. However, they often suffer from inaccuracies when dealing with irregularly shaped objects, such as curved ones [Liao et al., 2017, Liao et al., 2018, Zhu et al., 2021, Xie et al., 2019].

Part-based methods detect the small parts of texts (e.g., words) and then link these parts together to form larger units (e.g., sentences). They are good at detecting long texts, but their linking algorithms are quite complicated with hand-crafted parameters. Therefore, tuning them is quite difficult [Shi et al., 2017, Tang et al., 2019a].

Finally, segmentation-based methods make predictions on the pixel level to later combine them with component-grouping paradigms as post-processing algorithms. After the post-processing, mask output or bounding boxes can be obtained [Liao et al., 2022, Wang et al., 2019a, Lyu et al., 2018].

2.2.2 Text Recognition

A text recognition system converts a text instance image into a string sequence. Scene text recognition systems are a type of text recognizer in which text instance images come from natural scenes. Scene text recognition differs from document-level OCR systems because images, by their nature, make processing difficult. They have complex backgrounds, varying fonts, changing colors, and worse imaging conditions [Chen et al., 2021]. These issues are very similar to the challenges of the domain of the comic. Hence, I mostly use scene text recognition systems in this thesis.

Similar to other branches of computer vision, text recognition systems have witnessed a shift from processing handcrafted features to deep-learning-based models. In ABINet [Fang et al., 2021], vision and explicitly defined language models are combined for scene text recognition. The language model acts as a simulator for the cloze test, and an iterative correction strategy is applied to the visual model to get the best results. Another scene text recognizer that was used in this thesis is MASTER [Lu et al., 2021], which is capable of learning feature-feature and target-target relationships inside its encoder-decoder structure, resulting in better intermediate representations.

2.2.3 OCR for Comics

A preliminary work of [Rigaud et al., 2016] challenges of speech text recognition for comics are laid out, and a couple of approaches to tackle those are implemented. They conclude that generic OCR systems better recognize typewritten-like text, whereas specifically trained OCR models can perform better for skewed, uppercase, curvasive fonts. However, in this thesis, I show that specifically trained models can outperform generic OCR systems. That is a sign of how far the domain has progressed since then.

In [Gobbo and Matuk Herrera, 2020], pixel-level text annotations are curated for unconstrained text detection for the manga, and special evaluation metrics are measured benefiting from that. Following this, a more recent study, Comic Onomatopoeia Dataset (COO) [Baek et al., 2022] presents an onomatopoeia, textual

representations of sounds, states or objects, dataset for text detection, text recognition and link prediction tasks. COO provides onomatopoeia in Japanese to further push the limits of current irregular text detection and text recognition models. In contrast to my motivation of providing an extensive dataset for comics, they focus more on the performance of models for their tasks. However, this work is very compatible with my thesis. In that, the combined research on both datasets can fully capture the text modality of comics.

Hartel, Rita and Dunst, Alexander [Hartel and Dunst, 2021] constructed an OCR pipeline for comics that starts with segmenting comics page components such as panels and speech bubbles to extracting text from those components. On top, they analyzed textual properties of their OCR results concerning certain aspects such as genre affiliation and page length. This part of my thesis differs from theirs in using SOTA deep-learning-based text detection and text recognition models for OCR instead of open-source software Calamari [Wick et al., 2020]. Moreover, I publicly share the fine-tuned models, datasets, and OCR results for more than 2 million textboxes whereas their data can not be shared publicly due to copyright restrictions.

2.2.4 Datasets

Even though there are not many datasets available for the domain of comics due to copyright difficulties, only a few datasets have been made publicly available. Table 2.1 shows the complete list of datasets focused on comic books and manga based on my literature search.

The Graphic Narrative Corpus (GNC) [Dunst et al., 2017] is a unique dataset for both computational and social sciences. The dataset provides a variety of metadata, such as genre, author, and geographic origins, along with annotated data to the extent of eye gaze data. However, Unfortunately, the dataset can not be accessed publicly.

COMICS [Iyyer et al., 2017] is the dataset with the most comic books and is the primary dataset I use for this thesis. They also come up with downstream for the

domain of the comic. These are text cloze, visual cloze, and character coherence.

The Manga109 [Aizawa et al., 2020] dataset contains 109 volumes of manga by 93 different authors. These mangas were released from the 1970s to the 2010s and fall into 12 genres, such as fantasy, humor, and sports.

DCM772 [Nguyen et al., 2018] is one of the few publicly available datasets because of how it is collected. The dataset's comic books come from a free public domain collection called Digital Comic Museum [1]. This dataset annotates different character types, such as animal-like or human-like.

Comic Onomatopoeia (COO) [Baek et al., 2022] is a dataset to extend the Manga109 dataset with onomatopeia annotations. Since the COO dataset focuses on onomatopoeia, they provide a challenging text detection and recognition dataset for arbitrary text with various shapes, styles, and extreme curves. Furthermore, they introduce a novel task to link truncated texts.

eBDtheque [Guérin et al., 2013] is one the first and pioneering comic book datasets that involve American comics, mangas, and Franco-Belgian comic book albums. It provides panels, balloons, text lines, and pages.

Multimodal Emotion Recognition on Comics Scenes (EmoRecCom) [Nguyen et al., 2021b] contains a small subset of the COMICS dataset with their scene emotion labels of eight different emotions. Since its panels are coming from the COMICS dataset, it might be possible to improve the text quality of this dataset with our COMICS Text+ dataset.

2.3 Methodology

My approach consists of three main stages, transitions between each of which and steps are shown by green arrows in Figure 2.1. In the first stage, the shortcomings and the issues of text data in the COMICS dataset [Iyyer et al., 2017] were detected, and I developed the annotation tool for creating a brand new dataset by forking [Wada, 2018]. My primary motivation for doing so is that no metrics were provided in the works that proposed the dataset, [Iyyer et al., 2017] and [Iyyer, 2017]. In

[1] `http://digitalcomicmuseum.com`

Table 2.1: Overview of datasets in the comic domain.

Dataset Name	**Comic Book Release Dates**	**Top Level Item Count**	**Annotation Types**	**Public Availability**
The Graphic Narrative Corpus (GNC) [Dunst et al., 2017]	1975 - 2018	240 Comic Books	panel, balloon, caption, diegetic text, text, character, object	NOT AVAILABLE
COMICS [Iyyer et al., 2017]	1938 - 1956	3948 Comic Books	panel, balloon, captions, text	AVAILABLE
Manga109 [Aizawa et al., 2020]	1970 - 2010	109 Mangas	panel, balloon, caption, onomatopoeia, text, character, face, body	UPON REQUEST
DCM772 [Nguyen et al., 2018]	1938 - 1956	27 Comic Books	panel, balloon, caption, character, face, body, type of character	AVAILABLE
Comic Onomatopoeia(COO) [Baek et al., 2022]	1970 - 2010	109 Mangas	onomatopoeia, text	UPON REQUEST
eBDtheque [Guérin et al., 2013]	1905 - 2012	100 Pages	panel, balloon, text	UPON REQUEST
Multimodal Emotion Recognition on Comics scenes (EmoRecCom) [Nguyen et al., 2021b]	1938 - 1956	8258 Panels	multiple emotion labels for panels	AVAILABLE

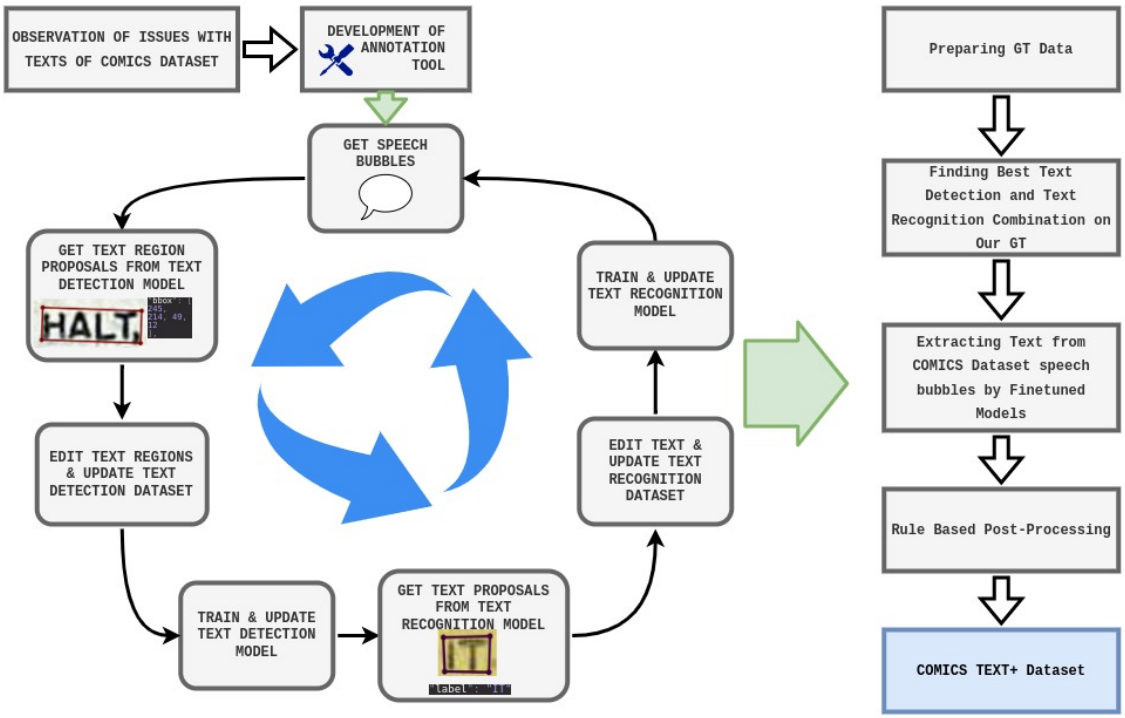


Figure 2.1: The complete process and pipeline of the proposed approach and dataset creation pipeline. Green arrows represent phase transitions. In the first phase, the OCR annotation tool was developed; in the second, text detection and recognition models were fine-tuned in parallel with data generation. In the final phase, text detection and recognition model pairs are evaluated on ground truth to find the best pair.

the second stage, panels and mostly speech bubbles cropped out pages of comics provided in the COMICS dataset are used to initiate the machine learning lifecycle. During this stage, I annotated data and trained out-of-the-box models successively until the model performance was stable. At the end of this stage, I created text detection and text recognition datasets using more than a thousand speech bubbles (see Table 2.3). In the third stage, I used the text detection and text recognition datasets to train text detection and text recognition models provided in MMOCR [Kuang et al., 2021]. As a result, I benchmarked their final performance on the dataset. I then used top-performing models to find the detection and recognition model combination that performs best on GT. GT was formed with carefully selected speech bubbles to measure the peak model performance. In the last phases of the third stage, the selected model combination is run over the COMICS dataset speech bubbles. Finally, the resultant data is post-processed to create *COMICS Text+ Dataset*.

2.3.1 Textual Flaws of COMICS Dataset

For the COMICS dataset, it is stated in [Iyyer et al., 2017] and [Iyyer, 2017] that the OCR data has detection and recognition flaws which were tried to be mitigated by post-processing. Those flaws are as follows:

- Detection of short words
- Detection of punctuation marks
- Recognition of the first letter of the word

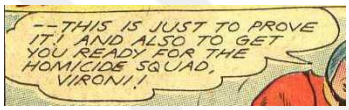
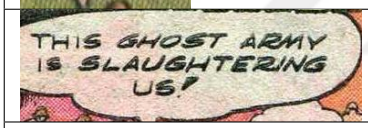
In addition to those cases, I realized there are more issues with the dataset. Such as:

- Skipping lines
- Total misrecognition words

- Not being able to recognize some fonts, e.g., italic fonts

In Table 2.2, the examples for all of the errors can be seen. Quantitative evaluation results of COMICS dataset OCR with respect to GT can be seen in Table 2.12. There, it can be seen that the character recall of the COMICS dataset OCR is lower than my final results. That backs up our qualitative analysis.

Table 2.2: Results of qualitative error analysis on COMICS Dataset OCR data.

Speech Bubble	COMICS Dataset OCR	Ground Truth
	/ 7 ./ anoa ¡ s 70 get you ready for 7 % e - vro // / yo	- - this is just to prove it ! and also to get you ready for the homicide squad , vironi !
	ok ,	click !
	is slaughter ,	this ghost army is slaughtering us !
	my oww wae , mamma , comantted	" my own wife , hannah , committed suicide ! i never understood that . . . "
	a sameersalaltano	in a flash , quick ! a somersault and
	that black id ride like of on to plummer out mule	i ' d like to ride plummer out of town on that black mule

2.3.2 Annotation Tool & Creation of New Comics Text Detection & Text Recognition Datasets

The annotation tool is a modified version of the graphical image annotation tool called LabelMe [Wada, 2018]. On top of its existing features, I added two more

modes, *Detect Text* and *Detect and Recognize Text*, that help to get inference from text detection and text recognition models to speed up the annotation process. As I fine-tuned the models with more and more data, this functionality eased and shortened annotation time.

This tool is integral to this part of the thesis. It helps the creation of new text detection and text recognition datasets and, thus, the training of text detection and text recognition models. The whole cyclic process of the tool being used is as follows:

- Randomly cropping speech bubbles or panels, at least a couple of images from a comic book, from the COMICS dataset
- Annotating for text detection by getting text region proposals from a pre-trained model
- Editing proposed text regions
- Training the text detection model, during the annotation process, I used DB-Net++ [Liao et al., 2022] and updated the text detection model for the text recognition pipeline
- Annotating for text recognition by getting text region and text proposals from the pre-trained text detection and text recognition models
- Editing proposed text regions and texts
- Training the text recognition model, during the annotation process, I used NRTR [Sheng et al., 2019] and updated the text recognition model for the text recognition pipeline

After each cycle, I measured the performance of the models with the respective training sets and GT. Results of that help to be aware of how annotations were done during the process and sometimes lead to changes in annotation policy (see Subsection 2.3.2).

Detection Mode

Figure 2.2: Text detection mode of the OCR annotation tool. A text detection model can predict text regions, words, and punctuations. Regions can be edited on demand if necessary.

In detection mode, given an image, polygons of the text regions can be inferred from a text detection model. Then the user can edit, add, or remove those polygons to create finely detailed annotations. In this mode, I created *COMICS Text+: Text Detection Dataset*.

Recognition Mode

Given an image, polygons of the text regions and their corresponding text can be inferred from a pipeline of text detection and text recognition models in recognition mode. Then the user can edit, add, or remove those polygons and texts to create finely detailed annotations. In this mode, I created *COMICS Text+: Text Recognition Dataset*.

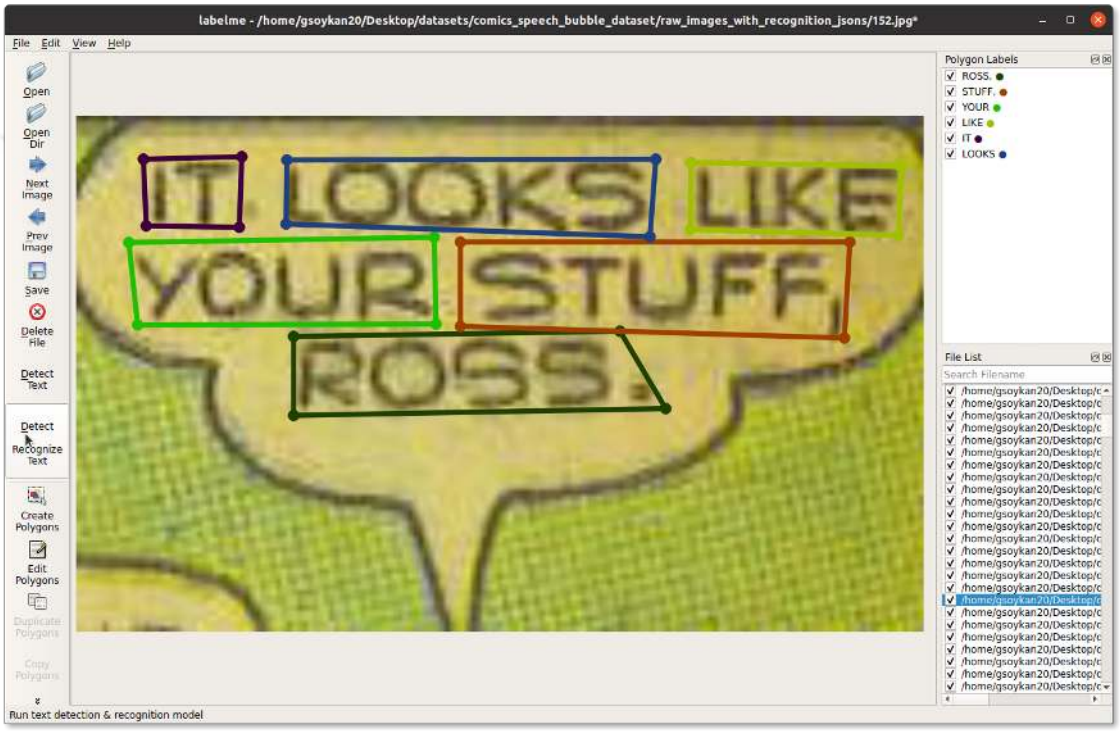


Figure 2.3: Text recognition mode of the OCR annotation tool. A text recognition model can infer recognition results. Outputs can be corrected if the results are wrong.

Conversion to Dataset Format

Once all the annotations are done, those can be exported in dataset formats with the conversions scripts in the pipeline, shared with this repository.

- For detection dataset format is *IcdarDataset* [Karatzas et al., 2015].
- For recognition dataset format is *OCRDataset* [2].

COMICS Text+: Text Detection & Recognition Datasets

COMICS Text+: Text Detection and *COMICS Text+: Text Recognition* datasets are created using the annotation tool and pipeline. The text detection dataset comprises annotated text regions in speech bubbles or panels. Those text regions are, due to the nature of the domain of the comics, mostly words. However, deciding the boundaries of text regions is challenging in this domain. My policies are shaped along with iteration cycles. Initially, I annotated text regions with high margins and included punctuation marks within a text region that contained a word. Because of that, models trained on data like this produced erroneous predictions of text regions that are often intersecting. Thus, my final policy for text detection annotation involves a low margin for words and exclusion of punctuation marks if they are not close enough to words, meaning their distance from the words can be at a maximum of two times of letter spacing if they are more distant than that they are annotated as another text region. Another factor that increases the complexity of annotations is that every comic book has different spacing between words and hyphens. I followed the same strategy with the hyphens as in the punctuation marks.

As for the text recognition dataset, I annotated text regions with their corresponding texts. However, here, since the important thing is recognizing text within the text regions, I carefully select regions with explicit texts. The challenges of text detection do not apply in this context. Another difference between the two datasets is that sometimes some text regions were skipped within an image because of their irregular text shapes or quality. Namely, they can be obscure or worn, or occluded.

[2] `https://mmocr.readthedocs.io/en/latest/basic_concepts/datasets.html`

Table 2.3: Statistics describing splits of dataset sizes constituting *COMICS Text+: Text Detection & Text Recognition Datasets*. The dataset is used to train and evaluate text detection and recognition models. In the experiments, the training set size can be different, but the test split is always used as it is.

	Training		**Validation**		**Test**	
	# Images	# Annotations	# Images	# Annotations	# Images	# Annotations
Text Detection	1012	18494	50	790	50	986
Text Recognition	906	15635	50	759	50	714

Since I cannot skip those text regions in the text detection dataset, text detection and text recognition datasets can have different images and annotations. The final statistics of the text detection and text recognition datasets can be seen in Table 2.3

2.3.3 Creating Text Ground Truth Data From COMICS Dataset

To measure the full performance of fine-tuned text detection and text recognition models, GT is needed. I created and organized GT data since no such data has been available for Golden Age comics. GT data combine tuples of a speech bubble or narrative boxes and corresponding texts. This differs from text detection and recognition datasets because it does not involve labels for text regions but covers all the text in the image. The speech bubbles for GT were carefully selected.

The policy for selecting speech bubbles was that all texts in the image should be utterly visible and read clearly. Selected images come from different comic books and different artistic styles. In some examples, character residues may be on the margin, but it is acceptable if it does not show the whole character. Thus, I can observe how the text detector behaves in residual texts. They should not be proposed as text regions. However, there are no half words or full characters to not bring irregular text and speech bubble detector problems to the general evaluation so that the models' performance can be truly measured without noisy data. Figure 2.4 shows some examples of the selected GT.

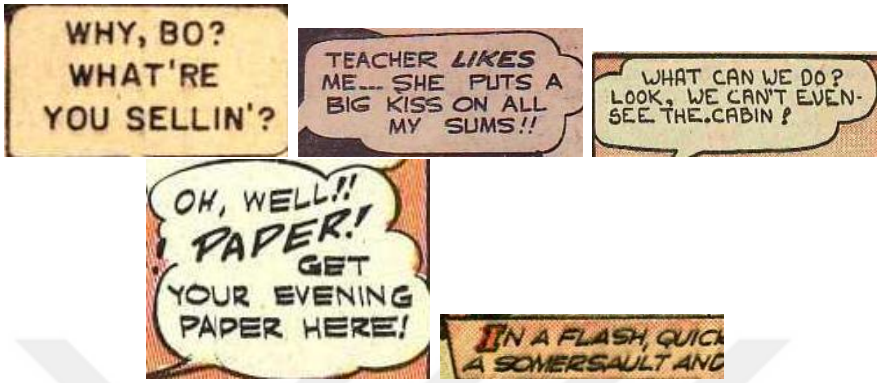


Figure 2.4: Five examples from selected speech bubbles for ground truth data.

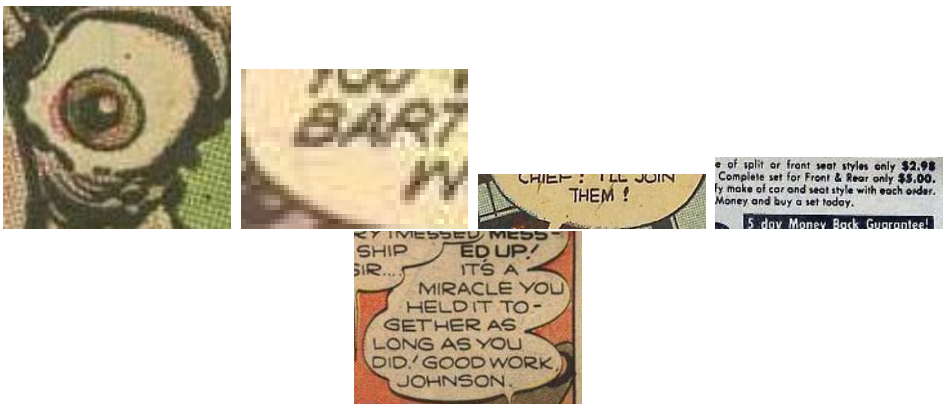


Figure 2.5: Five examples of speech bubbles that are not eligible for ground truth data.

The policy of selecting speech bubbles or narrative boxes was begun by randomly cropping multiple images from different comic series using the coordinates of text boxes provided in the COMICS dataset. I then eliminated some portions to refine the ground truth (GT). Figure 2.5 shows some examples of my filtered data from GT. I did not include images that had one or more of the following features:

- Containing partial words or texts
- Parts of the advertisement pages
- Do not contain any text
- Containing multiple panels
- Written in very irregular and specialized fonts

2.3.4 Best Detection - Recognition Models

A crucial step in the pipeline was finding the best-performing models on the GT. To do that, I trained fourteen text detection and ten text recognition models in the MMOCR toolkit [Kuang et al., 2021] and benchmarked them. After finding the best-performing models on the test split, the top three models on GT were evaluated to determine which combination of models gives the best results.

Detection Model Benchmark

I benchmarked recently published and performant fourteen pre-trained text detection models on our *COMICS Text+: Text Detection Dataset*. The results are shown in Table 2.4. The models are as follows:

- **`DB_r18`:** DBNet [Liao et al., 2020] with ResNet-18 [He et al., 2016] backbone pretrained on ICDAR2015 [Karatzas et al., 2015] dataset
- **`DB_r50`:** DBNet with ResNet-50 backbone pretrained on ICDAR2015 dataset

- **DBPP_r50:** DBNet++ [Liao et al., 2022] with ResNet-50 backbone pretrained on ICDAR2015 dataset
- **DRRG:** Deep Relational Reasoning Graph Network [Zhang et al., 2020] pretrained on CTW-1500 dataset [Yuliang et al., 2017]
- **FCE_IC15:** FCENet [Zhu et al., 2021] with ResNet-50 backbone pretrained on ICDAR2015 dataset
- **FCE_CTW_DCNv2:** FCENet [Zhu et al., 2021] with ResNet-50 and deformable convolutional networks (DCN) [Dai et al., 2017] backbone pretrained on CTW-1500 dataset
- **MaskRCNN_IC17 :** Mask R-CNN [He et al., 2017] pretrained on ICDAR2017 [Gomez et al., 2017] dataset
- **MaskRCNN_IC15:** Mask R-CNN pretrained on ICDAR2015
- **MaskRCNN_CTW:** Mask R-CNN pretrained on CTW-1500 dataset
- **PANet_CTW:** Pixel Aggregation Network (PAN) [Wang et al., 2019b] pretrained on CTW-1500 dataset
- **PANet_IC15:** PAN pretrained on ICDAR2015
- **PS_IC15:** Progressive Scale Expansion Network [Wang et al., 2019a] pretrained on ICDAR2015, (PSENet)
- **PS_CTW:** PSENet pretrained on CTW-1500 dataset
- **TextSnake:** TextSnake [Long et al., 2018] pretrained on CTW-1500 dataset

Table 2.4: Best text detection model results trained on the *COMICS Text+: Text Detection Dataset* sorted by hmean.

Text Detection Model	**Recall**	**Precision**	**Hmean**
DBPP_r50	93.3	97.1	95.2
FCE_CTW_DCNv2	92.9	96.7	94.8
MaskRCNN_IC17	92.5	96.8	94.6
PS_IC15	93.1	96.0	94.5
MaskRCNN_CTW	93.8	94.4	94.1
MaskRCNN_IC15	92.6	94.4	93.5
DB_r50	91.1	94.5	92.8
PS_CTW	91.6	93.9	92.7
DB_r18	89.7	95.5	92.5
PANet_IC15	90.3	93.9	92.0
TextSnake	85.9	94.8	90.2
FCE_IC15	90.2	89.3	89.8
DRRG	85.1	94.7	89.6
PANet_CTW	84.2	92.6	88.2

Training Details At training, I used a single NVIDIA V100 GPU. The batch sizes of each training varied to maximize the use of GPU. Adam optimizer is used with a learning rate of 0.0001. During training, gradient clipping is used with a max norm value of 0.5. Step learning rate scheduler is used at the third and fourth epoch by 0.1. Models have trained six epochs, and their best is selected for evaluation on the test set in their validation step according to h-mean. All experiments are configured by MMOCR [Kuang et al., 2021] training scripts.

Recognition Model Benchmark

I benchmarked recently published and performant fourteen pre-trained text recognition models on our *COMICS Text+: Text Recognition Dataset*. The results are shown in Table 2.5. The models are as follows:

- **ABINet:** ABINet [Fang et al., 2021] pretrained on Synth90k [Jaderberg et al., 2016] and SynthText [Gupta et al., 2016]
- **CRNN:** Convolutional Recurrent Neural Network (CRNN) [Shi et al., 2016a] pretrained on Synth90k
- **MASTER:** MASTER [Lu et al., 2021] pretrained on Synth90k, SynthText, and SynthAdd [Li et al., 2019]
- **NRTR_1/8-1/4:** NRTR [Sheng et al., 2019] with the height of feature from the backbone is 1/16 of the input image, where 1/8 for width and pretrained on Synth90k and SynthText
- **NRTR_1/16-1/8:** NRTR with the height of feature from the backbone is 1/8 of the input image, where 1/4 for width and pretrained on Synth90k and SynthText
- **RobustScanner:** RobustScanner [Yue et al., 2020] pretrained on multiple ICDAR datasets, Synth90k, SynthText, SynthAdd and more [3]
- **SAR:** Show-Attend-and-Read (SAR) [Li et al., 2019] pretrained on multiple ICDAR datasets, Synth90k, SynthText, SynthAdd and more [4]
- **SATRN:** Self-Attention Text Recognition Network (SATRN) [Lee et al., 2020] pretrained on Synth90k and SynthText
- **SATRN_sm:** SATRN with a reduced number of encoder and decoder layers pretrained on Synth90k and SynthText
- **CRNN-TPS:** CRNN-STN [Shi et al., 2016b] pretrained on Synth90k

[3] `https://mmocr.readthedocs.io/en/latest/textrecog_models.html#robustscanner`

[4] `https://mmocr.readthedocs.io/en/latest/textrecog_models.html#sar`

Table 2.5: Best text recognition model results trained on the *COMICS Text+: Text Recognition Dataset* sorted by 1-N.E.D.

Text Recognition Model	Char. Recall	Char. Precision	Word Acc.	Word Acc. Ignore Symbol	1-N.E.D
MASTER	99.6	99.5	95.9	98.3	99.2
NRTR_1/8-1/4	99.5	99.4	95.8	98.0	99.2
NRTR_1/16-1/8	99.4	99.2	95.1	97.5	99.2
RobustScanner	99.2	99.2	94.2	97.1	98.6
SAR	99.3	99.3	94.5	97.1	98.3
SATRN	99.1	99.2	94.2	96.5	98.1
SATRN_sm	98.7	98.8	92.9	94.9	97.7
ABINet	99.5	78.7	72.4	72.9	84.5
CRNN-TPS	98.6	78.5	70.8	71.5	84.1
CRNN	98.4	78.4	69.7	70.7	83.9

Training Details At training, I used a single NVIDIA V100 GPU. The batch sizes of each training varied to maximize the use of GPU. Adam optimizer is used with a learning rate of 0.0001. During training, gradient clipping is used with a max norm value of 0.5. Step learning rate scheduler is used at the third and fourth epoch by 0.1. Models have trained six epochs, and their best is selected for evaluation on the test set in their validation step according to *1-N.E.D.*. All experiments are configured by MMOCR [Kuang et al., 2021] training scripts.

Finding Best Text Detection and Text Recognition Models for OCR

After training on the text detection and text recognition datasets, I found that the text detection and text recognition benchmarks demonstrated close-performing models. Although their results were similar in terms of metrics, I realized that they exhibited different failure points. Due to this observation, I decided to assess the OCR performance of the top three combinations. Even though I had chosen DB-NET++ in the annotation-training-evaluation cycle based on qualitative evaluation, I subsequently discovered that the FCENET text detection backbone significantly outperformed others in terms of OCR metrics within the domain of comics (refer to

Table 2.6: Results of top-performing text detection and text recognition models trained on the *Comics Text+ Datasets* measured with respect to ground truth data.

Recognition / Detection Models	**NRTR_1/16-1/8**		**NRTR_1/8-1/4**		**MASTER**	
	Word Acc. Ignore Symbol	1 - N.E.D.	Word Acc. Ignore Symbol	1 - N.E.D.	Word Acc. Ignore Symbol	1 - N.E.D.
MaskRCNN_IC17	7.6	75.6	7.4	75.7	7.6	76.3
FCE	71.4	97.6	70.4	97.7	**72.2**	**97.8**
DB+	51.6	94.8	59.3	96.2	60.5	96.3

Table 2.6). From the results, it became clear that the critical factor for the current OCR of comics is the text detection model utilized. The OCR performance did not show substantial variations when keeping the same text detection model and changing the text recognition model. However, altering the text detection model while keeping the text recognition model constant led to significant changes in the OCR results.

Text Detection and Text Recognition and OCR Performances by Data Size

I investigated how the most successful text detection and text recognition models, based on OCR results of GT data, would yield results according to the varying number of training data. This way, how much better results can be achieved by using the minimum amount of data. Since annotating and labeling data is laborious and time-consuming, I aim to provide a baseline data count for different comic domains and future studies.

For text detection, I present the results in Figure 2.6 for precision, recall, and h-mean. The prominent finding is that since I already use a pretrained model, it adapts to the domain with as few as fifty training samples and within a few epochs. Typically, it takes hundreds of epochs for the text detection model to learn from random initialization. The results do not change significantly after the training sample count is increased to two hundred. However, as the amount of data increases, the model's performance continues to improve.

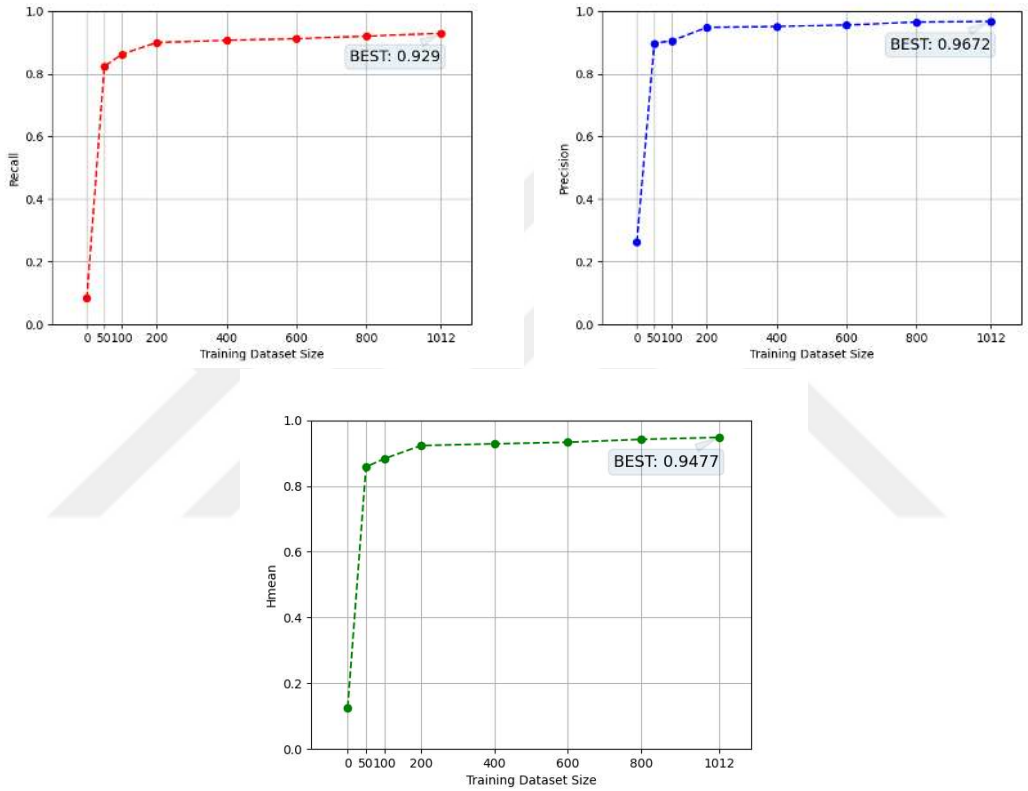


Figure 2.6: Recall, precision, and harmonic mean (hmean) graphs of the most performant text detection model, FCE_CTW_DCNv2, on the test dataset for varying training dataset size. After training with text regions from 200 speech bubbles, it can be observed that the model adapts to the domain.

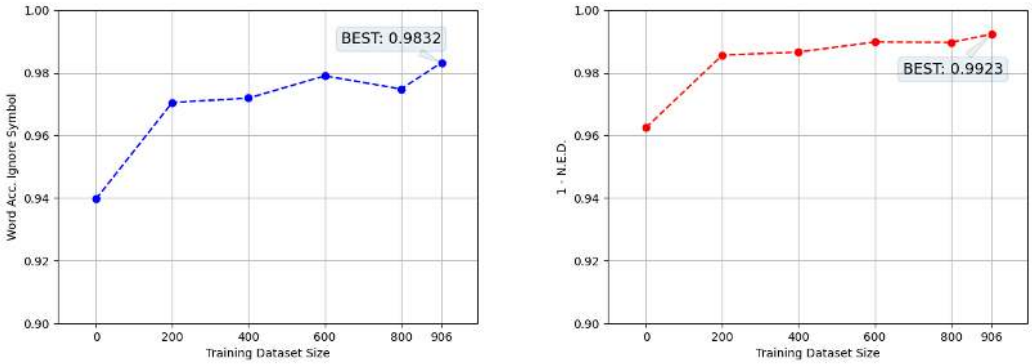


Figure 2.7: Word accuracy ignoring symbols and 1 - N.E.D. graphs of the most performant recognition model, MASTER, on the test dataset for varying training dataset size. Unlike text detection, the model performs reasonably well without being trained. However, similar to text detection, a significant improvement is observed when using data from 200 speech bubbles.

The same trend applies to text recognition (see Figure 2.7), as word accuracy is measured by ignoring symbols and one minus normalized edit distance. However, domain adaption in this context does not create a critical difference since the model performance only increases two % - three %. On the other hand, text detection increases eight - ninefold.

As the final step of performance measurement by varying dataset size, OCR performance is measured for text detection and text recognition models' combination (see Figure 2.8). On one minus normalized edit distance metric, parallel to the model's respective performance, I achieved near-top performance with two hundred training sample sizes. However, word accuracy by ignoring symbols reached that level of four hundred. This shows how sensitive the word accuracy is regarding model performance. Consequently, four hundred training samples for text detection and recognition can be a good baseline for comics.

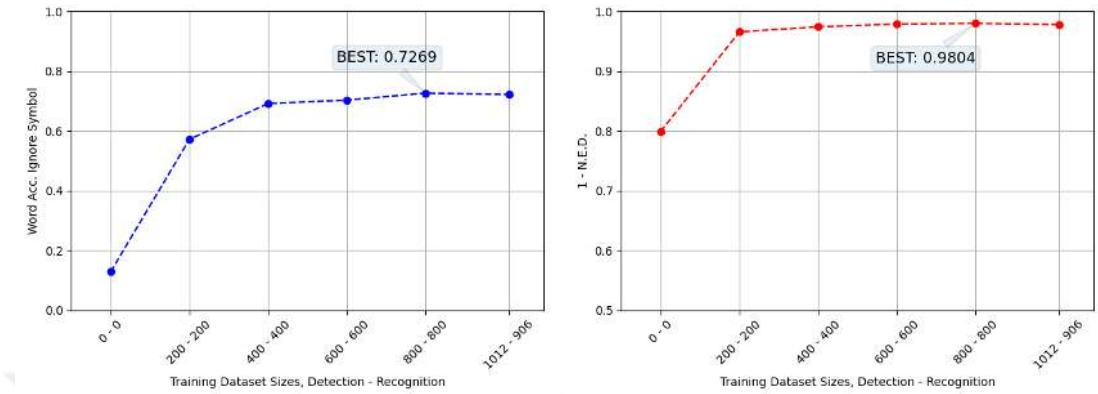


Figure 2.8: Word accuracy ignoring symbols, and 1 - N.E.D. graphs of the most performant detection and recognition model pair, FCE_CTW_DCNv2 - MASTER, on the ground truth data trained on varying training dataset size combinations.

Table 2.7: Some pairs of overlapping text boxes in the third comic series of the COMICS dataset [Iyyer et al., 2017]. The one with a smaller area within the pairs is filtered, and the other is kept. As expected, most of their characters are shared.

