

IMAGE ALIGNMENT FOR VISIBLE AND INFRARED IMAGES TAKEN BY AERIAL SYSTEMS



ALAIN PATRICK NDIGANDE

ÖZYEGİN UNIVERSITY

AUGUST, 2025

IMAGE ALIGNMENT FOR VISIBLE AND INFRARED IMAGES TAKEN BY AERIAL SYSTEMS

by
Alain Patrick Ndigande

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Computer Science

Advisor: Asst. Prof. Sedat Özer

Graduate School of Science and Engineering
Özyeğin University
İstanbul

August, 2025

IMAGE ALIGNMENT FOR VISIBLE AND INFRARED IMAGES TAKEN BY AERIAL SYSTEMS

Approved by:

Asst. Prof. Sedat Özer
Department of Computer Science
Özyeğin University

Assoc. Prof. Emre Sefer
Department of Artificial Intelligence and Data
Engineering
Özyeğin University

Prof. Hakan Ali Çırpan
Department of Electronics and
Communication Engineering
Istanbul Technical University

Approval Date: July 03, 2025

To my dear father, loving mother, and cherished friends.



DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Alain Patrick Ndigande

ABSTRACT

Aerial images acquired by satellites, unmanned aerial vehicles (Unmanned Aerial Vehicle (UAV)s) or by other means such as balloons, are used in many applications in today's world including entertainment, agriculture, delivery, surveillance, rescue, and disaster management applications. Recent advances in technology and storage hardware allowed cameras to record videos as opposed to individual images in aerial vehicles in multiple modalities. Consequently, there is a recent trend in the relevant research community to leverage such information, when the goal is aerial image or video analysis. As a particular example, timely and accurate segmentation and association of objects of interest in such aerial video frames has been an important research field, recently.

When the data is acquired by different cameras focusing on infrared and visible spectrum individually, a particular and an immediate problem is aligning the images (or videos) captured by such different cameras. Image alignment (or registration) is, essentially, the task where an image taken by a different camera is mapped onto the coordinate system of the other camera so that the same objects appear at the same coordinate on both images. In this thesis, we mainly focus on image alignment problem. We develop and propose a deep learning-based solution and compare its performance to traditional and other recent deep learning-based solutions from the relevant literature on multiple datasets. In addition to the image alignment, as additional tasks where such alignment can later be used as input, and as a part of future work, we provide a study on video instance segmentation algorithms that summarizes recent relevant advances and compares their performances.

Main contributions in this thesis include: (i) proposing a technique to align images acquired by different cameras on the same edge device, (ii) comparing and reporting the performance of such alignment algorithms on multiple datasets focusing on aerial

datasets, (iii) a comprehensive study summarizing the performance evaluations of recently proposed state-of-the-art Video Instance Segmentation (VIS) models on multiple datasets, (iv) presenting an approach on replacing the trainable layers in VIS applications with the non-trainable Fourier based layers.

Keywords: Multimodal image registration, image alignment, Visible-Infrared image registration, Video Instance Segmentation, Online/Offline VIS, Instance Association, Query-based VIS, UAV image processing



ÖZET

Uydu, insansız hava aracı (İHA) veya balon gibi araçlarla elde edilen hava görüntüleri, günümüzde eğlence, tarım, teslimat, gözetim, kurtarma ve afet yönetimi gibi pek çok alanda kullanılmaktadır. Teknoloji ve depolama donanımlarındaki son gelişmeler, hava araçlarındaki kameraların tek tek görüntüler yerine video kaydetmesine imkân tanımıştır. Bu doğrultuda, ilgili araştırma topluluğunda, hedef hava görüntüsü veya video analizi olduğunda bu zamana bağlı bilgidен yararlanma eğilimi artmıştır.

Özellikle, hava videolarındaki ilgi nesnelerinin zamanında ve doğru bir şekilde segmentasyonu ve takip edilmesi son yıllarda önemli bir araştırma alanı hâline gelmiştir. Veriler kızılötesi ve görünür spektrum tabanlı farklı kameralarla elde edildiğinde, acil ve temel bir sorun, aynı düzlem üzerindeki bu farklı kameraların görüntülerini (veya videolarını) birbirleriyle hizalamaktır. Görüntü hizalama (veya kayıt işlemi), temelde farklı bir kamera tarafından çekilen bir görüntüyü diğer kameranın koordinat sistemine haritalayarak aynı nesnelerin her iki görüntüde de aynı koordinatta görünmesini sağlamaktır. Bu tezde, nesne algılama ve video nesne örnek segmentasyonu gibi diğer görevler için yardımcı olabilecek görüntü hizalama üzerine odaklanıyoruz. Derin öğrenme tabanlı bir çözüm geliştiriyor ve öneriyoruz; bu çözümün performansını literatürdeki geleneksel ve diğer derin öğrenme tabanlı yöntemlerle birden fazla veri kümesi üzerinde karşılaştırıyoruz. Görüntü hizalamanın yanı sıra, gelecekteki çalışmalar kapsamında, çalışmaları özetleyen ve performanslarını karşılaştıran bir video örnek segmentasyonu çalışması da sunuyoruz.

Bu tezdeki ana katkılar şunlardır: (i) Farklı kameralarla aynı uç cihazda elde edilen görüntüleri hizalamaya yönelik bir tekniğin önerilmesi, (ii) Bu tür hizalama algoritmalarının birden çok veri kümesindeki performanslarının karşılaştırılıp raporlanması, (iii) Yakın zamanda önerilen son teknoloji Video Örnek Segmentasyonu (VIS) modellerinin çoklu veri kümeleri üzerindeki performans değerlendirmelerini özetleyen kapsamlı bir

alışmanın sunulması, (iv) VIS uygulamalarındaki eğitilebilir katmanların eğitilemeyen Fourier tabanlı katmanlarla değıştirilmesine yönelik bir yaklaşımın ortaya konması.

Anahtar Kelimeler: Multimodal image registration, image alignment, Visible-Infrared image registration, Video Instance Segmentation, Online/Offline VIS, Instance Association, Query-based VIS, UAV image processing



ACKNOWLEDGEMENTS

I would like to begin by extending my sincere gratitude to my advisor, Assistant Professor Sedat Özer, for his insightful guidance, unwavering support, and continual encouragement and mentorship throughout my graduate studies and research at Özyeğin University. His expertise and mentorship have been invaluable in shaping both my research direction and my professional growth.

I am very grateful to my instructors, colleagues, and all unnamed individuals whose valuable discussions, technical insights, and camaraderie have greatly contributed to my success. Their readiness to troubleshoot problems, share ideas, and provide feedback has helped refine my understanding of the subject matter which played a significant role in shaping my technical and research skills.

Special thanks to my family and friends for their continuous encouragement, patience, and belief in my aspirations, as their support was instrumental in helping me to achieve my goals. Their unfailing optimism and moral support, which sustained me during challenging times. Their encouragement and belief in my abilities have been constant motivators that propelled me forward.

I would like to extend my appreciation to the administrative and technical staff at Özyeğin University for providing the necessary resources and a stimulating environment that encouraged my growth and learning.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iv
ABSTRACT	v
ÖZET	vii
ACKNOWLEDGEMENTS	ix
LIST OF TABLES	xii
LIST OF FIGURES	xiv
LIST OF ACRONYMS AND ABBREVIATIONS	xvii
1 INTRODUCTION	1
1.1 Image Registration	1
1.2 Video Instance Segmentation	3
1.3 Contributions	4
2 DEEP IMAGE ALIGNMENT FOR UAV-TAKEN VISIBLE AND INFRARED IMAGE PAIRS	6
2.1 An Overview of Recent Image Registration Techniques	6
2.2 Preliminaries	9
2.3 VisIRNet (Proposed Approach) Network Structure	10
2.4 Used Loss Functions	15
2.5 Used Datasets	17
2.6 Evaluation Metrics	19
2.7 Quantitative Registration Results	20
2.8 Qualitative Registration Results	26
3 PERFORMANCE COMPARISON OF RECENT ADVANCES IN QUERY BASED VIDEO INSTANCE SEGMENTATION	28
3.1 A Summary of Video Instance Segmentation Methods	28
3.2 Transformer Based Architectures in VIS	31

3.3	Video Instance Segmentation Preliminaries	34
3.4	Instance Association and Contrastive Learning	37
3.5	Used Baseline Models	39
3.6	Experimental Setup	40
3.7	Used Video Instance Segmentation Datasets	41
3.8	Used VIS Evaluation Metrics.....	42
3.9	Experimental Results and Discussion	43
4	FOURIER BASED QUERY MIXING FOR ONLINE VIDEO INSTANCE SEG- MENTATION.....	47
4.1	Background: Fourier-Based Query Mixing	47
4.2	Revisiting Transformer Based VIS Pipelines	48
4.3	Fourier-Based Attention Layers for VIS	50
4.4	Experiments and Discussion.....	51
5	CONCLUSION.....	54
	REFERENCES	56

LIST OF TABLES

Table 2.1:	This table provides the layer-by-layer details of the feature embedding block. There are also skip connections between the layers in this architecture as shown in Figure 2.2.	11
Table 2.2:	This table provides the layer-by-layer details of the regression block as shown in Figure 2.2. Levels indicated by L are groups of <i>conv2D + BatchNormalization + Relu</i> . The <i>Conv2D</i> layers in each level have the same characteristics and filter dimensions. The number of used filters increases as we get deeper in the architecture.....	13
Table 2.3:	Ablation study on using different combinations of loss functions on two different datasets. The loss functions shown in each row are used for the MMFEB block, and the loss functions shown in each column ($\mathcal{L}_{Ace}$ and L_2^H) are used for the regression block in our model. Best results are shown in bold. Ace is the metric used to compute the results for each loss function combination. The last column shows the average Ace value for each loss function used in the MMFEB block. On average, $\mathcal{L}_{sim}$ yielded the best results.	17
Table 2.4:	A summary of the used datasets in our experiments is given. The training and test data are generated as explained in Section 2.4.	18
Table 2.5:	Comparative results of algorithms on each dataset in terms of Ace. Best results are shown in bold. The results illustrate that on average traditional SIFT performed on average in datasets of single or close modalities. Learning based algorithms DHN and MHN directly predict homography matrix without learning common representation also suffer especially on datasets such as Skydata and Google-maps.....	21
Table 2.6:	A comparison of the results of ModelA and ModelB at various hyperparameters including batchsizes and loss functions. The top table shows the Homography error in (a), while the bottom table shows the results as Average corner error in (b). Note that the results illustrating low homography error doesn't necessarily imply small registration error. we study the effect of batchsizes and loss functions. directly predicting homography matrix work but it does not minimize the registration error as predicting direct corners in our experiments. These results are obtained on Skydata dataset.	22

Table 3.1: This table presents a comparison of different video instance segmentation models based on their performance as the number of queries per image is varied. The evaluation is conducted on both the YouTube-VIS 2019 and OVIS validation sets. Results are grouped by query count for each method, providing insights into how the number of queries affects model accuracy and efficiency across different datasets.	43
Table 4.1: Results for YoutubeVis-2019 reported as mean average precision for both standard, and Fourier-based attentions on four VIS models.	53



LIST OF FIGURES

- Figure 1.1:** An overview of the image alignment process is shown. On the left, input RGB, I_{RGB} (192×192 pixels) and IR, I_{IR} (128×128 pixels) images are shown. The I_{IR} is shown in pseudocolors. Both images are given as input to the registration stage where the transformation parameters represented by the homography matrix (H) are predicted. After the registration process, the I_{IR} is transformed (warped) onto the I_{RGB} space by locating the positions of c_1 , c_2 , c_3 , and c_4 as c'_1 , c'_2 , c'_3 , and c'_4 . The warped I_{IR} is overlayed (where $\alpha = 0.4$) on the I_{RGB} 2
- Figure 1.2:** VIS is the task that handles the simultaneous segmentation, and temporal association of object instances across video frames. This figure demonstrates sample results on two videos where the objects in Frame t are associated with the segmented objects in the following frames. 3
- Figure 2.1:** In this figure, we summarize the architectures of various recently proposed deep alignment algorithms including DHN [1], MHN [2], CLKN [3] and Deep Lucas-Kanade Feature Map Network (DLKFM) [4]. While DHN and MHN predict the homography parameters $\mathbf{H}$; CLKN and DLKFM rely on using Lucas-Kanade (LK) based iterative approach and they use feature maps at different resolutions. By doing so, they predict homography in steps $\mathbf{H}_i$ where each step aims to correct the previous prediction. 8
- Figure 2.2:** This figure provides an overview of our proposed network architecture. Two parallel branches including RGB branch and IR branch (feature embedding blocks) extract the salient features for RGB and IR images, respectively. Those features are, then channel-wise concatenated and fed into the regression block for direct (ModelB) or indirect (ModelA) homography prediction. I.e., the model can be trained for learning the homography matrix in ModelB or to regress the corresponding coordinates of the four corners of the input IR image on the RGB image in ModelA. The output is 8 dimensional vector (for $\mathbf{H}$), if ModelB is used; and it is 8 dimensional vector where those 8 values correspond to the (x, y) coordinates of the 4 corners of the IR image, if modelA is used. The details of the feature embedding block are given on the top corner of the figure (also see Table 2.1). The details of the regression block are given in the lower right corner of the figure (also see Table 2.2). 12

Figure 2.3:	This figure shows how to select initial corner points on the registered image pairs and how to generate the training data. First a random image patch is taken from the originally registered IR image. Then the random corners of that patch is transformed into fixed coordinates and after that, the $\mathbf{H}$ matrix (and its inverse) performing that transformation is computed.	18
Figure 2.4:	Each plot shows the Average corner error for the algorithms including MHN, DHN, DLKFM, CLKN, SIFT and ours for a different dataset. The legends used in the plots are also given in the lower right corner of the figure.	23
Figure 2.5:	Average corner error distribution vs. count of image pairs for different models is shown on the test set of Skydata.	24
Figure 2.6:	This figure shows qualitative results on sample image pairs taken from different datasets. The first two columns show the input image pairs for the algorithms. The target image is 192×192 pixels and the source image is 128×128 pixels (which covers a scene that is a subset of the target image). The third column shows the ground truth version (192×192 pixels) of the source image on the coordinate system of the target image after being warped. The fourth column shows the ground truth (warped) where the source image is overlayed on the target image (192×192 pixels). The remaining 6 columns show the overlayed results (192×192 pixels), after applying registration with the algorithms in the order of SIFT, DHN, MHN, CLKN, DLKFM and our approach, respectively. Visually, each algorithms' result can be compared to the image in the fourth column.	25
Figure 3.1:	Query-based detection and instance segmentation architectures follow method proposed in DETR. Object queries are sufficient for a sufficient for a successful detection and segmentation. From object queries a class, mask and operational box heads are used to obtain instance masks and corresponding categories. Following the paradigm of detection, transform the fixed.	33
Figure 3.2:	Instance Association and Contrastive Learning in Query based VIS architectures.	37
Figure 3.3:	Youtube-VIS Qualitative results. Each rows represents the method of interest while the column shows different instance masks and instance IDs shown in different colours. The timesteps shown here are $t = [1, 30, 60, 90]$	45

Figure 3.4:	OVIS Qualitative results. Each rows represents the method of interest while the column shows different instance masks and instance IDs shown in different colours. The timesteps shown here are $t = [1, 15, 30, 45]$	46
Figure 4.1:	We provide the overview of the used instance segmentation approaches in all of the used VIS architectures in (A). There are two possible instance segmentation approaches shown in this figure: (i) segmentation via the classic attention mechanism as shown in (B); (ii) segmentation via Fourier-based attention as shown in (D). Multiple scale feature maps from the backbone go into pixel-decoder to arrive at higher-resolution mask features as shown in (A). The transformer decoder further takes these together with query features. Classes and masks are computed from both mask-features and context-aware instance embeddings. (C) illustrates the standard self-attention mechanism, while (D) shows the Fourier-based replacement module.	48
Figure 4.2:	Qualitative results for CTVIS on Youtube-VIS 2019 sequence <i>d7203fdab6</i> ; top pairs, and <i>ec6fd219</i> ; bottom pairs. The shown frames are the timesteps: 0, 30, 60, and 90 (from left to right). Object colors represent the object IDs. Here S.A. and F.A. stands Standard and Fourier Attention based models respectively.	52

LIST OF ACRONYMS AND ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
VIS	Video Instance Segmentation
CNN	Convolutional Neural Network
LK	Lucas-Kanade
DHN	Deep Homography Network
GAN	Generative Adversarial Network
CLKN	Cascade Lucas-Kanade Network
DLKFM	Deep Lucas-Kanade Feature Map Network
CNN	Convolutional Neural Network

1. INTRODUCTION

This section formally introduces the image registration and video instance segmentation tasks. It emphasizes the applications of aerial and unmanned aerial vehicle (UAV) by leveraging computer vision models.

1.1 Image Registration

Recent advancements in Unmanned Aerial Vehicle (UAV) technologies, computing and sensor technologies, allowed use of UAVs for various earth observation applications. Many UAV systems are equipped with multiple cameras today, as cameras provide reasonable and relatively reliable information about the surrounding scene in the form of multiple images or image pairs. Such image pairs can be taken by different cameras, at different view-points, different modalities or at different resolutions. In such situations, the same objects or the same features might appear at different coordinates on each image and, therefore, an image alignment (registration) step is needed prior to applying many other image based computer vision applications such as image fusion, object detection, segmentation or video instance segmentation as in [5, 6, 7].

The infrared spectrum and visible spectrum may reflect different properties of the same scene. Consequently, images taken in those modalities, typically, differ from each other. On many digital cameras, the visible spectrum is captured and stored in the form of Red-Green-Blue (RGB) image model and a typical visible spectrum camera captures visible light ranging from approximately 400 nanometers to 700 nanometers in wavelength [8, 9, 10]. Infrared cameras, on the other hand, capture wavelengths longer than those of visible light, falling between 700 nanometers and 10000 nanometers [11]. Infrared images can be further categorized into different wavelength ranges as in near-infrared (NIR), mid-infrared (MIR), and far-infrared (FIR) capturing different types of information in the spectrum [11, 12, 13, 14].

Figure 1.1: An overview of the image alignment process is shown. On the left, input RGB, I_{RGB} (192×192 pixels) and IR, I_{IR} (128×128 pixels) images are shown. The I_{IR} is shown in pseudocolors. Both images are given as input to the registration stage where the transformation parameters represented by the homography matrix (H) are predicted. After the registration process, the I_{IR} is transformed (warped) onto the I_{RGB} space by locating the positions of c_1, c_2, c_3 , and c_4 as c'_1, c'_2, c'_3 , and c'_4 . The warped I_{IR} is overlaid (where $\alpha = 0.4$) on the I_{RGB} .

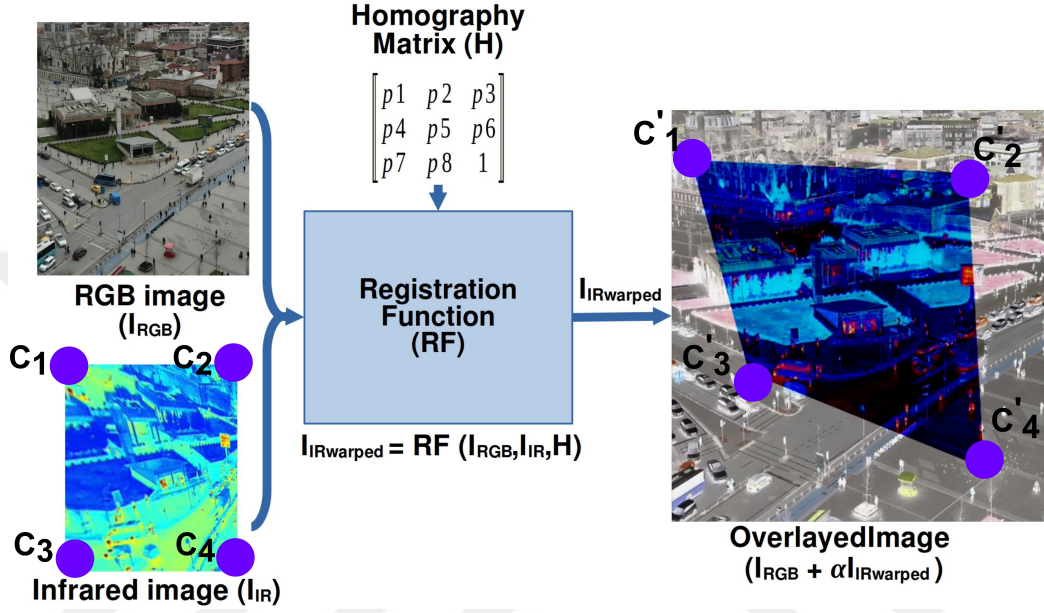


Image alignment is, essentially, the process of mapping the pixel coordinates from different coordinate system(s) into one common coordinate system. This problem is studied under different names including image registration and image alignment. We will also use the terms *alignment* and *registration* interchangeably in this thesis. Typically, alignment is done in the form of image pairs mapping from one image (source) onto the other one (target) [15]. Image alignment is a common problem that exists in many image-based applications where both of the target and source images can be acquired by sensors using the same modality or using different modalities. There is a wide range of applications of image alignment in many fields including medical imaging [16, 17], UAV applications [18, 19], image stitching [20] and remote sensing applications [21, 22, 23, 24, 25].

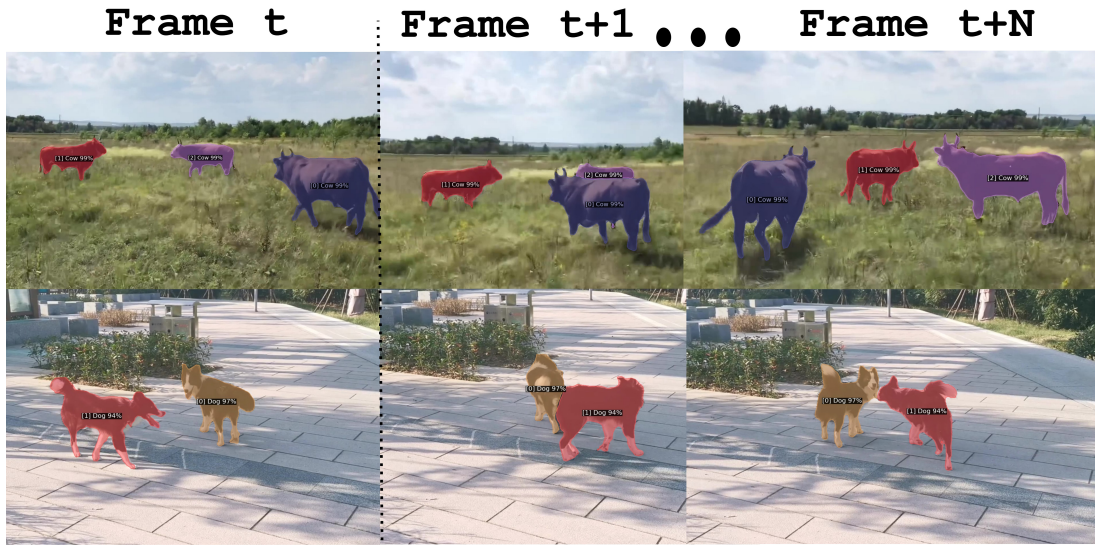
Image alignment, in many cases, can be reduced to the problem of estimating

the parameters of the perspective transformation between two images acquired by two separate cameras, where we assume that the cameras are located on the same UAV system. Fig. 1.1 summarizes such an image alignment process where the input consists of a higher resolution RGB image (e.g., 192×192 pixels) and a lower resolution IR image (e.g., 128×128 pixels visualized in pseudocolors in the figure). The output of the registration algorithm is the registered (aligned) IR image on the RGB image's coordinate system. As perspective transformation [26] is typically enough for UAV setups containing nearby onboard cameras, our registration process uses a registration function based on the Homography ($\mathbf{H}$) matrix. $\mathbf{H}$ contains 8 unknown (projection) parameters and the goal of the registration process is predicting those 8 unknown parameters, directly or indirectly.

1.2 Video Instance Segmentation

VIS is a rapidly evolving area of computer vision by going beyond the capabilities of the foundational tasks of object detection and semantic segmentation. In a classical

Figure 1.2: *VIS is the task that handles the simultaneous segmentation, and temporal association of object instances across video frames. This figure demonstrates sample results on two videos where the objects in Frame t are associated with the segmented objects in the following frames.*



detection task, objects in an image are identified and localized via bounding boxes [27], while a segmentation task takes a more fine-grained approach by classifying the exact pixels belonging to each object [28]. VIS extends these tasks from the image (spatial) to the video (spatio-temporal) domain, requiring not only accurate detection and segmentation of objects in individual frames but also robust association of these object instances across time in different video frames [28, 29]. A typical VIS output is shown on three frames selected from two videos in Fig. 1.2.

The inherent dynamics of video data, such as rapid motion, frequent occlusions, and appearance changes introduce additional challenges. Early works such as MaskTrack R-CNN [30], and many subsequent approaches, rely on separate segmentation and association modules (tracking heads on top of segmentation) frameworks to link instance masks across frames [31]. Moreover, many VIS models are designed as offline systems that process entire video clips at once [32], achieving high accuracy at a significant computational cost. In contrast, online VIS models, such as [33, 29, 34], operate on a frame-by-frame basis by making the association as soon as a new frame is received, making them more suitable for real-time applications.

1.3 Contributions

In this thesis, we propose a novel deep approach for registering IR and RGB image pairs, where instead of predicting the homography matrix directly, we predict the location of the four corner points of the entire image directly. This approach removes the additional iterative steps introduced by LK based algorithms and eliminates the steps of computing and finding important keypoints. (i) we introduce a novel deep approach for alignment problems of IR images onto RGB images taken by UAVs, where the resolutions of the input images differ from each other; (ii) we introduce a novel two-branch based deep solution for registration without relying on the Lukas-Kanade based iterative methods;

(iii) instead of predicting the homography matrix directly, we predict the corresponding coordinates of the four corner points of the smaller image on the larger image; (iv) we study and report the performance of our approach on multiple aerial datasets and present the state of the art results.

In this thesis, we further provide a comprehensive study summarizing the performance evaluations of recently proposed state-of-the-art VIS models on multiple datasets and we use the Fourier-mixing concept into state-of-the-art video instance segmentation models. We evaluate these concepts on quality of instance masks and models ability to maintain consistency over time on Youtube-VIS 2019 and OVIS and demonstrate video instance segmentation is still feasible by replacing the trainable layers in VIS applications with the non-trainable Fourier based layers.

2. DEEP IMAGE ALIGNMENT FOR UAV-TAKEN VISIBLE AND INFRARED IMAGE PAIRS

In this chapter, we first formalize the homography transformation and its estimation for aligning visible and infrared UAV images. We then discuss methods for image alignment alongside Lucas–Kanade based registration pipelines. Finally, we introduce our proposed approach and present the evaluation datasets, metrics and both comparative qualitative and quantitative results.

2.1 An Overview of Recent Image Registration Techniques

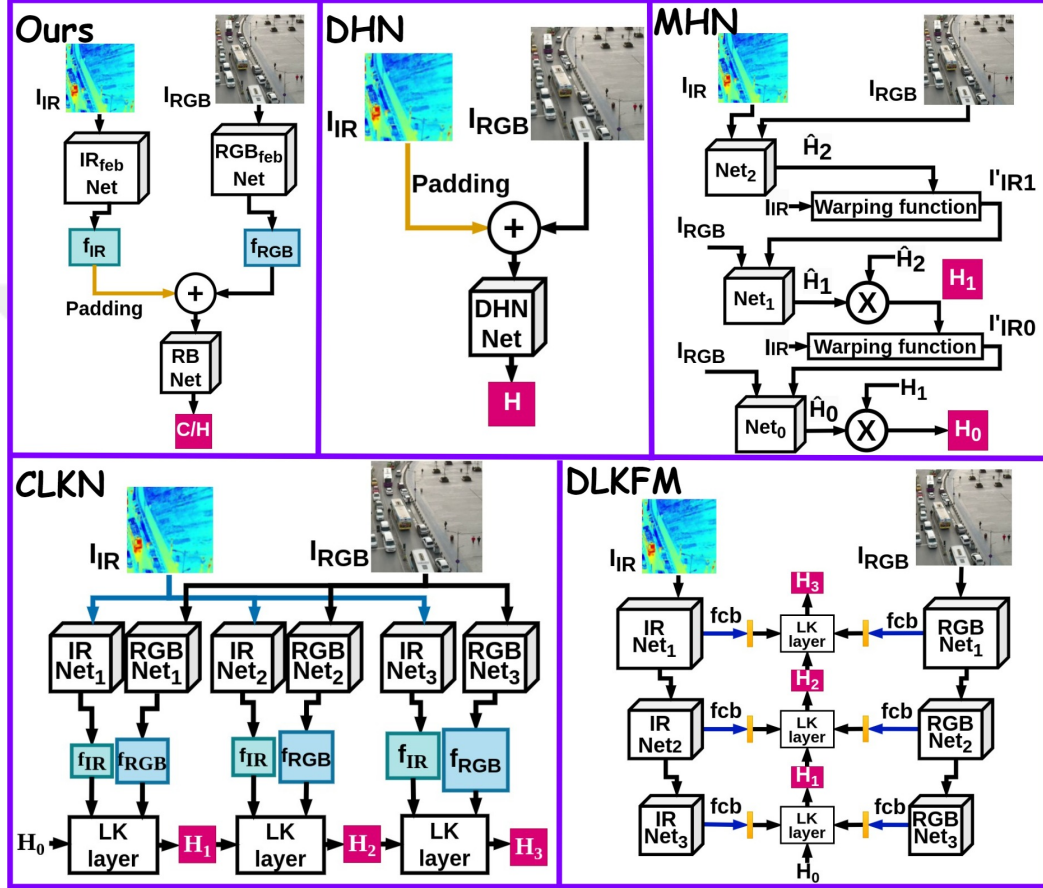
Many recent techniques performing image alignment rely on deep learning. Convolutional Neural Networks (Convolutional Neural Network (CNN)s) form a pipeline of convolutional layers where filters learn unique features at distinct levels of the network. For example, the authors of [1] proposed a Deep Image Homography Estimation Network (Deep Homography Network (DHN)) that uses CNNs to learn meaningful features in both images and it directly predicts the eight affine transformation parameters of the homography matrix. Later, the authors of [2] proposed using a series of networks to regress the homograph parameters in their approach. The latter networks in their proposed architecture aims to gradually improve the performance of the earlier networks. Their method builds on top of DHN [1]. Another work in [3], proposed incorporating the Lucas-Kanade (LK) algorithm in the deep learning pipeline.