Page No	Panel No	Textbox No	Text	x1	y1	x2	y2
5	2	0	i quickly drag the yorn body into ...	21	17	326	165
5	2	1	i quickly drag the yorn body into ...	24	21	326	165
20	0	1	lovers in all the dark corners ...	1	0	410	132
20	0	0	lovers in all the dark corners ...	0	0	410	121
30	2	2	not know the fish had been fed	0	529	384	587
30	2	3	not know ! the fish had been fed	0	540	384	587

Table 2.8: Statistics describing the filtered textboxes that are part of the COMICS dataset [Iyyer et al., 2017] but not in COMICS Text+, by the result of empty or erroneous inferences. Most images are not textboxes, and the others that increase character and word average are misdetections of textboxes in advertisement pages.

textbox count	char. avg.	word avg.	char. median	word median
9508	6 .43	2.12	3	1

2.3.5 Creating 'COMICS Text+ Dataset'

After the best models for OCR were determined and trained, I ran them for OCR on the images, cropped from panels by the coordinates of textboxes provided in the COMICS dataset [Iyyer et al., 2017]. However, the number of final textboxes with texts in COMICS Text+ differs from the COMICS. One reason is that the advertisement panels and their textboxes and textboxes that do not have matched texts from the dataset were filtered out. Advertisement pages are provided with the COMICS dataset. The other reason is that some of my inferences resulted in erroneous or empty texts. Those were also removed from the dataset to increase the overall quality of the dataset. Statistics of filtered textboxes from such reasons can be seen in Table 2.8, and some examples can be seen in Figure 2.9. After this, 2,208,006 text boxes remained. As a final preprocessing step, I filtered textboxes that overlap. Namely, in the COMICS dataset, some textboxes have highly intersecting textboxes. Although they were low in number, it was still a factor that lowered the dataset's quality. The existence of overlapping textbox predictions also supports my claim about the textbox detector and its effect on the dataset quality. In terms of area, I removed the smaller textbox in case they have more than 0.8 Intersection over Union (IoU) within the same panel; see Table 2.7. This step reduced the textboxes to 2,188,853, 1,702,140 dialogue, and 486,713 narratives. Whereas in the COMICS dataset, the number of textboxes is 2,545,728. I believe this preprocessing filtering increases the overall quality of the dataset.

Figure 2.9: Some examples of textboxes were filtered from the *COMICS Text+ Dataset*.

Postprocessing on Raw OCR Output

Two post-processing approaches were applied to extracted texts. The first is a rule-based approach, and the second is a neural spell-checking approach. The second failed to enhance the overall quality of the texts (see Subsubsection 2.3.5).

As for the rule-based approach, I improved the process of [Iyyer, 2017] by several additional steps. In Iyyer's thesis, two types of mistakes were targeted. The first is the mistake of recognizing a word's first letter (e.g., eleportation instead of teleportation). Although it is a valid case, those types of errors are also introduced in addition to the errors of their OCR system of choice but their problems with textbox detections. The second type of mistake that they were targeted was errors of single characters. Those types of errors can be in the OCR system, but they are also the result of the aforementioned faulty textbox detection coordinates. Because of that, I extended and modified their post-processing approach by accepting that there can be errors that include the first, initial, or final letters of the word.

Algorithm 1 The algorithm to create a dictionary for replacing tokens in the OCR output with PyEnchant suggestions.

```
procedure CreateTokenReplacementDict
    corpus ← set(nltk_corpus + brown_corpus)
    for each token ∈ tokens do
        if token.length ≥ 4 and
token ∉ corpus and CheckPyEnchant(token) then
            suggestions ← GetSuggestions(token)
            for each suggestion ∈ suggestions do
                if SuggestionStartsWith(token) or
SuggestionEndsWith(token) then
                    AddTokenReplacementDict(token, suggestion)
                end if
            end for
        end if
    end for
end procedure
```

For the first kind of error, I tokenize the OCR output and create a vocabulary of tokens sorted with the number of occurrences in a descending manner. For tokenization, NLTK's Punkt Tokenizer and word tokenizer [5] are used. Then, tokens between 10,001 to 100,000 ranks, with the assumption that misspelled words are less frequent, are subjected to the Algorithm 1 to create a dictionary where I can use it to replace those tokens in the output. This algorithm differs from Iyyer's [Iyyer, 2017] approach because I check whether those tokens are parts of corpora. In addition to NLTK's default corpus specifically, I add Brown Corpus [Francis and Kucera, 1979] because historically, the Golden Age of comics and the publish date of Brown Corpus is closer than other corpora, which are 50's and 61. Another difference is that instead of checking the token difference for the first letter of PyEnchant suggestions, I check whether the suggestion starts or ends with the token. That way,

[5] `https://www.nltk.org/`

errors of texts introduced by textbox detection errors can be fixed.

Algorithm 2 The algorithm for applying rule-based post-processing to OCR output to finalize the COMICS Text+ dataset.

```
procedure ApplyPostProcessing
    valid_characters ← {a, d, i, m, s, t, x}
    for each token ∈ all_tokens do
        if not CheckIfThereIsPuncBeforeOrAfter(token) then
            token ← ReplaceTokenIfInReplacementDict(token)
            if token.length = 1 and token ∉ valid_characters then
                Remove(token)
            end if
        end if
    end for
end procedure
```

Erroneous single characters are detected whether they have punctuation marks before or after the character. If they do not have punctuation marks before or after, then I check if they are in the set of valid single characters, which are *a, d, i, m, s, t, x*. Those characters frequently occur with a high chance part of the text. I check whether there is a punctuation mark before or after the character because there might be abbreviations of special names in the text (e.g., *F.B.I.*, *U.S.A.*). Then, I apply Algorithm 2 to post-process all of our OCR. output to prepare its final form. In my dataset, I share our raw OCR output along with the post-processed version of it. 110,838 OCR output of textboxes are affected by the first part of the post-processing; namely, some of their tokens are replaced by PyEnchant [6]. 131,833 OCR output of textboxes is updated with the second part of the process, meaning some of their single-character tokens are deleted. Both of those post-processing steps impact 18,580 OCR output as their intersections. I can only qualitatively measure the effects of rule-based post-processing because only a few GT samples are affected.

[6] `http://pyenchant.github.io/pyenchant/`

Table 2.9: Evaluation of Neuspell [Jayanthi et al., 2020] neural spelling correction toolkit's `BERT Checker` with respect to COMICS Dataset OCR and COMICS Text+ Data. 273 out of 500 ground truth data was used because that much of the data was exposed and, as a result, corrected by the Neuspell toolkit.

	Char. Recall	**Char. Precision**	**Word Acc.**	**Word Acc. Ignore Symbol**	**1-N.E.D**
COMICS Dataset OCR	92.1	96.6	10.6	48.0	92.9
COMICS Text+	97.2	97.3	28.2	62.3	97.5
Neuspell [Jayanthi et al., 2020] on COMICS Text+	95.3	95.0	1.5	25.6	93.5

Table 2.10: Evaluation of Contextual Spell Check contextual spell correction to COMICS Dataset OCR and COMICS Text+ Data. 270 out of 500 ground truth data was used because that much of the data was exposed to correction and, as a result, corrected by the Contextual Spell Check.

	Char. Recall	**Char. Precision**	**Word Acc.**	**Word Acc. Ignore Symbol**	**1-N.E.D**
COMICS Dataset OCR	92.4	96.5	10.4	47.4	93.1
COMICS Text+	97.0	97.4	34.1	65.2	97.8
Contextual Spell Check [Goel, 2021] on COMICS Text+	85.2	91.9	1.1	2.2	80.8

Neural Spell Checking Attempts Besides rule-based post-processing, I apply neural spell-checking and correction by NeuSpell [Jayanthi et al., 2020] and Contextual Spell Check [Goel, 2021]. Contextual Spell Check works first by finding the vocabulary words; over those words, BERT [Devlin et al., 2019] is used to get predictions. However, it immensely decreases the accuracy of our OCR output (see Table 2.10). I suspect that the tool's dictionary is not comprehensive or adequate for our domain, Golden Age comics, which leads to a decline in performance. NeuSpell uses neural models for spelling correction, which are trained by posing the task as a sequence labeling task. In this task, a correct word is tagged as itself, and a misspelled token is marked as its suitable correction. I use *BERT Checker* from their

Table 2.11: Evaluation of Contextual Spell Check [Goel, 2021] contextual spell correction and Neuspell [Jayanthi et al., 2020] with respect to COMICS Dataset OCR and COMICS Text+ Data. 189 out of 500 ground truth data was used because that much of the data is exposed and, as a result, corrected by both of the neural spell checkers.

	Char. Recall	**Char. Precision**	**Word Acc.**	**Word Acc. Ignore Symbol**	**1-N.E.D**
COMICS Dataset OCR	91.2	96.1	9.5	46.6	92.7
COMICS Text+	96.5	96.8	26.5	58.7	97.7
Contextual Spell Checker [Goel, 2021] on COMICS Text+	84.7	91.3	1.1	1.6	81.0
Neuspell [Jayanthi et al., 2020] on COMICS Text+	94.3	94.1	2.1	15.3	92.7

trained models. Although it performs better than Contextual Spell Check, it still reduces the quality of the OCR output (see Table 2.9). Both of those tools mark different textbox texts as subject to change. I chose the texts they both corrected and compared their results with the COMICS dataset OCR and my raw OCR output to compare them on the same data. My raw OCR output outperforms all (see Table 2.11).

2.4 Results & Discussion

2.4.1 Finalized Dataset Results

Table 2.12 shows the comparison of OCR results of COMICS [Iyyer et al., 2017] and *COMICS Text+*. COMICS TEXT+ improves on every evaluation metric in comparison to COMICS OCR. Even though there is a slight increase in character precision, a significant improvement can be seen in character recall. This, in return, leads to a considerable increase in word accuracy and one minus normalized edit distance metrics. The qualitative comparison between the two datasets can be seen in 2.13.

The reason behind getting these results is that labeled data for the specific

Table 2.12: Final comparison of the proposed dataset called *COMICS Text+* and COMICS dataset [Iyyer et al., 2017] OCR results, measured on ground truth data. The proposed dataset outperforms the COMICS dataset OCR in all metrics. The *Comics Text+* increases word accuracy by 337% and word accuracy by ignoring symbols by 140% on the COMICS dataset's speech bubbles.

	Char. Recall	**Char. Precision**	**Word Acc.**	**Word Acc. Ignore Symbol**	**1-N.E.D**
COMICS Dataset OCR	92.9	97.2	13.0	51.6	93.0
COMICS Text+	**97.5**	**97.9**	**40.5**	**72.3**	**97.7**

comic book domain is prepared. Following that, the SOTA text detection and text recognition models are fine-tuned. The initial performance of these models is poor on the same data, as seen in Figure 2.8. From this point of view, I can say that if I want accurate text data in the comics field, I have to fine-tune existing models by using data annotated for the comic domain. It is cumbersome and time-consuming to hand-label text detection and text recognition data. However, with a limited amount of data, around 400 images, SOTA models can achieve almost top performance, see Figures 2.6 and 2.7. Text detection models benefit most heavily from the fine-tuning process. Otherwise, they create a bottleneck for the OCR process. That contrasts text recognition models, which can perform relatively well even with pre-trained versions.

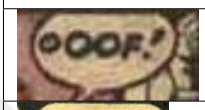
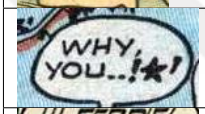
This study also shows how important quantitative analysis for OCR datasets is. Although the problems in the text data of the largest dataset, COMICS, in the comics field have been qualitatively revealed, they can be seen as inadequate. Along with the quantitative analysis, I show that the dataset is less accurate than claimed.

Error Analysis

Table 2.14 provides the most erroneous instances in the COMICS TEXT+ dataset from GT. Following faulty OCR patterns are identified in COMICS TEXT+:

- Symbols are hard to detect and recognize. Especially commas and dots are

Table 2.13: Qualitative comparison of samples from COMICS Text+ and COMICS datasets. Selected examples based on their 1-N.E.D. metric difference from ground truth data.

Speech Bubble	**COMICS Dataset OCR**	**COMICS Text+**	**Ground Truth**
	we seouΦht rein-force	- - - we brought re-inforce - ments ! a poose !	- - - we brought re-inforce - ments ! a posse !
	ooor ! 000f ,	ooof !	ooof !
	l / gh / ugh	ugh !	ugh !
	why ,	why you . . . i	why you . . ! * ’
	hon ’ s business ?	hi , ferdie ! how ’ s business ?	hi , ferdie ! how ’ s business ?
	c / se that odd weapon against us	look , lie ’ s go-ing to use that odd weapon against us !	look , he ’ s go-ing to use that odd weapon against us !
	and there ’ s	and there ’ s your pay . . . you	and there ’ s your pay . . . you fool !
	am crazy ea take him	im crazy , eh ? take him to the roof !	im crazy , eh ? take him to the roof !
	don ’ t be a ...	don ’ t be a . . ooooff	don ’ t be a . . ooooff !
	aboard . gone	come aboard - he ’ s gone !	come aboard - he ’ s gone !
	did !	he did !	he did !
	erry ’ s 74lew<unk>acr throwhag forma snowballs comer navamdy -	terry ’ s talent for throwing snowballs comes in handy !	jerry ’ s talent for throwing snowballs comes in handy !

often misrecognized interchangeably.

- Textures and background colors that are not very common can lead to text detection problems. This can be seen in the first and the third example of Table 2.14.

- Poorly detected bounding boxes lead to OCR issues. Speech bubble and narrative box detection or segmentation should be solved before the OCR step. Reflections of this issue can be seen as totally omitted or partially detected words, word groups, or multiple partial textboxes.

Better textbox detection or segmentation methods can solve problems caused by bounding box locations of text boxes. However, for the other problems, more extensive text detection and text recognition datasets for fine-tuning might be needed.

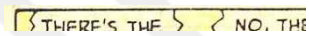
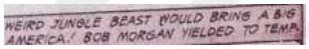
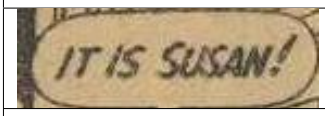
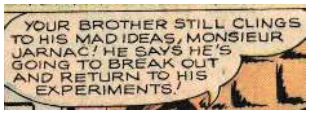
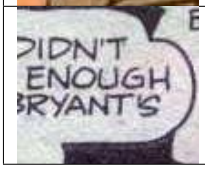
2.4.2 Results with Cloze Style Tasks

Iyyer et al. [Iyyer et al., 2017] propose three tasks (text cloze, visual cloze, and character coherence) to measure closure understanding of models given information about the previous three panels as context. If a model performs well, it can be deduced that it makes inferences between panel transitions to estimate the features of the following panel.

I reproduce the models and experimental setup as described in [Iyyer et al., 2017]. Our reproduction results can be seen in Table 2.15. For most of the experiment cases, the reproduction results are similar to those reported by a low margin. Given the reproduction results and experiment setup of character coherence, the validity of character coherence can be argued. Although the task is sound, the implementation shows that it only evaluates whether a text is before or after another text since no explicit character encoding or feature is provided. I think the difference can be stemmed from the dataset generation since the dataset itself is not shared but a script to generate many hyperparameters that are not shared.

The general setup for the tasks involves the following structure: features of context panels p_{i-3}, p_{i-2}, p_{i-1} are fed to model in order to predict panel p_i's features

Table 2.14: Qualitative error analysis results on COMICS Text+ dataset. Selected examples are from poorly performing speech bubbles based on the 1 - N.E.D. metric in ground truth data.

Speech Bubble	**COMICS Dataset OCR**	**COMICS Text+**	**Ground Truth**
	suddenly , ogg sees haga approaching	sudo ogg	suddenly , ogg sees haga approaching . . .
	pow	ma dow	pow
	there ' s the no , the	no the there ' s the	there ' s the no , the
	asluckx ste / s - int the room he i spelledby a terrific blow from behind	into felled terrific blow from	as lucky steps into the room he is felled by a terrific blow from behind
	weird jungle beast pould bring a big america bob morgan yelded 7 tema	weird jungle beast would bring a big yielded to temp . america . ! bob morgan	weird jungle beast would bring a big america ! bob morgan yielded to temp .
	itis susan	it ! susan !	it is susan !
	and there ' s	and there ' s your pay . . . you	and there ' s your pay . . . you fool !
	your brother still clings to his mad ideas , monsieur uarnac ! he says he ' s going to break out and return to his experiments .	your brother still clings monsieur to his mad ideas , jarnac ! he says he ' s going to break out and return to his experiments , !	your brother still clings to his mad ideas , monsieur jarnac ! he says he ' s going to break out and return to his experiments !
	williston parki	willist . park !	williston park !
	didnt enough ryants	i didn ' t enough suryant ' s	didn ' t enough bryant ' s

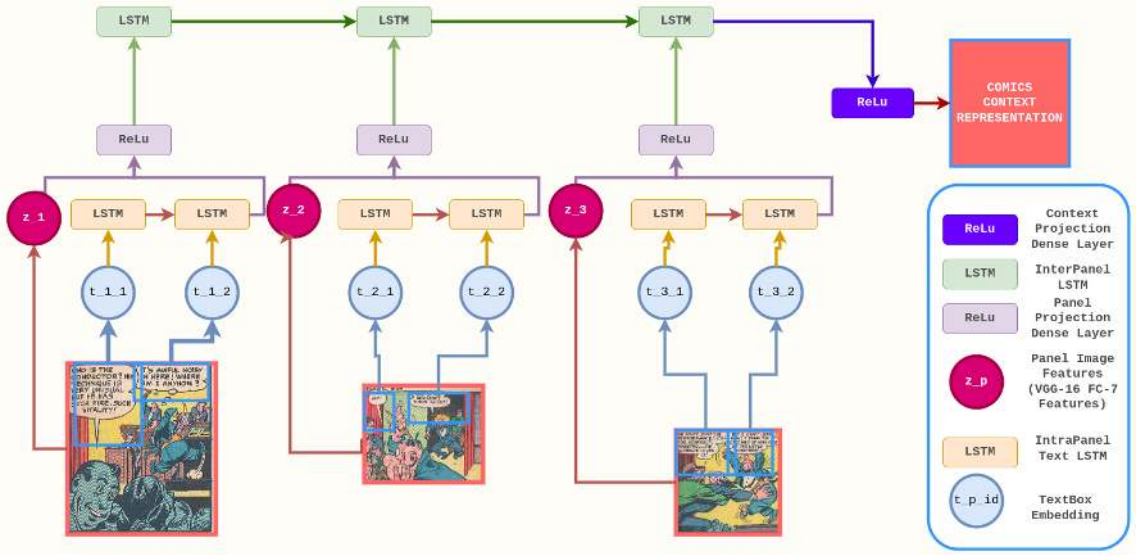


Figure 2.10: The extraction of comic sequence context representation from the architecture proposed by [Iyyer et al., 2017], utilizing both image and text modalities, is referred to as the **Image-Text** architecture. Architectures focused on a single modality are denoted as **Image-only** and **Text-only**, which maintain the same structure while excluding the processing of other modalities.

Table 2.15: Comparison of our replication results Iyyer et al.'s [Iyyer et al., 2017] results based on accuracy. Human baseline is also shared for only hard tasks since the easy task is trivial for humans. *NC-Image-text* models use only the final panel's image without any information from preceding context panels. No context model results strike the usefulness of the context for the cloze style tasks and narrative understanding.

MODEL	**Text Cloze**				**Visual Cloze**				**Char. Coherence**	
Task Difficulty	EASY		HARD		EASY		HARD			
	Original	Ours	Original	Ours	Original	Ours	Original	Ours	Original	Ours
Text-only	63.4	62.8	52.9	51.7	55.9	51.8	48.4	45.0	68.2	70.9
Image-only	51.7	50.7	49.4	45.6	85.7	79.5	63.2	56.1	70.9	70.7
Image-text	68.6	62.1	61.0	57.2	81.3	83.6	59.1	59.5	69.3	71.5
NC-Image-text	63.1	57.3	59.6	56.5	-		-		65.2	75.6
Human	-		84		-		88		87	

from single modality. This setup can further be enhanced by exploiting an arbitrary number of context panels. However, that is a study of future works. Prediction of the final panel's features is made with a cloze-style framework. Three candidates are presented for all experiments, and models use extracted context information to score the correct candidate higher than the rest. Figure 2.10 illustrates the context extraction process. The three tasks can be described as follows:

- **Text Cloze** is task of predicting final, p_i, panel's text. To reduce the task's complexity and to measure models' ability to learn from interpanel transitions, final panels are selected among those with only a single textbox. Architectures using visual modality, *image-text and image-only* applies masking to textbox areas so that the model cannot "see" text content and provides artwork features along with context to score textbox candidates.

- **Visual Cloze** is task of predicting final, p_i, panel's artwork. Differently from *text cloze* task, models are not provided with text features of the final panel.

- **Character Coherence** is the task of reordering two dialogue text boxes for the final panel.

Each task has two variations based on difficulty. In the *easy* case, the wrong candidates are selected from comic books that are different from the correct candidates. In contrast, candidates of the *hard* case come from the same comic book.

By using my implementation, experiments' text data are replaced with *COMICS TEXT+*. Test results of this training are shared in Table 2.16. Compared to the reproduction results with COMICS OCR data, an increase in accuracy can be observed for all text cloze task cases. SOTA results are achieved using the same models as Iyyer et al.'s [Iyyer et al., 2017] results. Except for character coherence and text-only model visual cloze case and image-text text cloze for the easy setting case. There is no or inconsiderable improvement in visual cloze tasks. From this point of view, I can say that when working with this architecture, text modality has limited effectiveness on visual cloze.

Although the improvement in the text-cloze task is notable, they are not groundbreaking, even if they are SOTA. It shows that the used model architecture and current experimental setup reached their maximum. Therefore, model architectures and experimental setups that are more suitable for the multimodal processing of comics are needed for the comic domain. In the Chapter 5, I propose such an approach.

Table 2.16: The results with replacing the COMICS OCR dataset with *COMICS TEXT+* dataset using the same model architecture and training procedure. The star indicates SOTA, the single up arrow indicates a slight increase, and the double up arrows indicate considerable improvement concerning my replication results.

MODEL	**Text Cloze**		**Visual Cloze**		**Char. Coherence**
Task Difficulty	EASY	HARD	EASY	HARD	
Text-only	64.5 ↑↑ ⋆	55.0 ↑↑ ⋆	51.6 ↑	45.5 ↑	67.0
Image-only	52.6 ↑↑⋆	50.3 ↑↑ ⋆	-	-	-
Image-text	64.1 ↑↑	61.4 ↑↑ ⋆	83.5 ⋆	59.0 ⋆	68.0

2.4.3 Performance on Modern Comics

The models trained for COMICS Text+ are used to extract text from speech bubbles that are not coming from Golden Age comics but from "Modern Age" comics, specifically, comics that were published after 2000.

The inference procedure is to crop speech bubbles by my speech bubble detector and then run the models on speech bubbles, not the whole pages or panels, so the same procedure would be applied to the dataset.

However, the performance turned out to be poor. Unfortunately, I cannot share any modern speech bubbles within the dataset for copyright reasons. This drop in performance is primarily because of text detector errors parallel to our study findings. I discovered that text detectors are the bottleneck for the OCR of comics.

The model, as just trained on our dataset, cannot separate text groups or words as well as Golden Age comics.

It means the publicly shared models can be used with Golden Age comics. However, If it is wanted to be used in comics of another period or style, the model needs to be retrained. In this study, I also found out that a few hundred annotated data is enough for the models to perform very close to their peak performance.

2.5 Conclusion and Future Work

I present the COMICS TEXT+ dataset, which is an improved refabrication of OCR data of the COMICS dataset [Iyyer et al., 2017]. Unlike COMICS, I don't use any paid and proprietary software for OCR. Instead, I create my pipeline to generate text detection and text recognition data to fine-tune SOTA deep-learning-based text detection and text recognition models to set up an end-to-end OCR pipeline. I also present the following datasets: *COMICS Text+: Text Detection* and *COMICS Text+: Text Recognition*. Finally, OCR data is extracted, and rule-based post-processing is applied to augment the data further, resulting in an objectively superior text dataset for the domain of the comic, making it the most extensive and accurate dataset available.

Although this is a step in the direction, there is a lot to be done for the computational approaches to comics. To leverage the multimodality of the comics and to have better generalizable models, comic series from different eras should be accessed and processed. Text detection models for comics should be focused on since it is one of the bottlenecks of OCR for comics. Comics components such as text boxes, panels, characters, and other low-level features should be processed with much better accuracy so that high-level processing of comics is enabled. I also show that current neural spell-checkers cannot be used off the shelf to improve OCR quality. Unified OCR models can be proposed. Namely, in this study, I focus on dialogues and narratives. However, in studies like [Baek et al., 2022], onomatopoeias are focused. If all of the text modalities can be extracted, then there will be a better chance to understand the visual language of comics. Currently, there is no adequate comics

processing backbone model to work on to process comics for high-level features such as narrative understanding, character relations, story understanding and generation, and so on.

Chapter 3

ASSOCIATING FACE AND BODY WITH SPEECH BUBBLES

3.1 Introduction

In comic books, speech bubbles are frequently used to represent character dialogue. However, matching the right character with their specific speech bubble can be difficult, especially when several characters are on a panel or page. This part of my thesis intends to investigate the efficacy of employing deep learning models to link the faces and bodies of characters with their corresponding speech bubbles in comics in order to address this issue. This task is a crucial part of my thesis since my approach overall claims to be character-centric, and without speech bubble to character association, it would be lacking in laying out the structure for comics. In order to fully comprehend the narrative structure and character interactions in comics, linking speech bubbles with characters is essential since it enables the understanding of the plot and character development.

Previous studies have looked into various methods, such as geometric graph analysis [Rigaud et al., 2015b], for connecting characters to speech bubbles. However, there has been limited research on the use of deep learning models for this purpose. In the COMIC MTL [Nguyen et al., 2019] method, a deep learning model is proposed that can detect and segment components such as faces, bodies, panels, narrative boxes, and speech bubbles. Moreover, the model can associate speech bubbles with the corresponding characters. Though my focus is to maximize the accuracy of character-to-speech bubble association, detection and segmentation output are crucial for my study.

This chapter of my thesis aims to enhance the COMIC MTL model's performance. COMIC MTL gives crucial information about a comic page when integrated

with other elements of this thesis, such as speech bubble segmentation with OCR (Chapter 2) and character face and body detection for character recognition and re-identification tasks (Chapter 4). To do this, I changed the relation network that assigns the links between characters and speech bubbles from various aspects. The efficiency of these improvements in connecting characters with speech bubble bubbles is then tested using the extended DCM772 dataset [Nguyen et al., 2018]. I also examine how various design decisions affect the performance of my model and compare it with the performance of the original COMIC MTL model.

3.2 Related Work

Comic MTL, introduced by Nguyen et al. [Nguyen et al., 2019], is a deep neural network that can detect and segment various components of comic book images, including panels, speech bubbles, characters, and narrative boxes. The model is trained using a novel loss function that optimizes the network for multiple tasks simultaneously, resulting in improved performance compared to existing methods. This approach was evaluated on benchmark datasets and demonstrated superior performance over other approaches.

In the field of object detection, Faster R-CNN [Ren et al., 2015] and Mask R-CNN [He et al., 2017] are two popular models that have been used for comic book analysis. Faster R-CNN utilizes a region proposal network to generate potential object regions, which are then classified using a convolutional neural network. Mask R-CNN extends Faster R-CNN by adding a segmentation branch to the network, allowing the model to generate pixel-level masks for detected objects. Similarly, Comic MTL is an extension of Mask R-CNN, specifically designed for comic book analysis. More details about Comic MTL can be found in section 3.3.1, as it forms the backbone of our proposed approach.

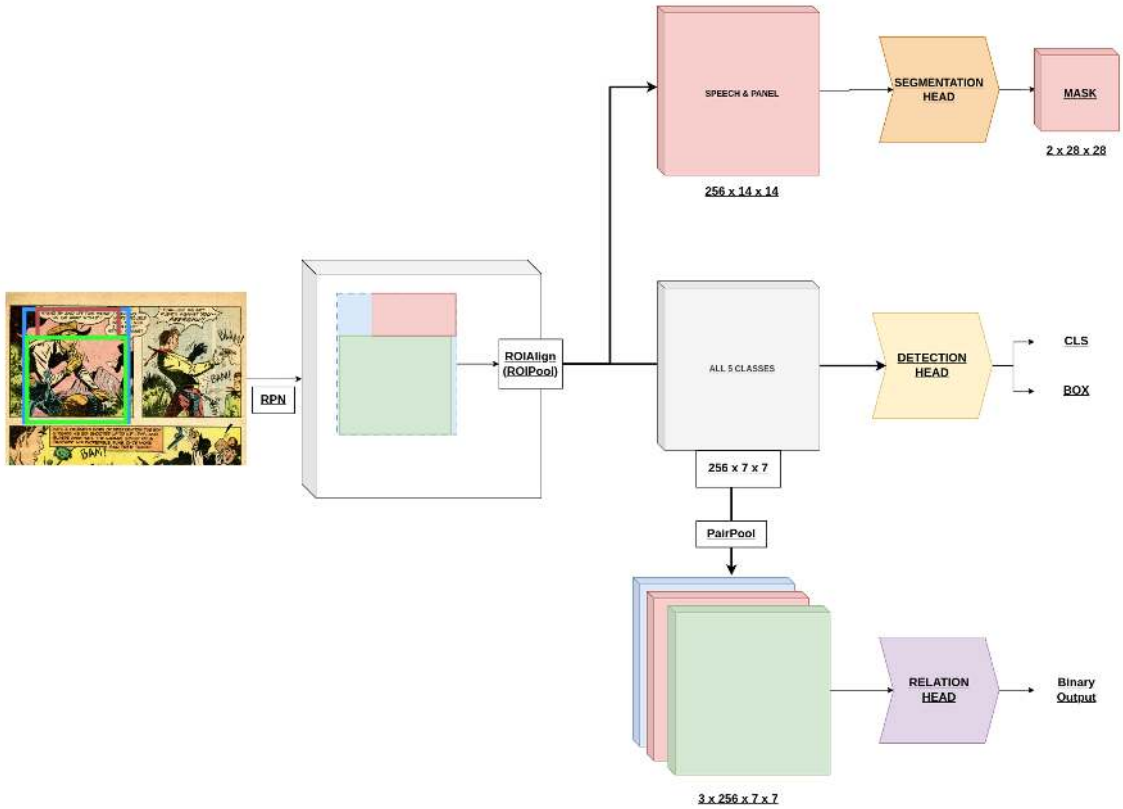


Figure 3.1: Overview of Comic MTL framework [Nguyen et al., 2019]. The figure shows the architecture of the model, which consists of three main stages: (1) the Region Proposal Network (RPN) stage, (2) the Region of Interest (RoI) Align stage, and (3) Three parallel branches for detection, segmentation, and relation (association) tasks The model takes a comic page as input and generates bounding boxes for characters' faces, bodies, panels, speech bubbles, narrative boxes, segmentations for speech bubbles and panels, and determines the association scores between speech bubbles and characters' faces, bodies, using a relation branch.

3.3 Methodology

3.3.1 Comic MTL Overview

This section explains the Comic MTL [Nguyen et al., 2019] model for analyzing comic page images to extract characters, panels, speech balloons, narrative text boxes, and the associations between characters and balloons. The model is built upon the Mask R-CNN model in instance segmentation. Two modifications are made to the Mask R-CNN model. Firstly, the loss function of the mask branch is modified to simultaneously learn detection and segmentation tasks and consider the origin of annotations because detection and segmentation heads have different inputs. Secondly, an additional branch is added to detect associations between balloons and characters using a Pair-Pool component and binary classifier. The classifier outputs probabilities of a pair being linked or not. The extraction of relevant features for the pairs of balloon-character is required. The modifications are further explained in the following paragraphs and visualized in 3.1.

The Comic MTL model addresses a detection/segmentation problem with classes that have both bounding box and segmentation mask annotations. The Mask R-CNN can provide predictions for both bounding boxes and masks, but it needs mask annotations for all classes in order to be trained effectively. To learn jointly the detection and segmentation tasks from bounding boxes and object masks, the authors enhance Mask R-CNN by only using mask annotations for objects from classes with mask annotations. Those classes are speech bubbles and panels. Instance segmentations of speech bubbles are part of the extended DCM772 dataset. As for panels, it is assumed that the bounding box itself is the segmentation mask. The multi-task loss is defined by a binary cross-entropy loss for mask prediction, and the loss is only applied to regions of interest with mask annotations. The binary cross-entropy loss is a function of the ground-truth mask and the predicted mask for a particular object class. The loss function includes only those classes that have ground-truth segmentation masks.

The relationship analysis between balloons and characters is considered a binary classification problem. Each balloon-character pair is classified as either having

a link or not having a link, where a linked pair means that the character speaks the text of the corresponding balloon. It is assumed that a character may link to multiple balloons, but a balloon can only link to one character. The entire image is processed rather than each panel separately, and it is assumed that the model learns the relevant features automatically from training examples. A new branch is added to the Mask R-CNN model to create a binary classifier for relation analysis.

The Pair-Pool component in the Comic MTL methodology processes a pair of bounding boxes instead of taking each box independently. Positive pairs are added to the pool, which are pairs of boxes corresponding to a linked balloon and character in the ground truth, and negative pairs are randomly added to the pool until they reach the same number as positive pairs. The number of pairs is then fixed and used as the input size for the additional branch. The multi-task loss for each sampled Region of Interest (RoI) is defined as the sum of four components: classification loss, bounding box regression loss, segmentation mask loss, and relationship loss. The relationship loss is optimized using binary cross-entropy loss and is calculated based on the predicted and ground-truth relation classes for each pair in the pool.

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} + L_{\text{rel}} \tag{3.1}$$

Unlike other branches, detection, and mask, that use a single shared feature, the new relation branch uses a combined feature of (1) the visual features of the two bounding boxes and their union, I call this *encapsulation box* and (2) the spatial features. The model reuses visual features from Mask R-CNN's shared features but encodes spatial features using 5-dimensional vectors that are not affected by translation or scale changes. These features can approximate the relative distance relationship between the two bounding boxes. The visual and spatial features are concatenated at the final layer to form the final features that are fed to the classifier head.

3.3.2 Improvements on Comic MTL

In this subsection, I will discuss the improvements made to the Comic MTL. These improvements cover various aspects of the methodology, including preprocessing and dataset, training optimization, positive-negative link (association) sampling, backbone model, relation branch, and post-processing technique called *character consistency*. The improvements aim to enhance the accuracy and efficiency of the Comic MTL model in recognizing and linking balloons and characters in comics. I will delve into each of these improvements and highlight their contributions to the overall performance of the methodology.

Preprocessing and Dataset

The Extended DCM 772 dataset used in previous work was found to be flawed, raising questions about the validity of the reported results. Several filtering steps were taken to address these issues to improve the dataset's quality.

Figure 3.2: Examples of filtered pages from the Extended DCM 772 [Nguyen et al., 2018] dataset. An advertisement page on the left, a cover page in the middle, and an erroneous annotation case on the right.

First, pages without faces were excluded from the dataset, even if other annotations were present. Pages without characters or speech bubbles were also excluded.

Additionally, pages with less than three panels were eliminated, as they often consisted of advertisements or cover pages. Examples of such pages are in Figure 3.2. Furthermore, some samples were manually eliminated if they contained errors in panel annotations.

To improve the accuracy of the dataset, positive links, where the body or face has a has-a-link relationship with the speech bubble, between faces and speech bubbles were evaluated using logical rules, where a face linked to a speech bubble was associated with the corresponding character. Additionally, face and character IDs were matched using intersection ratings, with faces without an ID assigned to the character with the highest intersection rating, provided that the rating was above a certain threshold (0.2).

After applying these filtering and improvement steps, the final train-validation-test split sizes were set to 544-25-25 pages, respectively, which was initially 772. The resulting dataset was then used to train and evaluate the proposed model, as described in the following sections.

Training Optimization

AdamW [Loshchilov and Hutter, 2017] was used in place of the stochastic gradient descent (SGD) optimizer to enhance the model's training procedure. The AdamW optimizer is a variation that incorporates weight decay into the optimization process to help avoid overfitting. Weight decay improves generalization performance by regularizing the model. Additionally, it has been demonstrated that the AdamW optimizer converges more quickly than SGD, which facilitates accelerated training.

I introduced the *CosineWarmupScheduler* [Loshchilov and Hutter, 2016] to optimize the training process further. The CosineWarmupScheduler is a learning rate scheduler that gradually increases the learning rate during the warm-up phase and decays it using a cosine annealing schedule. The warm-up phase allows the model to adjust to the learning rate slowly, which leads to faster convergence. The cosine annealing schedule helps the model avoid getting stuck in local minima by gradually reducing the learning rate.

Additionally, I used a number of augmentations during training, including random cropping, flipping, and rotation. This is intended to increase the diversity of the training data and make the model more resilient to changes in the input data for the test time.

```
A.Compose([
  A.ToFloat(max_value=255, always_apply=True),
  A.OneOf([
      A.LongestMaxSize(self.hparams.img_dims_hw[0]) if self.hparams.img_dims_hw[1] is None else A.Resize(
          self.hparams.img_dims_hw[0], self.hparams.img_dims_hw[1]),
      A.RandomResizedCrop(self.hparams.img_dims_hw[0],
                          self.hparams.img_dims_hw[1] if self.hparams.img_dims_hw[1] is not None else
                          self.hparams.img_dims_hw[0],
                          interpolation=cv2.INTER_CUBIC, scale=(0.2, 1.0)),
  ], p=1),
  A.OneOf([
      A.GaussianBlur(blur_limit=(5, 7), p=1),
      A.MotionBlur(p=1),
      A.MedianBlur(blur_limit=3, p=1),
      A.Blur(blur_limit=3, p=1),
  ], p=0.1),
  A.ToGray(p=0.2),
  A.OneOf([
      A.ToSepia(p=1),
      A.RandomSnow(brightness_coeff=2, p=1),
      A.RandomFog(fog_coef_lower=0.2,
                  fog_coef_upper=0.4, p=1.0),
      A.ColorJitter(p=1.0),
  ], p=0.1),
  A.HorizontalFlip(),
  A.Lambda(image=albu_clip_0_1),
  ToTensorV2(),
],
  bbox_params=A.BboxParams(format='pascal_voc',
                           label_fields=['labels', 'area', 'iscrowd', 'bb_ids', 'indices'],
                           min_area=16,
                           min_visibility=0.1))
```

Figure 3.3: Code snippet displaying the list of augmentations, implemented with Albumentations library [Buslaev et al., 2020], used for improving the Comic MTL model.

These improvements to the training optimization process helped to improve the performance of the comic MTL model. The model achieved better accuracy and faster convergence during training by introducing a better optimizer, learning rate scheduler, and augmentations.

Positive & Negative Link (Association) Sampling

A positive link indicates that a face or body is linked to a speech bubble, whereas a negative link indicates the opposite. My approach aimed to address the imbalance,

which may be the location or number of samples, between face and body samples in the existing random sampling methods.