The authors of [4] used a CNN based network and introduced a learning based Lucas-Kanade block. In their work, they designed modality specific pipelines for both source and template images, respectively. At the end of each block, there is a unique feature construction function. Instead of using direct output feature maps, they constructed

features based on Eigen values and Eigen vectors of the output feature maps. The features constructed from the source and template network channels have a similar learned representation. Transformation parameters found at a lower scale are given as input to the next level and the LK algorithm iterates until a certain threshold is reached. In another work, the authors of [35] utilized disentangled convolutional sparse coding to separate domain-specific and shared features of multi-modal images for improved accuracy of registration. Multi-scale Generative Adversarial Networks (Generative Adversarial Network (GAN)s) are also used to estimate homography parameters as in [36].

The architectural comparisons of the above-mentioned multiple networks is provided in Fig.2.1. In DHN [1], the image to be transformed (it is noted as I_{IR} in the figure) is padded to have the same dimensions as the target image (I_{RGB}) and they are concatenated channel-wise. The concatenated images are given to the Deep Homography Network (DHN) for the direct regression of the 8 values of the homography matrix. On the other hand, Multi-scale homography estimation (MHN) [2] adapts using a series of networks (Net_i). The inputs for Net_2 are a concatenation of I_{IR} and I_{RGB} . For the succeeding levels, first, the warping function performs the projective inverse warping operation on the infrared image (I_{IR}) via the homography matrix which was predicted at the previous level. The resulting image (I'_{IR_i}) is first concatenated with I_{RGB} and then given as input to the Net_i . For the following levels, the current matrix and previously predicted matrices are multiplied to form the final prediction. This way MHN aims to learn correcting mistakes made in the earlier levels. Cascaded Lucas-Kanade Network (Cascade Lucas-Kanade Network (CLKN)) [3] uses separate networks for each modality. They use levels of different scales in the form of feature pyramid networks, and perform registration from smallest to the largest. The homography matrix from the earlier LK-layer is given as input to the next. Deep Lucas-Kanade Feature Maps (DLKFM) [4] also performs coarse to fine registration as shown in Fig.2.1. It uses a special feature construction block

Figure 2.1: In this figure, we summarize the architectures of various recently proposed deep alignment algorithms including DHN [1], MHN [2], CLKN [3] and DLKFM [4]. While DHN and MHN predict the homography parameters $\mathbf{H}$; CLKN and DLKFM rely on using Lucas-Kanade (LK) based iterative approach and they use feature maps at different resolutions. By doing so, they predict homography in steps $\mathbf{H}_i$ where each step aims to correct the previous prediction.



called (**fc****b**). The (**fc****b**) block takes in the feature maps and transforms them into new feature based on the Eigen vectors and covariance matrix. The constructed features capture principal information and the registration is performed on the constructed feature maps, thus, it aims to increase the accuracy of the LK-layer. Our approach uses separate feature embedding blocks to process each modality separately. It is trained to extract modality specific features so that the output feature maps of different modalities can have similar feature representations.

2.2 Preliminaries

Perspective Transformation: Here, by perspective transformation, we mean a linear transformation in the homogenous coordinate system which warps the source image onto the target image. Homography matrix consists of the transformation parameters needed for the perspective transformation. The elements of the 3×3 dimensional homography matrix represent the amount of rotation, translation, scaling and skewing motions. Homography matrix $\mathbf{H}$ is defined as follows:

$$\mathbf{H} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix} \quad (2.1)$$

where the last element (p_9) is set to 1 to ensure the validity of conversion from homogeneous to the cartesian coordinates.

Warping function: maps a set of coordinates $[(x_i, y_i), \dots]$ to another coordinate system via $\mathbf{H}$. Let $c_i = (x_i, y_i)$ be the location of a point in the coordinates set C of the source image. Let $W(c, P)$ be the warping function that warps given coordinate c with parameter set P of $\mathbf{H}$ to the target image:

$$c'_i = W(c_i, P) \quad (2.2)$$

The warping process is a linear transformation in homogeneous coordinate system. Therefore, the Cartesian coordinates are first transformed into the homogeneous coordinate system by adding the extra z dimension to the 2D Cartesian pixel coordinates. Let c_i be the pixel with x_i, y_i coordinates. Homogeneous coordinate of c_i can be represented by setting z - axis to 1 i.e., $c_{h_i} = (x_i, y_i, 1)$. Once we have the homography matrix, we warp any given i^{th} pixel location c_i represented by (x_i, y_i) to its warped version c_i^{warped} on the other

image's Cartesian coordinate as follows;

$$c_{h_i}^{warped} = W(c_i, P) \iff \begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (2.3)$$

where x'_i, y'_i, z'_i , are warped homogeneous coordinates of c_i^{warped} which can be converted to Cartesian coordinates by simply division by the z'_i value. Therefore, we can obtain the final warped 2D pixel coordinates in Cartesian coordinates as follows: $c'_i = (x'_i, y'_i)$, where:

$$x_i^{warped} = \frac{x'_i}{z'_i} \iff \frac{p_1x_i + p_2y_i + p_3}{p_7x_i + p_8y_i + 1} \quad (2.4)$$

$$y_i^{warped} = \frac{y'_i}{z'_i} \iff \frac{p_4x_i + p_5y_i + p_6}{p_7x_i + p_8y_i + 1} \quad (2.5)$$

2.3 VisIRNet (Proposed Approach) Network Structure

In our proposed approach, we aim at performing accurate, single and multi-modal image registration which is free of the iterative nature of LK-based algorithms. We name our network VisIRNet, where we aim to predict the location of the corners of the input image on the target image directly, since having four matching points is sufficient to compute the homography parameters. In our proposed architecture, we assume that there are two input images with different resolutions. The overview of our architecture is given in Fig.2.2. Our approach first processes two inputs separately by passing them through their respective feature embedding blocks and extracts representative features. Those features are then combined and given to the regression block as input. The goal of the regression block is computing the transformation parameters accurately. The output of the regression block is eight dimensional (which can represent the total number of homography parameters or the coordinates of the four corner points of the source image on the target image).

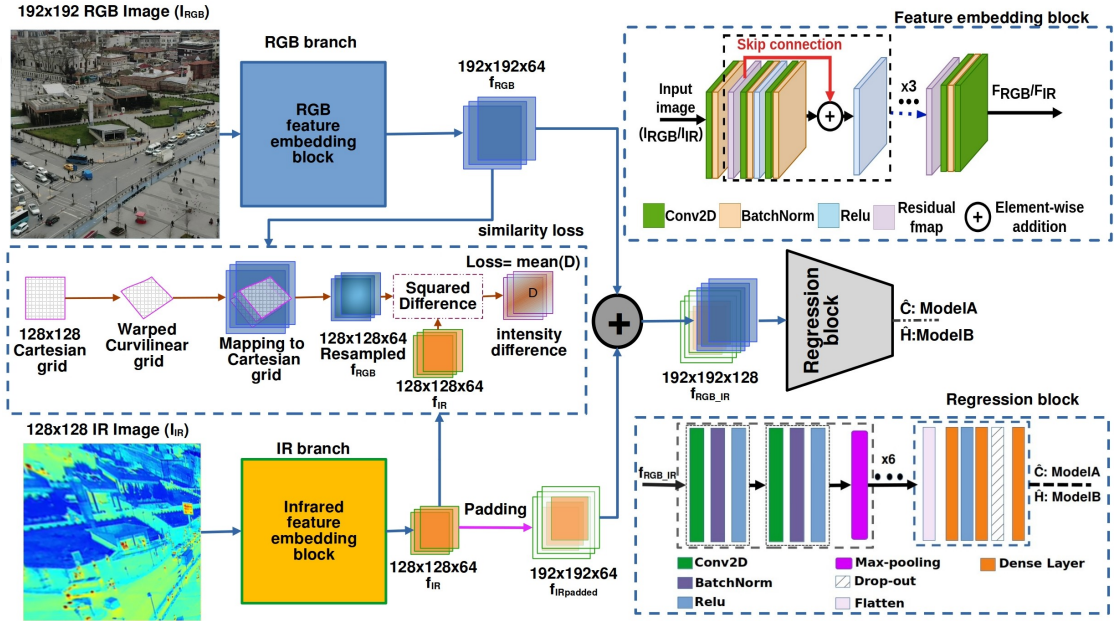
Table 2.1: *This table provides the layer-by-layer details of the feature embedding block. There are also skip connections between the layers in this architecture as shown in Figure 2.2.*

Layer	Filter Number	Filter-dims	Stride	Padding	Repetition
Conv2D	64	3x3	1	SAME	x1
BN	-	-	-	-	
Conv2D	64	3x3	1	SAME	x3
BN	-	-	-	-	
Relu	-	-	-	-	
Conv2D	64	3x3	1	SAME	
BN	-	-	-	-	
Relu	-	-	-	-	
Conv2D	64	3x3	1	SAME	x1
BN	-	-	-	-	
Conv2D	64	3x3	1	SAME	

Our proposed network is composed of multi-modal feature embedding blocks (MMFEB) and a regression block (see Fig.2.2). The regression block is responsible for predicting the 8 homography matrix parameters directly or indirectly. In this thesis, we study the performance of two variants of our proposed model and we call them ModelA and ModelB. ModelA predicts the coordinates of the corner points while its variant, ModelB, predicts the direct homography parameters. In ModelA, 4 corners are enough to find the homography matrix. Therefore the last layer has 8 neurons for the four (x, y) corner components for the ModelA, or for the eight unknown homography parameters for the modelB.

Multi-modal Feature Embedding Backbone: MMFEB is responsible for producing a combined representative feature set formed of fine level features for both of the input images. The network then, will use that combined representative feature set to transform the source image onto the target image. We adapt the idea of giving RGB and infrared modalities separate branches as in [4]. We use two identical networks (branches) with same structure but with different parameters for RGB and infrared images, respectively. Therefore, the multi-modal feature embedding block has two parallel branches

Figure 2.2: This figure provides an overview of our proposed network architecture. Two parallel branches including RGB branch and IR branch (feature embedding blocks) extract the salient features for RGB and IR images, respectively. Those features are, then channel-wise concatenated and fed into the regression block for direct (ModelB) or indirect (ModelA) homography prediction. I.e., the model can be trained for learning the homography matrix in ModelB or to regress the corresponding coordinates of the four corners of the input IR image on the RGB image in ModelA. The output is 8 dimensional vector (for $\mathbf{H}$), if ModelB is used; and it is 8 dimensional vector where those 8 values correspond to the (x, y) coordinates of the 4 corners of the IR image, if modelA is used. The details of the feature embedding block are given on the top corner of the figure (also see Table 2.1). The details of the regression block are given in the lower right corner of the figure (also see Table 2.2).



with identical architectures (however they do not share parameters), namely RGB-branch and infrared branch. We first train the multi-modal feature embedding backbone by using average similarity loss $\mathcal{L}_{sim}$ (see Eq. 2.6). To compute the similarity loss, we first generate a 128×128 recti-linear grid, representing locations in infrared coordinate system as in spatial transformers [37]. Then, we use the ground truth homography matrix to warp the grid onto the RGB coordinate system resulting in a warped curvilinear grid representing projected locations. We use bi-linear interpolation [38, 39] to sample those warped locations on the RGB feature maps (f_{RGB}). After that, we can compute the similarity loss

Table 2.2: This table provides the layer-by-layer details of the regression block as shown in Figure 2.2. Levels indicated by L are groups of $\text{conv2D} + \text{BatchNormalization} + \text{Relu}$. The Conv2D layers in each level have the same characteristics and filter dimensions. The number of used filters increases as we get deeper in the architecture.

Level	Number of filters / Units	Filter-dims	Stride	Padding	Activation
L1	32	3x3	1	SAME	
max-pool	-	2x2	2	SAME	
L2	64	3x3	1	SAME	
max-pool	-	2x2	2	SAME	
L3	64	3x3	1	SAME	
max-pool	-	2x2	2	SAME	
L4	128	3x3	1	SAME	
max-pool	-	2x2	2	SAME	
L5	128	3x3	1	SAME	
max-pool	-	2x2	2	SAME	
L6	256	3x3	1	SAME	
Flatten					
Dense	1024	-	-		Relu
Dense	1024	-	-		Linear
Dropout	20%	-	-		-
Dense	8	-	-		Linear

between IR feature maps and re-sampled RGB feature maps. Algorithm 1 provides the algorithmic details of calculating the similarity loss for the feature embedding block.

MMFEB is trained by using the $\mathcal{L}_{sim}$ (see Eq. 2.6) which is detailed in Algorithm 3. Steps for training the MMFEB are given in Algorithm 1. Regression block is trained with homography loss ($\mathcal{L}_2^H$) in combination with average corner error ($\mathcal{L}_{Ace}$) (see the subsection "Average Corner Error (Ace)" below for the definitions of $\mathcal{L}_{Ace}$), yielding the total loss $\mathcal{L}$ to train our model. Table 2.1 summarises the structure of our used MMFEB.

Regression block: The second main stage of our pipeline is the regression block which is responsible for making the final prediction. The prediction can be the four corner locations, if ModelA; or the unknown parameters of the homography matrix, if ModelB.

f_{RGB} and f_{IR} are the feature maps extracted by passing the RGB image and infrared image through their respective feature embedding blocks in the feature embedding block. Note that f_{RGB} and f_{IR} have different dimensions. Therefore, we apply zero-padding to the lower dimensional feature maps (f_{IR}) so that we can bring its dimensions to the dimensions of f_{RGB} , resulting $f_{IRpadded}$. We concatenate (channel-wise) $f_{IRpadded}$ to f_{RGB} feature maps coming from infrared and RGB feature embedding blocks and use that as input for the regression block.

The architecture for regression block is further divided in two sub-parts as shown in Fig. 2.2. The first part is composed of 6 levels. Apart from the last level, each level is composed of 2 sub-levels followed by a max-pooling layer. Sub-level is a convolution layer followed by a batch normalization layer followed by a relu activation function. sub-levels m and n of a level l are identical in terms of the filters used, kernel size, stride and padding used for level l . the 6th level does not have a max-pooling layer. Second part has two 1024-dense layers with relu as activation function followed by a dropout layer and 8-dense output layer for 8 parameters of homography matrix or corner components. Feature maps from the previous part are flattened and given to the second part where homography matrix parameters or corner components are predicted according to the model used. Table 2.2 gives detailed information for the first and the second parts of the regression head.

Algorithm 1 Training steps of the MMFEB

Inputs: I_{RGB}^*, I_{IR}^* ▷ * indicates whole training set
for $e \leftarrow 0$ to $epochs$ **do**
 for $batch \leftarrow 0$ to $datasetSize/batchsize$ **do**
 $I_{RGB} = I_{RGB}^*[batch]$
 $I_{IR} = I_{IR}^*[batch]$
 $H \leftarrow groundTruthHomography$
 $f_{IR} \leftarrow RGBbranch(I_{RGB})$
 $f_{RGB} \leftarrow IRbranch(I_{IR})$
 $simLoss \leftarrow \mathcal{L}_{sim}(f_{IR}, f_{RGB}, H)$
 $Backprop(simLoss)$ Using *AdamOptimizer*
 end for
end for

Algorithm 2 Training step of the regression block

```
let M be RegressionBlock
Inputs:  $I_{RGB}^*, I_{IR}^*$  ▷ * indicates whole training set
for  $e \leftarrow 0$  to  $epochs$  do
  for  $batch \leftarrow 0$  to  $datasetSize/batchsize$  do
     $I_{RGB} = I_{RGB}^*[batch]$ 
     $I_{IR} = I_{IR}^*[batch]$ 
     $H \leftarrow groundTruthHomography$ 
     $f_{IR} \leftarrow RGBbranch(I_{RGB})$ 
     $f_{RGB} \leftarrow IRbranch(I_{IR})$ 
  Ensure:  $f_{RGB}.shape = 192 \times 192 \times 64$ 
  Ensure:  $f_{IR}.shape = 128 \times 128 \times 64$ 
     $f_{IRpadded} = zeroPadd(f_{IR})$ 
     $f_{RGB\_IR} = concat(f_{RGB}, f_{IRpadded})$ 
     $\hat{H} = \mathbf{M}(f_{RGB\_IR})$ 
     $Loss = \mathcal{L}(\hat{H}, H)$ 
     $Backprop(Loss)$  using AdamOptimizer
  end for
end for
```

Algorithm 3 Computing the $\mathcal{L}_{sim}$ loss

```
 $f_{RGB} \leftarrow RGBbranch(I_{RGB})$ 
 $f_{IR} \leftarrow IRbranch(I_{IR})$ 
 $H \leftarrow groundTruthHomography$ 
 $grid_{n \times n} \leftarrow 2x2gridwithIrdimensions$ 
Ensure:  $warpedGrid = warpGrid(grid_{n \times n}, H^{-1})$ 
 $f'_{RGB} = BilinearSampler(f_{RGB}, warpedGrid)$ 
 $\mathcal{L}_{sim} \leftarrow 0$ 
for  $i = 0, i \leq n \times n$  do
   $Ir_i = f_{IR}[i]$ 
   $Rgb_i = f'_{RGB}[i]$ 
   $P_{diff} \leftarrow Ir_i - Rgb_i$ 
   $\mathcal{L}_{sim} \leftarrow \mathcal{L}_{sim} + P_{diff}^2$ 
end for
 $\mathcal{L}_{sim} \leftarrow \mathcal{L}_{sim} / (n \times n)$ 
```

2.4 Used Loss Functions

While MMFEB uses *similarity loss*, we used two loss terms based on the corner error and homography for the regression head.

Similarity loss: The similarity loss is used to train MMFEB and is defined as follows:

$$\mathcal{L}_{sim} = \frac{1}{x * y} \sum_{x=0}^n \sum_{y=0}^n (f'_{RGB}(x, y) - f_{IR}(x, y))^2 \quad (2.6)$$

where, $f_{IR/RGB}(x, y)$ is the value at (x, y) location for respective image feature maps. $f'_{RGB}(x, y)$ is the value at (x, y) location on the re-sampled RGB feature maps. Note that the (x, y) is a location on the coordinate system constrained by the infrared image height and width. The algorithmic details of the similarity loss are provided in Algorithm 3.

L_2 Homography loss term: ModelB is trained to predict the values of the elements of the homography matrix. Therefore, its output is the 8 elements of a 3×3 matrix (where the ninth element is set to 1). The homography based loss term: L_2^H is defined as follows: let $[p_i : (\text{for } i = 1, 2, 3, 4, 5, 6, 7, 8), 1]$ be the elements of a 3×3 $\mathbf{H}$ ground truth homography matrix. Similarly, let $[\hat{p}_i : (\text{for } i = 1, 2, 3, 4, 5, 6, 7, 8), 1]$ be elements of 3×3 $\hat{\mathbf{H}}$, the predicted homography matrix. Then, $\mathcal{L}_2^H = \frac{1}{8} \sum_{i=1}^8 (p_i - \hat{p}_i)^2$ where, $\mathcal{L}_2^H$ represents the homography loss based on the L_2 distance.

Average Corner Error (Ace): Ace is computed as the average sum of squared differences between the predicted and ground truth locations of the corner points. For ModelB, we use predicted homography matrix to transform the 4 corners of infrared image onto the coordinate system of RGB image and together with ground truth locations we compute $\mathcal{L}_{Ace}$. Let e_i be a corner at the (x_i, y_i) coordinates on the infrared image and let e'_i be its warped equivalent on the RGB coordinate space such that $e'_i = W(e_i, P)$ where W is the warping function.

$$\mathcal{L}_{Ace} = \frac{1}{4} \sum_{i=1}^4 D(e_i, e'_i)^2 = \frac{1}{4} \sum_{i=1}^4 (W(e_i, P) - W(e_i, \hat{P}))^2 \quad (2.7)$$

where D is defined as: $D(e_i, e'_i) = W(e_i, P) - W(e_i, \hat{P})$, and where P and $\hat{P}$ are ground truth and predicted vectorized homography matrices, respectively. The total loss for ModelB, then, is computed as $\mathcal{L} = \mathcal{L}_2^H + \gamma \mathcal{L}_{Ace}$ where γ is weight factor (a hyperparameter).

In ModelB, we predict the x and y locations of the 4 corner points, instead of computing the homography matrix. This makes it possible for the network to learn to predict exact locations (landmarks) instead of focusing on one solution. As shown in our experiments, (see Fig. 2.6 for qualitative and Fig. 2.5 for quantitative results), ModelA

Table 2.3: Ablation study on using different combinations of loss functions on two different datasets. The loss functions shown in each row are used for the MMFEB block, and the loss functions shown in each column ($\mathcal{L}_{Ace}$ and L_2^H) are used for the regression block in our model. Best results are shown in bold. Ace is the metric used to compute the results for each loss function combination. The last column shows the average Ace value for each loss function used in the MMFEB block. On average, $\mathcal{L}_{sim}$ yielded the best results.

	SkyData		VEDAI		Average
	$\mathcal{L}_{Ace}$	L_2^H	$\mathcal{L}_{Ace}$	L_2^H	
$\mathcal{L}_{sim}$	18.6	21.6	19.1	128.3	36.85
$\mathcal{L}_{MAE}$	18.5	35.8	18.5	178.1	62.7
$\mathcal{L}_{SSIM}$	18.5	20.1	19.1	134.2	47.9

converges faster and yields better results, while minimizing outliers. We use a slightly modified version of $\mathcal{L}_{Ace}$ for ModelA such that $\hat{e}_i$ becomes the ground truth corner coordinate in RGB coordinate space. For ModelA, $\mathcal{L}_{Ace}$ is defined as follows:

$$\mathcal{L}_{Ace} = \frac{1}{4} \sum_{i=1}^4 (e_i - \hat{e}_i)^2 \quad (2.8)$$

In addition to these loss functions, we also used additional loss functions in the MMFEB block during our ablation study. Those functions are $\mathcal{L}_{MAE}$ and $\mathcal{L}_{SSIM}$. They are briefly defined below.

$$\mathcal{L}_{MAE} = \frac{1}{x * y} \sum_{x=0}^n \sum_{y=0}^n |f'_{RGB}(x, y) - f_{IR}(x, y)| \quad (2.9)$$

$$\mathcal{L}_{SSIM} = 1 - SSIM(f'_{RGB}(x, y), f_{IR}(x, y)) \quad (2.10)$$

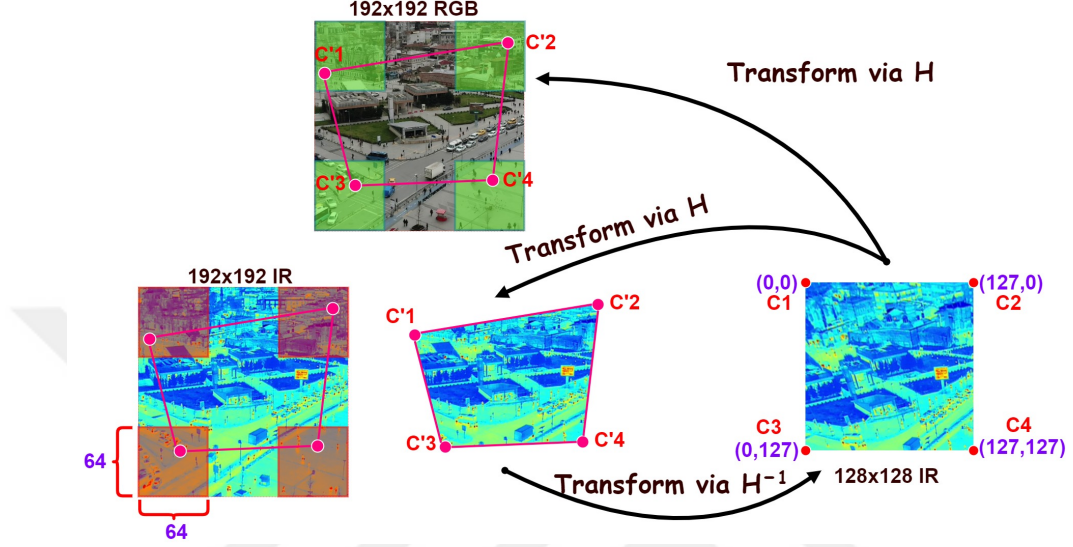
where SSIM is used as also used and defined in [5].

2.5 Used Datasets

In our experiments we use Skydata¹ containing RGB and IR image pairs, MSCOCO [40], Google-Maps, and Google-Earth (as taken from DLKFM [4]), VEDAI [41] datasets.

¹www.skydatachallenge.com

Figure 2.3: This figure shows how to select initial corner points on the registered image pairs and how to generate the training data. First a random image patch is taken from the originally registered IR image. Then the random corners of that patch is transformed into fixed coordinates and after that, the H matrix (and its inverse) performing that transformation is computed.



Please refer Table 2.4 for more details about the used datasets in our experiments. Sky-Data is originally a video-based dataset which provides each frame of the videos in image format.

Generating the training and test sets: To train the algorithms we need unregistered and registered (ground truth) image pairs. For SkyData, we randomly select m frame pairs for each video sequence.

Dataset	Modality	Training set	Test Set
SkyData	RGB + Infrared	27700	7990
MSCOCO [40]	Single modality (RGB)	82600	6400
Google Maps	RGB + map (vector)	8800	888
Google Earth	RGB + RGB	8750	850
VEDAI [41]	RGB + Infrared	8722	3738

Table 2.4: A summary of the used datasets in our experiments is given. The training and test data are generated as explained in Section 2.4.

For each dataset that we use, we generate the training and test sets as follows: (i)

Select a registered image pair at higher resolutions. (ii) Sample (crop) regions around the center of the image to get smaller patches of 192x192 pixels. This process is done in parallel for visible and infrared images. (iii) If the extracted patches are not sufficiently aligned, manually align them. (iv) For each pair, select a subset of the IR image, by randomly selecting 4 distinct locations on the image. (v) Find perspective transformation parameters that map those randomly chosen points to the following fixed locations: $(0,0)$, $(n-1,0)$, $(n-1,n-1)$, $(0,n-1)$ so that they can correspond to the corners of the unregistered IR image patch; where we assume that the unregistered I_{IR} is $n \times n$ dimensional (in our experiments n is set to 128). This process creates an unregistered infrared patch (from the already registered ground truth) that needs to be placed back to its true position. (vi) Use those 4 initially selected points as the ground truth corners for the registered image. (vii) Repeat process k times to create k different image pairs. This newly-created dataset is then split into training and test sets. (viii) The RGB images are used as the target set and the transformed infrared patches are used as the source set (for both training and testing). This process is done on randomly selected registered pairs for each dataset. Fig.2.3 also illustrates this process on a pair of RGB and IR images. The list of all the used data sets and their details are summarized in Table 2.4.

2.6 Evaluation Metrics

As shown in Table 2.5, we quantitatively evaluate the performance of our models using Ace and homography error. We compute each algorithm's result distribution in terms of quantiles, mean, standard-deviation and min-max values for a given test set. Quartiles are a set of descriptive statistics which summarize central tendency and variability of data [42]. Quartiles are a specific type of quantiles that divide the data into four equal parts. The three quartiles are denoted as Q1, Q2 (which is also known as the median), and Q3. The 25% (Q1), 50% (Q2) and 75% (Q3) percentiles indicate that k% of

the data falls below the k^{th} quartile (the bottom right illustration in Fig.2.4 also illustrates these terms). To find quartiles; we first sort elements in data being analysed in ascending order. The first quartile is the number of samples that fall below the dataset size*(1/4) element. Likewise the second quartile is the count of elements that fall below dataset size * (2/4) and the third quartile is dataset size * (3/4) th element in sorted dataset. The samples that fall out of ($Q1-1.5 * IQR$ and $Q3+1.5IQR$) where IQR is inter-quartile range, are considered outliers. The box plot as in Figure 2.4 illustrates the above mentioned description visually.

Table 2.3 shows an ablation study on using different loss functions in each block in our architecture. The used metric in the table is Ace and the best values are shown in bold. The loss functions in each row are used to train the MMFEB block includes $\mathcal{L}_{sim}$, $\mathcal{L}_{MAE}$ and $\mathcal{L}_{SSIM}$. The loss functions used for the regression block are $\mathcal{L}_{Ace}$ and L_2^H . In the table, the last column shows the average error for both $\mathcal{L}_{Ace}$ and L_2^H (over two datasets including SkyData and VEDAI) for each of the used loss functions in the MMFEB block.

Next, we provide experimental results on the effect of the hyperparameters that we studied for both ModelA and ModelB.

2.7 Quantitative Registration Results

Table 2.6 summarizes those results. In particular, we studied the effect of using different loss functions (L_1 , L_2 and L_{Ace}) and using different batch sizes for both models. All the experiments were done on the SkyData set. The best results are shown in bold. Overall, ModelA showed promising results achieving better results when compared to the ModelB. Therefore, for the rest of our experiments, we kept using ModelA only.

Fig.2.4 uses a box plot, also known as a box-and-whisker plot, to display the distribution of average corner error for different datasets and for different models. It

Table 2.5: Comparative results of algorithms on each dataset in terms of Ace. Best results are shown in bold. The results illustrate that on average traditional SIFT performed on average in datasets of single or close modalities. Learning based algorithms DHN and MHN directly predict homography matrix without learning common representation also suffer especially on datasets such as Skydata and Google-maps.