To improve the positive and negative link sampling, I introduced *Sorted PairPool* sampling, which ranks the box proposals according to the intersection rate with GT boxes and uses the best ones. However, I noticed an imbalance between face and body samples in the Sorted Pair Pool Sampling, with most of the top samples coming from bodies. To address this, I introduced *Balanced Sorted PairPool* Sampling and adjusted the parameter a, which is used to determine whether a bounding is selected for sampling by checking IoU with the GT box, to achieve a better balance between face and body samples to get 2N samples.

In addition to these methods, I also developed **Sliding Window Negative Sampling**, which augments positive samples by converting them to negative samples in a way that by sliding the face or body bounding box up and down, left and right, to create negative samples. For faces, if the sliding window intersects with the body of the face to some degree, that is ignored.

Figure 3.4: An example of *sliding window negative sampling* method. The positive samples can be seen on the left, and the others are left and right-slide windows to generate negative samples. The red rectangle indicates the speech bubble bounding box proposal, green for the face, and blue for the encapsulation box.

Bubble Mirrored Negative Sampling is another technique I employed, which involves taking the symmetric face or character bounding box horizontally and ver-

tically relative to the speech bubble center and using it as a negative sample. These negative sampling methods are conceived due to the observation of slid face or character bounding boxes having a high probability of being associated with the speech bubble. We believe this technique helps the model to attend more to the direction of the tail and face or body element.

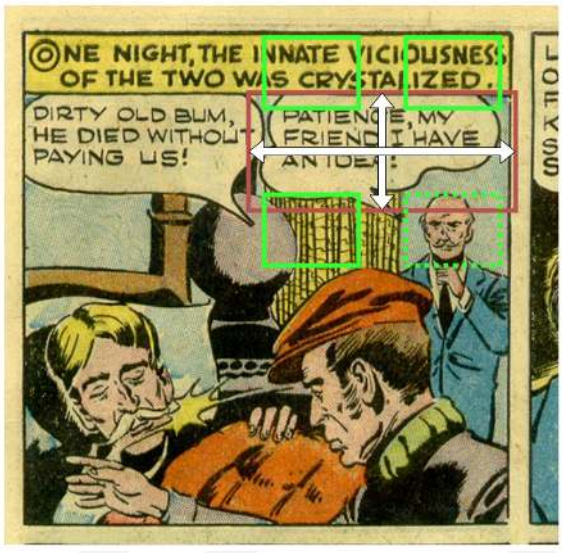


Figure 3.5: An example of *bubble mirrored negative sampling* method. Speech bubble and face bounding box proposals are indicated by red and dashed green rectangles, respectively. To generate negative samples, green bounding boxes are created by sliding the face box with respect to the speech bubble bounding box.

Finally, as for negative links, two types of negative links are exploited. The first is by pairing a speech bubble with a face or character that does not belong to it but belongs to another speech bubble. Additionally, I used faces or characters that are on the same panel but do not have a link with any speech bubble as negative samples; they are considered as **absent negative links**. These techniques helped to improve the dataset's quality by ensuring a better balance between positive and negative links.

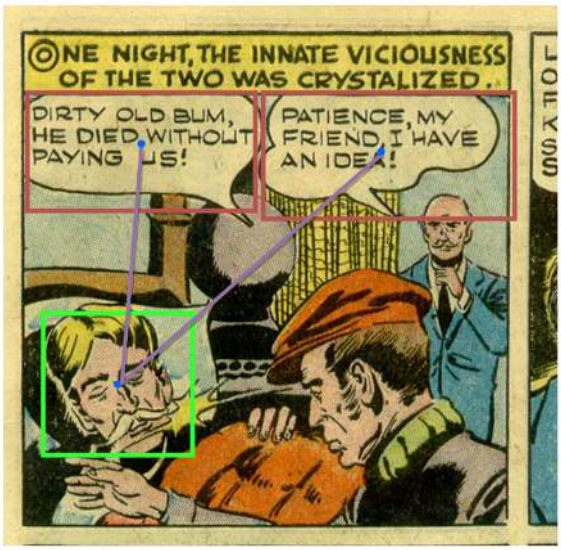


Figure 3.6: *Absent negative links* in a single panel. In absent negative links, speech bubbles are associated with a character with no speech bubble associated with it.

The Backbone Model

The proposed backbone model used in the Comic MTL was a MASK R-CNN network pre-trained on the ImageNet dataset. In contrast, I used the following models and weights as the backbone for my experiments. The results showed a significant improvement when using the weights in the third option, which can be seen in the Results section.

- *FasterRCNN ResNet50 FPN Weights COCO V1* [1]: This model does not have a mask branch so it is just for testing the effect of detection and relation head.

- *MaskRCNN ResNet50 FPN - Weights COCO V1*[2]: This model is the most similar that is used in Comic MTL. It has the same architecture but pretrained

[1] `https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn.html#torchvision.models.detection.FasterRCNN_ResNet50_FPN_Weights`

[2] `https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html#torchvision.models.detection.MaskRCNN_ResNet50_FPN_Weights`

with COCO.

- *MaskRCNN ResNet50 FPN V2 - Weights COCO V1*[3]: This model relies on vision transformers [Li et al., 2021], and produces our best and SOTA results.

Note that the above models are pre-trained on the COCO dataset and are available in the PyTorch Vision library.

Relation Branch

There are two major update areas with the relation branch. One is the encapsulation box masking idea. The other is to update the relation branch network architecture.

Encapsulation Box Masking is a technique used to mask the features of an encapsulation box. Each encapsulation box feature is composed of 256 x 7 x 7 feature maps. The 7 x 7 part stands for *Height* and *Weight* after the convolution operation; because of that, these maps can be traced back to the image patches of the encapsulation box. It is redundant to use the entire grid of the encapsulation box features, as it contains information that is not related to the face (body) or speech bubble with which the association is being queried. Therefore, this method can be considered a noise-filtering mechanism.

Since the encapsulation box features contain spatial information, a 7 x 7 mask was created in order to be multiplied with the features in an element-wise manner. The mask regions corresponding to the speech bubble and face (body) bounding boxes were set to 1, and the rest of the regions were set to 0. Several versions of this method were tried, including connecting the regions between the speech and face (body) bounding boxes using the modified Bresenham line-drawing algorithm [Bresenham, 1965] and marking connecting regions as 1 as well.

Different **Relation Branch Neural Network Architectures** are implemented and tested. However, most of them failed to give competitive results with the

[3] `https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn_v2.html#torchvision.models.detection.MaskRCNN_ResNet50_FPN_V2_Weights`

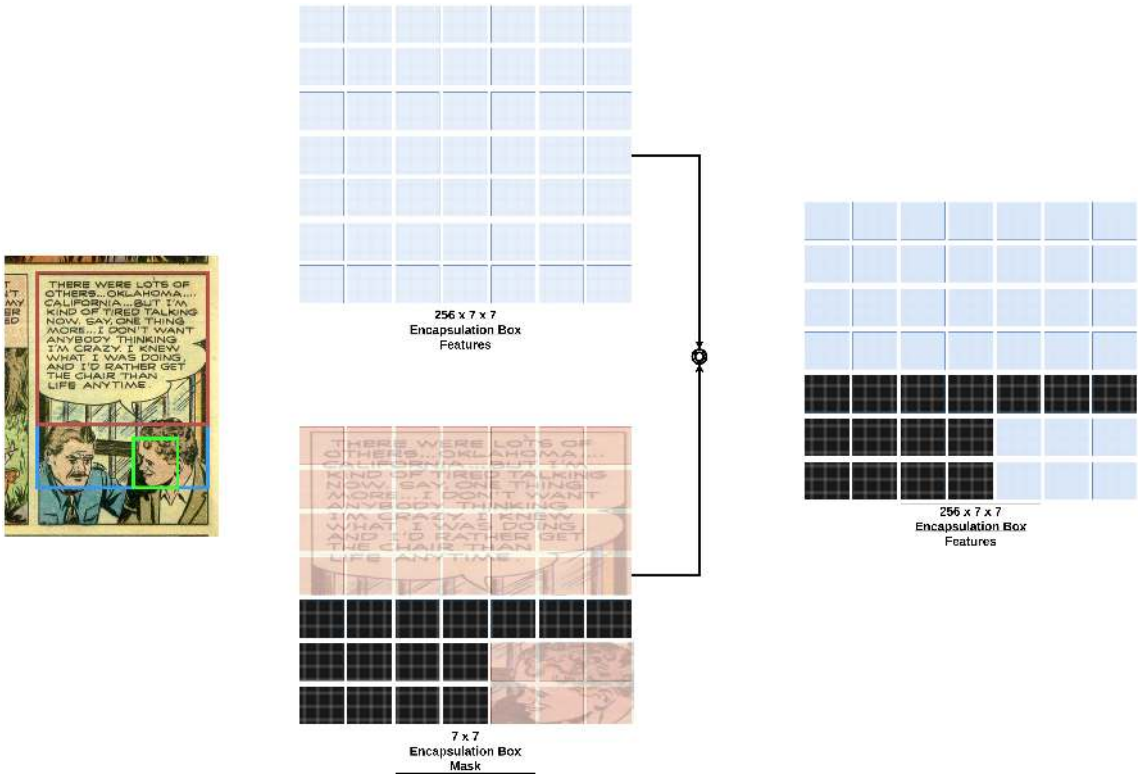


Figure 3.7: Example of *Encapsulation Box Masking* applied to ROI-Aligned features of encapsulation box. The 7 x 7 mask is multiplied element-wise with the encapsulation box features, a tensor with dimensions 7 x 7 x 256, with broadcasting, setting only the regions corresponding to the speech bubble and face (body) bounding box to 1 and the rest to 0. This technique helps to filter out the noise and irrelevant information, improving the accuracy of the model's associations.

baseline. Some of those models include the implementations and inspirations from proposed architectures from [Hu et al., 2018] and [Santoro et al., 2017]. The baseline and the most performant models are named as follows:

- **MLP:** This is my baseline model. It is the baseline because it is the most straightforward model, and I assume it was used in Comic MTL. The reason for my assumption is that Comic MTL does not specify relation branch architecture; there is a single figure indicating it is a two-layer MLP.

  After the pair-pool operation, the features are flattened and passed through the two-layered MLP. Before the output layer, spatial features are concatenated, and binary output is obtained. The model can be seen in 3.8.

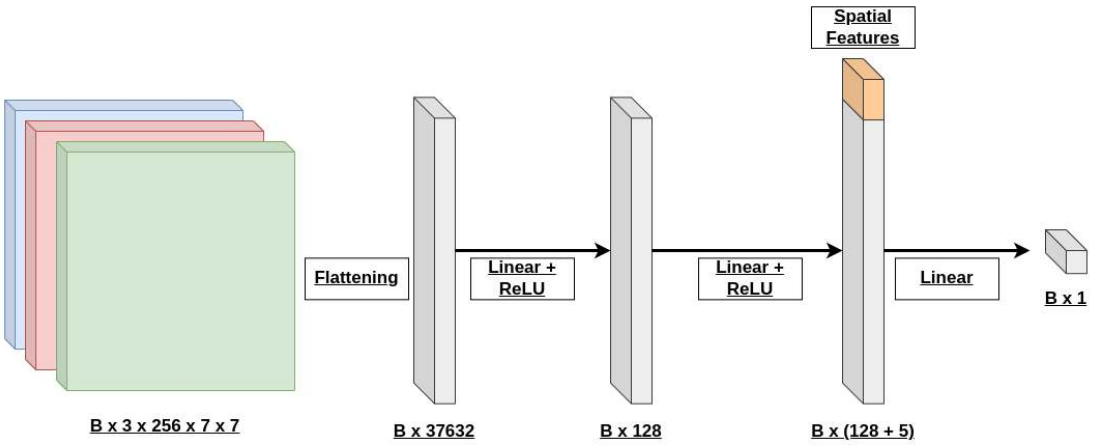


Figure 3.8: Relation branch's MLP head architecture.

- **Pair-Triple-Wise Convolutional Network (PTW-ConvNet):** I have a set of feature-maps for speech bubbles, faces, and encapsulation boxes denoted as s_i, f_i, and e_i respectively. I got inspired by [Tang et al., 2019b] to incorporate pair-wise and triplet-wise features of these elements. To compute these features, I multiplied the elements as follows: $sf_i = s_i \cdot f_i$, $se_i = s_i \cdot e_i$, $fe_i = f_i \cdot e_i$, $t_i = s_i \cdot f_i \cdot e_i$ where $\cdot$ denotes element-wise product.

After fusing the features with one another, I merged them and used them as the input of a Convolutional Neural Network (CNN). The CNN is represented by $d_i = \text{ConvNet}([\mathbf{sf}_i; \mathbf{se}_i; \mathbf{fe}_i; \mathbf{trp}_i])$. The architecture of the model can be seen in Figure 3.9.

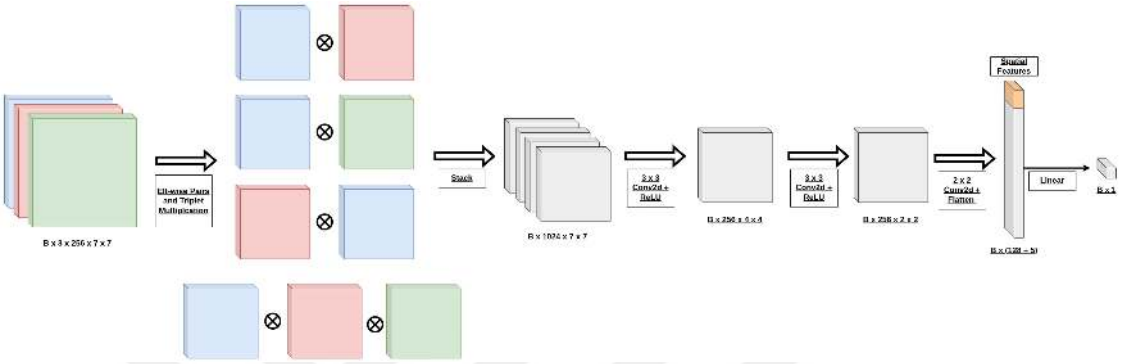


Figure 3.9: Relation branch PTW-ConvNet head architecture.

- **FastRCNNConvFCHead:** I used the box-head structure proposed in the [Wu et al., 2019] and [Li et al., 2021]. The only difference was the output dimension size. I passed the output of this layer through a linear layer and a ReLU before the output layer. Since the final structure of our my MTL model is based on [Li et al., 2021], I can say that the actual box-head and relation head are composed of common components, but they do not share parameters. The model's architecture can be seen in Figure 3.10.

- **MHA on RoI-Align Features:** I based the architecture of the relation head on the core idea of the Vision Transformer (ViT) [Dosovitskiy et al., 2020]. However, instead of using a linear projection layer to process image patches, I treated RoI-Aligned features' height and width dimensions as tokens of size 256.

  This means that the $7x7x3$ tokens are used as input to a transformer encoder model. The outputs of the transformer encoder are concatenated and fed to

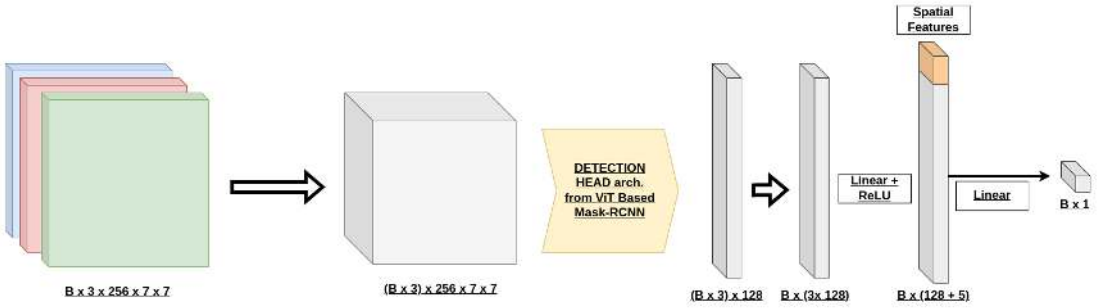


Figure 3.10: Relation branch using *FastRCNNConvFCHead* architecture, Detectron2's [Wu et al., 2019] box-head.

an MLP to produce the relation head output. Figure 3.11 demonstrates the model's architecture.

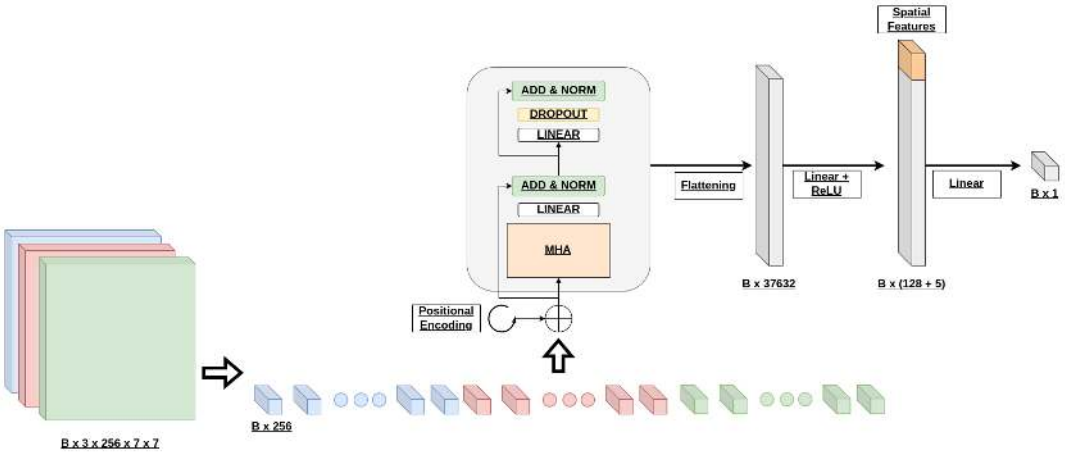


Figure 3.11: Relation branch using multi-head attention, treating channel dimension of RoI-Aligned features as tokens.

Character Consistency Post-processing and Evaluation

The way I get the relation outputs of the model is as follows: I separate the body, face, and speech bubble boxes that the model predicts. Here, I select the predicted boxes with a box score greater than 0.5. Then, I match each speech bubble with all bodies and faces one by one, pass it through the relation branch, and get the relation score, and then it is mapped between 0 and 1 with the sigmoid function. Relations with probabilities between 0 and 1, greater than 0.75, are pooled while others are removed.

Then, for each speech bubble, I divide all its associated boxes into groups according to whether it is a body or a body. I choose those with the highest probability in both groups. Because it is assumed that each speech bubble belongs to only one character, it can only have a single face and/or body.

With this method, it is possible to evaluate the relation performance of the model, and the evaluation was done with this method in the 3.4 section. For evaluation, the relevant boxes must match with 0.6 IoU. If there is no match, it is considered that our model has made a wrong prediction for that relation. However, if the boxes are predicted correctly, the relation between them is checked to determine whether there is a GT relation, and according to that, it is true or false.

However, there may be better ways to get the best results from the model, such as **Character Consistency Post-processing** that I proposed. With this method, I wanted to ensure the model gives the closest result to ground truth. This method works like the following:

- The characters given by the model are found one by one. The object mentioned as a character is a combination of face and body. If a body and face intersect at a specific rate, it is the face of that body. In the case of multiple intersections, the face that intersects with the body at the highest rate is assigned to that body. There can also be bodies without a face and faces without a body. Those are still considered character instances.

- Then, the relation score to all speech bubbles for each character is checked.

- If the item has a face and a body, the mean of the character's scores are calculated.
- Then, character components with the highest average score are matched with the speech bubble.
- If the speech bubble used here is not related to another character at a higher level, it can be decided that this is the character's bubble.
- This process continues until no speech bubble meets the criteria that are stated above.

The results of the post-processing technique can be seen in Figure 3.12.

Figure 3.12: *Character Consistency Post-Processing* applied to a comic page. The before and after images are shown on the left and right. The effect of the post-processing can be seen best in the mid-right and bottom-right panels of the page. Labels 1, 2, and 4 represent body, face, and speech bubble, respectively.

3.4 Results & Discussion

I have conducted a series of experiments using two different training setups: **slow-training** and **fast-training**. In the slow-training setup, I train the entire dataset consisting of train-val-test splits with 544-25-25 comic pages, respectively, for up to 40 epochs. In contrast, the fast-training setup involves training on a smaller dataset with train-val-test splits of 64-24-24 pages, respectively, for only 10 epochs.

In both training setups, I first freeze the backbone and only train the heads for the first ten epochs. Afterward, I train the entire model in the following epochs.

The fast-training setup serves as a helpful approach for serial experimentation, allowing me to quickly evaluate my model's performance with various hyperparameters and configurations. Once I have identified the most promising configurations through fast-training, I then perform slow-training to obtain more reliable results with optimal settings.

3.4.1 Fast-Training

Pair-pool Sampling

I conducted several experiment setups to measure the effects of sampling methods and approaches with fast-setting to understand how they affect overall training performance. The metrics and evaluation strategy are mentioned in paragraph 3.3.2.

Table 3.1 provides the results of different sampling strategies for face or body and speech bubble association. Experimenting with different Pair-pool strategies was significant in evaluating and selecting training data for the relation branch. Pair-pool curates the training data for the relation branch considering box head output.

The First N strategy is when the boxes to be matched are sorted by box score from the RoI head's outputs. This sample selection is obtained by taking N-positive and N-negative samples with the highest box scores. The First N method is considered to be the baseline for this setup. Following that **Randomized** is a random selection of N samples among possible positive and negative sample candidates. The randomized method results show that having a higher box score does not necessarily

mean training the model better and thus being a good input for the model. **Balanced** method ensures that the face and body have an equal number of elements within positive and negative samples. Namely, there are N//2 face and body samples for N positive or N negative samples. However, class balancing reduces the performance of the model with the fast-setting since solving the face and speech bubble association problem is harder than body and speech bubble association. I argue that this is because faces have less intersection with the speech bubble bounding box; in return, they do not share many features. Additionally, in comparison to bodies, faces can have varying orientations, expressions, and partial occlusions, making it difficult for the model to establish reliable associations. Due to that, the performance of the face is almost always lesser than its body counterpart, not just for pair-pool experiments but in others, too.

The **Sorted** option in the table indicates that the candidate samples are sorted based on their Intersection over Union (IoU) values with their corresponding ground truth (GT) counterparts. The sorting operation is performed, and the top N samples are selected. When combined with the balanced approach, the first N//2 samples are chosen for bodies, and the remaining N//2 samples are selected for faces. This strategy yields the best performance across most metrics, except for face precision.

The superior performance of the Sorted option can be attributed to the fact that when the selected samples are closer to their target boxes, the relation branch can be effectively trained, leading to improved results. By prioritizing the samples with higher IoU values, the model focuses on associations that are more likely to be correct, resulting in better overall performance.

Impact of Adding More Negative Samples

I evaluated the performance of my model on the association of bodies and faces with speech bubbles by adding more negative samples. The base case was when N samples constituted positive and negative; N was 76. The results in figure 3.13 showed that the model achieved reasonable precision, recall, and F1 scores for body associations. However, as we discussed, it faced more significant challenges in accurately

Table 3.1: Performance comparison of Pair-Pool sampling strategies with fast-setting.

Method	BODY			FACE		
	Recall	Precision	F1	Recall	Precision	F1
First N	50.00	52.88	51.40	40.31	34.21	37.01
Randomized	50.90	54.10	52.45	41.08	**43.44**	42.23
Randomized & Balanced	51.36	53.81	52.55	37.20	36.09	36.64
Sorted	**51.36**	**54.32**	**52.80**	**48.06**	40.00	**43.66**
Sorted Balanced	50.45	53.11	51.17	46.51	36.14	40.67
Sorted & Weighted Loss	46.36	48.11	47.22	36.43	31.54	33.81

associating faces with speech bubbles.

The rationale behind measuring the effect of adding more negative samples was related to how the model is used for inference. During inference, the model has to identify a relation of speech bubble with an element correctly. However, in most cases, there are more wrong options than correct ones since there can only be one face and one correct body. When additional negative samples were introduced during training, the model improved body precision and recall. It peaked when the additional samples were 150, almost equivalent to a 1:3 ratio between positives and negatives. However, this improvement was not consistent for face associations.

Encapsulation Box Mask Results

In this section, I analyzed the performance of various encapsulation masking strategies for associating bodies and faces with speech bubbles using the fast-setting. The results are summarized in Table 3.2.

First, I evaluated the performance without any masking (Without Mask). The results showed moderate body recall, precision, and F1 score performance. However, the face scores are still higher than the masked versions.

I then explored different versions of encapsulation masking. Version 1 marked only the relevant regions, which correspond to face or body and speech bubbles, as

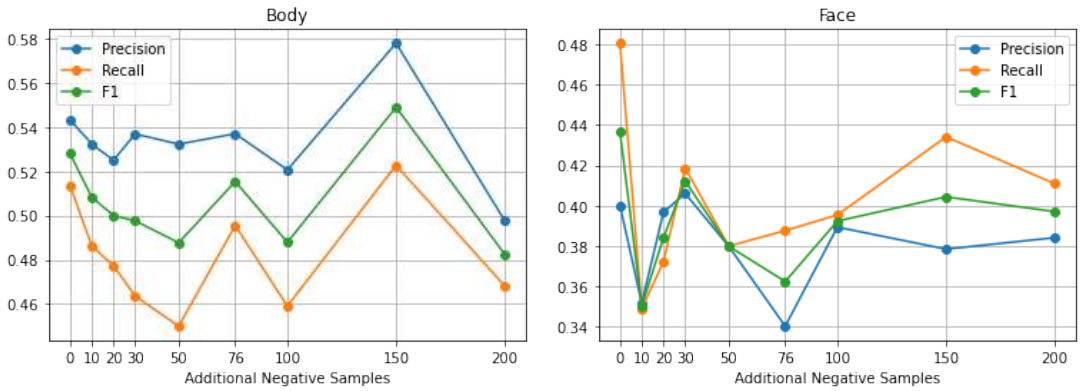


Figure 3.13: The impact of increasing the number of negative samples on precision, recall, and F1 scores for body and face association with speech bubbles, using the fast-training setup in the relation network. The base number of negative samples is 76.

one while masking the rest. This approach improved the body metrics but slightly degraded face performance.

In version 2, I introduced additional masking by drawing lines from box centers. Although it decreased body and face scores, it provided valuable insights that led to version 3.

In version 3, I relaxed the masking by marking surrounding areas within a 1-unit distance from the lines drawn in version 2. This further improved body and face scores.

Finally, version 4 modified version 1 by assigning masked areas a value of 0.25 instead of 0. This adjustment yielded improved face performance in return, resulted in a slight decrease in body scores. This makes sense of how much space a possible face occupies in the 7 x 7 grid compared to a body.

The different masking strategies had mixed effects on the model's performance, with some strategies improving specific metrics while affecting others. Careful selection of the appropriate masking strategy is crucial, considering the task requirements and trade-offs. Though these results justify why encapsulation mask is crucial, mask versions' performance does not directly translate to slow-setting. In the slow-setting,

I found that mask version 3 performs best.

Table 3.2: Performance comparison of various encapsulation masking strategies with fast-setting.

Method	BODY			FACE		
	Recall	Precision	F1	Recall	Precision	F1
Without Mask	51.36	54.32	52.80	**48.06**	40.00	**43.66**
1	**54.54**	**57.41**	**55.94**	42.63	38.19	40.29
2	49.54	53.69	51.53	40.31	37.14	38.66
3	53.63	51.52	52.56	45.73	35.32	39.86
4	53.18	56.52	54.80	44.18	**40.14**	42.06

Sliding Window Negative Samples

The results of the experiments comparing the number of sliding window negative samples during training, along with the inclusion of encapsulation box masking, are presented in Table 3.3.

In the experiments, I varied the count of sliding window negative samples and examined their impact on the performance metrics for both body and face association with speech bubbles.

I increased the count of sliding window negative samples to 5 without applying encapsulation box masking. This adjustment led to an improvement in body recall and F1 score. However, face recall, precision, and F1 score experienced a significant drop.

I introduced encapsulation box masking to count five sliding window negative samples to address the decline in face metrics. This combination yielded much better results compared to the previous method. Specifically, extreme drops in the face scores were mitigated. This also shows the importance of encapsulation box masking and the sensitivity of face and speech bubble association problems. Additionally, body recall and F1 score were slightly increased.

I further increased the count of sliding window negative samples to 10 without masking and observed a decline in performance across all metrics. However, when encapsulation box masking was applied along with ten sliding window negative samples, the performance improved for both body and face association.

Continuing the experimentation, I increased the count of sliding window negative samples to 15 without masking, increasing scores compared to counts of 5 and 10. As the number of sliding window samples increases, they lose their status as outliers. Nevertheless, when encapsulation box masking was applied with 15 sliding window negative samples, the performance improved compared to the previous method. Body and face scores both exhibited improvements except body precision.

Including sliding window negative samples during training surely does not boost the model's performance in the fast-setting. It may seriously harm the model's performance. However, that can be blocked by encapsulation box masking. Thus, it can help the model to differentiate which parts of the feature provide the association information since it was my observation that a model without any guidance could output a non-face area as an association. These findings highlight the importance of carefully selecting the count of sliding window negative samples and incorporating encapsulation box masking to strike a balance between body and face association.

Bubble Mirrored Negative Samples

The performance comparison of the quantity of bubble-mirrored negative samples during training with the fast-setting is shown in Table 3.4. The table assesses how encapsulation box masking and various counts of bubble-mirrored negative samples affect the model's body and face association performance.

A similar trend with sliding window negative samples can be seen in the results. Without the encapsulation mask, there is no improvement in any of the F1 scores. However, when encapsulation bıx masking is introduced, the F1 score surpasses all other cases and the case of having encapsulation box masking without bubble-mirrored negative samples. That shows a 55.94% F1 score in its maximum performance. Notably, the same positive performance increase cannot be seen when

Table 3.3: Performance comparison of the number of sliding window negative samples during training with fast-setting. Results with encapsulation box masking are also included in the table to show the combined effect of both techniques and as mitigation for drops in results of face and speech bubble association.

Count	Encapsulation	BODY			FACE		
	Box Masking	Recall	Precision	F1	Recall	Precision	F1
0		51.36	54.32	52.80	**48.06**	40.00	**43.66**
5		53.18	54.93	54.04	34.88	24.19	28.57
5	✓	**53.63**	54.63	**54.12**	46.51	37.26	41.37
10		43.18	51.91	47.14	31.78	33.06	32.41
10	✓	52.27	52.99	52.63	47.28	30.80	37.30
15		42.72	**57.66**	49.08	38.76	35.46	37.03
15	✓	50.45	54.68	52.48	43.41	**41.79**	42.58

the bubble-mirrored sample count is increased by more than 5. I think that is because it is enough to provide five samples to teach the model bubble symmetry. More than that would limit the samples with actual faces or bodies.

The findings suggest that encapsulation box masking and the quantity of bubble-mirrored negative samples significantly impact how well the body and face association model performs. While encapsulation box masking and a moderate amount of bubble-mirrored negative samples enhance the model's performance, a higher amount of bubble-mirrored negative samples could make it more difficult to distinguish between bodies and faces. These results suggest improving the training procedure and preventing performance declines in the speech bubble association task.

Absent Negative Links

Table 3.5 compares the base model's performance in body and face association tasks. Different experimental settings were explored, including varying counts of *absent negative links* and the application of encapsulation box masking. The results

Table 3.4: Performance comparison of the number of *bubble mirrored negative samples* during training with fast-setting. Results with encapsulation box masking are also included in the table to show the combined effect of both techniques and as mitigation for drops in results of face and speech bubble association.

Count	Encapsulation	BODY			FACE		
	Box Masking	Recall	Precision	F1	Recall	Precision	F1
0		**51.36**	54.32	52.80	48.06	40.00	43.66
5		46.36	50.49	48.34	34.88	29.22	31.80
5	✓	53.63	**59.89**	**56.59**	**49.61**	39.50	**43.98**
10		50.00	57.29	53.39	34.10	**47.82**	39.81
10	✓	50.45	53.11	51.74	44.18	39.04	41.45
15		45.45	53.19	49.02	38.76	29.41	33.44
15	✓	45.90	54.59	49.87	37.98	40.49	39.20

Table 3.5: Performance comparison of the number of *absent negative links* during training with fast-setting. Results with encapsulation box masking are also included in the table to show the combined effect of both techniques and as mitigation for drops in results of face and speech bubble association.

Count	Encapsulation	BODY			FACE		
	Box Masking	Recall	Precision	F1	Recall	Precision	F1
0		51.36	54.32	52.80	**48.06**	40.00	**43.66**
5		47.27	56.83	51.61	38.76	37.59	38.16
5	✓	48.63	57.52	52.70	40.31	**45.21**	42.62
10		44.54	58.33	50.51	31.00	31.49	31.25
10	✓	50.90	**65.11**	**57.14**	31.78	41.83	36.12
15		48.18	48.84	48.51	40.31	35.86	37.95
15	✓	**52.27**	53.99	53.11	36.43	38.84	37.60

demonstrate that incorporating *absent negative links* and utilizing encapsulation box masking techniques impact the model's performance positively in terms of body and speech bubble association.

In the body association task, including 5 *absent negative links* along with encapsulation box masking yields improved precision and F1 score. Additionally, in the face association task, encapsulation box masking has a positive effect on precision. These findings suggest that the careful inclusion of negative samples and the use of appropriate masking techniques enhance the model's performance. The phenomenon of declined face and speech bubble association performance continued with this data augmentation technique. This again shows the difficulty of solving the face and speech bubble association problem.

This technique is important in effectively addressing the face or body and speech bubble association problem. The outcomes highlight the significance of thoughtful data preparation and the implementation of suitable methodologies to achieve improved performance in these association tasks. Using absent negative links improves the overall body and speech bubble association performance but leads to a decline in the face. This can also be seen in Table 3.7. In that, when absent negative links are used more than ten and mixed with other negative samples to be selected for which pair-pool sampling strategy is applied, it has the same effect.

Relation Branch

Table 3.6 presents the performance comparison results among different neural network architectures with various layer dimensions in the context of relation branch modeling. This comparison aimed to evaluate the effectiveness of these models during fast-setting training. The models considered in the analysis are MLP, PTW-ConvNet, FastRCNNConvFCHead, and MHA on RoI-Align Features; the details of these architectures can be found in paragraph 3.3.2.

The MLP models serve as the baseline and provide a reference point for comparison. They demonstrate reasonable performance for body association but perform poorly for the face.

Table 3.6: Performance comparison of the different relation branch neural network architectures with different layer dimensions during fast-setting training. The best results for each metric are shown in bold.

Model	Final Linear	BODY			FACE		
	Layer Dim	Recall	Precision	F1	Recall	Precision	F1
MLP	128	**52.72**	57.42	54.97	31.78	38.31	34.74
MLP	512	**52.72**	62.03	57.00	**43.41**	42.42	42.91
PTW-ConvNet	128	50.45	59.04	54.41	41.08	40.45	40.76
FastRCNNConvFCHead	128	43.18	**65.97**	52.19	39.53	48.57	**43.59**
FastRCNNConvFCHead	512	51.81	64.40	**57.43**	36.43	**48.95**	41.77
MHA on RoI-Align Features	128	45.45	54.05	49.38	18.60	19.83	19.20
MHA on RoI-Align Features	512	50.45	59.67	54.68	41.86	43.54	42.68

The PTW-ConvNet model, inspired by the idea of incorporating pair-wise and triplet-wise features, shows competitive performance in the body and face association tasks. While its results are not the highest in any particular metric, the model consistently performs well across multiple evaluation measures. This shows that using pairs of features for the association task is promising.

The FastRCNNConvFCHead models exhibit different strengths based on their layer dimensions. The model with a layer dimension of 128 achieves high precision in the body association task, indicating its ability to predict positive links accurately. On the other hand, the model with a layer dimension of 512 achieves a notable F1 score in the body association task, demonstrating its overall effectiveness in capturing relations. In the face association task, the FastRCNNConvFCHead model with a layer dimension of 128 achieves the highest F1, implying its proficiency in correctly identifying related elements. Interestingly, when this model's final layer dimension is increased, it loses some of its ability to capture face associations. Thus, this results in lower F1 for face association.

The MHA on RoI-Align Features models, while showing lower performance compared to other architectures, provide insights into the limitations of this approach. These models yield relatively lower recall, precision, and F1 scores in both the body and face association tasks. This suggests that the direct application of the MHA-

based approach on RoI-Align Features may not be as effective as the other models in this specific task. On the other hand, with the final layer dimension 512, it performs competitively with the other models. Using a multi-head attention mechanism and transformer encoders on RoIAlign features is a promising direction. However, It is a known issue to set transformers up and optimize their training stably. Therefore, more experiments are needed. It should be considered as future work.

Another finding is that increasing the final linear dimension leads to more performant models, almost with all models. This is because these experiments were conducted in the fast-setting. Hence, the training dataset size was 64 comic pages. Because, with slow-setting, this is not the case, possibly leading to overfitting.

The results highlight the importance of selecting the appropriate neural network architecture for relation branch modeling. Each architecture has its strengths and weaknesses, and the choice should be based on the specific requirements and characteristics of the task. Given the complexity of face association tasks and the FastRCNNConvFCHead model's performance, I continued experiments for the slow setting with FastRCNNConvFCHead.

3.4.2 Slow-Training

Table 3.7 provides a comparison of the performance of *FastRCNNConvFCHead* model as the relation branch under different settings during slow-setting training. It explores the impact of negative sampling methods and augmentation on the model's performance. In all of the experiments, encapsulation box masking version 3 is used. As for the pair-pool sampling strategy, sorted-balanced is chosen. As input image size 800 is decided for height, the width is dynamic based on aspect ratio. The number of additional negative samples is fixed at 150 because it resulted in the best body F1 score during the fast-setting experimentation stage.

The results indicate that the choice of the negative sampling method plays a crucial role in the model's performance. Incorporating negative sampling methods improves performance. However, there is a trade-off between the performance of the body and face with respect to absent negative links. An increase in absent negative

Table 3.7: Comparison of *FastRCNNConvFCHead* Performance with Different Settings during Slow-Setting Training. The *Mixed* option combines absent negative links with negative links and selects negative samples using sorted-balanced Pair-Pooling. The best results for each metric are highlighted in bold.

Negative Sampling Methods				Aug.	BODY			FACE		
Bubble	Sliding	Absent	Additional		Recall	Precision	F1	Recall	Precision	F1
0	0	0	150		62.89	64.35	63.61	59.23	55.39	57.24
5	5	10	150		60.18	64.56	62.29	**65.38**	**64.39**	**64.88**
5	5	10	150	✓	58.82	64.35	61.46	60.00	62.40	61.17
0	0	Mixed	150		61.53	66.99	64.15	63.07	61.19	62.12
0	0	Mixed	150	✓	60.18	62.73	61.43	57.69	58.59	58.14
5	5	Mixed	150		**63.34**	**68.96**	**66.03**	63.84	63.35	63.60
5	5	Mixed	150	✓	60.18	66.16	63.03	57.69	59.52	58.59

links helps the model differentiate bodies better at a minor face performance cost. However, the gain in body performance is significant.