(a) SkyData Results

	mean	std	min	25%	50%	75%	max
MHN [2]	14.5	67.29	1.19	8.11	11.36	15.4	4296.21
DHN [1]	77.96	854.59	4.1	48.34	65.95	82.19	76119.41
DLKFM [4]	93.4	2894.7	7.32	32.08	40.73	51.95	258091.03
Ours	3.83	1.77	0.78	2.64	3.46	4.61	19.19
CLKN [3]	77.31	862.85	5.47	38.66	47.72	57.74	73661.8
SIFT [43]	43477.63	201670.37	2.13	232.88	1275.96	1285.6	100000.0

(b) VEDAI [41] Results

	mean	std	min	25%	50%	75%	max
DLKFM [4]	382.4	3363.72	10.93	101.43	120.09	187.6	189375.0
Ours	24.76	4.77	4.09	21.59	24.88	28.03	42.77
MHN [2]	374.53	5519.56	8.74	56.4	103.68	141.99	319559.19
SIFT [43]	40221.01	195588.26	0.11	2.07	88.96	1264.86	100000.0
DHN [1]	163.87	427.16	41.59	128.8	137.43	146.43	19138.92
CLKN [3]	99.49	709.1	2.45	37.97	52.29	75.13	30289.16

(c) GoogleEarth Results

	mean	std	min	25%	50%	75%	max
DHN [1]	1073.9	25214.41	8.85	52.1	67.17	83.78	733889.5
Ours	10.91	3.61	4.19	8.25	10.5	12.9	31.13
SIFT [43]	1334.41	34297.53	0.51	2.63	8.45	108.25	100000.0
DLKFM [4]	27.65	121.65	0.36	7.41	18.5	28.96	2733.34
MHN [2]	118.86	257.47	12.62	45.06	60.91	106.44	4552.5
CLKN [3]	14.58	36.57	0.38	3.15	6.51	14.51	730.46

(d) GoogleMaps Results

	mean	std	min	25%	50%	75%	max
MHN [2]	319.68	1003.02	9.48	40.68	73.54	208.8	12910.34
SIFT [43]	178912.9	382210.28	48.79	1273.04	1281.86	1292.6	100000.0
CLKN [3]	123.96	453.65	13.22	56.43	66.9	80.77	8496.15
DHN [1]	131.27	410.38	12.28	30.27	43.08	115.07	9866.02
Ours	9.57	4.15	2.6	6.82	8.68	11.28	33.67
DLKFM [4]	77.78	183.6	0.47	20.27	38.76	66.28	3251.74

(e) MSCOCO[40] Results

	mean	std	min	25%	50%	75%	max
DLKFM [4]	67.16	2515.37	0.06	0.44	8.31	31.12	200374.44
Ours	3.67	2.45	0.64	2.28	2.99	4.13	27.14
CLKN [3]	6.45	8.96	0.1	1.68	3.86	8.01	280.96
DHN [1]	622.38	7493.6	3.0	141.93	194.88	383.56	580642.19
SIFT [43]	3308.58	57236.2	0.07	0.37	0.65	1.38	100000.0
MHN [2]	15.5	7.31	1.84	10.17	14.39	19.32	90.29

Table 2.6: A comparison of the results of ModelA and ModelB at various hyperparameters including batchsizes and loss functions. The top table shows the Homography error in (a), while the bottom table shows the results as Average corner error in (b). Note that the results illustrating low homography error doesn't necessarily imply small registration error. we study the effect of batchsizes and loss functions. directly predicting homography matrix work but it does not minimize the registration error as predicting direct corners in our experiments. These results are obtained on Skydata dataset.

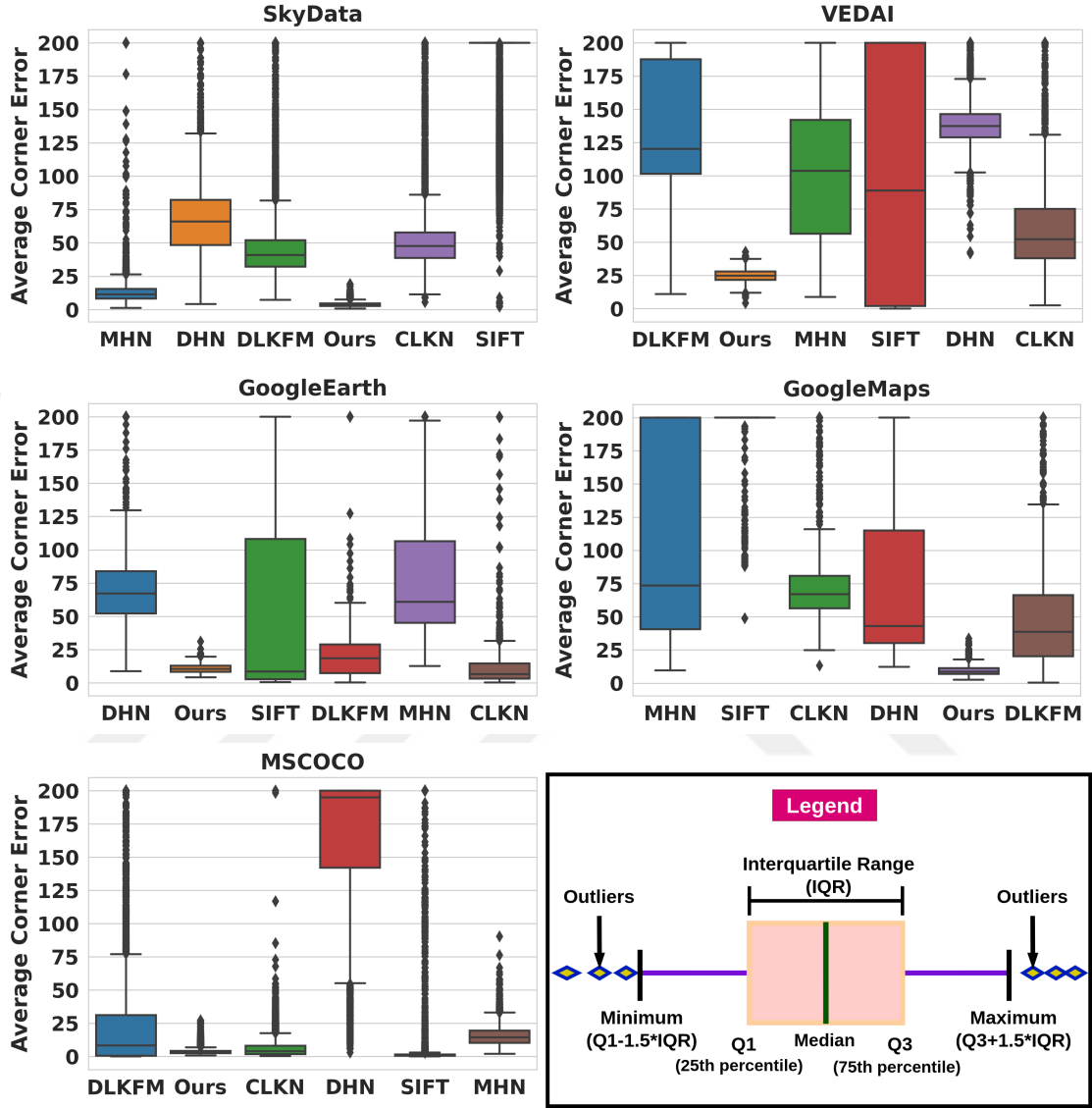
(a) Homography loss (L_2^H)

Model	BatchS	loss	mean	std	min	25%	50%	75%	max
ModelB	8	L1	0.6	0.44	0.03	0.32	0.49	0.76	8.38
ModelB	8	L2	1.84	4.49	0.0	0.37	0.89	2.03	219.33
ModelA	8	Ace	2.49	4.15	0.0	0.48	1.23	2.85	63.65
ModelB	16	L1	0.6	0.4	0.03	0.34	0.51	0.76	6.08
ModelB	16	L2	1.94	4.05	0.0	0.3	0.8	2.02	92.81
ModelA	16	Ace	2.14	4.1	0.0	0.4	1.04	2.42	176.51
ModelB	32	L1	0.7	0.46	0.02	0.4	0.61	0.9	8.17
ModelB	32	L2	2.79	5.3	0.0	0.5	1.31	3.08	149.63
ModelA	32	Ace	2.98	4.71	0.0	0.57	1.51	3.5	81.88
ModelB	64	L1	0.73	0.49	0.03	0.39	0.61	0.93	7.03
ModelB	64	L2	3.62	5.94	0.0	0.71	1.82	4.09	107.96
ModelA	64	Ace	3.65	5.17	0.0	0.91	2.22	4.57	157.6

(b) Average corner error

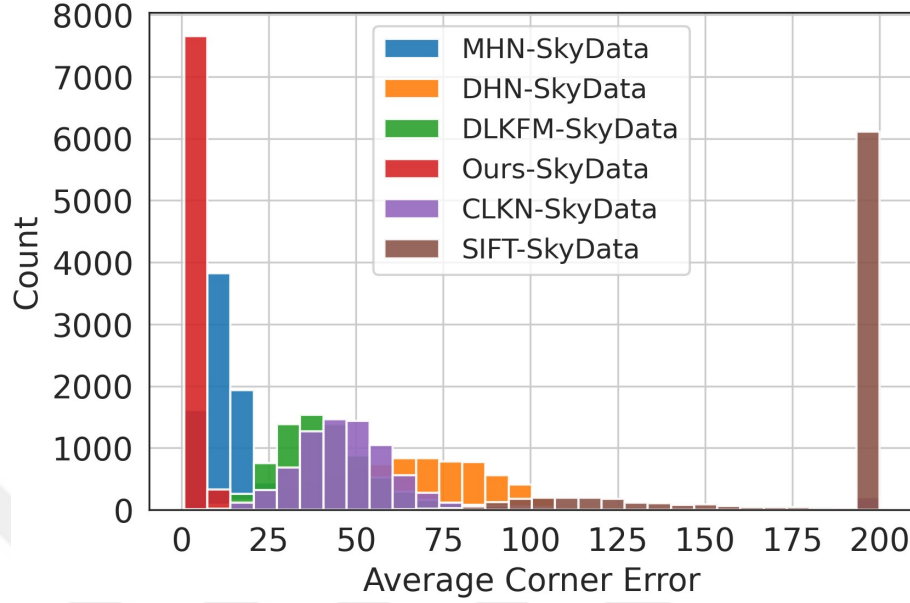
Model	BatchS	Loss	Mean	Std	Min	25%	50%	75%	Max
ModelB	8	L1	21.38	44.01	2.53	8.65	12.57	18.27	235.99
ModelB	8	L2	25.15	119.95	2.30	11.83	13.11	15.53	893.01
ModelA	8	Ace	3.96	1.64	0.93	2.84	3.68	4.72	19.59
ModelB	16	L1	25.83	108.53	1.66	8.02	10.97	16.27	743.54
ModelB	16	L2	31.27	143.28	2.29	10.52	13.65	18.47	1180.64
ModelA	16	Ace	3.83	1.77	0.78	2.64	3.46	4.61	19.19
ModelB	32	L1	25.17	101.33	1.87	6.08	8.80	16.97	803.25
ModelB	32	L2	6.50	4.11	2.05	5.43	6.40	7.35	15.07
ModelA	32	Ace	5.32	1.87	1.35	4.04	5.02	6.22	22.21
ModelB	64	L1	30.60	177.66	3.41	11.07	11.65	12.41	1298.48
ModelB	64	L2	31.86	135.19	1.74	8.10	13.46	19.23	981.56
ModelA	64	Ace	5.01	2.01	0.95	3.62	4.65	6.0	18.94

Figure 2.4: Each plot shows the Average corner error for the algorithms including MHN, DHN, DLKFM, CLKN, SIFT and ours for a different dataset. The legends used in the plots are also given in the lower right corner of the figure.



provides a summary of key statistical measures such as the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. The length of the box indicates the spread of the middle 50% of the data. The line inside the box represents the median (Q2). The whiskers extend from the box and represent the variability of the data beyond the quartiles, in our case they represent $Q1 - 1.5 * IQR$ and $Q3 + 1.5 * IQR$. Individual data

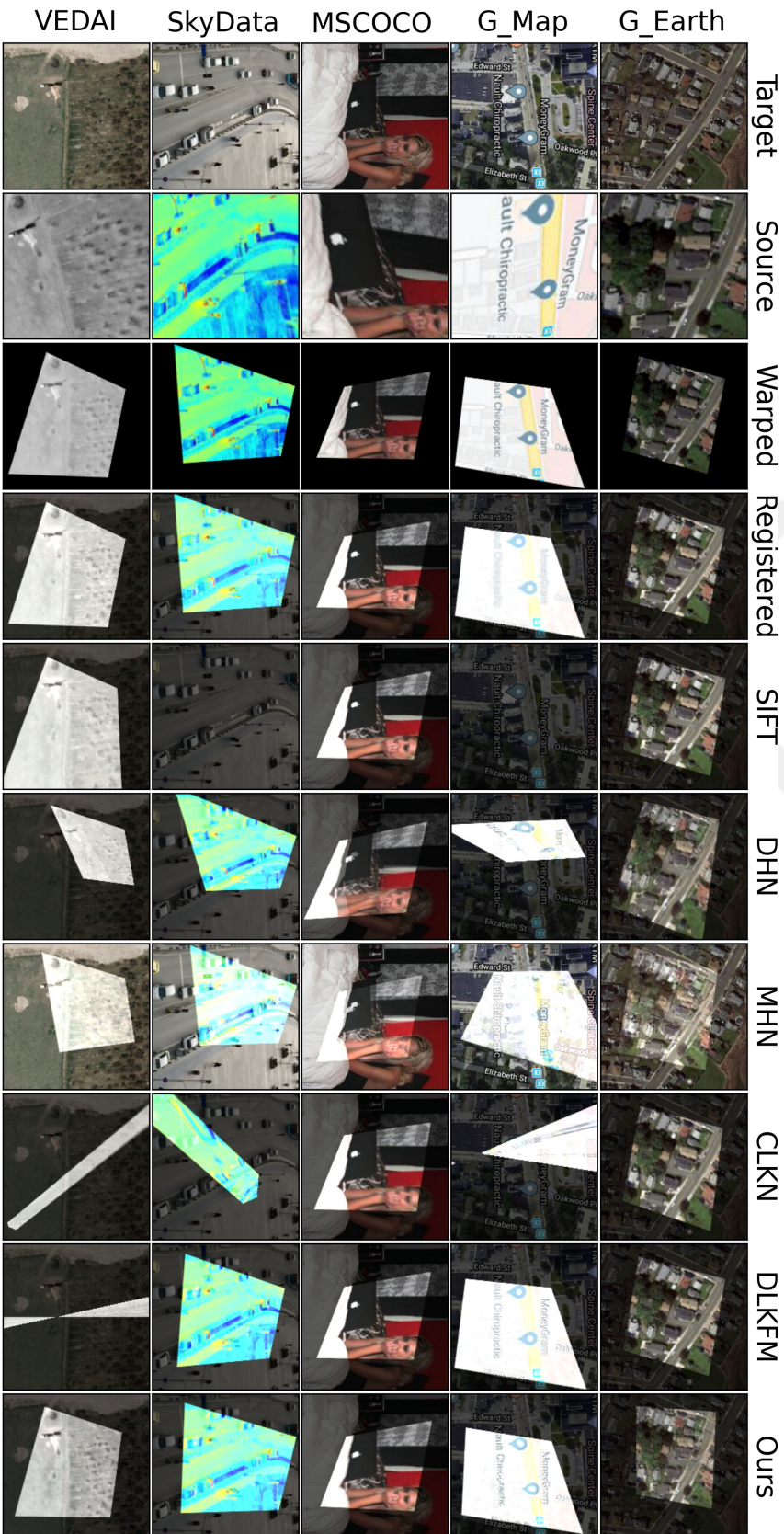
Figure 2.5: Average corner error distribution vs. count of image pairs for different models is shown on the test set of Skydata.



points that lie outside the whiskers are considered outliers and are plotted with diamonds. The figure compares the results for 6 algorithms on 5 different datasets.