Additionally, the impact of data augmentation is examined, where the "Aug." column indicates whether augmentation was applied. Although I have tried several different augmentation settings, the results did not improve unexpectedly. The whole list of augmentations can be found in paragraph 3.3.2.

I performed experiments using subsets of these augmentations; however, the results showed no improvement. This lack of improvement can be attributed to the dataset itself, which is relatively small and needs more diversity in covering all aspects and styles of comics. Consequently, the dataset exhibits low variance, which could limit the effectiveness of augmentations in enhancing the model's performance.

Overall, the results emphasize how crucial it is to choose the best mixture of augmentation and negative sampling methods to get the best performance out of the FastRCNNConvFCHead model during slow-setting training.

3.4.3 Final Results

Table 3.8 compares the results obtained from Comic MTL and my approach in various segmentation, detection, and association tasks. The recall, precision, and F1 score performance metrics are reported for each task. For the final results, the best-performing model in terms of body F1 score is chosen, and association scores are reported after character consistency post-processing.

In terms of speech bubble segmentation, my approach is significantly better. However, it should be noted that for speech bubble detection [Nguyen et al., 2019] does not provide any results. So, it needs to be clarified how they measured speech bubble segmentation. Namely, whether false positives are included in their evaluation needs to be clarified.

Both Comic MTL and the proposed approach demonstrate high performance regarding panel detection. However, they report higher precision and, thus, higher F1 scores. As for narrative box detection, both Comic MTL perform significantly better. The reason for this discrepancy arguably stems from the fact that the dataset I used was the subset of DCM772, and also, in their work, the image dimensions are not provided. I employed a height of 800 for my approach, while the width was determined according to the aspect ratio.

Additionally, it should be noted that panel and narrative box detections are relatively large components. Additionally, I focused heavily on improving speech bubble association tasks in our experiments. That, in return, might affect the model negatively for panel and narrative box detection tasks.

However, for the others, comparatively smaller components, character (body) and face, I have superior scores, although using a smaller dataset. This might be one of the reasons why I report more recall with the association tasks. My approach outperforms Comic MTL in both tasks, especially showing remarkable progress in body-to-speech bubble association.

The proposed approach consistently achieves competitive or superior results compared to Comic MTL in various segmentation, detection, and association tasks. These findings demonstrate the effectiveness and superiority of my approach to

comic analysis.

Table 3.8: Comparison of the results obtained from the original Comic MTL model [Nguyen et al., 2019] and my enhanced approach's segmentation, detection, and association tasks. My approach achieves SOTA results in the association tasks.

	Recall		Precision		F1	
	Comic MTL	Ours	Comic MTL	Ours	Comic MTL	Ours
Speech Bubble Segmentation	89.50	**97.00**	94.87	**95.80**	92.11	**96.40**
Speech Bubble Detection	-	92.60	-	95.40	-	94.00
Panel Detection	**98.71**	96.80	**96.84**	92.40	**97.76**	94.60
Character Detection	67.56	**72.60**	76.21	**82.10**	71.62	**77.00**
Face Detection	72.39	**79.50**	**82.12**	81.00	76.95	**80.30**
Narrative Box Detection	**84.38**	81.40	**91.22**	82.60	**87.66**	82.00
Balloon-Character Association	45.64	**63.30**	**71.01**	70.00	55.57	**66.50**
Balloon-Face Association	53.47	**61.50**	**72.00**	67.80	61.36	**64.50**

3.5 Conclusion

In conclusion, this part of my thesis has aimed to achieve state-of-the-art (SOTA) results in comic analysis, especially for face and body to speech bubble association tasks, by utilizing new negative sampling methods, optimizing the pair-pool process, a novel masking strategy for the relation branch, and post-processing methods. Building upon the COMIC MTL structure, I optimized all the components, starting from the preprocessing dataset to the neural network of the architecture of the relation branch, in order to enhance the overall performance.

The significance of this research lies in achieving superior results and advancing the field of comic analysis. The developed methods and strategies can be further explored and extended to tackle new challenges and tasks within comics. Some of those challenges are part of this thesis, such as character recognition in comics, which would enable me to track characters across panels in a way that their speech bubble content is associated with them. Thus enabling converting comics into dialogue-like structures. Additionally, it is crucial to have a model that can process comic pages

to their components. It complements the first part of my thesis, which is about improving the OCR performance of speech bubbles. Since the MTL model provides speech bubble segmentations, the OCR performance of models can be further improved.

Chapter 4

IDENTITY-AWARE SEMI-SUPERVISED LEARNING FOR COMIC CHARACTER RE-IDENTIFICATION

4.1 Introduction

Comic character re-identification verifies and links characters across different panels or scenes in a comic book or strip. This task entails accurately connecting instances of a character, even when their appearance varies in terms of pose, expression, or context. Sometimes, characters may be portrayed in close-up shots, focusing predominantly on their faces. In contrast, their faces might not be visible in other instances due to pose variations. Hence, character re-identification in comics is a challenging area that calls for thorough exploration and novel solutions. Despite the significant research in character recognition, the task of character re-identification across comic panels still needs to be explored. Existing literature primarily concentrates on character recognition, particularly emphasizing face recognition and retrieval tasks.

In the context of comic character re-identification, the primary challenge is the scarcity of accurately annotated data for character faces and bodies, particularly when compared to the available data for character detection [Inoue et al., 2018, Zheng et al., 2020b, Nguyen et al., 2018]. Assigning identities to comic book characters is challenging due to their arbitrary appearances across pages, resulting in few datasets like Manga109 [Aizawa et al., 2020] having character identity annotations.

To address this challenge, I employed a multi-step approach. I began by using a SOTA model [Topal et al., 2022] to detect character instances within comic pages. Subsequently, I developed an identity-aware self-supervised model using contrastive learning. This model generates unified embeddings for comic faces and bodies, en-

suring consistent identity representation for these character instances. Building on this, I further refined the framework by fine-tuning a linear projector. This projector uses identity annotations collected from sequences of four panels and incorporates metric learning principles and loss functions to extract identity representations. Finally, I clustered identity representations for a given number of characters, where each cluster represents a distinct identity. The core contributions of this work are as follows:

- I present a novel *Identity-Aware* self-supervised framework designed to simultaneously process both facial and bodily features of comic book characters resulting in unified character embeddings. This approach exploits the inherent similarity between facial and bodily representations, enabling alignment between the two with *identity-awareness loss*. This approach contributes significantly to the success of solving the character re-identification problem.

- By using semi-supervised learning, I fine-tuned a linear layer on top of my identity-aware self-supervised backbone with metric learning techniques, losses and miners, to produce identity features from character embeddings. By clustering identity features, I address the task of comic character re-identification, demonstrating the potential to achieve robust results even with a restricted amount of annotated data on inter-series and, for the first time, in-series evaluation metrics.

- I created two new datasets. The first one is the *Comic Character Instances Dataset* containing over a million character instances, which I generated by applying the SOTA *DASS Detector* [Topal et al., 2022] framework to the COMICS [Iyyer et al., 2017] dataset. The second dataset, named *Comic Sequence Identity Dataset*, includes annotations of identities within more than 3000 sets of four consecutive comic panels that I collected.

4.2 Related Work

Person Re-Identification (ReID) is a computer vision task involving recognizing and tracking individuals across camera viewpoints and scenes. Its primary objective is to establish the identity of a person in one camera view and match it with the same person in other camera views, even when facing challenges like changes in posture, variations in camera perspectives, and instances of occlusion [Wei et al., 2022b]. In comics, character re-identification is done on panels instead of videos or photographs as they innately represent characters from different viewpoints or scenes. In this sense, the related work contains work related to comic character re-identification.

4.2.1 Character Recognition

The "Progressive Deep Feature Learning" framework proposed in [Qin et al., 2019] emphasizes the acquisition of a coarse feature representation from unlabeled data, followed by fine-tuning with labeled data. Notably, their approach demonstrates effective performance in manga character recognition, showcasing the potential of leveraging unlabeled data.

The task of learning from diverse modalities is tackled by [Zheng et al., 2020a], who present a meta-continual learning approach for joint face recognition across various styles including sketches, cartoons, caricatures, and real-life photographs. Their shared feature representation and subsequent type-specific classifiers exhibit strong recognition capabilities even in scenarios with limited training data.

4.2.2 Character Labeling in Animation

In the realm of character labeling within animation, [Nir et al., 2022] introduces a self-supervised approach named CAST (Character Labeling in Animation using Self-supervision by Tracking). CAST leverages motion tracking to label characters within animations. This innovative pipeline encompasses automatic shot segmentation, detection, multi-object tracking, and contrastive-based refinement, ultimately defining unique semantic embeddings for animation styles. This approach enhances

the understanding of animated content through robust semantic representations.

4.2.3 Unsupervised Manga Character Re-identification

An analogous work to my study is the work in [Zhang et al., 2022], which investigates unsupervised domain adaptation methods for character re-identification using the Manga109 dataset [Aizawa et al., 2020]. Notably, the presence of character identities in Manga109 aids their approach, assisting in image selection and training. In a similar vein, this paper proposes an unsupervised method for manga character re-identification. By extracting face and body features and employing clustering algorithms, it successfully achieves promising re-identification performance without the need for labeled data.

4.3 Methodology

My methodology consists of two main parts. First, I explain how I curated the data for faces and bodies of the COMICS [Iyyer et al., 2017] to train a novel identity-aware self-supervised model as the backbone for the reidentification model. Secondly, I go through data preparation and mining to train the reidentification model with fine-tuning on a limited amount of data that is collected from *Comic Sequence Identity Annotator*.

4.3.1 Self-supervised Learning of Comic Characters

Dataset for Self-supervision

The main data source is the panels of the COMICS dataset. I ran the *XL Stage-3 w/ Mix of Datasets* model of the DASS [Topal et al., 2022] approach on all panels. I specifically chose this model checkpoint because this model gave the best results for both face and body on the DCM772 [Nguyen et al., 2018] dataset, which is the domain-wise closer to the COMICS.

The scores and bounding boxes can be obtained from the predictions of the DASS model. There were a few post-processing steps to remove and organize the faces and bodies. I did not include a prediction for the self-supervision dataset if the

bounding box areas were less than 64 and the detection score was lower than 0.95. During training, the face images are scaled by 1.2, then cropped as a square by the longest side to capture the head of the character completely. Then, the faces and bodies were detected as character instances, which were automatically labeled by a rule-based algorithm. I did this as follows: If there is a body in the panel where any face intersects over 0.95, I indexed this face and body as a character instance. If a face intersects with more than one body bounding box at the rate I specified, those were assigned to the body with the minimum difference between the top-y coordinates. Because the faces are naturally located at the top of the body, some samples from this dataset can be seen in 4.1.

Table 4.1: A section of the *Comic Character Instances Dataset* from series 1551, page 1, panel 2. This dataset contains detections from the DASS model. If a face and body belong to the same character instance, they share the same 'char_index', whereas the 'index' column is used to access the actual file path of the cropped image.

char_index	**type**	**index**	**x_0**	**y_0**	**x_1**	**y_1**	**score**	**series_id**	**page_id**
0	face	0	520	189	691	395	0.98	1551	1_2
0	body	1	491	92	717	541	0.93	1551	1_2
1	face	1	110	66	156	117	0.96	1551	1_2
1	body	0	47	30	222	560	0.99	1551	1_2
2	face	2	246	124	272	153	0.72	1551	1_2
3	face	3	517	420	538	445	0.65	1551	1_2
3	body	3	498	413	585	541	0.87	1551	1_2
5	body	2	185	433	253	562	0.90	1551	1_2

During the self-supervised pretraining phase, the below datasets were used, and total counts can be seen in Table 4.2:

- **Face 100k**: Randomly selected 100,000 samples from all suitable face images.

- **Body 100k**: Randomly selected 100,000 samples from all suitable body images.

- **Face + Body 100k**: This dataset merges *Face 100k* and *Body 100k* datasets. Thus resulting in 173,828 instances of face and body images.

- **Face + Body** This dataset merges all suitable faces and bodies. **Face + Body** datasets contain the character instances containing face and body predictions in pairs.

Table 4.2: Total number of image instances for the dataset used in the self-supervised pretraining phase. The term '100k' refers to using 100,000 faces and bodies. Most of the experiments were trained with a 100k dataset since using all of the datasets takes almost a week to train.

Dataset Type	**Count**
Face	917,092
Body	1,139,956
Face+Body	1,265,710
Face+Body 100k	173,828

Self-supervised Models: Vanilla SimCLR

The vanilla SimCLR model refers to combination of v1 and v2 of SimCLR frameworks [Chen et al., 2020a, Chen et al., 2020b]. My vanilla SimCLR is reproduced mostly from v1, and the capacity of the projection head is increased, as suggested by v2. The used loss function is the same as the SimCLR, meaning NT-Xent loss is used. The equations in 4.1 are the loss for a single element in the batch. The similarity metric is cosine similarity.

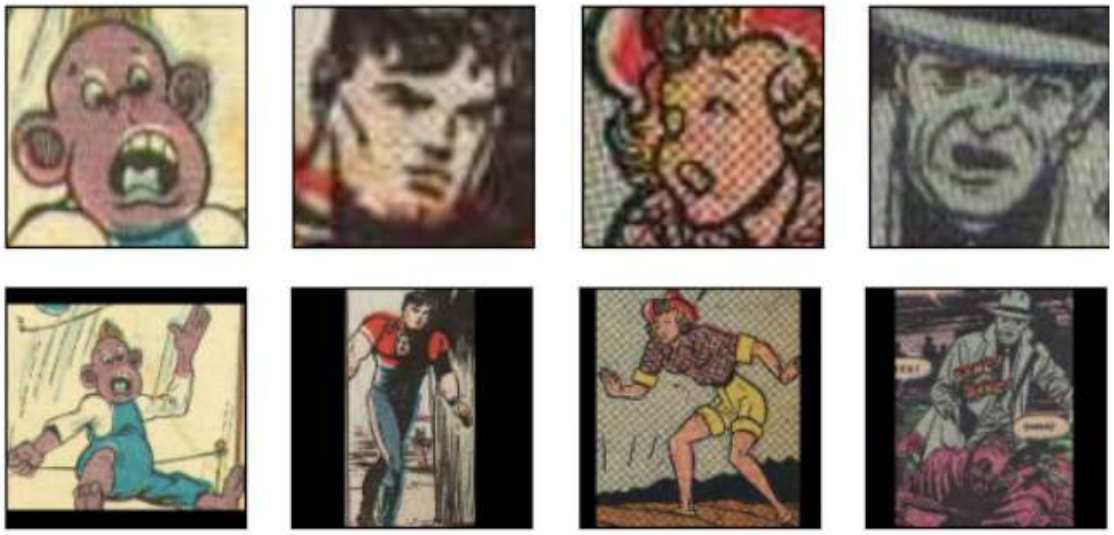


Figure 4.1: Four face and body samples from the dataset, *Comic Character Instances Dataset*, were used to pre-train the self-supervised backbone. As can be seen, there are various types of characters ranging from animals to superheroes.

To briefly summarize SimCLR, it learns powerful representations from unlabeled data by leveraging contrastive learning and strong data augmentation. It introduces a contrastive loss function, NT-XEnt, that encourages the network to bring similar samples closer while pushing different samples apart in a learned embedding space. This is achieved by maximizing the agreement between positive pairs and minimizing the agreement between negative pairs. With carefully designed training procedures, it is likely to produce high-quality representations. SimCLR has proven useful and outperformed previous methods in various computer vision tasks, demonstrating its ability to learn meaningful representations without relying on large labeled datasets.

$$l_{ij}^{\text{NT-Xent}} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{4.1}$$

$$l_{ij}^{\text{NT-Xent}} = -\text{sim}(z_i, z_j)/\tau + \log \left(\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau) \right) \tag{4.2}$$

Self-supervised Models: Identity-Aware SimCLR

In literature, comic character recognition is usually done by the characters' faces. The face-body combination module uses face and body information in Unsupervised Manga Character Re-identification [Zhang et al., 2022]. However, they train separate networks for each part and use each part's information to enhance their soft labels by comparing the probabilities of classes coming from face and body modules. Although it is one way to do it, they are not combining features coming from the face and body.

To solve those issues, inspired by the contrastive learning approaches. I propose **Identity-Aware SimCLR** framework, in which the main framework can also be used with other self-supervised learning algorithms.

The Identity-Aware SimCLR approach uses a shared network architecture to perform three tasks aiming to capture visual similarities within specific image parts and encode identity-related information.

The first task involves contrastive training of comic face images with strong augmentations, following the augmentation strategy employed in SimCLR. The network learns to produce representations of comic book faces by maximizing the similarity between augmented versions of the same face and minimizing the similarity between different faces.

The second task does the same for body images. However, the augmentations are the same except for the image dimensions for resizing. I use 96x96 images for the face, and for the body, it is 128x128.

The third task introduces pairs of face and body images depicting the same character. These images undergo weak transformations to ensure that the embeddings of both face and body images exhibit similarity. This encourages the network to encode identity-related information in face and body representations.

By jointly optimizing these tasks using the NT-Xent loss, the Identity-Aware SimCLR approach enables the network to learn representations that capture visual similarities within face and body parts while incorporating identity-related information across them.

In other words, the Identity-Aware SimCLR approach learns to identify visual features that are unique to faces and bodies, as well as features that are shared between faces and bodies. This allows the network to distinguish between different characters, even if they are only partially shown, and encode aligned identity features for face and body features, enabling the smooth fusion of face and body features with summation operation. The loss function for Identity-Aware SimCLR can be seen in Equation 4.6.

Algorithm 3 Strong augmentations used with Identity-Aware SimCLR.

Input: N, min_scale
Output: Strong augmentations
Procedure StrongAugmentations
transforms $\leftarrow$ Compose([RandomHorizontalFlip(),
RandomResizedCrop(size $= N$, scale $=$ (min_scale, 1)),
RandomApply([ColorJitter($brightness = 0.5, contrast = 0.5, saturation = 0.5, hue = 0.1$)], $p = 0.8$),
RandomGrayscale($p = 0.2$),
GaussianBlur(kernel_size $= 9$),
ToTensor(),
Normalize((0.5,), (0.5,))])
Return transforms

$$\mathcal{L}ij^{f} = -\log\left(\frac{\exp(\text{sim}(z_i^f, z_j^f)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i] \exp(\text{sim}(z_i^f, z_k^f)/\tau)}\right) \tag{4.3}$$

$$\mathcal{L}ij^{b} = -\log\left(\frac{\exp(\text{sim}(z_i^b, z_j^b)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i] \exp(\text{sim}(z_i^b, z_k^b)/\tau)}\right) \tag{4.4}$$

$$\mathcal{L}ij^{id} = -\log\left(\frac{\exp(\text{sim}(z_i^{f_w}, z_j^{b_w})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i] \exp(\text{sim}(z_i^{f_w}, z_k^{b_w})/\tau)}\right) \tag{4.5}$$

Algorithm 4 Weak augmentations used with Identity-Aware SimCLR. N is 96 for faces and 128 for bodies. *min scale* is 0.2 for faces and 0.5 for bodies.

Input: N, min_scale
Output: Weak augmentations
Procedure WeakAugmentations
transforms $\leftarrow$ Compose([LongestMaxSize(max_size $= N$),
PadIfNeeded(min_height $= N$, min_width $= N$, value $= (0, 0, 0)$),
RandomResizedCrop(size $= N$, scale $=$ (min_scale, 1.0), $p = 0.25$),
HorizontalFlip($p = 0.5$),
ShiftScaleRotate(shift_limit $= 0.05$, scale_limit $= 0.05$, rotate_limit $= 15$, $p = 0.5$),
OneOf([GaussianBlur(blur_limit $= (5, 7)$, $p = 1$),
MotionBlur(p=1),MedianBlur(blur_limit $= 3$, $p = 1$),
Blur(blur_limit=3, p=1),], $p = 0.1$),
ToGray($p = 0.2$),
Normalize($(0.5,), (0.5,)$),
ToTensor(),])
Return transforms

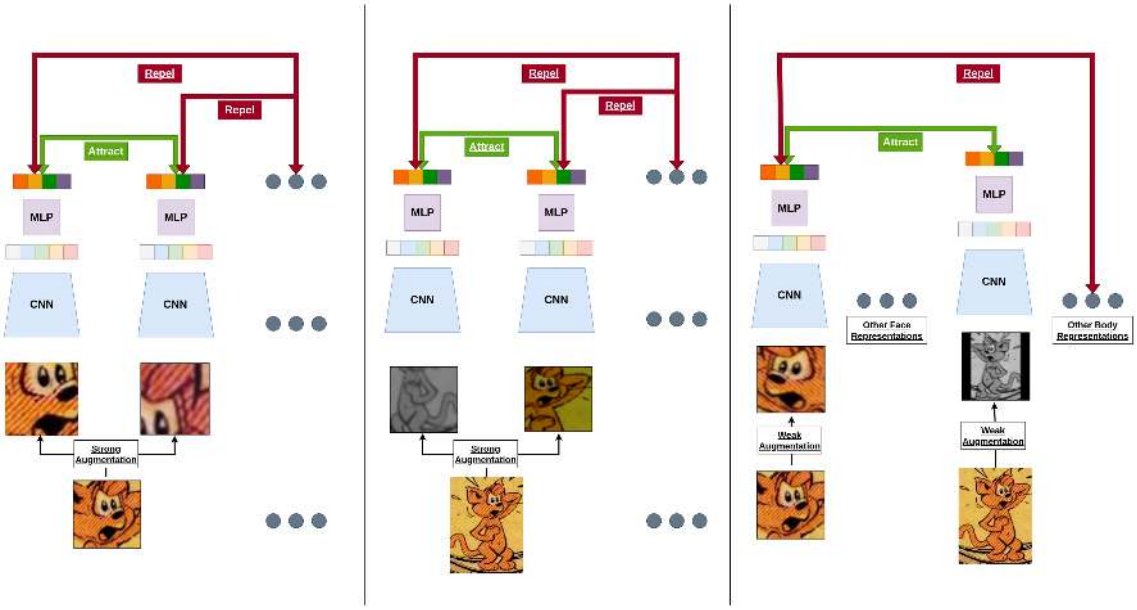


Figure 4.2: Overview of the *Identity-Aware SimCLR* approach. The same network architecture is utilized for three tasks, each employing the NT-Xent loss. From left to right: (1) Contrastive training of comic face images with strong augmentations, following SimCLR's augmentation strategy. (2) Contrastive training of comic body images with strong augmentations. (3) Contrastive training of face and body pair images with weak transformations, ensuring that the embeddings of face and body images depicting the same character exhibit similarity. This encourages both embeddings to encode identity-related information.

The overall function for identity-aware NT-Xent loss is given by:

$$\mathcal{L}ij = \mathcal{L}ij^{f} + \mathcal{L}ij^{b} + \mathcal{L}ij^{id} \tag{4.6}$$

Training Details

The training details for the Identity-Aware SimCLR approach are summarized as follows:

- Hardware Configuration: The training process utilizes two NVIDIA V100 GPUs, each with a 32GB configuration.
- Parallelization Strategy: The Distributed Data-Parallel (DDP) strategy is employed, which enables efficient training across multiple GPUs with synchronized batch normalization.
- Batch Size: A batch size of 320 is used for each GPU, resulting in a total effective batch size of 640.
- Training Epochs: The models are trained for a maximum of 500 epochs.
- Gradient Clip Value: A gradient clip value 0.5 is applied to prevent the gradients from exploding during training.

The model architecture used in the Identity-Aware SimCLR approach is based on ResNet50 [He et al., 2016]. A projection head replaces the final fully connected layer of ResNet50. The projection head is designed to be deeper, specifically a 3-layered MLP, as suggested by SimCLRv2 [Chen et al., 2020b]. It is capable of normalizing projections with L2 Norm. The output dimension of the projection head is set to 128.

The optimizer used was AdamW [Loshchilov and Hutter, 2017], with a learning rate 5e-4 and weight decay of 0.01. The learning rate was scheduled using the Cosine Annealing scheduler with a minimum learning rate of lr / 50. The temperature value

for the loss function is set to 0.07. It is worth noting that different temperature values, such as 0.5, were experimented with, but they resulted in significantly worse performance in terms of test loss and top-k accuracy values.

Two primary datasets are utilized regarding the datasets used in training, as explained in paragraph 4.3.1. One dataset includes all face-body pairs, while the other contains 100,000 faces and 100,000 bodies. During training, the train-val-test split ratios are set to 0.92, 0.04, and 0.04, respectively.

4.3.2 Semi-supervised Learning with Metric Learning for ReIdentification

Datasource and Preperation

Table 4.3: Statistics of the *Comic Sequence Identity Dataset*, utilized for fine-tuning with metric learning in character re-identification task.

Sequence	**Char. Instance**	**Face**	**Body**	**Char. Id.**	**Avg. Char. in Seq.**	**Avg. Char. Id. in Seq.**
3169	13716	10919	13359	6720	4.33	2.12

The *Comic Sequence Identity Dataset* aims to provide a collection of character instances with varying facial and bodily similarities or dissimilarities within a sequence of 4 panels. This dataset aims to strike a balance between easily annotated character identities and create a valuable resource for effective machine learning of characters, ultimately contributing to solutions for character recognition and reidentification tasks. The statistics of the dataset are presented in Table 4.3, highlighting that each sequence contains more than two unique characters, with an average of over four instances per sequence. This curation methodology aligns well with the requirements of metric learning, which involves accurately measuring similarity or dissimilarity between data points.

It can be challenging to perform dataset evaluation by splitting it into train-validation-test sets for two main reasons. Firstly, there is inter-series variance, which

means that drawing styles, character depictions, and even the quality of digital copies may differ across different comic series. Secondly, the data collection process involves acquiring character identities in an intra-sequence discriminative manner. Different character identities assigned to two sequences from the same series may refer to the same character. The handling of this issue is explained in the mining strategy section of the thesis.

To address these challenges, the following strategy is applied for splitting the data:

- Group the sequences by their comic series ID.
- Sort the sequences in ascending order based on the number of sequences for each series.
- Begin including series and their sequences, starting from the one with the fewest sequences.
- Repeat this process until the 800-sequence threshold is reached.
- Randomly sample two sets from the included sequences as validation and test splits.
- Use the remaining sequences as the training split.

By employing this strategy, series with a larger number of sequences as training samples can be utilized. This increases the likelihood of capturing various face and body gestures within a series. On the other hand, including a substantial number of different series in the validation and test splits enables the assessment of the model's generalizability, as these series are not included in the training and consist of a diverse range of comic series.

Augmenting identity annotations by linking sequences.

To utilize the dataset more efficiently and extract more information, I increased the number of similar-dissimilar pairs that can be obtained by linking characters between

sequences. One factor that facilitates this is the distribution of sequences belonging to a series in the train split using our train-validation-test split method; thus, there can be an intersection of panels for sequences. The identity annotation augmentation can be explained as follows: each character instance has a unique UUID independent of the character identity. Thanks to these UUIDs, even if the quartet sequences are different, I can transfer inter-sequence identity over the character instances in the intersection panels if there are one or more panels at their intersection.

To give an example, let sequence 1 be $N-3$, $N-2$, $N-1$, N_{th} panels, and sequence 2 be N, $N+1$, $N+2$, $N+3$. At the intersection of the two sequences, there is the panel N. For character A in the panel N, if there are n instances of the same character in sequence one and m in sequence 2, also, if there are l different characters in sequence one and k in sequence 2. Without augmentation, the number of similar pairs for sequence one would be $\binom{n}{2}$ pairs, and for sequence 2, it would be $\binom{m}{2}$ pairs. However, it would result in $\binom{m+n}{2}$ pairs with augmentation. Similar logic can be applied to dissimilar pairs. Without augmentation, sequence one would have $m \times k$ dissimilar pairs, and sequence two would have $n \times l$ dissimilar pairs, while with augmentation, it would be possible to use $(m+n) \times (k+l)$ pairs. This number is clearly more than the case without augmentation.

A meta-mining strategy for metric learning.

Mining in metric learning is the process of selecting or filtering samples, such as pairs, triplets, and groups of elements, depending on the loss function, to train the model in the most informative way possible. However, in my thesis, I implemented a **meta-mining** strategy that precedes the mining strategy. This choice was made because the character identity labels used in this study are not hard labels, meaning they are not definitive but rather discriminative within a sequence. Despite being augmented by linking, these labels are not guaranteed to be definitive. Consequently, using annotated character identities directly in the mining process could potentially lead to catastrophic errors during training. For example, consider the case of two character identities, A and B, from disconnected sequences, which may actually

belong to the same identity. To mitigate such issues, the meta-mining strategy consists of the following key steps:

- Initially, I identify all the longest possible dissimilar character identity lists for a given character identity. Each group of dissimilar characters is assigned a unique ID. This process can be conceptualized as a graph, where each node represents a character identity, and edges correspond to annotations within the dataset that indicate dissimilarity between characters. Given a node in the graph, I determine all possible nodes, cliques, that are directly connected to it.

- Within each dissimilar character group, I assign each character instance to its parent character identity as a class, serving as identifiers for the characters.

- Next, I extract the associated embeddings from the model for each character instance, utilizing features such as their face, body, or combination.

- Subsequently, I apply a selected mining algorithm to the embeddings, aiming to identify the most informative similar and dissimilar pairs within each subset. The output of the mining algorithm provides tuples or triplets of indices that represent these pairs, capturing their inherent relationships.

- Finally, I combine all the indices corresponding to a minibatch to construct a comprehensive collection of similar and dissimilar pairs for the entire batch. This expanded set of pairs enables the extraction of more informative samples and facilitates more effective character recognition and re-identification.

By employing this meta-mining strategy, I can enhance the mining process by leveraging the relationships between characters across different sequences and effectively linking the characters between sequences. Consequently, this approach enables the extraction of a larger quantity of meaningful pairs, enhancing the quality and quantity of information available for character recognition and re-identification. This contribution significantly improves the overall performance and effectiveness of the metric learning framework employed in my thesis.

Accuracy Metrics

The following evaluation metrics are commonly used in information retrieval, and they describe embedding space well [Musgrave et al., 2020a, Musgrave et al., 2020b].

- **Mean Average Precision (MAP)**: The rank positions of relevant documents, denoted as $K_1, K_2, \ldots, K_R$, are taken and then the Precision@K for each rank position is computed. The average precision is then obtained by averaging the *Precision@K* values. Thus, MAP is the mean of all average precisions.

- **Mean Average Precision at R (MAP@R)**: Average of the precision values at each rank position up to R. For a single query, it is calculated as below, and P(i) represents the precision at rank position i, with a value of 1 if the ith retrieval is correct, and 0 otherwise:

$$\text{MAP@R} = \frac{1}{R}\sum_{i=1}^{R} P(i)$$

- **Mean Reciprocal Rank (MRR)**: The first relevant document's rank position, K, is crucial for this metric. The Reciprocal Rank (RR) score is calculated as the reciprocal of the rank position, given by $\frac{1}{K}$. The MRR is then computed as the average RR score across multiple queries.

- **Precision@1 (P@1)**: Evaluation of whether the first nearest neighbor is correctly identified.

- **R-precision (R-P)**: It measures the precision of the top-R retrieved items, where R corresponds to the total relevant items in the dataset. It is calculated as the ratio of the number of relevant items among the top-R retrieved items to R.

$$\textit{R-precision} = \frac{\text{Number of relevant items in the top-}R\text{ retrieved items}}{R}$$

Evaluation Strategies

In Paragraph 4.3.2, I briefly explained the accuracy metrics suitable for the tasks. However, they alone do not necessarily indicate the models' performance. Instead, how they are employed makes a difference. With that respect, I came up with two different evaluation strategies. They are local (in-sequence) and global (query-reference) approaches. These strategies consider different perspectives and considerations when evaluating the models' effectiveness.

- **Local (In-Sequence)**: This strategy measures accuracy metrics in a way that they are applied to character instances in a sequence. What I mean by a *sequence* is the collection of character instances in which it is known they belong to different identities definitely. How such combinations of characters are found is explained in Paragraph 4.3.2. However, most simplistically, it can be thought of as a measurement for characters within four sequential panels; hence, on average, it includes on average four instances of characters belonging to 2 different identities (see Table 4.3). The reason why I employ this strategy is to be able to show the performance of our models for character reidentification across a series of sequential panels. Local performance evaluation is significant for the task, reflecting the spatiotemporal character reidentification performance.

- **Global (Query - Reference)**: This strategy is employed to evaluate the discriminative power models more broadly. So, instead of measuring the accuracies over a sequence, a set of query and reference character instances are selected from all the comic series in the test set. For a given query character, its pair or pairs are attempted to be found or retrieved by their similarity (L2 or cosine). Query and reference sets are constructed as follows: For each series in the test dataset, the algorithm identifies the longest suitable character combination (based on the sum of character instance lengths) for each character identity within the series. Once the longest combination is found, for every character within, one instance is assigned to a query, while the remain-

ing instances are assigned as references. If a character has only one instance, it is directly assigned as a reference. The test dataset partition contains 134 comic series comprising 303 distinct character identities. The dataset consists of **126 query instances** and **361 reference instances**. Out of the 126 queries, there are 184 references associated with the same character identity, meaning there are almost 1.5 reference pairs for a given query instance. The remaining 177 characters have only one instance each, intentionally included in the reference set to make it more challenging.

Experimental Setups and Training Details

Three main experimentation setups were devised. The first one involves evaluating variations of SimCLR [Chen et al., 2020a], including models trained specifically for identity-aware character self-supervision over datasets involving face, body, or both. This setup allows the assessment of the effectiveness of the identity-awareness method and compares its performance against vanilla SimCLR models. I aim to make a meaningful and measurable comparison of self-supervised backbone selection.

The second setup's focus is on the comparison of the proposed method with other baselines. Those baselines include a ResNet50 [He et al., 2016] backbone, extracting color histogram vectors, random baseline along with two variations of representations from face-body identity aligned model.

The third experimental setup focuses on keeping the backbone selection constant while investigating various strategies for mining, metric loss functions, and the utilization of representations, etc., for character reidentification. This setup serves as a hypothetical scenario to understand which factors contribute to better character reidentification performance while holding the backbone selection fixed.

Comparing Different Backbones and Data Types: This is the first setup. In this setup, four models were fine-tuned. Two of them have a vanilla SimCLR backbone. If the model was trained only on the face using vanilla SimCLR, it is referred to as the *Face Only* model. On the other hand, if the model was trained only on the bodies, it is called *Body Only* model. When using these two models

without training identity-awareness to obtain representations from both the face and body, I refer to it as *Face and Body Separately*. This model did not undergo identity-aware training, but rather, I fused the representations trained face-only and body-only models. The third fine-tuned model can be considered a baseline model for identity-awareness validation. It was trained simultaneously on the face and body backbones but without using identity-awareness loss. I refer to this model as *Face and Body Unaligned*. Similarly, if the identity-awareness loss was used with the same settings during the backbone training, it is called *Face and Body Aligned*.

In the experiments, I trained those models on datasets consisting of only face data, only body data, or a combination of both. I have described the nature of the datasets in Paragraph 4.3.2.

The training details of this fine-tuning process are as follows. The training process incorporates the use of *Triplet-Margin loss* [Balntas et al., 2016] with a margin value of 0.1, utilizing the L2 distance metric. To facilitate the mining process, *Multi-Similarity Miner* [Wang et al., 2019c] is employed with an ϵ value of 0.1. The training process consisted of a maximum of 20 epochs. Early stopping is applied based on the MAP@R metric for the validation set; the gradient clip value was set to 0.5. The model utilized a 256-dimensional linear layer to be fine-tuned. The final identity embeddings were L2 normalized. The linear layer is appended to the middle layer of the projection head of the SimCLR backbone model. The learning rate was set to 0.00075, and weight decay was set to 0.05. The training employed a scheduler gamma of 0.95 with an exponential LR scheduler. When both face and body are included, the fusion strategy was summing the embeddings from each. That ensured the fine-tuned linear layers (identity projector) for a single data type or face and body data had the same number of parameters. The data module included intense image transformations, such as various blurring operations, random resized cropping, flips, shifting, scaling, and rotating, with an image dimension of 128x128 for the body and 96x96 for the face. The same training settings were used as a template for all other fine-tuning experiments. The updated hyper-parameters are stated in experimental setups.

Baseline Experiments The baseline models used for comparison over face and body data are as follows:

- **Random**: This baseline involved generating a random vector of size 256 from a normal distribution for each image. This baseline is essential for evaluating the effectiveness of the learning process.

- **Color Histogram**: For each image, the color histogram of each color channel is computed and then concatenated to create a single vector. The bin size was 32, so the resulting vector was of size 96. This baseline provides a representation of the color distribution in the images.

- **Pre-trained ResNet50**: In the self-supervision phase of the training, a ResNet50 backbone was randomly initialized. To demonstrate the effectiveness of my approach, a ResNet50 pre-trained on the ImageNet-1k dataset [Deng et al., 2009] is used. I replaced the last fully connected layer of the pre-trained model with a trainable linear layer of output size 256. The backbone was frozen, making this model identical to the identity-aware self-supervised model. By comparing the performance of this pre-trained model with the proposed approach, the impact of the method can be evaluated.

- **Face & Body Aligned As-Is**: With this baseline no further training was applied. The self-supervised backbone was used as-is, keeping the self-supervision projector in place. Since the identity-awareness approach was taken, it is expected to have good discriminative power for reidentification tasks. However, since the model is used as-is, the final identity representation is 128-d.

- **Face & Body Aligned - Training Last Linear**: The difference between this approach and as-is was that the final linear layer of the model was not frozen, and a fine-tuning process was applied.

- **Face & Body Aligned**: This is the model used in *Comparing Different Backbones and Data Types* approach; the intermediate representations from

the SSL-trained model are fed to a linear projector of output size 256 to fine-tune the model.

These baselines provide a basis for evaluating and comparing the performance of our identity-aware semi-supervised model in various scenarios.

Exploring Strategies for Character Reidentification with Fixed Backbone Selection In this section, I explain the aspects investigated to understand which factors affect the model's performance.

Firstly, various loss functions and miner combinations are explored.