Table 2.5 illustrates the results of using different approaches for each dataset, separately. The MSCOCO (e) results being a single modality dataset, SIFT perform relatively better but there are cases where the algorithm could not find homography due to insufficient pairs. Google earth in (c) also has RGB image pairs but from different seasons. SIFT algorithm is still able to pick enough salient features therefore the performance is still reasonable. (d) Google maps, (a) SkyData and (b) VEDAI have pairs of significant modality difference. Deep learning based approaches were able to perform registration often with high number of outliers. Our approach was able to perform registration on both single and multi-modal image pairs, specifically we were able to keep the max error minimum as opposed to LK-based approaches. In cases where enough pairs were not found SIFT is unable to estimate homography matrix. We assign a constant 10000.0 as

Figure 2.6: This figure shows qualitative results on sample image pairs taken from different datasets. The first two columns show the input image pairs for the algorithms. The target image is 192×192 pixels and the source image is 128×128 pixels (which covers a scene that is a subset of the target image). The third column shows the ground truth version (192×192 pixels) of the source image on the coordinate system of the target image after being warped. The fourth column shows the ground truth (warped) (192×192 pixels), after applying registration with the algorithms in the order of SIFT, DHN, MHN, CLKN, DLKFM and our approach, respectively. Visually, each algorithms' result can be compared to the image in the fourth column.



the error. Learning based algorithms DHN and MHN directly predict homography matrix without learning common representation also suffer especially on datasets such as Skydata and Google-maps. This is due to the models being unable to create meaningful correspondences for input and target images as a result of modality difference level. Note that our approach has a small standard deviation as opposed to LK based approaches. LK techniques often significantly deviate from the solution depending on number of iterations they are run and initial parameters they receive.

2.8 Qualitative Registration Results

Fig.2.5 shows the performance of 6 methods (SIFT, DHN, MHN, CLKN, DLKFM, Ours) on the SkyDataV1 dataset, in terms of average corner error. Skydata has RGB and infrared image pairs. In this figure, we aim to show that feature based registration techniques such as SIFT perform poorly whereas methods that leverage neural networks and learn representations are superior.

Fig.2.6 gives detailed qualitative results of our experiments. Each row represent a sample taken from a different dataset. The columns represent inputs and results for different approaches. Target is (192x192) (first column), and source (second column) (128x128) are input image pairs. Warped (third column), is the ground truth projection of source to the coordinate system of target image and Registered (fourth column) is the warped image overlayed on the target image as shown. Columns from 5 to 10 shows the registered and overlayed results for SIFT, DHN, MHN, CLKN, DLKFM, and Ours (ModelA) for the given input pair. While almost all algorithms relatively done well on Google Earth pair (which provides similar modalities for both target and source images), when the modalities are significantly different, as in the SkyData, Google Maps and VEDAI pairs, the figure shows that SIFT, CLKN, MHN, DHN and DLKFM algorithms can struggle for aligning them and they may not converge to any useful result near the ground truth (see

SIFT and CLKN results), while our approach converges to the ground truth by yielding small ACE error for each of those sample pairs.



3. PERFORMANCE COMPARISON OF RECENT ADVANCES IN QUERY BASED VIDEO INSTANCE SEGMENTATION

Video instance segmentation (VIS) is a fundamental and evolving area of computer vision research. VIS requires the simultaneous detection, segmentation, and temporal association of object instances across video frames [30, 44, 45, 46, 47, 48] as shown in figure 1.2. Regardless of the impressive and almost saturated performance in image-based instance segmentation models [28, 49, 50, 51, 52], VIS has proven to be a challenging task due to the dynamic nature of video data, where objects may go through rapid motion changes, occlusion, and appearance changes. The task was first formalized with the introduction of MaskTrack R-CNN[30], which extended the widely adopted Mask R-CNN [53] model to the video domain by introducing a tracking head that linked objects' instances across frames. MaskTrack R-CNN relied on separate segmentation and association modules which limited its ability to efficiently model long-term dependencies across frames.

3.1 A Summary of Video Instance Segmentation Methods

VIS is the task of detecting, segmenting, and tracking object instances across a sequence of video frames. In VIS, the goal is to generate pixel-precise instance masks for each object in every frame and to track the identity of those instances across frames. VIS methods can be divided into two major categories; online and offline video instance segmentation.

Online VIS methods, detect, segment instance objects in a frame-by-frame manner, and associating these instances across time. Online models are typically built on image-level segmentation models such as Mask R-CNN [53]. Unlike object detection, which only provides bounding boxes, VIS requires detailed masks for each object and

tracks this over time. Motivated by recent transformer-based image detection and segmentation models [54, 55, 56, 57], a number of query-based online methods have been proposed. These methods tackle the VIS task by taking advantage of consistent instance query embeddings and associating corresponding instance queries from frame to frame.

Most query-based methods presented [58, 59, 60, 61], extend on Masked-attention Mask Transformer (Mask2Former [62]) architecture to the video domain. Mask2Former consists of a backbone feature extractor[63, 56], a pixel decoder and a transformer decoder [64]. The backbone feature extractor is usually a pretrained resnet or swin-transformer to extract image features before further processing through the network. Pixel decoder build richer feature maps for mask prediction. FPN [65] used in MaskFormer [66] processes and combines feature maps from different scales, resulting in improved mask quality and allows capturing instances of different sizes. Following DETR [54], instance masks can be encoded with a C dimensional object queries. Transformer decoder then takes image features and processes object queries. Mask2Former uses masked attention in the transformer decoder which attends to local features instead of cross-attention used in standard transformer decoder leading to faster convergence and reduced computation load.

Online models have the inherent advantage of handling long video sequences and ongoing videos[67, 68, 69, 70, 70, 71, 30]. In their work entitled in defense of online models [72], Wu et al. demonstrated that the performance gap between online and offline methods is caused by error-prone association mechanisms between frames caused by similar appearance among different instances in the feature space. Thus, they proposed a contrastive learning-based online method that learns more discriminative instance embeddings for association and benefits from history information by introducing a memory bank for stability.

In another work, Heo et al. discussed that the discrepancy between training and inference is the main limitation in VIS methods. They proposed a generalized framework

[60] that introduced a memory and a label assignment mechanism that enhance the relationships between separate frames. MinVIS [61] highlighted the vital role of embeddings in instances of association between multiple frames. Using Mask2Former[50], they suggested that a strong query-based image instance segmentation model inherently produces distinctive enough instance queries and demonstrated competitive performance without video-based training or introducing additional contrastive loss functions. In DVIS, Zhang et al.[59] discuss online vis methods underperform due to the under utilization of temporal information. They propose a strategy that divides VIS into independent subtasks; segmentation, tracking, and refinement. In their subsequent work DVIS-DAQ[73], they discussed addressing disappearing and reappearing instances by introducing dynamic anchor queries. Ying et al.[58] studied a method of aligning training and inference in terms of choosing contrastive items for contrastive loss by performing momentum embedding averaging [74] and using the memory bank and adding noise to the embeddings when updating the memory bank to simulate tracking failure scenarios during inference time [58]. This extension leads to reliable comparison between instance embeddings and stable representation of historical embeddings that allow better modeling of challenges such as occlusion, deformations, and re-identifications.

Video Instance Segmentation has been advanced by offline models [32, 75, 76, 77, 78, 31]. Offline VIS methods segment and classify all video frames simultaneously and make instance associations in a single step. Often, as long videos would not fit in memory for processing, many methods split the video into several segments and perform VIS at clip level [78, 60]. Semi-Online methods usually require a heuristic mechanism to connect instances between adjacent clips.

The video instance segmentation transformer (VisTR) [32] extended the DETR[54] architecture to the video domain, using queries to directly predict object instances and

their associations between frames in an end-to-end fashion. VisTR [32] set a new benchmark by showing the effectiveness of transformer-based approaches in capturing spatio-temporal dependencies and extending it to segmentation and association (VIS). Motivated mainly by the computational requirements posed by VisTR, numerous follow-up studies extended VisTR keeping the paradigm of taking video as input and outputting masks with their trajectories in an end-to-end fashion. IFC[78] proposed an inter-frame communication transformer that uses memory tokens to convey information between frames, significantly reducing memory and computational requirements. The features for each frame are enriched and correlated with other frames, thus exchanging information. SeqFormer [79] uses a stand-alone instance query to capture the temporal sequence of instances across video frames. Each frame is processed independently, and mask sequences are dynamically predicted without tracking branches. TeViT [80] introduced the messenger mechanism for the early temporal and spatiotemporal query interaction to align video instances and queries efficiently modeling temporal information at the frame level and at the instance level. In Stem-seg [76], Athar et al. modeled spatio-temporal embeddings, for instance, segmentation in whole videos using 3D CNN architecture. Based on the hypothesis that object-oriented information constrains clues for understanding the entire sequence, VITA[81] uses object tokens instead of dense spatio-temporal features. VITA uses builds on transformer-based image segmentation model and object tokens are distilled from object-specific context.

3.2 Transformer Based Architectures in VIS

VIS is closely related to the Multi-Object Tracking and Segmentation (MOTS) challenge [82], where the goal is also to track and segment multiple objects across video frames. Both tasks share the requirement of maintaining object identities over time while accurately segmenting each instance. VIS typically involves a broader range of object

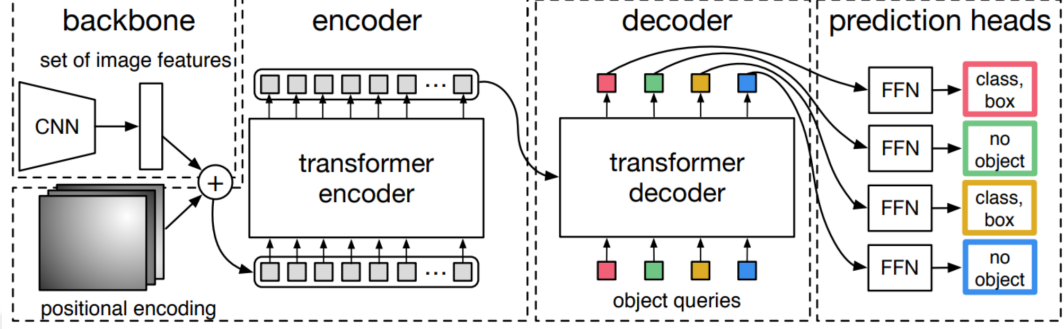
categories and often operates in less constrained environments compared to MOTS. While MOTS is designed considering specific applications such as autonomous driving, VIS covers more diverse scenarios such as dealing with varying object scales and complex interactions, and appearances.

As VIS continued to evolve, the focus shifted toward more integrated approaches that could simultaneously address both segmentation and association. Effective association is highly dependent on the strong representation of instances. This led to the exploration of transformer architectures, which have revolutionized various computer vision tasks because of their ability to model relationships through self-attention mechanisms. The seminal work "Attention Is All You Need"[64] introduced the transformer model in the context of natural language processing, but its core principles were soon adopted for vision tasks, inspiring new directions in VIS.

DETR [83] was an explicit advancement that used transformers for object detection in images with queries and thus eliminated the need for region proposals and post-processing steps like non-maximum suppression. Building on this, VIS models extended the transformer architecture to the video domain, using queries to directly predict object instances and their associations across frames. VisTR [32] set a new benchmark by showing the effectiveness of transformer-based approaches in capturing spatio-temporal dependencies and extending it to segmentation and association (VIS).

Most state-of-the-art transformer-based VIS methods, including VisTR[32], are offline. Full video sequences (or clips) are used to optimize instance association. While these approaches dominated online counterparts, they are not practical for real-time applications use-cases due to computational and memory constraints. To train such method massive GPUs and extended time are needed as videos are processed in full. Online methods, on the other hand, operate on a frame-by-frame basis, which is practical and significantly reduces the training time but they lagged behind in terms of accuracy. The need for

Figure 3.1: Query-based detection and instance segmentation architectures follow method proposed in DETR. Object queries are sufficient for a successful detection and segmentation. From object queries a class, mask and operational box heads are used to obtain instance masks and corresponding categories. Following the paradigm of detection, transform the fixed.



efficient online VIS models drove further research into exploring the reason behind and over time optimized query-based methods for faster, more lightweight implementations have been proposed.

In this work, we focus on the performance of query-based instance association in online VIS frameworks. By analyzing how query parameterization influences segmentation accuracy and consistent association across frames. Our experiments were carried out on YouTube-VIS 2019 and OVIS datasets.

Transformer architecture has been adopted for image recognition tasks (detection [54, 84, 85], classification[86, 87, 88, 89, 90, 91], segmentation [92, 93, 94], video instance segmentation [32, 95]). Vision Transformers (ViT) are generalized transformer architecture that, unlike traditional CNNs, which rely on convolutional layers to process images, use the transformer architecture [64] that was originally developed for and commonly used in natural language processing.

The transformer architecture, introduced by Vaswani et al. [64] in the paper "Attention Is All You Need," has revolutionized the fields of NLP and computer vision through the use of a self-attention mechanism, which enables the model to weigh the importance of each token relative to others, allowing it to capture dependencies across

different parts of the inputs depending on the context. Transformer models could consist of encoder and decoder structures or both depending on the task [96, 97, 98]. a) Encoder-only models are primarily used to understand and extract contextual information from input sequences [99]. b) Decoder-only models are typically used for generative tasks where the model needs to produce a sequence [100]. c) encoder-decoder models for tasks that require transforming one sequence into another, such as machine translation, summarization, and sequence-to-sequence [101]. The encoder, a stack of layers with multihead self-attention mechanisms and feedforward neural networks as main components, is responsible for processing the input sequence. The decoder could additionally consist of a masked multi-head self-attention mechanism which prevents positions from attending to subsequent positions and is used in generating the output sequence.

Transformers extended for vision tasks can be used in complex image analysis tasks video instance segmentation, which often contain complex scenes with multiple areas of interest by capturing both spatial and temporal long-range dependencies within frames. For VIS typically, a video sequence is usually passed through pre-trained backbones [63, 56] to extract features which are then passed to the transformer, for instance embeddings. These embeddings must have descriptive instance features across frames for a successful association of instances.