- Triplet-Margin loss [Balntas et al., 2016] is employed with the Multi-Similarity miner. Margin values of 0.2 and 0.5 are tested, along with their corresponding ϵ values. The L2 distance is used as the distance metric. The variables "a", "p", and "n" refer to the anchor, positive, and negative samples in Equation 4.7.

$$\mathcal{L}_{\text{triplet-margin}} = \frac{1}{N}\sum_{i=1}^{N}\max(0, d(a_i, p_i) - d(a_i, n_i) + \text{margin}) \tag{4.7}$$

- Contrastive loss [Hadsell et al., 2006]: The L2 distance metric is used without any mining function. In Equation 4.8 "y" is 0 for similar pairs and 1 for dissimilar pairs, penalizing any distance between similar pairs and pushing the dissimilar pairs by margin value.

$$\mathcal{L}_{\text{contrastive}}(y, d) = (1 - y) \cdot \frac{1}{2} \cdot d^2 + y \cdot \frac{1}{2} \cdot \max(0, \text{margin} - d)^2 \tag{4.8}$$

- Multiple Losses: Combination of Tuplet-Margin loss and Intra-Pair Variance loss [Yu and Tao, 2019] with weights of 1 and 0.5, respectively. No mining function is used.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{tuplet-margin}} + \frac{1}{2} \cdot \mathcal{L}_{\text{intra-pair variance}} \tag{4.9}$$

- Multi-Similarity loss with Multi-Similarity miner [Wang et al., 2019c]: α, β, and base parameters are set to 2, 50, and 0.5, respectively. L2 distance is used

for distance calculation, with an ϵ value of 0.1 for mining.

$$\mathcal{L}_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\sum_{k\in\mathcal{P}_i}e^{-\alpha(S_{ik}-\lambda)}\right] + \frac{1}{\beta}\log\left[1+\sum_{k\in\mathcal{N}_i}e^{\beta(S_{ik}-\lambda)}\right]\right\} \tag{4.10}$$

The Multi-Similarity miner selects negative pairs with similarity greater than the hardest positive pair minus a margin (ϵ) and positive pairs with similarity less than the hardest negative pair plus a ϵ.

Then, as a follow-up setup, by using the setting with contrastive loss, I conducted experiments to measure the effects of the following:

- Batch size, in my setting, batch size stands for the number of series that are used to sample suitable character combinations.

- Inclusion of sequences that only have single character identity. For some losses, it is required to have multiple identities, for those elements from other comic series are utilized.

- Trainable padding vectors. By default, when there is a missing component of a character, a face, or a body, a zero vector is used as padding. For this experiment type, padding vectors are trained as model parameters, initialized with Xavier uniform function (Glorot initialization) [Glorot and Bengio, 2010].

- Random masking of face or body. Since it is possible to have a missing face or body for a character, I experimented with randomly not using a face or a body when an element has both to see whether it would make the model more resilient to missing components and act as a regularization.

- Final linear layer output size. The final identity representation dimension of the model is changed to find the best output dimension.

- In the experiments, the fusion strategy for face and body embeddings was the summation of the vectors. However, to challenge that, experiments with the

following strategies were conducted: concatenation of vectors, using trainable parameters (α, β) for face and body features as coefficients that would sum up to 1, this is named as *Coeffiecent Sum*, using trainable parameters that has the same dimension of feature vectors to weight every single dimension of the features (*Weighted Sum*), and self-attention mechanism.

Another type of experimentation I did was augmenting the meta-mining strategy (see Paragraph 4.3.2). The main observation that led to this experiment was the performance of the trained models with miners. The loss functions used for this experiment were Triplet-Margin Loss and Multi-similarity loss. Though they performed well on local evaluation metrics, they had poor performance on global evaluation metrics. Because of that, for every character combination in a series, a single character instance from each other series within the batch was added to increase the global discrimination levels. This type of experiment is called *mix-series*.

4.3.3 Identity Assignment Across Panels with Character Clustering for ReIdentification

Within a 4-panel sequence (the decision of the number of panels is entirely arbitrary, as I have used 4-panel sequences in other parts of the research, and the method can accommodate an arbitrary number of panels or character count), the face and body images of the given characters are processed using a model fine-tuned with metric learning losses, outputting character identity embeddings. Then, these embeddings are fed to a clustering algorithm to determine which embedding belongs to which cluster.

The main challenge with clustering is that algorithms like K-means clustering, where the number of clusters is known in advance, are unsuitable since the number of different character identities in a given sequence is unknown. Therefore, I exploited methods such as DBSCAN [Ester et al., 1996] or Agglomerative Clustering [1]. Ultimately, I proceeded with Agglomerative Clustering (scikit-learn [Pedregosa

[1] Agglomerative Clustering in scikit-learn

et al., 2011] implementation), as DBSCAN is more effective when the sample size is large.

An important point to note is that a distance threshold for these methods must be provided. This threshold represents the maximum acceptable distance between two points or clusters. If a point is farther away, it is considered a member of a new cluster. Therefore, it is assigned to another character or a new identity. When determining the distance threshold, the statistics of distances for similar and dissimilar pairs in the test set can be calculated for the chosen model, and a decision can be made by comparing them, or one can use the ROC curve to decide the threshold. For example, in Section 4.4.5, the model I used (trained with Triplet-Margin loss + mix-series) has an average Euclidean distance of 0.77 for similar pairs and 0.97 for dissimilar pairs. Taking this into consideration, I selected a threshold value of 0.82.

I used AgglomerativeClustering with the "average" linkage criterion in the algorithm, which uses the average of the distances of each observation of the two identity sets. As a result of the clustering algorithm, assigned cluster numbers or character identities are obtained for each character embedding, thus for each character instance.

4.4 Results & Discussion

4.4.1 Self-supervised Learning of Comic Characters

This section presents the evaluation results of the self-supervised models trained on comic character images. The evaluation includes test metrics such as loss values, retrieval top-k (k=1 and k=5), and mean position based on cosine similarity within the batch. Although the ultimate measure of the utility of self-supervised models lies in their performance on downstream tasks, the chosen loss function, NT-Xent, aims to bring closer the projection features of transformed images belonging to the same source image. Hence, it is expected to be well-aligned with the downstream task of character reidentification using metric learning. Therefore, the results (see Table 4.4) obtained from self-supervised learning training provide key insights for

assessing the effectiveness of the models.

Table 4.4: Results of self-supervised training experiments conducted on face, body, and combining face and body for comic characters. The table presents recorded loss values, top-k accuracy metrics, and mean position for each task. The models served as the backbone for character re-identification tasks. Image dimensions were 96x96 for faces and 128x128 for bodies. The experiments were performed on a dataset of 100k samples, except for the full setting. Projections were subjected to L2 normalization. Training was conducted using a batch size of 320x2 on 2 GPUs. Note: * indicates that the F&B ALIGNED full experiment is not completed; the results shown are from the 81st epoch.

Task	**Loss**			**Top-1 Acc.**			**Top-5 Acc**			**Mean Position**		
	Face	Body	Identity	Face	Body	Identity	Face	Body	Identity	Face	Body	Identity
FACE	0.198	-	-	0.970	-	-	0.993	-	-	1.207	-	-
BODY	-	0.409	-	-	0.938	-	-	0.979	-	-	1.553	-
F. & B. UNALIGNED	0.194	0.398	6.540*	**0.977**	0.945	0.224	**0.995**	0.983	0.313	**1.119**	1.550	147.938
F. & B. ALIGNED	**0.192**	**0.368**	**0.207**	0.975	**0.949**	**0.973**	0.994	**0.986**	**0.995**	1.126	**1.490**	**1.094**
F. & B. ALIGNED FULL*	0.238	0.385	0.245	0.966	0.948	0.967	0.990	0.983	0.994	1.245	1.434	1.142

From the face results, it can be deduced that the models trained with face and body data exhibit superior performance in loss and top-k accuracy. Therefore, regardless of alignment, the availability of body data during training enhances face representations of the models. Nevertheless, the semi-supervised learning results may vary depending on the downstream task.

A similar pattern can be observed for performance comparison of the body results. Including face data during training leads to the improvement of body representations. The presence of face data enables the models to focus more on identity-related information, while the body data provides insights into the overall character structure. Although the results of the *Face and Body Unaligned* model in terms of identity is considerably lower than the aligned one, it still results in 0.224 top-1 accuracy given a batch size of 320 for a single GPU, supporting my claim.

To assess the effects of *Identity-Alignment*, it is best to compare the performance of the models in terms of their ability to capture identity-related information. The identity-aware module is far superior to its unaligned counterpart. During the training, they were fed with the same face and body dataset, and thus, this difference solely stems from *Identity-Awareness Loss*. It can be argued that the relatively better performance of the unaligned models, even without the identity-awareness loss, creates a favorable environment for the success of this additional loss.

The face performance of the unaligned model is slightly better than the aligned one. Comparatively, aligned is better in all metrics when it comes to body data. Considering that the identity-alignment procedure contributes more to distinguishing body features than the face in terms of self-supervision performance. However, it is essential to note that the identity-aligned model consistently outperforms the unaligned model in all evaluation metrics for the character reidentification task.

4.4.2 Semi-supervised Learning with Metric Learning Comparative Results for Different Backbones and Data Types

The results of the semi-supervision experiments can be interpreted in two main ways:

I examine the performance based on the specific data types used, such as face-

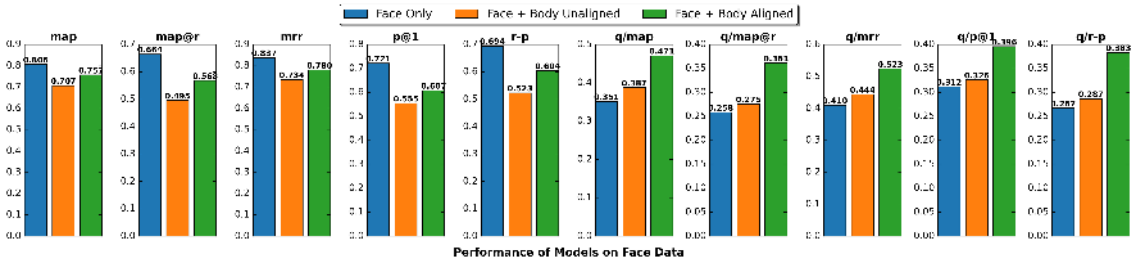


Figure 4.3: Performance comparison of the fine-tuned model using different self-supervised backbones on face data. Backbones are Face Only, Face + Body Unaligned, and Face + Body Aligned. Alignment indicates identity alignment. Metrics that start with *q/* indicate query evaluation representing *global* evaluation. The results highlight the superior performance of the Face + Body Aligned model across all global metrics, indicating the effectiveness of incorporating both face and body information. The face-only model is better in *local* evaluations.

only, body-only, or a combination of both. This analysis can be visualized by studying the figures (see Figures 4.3, 4.4, and 4.5) and comparing the horizontal trends. I refer to this as a "data type-dependent" analysis.

Regardless of the data type, I evaluate the general performance by comparing values on Figures 4.3, 4.4, and 4.5 vertically. This allows me to determine the most effective approach and data type for solving the problem under ideal conditions, where, hypothetically, all character instances in a comic book have both face and body data available.

I comprehensively evaluate comparative results for different backbones, trained with different approaches, separate, unaligned, or aligned self-supervision and data types by adopting both data type-dependent and data type-independent approaches. This analysis sheds light on the performance variations based on the specific data types employed and the overall effectiveness of different approaches and data types in addressing the character re-identification problem.

Data Type Dependent Discussion

For the face data, in local metrics, the model trained only on face data consistently outperforms the other models. However, interestingly, from the global perspective, the model trained with face and body data, specifically the aligned model, performs the best. Surprisingly, the unaligned model performs better than the model trained with only face concerning the global metrics. This difference in performance can be attributed to several factors. The **identity-awareness loss** may have enabled the model to learn more discriminative features related to character identities. By leveraging information from the face, the aligned model might have acquired specific contextual cues relevant to the entire character, leading to better encoding information such as body weight, drawing nuances, complexity, etc. On the other hand, the face-only model may excel at extracting **spatial and temporal context**, resulting in better performance in the local evaluation.

From the local perspective, although the aligned module performs better than the unaligned module, it falls short compared to the model's results trained exclusively on face data. Considering this, as well as examining the results of the unaligned module, it can be inferred that the backbone model trained with both face and body data is exposed to difficulties in extracting certain face features compared to the model trained solely on face data.

In contrast to the face evaluation, when considering the performance of models solely based on body data, there is no significant difference in the ranking of models in both local and global evaluations. Specifically, the aligned module outperforms the unaligned module, followed by the vanilla body model. This suggests that extracting identity information solely from the body is more challenging than the face. However, as the *face is seemingly the primary carrier of identity information*, when the model learns information related to the face from a training procedure where it can access faces and bodies, the performance improves even when only the body is used. With the incorporation of identity awareness, the model achieves its maximum potential in performance.

For face and body data, the trend observed in the evaluation using only face

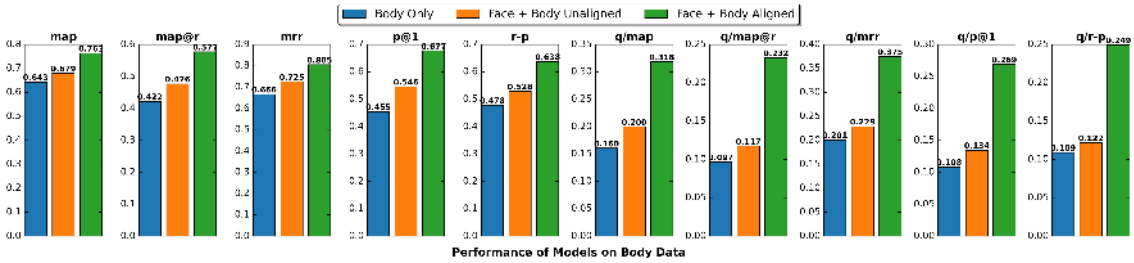


Figure 4.4: Performance comparison of the fine-tuned model using different self-supervised backbones on body data. Backbones are Body Only, Face + Body Unaligned, and Face + Body Aligned. Alignment indicates identity alignment. Metrics that start with *q/* indicate query evaluation representing *global* evaluation. The results highlight the superior performance of the Face + Body Aligned model across all metrics, indicating the effectiveness of incorporating both face and body information.

data holds true. Namely, separately trained SSL backbones with a fine-tuned linear layer for face and body data perform better than simultaneous training of faces and bodies. In parallel to that, the aligned module outperforms the unaligned module. Similarly, in the global context, the performance ranking is aligned, unaligned, and the separately trained models. However, the results show a notable and positive difference compared to the face-only evaluation. Specifically, there is no significant difference between the aligned and separate models regarding local results. For instance, the MAP values are nearly identical at 0.788, with a slight difference of 0.017 in MAP@R values.

I attribute this to the observation in analyzing the face-only data, where the vanilla face model is more likely to extract spatiotemporal information. The aligned model, although not performing as well as the vanilla model locally when considering only the face, achieves similar, thus better, results when both face and body data are utilized. This suggests that the aligned model effectively distributes the spatiotemporal information across the face and body. Therefore, utilizing two different sources of information, face and body together, allows the aligned model to

maximize its potential in both local and global contexts.

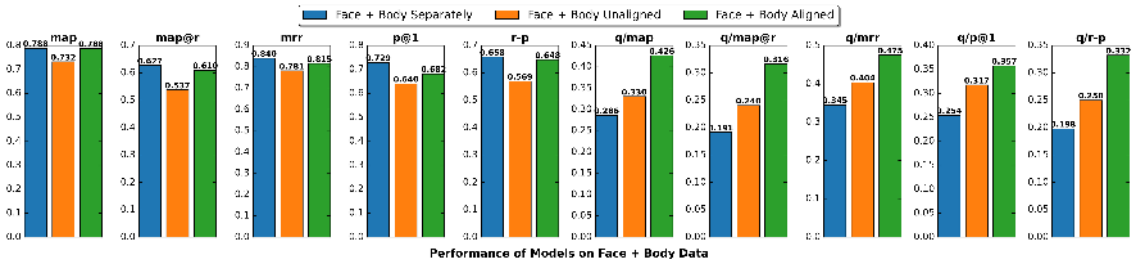


Figure 4.5: Performance comparison of the fine-tuned model using different self-supervised backbones on face and body data combined. Backbones are Face + Body Separately (Using two different SSL backbone models for each to get embeddings), Face + Body Unaligned, and Face + Body Aligned. Alignment indicates identity alignment. Metrics that start with *q/* indicate query evaluation representing *global* evaluation. The results highlight the superior performance of the Face + Body Aligned model across all global metrics, indicating the effectiveness of incorporating both face and body information. The face-only model is slightly better in *local* evaluations.

In conclusion, the aligned module consistently outperforms the unaligned module in both local and global evaluation scenarios, demonstrating the effectiveness of the identity-aware training approach. The aligned module achieves superior comic character reidentification performance by leveraging character-specific features and considering contextual information from both face and body data. These findings highlight the potential of identity-aware self-supervised training in enhancing comic character reidentification.

Data Type Independent Discussion

If results are evaluated in a data type-independent manner, the following observations can be made. In a scenario where all comic characters' face and body information are available for sequential reidentification, the best results are achieved by

using only the face data and training models exclusively on face data. This is evident from the higher metrics values such as mean average precision (MAP), MAP at R, and R-precision obtained on the face-only data compared to the face and body data evaluation results. On the other hand, when it comes to global reidentification, using only the face information and employing the face and body aligned model yields the best performance.

However, not all comic characters have detectable faces or bodies. In close-up drawings, characters may only have visible faces, while in some scenes, characters may be turned away or positioned at a distance, making their faces indistinguishable. In my analysis of this issue, I noticed that when a character's face is detected, there is a 98% chance of also detecting their body. However, when a character's body is detected, there is only an 80% chance of their face being present or detectable. Therefore, the performance on both face and body results must be considered when selecting an instrumental model. While the local performance of separate and aligned models is similar, the aligned model outperforms significantly in global evaluation. Furthermore, the aligned model has approximately half the parameters compared to the separate model since it incorporates a single backbone, which is advantageous regarding the model size.

4.4.3 Baseline Comparison

Table 4.5: Baseline models' results in terms of performance metrics for the local and global evaluation.

Model Name	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
Random	0.575 $\pm$ 0.002	0.306 $\pm$ 0.007	0.617 $\pm$ 0.005	0.356 $\pm$ 0.009	0.388 $\pm$ 0.009	0.022 $\pm$ 0.003	0.002 $\pm$ 0.002	0.028 $\pm$ 0.003	0.004 $\pm$ 0.005	0.003 $\pm$ 0.002
Color Histogram	0.582	0.327	0.628	0.4089	0.389	0.058	0.029	0.080	0.039	0.033
ResNet50	0.704	0.497	0.746	0.588	0.540	0.212	0.138	0.251	0.158	0.149
Aligned (As-Is)	0.668	0.462	0.703	0.498	0.530	**0.668**	**0.462**	**0.703**	**0.498**	**0.530**
Aligned (Last Layer)	0.742	0.531	0.780	0.621	0.575	0.450	0.352	0.533	0.452	0.359
Aligned	**0.787**	**0.609**	**0.815**	**0.681**	**0.647**	0.425	0.316	0.474	0.357	0.332

Table 4.5 presents the performance metrics of the baseline models evaluated in terms of local and global performance.

The Random model represents the minimum achievable performance for the tasks. All metrics are close to zero for global evaluation since there are 361 reference instances and 1.5 candidates for every query. As for the local evaluation, P@1 is also quite consistent, pointing to an average sequence consisting of 2 pairs of characters. So, for a given query, the valid candidate has around 1/3 chance of being selected randomly.

Regarding local evaluation, the Color histogram vectors achieved scores that indicate a poor performance compared to the other baselines as they are close to the random baseline. This suggests that more than color distribution information is needed for accurate comic character re-identification.

The ResNet50 model with linear layer is the direct competitor for the identity-aware semi-supervised approach. It even outperforms the Face and Body Aligned (As-Is) model for local evaluation. The As-Is model lacks fine-tuning with metric learning, whereas ResNet50 undergoes training. However, the As-Is model excels in all metrics for global evaluation, showcasing the effectiveness of the identity-awareness method.

The results changed remarkably when the same model's final linear was trained. This can be seen from the difference in results with Aligned (As-Is) and (Last Layer). The model performs better in local evaluation at the expense of global performance. This shows that performance on a global scale and local scale have different characteristics, and the metric learning pipeline focuses on local with the Triplet-Margin loss, reducing the model's global performance.

The same trend is available between the Aligned (Last Layer) and Aligned model, in which their only difference is that the final linear layer is trained from scratch instead of transferring weights from SSL.

The Aligned model, leveraging both local and global features, achieves the best performance among the baseline models in terms of local performance. However, learning spatiotemporal differences causes the model to 'forget' the differentiation of characters on the global scale.

In conclusion, the results confirm the effectiveness of the proposed approach, outperforming other baselines and providing insights about the difference between local

and global scales for further advancements in self-supervised and semi-supervised learning for image retrieval and re-identification tasks.

4.4.4 Exploring Strategies for Character Reidentification with Fixed Backbone Selection

This section explains the findings and results of various factors affecting the model's fine-tuning performance. This analysis helps to find out which training details and hyper-parameters work best.

Losses and Miners

Table 4.6: Fine-tuning performance comparison using different loss functions. Multi-Similarity Miner [Wang et al., 2019c] was employed during training, specifically for Triplet-Margin, with corresponding margin values provided and Multi-Similarity losses.

Loss	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
Contrastive	$0.777_{\pm 0.004}$	$0.584_{\pm 0.013}$	$0.804_{\pm 0.005}$	$0.653_{\pm 0.008}$	$0.627_{\pm 0.018}$	$\mathbf{0.646_{\pm 0.003}}$	$\mathbf{0.517_{\pm 0.021}}$	$\mathbf{0.699_{\pm 0.007}}$	$\mathbf{0.584_{\pm 0.021}}$	$\mathbf{0.529_{\pm 0.026}}$
Intra-Pair Variance	$0.740_{\pm 0.015}$	$0.535_{\pm 0.026}$	$0.765_{\pm 0.016}$	$0.591_{\pm 0.029}$	$0.586_{\pm 0.027}$	$0.597_{\pm 0.026}$	$0.450_{\pm 0.028}$	$0.646_{\pm 0.028}$	$0.517_{\pm 0.037}$	$0.463_{\pm 0.025}$
Triplet-Margin (0.2)	$\mathbf{0.796_{\pm 0.020}}$	$0.625_{\pm 0.039}$	$\mathbf{0.816_{\pm 0.022}}$	$0.681_{\pm 0.037}$	$0.660_{\pm 0.038}$	$0.400_{\pm 0.014}$	$0.289_{\pm 0.018}$	$0.462_{\pm 0.014}$	$0.350_{\pm 0.018}$	$0.301_{\pm 0.022}$
Triplet-Margin (0.5)	$0.794_{\pm 0.017}$	$\mathbf{0.638_{\pm 0.030}}$	$0.814_{\pm 0.017}$	$\mathbf{0.689_{\pm 0.028}}$	$\mathbf{0.672_{\pm 0.028}}$	$0.267_{\pm 0.015}$	$0.177_{\pm 0.020}$	$0.320_{\pm 0.028}$	$0.224_{\pm 0.047}$	$0.186_{\pm 0.018}$
Multi-Similarity	$0.784_{\pm 0.016}$	$0.597_{\pm 0.032}$	$0.811_{\pm 0.016}$	$0.671_{\pm 0.034}$	$0.638_{\pm 0.029}$	$0.479_{\pm 0.028}$	$0.362_{\pm 0.036}$	$0.530_{\pm 0.018}$	$0.411_{\pm 0.032}$	$0.376_{\pm 0.035}$

Table 4.6 presents the results of the fine-tuning experiments using different loss functions and miners.

Initially, comparing the results of loss functions with and without miners reveals the following observations: the losses Triplet-Margin and Multi-Similarity, which incorporate miners through the meta-mining strategy, demonstrate improved local performance. However, this improvement comes at the cost of a decrease in global performance. To address this trade-off, I conducted an additional experimental setup called *mix-series* (see Paragraph 4.4.4) in an attempt to enhance global performance. When no miner is employed, all characters can be paired with each other based on their labels. However, as discussed previously, this can result in negative pairs

that actually belong to the same identity due to the data collection method. This discrepancy could be one of the reasons why loss functions utilizing miners prior to the loss calculation yield superior local performance.

On the other hand, the contrastive loss function exhibits better performance in the global context. It demonstrates competitive results in the local context, indicating its effectiveness in capturing the relationships between face and body images and embedding them into a shared feature space. Similarly, the intra-pair variance loss follows a similar trend but with slightly lower performance. Considering the variance within character pairs during training may decrease model performance, particularly when some labels are inherently incorrect. Consequently, the intra-pair variance loss might be less noise tolerant than the contrastive loss.

The Triplet-Margin loss achieves the highest scores in the local evaluation metrics. However, when the margin values are increased, the performance on global metrics deteriorates further. On the other hand, the multi-similarity loss performs comparably well on the local metrics and demonstrates better performance on the global metrics.

Batch size

Table 4.7: Fine-tuning performance of contrastive loss function with respect to batch size.

Batch Size	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
16	$\mathbf{0.777}_{\pm 0.004}$	$0.584_{\pm 0.013}$	$0.804_{\pm 0.005}$	$0.653_{\pm 0.008}$	$\mathbf{0.627}_{\pm 0.018}$	$\mathbf{0.646}_{\pm 0.003}$	$\mathbf{0.517}_{\pm 0.021}$	$\mathbf{0.699}_{\pm 0.007}$	$\mathbf{0.584}_{\pm 0.021}$	$\mathbf{0.529}_{\pm 0.026}$
32	$0.773_{\pm 0.006}$	$\mathbf{0.585}_{\pm 0.013}$	$\mathbf{0.806}_{\pm 0.009}$	$\mathbf{0.660}_{\pm 0.019}$	$\mathbf{0.627}_{\pm 0.010}$	$0.645_{\pm 0.017}$	$0.511_{\pm 0.028}$	$0.694_{\pm 0.023}$	$0.568_{\pm 0.039}$	$0.526_{\pm 0.028}$
64	$0.766_{\pm 0.004}$	$0.574_{\pm 0.008}$	$0.797_{\pm 0.006}$	$0.644_{\pm 0.010}$	$0.618_{\pm 0.012}$	$0.637_{\pm 0.011}$	$0.503_{\pm 0.022}$	$0.677_{\pm 0.009}$	$0.552_{\pm 0.017}$	$0.520_{\pm 0.021}$

I aimed to analyze the results based on different batch sizes. However, it is essential to clarify that the term "batch size" used here does not refer to the standard definition but instead denotes the number of comic series utilized in each iteration. From each comic series, the combinations of labeled examples that are compatible with each other are selected. With this clarification in mind, results can be inter-

preted. In general, keeping the batch size small yields better results. While the best results are observed within the range of 16 and 32 in the local context, the results are quite close. In terms of the global perspective, the experiment with a batch size of 16 consistently provides the best results, but the results obtained with a batch size of 32 are close behind. Since these results are based on contrastive loss, I hesitate to generalize them to other loss functions. The contrastive loss was chosen because of its ability to provide the most consistent results in local and global contexts.

Single Char. Identity, Trainable Paddings, Random Masking

Table 4.8: Performance comparison of trainable face and body paddings, random masking of face and body, and addition of characters from sequences with only a single identity to base training setting on local and global metrics for contrastive loss function.

Method	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
Base	$\mathbf{0.777}_{\pm 0.004}$	$0.584_{\pm 0.013}$	$0.804_{\pm 0.005}$	$0.653_{\pm 0.008}$	$0.627_{\pm 0.018}$	$\mathbf{0.646}_{\pm 0.003}$	$\mathbf{0.517}_{\pm 0.021}$	$\mathbf{0.699}_{\pm 0.007}$	$\mathbf{0.584}_{\pm 0.021}$	$\mathbf{0.529}_{\pm 0.026}$
Trainable Padding	$0.775_{\pm 0.008}$	$\mathbf{0.588}_{\pm 0.012}$	$\mathbf{0.807}_{\pm 0.011}$	$\mathbf{0.659}_{\pm 0.015}$	$\mathbf{0.635}_{\pm 0.011}$	$0.606_{\pm 0.019}$	$0.460_{\pm 0.022}$	$0.651_{\pm 0.023}$	$0.516_{\pm 0.022}$	$0.475_{\pm 0.022}$
Random Masking	$0.768_{\pm 0.008}$	$0.573_{\pm 0.022}$	$0.797_{\pm 0.008}$	$0.639_{\pm 0.016}$	$0.620_{\pm 0.024}$	$0.630_{\pm 0.025}$	$0.492_{\pm 0.033}$	$0.681_{\pm 0.034}$	$0.563_{\pm 0.054}$	$0.503_{\pm 0.030}$
Single Ids	$0.742_{\pm 0.016}$	$0.545_{\pm 0.023}$	$0.771_{\pm 0.018}$	$0.603_{\pm 0.034}$	$0.597_{\pm 0.018}$	$0.579_{\pm 0.030}$	$0.427_{\pm 0.049}$	$0.630_{\pm 0.028}$	$0.487_{\pm 0.055}$	$0.440_{\pm 0.046}$

The results presented in Table 4.8 provide insights into the performance of different methods applied to the base training setting using the contrastive loss function. Other methods did not increase performance except for the trainable face and body padding method for missing images. Although trainable paddings showed minor performance upgrades in a local setting, on all global metrics, results declined. I argue that the success of 0 paddings is because the face and body embeddings from the self-supervised backbone model are summed in the base method, and the identity-awareness method in the SSL period tries to bring closer face and embeddings concerning cosine similarity. Therefore, adding a non-zero vector to either face or body embeddings can divert the identity information, whereas a 0 vector would not affect the direction of the vector. This may explain the drop in global evaluation metrics.

The random masking method, which introduces randomness by masking either face or body images given a character with both, demonstrates lower performance than the base method. The masking rates were 0.4 in total for face and body distributed equally. Instead of a regularization effect, this process increased noise, affecting the model's ability to learn meaningful representations. Thus, the best scenario for training is to have both faces and bodies of characters when both are available.

Lastly, the single identity method, including characters from sequences with only a single identity, yields the lowest performance across most evaluation measures. This suggests that a diverse range of identities in the training data is essential for capturing the model's generalizability and improving its performance.

Output Dimension

Table 4.9: Performance comparison of output dimension size on local and global metrics for contrastive loss function.

Output Dim.	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
64	$0.776_{\pm 0.007}$	$0.579_{\pm 0.006}$	$\mathbf{0.811_{\pm 0.015}}$	$\mathbf{0.664_{\pm 0.023}}$	$0.619_{\pm 0.005}$	$0.617_{\pm 0.023}$	$0.473_{\pm 0.032}$	$0.669_{\pm 0.019}$	$0.529_{\pm 0.026}$	$0.493_{\pm 0.033}$
128	$\mathbf{0.778_{\pm 0.012}}$	$\mathbf{0.588_{\pm 0.017}}$	$0.808_{\pm 0.015}$	$0.661_{\pm 0.025}$	$\mathbf{0.630_{\pm 0.013}}$	$0.627_{\pm 0.022}$	$0.490_{\pm 0.025}$	$0.677_{\pm 0.025}$	$0.552_{\pm 0.041}$	$0.506_{\pm 0.023}$
256	$0.777_{\pm 0.004}$	$0.584_{\pm 0.013}$	$0.804_{\pm 0.005}$	$0.653_{\pm 0.008}$	$0.627_{\pm 0.018}$	$\mathbf{0.646_{\pm 0.003}}$	$\mathbf{0.517_{\pm 0.021}}$	$\mathbf{0.699_{\pm 0.007}}$	$\mathbf{0.584_{\pm 0.021}}$	$\mathbf{0.529_{\pm 0.026}}$
512	$0.768_{\pm 0.009}$	$0.570_{\pm 0.015}$	$0.799_{\pm 0.013}$	$0.645_{\pm 0.029}$	$0.616_{\pm 0.018}$	$0.624_{\pm 0.030}$	$0.495_{\pm 0.043}$	$0.673_{\pm 0.030}$	$0.556_{\pm 0.043}$	$0.508_{\pm 0.041}$

The following observations can be made by referring to Table 4.9. The output dimension of the linear layer, which is added on top of the SSL backbone and fine-tuned, does not appear to be a significant factor affecting local performance within the range of 64 to 512, favoring the lower. However, for the global evaluation, an output dimension of 256 performs slightly better than the others, albeit with a marginal difference. Considering the disparity between local and global evaluations, it can be inferred that a lower parameter count may suffice for in-sequence character reidentification. In contrast, for a robust discriminator with strong global discriminative power, the model output size should be 256.

Fusion Strategies

Table 4.10: Performance comparison of fusion methods for face and body features on local and global metrics for contrastive loss function.

Fusion Method	**Local**					**Global**				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
sum	**0.777**$_{\pm 0.004}$	0.584$_{\pm 0.013}$	0.804$_{\pm 0.005}$	0.653$_{\pm 0.008}$	**0.627**$_{\pm 0.018}$	**0.646**$_{\pm 0.003}$	**0.517**$_{\pm 0.021}$	**0.699**$_{\pm 0.007}$	**0.584**$_{\pm 0.021}$	**0.529**$_{\pm 0.026}$
cat	0.771$_{\pm 0.012}$	**0.589**$_{\pm 0.024}$	**0.810**$_{\pm 0.007}$	**0.676**$_{\pm 0.018}$	0.623$_{\pm 0.023}$	0.611$_{\pm 0.013}$	0.470$_{\pm 0.016}$	0.660$_{\pm 0.018}$	0.534$_{\pm 0.019}$	0.482$_{\pm 0.012}$
weighted sum	0.771$_{\pm 0.014}$	0.571$_{\pm 0.023}$	0.804$_{\pm 0.018}$	0.651$_{\pm 0.036}$	0.616$_{\pm 0.023}$	0.633$_{\pm 0.017}$	0.493$_{\pm 0.022}$	0.682$_{\pm 0.014}$	0.556$_{\pm 0.021}$	0.509$_{\pm 0.020}$
coeff. sum	0.656$_{\pm 0.011}$	0.460$_{\pm 0.024}$	0.690$_{\pm 0.018}$	0.499$_{\pm 0.030}$	0.518$_{\pm 0.023}$	0.299$_{\pm 0.044}$	0.222$_{\pm 0.053}$	0.356$_{\pm 0.061}$	0.272$_{\pm 0.071}$	0.231$_{\pm 0.054}$

Table 4.10 comprehensively compares fusion methods used to combine face and body features within the framework of the contrastive loss function.

The results indicate that the simple vector addition method (sum) achieves the highest scores in the global evaluation setting. Interestingly, in the local evaluation, the sum method and the concatenation of features method (cat) perform comparably well, despite the cat method having twice the parameter size of the sum method for the linear projector. This finding suggests that the effectiveness of fusion methods extends beyond the consideration of parameter size alone.

The superior performance of the summation fusion method can be attributed to its ability to effectively capture the complementary information and interactions between the face and body modalities. By combining the features additively, the summation fusion method facilitates better integration with the aligned features, leveraging the identity-aware self-supervised backbone.

In contrast, the weighted sum fusion method produces comparable results to the sum method but falls behind in all metrics. This observation suggests that all features derived from the backbone exhibit relatively equal importance, as the weighted sum method was designed to train two vectors with the same size and employ softmax to perform element-wise multiplication with the face and body features.

Finally, the fusion method involving two trained scalar coefficients before summation (coeff. sum) demonstrates the lowest performance across all metrics, indicating

its limited effectiveness in combining face and body features for the contrastive loss function.

Besides these techniques, self-attention-based mechanisms were also applied. For instance, one of the attempts was to use face embeddings as a query to get attention values over keys and values of face and body embeddings. This, in essence, was a dynamic version of coefficient summation. However, it did not produce promising results.

Mix Series for Meta-mining

Table 4.11: Performance comparison of mix-series for the loss functions that use miners on local and global metrics.

Method	Local					Global				
	map	map@r	mrr	p@1	r-p	map	map@r	mrr	p@1	r-p
Triplet-Margin	**0.796**$_{\pm 0.020}$	**0.625**$_{\pm 0.039}$	0.816$_{\pm 0.022}$	0.681$_{\pm 0.037}$	**0.660**$_{\pm 0.038}$	0.400$_{\pm 0.014}$	0.289$_{\pm 0.018}$	0.462$_{\pm 0.014}$	0.350$_{\pm 0.018}$	0.301$_{\pm 0.022}$
+ mix series	0.790$_{\pm 0.015}$	0.614$_{\pm 0.021}$	**0.817**$_{\pm 0.017}$	**0.684**$_{\pm 0.023}$	0.651$_{\pm 0.022}$	**0.619**$_{\pm 0.030}$ ↑↑	**0.483**$_{\pm 0.050}$ ↑↑	**0.662**$_{\pm 0.029}$ ↑↑	**0.544**$_{\pm 0.048}$ ↑↑	**0.493**$_{\pm 0.052}$ ↑↑
Multi-Similarity	0.784$_{\pm 0.016}$	0.597$_{\pm 0.032}$	0.811$_{\pm 0.016}$	0.671$_{\pm 0.034}$	0.638$_{\pm 0.029}$	0.479$_{\pm 0.028}$	0.362$_{\pm 0.036}$	0.530$_{\pm 0.018}$	0.411$_{\pm 0.032}$	0.376$_{\pm 0.035}$
+ mix series	0.789$_{\pm 0.021}$	0.597$_{\pm 0.041}$	0.812$_{\pm 0.023}$	0.670$_{\pm 0.044}$	0.635$_{\pm 0.039}$	0.591$_{\pm 0.023}$ ↑	0.452$_{\pm 0.049}$ ↑	0.628$_{\pm 0.030}$ ↑	0.500$_{\pm 0.063}$ ↑	0.464$_{\pm 0.043}$ ↑

The results presented in Table 4.11 highlight the impact of the mix-series method on the performance of the triplet-margin and multi-similarity loss functions in conjunction with the mining process.

The mix-series approach involves adding a character instance from every other series in the batch to the mining pool, allowing for the selection of triplets that mitigate the performance drop in the global evaluation.

Without the mix-series technique, the model trained with the Triplet-Margin loss function performs slightly better in the local evaluation. However, in a global context, the results are reversed, meaning multi-similarity loss performs better.

By enhancing the triplets with elements from other series, both models exhibit significantly improved performance in global evaluation metrics. This helps the model remember how to differentiate characters in a more general sense beyond just in-sequence discrimination. Importantly, this modification does not lead to any performance drops in the local evaluation.

The mix-series approach enhances the meta-mining strategy by incorporating instances from different comic series. This demonstrates the effectiveness of leveraging cross-series information to improve the discrimination capabilities of the models.

Considering all experiments, it can be concluded that the Triplet-Margin loss + mix-series method yields the best results in terms of local evaluation, while the Contrastive loss outperforms significantly in a global context. To achieve optimal results, it is advisable to use the base training setting. However, if there is a desire to improve local performance slightly, adjusting the output dimension size to 128 and the batch size to 32 can be considered, although this may slightly compromise global performance.

4.4.5 Qualitative Evaluation

In this section, qualitative evaluation is performed on 4-panel sequences. Both successful and problematic examples can be seen in Figures 4.6 and 4.7.

My observations are as follows: In the well-performing examples, characters generally have face and body representations, and their drawing distances are relatively consistent with the scene. Even if they are not the same, there is consistency in facial expressions.

The problematic examples help to understand the challenges faced by the model. However, identifying specific patterns is challenging. The following can be observed if the problematic examples are examined case by case. The numbering is aligned with the order of sequences in Figure 4.7:

1. The first two panels in the first sequence indicate that the model performs very well, even separating overlapping characters. However, in the third panel, the male character is assigned to the female identity in the first two panels. My assumption here is that since the female character does not have a body, the model might have made the similarity based on the skin color and complexity and the presence of a body. This might suggest that it is important for both face and body to be detected in the examples.

Figure 4.6: Examples of 4-panel sequences that the model demonstrates successful character re-identification.

2. The example where the model performs the worst in the sequence is when the child character turns around in the first panel and faces forward in the second panel. The model may have difficulty distinguishing complete rotations. For the other character, Negative outcomes can be based on the difficulty of determining identities only based on bodies and having a non-human character.

3. Here, the mistake made by the model could be the character both turning its back and changing clothes. Otherwise, the model successfully assigns the correct identity to the character instances.

4. In the last sequence, characters who are quite similar physically and in terms of their faces are presented but are assigned the same identity.

4.5 Conclusion & Future Work

I addressed the challenging problem of character reidentification across panels in comics. This work is one of the few that explicitly targets character reidentification in comics and is the only one that fuses face and body information while doing that.

I employed semi-supervised learning combining self-supervised learning and metric learning techniques for character reidentification with a novel identity-aware self-supervised framework enabling simultaneous processing of both faces and bodies in comics, capturing critical visual cues for character reidentification. The challenge of scarce labeled character identity information in comics was overcome by leveraging semi-supervised learning with limited annotated data that I collected.

There are two main areas for future work. The first is to refine the comic character reidentification by several more advances, such as expanding the character identity annotation and maybe, instead of getting partial labels from comic sequences, getting complete identity information from comic series, which would lead to the training of more robust models. Another aspect of the reidentification task can be an exploration of multi-modal features for the tasks, such as textual information from speech bubbles. Since the speech bubble to character association is part

Figure 4.7: Examples of 4-panel sequences that highlight the challenges faced by the model for character re-identification.

of this thesis, these ideas can be incorporated into a more generalized multi-task learning framework that would also be used for character recognition and reidentification. Lastly, integrating temporal information from panels can enhance the models' capabilities.

Another promising direction for future work is the utilization of character identity information for various tasks beyond character reidentification. Incorporating character identity knowledge into scene-understanding tasks can provide a deeper understanding of the relationships and interactions between characters within a comic. Additionally, character identities can be leveraged for cloze-style tasks, where missing information about characters can be inferred based on their identities and the context of the comic. Furthermore, character identity information can be employed in generating comic texts and images, creating more coherent and contextually consistent narratives. Exploring these possibilities could enhance comics' comprehension, generation, and interactive aspects, opening up new avenues for research and creativity. I discuss these issues in more detail in the chapter 5.

Chapter 5

COMICBERT: A TRANSFORMER MODEL AND PRE-TRAINING STRATEGY FOR CONTEXTUAL UNDERSTANDING IN COMICS

5.1 Introduction

A foundation model for comic processing has not yet been proposed. The closest study to this is the method suggested in [Iyyer et al., 2017]. A multimodal sequential model is proposed with cloze-style tasks, but it is far from human-level performance due to both the dataset and method used.

Considering these, I am trying to establish a method that can be a foundation model and a self-supervised objective to pre-train it. The base model can be called a *Comicsformer*. Comicsformer is a transformer encoder structure that can process visual information of panels, speech bubble texts, narrative text, and ultimately character visual and identity information for sequences of comics panels. The self-supervision task that can train Comicsformer in order to make it a foundation model is called *Masked Comic Modeling*, inspired by the masked language modeling task in BERT [Devlin et al., 2019]. Thus, overall, the general framework can be called *ComicBERT*. To the best of my knowledge, I am the first to propose these tasks and models for comics in the literature.

I use the cloze-style tasks, visual-cloze, text-cloze, and character coherence, introduced in [Iyyer et al., 2017] to fine-tune the model and measure the success of it. In addition to these, I propose two new novel tasks that will be useful in comic book understanding. One of them is scene-cloze; unlike the cloze-style tasks I just mentioned, it tries to predict all of the elements in the next scene, not some of them. The other is contextual character to speech attribution; this task allows us to understand whether a given speech bubble context belongs to the existing character

when the sequential panel context is given.

I believe these new tasks, contextual character-to-speech attribution and scene-cloze, can become tools that will be useful to comics illustrators; they can be utilized to decide what a particular character has to say or what next panel would be given context information from previous n panels.

The Comicsformer structure and ComicBERT can be trained later with different tasks and information. It can be used to predict or process visual grammar information. Alternatively, the ComicBERT framework can be considered a universal comic processor and used as a foundation model, which is its primary motivation.

5.2 Related Work

Few works in the comic book processing field focus on the multimodal nature of comics. One such work is called *Manga-MMTL* [Nguyen et al., 2021a]. In that work, the authors propose a method to combine a comic character's visual features of faces and their corresponding text to extract multimodal character features. The learned character features are then evaluated by information retrieval methods to prove their effectiveness. However, in order to train their framework, one of the tasks is to classify comic albums, which may be unavailable for some datasets. Also, it is focused on the faces of the characters and dialogues only. Thus, it cannot provide context for the panel or the sequence of panels.

Another architecture that exploits the multimodal nature of comics is *MPDialog* [Agrawal et al., 2023]. In this work, persona-based dialogue generation for comic strips is proposed. Considering their overall architecture, this is the closest work to ours in terms of using a transformer-based architecture. However, they use a multimodal transformer decoder with panel image features, a detailed persona description text, and speech bubble content associated with identities. So, instead of using character embeddings as an input model, they are mapped to textual descriptions, which are hard to annotate. So, this can be a good candidate for the downstream task for comics. However, the evaluation is particularly hard and does not necessarily result in context embeddings out of the model to be used for other

tasks. Whereas the tasks proposed in [Iyyer et al., 2017] and the proposed tasks in this thesis force the model to learn contextual information of panel sequences and can be combined to extract generalized context features for comics.

Regarding the utilization of contrastive learning to enhance the representation capabilities of BERT-like models, I would like to discuss two previous works. The first one is the *TaCL (Token-aware Contrastive Learning)* framework proposed by Su et al. [Su et al., 2021]. Their approach employs two instances of the same pre-trained BERT variant: one instance is frozen and designated as the teacher model, while the other is trained using masked language modeling and next-sentence prediction tasks in conjunction with their proposed TaCL loss. TaCL incorporates contrastive learning between the representations of the student and teacher models; however, the input tokens of the teacher model are left unmasked. The authors suggest this approach leads to a more discriminative distribution of token representations. In another study [Sunder et al., 2022], tokenwise contrastive learning is applied to align the representations of a BERT model with a speech encoder augmented with cross-attention mechanisms and non-contextual word embeddings. Once again, the BERT model serves as a teacher to transfer token-level embeddings to speech representations.

5.3 Methodology

The successful completion of this phase of my thesis is attributed to the preceding three phases, namely quality OCR from comic pages (Chapter 2), component detection from comic pages, and associating speech bubbles with characters (Chapter 3), along with generating character embeddings with identity information from faces and bodies of characters (Chapter 4). This integrated effort allowed for the creation of the required dataset, involving the processing of comic book pages sourced from the COMICS dataset [Iyyer et al., 2017].

This section provides an in-depth exploration of the Comicsformer architecture, its design, and its application to the masked comic modeling task. The development of the ComicBERT foundation model is also presented, along with the strategies

employed for pretraining and fine-tuning. Furthermore, comprehensive details regarding the downstream tasks are provided, demonstrating the versatility and effectiveness of the proposed approach.

5.3.1 Data preparation

The overall process is as follows:

- Application of the MTL model to detect speech bubbles above 0.5 score threshold, character faces and bodies, narrative boxes, and panels. Additionally, the MTL model can segment instances of speech bubbles and associate speech bubbles and characters.
- The panels were ordered by z-order.
- The components are associated with panels.
- Speech bubbles within the panels are ordered by the z-order.
- Speech bubble context extraction was done by the e2e OCR model trained with Comics Text+ datasets using speech bubble segmentations.

At the end of the process, given a comic page, I am able to get panels in an ordered manner by their associated components. The speech order and associated speaker are known within a panel. The detected faces and bodies are grouped as characters.

Handling Characters and Associating with Speech Bubbles

First, I performed pairing for bodies and faces. The algorithm used for pairing works as follows: I calculate the intersection rate of faces within bodies. If it is higher than 20%, they are added to the candidate list. From the candidates, I select the one with the highest intersection rate. However, in some cases, there can be multiple faces within a body bounding box. In such situations, the y-coordinate of the faces is used to choose the one higher up in the body bounding box. If there is no intersection,

faces or bodies alone can represent the entire character instance. Some paired faces and bodies can be seen in Figure 5.1

Figure 5.1: Examples of cases with multiple faces in the bounding box of body detection.

After extracting character instances, finding their associations with the existing speech bubbles is necessary. The Multi-Task Learning (MTL) model provides relation scores between all speech bubbles and all faces and bodies.

I filtered out the ones with relation scores below 0.2. Then, I successfully attempted to assign relations to character instances based on these scores.

Since I had already determined which panel each component belonged to, I leveraged this information. Therefore, I filtered the relations for speech bubbles, faces, and bodies based on whether they belong to the same panel. As a result, the possibility of a speech bubble being assigned to a character outside of its panel was eliminated.

Next, I checked the relation scores for the components that make up each character (i.e., face and body). I selected the highest score associated with the character

instance's chosen speech bubble.

While associating each speech bubble with a character, initially, each character should be assigned the highest-scoring speech bubble that corresponds to them. Then, any remaining speech bubbles may belong to the same character as their second or third speech bubble.

At this point, I employed the same algorithm again. However, if a speech bubble has already been assigned to a character, I added a penalty of up to 0.35 in subsequent calls for that character's possible next assignments. The purpose of this penalty is to prevent the model from making incorrect associations, as the likelihood of one of two characters owning both speech bubbles in the same panel is lower than the possibility of them both sharing a single speech bubble (see Figure 5.2).

Z-order Calculation

To simulate the Z-order, I performed the following steps. First, I obtained the union box of the bounding boxes and then partitioned this union box into an n x n grid. The center points of the bounding boxes are then snapped to the grid points. Next, each bounding box is considered as a point with integer values for the x and y coordinates on the grid. Then these points are sorted first based on the column values and then based on the row values [1] to calculate the Z-order. The value of n is 4, which applies to both panels and speech bubbles.

Component Association with Panels

Every detected component is assigned to a panel with the following logic: their intersection rate with each panel is calculated, along with their distance based on the center coordinates. Then, the one with the maximum intersection rate is assigned. In cases where a bounding box has the same intersection with multiple panels, the one with the minimum center distance is selected. If the intersection rate is below 0.2, then that bounding box is left unassigned and is not used for our experiments.

[1] `https://numpy.org/doc/stable/reference/generated/numpy.lexsort.html`

Figure 5.2: Two examples of cases where there are two speech bubbles for two characters. There is a fixed association on the left; before the penalty, both speech bubbles were assigned to one of the characters, thanks to the heuristic for penalizing characters with multiple speech bubbles. However, on the right, there is a not associated speech bubble due to an undetected body, and penalizing helped make the score below the threshold value so that it was not assigned to the only detected body in the scene.

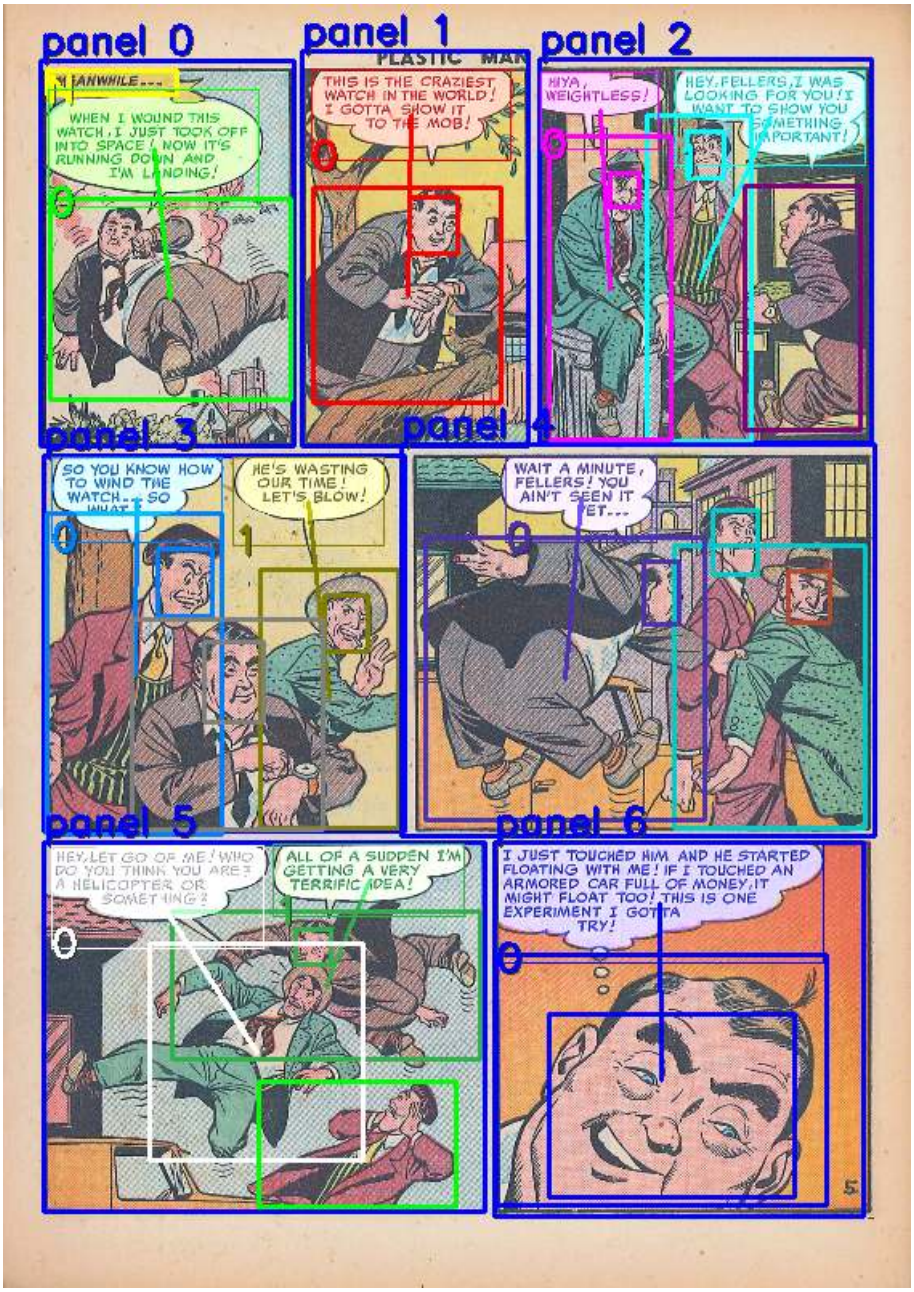


Figure 5.3: A sample comic page is used to demonstrate the finalized inference process of the MTL model for the data preparation step in the Comicsformer model. The panels are z-ordered, and speech bubbles within a panel are displayed with their order indicated at the bottom left. Characters' components are colored with the same color, and speech bubble associations are shown with lines originating from their centers.

OCR and Text Extraction

To extract text from comic books, the speech bubble segmentations from the MTL model are utilized. Instead of directly using the text from *Comics Text+*, which is obtained from speech bubble detections, the decision is made to use segmentations. This choice results in improved text extraction because the image used with OCR models represents the pure content of the speech bubble without any background or overlapping speech bubbles ideally. As a result, the extracted text is of even higher quality compared to *Comics Text+*, which already outperforms the original COMICS dataset's texts.

The segmentation mask is extracted for every speech bubble instance if it surpasses the 0.6 score threshold. Subsequently, morphological transformations are applied to the binary mask using the cv2 library [2] in order to smooth out the edges and fill small gaps. Some examples can be seen in Figure 5.4.

5.3.2 Dataset

As a result of the data preparation method mentioned above, I would like to share the sizes of the resulting dataset's components to provide an understanding:

- Narrative Text: 388,079
- Speech Bubble Text: 1,525,553
- Character Instances: 2,201,437
- Panel: 1,145,654

The above elements are sourced from approximately 4,000 (3959) comic magazines.

Additionally, for a brief textual analysis, I'd like to share the word frequency distributions. The most frequent words in both speech and narrative text can be observed in Figures 5.6 and 5.5.

[2] `https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html`

Figure 5.4: The *Comic Text+ OCR models* use examples from speech bubble segmentations as shown above. Utilizing speech bubble segmentations instead of detections is particularly useful for irregularly shaped speech bubbles.

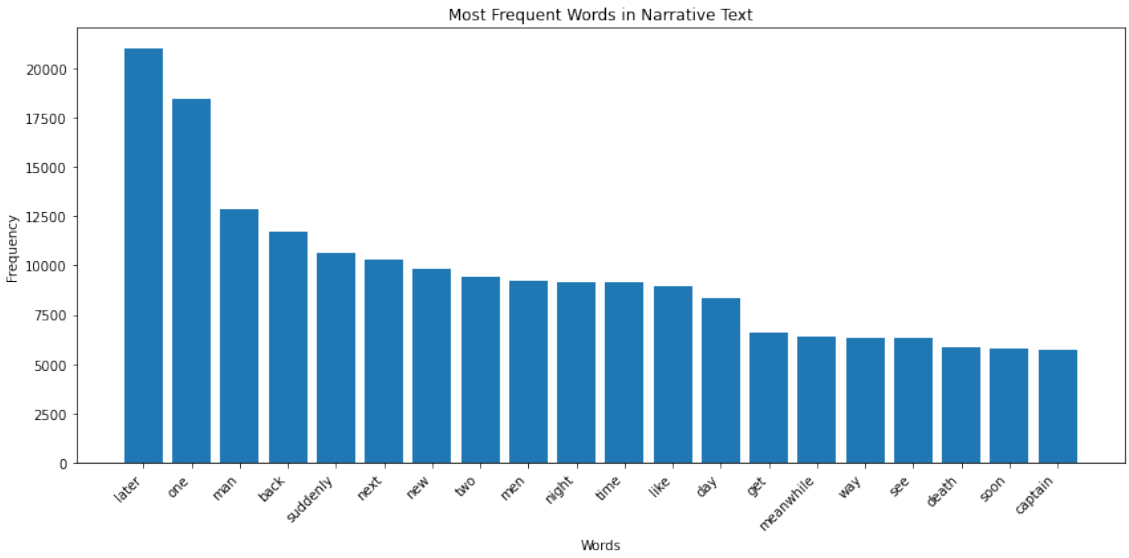


Figure 5.5: Frequency distribution of the top 20 common words, apart from stopwords and punctuation, in narrative boxes.

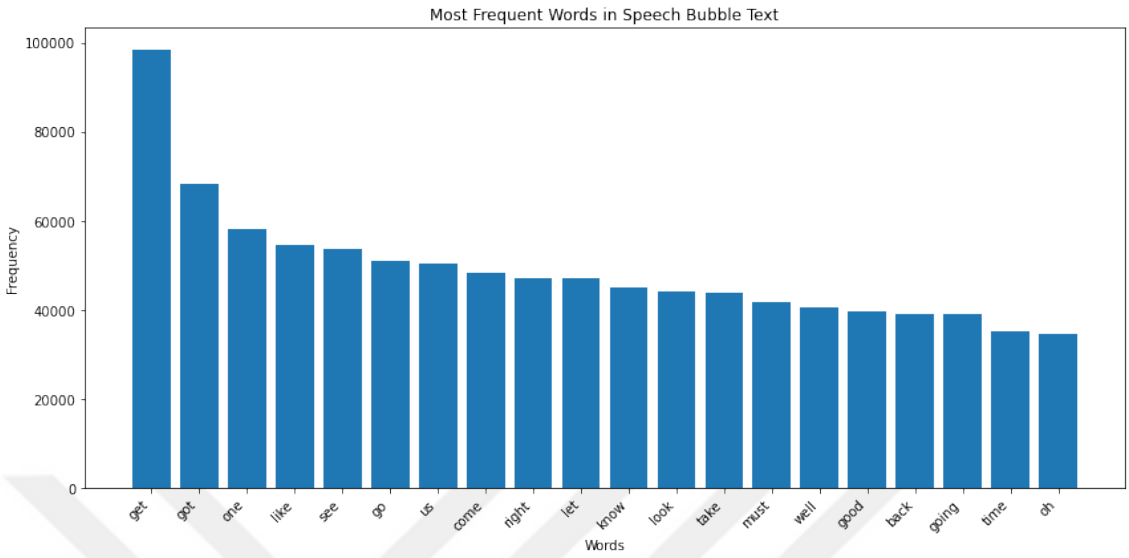


Figure 5.6: Frequency distribution of the top 20 common words, apart from stopwords and punctuation, in speech bubbles.

5.3.3 Comicsformer Architecture

Comicsformer is a multimodal transformer encoder [Vaswani et al., 2017] architecture that takes sequential comic book panels' units as input. The hyperparameters used in the experiments are as follows: input dimension is 384, headcount is 4, hidden layer dimension is 1536, and 6 transformer sub-encoder layers are used, the dropout rate is 0.1, and activation function is ReLU. Additionally, layer normalization is performed with an epsilon value of 1e-5, and it is applied after attention and feedforward operations.

On average, Comicsformer has a relatively small scale compared to other models in the literature. The main factor enabling this is how I construct the input embeddings. In standard NLP architectures, word or subword token embeddings are used. However, the situation is different in comic processing because fixed vocabulary cannot be used. Therefore, our input embeddings come from other pre-trained models. These models are run on comic components, and embeddings are extracted for each "comic unit." The goal of Comicsformer's output is to obtain contextualized comic

unit embeddings. The success or failure of our experiments is directly related to the success of contextualized comic unit embeddings. Both the self-supervised pre-training ComicBERT and the cloze-style tasks require this. In summary, what makes Comicsformer a special type of transformer encoder is the nature of the inputs, the pre-training strategy, and how the outputs are used for downstream tasks.

Encoding Modalities and Inputs of the Comicsformer

The comic units used in Comicsformer are as follows: panels, characters (face and body detection pairs), speech bubble texts, and narrative texts. Since they are different data types, I treat them as separate modalities. The input consists of sequential comic book panels, and the units, as mentioned earlier, are utilized for each panel. The decomposition of a panel is done as follows, and this process can also be visually followed in Figure 5.7. The general processing method begins by selecting a pre-trained model for each modality. During experiments, these selected models are kept frozen. Then, the output of the pre-trained model is projected to the Comicsformer input dimension using a linear layer. The projection layers are trainable.

The pre-trained models used for each modality are as follows:

- **Panels**: EfficientNet [Tan and Le, 2019] is used to process the panel images. Since the baseline model is B0, I decided to use that for having low computational cost. During the transformation phase of the panel images, the following is applied: resizing by longest size to 256, padding to make the image square, then center cropping to have 224x224, normalization.

- **Characters**: The model discussed in the Character Re-identification section (see Chapter 4) is the one that enables me to utilize the *Character modality* in my thesis work. Character instances comprise a character's face and body images, where both may not necessarily appear together (due to occlusion or the character facing away). Thanks to the Character ReID model, such cases can be handled effectively. The Character ReID model consists of an identity-

aware self-supervised ResNet-50 backbone, fine-tuned with an identity network. During character processing, I concatenate the network's final output, a 256-dimensional identity embedding, with the 2048-dimensional backbone embedding before the identity head. Then, I pass this concatenated vector through a projection layer. The reason for doing this is to provide Comicsformer with both the features from the character's self-supervised backbone and the identity information. For body images, the following transformations are applied: resizing by longest size to 128, padding to make the image square, Whereas, for face images, since they were already square images, they are resized to 96. Both images are normalized with 0.5 mean and std values.

- **Texts of Speech Bubbles and Narrative Boxes**: My approach is as follows for language processing. It is not feasible to use separate embeddings for each token in the texts that'd be processed and use them directly with Comicsformer, as this would significantly increase the context window size. Therefore, I opted to use sentence or paragraph embeddings. BERT [Devlin et al., 2019] and its derivatives seemed sensible for this purpose. Additionally, Sentence Transformers [Reimers and Gurevych, 2019], which are trained in a way suitable for MCM objective, were also strong candidates. In the experiments, I tried BERT and two variants of Sentence Transformers, which can be accessed with the following aliases in the Hugging Face's transformers [Wolf et al., 2020] library: 'bert-base-uncased', 'sentence-transformers/all-MiniLM-L6-v2', 'sentence-transformers/all-distilroberta-v1'. Among these options, I conducted my experiments using DistilRoBERTa [Sanh et al., 2019] variant, which showed the best performance and did not cause much computational overhead because of the distillation process. The self-supervised contrastive learning objective used in Sentence Transformers aligns well with the objectives in my thesis, thus making it successful. The text of speech bubbles and narrative boxes were processed using the same pre-trained language model to obtain text embeddings.

Now, I need to discuss the processing steps after obtaining each modality's so-

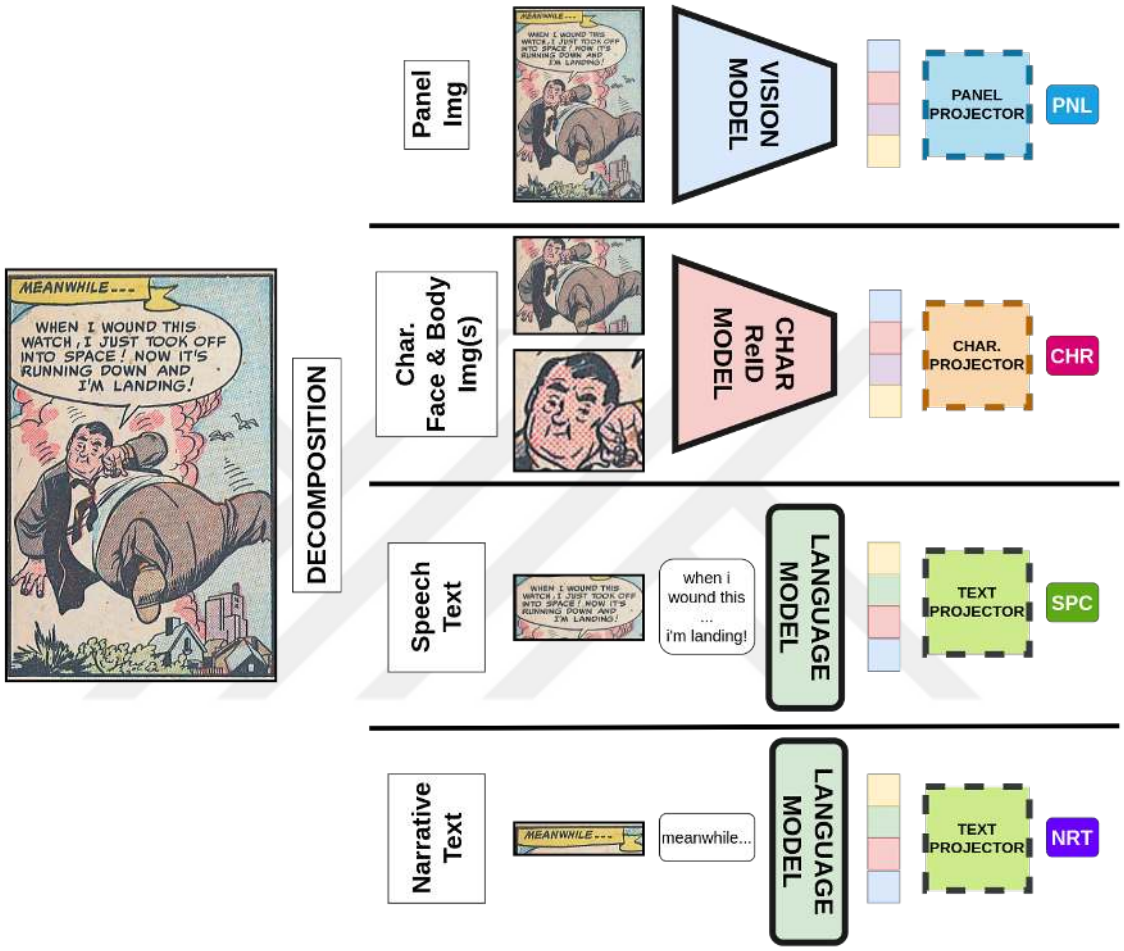


Figure 5.7: Decomposition of a panel as Comicsformer input. The sequential panels are the primary inputs to the Comicsformer. How each panel is used and transformed to be a fundamental unit for Comicsformer is crucial. When converting each panel into the input of Comicsformer, the entire panel image, character instances, speech bubble texts, and narrative texts are utilized. Frozen pre-trained models are employed for each data type or modality, and their outputs are projected to the Comicsformer input dimensions using linear layers.

called *token embeddings* from a panel. The token embeddings for a panel consist of embeddings for the panel itself, three character-speech pairs, and the narrative. These collectively form the *panel unit tokens*. Additionally, it is important to note that in case the MTL model does not establish a character association for a speech bubble, the character token is represented using a padding embedding. Similarly, padding tokens are used in place of speech tokens for characters without speech. Moreover, if a character has multiple speech bubbles and corresponding text, the speech is paired based on speech (order). For instance, the embedding for the character, followed by the embedding for the first speech, and then the embedding for the character again, and the second speech, and so on, meaning character embedding can be reused in case of multiple speeches. As Comicsformer employs a transformer architecture, it can use any number of sequential panels within the context window. However, I adopt the approach of the referenced work [Iyyer et al., 2017], which uses three context panels.

To separate each panels' unit tokens, [SEP] (Separator) token embeddings are inserted. This is in line with how BERT [Devlin et al., 2019] handles the "Next Sentence Prediction" task. In this scenario, a total of 26 token embeddings are used for each sequence. Before feeding the token embeddings to Comicsformer as input, I sum them with positional embeddings [Vaswani et al., 2017] and modality embeddings. While positional embeddings are constant values and not trained during training, modality embeddings are specific to each modality (panel, character, speech, narrative) and are trainable embeddings initialized with random values sampled from a uniform distribution within the range of -0.1 and 0.1. After these steps, I apply layer normalization to complete the input formation. The whole process can be seen in Figure 5.8.

5.8

5.3.4 Masked Comic Modeling and ComicBERT

Masked Comic Modeling (MCM) can be explained in the following way. Let S be the sequence of units in a comic page comprising panels, characters, speech, and

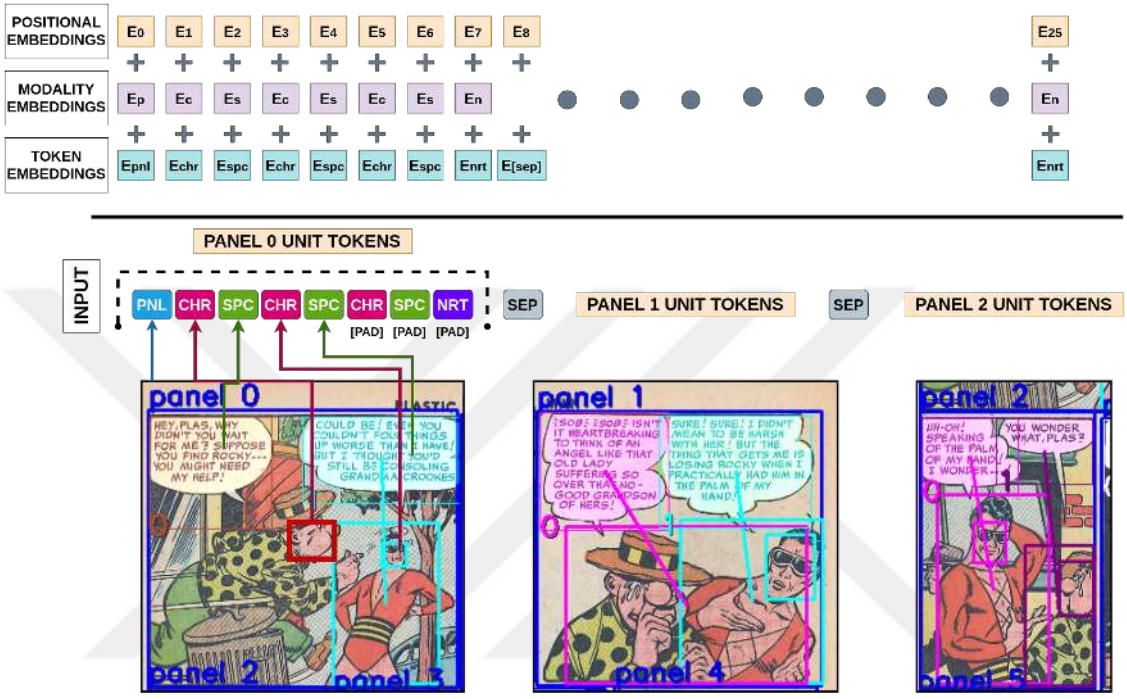


Figure 5.8: Formation of the Comicsformer input. For each panel, modality projections are used as token embeddings. In the experiments, three sequential panels were used, and special separator ([SEP]) tokens were employed to separate the panels. The sequential structure of the comics was reflected in the model using positional embeddings. Trainable special modality embeddings were used on tokens coming from modalities to facilitate the discrimination abilities of the model. The token order for each panel is as follows: panel, three character and speech token pairs, and the final narrative token. Character and speech token pairs were provided in the order of speech bubbles. In case of missing data, pad tokens were used, and padding masking prevented them from affecting the computations

narrative texts. Each unit u within the sequence S is represented by a vector $\mathbf{u}$ in an input embedding space. The MCM task aims to contextualize these embeddings.

Let T be a set of indices representing tokens within S. For each token index t in T, I define a masking operation M_t, which masks the corresponding token's embedding in $\mathbf{u}_t$. The objective can then be considered to recover or retrieve the masked embeddings using contextual information. The masking rate in my experiments is 0.2 of the sequence S.

Given the Comicsformer model, the correspondent output token embedding, $\mathbf{o}_t$, of a masked token $\mathbf{u}_t$ is passed through a feed-forward network to obtain a transformed embedding $\mathbf{h}_t$. This transformation can be expressed as:

$$\mathbf{o}_{1:n} = \text{Comicsformer}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$$

$$\mathbf{h}_t = \text{FFN}(\mathbf{o}_t)$$

The feed-forward network (FFN) utilized in the model comprises two linear layers with identical input and output dimensions, aligned with Comicsformer's output dimension. The FFN incorporates a GELU [Hendrycks and Gimpel, 2016] activation function and layer normalization in between linear layers, contributing to the augmentation of the model's internal representation transformation capabilities and acting as a projection head.

MCM aims to maximize the similarity between $\mathbf{h}_t$ and the corresponding unit's encoding while minimizing the similarity with other masked unit encodings within the sequence and batch. Let M be all masked input embeddings. Then, this can be formulated as an optimization problem for a single masked embedding:

$$\text{maximize} \sum_{t \in M} \left(\text{similarity}(\mathbf{h}_t, \mathbf{u}_t) - \lambda \sum_{u' \in M \setminus \{\mathbf{u}_t\}} \text{similarity}(\mathbf{h}_t, \mathbf{u}') \right)$$

where similarity$(\cdot)$ measures the similarity between two embeddings and λ is a balancing parameter.

Because of how the problem is formulated, it has transformed into a contrastive learning problem. Therefore, contrastive learning can be applied to train Comicsformer using NT-Xent [Chen et al., 2020a] loss or similar contrastive losses. In my

experiments, I used the NT-Xent loss. Thus, the NT-Xent loss for the MCM problem can be conceptualized as follows for ith masked unit in the batch within in total N masked units:

$$\mathcal{L}_{\text{MCM},i} = -\log\left(\frac{\exp(\text{sim}(h_i, u_i)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i]\exp(\text{sim}(h_i, u_k)/\tau)}\right) \tag{5.1}$$

Cosine similarity is used as the similarity function during my experiments.

When MCM is used with the Comicsformer architecture, the whole framework is called ComicBERT. The framework can be visualized as in Figure 5.9, and the main learning algorithm can be followed in Algorithm 5.

Algorithm 5 Main learning algorithm of *Masked Comic Modeling* (MCM)

Require: Batch size B, constant τ, modality encoder e, masking function m, Comicsformer model c, feed-forward network f, NT-Xent loss function nt.

for sampled minibatch $\{S_k\}_{k=1}^{B}$ **do**
  $\mathrm{u} \leftarrow \mathrm{e(S)}$ ▷ Encode input sequences
  $\mathrm{u}_m, \mathrm{i} \leftarrow \mathrm{m(u)}$ ▷ Apply masking to encoded units and get masked indices
  $\mathrm{o} \leftarrow \mathrm{c}(\mathrm{u}_m)$ ▷ Generate Comicsformer representations
  $\mathrm{h} \leftarrow \mathrm{f(o)}$ ▷ Obtain FFN features
  $\mathrm{u}_{um} \leftarrow \mathrm{u}[i]$ ▷ unmasked panel unit encodings
  $\mathrm{h}_m \leftarrow \mathrm{h}[i]$ ▷ masked representations
  $\mathrm{u}_{um} \leftarrow \textsc{Flatten}(\mathrm{u}_{um})$
  $\mathrm{h}_m \leftarrow \textsc{Flatten}(\mathrm{h}_m)$
  $\mathcal{L} \leftarrow \mathrm{nt}(\mathrm{u}_{um}, \mathrm{h}_m)$ ▷ Compute NT-Xent loss
  Update networks e, c, f to minimize $\mathcal{L}$ ▷ Backpropagate and update
end for
return Comicsformer $c(\cdot)$, Modality encoder $e(\cdot)$, and discard feed-forward network $f(\cdot)$

The contextualized embeddings $\mathbf{o}_t$ obtained through MCM enhance the understanding of the relationships between different units on the comic page. The resulting

pre-trained model, ComicBERT, can be employed for various downstream tasks in comics understanding and processing.

5.3.5 Downstream Tasks Overview

In this thesis, there are two more sections where I got results for cloze-style tasks (Sections 6, 2.4.2). In those sections, I explained the cloze-style task concepts. They need to be summarized again. Cloze-style tasks are first proposed in [Iyyer et al., 2017] to measure deep learning models' ability to capture context and meaning from comics. The tasks follow a structure where features of context panel $p_{i-3}, p_{i-2}, p_{i-1}$ are used to predict the features of panel p_i from a single modality. The setup can be extended to incorporate an arbitrary number of context panels, a potential area for future exploration.

The final panel's, p_i, features are predicted using a cloze-style framework. In all experiments, three candidates are presented as options, and models leverage context information to assign a higher score to the correct candidate. The tasks are described as follows:

- **Text Cloze:** Predicting the single speech bubble's text of the final panel p_i. This task also uses the final panel's visual information with models, including panel modality. However, they are used with masked textbox areas to focus on artwork features while considering the context for scoring.

- **Visual Cloze:** Predicting artwork of the final panel p_i. Unlike *text cloze*, text features of the final panel are not provided.

- **Character Coherence:** Reordering two dialogue text boxes for the final panel.

Task Variations: Each task has two difficulty levels, apart from character coherence. In the *easy* case, incorrect candidates are chosen from different comic books, while the *hard* case uses candidates from the same comic book.