3.3 Video Instance Segmentation Preliminaries

VIS is the task of detecting, segmenting, and tracking object instances in a sequence of video frames. In VIS, the goal is to generate pixel-precise instance masks for each object in every frame and to track the identity of those instances across frames. The length of videos and even different challenging factors such as occlusions, deformations, and rapid motion changes must be addressed properly by a successful VIS model. Unlike object detection, which only provides bounding boxes, VIS provides detailed masks for

each object.

$$\mathbf{X} = \{\mathbf{X}_t\}_{t=1}^T \quad \mathbf{X}_t \in \mathbb{R}^{W \times H \times C} \quad (3.1)$$

Given a video with a set of T frames of $\mathbb{R}^{W \times H \times C}$ where W is the width, H is the height, and C is the number of channels (e.g., $C = 3$), and $\mathbf{X}_t$ represents the frame at time t , the goal is to identify, segment and track the instances between frames.

For each frame at timestep t , the model outputs a binary mask for each instance j in the frame t denoted by:

$$\mathbf{M}_{t,j} \in \{0, 1\}^{W \times H}$$

Where each pixel value in the mask $\mathbf{M}_{t,j}(x, y)$ indicates whether the pixel (x, y) belongs to the j -th instance or not.

Similarly, the model outputs a label $l_j \in \{1, \dots, K\}$, where K is the number of instance categories present in the dataset (i.e., 40 and 25 for YoutubeVIS and OVIS respectively). Additionally, a tracking id τ_j is assigned to each instance and further needs to be associated through the subsequent frames to ensure temporal consistency. Thus, the complete output $\mathbf{Y}_t$ of the model for frame t is a set of instance masks, corresponding instance classes, and their tracking ids:

$$\mathbf{Y}_t = \{(\mathbf{M}_{t,j}, l_j, \tau_j)\}_{j=1}^{N_t} \quad (3.2)$$

Where N_t is the number of instances in frame t .

The objective during training is then to minimize a loss function between the ground truth and predicted outputs for each frame. Let the ground-truth for frame t be denoted as:

$$\mathbf{Y}_t^{GT} = \{(\mathbf{M}_{t,j}^{GT}, l_j^{GT}, \tau_j^{GT})\}_{j=1}^{N_t^{GT}} \quad (3.3)$$

The loss function is typically composed of mask loss, classification loss, and tracking (association) loss.

Mask loss L_{mask} : Measures pixel-wise mismatches between the predicted mask $\mathbf{M}_{t,j}$ and the ground-truth mask $\mathbf{M}_{t,j}^{GT}$. In traditional bounding-box-based detection tasks, it would be computed based on variants of intersection over union (IOU) between the ground-truth and prediction boxes. For instance segmentation, the finer details are captured by evaluating pixel-wise discrepancy. Such can be done using functions such as binary cross-entropy loss:

$$L_{\text{mask}} = \sum_{t=1}^T \sum_{j=1}^{N_t} \text{BCE}(\mathbf{M}_{t,j}, \mathbf{M}_{t,j}^{GT}) \quad (3.4)$$

Classification loss L_{class} : Well-studied functions that compare the predicted object category l_j with the ground-truth label l_j^{GT} , typically using a cross-entropy loss:

$$L_{\text{class}} = \sum_{t=1}^T \sum_{j=1}^{N_t} \text{CE}(l_j, l_j^{GT}) \quad (3.5)$$

Association loss L_{assoc} : The term to ensure correct associations between instances across frames. Ensures that the predicted association id τ_j matches the ground truth ID τ_j^{GT} , enforcing temporal consistency. In many query-based videos instance segmentation pipelines, the matching is done based on the learned instance embeddings. To have discriminative embeddings for dissimilar instances, other loss terms, i.e. contrastive loss functions, are utilized. We will discuss contrastive learning later for now, a generalized association loss between frames can be formulated as follows:

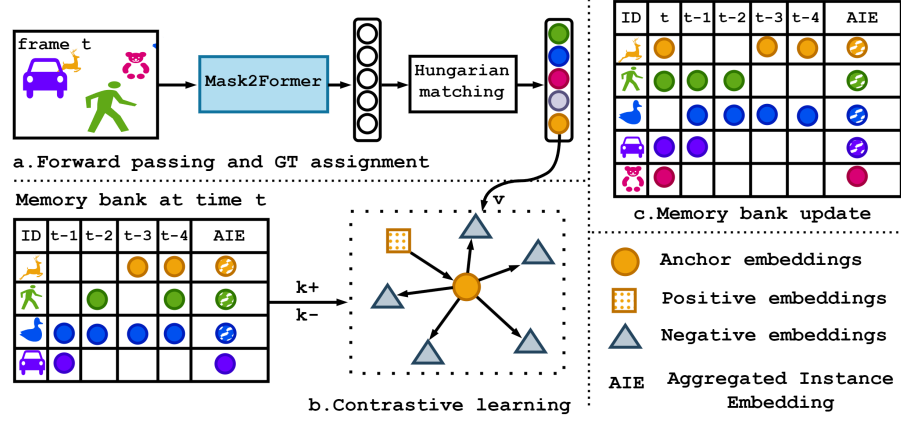
$$L_{\text{assoc}} = \sum_{t=1}^T \sum_{j=1}^{N_t} \text{ID_Loss}(\tau_j, \tau_j^{GT}) \quad (3.6)$$

The total loss L_{total} for the video instance segmentation model then becomes the weighted sum of the above losses:

$$L_{\text{total}} = \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{class}} L_{\text{class}} + \lambda_{\text{assoc}} L_{\text{assoc}} \quad (3.7)$$

where λ_{mask} , λ_{class} , λ_{assoc} are hyperparameters that control the contribution of each term.

Figure 3.2: Instance Association and Contrastive Learning in Query based VIS architectures.



3.4 Instance Association and Contrastive Learning

In query-based video instance segmentation methods, the discrimination of instance embeddings is crucial for accurate association of instances across video frames. The task of associating the same object in consecutive frames relies heavily on being able to distinguish between instances, even in the presence of challenges such as occlusions, variations in appearance, and background noise. Contrastive learning has shown success in discriminative representation learning [102, 74, 103]. As such, the same technique can be employed to enhance the discriminative instance embeddings.

For a video sequence of frames $\mathbf{X} = \{\mathbf{X}_t\}_{t=1}^T$, the model outputs instance embeddings $\mathbf{E}_t = \{\mathbf{e}_{t,i}\}_{i=1}^{N_t}$, where N_t is the number of instances in frame t , and $\mathbf{e}_{t,i} \in \mathbb{R}^d$ represents the d -dimensional embedding of the i -th instance in frame t .

Let $(\mathbf{e}_{t,i}, \mathbf{e}_{t',i})$ be the embeddings of the same instance i in frames t and t' . This is a positive pair and should have similar embeddings. Let $(\mathbf{e}_{t,i}, \mathbf{e}_{t',j})$ be the embeddings of different instances i and j . Negative pairs should have dissimilar embeddings.

The contrastive loss is applied on positive and negative pairs of embeddings. In the context of VIS an anchor object is chosen and an embedding is considered positive if the same instance is present in the next frame, and the rest is considered negative embeddings

[58]. In recent studies such as IDOL[72], positive and negative examples are dynamically chosen from the reference frame through an optimal transport mechanism [104]. Finding sets sourced from reference frame insufficient, Ying et al. [58] extended by leveraging a memory bank to capture history and adding noise when updating the memory. Overall contrastive loss can be formulated as follows:

$$L_{\text{contrast}}(\mathbf{e}_{t,i}) = -\log \frac{\exp(\text{sim}(\mathbf{e}_{t,i}, \mathbf{e}_{t',i})/\tau)}{\sum_{j=1}^{N_t} \exp(\text{sim}(\mathbf{e}_{t,i}, \mathbf{e}_{t',j})/\tau)} \quad (3.8)$$

Where $(\mathbf{e}_{t,i}, \mathbf{e}_{t',i})$ form a positive pair (same instance in different frames), and $(\mathbf{e}_{t,i}, \mathbf{e}_{t',j})$ with $j \neq i$ form negative pairs (different instances). Here, $\text{sim}(\mathbf{e}_{t,i}, \mathbf{e}_{t',j})$ is the similarity measure between two instance embeddings and can be measure using cosine similarity measure.

$$\text{sim}(\mathbf{e}_{t,i}, \mathbf{e}_{t',j}) = \frac{\mathbf{e}_{t,i} \cdot \mathbf{e}_{t',j}}{\|\mathbf{e}_{t,i}\| \|\mathbf{e}_{t',j}\|}$$

The task is to learn instance embeddings such that embeddings for the same instance across frames are close in the embedding space, while embeddings for different instances (or negative pairs) are far apart.

The training process involves contrasting the embeddings of instance queries using a contrastive loss function. Positive pairs (the same instance in different frames) are encouraged to be close, while negative pairs (different instances or incorrect associations) are pushed farther apart. The model is trained to minimize the distance between similar (positive) instance representations in the feature space while pushing apart dissimilar (negative) instances. In the context of online video instance segmentation, this is typically applied to learn instance representations, which are used to associate objects across frames. While instance queries of same object across frames are considered positive pairs are pushed to have similar representations the rest is taken as negative examples and pushed to be dissimilar in the embedding space even though they may belong to the same classes.

Some video instance segmentation methods heavily rely on cues such as semantic consistency, spatial correlation, detection confidence for instance associations between frames. Other methods rely on predicted masks and metrics, such as IOU, for associations. This is intuitive, however, the use of masks and bounding box is error-prone in long videos and in challenging scenes with lots of disappearance and reappearances. In query-based VIS using object queries for instance associations is preferred as the learned instance embeddings encodes more discriminative feature. IDOL [72] uses a temporally weighted SoftMax score and a memory bank-based association strategy. The object queries are temporally weighted before computing the similarity measure with the predicted instances. Instance assignment is performed on the basis of the similarity score.

3.5 Used Baseline Models

State-of-the-art VIS models are selected for experimentation and evaluation. With considerations of both computational resources and time, transformer-based online architectures are selected. The criterion for selecting models was the availability of codebases, relevancy, recency, popularity, support for datasets of interest, and reproducibility of the reported metrics. The selected models were all implemented using PyTorch and are trained or fine-tuned on both VIS datasets. DVIS [59] 2023 proposed a decoupled VIS framework that incorporates memory for history. In a more recent DVIS-DAQ [73] 2024 follow-up work, they introduced dynamic anchor queries (DAQ) to shorten the transition gap between anchor and target queries. we use DVIS as it fitted all our criterion for model selection. IDOL [72] 2022 proposes using an additional contrastive head to enforce distinctive query embeddings. IDOL uses optimal transport to select positive and negative examples that need contrastive loss. MinVis [61] 2022 offers state-of-the-art performance without video-based architectures by only using object query embeddings. GenVis [60] 2022, introduces a novel target label assignment and memory in instance associations

in an attempt to improve performance in long videos and challenging scenes. Finally, CTVIS [58] 2023, proposes a new method of selecting positive and negative items for the contrastive loss.

3.6 Experimental Setup

The objective of this use case is to analyze the performance of online methods video instance segmentation (VIS) frameworks as an effect of different object query parameterizations. The contribution of each method can be better understood by evaluating how different query parameterization strategies impact segmentation accuracy, temporal consistency, and computational efficiency in online VIS models. In general, the following steps are taken; 1) model selection and implementation, 2) dataset preparation, 3) experimentations and analysis, 4) performance evaluation. After selecting state-of-the-art online vis models we experiment on query parameterization and report the findings.

The impact of the association of the query-based instance is studied by systematically varying the query parameterization in the selected models. Specifically, what effect will changing the total number of queries processed per frame have on the performance of VIS models. Other possible parameters could be the dimensionality of the queries used in the transformer model and different strategies for initializing queries, such as random initialization versus learned embeddings.

Strategies for associating instance queries across the frames are kept the same as the model’s reported native method allowing us to better weigh the contribution introduced in each work. Models are forced to have similar configurations in terms of dataset splits, queries, detections per frame, etc. as much as possible. Models are trained and tested on the same datasets, with particular attention given to how changes in query parameterization affect the ability to accurately segment and associate instances.

All experiments were carried out on an SLURM cluster with multiple nodes, each

equipped with different CPU, memory, and GPU configurations. Overall the cluster has the following configurations: 2 x (20 CPUs, 122GB memory, 2 A6000 GPUs), 2 x (24 CPUs, 24GB memory, 2 RTX 3090 GPUs). 2 x (16 CPUs, 60GB memory, and 1 RTX 3090 GPU), 1 x (16 CPUs, 45GB memory, 1 P6000 GPU). The GPU requirements for each model determined the distribution of training between nodes. The duration of training varied according to the model and configuration, ranging from 4 to 18 hours.

3.7 Used Video Instance Segmentation Datasets

The experiments are conducted on two benchmark datasets that represent a range of challenging scenarios for VIS. YouTube-VIS 2019 [30] includes videos with 40 object categories and overall has relatively less challenging scenes and shorter sequences. Youtube-VIS 2019 was the pioneering challenge for video instance segmentation. Youtube-VIS has videos with 40 object categories. It consists of 2238, 302 and 343 training, validation, and test high-resolution video clips.

Occluded Video Instance Segmentation (OVIS) [105] came to be in 2022 and focuses on scenarios with heavy occlusions, making it ideal for testing the models' ability to maintain consistent instance associations despite partial or complete occlusions. OVIS [105] has 25 object categories and is more challenging in comparison to the Youtube-VIS dataset. OVIS has 607, 140, 154 training, validation, and test videos, respectively. It has much longer videos and contains scenes with more occlusions disappearance and reappearance rates. Both datasets are pre-processed to ensure consistent input formats, making the comparisons justified as models will see the same data.

Annotations: All annotations adopt the YouTube-VIS/COCO-style JSON schema, organized under six top-level keys: 'info', 'licenses', 'categories', 'videos', 'images', and 'annotations'. The 'info' section records metadata such as version, year, and creation date, while 'licenses' enumerates usage terms. Under 'categories', the semantic classes

are defined by 'id', 'name', and 'supercategory'. The 'videos' array lists each video's 'id', 'name', 'fps', 'width', and 'height', and 'images' provides frame records with 'id', 'file_name', 'height', 'width', 'video_id', and 'frame_id'. In the 'annotations' section, each object instance on an annotated frame includes mask 'segmentation' (polygon or RLE), '[x, y, width, height]' 'bbox', pixel 'area', and binary 'iscrowd' flag. Additionally, 'category_id' links to the semantic class, while 'track_id' and unique 'instance_id' maintain consistent identities across frames.