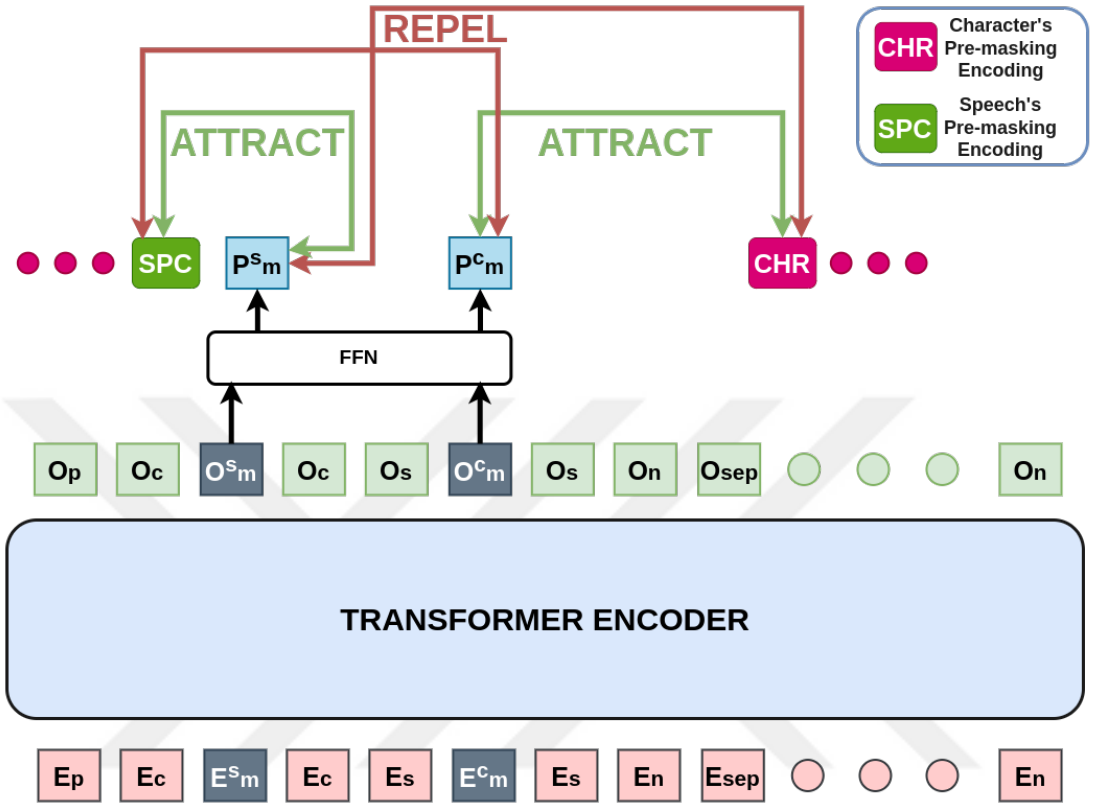


Figure 5.9: Overview of *Masked Comic Modeling (MCM)* self-supervision task for comic representation learning. Similar to BERT [Devlin et al., 2019], the goal is to obtain contextualized *unit* embeddings. These units can be words or subwords (tokens) in language models. However, in the context of Comicsformer, these units correspond to panels, characters, speech, and narrative texts. Unlike language models, it is not possible to determine which unit a masked token belongs to directly from a vocabulary. Therefore, contrastive learning methods are utilized. After passing a masked token's Comicsformer output through a feed-forward network, the resulting embedding maximizes the similarity with the unit's modality encoding while minimizing the similarity with other modality encodings (both within the same sequence and across the batch). The pre-trained framework achieved through Comicsformer with MCM is referred to as ComicBERT.

Additional Tasks

In addition to the abovementioned tasks, I propose two new tasks in this thesis: *Scene Cloze* and *Contextual Character to Speech Attribution*.

- **Scene-Cloze:** Unlike other cloze-style tasks, this task doesn't predict a single feature of the final panel, p_i, such as image features or speech bubble text. Instead, it utilizes all the information present in the final panel. The context of p_i is extracted and compared with the context of previous panels to measure their similarity. The task involves predicting the next scene based on the context from the previous n panels.

- **Contextual Character to Speech Attribution:** In this task, by fusing the features of any character from the final panel with context embeddings, it aims to measure the similarity between context-enhanced character embedding with any given text embedding, ideally belonging to a speech bubble. It attempts to maximize the similarity of the speech bubble content associated with the character. It is different from the Character Coherence task since this does not do character speech reordering in a panel, but rather, it attempts to score the likelihood of a character saying any speech bubble content given the context from the previous panels.

Context Extraction

Paragraph 5.3.3 describes how the Comicsformer input was constructed. When this input is provided to the Transformer encoder, we obtain output features for each position. Then, by performing mean pooling, a single context precursor vector is obtained. Subsequently, the context precursor is passed through the context projector layer to obtain the final context features. The context projector consists of a linear layer and an L2 normalization operation. It should be mentioned that the features used in mean pooling are the features of non-padded positions. Furthermore, Figure 5.10 illustrates the process of context extraction from Comicsformer.

Scoring and Option Embeddings

It is necessary to explain the process of obtaining option embeddings separately for each task, which is discussed after the scoring explanation. Considering how context embedding is extracted, the scoring process for an option is as follows:

$$\text{option\_logit} = \mathbf{O} \cdot \mathbf{c}$$

Since we perform L2 normalization on both the option and the context vectors, the option logit equals the cosine between those two vectors; in fact,

$$\text{option\_logit} = \cos\theta$$

As the cross-entropy loss is used as the loss function and there are three options, which is the case for all cloze-style tasks, the final scores are as follows:

$$\text{scores} = \text{softmax}([\text{option\_logit}_0, \text{option\_logit}_1, \text{option\_logit}_2])$$

The extraction of options' embedding process for each task is as follows:

- **Text-Cloze:** The Final panel is encoded as Comicsformer input and fed to the Comicsformer. Then the output feature at the position of the speech bubble, which is the third position, since the final panels should only have a single speech bubble text for the text cloze, is obtained and then processed with the output projector, which has the same architecture as context projector, meaning a linear layer followed by L2 normalization. Depending on the model type, encoded panel features are masked; for instance, if it is the panel-only (image-only) model, then character input features are masked in the input.

- **Visual-Cloze:** Almost the same as Text-Cloze, but with the difference that textual (speech bubble, narrative) input features for the final panel are masked. Since the task requires not using text information from the final panel apart

from the visual features when character modality is involved, it is not masked in the input. However, only the output feature of the panel token is used from the outputs.

- **Scene-Cloze:** Since there are no restrictions for the scene cloze, all modalities of the model type are used. Similar to context extraction, the mean pooling operation is applied to get the final output features before the output projector.

- **Character Coherence:** This is similar to scene-cloze in the sense that all the available modality information is used in the input of the Comicsformer. However, only the character and speech output features are used during the mean pooling operation. The options construction differs from the other tasks since they use information from other panels. The panel information is used as is for the correct options for character coherence. However, for the wrong option, the speech input token feature positions are exchanged between the first and second.

- **Contextual Character to Speech Attribution:** This task is different from the others because it does not use the Comicsformer architecture for options. Context features are fused with the character projections coming from pre-trained models. Then, their similarity with speech projection is measured. Both projections are further processed with linear layer and L2 normalization before similarity measurement.

Proper masking is applied for each model type, panel, character, and text(speech, narrative) can be thought of as different modalities. Based on that, during the context and option feature extraction, masking was applied based on the available modality. Overall, the model types are as follows:

- **Text-only**: Uses only text modality, speech, and narrative.

- **Char-only**: Uses only character modality.

- **Image-only (Panel-only)**: In the original task definition, it was called Image-only when the model uses only panel visual features. However, I used character information based on the characters' images. Hence, the modal is changed to Panel-only.

- **Image-Text (Panel-Text)**: Uses panel visual information and text modality.

- **Panel-Char**: Uses panel visual information and character visual features.

- **Char-Text**: Uses character visual features with text features.

- **Panel-Char-Text**: Uses all available information from the three modalities.

5.3.6 Pre-training and Fine-tuning Details

Self-Supervised ComicBERT Pre-Training

The training process for the ComicBERT framework involves two settings. In the fast setting, the model is trained for 50 epochs on the first 20,000 comic pages, while in the slow setting, training extends to 500 epochs on the first 100,000 comic pages until the 3000th series of Comics [Iyyer et al., 2017], with early stopping triggered at 100 epochs. Gradient clipping is applied with a value of 0.5 to stabilize training. The model is configured to use panel, character, speech bubble, and narrative information, and a masking rate of 0.2 is used along with the NT-Xent loss function. None of the token types are exempt from the masking strategy. A learning rate 0.0001 is employed with a weight decay of 0.01 to control overfitting. The AdamW [Loshchilov and Hutter, 2017] optimizer is used with a cosine schedule with a warmup (0.1 of max. epochs was warmup). The Comicsformer properties include an input dimension of 384, 4 heads in multi-head attention, a feedforward dimension of 1536, 6 encoder layers in the encoder, and a dropout of 0.1. The ComicBERT framework processes context from three panels, and testing is performed on a separate set of 10,000 pages; these pages come from series above 3500 from the Comics dataset,

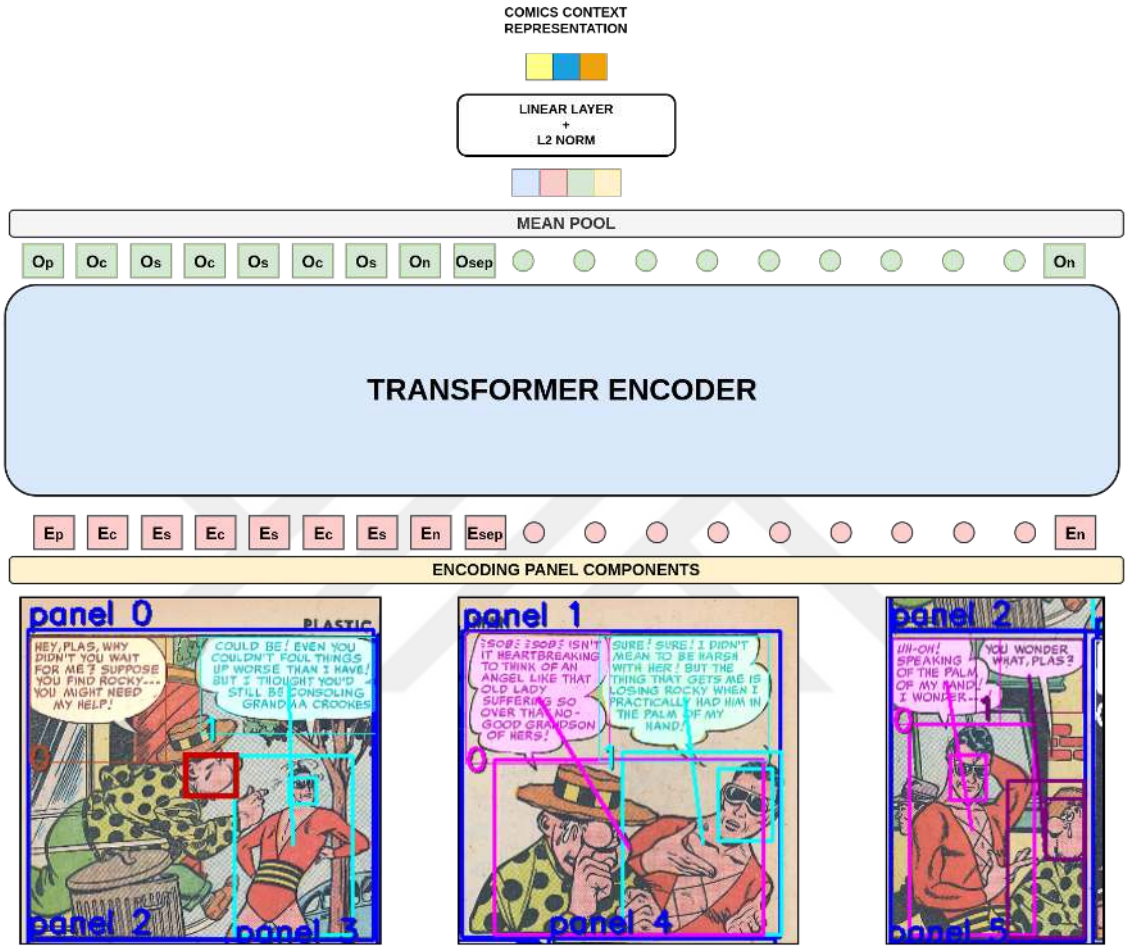


Figure 5.10: Obtaining the context embedding with Comicsformer from comic sequences. The transformer encoder outputs at the core of Comicsformer are fused using mean pooling. During mean pooling, the outputs corresponding to padding tokens are not considered. The resulting mean-pooled output is passed through a linear layer (in the experiments, the input-output dimensions are 384x256). Finally, L2 normalization is applied. L2 normalization lets us directly obtain the cosine similarity since downstream tasks use dot product.

which has close to 4000 series. A batch size of 128 is used, and early stopping is implemented with a patience of 20 epochs, monitoring minimum validation loss.

As mentioned earlier, the generated sequences from the pages undergo filtration before training. This filtration process removes sequences with fewer than two speech bubbles and two characters. This is a modality regularization technique. Additionally, it eliminates irrelevant sequences originating from advertisement pages, as the Comics dataset might include such pages. The first 3000 series are allocated for training, series 3000-3500 for validation, and from series 3500 to the end for testing. In the fast setting, the number of sequences used in training is as follows: train dataset size: 57,250, val dataset size: 5,818, test dataset size: 30,525. On the other hand, for the slow setting, the train dataset size is 292,288, the validation dataset size is 15,594, and the test dataset size is 30,525.

Task Specific Fine-Tuning of Comicsformer

The Comicsformer properties are the same as those used in the pre-training of ComicBERT. Most of the hyperparameters are the same. Those that differ are as follows: The maximum number of epochs is 20. The projection layer output dimension is 256 if ComicBERT weights were used in training and they were not frozen. The Masked Comic Modeling objective was active if transfer learning from ComicBERT was applied. The early stopping was active with a patience of 5 epochs based on the maximum validation accuracy value during training.

For every task-specific training, page-wise training, validation, and test split counts were 20,000, 2,000, and 10,000. Sequence filtering strategies were employed. For every task, like ComicBERT, sequences with fewer than two speech bubbles and two characters in their context panels are removed. Additionally, every task has an additional filter for the final panel based on the number of characters associated with a speech bubble. Text-cloze requires 1; character coherence requires 2; scene-cloze and character-to-speech attribution requires at least 1. whereas, for visual-cloze, no such filter exists. Furthermore, for text-cloze and character coherence tasks, speech bubble areas for the final panel are masked.

No-Context (NC) Models

The No-Context (NC) model is a model type proposed by Iyyer et al. [Iyyer et al., 2017] to measure the impact of context. The NC model has only been employed in the Panel-Text (Image-Text) model. It can be regarded as an ablation study specific to context. Instead of context, the visual features of the final panel p_i are used in place of context for this model.

5.4 Results & Discussion

5.4.1 Comparison of ComicBERT Results and Previous Results

Table 5.1: Comparison of the results with ComicBERT pre-trained Comicsformer architecture and Iyyer et al.'s [Iyyer et al., 2017] results based on accuracy on cloze-style tasks proposed in [Iyyer et al., 2017]. Human baseline is also shared for only hard tasks since the easy task is trivial for humans. *NC-Image-text* models use only the final panel's image without any information from preceding context panels. No context model results strike the usefulness of the context for the cloze style tasks and narrative understanding.

MODEL	**Text Cloze**				**Visual Cloze**				**Char. Coherence**	
Task Difficulty	EASY		HARD		EASY		HARD			
	Original	Ours	Original	Ours	Original	Ours	Original	Ours	Original	Ours
Text-only	63.4	**66.4**	52.9	**57.8**	55.9	55.9	**48.4**	47.0	**68.2**	65.1
Image-only	51.7	**80.9**	49.4	**56.6**	85.7	**88.7**	63.2	**64.8**	**70.9**	64.7
Image-text	68.6	**84.0**	61.0	**62.7**	81.3	**89.1**	59.1	**65.8**	**69.3**	66.1
NC-Image-text	**63.1**	52.3	**59.6**	48.5	-		-		**65.2**	65.0
Human	-		84		-		88		87	

Table 5.1 presents the results of this comparison. The model, referred to as "Ours," is the ComicBERT model pretrained with the fast-setting, then fine-tuned with transfer learning for task-specific adaptation. The results can be interpreted from three perspectives: NC (No-Context) results, results excluding NC for text-cloze and visual-cloze, and character coherence results.

- **NC Results:** These can be observed specifically for the text-cloze task. Their results are better than ComicBERT's in the easy and hard settings. From this, it can be inferred that my model generally depends more on context. Learning does not occur thoroughly without context information. This could be attributed to parameter size, as their model is LSTM-based and thus has significantly fewer parameters.

- **Text-Cloze and Visual-Cloze:** In this case, ComicBERT performs notably better, especially with panel information. However, the improvement in the text-only model is minimal. In the hard setting, while the improvement is not as significant as in the easy setting, it is still noticeable. Nevertheless, the results remain distant from the human baseline. One of the reasons I wanted to incorporate character information is due to this upper bound. It can be said that approaching the human baseline using only panel and text information is challenging. Nonetheless, ComicBERT is still able to surpass previous results.

- **Character Coherence:** This task can be arguably considered controversial. When the task is proposed, it's framed around characters, yet Iyyer et al. [Iyyer et al., 2017] do not use any character information. They attempt to perform text ordering only, which is dependent on the z-order. Thus, this task may merely reflect z-order accuracy. While my results are close to theirs, they are lower, and regardless, the results do not improve much further with character information or longer pre-training.

5.4.2 Multi-Modal Results of ComicBERT and Comicsformer

In this section, I present the results for the text-cloze, visual-cloze, scene-cloze, and character coherence tasks, as shown in Tables 5.2, 5.3, 5.4, and 5.5. The results encompass the outcomes of two approaches: Comicsformer, which I trained directly with task-specific training without pre-training, and ComicBERT, which, as the name suggests, is fine-tuned from a model pre-trained with Masked Comic Modeling. I will not share results for the Contextual Character and Speech Attribution task as

Table 5.2: Performance Comparison of Text-Cloze Task between Comicsformer and ComicBERT. ComicBERT is pre-trained using the Masked Comic Modeling task before and during its training for the text-cloze task.

Model	**Text Cloze**			
	easy		*hard*	
	Comicsformer	ComicBERT	Comicsformer	ComicBERT
Text-only	34.1	**66.4**	33.7	**57.8**
Char-only	55.6	**88.3**	33.6	**60.7**
Panel-only	63.0	**80.9**	33.5	**56.6**
Panel-Text	59.2	**83.9**	32.9	**62.7**
Panel-Char	68.1	**91.1**	33.6	**65.0**
Char-Text	55.2	**89.0**	33.4	**66.6**
Panel-Char-Text	69.8	**92.3**	33.2	**68.2**

Table 5.3: Performance Comparison of Visual-Cloze Task between Comicsformer and ComicBERT. ComicBERT is pre-trained using the Masked Comic Modeling task before and during its training for the visual-cloze task.

Model	**Visual Cloze**			
	easy		*hard*	
	Comicsformer	ComicBERT	Comicsformer	ComicBERT
Text-only	52.9	**55.9**	33.5	**46.9**
Char-only	84.8	**91.9**	54.9	**71.6**
Panel-only	83.5	**88.7**	33.7	**64.8**
Panel-Text	82.6	**89.1**	33.4	**65.8**
Panel-Char	87.0	**93.7**	56.2	**75.9**
Char-Text	83.9	**91.9**	58.1	**74.3**
Panel-Char-Text	85.6	**93.9**	51.2	**74.9**

Table 5.4: Performance Comparison of Scene-Cloze Task between Comicsformer and ComicBERT. ComicBERT is pre-trained using the Masked Comic Modeling task before and during its training for the scene-cloze task.

Model	**Scene Cloze**			
	easy		*hard*	
	Comicsformer	ComicBERT	Comicsformer	ComicBERT
Text-only	66.7	**72.4**	33.0	**62.4**
Char-only	85.1	**93.3**	51.8	**73.9**
Panel-only	82.4	**88.9**	33.3	**64.8**
Panel-Text	83.3	**91.1**	33.0	**70.1**
Panel-Char	85.2	**94.9**	51.7	**76.2**
Char-Text	84.6	**94.2**	52.0	**77.6**
Panel-Char-Text	86.9	**95.6**	51.3	**79.3**

Table 5.5: Performance Comparison of Character Coherence Task between Comicsformer and ComicBERT. ComicBERT is pre-trained using the Masked Comic Modeling task before and during its training for the character coherence task.

Model	**Char. Coheren.**	
	Comicsformer	ComicBERT
Text-only	65.1	**66.7**
Char-only	64.4	**66.4**
Panel-only	64.8	**66.7**
Panel-Text	66.1	**67.1**
Panel-Char	**66.0**	65.9
Char-Text	64.9	**66.1**
Panel-Char-Text	66.1	**66.4**

I did not obtain significant results, leaving it as a potential avenue for future work.

During the evaluation, apart from character coherence, the focus should be on the cloze-style tasks. Regarding character coherence, the situation I mentioned in the previous section still holds, even to the extent that it can be considered as a validation of its improper setup. Despite pre-training, the results show only a very slight improvement.

Conversely, looking at the cloze-style tasks yields promising outcomes for the proposed approach, using character information, Comicsformer architecture as a base model, and Masked Comic Modeling pre-training. In the previous section, it was noted that the ComicBERT approach exhibited improvement compared to previous results. With the addition of character information, it can be confidently stated that the results in the *Panel-Char-Text* model have significantly improved over the *Panel-Text* model. Across all cloze-style tasks, the *Panel-Char-Text* model surpasses 90% accuracy, showcasing state-of-the-art performance. However, it should be noted that the most crucial improvement is observed in the hard setting, which requires the most contextual information due to the hard setting setup. Remarkable improvement is observed with the ComicBERT framework. The model achieves an accuracy of 68.2 for text-cloze and 74.9 for visual-cloze. Examining these findings, it can be said that my results are getting closer to the human baseline. Moreover, it demonstrates the importance of character information in comprehending comics. As the scene-cloze task is novel, the results here will serve as a baseline for future work. In the hard setting, I achieved nearly 80% accuracy in this task. This implies that when the model is provided with context, it can successfully identify the possible content of the 4th panel within the same comic book with an accuracy of nearly 80%. This displays the potential for using this framework in creating comics and highlights how deep learning models can successfully process complex media like comics.

Considering the contributions of different modalities, when examined individually for the three cloze-style tasks, text consistently yields the lowest accuracy, while character consistently exhibits the highest performance. Additionally, when comparing pairs of modalities, it is evident that *Panel-Text* consistently achieves

lower results than the other combinations. This reaffirms the critical importance of integrating character information for effective comic understanding.

5.4.3 ComicBERT Pre-Training Results and The Best Results

Table 5.6: Pretraining Results of the Masked Comic Modeling (MCM) Task for ComicBERT. Two distinct settings were employed during the experiments. The upper setting involved fewer sequences and shorter training duration, optimizing for time efficiency (fast setting). The fast setting was utilized for most experiments, except for the results in Table 5.7. For each setting, the batch size was 128.

Seq. Count	**# Epoch**	**NT-Xent Loss**	**Top-1 (%)**	**Top-5 (%)**	**Mean Pos.**
57,250	50	3.3	26.3	48.6	23.7
292,288	100	2.8	37.1	58.4	18.9

During the ComicBERT pre-training phase, I employed two distinct settings, namely slow and fast, based on the duration of training. The fast setting was trained for 50 epochs and utilized a smaller dataset. This training was completed within approximately 24 hours on an NVIDIA V100 GPU. In contrast, the slow setting took nearly a week on the same GPU for training. A substantial difference becomes evident when observing the pre-training outcomes for both settings (see Table 5.6). However, this contrast is not equally reflected in the results of the cloze-style tasks. The results are presented in Table 5.7, which were achieved using ComicBERT trained with the slow setting. Nevertheless, its contribution over the fast setting remains marginal. Sharing the results of these two experimental settings could guide future research involving similar experiments, saving time and serving as a reference. Additionally, my best results represent the new SOTA, except for the character coherence task.

Table 5.7: The top results of ComicBERT on cloze-style and Character Coherence tasks. The ComicBERT model utilizes information from panels, characters, and text, pre-trained for 100 epochs on nearly 300,000 sequences. These results are also the current SOTA results.

Task	**Task Difficulty**	
	easy	*hard*
Text-Cloze	93.7	69.5
Visual-Cloze	95.7	77.1
Scene-Cloze	96.7	80.6
Char. Coheren.	67.1	

5.5 Conclusion & Future Work

The evaluation of the proposed framework, ComicBERT, revealed its promising potential in extracting contextual information from comic pages, specifically from the sequences of panels. I achieved SOTA results in various cloze-style tasks by incorporating the Comicsformer architecture and pre-training it using Masked Comic Modeling. The integration of character information proved crucial, consistently improving the model's performance and pushing it closer to human-level understanding.

Despite these achievements, certain aspects require further attention and exploration. For instance, the contextual character-to-speech attribution task presented challenges and yielded limited results. It remains a potential avenue for future investigation, leading to improved character-based content generation.

In terms of future work, the following directions could be pursued:

- **Enhanced Contextual Understanding:** Further refinement of the contextual character-to-speech attribution task may lead to better character-centered content generation. Exploring more sophisticated fusion techniques and architectures could contribute to this aspect.

- **Multimodal Interaction:** Investigating the interplay between modalities within comics, such as image, text, and character features, could show deeper insights into the underlying narrative structure.

- **Interactive Comics Generation:** The insights gained from my framework could pave the way for interactive comics generation tools. Such tools would allow creators to receive real-time suggestions and visualizations as they design their comics.

- **Generalization to Other Media:** The techniques and models developed in this thesis are not limited to comics alone. They could potentially be extended to other visual media forms, like storyboards, graphic novels, and even video content.

- **Cultural and Style Adaptation:** Exploring the adaptation of my framework to different comic styles, genres, and cultural contexts could offer a broader perspective on its capabilities and limitations.

In conclusion, this part of my thesis has laid the foundation for advancing the understanding of comics through the application of deep learning models. The achievements in cloze-style tasks, the development of the ComicBERT framework, and the insights gained from multimodal analysis open new avenues for the intersection of technology and art regarding comics.

Chapter 6

SOLVING TEXT-CLOZE TASK WITH PROMPTING LANGUAGE MODELS

Iyyer et al. [Iyyer et al., 2017] propose the text-cloze task, visual-cloze, and character coherence tasks to measure closure understanding of a model provided context information extracted from the previous three panels. Therefore, the text-cloze task involves using a cloze-style framework to predict the text in the final panel (p_i) by using context panels (p_{i-3}, p_{i-2}, p_{i-1}) to score three candidate texts.

I reformulated the problem for language models with two approaches. The first approach is to design and use a question-answer framework in a natural language prompt and get answer predictions from a language model. However, smaller-sized language models require some form of fine-tuning for such frameworks. Thus, I came up with the loss-based approach.

6.1 Methodology

6.1.1 Answer Scoring Strategies

Loss-based Answer Scoring

Loss-based answer scoring depends on the relation between cross-entropy loss and perplexity. Let $\mathcal{D}$ be the test set of size N and θ be the parameters of the language model. Then, the cross-entropy loss for the language model on $\mathcal{D}$ is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log p(w_i|\boldsymbol{w}_{<i};\theta) \tag{6.1}$$

where w_i is the i-th word in the test set, $\boldsymbol{w}< i$ represents the words that come before w_i in the test set, and $p(w_i|\boldsymbol{w}< i;\theta)$ is the predicted probability of w_i given the previous words and the parameters θ.

The perplexity of the language model on the test set $\mathcal{D}$ is defined as:

$$\text{Perplexity}(\mathcal{D}) = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log\frac{1}{p(w_i|\boldsymbol{w}_{<i};\theta)}\right) \tag{6.2}$$

The relationship between perplexity and cross-entropy loss is given by:

$$\text{Perplexity}(\mathcal{D}) = \exp\left(\mathcal{L}(\theta)\right) \tag{6.3}$$

This implies that minimizing the cross-entropy loss on the training set will result in a lower perplexity on the test set, which is the ultimate objective of language modeling. In other words, language models are more likely to produce texts with lower loss values. Based on this information, the loss-based answer scoring method is as follows:

1. Generate a base prompt based on the information and texts of previous panels.
2. Concatenate each answer with the base prompt separately, resulting in three base-prompt + answer pairs.
3. Calculate the loss value for each pair.
4. Order the losses from low to high.
5. Select the answer with the lowest loss value as the final output.

This approach leverages the relationship between loss and language probability to select the most probable answer based on the given information and previous texts.

Multiple Choice Question-Answering Framework

This approach involves engineering prompts in a way that presents all three potential answers in a multiple-choice format, prompting the model to predict the correct answer option. The answer format for this approach is a single token, and the answer search space consists of discrete values of *A, B, C*. I drew inspiration from

various tasks such as question-answering, conversation, and text classification to create prompts for this approach. The format of the prompt can significantly impact the prediction results, and while there are various prompt templates for different tasks, prompts for the text-cloze task in comics have not been explored. Given the nature of comics, I was influenced by conversational prompts, which allowed for the introduction of character identities into the prompts. Additionally, since the text-cloze problem involves three potential answers, the prompt was structured in a question-answering format. Finally, since the answer options are predetermined, text classification methods were used with prompts to select the correct answer. Figure 6.2 shows examples of two approaches.

6.1.2 Data Source

As the primary data source, I am using COMICS dataset [Iyyer et al., 2017] with an update for the texts of it. Text data of COMICS is augmented by *COMICS Text+ Dataset* (Chapter 2). To provide character information to the models, I also developed a web-based annotation tool to get associations of character identities with speech bubbles. The annotation tool is called Comics Sequence Identity Annotator.

Comics Sequence Identity Annotator

The Character Identity Annotation Tool for Comics Sequences is a helpful tool template for researchers, creators, and enthusiasts in comic studies. This tool is necessary because it helps establish a consistent and accurate character identity throughout a comic sequence.

This tool ensures that all characters in a comic sequence are correctly identified, particularly in cases where there may be multiple characters with similar appearances. By providing an interface that allows users to group characters' bodies, faces, and speech bubbles, the tool helps minimize confusion and ambiguity that can arise when characters are not properly identified.

Additionally, this tool can be used in various research applications, such as analyzing character interactions, tracking character development throughout a series,

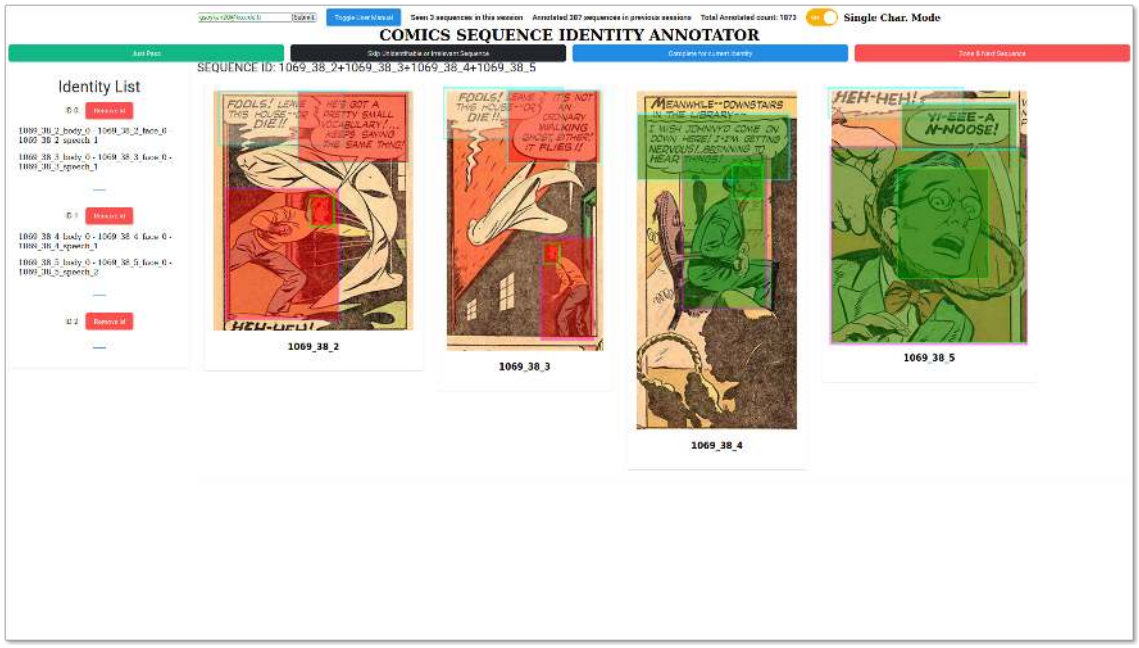


Figure 6.1: **Character Identity Annotation Tool for Comics Sequences.** The tool provides an interface for associating bodies, faces, and speech bubbles of characters with the same identity groups.

or identifying recurring themes and motifs. Creators can also use it to ensure continuity in their storytelling and maintain consistency in their characters' appearances and identities. I also used this tool and annotations collected by it for the comic character re-identification task within this thesis (Chapter 4)

The Character Identity Annotation Tool for Comics Sequences offers users two distinct modes for annotation: Single Character Mode and Multiple Character Mode.

- **In Single Character Mode**, users can focus on annotating the identity of a single character in a single panel. This mode is useful for annotator speed and for capturing characters with the same identity. It can be estimated that almost one-third of the sequences that have a single body or face in a single include just one identity. The estimation is based on ground-truth 200+ sequences randomly picked among the dataset uploaded to the tool.

- **In Multiple Character Mode**, users can annotate multiple identities in a single panel simultaneously. This mode is useful when dealing with comics with many characters or when studying the interactions between characters.

The tool offers flexibility and ease of use by providing these two modes to accommodate different annotation needs and preferences.

The methodology used for sequence creation in the tool follows the approach proposed by Iyyer et al. [Iyyer et al., 2017] for cloze tasks. Concretely, I collected four consecutive panels from a comic page whenever such a grouping was possible. More than 10,000 sequences have been uploaded to the system; of these, over 3,000 have been manually annotated.

To enable users to label the sequences effectively, bounding boxes for bodies, faces, and speech bubbles are available for each panel. The speech bubbles are sourced from the COMICS [Iyyer et al., 2017] dataset, while the DASS detects the bounding boxes for bodies and faces [Topal et al., 2022] framework. This framework is state-of-the-art for detecting faces and bodies in drawings, particularly comics.

6.1.3 Sequence Selection Criteria

From the manually annotated sequences, which numbered over 3,000, I applied a filtering process to identify suitable sequences for evaluation. Specifically, the following two conditions were used to select the sequences:

- The sequence should include at least one character in the final panel associated with a text box.

- The sequence should be part of the text-cloze task dataset generated by the scripts shared by Iyyer et al. [Iyyer et al., 2017]. This filter was added to ensure that the evaluation would be performed on a subset of the primary dataset, allowing for a direct comparison of results.

After the selection process, 279 sequences were identified as suitable for evaluation. For each selected sequence, relevant information, such as the panel order, panel text content, and the type of text (narrative or dialogue), was extracted. If available, the character ID associated with the text box was also used.

6.1.4 Used Large Language Models

This chapter of the thesis utilized several large language models to investigate the effectiveness of prompting in text-cloze tasks for comics. The models used were GPT-2, GPT-Neo 125M, GPT-2 Medium, GPT-2 Large, GPT-2 XL, GPT-Neo 2.7B, GPT-J 6B [Radford et al., 2019, Black et al., 2021, Wang and Komatsuzaki, 2021] from Huggingface's transformers [Wolf et al., 2020], and also, text-ada-001 (Ada), text-babbage-001 (Babbage), text-curie-001 (Curie), text-davinci-003 (Davinci) models [Brown et al., 2020] from OpenAI [1].

The inclusion of the GPT-2 models in the study was based on their strong performance in previous studies of zero-shot tasks, where they produce competitive or SOTA results depending on the tasks, such as reading comprehension, summarization, and question answering [Radford et al., 2019].

[1] `https://platform.openai.com/docs/model-index-for-researchers`

Similarly, the GPT-Neo 2.7B and GPT-J 6B models were included due to their impressive performance in generating natural language text, including text completion tasks. These models are also notable for their large size and capacity to handle complex language tasks.

Lastly, the four OpenAI models, Davinci, Curie, Babbage, and Ada, differ in their capabilities and speed. Davinci is the most powerful and capable of solving complex text analysis problems such as summarization for a specific audience, creative content generation, and understanding the intent of the text. Curie is also powerful, with nuanced capabilities such as sentiment classification, question-answering, and language translation. Babbage is suitable for simple classification and semantic search tasks, while Ada is the fastest model for tasks like parsing text, address correction, simple classification, and keywords. It is important to note that any task performed by a faster model like Ada can also be performed by a more powerful model like Curie or Davinci [2]. For my study, only the Davinci model was able to perform well on the multiple-choice question-answering prompt framework.

6.1.5 Experimental Setup

The experimental setup was designed with several stages to carefully craft prompts for a loss-based answer-scoring strategy and multiple choice strategy to evaluate the performance of the models. In the first stage, the best prompt for the loss-based answer-scoring strategy is explored with prompt engineering. These prompts included instructions to the models, information for each panel, and specifications of panel details such as text, text type, and character identity. Since the models used in the first stage were smaller than 7B parameters, they were unable to provide accurate output for a multiple-choice format, favoring one option over the others in terms of log-likelihood.

The initial template for the first stage consisted of combinations of instructions, panel information, with or without character identity, and optional output specification. The best template was selected based on its performance on the GPT-2

[2] `https://platform.openai.com/docs/models/davinci`

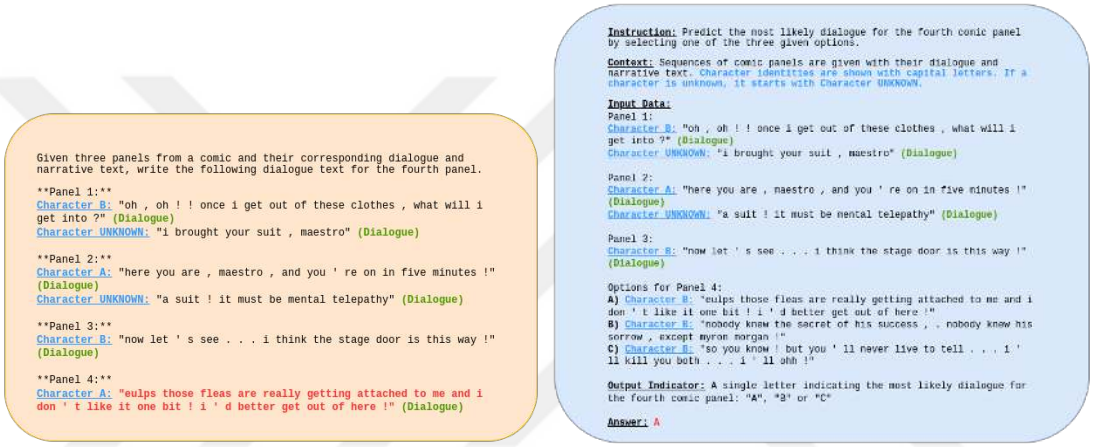


Figure 6.2: Comparison of Prompt ID 6 and Prompt ID 9 for Language Models. Prompt ID 6 (on the left) uses each option as a prompt-answer pair to calculate loss, with the option generating the lowest perplexity selected as the model's prediction. Prompt ID 9 (on the right) expects the model to generate the correct option from a set of choices (A, B, and C) and is only effective with large language models (10B+ parameters). Character identities are indicated in blue, while the type of speech bubble (Narrative or Dialogue) is indicated in green. The red text indicates the answer option or text generated by the model.

model. The performance of the prompts on GPT-2 can be seen in Figure 6.6, and the template can be seen in Figure 6.2. The performances of other language models and the effect of character ID within a prompt were also measured, as shown in Figure 6.7. Figure 6.8 shows all prompts used for this stage.

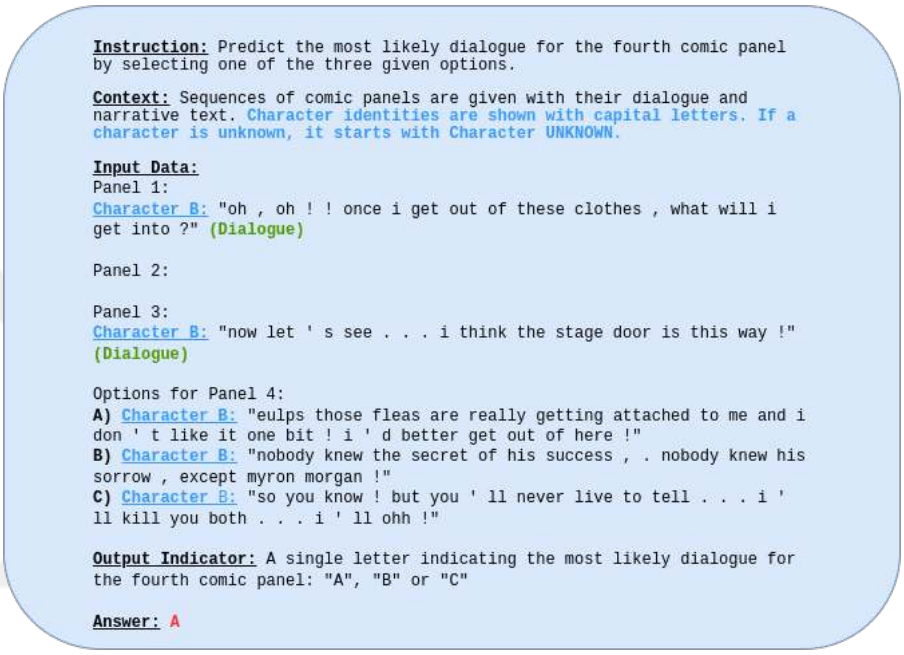


Figure 6.3: Prompt ID 10. This prompt was formed by removing dialogues of every character except the character in the final panel, who was queried, from Prompt ID 9. This was done to analyze the effect of dialogues from previous panels while predicting the dialogue of the last panel for EASY and HARD settings and their contribution to creating a coherent story.

In the second stage, the largest language model, Davinci, with approximately 175B parameters, was used for a multiple-choice question-answering framework. The prompt was further developed and more structured to take advantage of Davinci's superior content understanding ability. Several prompt versions were engineered to measure Davinci's performance, including prompts without character identities, prompts with character identities, prompts with only the texts of characters in the last panel, and zero-shot chain-of-thought (CoT) [Wei et al., 2022a] versions of the prompt. These experiments were conducted to understand the effect of character

identities and other characters on the story flow of the comic sequence. Zero-shot CoT was also explored to determine whether it could improve performance. These prompts can be seen in Figures 6.2, 6.3, and 6.4. After identifying the best prompt for the Davinci model, a few-shot prompting experiment was conducted to observe its effects on comic understanding. Figure 6.9 shows all prompts used for this stage.

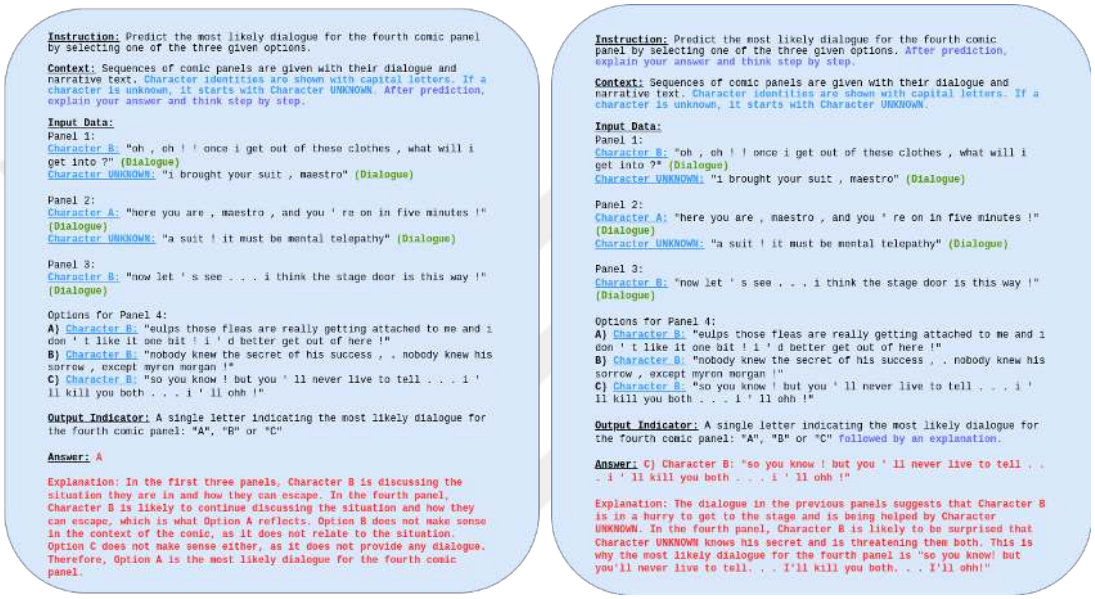


Figure 6.4: Comparison of Prompt ID 11 and Prompt ID 12 for Zero-Shot Chain of Thought (CoT) [Wei et al., 2022a] Prompting. Both prompts were designed to support zero-shot CoT generation but resulted in lower model performance. Purple text in the prompt indicates zero-shot CoT additions.

6.2 Results & Discussion

Table 6.1 presents Iyyer et al.'s [Iyyer et al., 2017] results for text-cloze tasks and shows human-level performance for the hard setting. My experimental setup for prompting language models to solve the text-cloze task belongs to the text-only model category, as the models only exploit the text modality. In the first stage of our experiments, I utilized a loss-based answer scoring strategy with GPT-2, and my best prompt achieved promising results for both tasks with a success rate

of 57.7% and 50.1% using prompt ID 6. This prompt includes character identities. However, prompt ID 7, which follows the same pattern but does not include character identities, does not fall short in comparison, as its performance for both tasks is only around 1% lower, as shown in Figure 6.6. Notably, the experiments with language models were conducted in a zero-shot fashion, indicating the generalization capabilities of language models for comics as well. During the prompt search, several important points for GPT-2 were observed:

- The sophistication of the output before panel four negatively affects performance, as evidenced by the difference between Prompt ID 2 and 1.

- The way character identity information is presented has a significant impact on performance. With the exception of prompt IDs 1, 2, and 7, all others include character identity information. However, in some cases, character identity information can even degrade performance, as seen in the difference between prompt IDs 2 and 3.

- The presentation of panel number information also affects performance. Leaving a space after the panel information increases the model's performance, as shown by the performance difference between Prompt 5 and 6.

I present the outcomes of the experimental setup where I observed the effects of few-shot learning (prompting). Figure 6.5 illustrates the variations in results for the text-cloze easy and hard settings based on the changing value of k, representing the number of few-shot exemplars. Upon examining the range of changes from 0 to 5, it is generally evident that an increase in the number of examples corresponds to an improvement in performance. Consequently, when providing 5 examples, it is observed that the text-davinci-003 model achieves the best results through prompting. Specifically, an accuracy of 80.6% is attained for the easy setting and 69.0% for the hard setting.

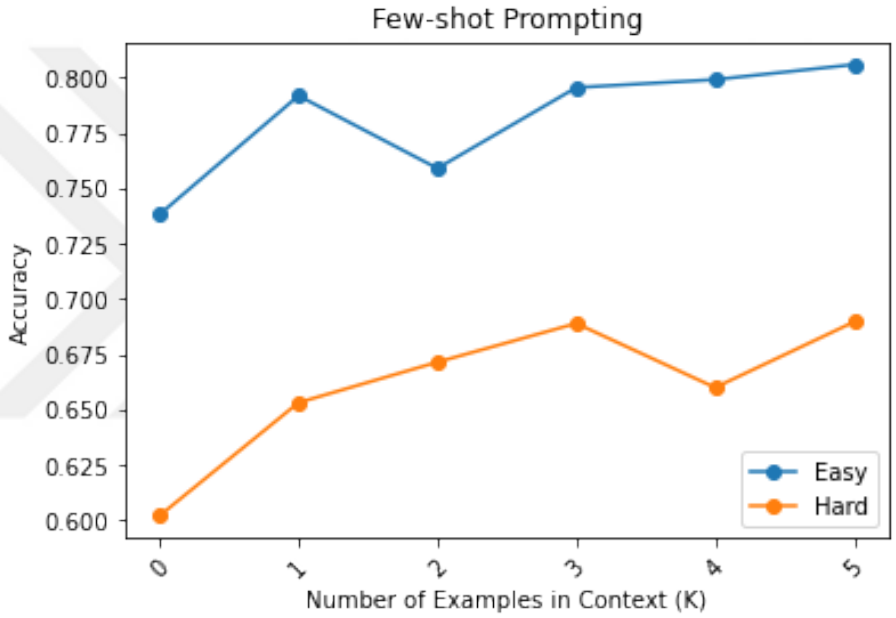


Figure 6.5: Effect of few-shot prompting [Brown et al., 2020] on accuracy for easy and hard settings in the text-cloze task. When k was set to 5, the results achieved the best results with prompting. Even when k was 1, it improved the results and surpassed the performance of the previous SOTA.

6.3 Conclusion

I identified the most effective prompt setting, which can serve as a template for future tasks involving comic prompting and similar applications. I observed model size's effects and character IDs' inclusion in the prompts. It emerged that providing character IDs to the language model as text did not significantly contribute, indicating the necessity to provide character information to models differently. Accordingly, I implemented this with ComicBERT and observed considerable improvements in the results.

Furthermore, the impact of Few-shot prompting on the results was observed, surpassing the performance of existing text-cloze results in the literature. When comparing these results with those in Section 5.4, it is evident that ComicBERT yields significantly better outcomes in the easy setting. In contrast, it achieves nearly similar results in the hard setting, though it was slightly better. It should be noted that these results are obtained using only text modality with LLM, but considering that the used LLM has 175 billion parameters. In comparison, ComicBERT has only 150 million parameters, so it is much smaller in terms of model capacity. Considering the differences in pretraining tasks and training procedures, it's important to acknowledge the distinction. From this, it can be concluded that surpassing the results obtained with ComicBERT using larger-scale models is possible, thereby approaching the human baseline more closely.

Table 6.1: The best results for the text-cloze task in comics using the neural network architecture proposed with the COMICS [Iyyer et al., 2017] dataset, both in easy and hard settings. The results were obtained using our *COMICS TEXT+* dataset, except the image-text model in the easy setting. Human-level performance for the hard setting is shown for reference.

MODEL	**Text Cloze**	
Task Difficulty	EASY	HARD
Text-only	64.5	55.0
Image-only	52.6	50.3
Image-text	68.6	61.4
Human	-	84

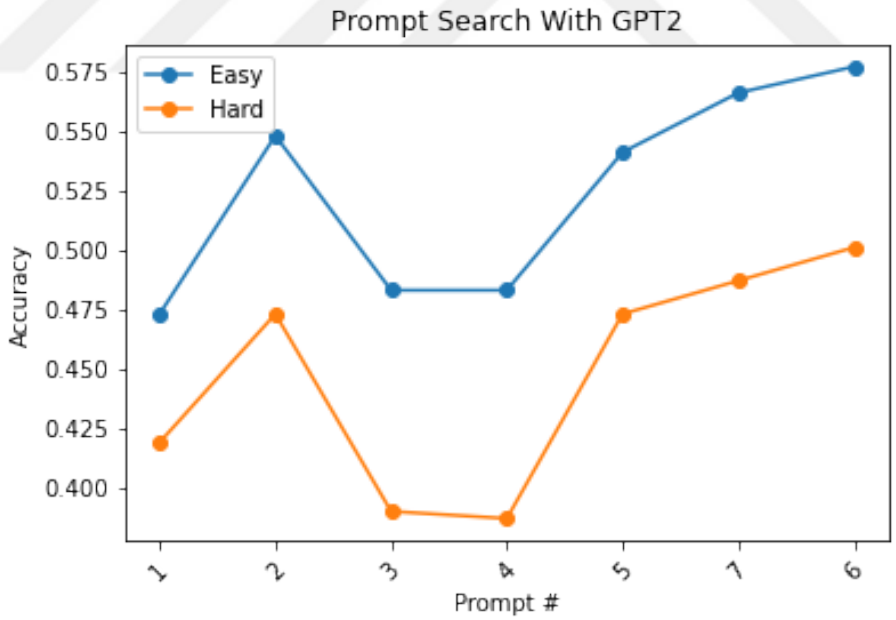


Figure 6.6: Comparison of prompt accuracy for Text-Cloze Task Difficulty Levels. The line chart shows the accuracy of easy and hard difficulty options for the text-cloze task, with the prompt number on the X-axis and accuracy on the Y-axis. The chart represents the search for better prompts for the text-cloze task. Prompt ID 6 is the version of Prompt ID 7, including character identities.

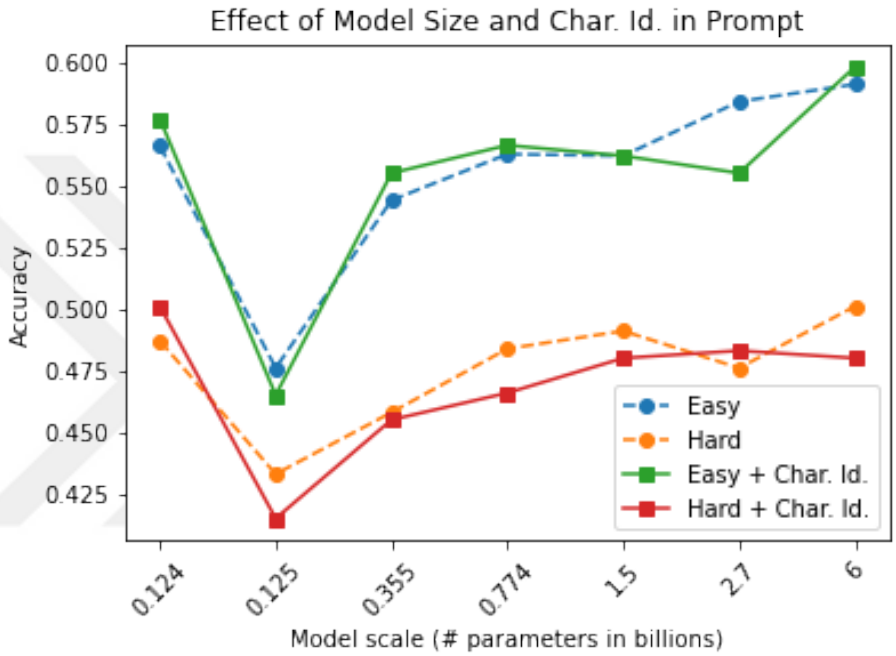


Figure 6.7: Performance comparison of Prompt ID 6 and 7 across language model sizes and task difficulty. The figure shows the performance of Prompt ID 6 and 7 across varying language model sizes and task difficulty. Adding character identities to the prompt improves the performance for easy settings but has an adverse effect on hard settings. The models shown from left to right are: GPT-2, GPT-Neo 125M, GPT-2 Medium, GPT-2 Large, GPT-2 XL, GPT-Neo 2.7B, and GPT-J 6B [Radford et al., 2019, Black et al., 2021, Wang and Komatsuzaki, 2021].

Figure 6.8: All loss-based answer scoring prompts by ID.

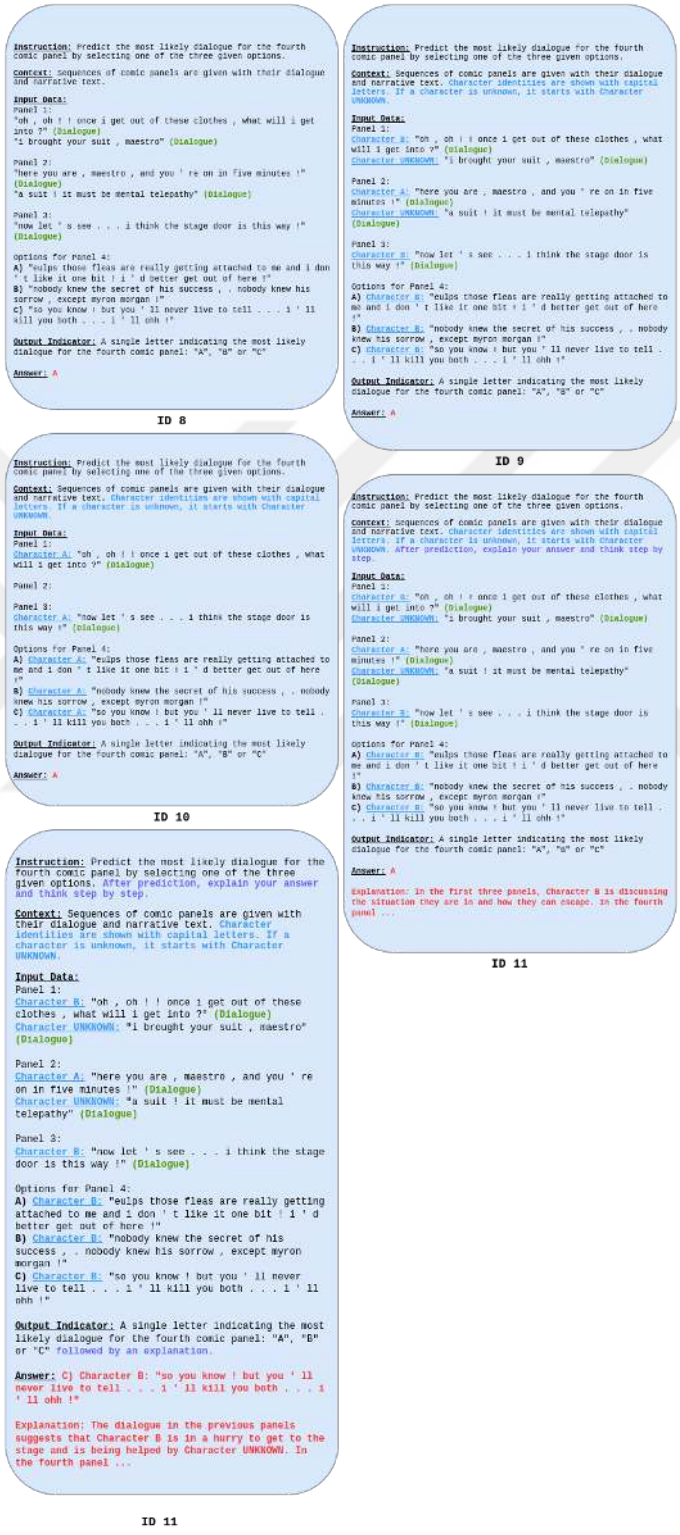


Figure 6.9: All multiple choice question-answering prompts by ID.

Table 6.2: Performance comparison of the text-davinci-003 [Brown et al., 2020] language model for our structured prompt types. The table shows the accuracy scores (in percentage) on two difficulty levels (EASY and HARD) for five prompt types. Prompt ID 8, which involves structured QA without character IDs, achieved the highest score on the HARD difficulty level, while Prompt ID 9, which includes character IDs in the prompts, achieved the highest score on the EASY level. Zero-shot CoT [Kojima et al., 2022] Structured QA v1 and v2 (Prompts 11 and 12, respectively) show promising results but lower scores than the other prompts.

Prompt Id.	**Prompt Desc.**	**EASY**	**HARD**
8	Structured QA	71.3	**61.3**
9	Structured QA with Char. Ids.	**73.8**	60.2
10	Structured QA with only answer Char. Ids.	60.2	60.2
11	Zero-shot CoT Structured QA v1	70.6	60.5
12	Zero-shot CoT Structured QA v2	66.6	56.2

Chapter 7

CONCLUSION

In the first phase of the thesis, I established a gold standard for extracting text data, particularly from speech bubbles and narrative box contents in comic books, which is used for OCR models. I benchmarked various text detection and recognition models. Furthermore, I curated suitable text detection and recognition datasets aligned with this gold standard. These datasets were then used to fine-tune appropriate OCR models, resulting in the creation of a text dataset for the golden age of comics, *COMICS Text+*. Through quantitative analysis, I demonstrated the superior quality of this dataset compared to the text data of *COMICS* dataset [Iyyer et al., 2017].

The second phase of the thesis was centered around multi-task learning. The tasks encompassed detection, segmentation, and association (relation). The detection component focused on detecting panels, speech bubbles, narrative boxes, character faces, and bodies. Meanwhile, the segmentation aspect revolved around speech bubbles and panels. The association (relation) task aimed to associate comic character faces and bodies with speech bubbles. I built upon the *COMIC MTL* [Nguyen et al., 2019] framework, enhancing it comprehensively from preprocessing to architectural components. I developed novel sampling and masking strategies and improved the outcomes further by applying the *Character Consistency Post-processing* method. Consequently, I achieved SOTA results in the comic character face and body-to-speech bubble association tasks while observing improvements in the detection and segmentation aspects of the MTL framework.

In the third phase of the thesis, my focus was on comic character reidentification. For the first time in the literature, the comic character reidentification task was attempted to be solved by utilizing both face and body information. Due to the scarcity of annotated data, I adopted a semi-supervised approach. In that, I devel-

oped the Identity-Aware Self-Supervision framework. Leveraging this framework, I employed a pre-trained model as a backbone, and through metric learning techniques such as miners and losses, I conducted semi-supervised fine-tuning on the limited data I collected as *Comic Sequence Identity Dataset* thru an annotation tool I developed. The efficacy of the comic character identity representations derived from the final model was assessed from intra-series and inter-series perspectives. Furthermore, by using the self-supervised backbone, unified and identity-aligned character embeddings can be extracted. Additionally, the backbone's ability to process face and body in the same network leads to parameter efficiency.

In the final stage of the thesis, utilizing the models and approaches I trained and developed in the preceding three sections, I extracted sequential panel data. The content of this data includes panel, speech bubble, narrative box, character face and body detections, speech bubble segmentations, and speech bubble-to-character associations. Additionally, I was able to derive character embeddings using the reidentification backbone and model on character instances. This data originated from the golden age of comic pages.

Following data preparation for the final phase, I developed a specialized transformer encoder model for processing comic data named *Comicsformer*. Additionally, I formulated the *ComicBERT* framework based on Comicsformer by pre-training it with a novel, comic-specific self-supervision task I referred to as *Masked Comic Modeling*. ComicBERT can be considered a multimodal foundation model capable of processing panels, character information, speech bubbles, and narrative box texts. By leveraging these capabilities, I achieved SOTA results in cloze-style tasks, thus displaying its effectiveness in extracting context and understanding sequential comic panels. This multimodal approach brings human-level performance within reach, particularly in text-cloze and visual-cloze tasks. Furthermore, I introduced a new task called *Scene-Cloze*, a task to predict the next panel as a scene with all of its content, and shared baseline results for this task for further research.

7.1 Future Work

In this section, I outline several potential avenues for future research and development that can build upon the foundations established in this thesis.

- **Enhancing Multimodality:** Expanding the utilization of multimodality within the proposed framework presents a promising direction for future work. At a fundamental level, considering the visual and linguistic aspects of the problem can be broken down into submodalities. For instance, additional components such as onomatopoeias, motion lines, and character poses can contribute to text or visual representations. Additionally, investigating multimodal interactions and their impact on narrative comprehension could lead to insightful studies, such as determining which modality predominantly influences narrative understanding.

- **Developing More Generalizable Models:** One of the limitations of this thesis is its focus on golden age comics. Future research could strive to create models and datasets that cater to a broader range of comic styles and periods, spanning European, Asian, and American comic works.

- **Cultural and Style Adaptation:** Investigating the adaptability of the proposed framework to diverse comic styles, genres, and cultural contexts could provide valuable insights into its applicability and boundaries. In addition, style transfer from one culture or drawing style might be interesting. It could even be considered as a translation task; instead of just translating text from one language to another for comics, more comprehensive translation schemas might be created.

- **Leveraging the Structure of Comics for Narrative Understanding:** The architecture of ComicBERT is extremelty suitable for exploring the *Visual Language of Comics* [Cohn, 2013]. Because visual language theory lays out the relationship between graphic and narrative structures as mapping between them. Exploration for this direction could involve collecting relevant data

and designing suitable self-supervision tasks where the aim is to predict the visual grammar of the sequence. This direction has the potential to unlock the underlying narrative understanding aspects of comics and even our visual perception.

- **Character Relations Exploration:** Exploring the identification and analysis of character relations within comics presents an exciting direction. The structure of ComicBERT could be utilized to process character outputs since it produces one-to-one embeddings, potentially shedding light on character interactions and dynamics.

- **Story Understanding and Generation:** Expanding into story-level comprehension and generation offers significant potential. In addition to cloze-style tasks, approaches like question-answering (QA) could be explored. Narrative text generation and even overall plot generation are potential areas of investigation. Furthermore, generating sequences of panels given a plot or enabling character-centered content generation are valuable prospects.

- **Generalization to Other Media:** The techniques and models developed in this thesis are not limited to comics alone. They could potentially be extended to other visual media forms, such as storyboards, graphic novels, and even video content. Since they share a very similar sequential nature.

BIBLIOGRAPHY

[Agrawal et al., 2023] Agrawal, H., Mishra, A., Gupta, M., et al. (2023). Multimodal persona based generation of comic dialogs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14150–14164.

[Aizawa et al., 2020] Aizawa, K., Fujimoto, A., Otsubo, A., Ogawa, T., Matsui, Y., Tsubota, K., and Ikuta, H. (2020). Building a manga dataset "manga109" with annotations for multimedia applications. *IEEE MultiMedia*, 27(2):8–18.

[Augereau et al., 2017] Augereau, O., Iwata, M., and Kise, K. (2017). An overview of comics research in computer science. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 3, pages 54–59. IEEE.

[Baek et al., 2022] Baek, J., Matsui, Y., and Aizawa, K. (2022). Coo: Comic onomatopoeia dataset for recognizing arbitrary or truncated texts. *arXiv preprint arXiv:2207.04675*.

[Balntas et al., 2016] Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3.

[Black et al., 2021] Black, S., Leo, G., Wang, P., Leahy, C., and Biderman, S. (2021). GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

[Bresenham, 1965] Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30.

[Brienza, 2010] Brienza, C. (2010). Producing comics culture: a sociological approach to the study of comics. *Journal of Graphic Novels and Comics*, 1(2):105–119.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Buslaev et al., 2020] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, 11(2):125.

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

[Chen et al., 2020b] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. (2020b). Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*.

[Chen et al., 2021] Chen, X., Jin, L., Zhu, Y., Luo, C., and Wang, T. (2021). Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–35.

[Cohn, 2013] Cohn, N. (2013). *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. A&C Black.

[Dai et al., 2017] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Dunst et al., 2017] Dunst, A., Hartel, R., and Laubrock, J. (2017). The graphic narrative corpus (gnc): design, annotation, and analysis for the digital humanities. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 3, pages 15–20. IEEE.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

[Fang et al., 2021] Fang, S., Xie, H., Wang, Y., Mao, Z., and Zhang, Y. (2021). Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition.

[Francis and Kucera, 1979] Francis, W. N. and Kucera, H. (1979). Brown corpus manual. *Letters to the Editor*, 5(2):7.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

[Gobbo and Matuk Herrera, 2020] Gobbo, J. D. and Matuk Herrera, R. (2020). Unconstrained text detection in manga: A new dataset and baseline. In *European Conference on Computer Vision*, pages 629–646. Springer.

[Goel, 2021] Goel, R. (2021). Contextual Spell Check.

[Gomez et al., 2017] Gomez, R., Shi, B., Gomez, L., Numann, L., Veit, A., Matas, J., Belongie, S., and Karatzas, D. (2017). Icdar2017 robust reading challenge on coco-text. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1435–1443.

[Guérin et al., 2013] Guérin, C., Rigaud, C., Mercier, A., Ammar-Boudjelal, F., Bertet, K., Bouju, A., Burie, J.-C., Louis, G., Ogier, J.-M., and Revel, A. (2013). ebdtheque: a representative database of comics. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1145–1149.

[Gupta et al., 2016] Gupta, A., Vedaldi, A., and Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Hadsell et al., 2006] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

[Hartel and Dunst, 2021] Hartel, R. and Dunst, A. (2021). An ocr pipeline and semantic text analysis for comics. In *International Conference on Pattern Recognition*, pages 213–222. Springer.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual

learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

[Herbst et al., 2011] Herbst, P., Chazan, D., Chen, C.-L., Chieu, V.-M., and Weiss, M. (2011). Using comics-based representations of teaching, and technology, to bring practice to teacher education courses. *ZDM*, 43(1):91–103.

[Hu et al., 2018] Hu, H., Gu, J., Zhang, Z., Dai, J., and Wei, Y. (2018). Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597.

[Inoue et al., 2018] Inoue, N., Furuta, R., Yamasaki, T., and Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009.

[Iyyer et al., 2017] Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., au2, H. D. I., and Davis, L. (2017). The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives.

[Iyyer, 2017] Iyyer, M. N. (2017). *Discourse-Level Language Understanding with Deep Learning*. PhD thesis.

[Jaderberg et al., 2016] Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20.

[Jayanthi et al., 2020] Jayanthi, S. M., Pruthi, D., and Neubig, G. (2020). NeuSpell: A neural spelling correction toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 158–164, Online. Association for Computational Linguistics.

[Karatzas et al., 2015] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., et al. (2015). Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE.

[Kojima et al., 2022] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

[Kuang et al., 2021] Kuang, Z., Sun, H., Li, Z., Yue, X., Lin, T. H., Chen, J., Wei, H., Zhu, Y., Gao, T., Zhang, W., Chen, K., Zhang, W., and Lin, D. (2021). Mmocr: A comprehensive toolbox for text detection, recognition and understanding. *arXiv preprint arXiv:2108.06543*.

[Laubrock and Dunst, 2020] Laubrock, J. and Dunst, A. (2020). Computational approaches to comics analysis. *Topics in cognitive science*, 12(1):274–310.

[Lee et al., 2020] Lee, J., Park, S., Baek, J., Oh, S. J., Kim, S., and Lee, H. (2020). On recognizing texts of arbitrary shapes with 2d self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 546–547.

[Li et al., 2019] Li, H., Wang, P., Shen, C., and Zhang, G. (2019). Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8610–8617.

[Li et al., 2021] Li, Y., Xie, S., Chen, X., Dollar, P., He, K., and Girshick, R. (2021). Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*.

[Liao et al., 2018] Liao, M., Shi, B., and Bai, X. (2018). Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690.

[Liao et al., 2017] Liao, M., Shi, B., Bai, X., Wang, X., and Liu, W. (2017). Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*.

[Liao et al., 2020] Liao, M., Wan, Z., Yao, C., Chen, K., and Bai, X. (2020). Real-time scene text detection with differentiable binarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11474–11481.

[Liao et al., 2022] Liao, M., Zou, Z., Wan, Z., Yao, C., and Bai, X. (2022). Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Long et al., 2018] Long, S., Ruan, J., Zhang, W., He, X., Wu, W., and Yao, C. (2018). Textsnake: A flexible representation for detecting text of arbitrary shapes. pages 20–36.

[Loshchilov and Hutter, 2016] Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[Loshchilov and Hutter, 2017] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[Lu et al., 2021] Lu, N., Yu, W., Qi, X., Chen, Y., Gong, P., Xiao, R., and Bai, X. (2021). MASTER: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*.

[Lyu et al., 2018] Lyu, P., Liao, M., Yao, C., Wu, W., and Bai, X. (2018). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83.

[Minaee et al., 2021] Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

[Musgrave et al., 2020a] Musgrave, K., Belongie, S., and Lim, S.-N. (2020a). A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer.

[Musgrave et al., 2020b] Musgrave, K., Belongie, S. J., and Lim, S.-N. (2020b). Pytorch metric learning. *ArXiv*, abs/2008.09164.

[Nguyen et al., 2018] Nguyen, N.-V., Rigaud, C., and Burie, J.-C. (2018). Digital comics image indexing based on deep learning. *Journal of Imaging*, 4(7).

[Nguyen et al., 2019] Nguyen, N.-V., Rigaud, C., and Burie, J.-C. (2019). Comic mtl: optimized multi-task learning for comic book image analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, 22:265–284.

[Nguyen et al., 2021a] Nguyen, N.-V., Rigaud, C., Revel, A., and Burie, J.-C. (2021a). Manga-mmtl: Multimodal multitask transfer learning for manga character analysis. In *International Conference on Document Analysis and Recognition*, pages 410–425. Springer.

[Nguyen et al., 2021b] Nguyen, N.-V., Vu, X.-S., Rigaud, C., Jiang, L., and Burie, J.-C. (2021b). Icdar 2021 competition on multimodal emotion recognition on comics scenes. In *International Conference on Document Analysis and Recognition*, pages 767–782. Springer.

[Nir et al., 2022] Nir, O., Rapoport, G., and Shamir, A. (2022). Cast: Character labeling in animation using self-supervision by tracking. In *Computer Graphics Forum*, volume 41, pages 135–145. Wiley Online Library.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Qin et al., 2019] Qin, X., Zhou, Y., Li, Y., Wang, S., Wang, Y., and Tang, Z. (2019). Progressive deep feature learning for manga character recognition via unlabeled training data. In *Proceedings of the ACM Turing Celebration Conference-China*, pages 1–6.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[Rigaud et al., 2015a] Rigaud, C., Le Thanh, N., Burie, J.-C., Ogier, J.-M., Iwata, M., Imazu, E., and Kise, K. (2015a). Speech balloon and speaker association for comics and manga understanding. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 351–355. IEEE.

[Rigaud et al., 2015b] Rigaud, C., Le Thanh, N., Burie, J.-C., Ogier, J.-M., Iwata, M., Imazu, E., and Kise, K. (2015b). Speech balloon and speaker association for comics and manga understanding. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 351–355. IEEE.

[Rigaud et al., 2016] Rigaud, C., Pal, S., Burie, J.-C., and Ogier, J.-M. (2016). Toward speech text recognition for comic books. In *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, pages 1–6.

[Sanh et al., 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

[Santoro et al., 2017] Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30.

[Sheng et al., 2019] Sheng, F., Chen, Z., and Xu, B. (2019). Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 781–786. IEEE.

[Shi et al., 2017] Shi, B., Bai, X., and Belongie, S. (2017). Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2550–2558.

[Shi et al., 2016a] Shi, B., Bai, X., and Yao, C. (2016a). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*.

[Shi et al., 2016b] Shi, B., Wang, X., Lyu, P., Yao, C., and Bai, X. (2016b). Robust scene text recognition with automatic rectification.

[Su et al., 2021] Su, Y., Liu, F., Meng, Z., Lan, T., Shu, L., Shareghi, E., and Collier, N. (2021). Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

[Sunder et al., 2022] Sunder, V., Fosler-Lussier, E., Thomas, S., Kuo, H.-K. J., and Kingsbury, B. (2022). Tokenwise contrastive pretraining for finer speech-to-bert alignment in end-to-end speech-to-intent systems. *arXiv preprint arXiv:2204.05188*.

[Tan and Le, 2019] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model

scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

[Tang et al., 2019a] Tang, J., Yang, Z., Wang, Y., Zheng, Q., Xu, Y., and Bai, X. (2019a). Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern recognition*, 96:106954.

[Tang et al., 2019b] Tang, K., Zhang, H., Wu, B., Luo, W., and Liu, W. (2019b). Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6619–6628.

[Topal et al., 2022] Topal, B. B., Yuret, D., and Sezgin, T. M. (2022). Domain-adaptive self-supervised pre-training for face & body detection in drawings. *arXiv preprint arXiv:2211.10641*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wada, 2018] Wada, K. (2018). labelme: Image polygonal annotation with python. `https://github.com/wkentaro/labelme`.

[Wang and Komatsuzaki, 2021] Wang, B. and Komatsuzaki, A. (2021). GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

[Wang et al., 2019a] Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., and Shao, S. (2019a). Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345.

[Wang et al., 2019b] Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu,

G., and Shen, C. (2019b). Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, pages 8439–8448.

[Wang et al., 2019c] Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. (2019c). Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030.

[Wei et al., 2022a] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022a). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

[Wei et al., 2022b] Wei, W., Yang, W., Zuo, E., Qian, Y., and Wang, L. (2022b). Person re-identification based on deep learning—an overview. *Journal of Visual Communication and Image Representation*, 82:103418.

[Wick et al., 2020] Wick, C., Reul, C., and Puppe, F. (2020). Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. *Digital Humanities Quarterly*, 14(1).

[Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

[Wu et al., 2019] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. `https://github.com/facebookresearch/detectron2`.

[Xie et al., 2019] Xie, L., Liu, Y., Jin, L., and Xie, Z. (2019). Derpn: Taking a further step toward more general object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9046–9053.

[Yu and Tao, 2019] Yu, B. and Tao, D. (2019). Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6490–6499.

[Yue et al., 2020] Yue, X., Kuang, Z., Lin, C., Sun, H., and Zhang, W. (2020). Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *European Conference on Computer Vision*.

[Yuliang et al., 2017] Yuliang, L., Lianwen, J., Shuaitao, Z., and Sheng, Z. (2017). Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*.

[Zhang et al., 2020] Zhang, S.-X., Zhu, X., Hou, J.-B., Liu, C., Yang, C., Wang, H., and Yin, X.-C. (2020). Deep relational reasoning graph network for arbitrary shape text detection. pages 9699–9708.

[Zhang et al., 2022] Zhang, Z., Wang, Z., and Hu, W. (2022). Unsupervised manga character re-identification via face-body and spatial-temporal associated clustering. *arXiv preprint arXiv:2204.04621*.

[Zheng et al., 2020a] Zheng, W., Yan, L., Wang, F.-Y., and Gou, C. (2020a). Learning from the past: Meta-continual learning with knowledge embedding for jointly sketch, cartoon, and caricature face recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 736–743.

[Zheng et al., 2020b] Zheng, Y., Zhao, Y., Ren, M., Yan, H., Lu, X., Liu, J., and Li, J. (2020b). Cartoon face recognition: A benchmark dataset. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2264–2272.

[Zhu et al., 2021] Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., and Zhang, W. (2021). Fourier contour embedding for arbitrary-shaped text detection. In *CVPR*.