3.8 Used VIS Evaluation Metrics

Both Youtube-VIS 2019 and OVIS datasets have open challenge evaluation servers that we take advantage of to test the performance of our selected models on respective validation sets. The annotations for development sets are unavailable to the public, making the evaluation transparent for all models. Each model configuration is evaluated using a VIS metrics; Average Precision (AP) and Average Recall (AR) aggregated over different IOU thresholds. AP measures how well a model can detect and segment instances across various Intersection over Union (IoU) thresholds. In the context of VIS, AP is extended to video sequences, where the goal is to correctly identify and segment instances over time. AP is computed as the area under precision-recall curve at a specific IOU threshold. Precision measures the ratio of true positives (TP) over all positive detections (true and false positives). Recall is the measure of how correctly the model can detect and segment instances at various IOU thresholds. Recall is the ratio of true positives to both true positives and missed detections (FN). In the context of VIS all averaged across the multiple frames. The results are compared across different model configurations to identify the key factors that increase or decrease the performance of query-based instance association in both offline and online VIS settings. AP is averaged across multiple IoU thresholds (e.g., [0.50, 0.55, 0.60, ..., 0.95]), and is usually referred to as mAP (mean AP). We report

Table 3.1: This table presents a comparison of different video instance segmentation models based on their performance as the number of queries per image is varied. The evaluation is conducted on both the YouTube-VIS 2019 and OVIS validation sets. Results are grouped by query count for each method, providing insights into how the number of queries affects model accuracy and efficiency across different datasets.

Qc	Method	Backbone	YouTube-VIS 2019					OVIS				
			AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀	AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀
Q ₃₂	CTVIS	R50	0.486	0.704	0.518	0.480	0.586	0.242	0.425	0.240	0.140	0.328
	DVIS	R50	0.493	0.736	0.530	0.467	0.574	0.117	0.257	0.086	0.070	0.147
	GenVis	R50	0.341	0.505	0.370	0.391	0.457	0.145	0.304	0.134	0.097	0.209
	MinVis	R50	0.491	0.726	0.539	0.464	0.557	0.234	0.437	0.222	0.125	0.281
	IDOL	R50	0.452	0.683	0.483	0.455	0.561	-	-	-	-	-
Q ₆₄	CTVIS	R50	0.481	0.715	0.505	0.480	0.589	0.193	0.371	0.184	0.114	0.284
	DVIS	R50	0.493	0.726	0.527	0.470	0.570	0.101	0.229	0.080	0.065	0.132
	GenVis	R50	0.355	0.521	0.384	0.402	0.481	0.161	0.325	0.142	0.096	0.220
	MinVis	R50	0.483	0.710	0.536	0.470	0.554	0.258	0.481	0.255	0.141	0.295
	IDOL	R50	0.465	0.705	0.509	0.458	0.565	0.171	0.342	0.156	0.107	0.302
Q ₁₀₀	CTVIS	R50	0.490	0.712	0.511	0.500	0.617	0.203	0.394	0.185	0.126	0.306
	DVIS	R50	0.500	0.730	0.537	0.470	0.583	0.050	0.086	0.054	0.032	0.065
	GenVis	R50	0.349	0.526	0.386	0.394	0.466	0.153	0.319	0.134	0.102	0.215
	MinVis	R50	0.473	0.710	0.504	0.453	0.560	0.251	0.470	0.240	0.143	0.295
	IDOL	R50	0.478	0.716	0.528	0.458	0.563	0.255	0.475	0.249	0.141	0.369
Q ₃₀₀	CTVIS	R50	0.464	0.678	0.514	0.481	0.590	0.172	0.333	0.162	0.118	0.272
	DVIS	R50	0.498	0.720	0.547	0.475	0.591	0.303	0.569	0.286	0.148	0.362
	GenVis	R50	0.339	0.503	0.365	0.384	0.460	0.167	0.329	0.151	0.110	0.236
	MinVis	R50	0.480	0.705	0.521	0.462	0.568	0.267	0.494	0.249	0.142	0.309
	IDOL	R50	0.468	0.697	0.508	0.474	0.577	0.273	0.478	0.282	0.146	0.372

mean AP, AP₅₀, AP₇₅ and average recall (AR₁ and AR₁₀) for max detections 1 and 10 respectively.

3.9 Experimental Results and Discussion

The length of videos and even different challenging factors, such as occlusions, deformations, and rapid motion changes, increase the complexity of the problem and must be addressed properly by a successful VIS model. Several methods use convolutional-based architectures to segment object instance in frames and later use heuristics to associate these instance based on clues such as mask overlapping ratios (IOU) and the appearance similarity. Hand-designed heuristics may fail in challenging scenarios, such as

occlusions and rapid changes in motion and appearance.

The results shown in table 3.1 reveal that as the complexity of the frames increases, the performance is more likely to vary as the object queries count change. Generally, the choice of backbones as feature extractors for segmentation models is ResNet-(50 or 101) [63] and hierarchical vision transformer using shifted windows (swin-transformer) backbone [56]. Due to the computational requirements, we constrained the experiments to using ResNet-50 as backbone feature extractor for all methods. All models were initialized with their respective pretrained weights (specifically, coco pretrained segmentation weights) for the backbone. CTVIS performs best at at object queries count $Q_c = 100$ with $AP = 0.490$ on Youtube-VIS 2019 and $Q_c = 32$ on OVIS dataset with $AP = 0.242$. DVIS has best performance at $Q_c = 32$ with $AP = 0.493$ and $Q_c = 300$ with $AP = 0.303$ for Youtube-VIS and OVIS respectively. GenVis has best performance at $Q_c = 64$ with $AP = 0.355$ and $Q_c=300$ with $AP = 0.167$. MinVis has $AP = 0.491$ and $AP = 0.267$ for $Q_c = 32$ and $Q_c = 300$ on Youtube-VIS and OVIS respectively. IDOL has $AP = 0.478$ and $AP = 0.273$ at $Q_c = 100$ and $Q_c = 300$. Taking both average precision and average recall into account, there is slightly an increase in performance as the object queries count increases. On average, all methods show less variance on Youtube-VIS 2019 in comparison to OVIS dataset. OVIS dataset has more challenging scenes and sequences have occlusions and instances disappear and reappear more often. All methods are using their respective pre-trained weights, but there is an obvious performance drop on OVIS dataset which can be attributed to the elevated complexity in sequences.

Figure 3.3: Youtube-VIS Qualitative results. Each row represents the method of interest while the column shows different instance masks and instance IDs shown in different colours. The timesteps shown here are $t = [1, 30, 60, 90]$.



4. FOURIER BASED QUERY MIXING FOR ONLINE VIDEO INSTANCE SEGMENTATION

In this section, we study Fourier-Based Query Mixing (FBQM), a method that replaces standard trainable attention mechanisms with parameter-free (i.e., non-trainable) Fourier transforms for token mixing in VIS. By leveraging the computational efficiency of Fourier transforms, it becomes possible to mitigate the heavy (quadratic) computational load typically associated with self-attention while aiming for a similar segmentation accuracy. We study the performance of FBQM approach by integrating it into four state-of-the-art transformer-based VIS frameworks and its effect on the segmentation performance. We report our preliminary results obtained on YouTube-VIS 2019 dataset.

4.1 Background: Fourier-Based Query Mixing

Transformer-based VIS architectures have demonstrated state-of-the-art performance, and they rely on full attention layers. However, they operate at a level requiring quadratic computational complexity [106, 107]. Recent efforts have aimed to reduce the complexity of self-attention through various approximations as in [107, 108]. Among such approximation techniques, the work in [109] proposed a Fourier Transform based approach where they replaced the self-attention layers with simple Fourier transform operations to mix tokens for natural language processing (NLP) tasks. This approach replaces the expensive query, key, value matrix operations in attention layers with efficient Fast Fourier Transform (FFT) [110] operations.

Object detection provided the groundwork for later advances in instance segmentation tasks [27]. With the outstanding performance of transformers, they were heavily adapted and used in many recent vision tasks as in [32, 54]. VIS applies the instance segmentation task on videos by performing both segmentation and the association tasks

Figure 4.1: We provide the overview of the used instance segmentation approaches in all of the used VIS architectures in (A). There are two possible instance segmentation approaches shown in this figure: (i) segmentation via the classic attention mechanism as shown in (B); (ii) segmentation via Fourier-based attention as shown in (D). Multiple scale feature maps from the backbone go into pixel-decoder to arrive at higher-resolution mask features as shown in (A). The transformer decoder further takes these together with query features. Classes and masks are computed from both mask-features and context-aware instance embeddings. (C) illustrates the standard self-attention mechanism, while (D) shows the Fourier-based replacement module.



over all the video frames.

Inspired by FNet’s performance on NLP tasks, we study the applicability and performance of Fourier transform approximations in VIS tasks. By substituting traditional self-, and cross-attention layers with Fourier transforms, we aim to reduce the computational burden in VIS applications. This is especially beneficial for online VIS models, where real-time performance is important.

4.2 Revisiting Transformer Based VIS Pipelines

In this thesis, we consider and use four recent transformer based online VIS models including DVIS [34], CTVIS [29], GENVIS [111], and MinVIS [33]. In each of those models, we replaced the attention mechanisms with Fourier blocks. The details of a Fourier block as used in our approach is given in Fig. 4.1-D, while Fig. 4.1-A provides

the overall architecture used for the instance segmentation in each of those VIS models. Online VIS models, as used in this thesis, share several components that function as a whole in end-to-end fashion [62]. These components include a backbone that extract features, a pixel-decoder that enriches the features, and a segmentation-classification head that predicts the instance masks and instance categories. **Backbone module** is used to extract deep convolutional features. In all the selected VIS methods, we use ResNet-50 as the backbone. The backbone is responsible for generating feature maps that used in the following modules. While processing the inputs to of the backbone, the feature maps obtained at different levels (in particular, Res2, Res3, Res4, and Res5) are retained for later to be used in building high-resolution feature maps. **Pixel-decoder module** receives feature maps obtained at different levels in the backbone and further processes them via pixel decoders (such as the feature pyramid network-FPN [112], or multi-scale deformable attention-based pixel decoder-MSDeformAttnPixelDecoder [62]) to produce high-resolution feature maps. **Transformer module** produces Q query embeddings by taking feature maps from the backbone and initial query embeddings and additionally attention mask for Mask2Former [62] (Mask2Former was used in all of the studied VIS architectures). The transformer module is based on the standard transformer decoder architecture [64]. It comprises of both self- and cross-attention blocks and will process feature maps from the backbone and different levels of pixel-decoder and attention mask to encode context information in object queries. **Segmentation head** comprises of multi-layer perceptron, that takes feature from the pixel-decoder and context-aware instance queries from the transformer decoder module and produces $h \times w$ binary mask per object query. **Classification module** is responsible with producing $Q \times k$ class probabilities, where k is the number of classes including the non-object class, and Q is the total instance query count. **Instance Association Module** is responsible for associating instances in successive frames. All the methods share the notion of per-frame instance segmentation and

use instance embeddings. However, their association mechanisms differ mainly in where the matching happens. GenVIS[111] directly propagates previous instance queries as new frames come in. Other considered VIS techniques in this thesis explicitly perform matching step using similarity metric and assignment (Hungarian matching) based on instance embeddings. This introduces a notion of memory that holds past instance embeddings which is gradually updated as new instances are detected.

In each of the studied VIS methods, we only modify the standard self-, and cross-attention mechanisms by replacing them with Fourier-based attention in the transformer module for the instance segmentation (spatial) part of a VIS task.

4.3 Fourier-Based Attention Layers for VIS

In the Fourier-Based Attention layer implementation, each standard self-, and cross-attention layers are replaced by a Fourier self-, and cross-attention layers respectively as described in [109]. In a standard attention mechanism, shown in Fig. 4.1, a set of keys, queries, and value matrices are used in attention mechanism in order to arrive at context-aware vectors as illustrated in Eq. 4.1.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_{model}}} \right) V \quad (4.1)$$

where Q , K , and V are the query, key, and the value embeddings respectively, and d_{model} is the embedding dimension.

In the case of self-attention, the queries, keys, and values all come from the same input sequence. However, for cross-attention, the keys and values come from a same source (usually encoder output), but the queries (for example, initial instance queries for VIS tasks) come from a separate module [64]. Instead of computing pairwise dot-product attention, the Fourier layer applies FFT [110] to the input token sequence. The FFT is an efficient implementation of the Discrete Fourier Transform (DFT) [113]. For a discrete

time-domain signal $x[n]$ of length N , the DFT of $x[n]$ is defined as $X[k]$ and is computed as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-2\pi i \frac{kn}{N}}, \quad k = 0, 1, \dots, N-1, \quad (4.2)$$

where $X[k]$ is the k -th frequency component, $x[n]$ is the input signal, N is the total number of points in the FFT. $X[k]$ is a complex value for each frequency k and can be defined as the sum of its real $Re(X[k])$ and imaginary $Im(X[k])$ components. In our implementation, these components are channel-wise concatenated and are projected back to embedding dimension (d_{model}) via a linear project layer. Similarly, the cross-attention layers in the decoder are replaced by Fourier Cross-Attention layers.

After the Fourier transform and projection, we preserve the subsequent normalization (LayerNorm) and feed-forward network components from the original self-attention block.

4.4 Experiments and Discussion

In our experiments, the key change is the replacement of the attention layers with Fourier-based layers, while keeping all other components of a VIS pipeline intact. We conduct our experiments on YouTube-VIS 2019 [30] dataset. All the VIS models are using ResNet-50 as the backbone. The hyper-parameters used during the training (such as the learning rate, batch size, and iteration count) are kept as consistent as possible across models.

Our preliminary experimental evaluation of Fourier Based Query Mixing integrated into online VIS frameworks yields promising results. Quantitative evaluations on YouTube-VIS 2019 Table 4.1 indicate that it is possible to train VIS models with Fourier-based attention while there is still room for improvement on training the Fourier-based

Figure 4.2: Qualitative results for CTVIS on Youtube-VIS 2019 sequence d7203fdab6 ;top pairs, and ec6fd219; bottom pairs. The shown frames are the timesteps: 0, 30, 60, and 90 (from left to right). Object colors represent the object IDs. Here S.A. and F.A.



Table 4.1: Results for YoutubeVis-2019 reported as mean average precision for both standard, and Fourier-based attentions on four VIS models.

Attn.	Method	Back.	YouTube-VIS 2019				
			AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀
Standard	CTVIS	R50	0.49	0.71	0.51	0.50	0.62
	DVIS	R50	0.50	0.73	0.54	0.47	0.58
	GenVis	R50	0.35	0.53	0.39	0.39	0.47
	MinVis	R50	0.47	0.71	0.50	0.45	0.56
Fourier	CTVIS	R50	0.35	0.51	0.38	0.39	0.44
	DVIS	R50	0.05	0.07	0.05	0.19	0.19
	GenVis	R50	0.12	0.20	0.11	0.23	0.25
	MinVis	R50	0.36	0.56	0.38	0.38	0.46

models. CTVIS [29] and MinVis [33] yielded the highest results meaning that Fourier-based attention can be used to train VIS models. DVIS [34] however, show a drastic gap of 0.455 which indicates that there might be a different approach to use in choosing the hyperparameters when the VIS models are trained with Fourier-based attention. Qualitatively, standard attention-based models are able to predict more instances and are able to maintain the temporal consistency as shown in top-row of Fig. ???. Fourier-based models, on the other hand, are able to detect most instances in the frames, however experience more identity switches and false-positives as show in bottom-row. Overall, qualitative and qualitative results suggest that FBQM can be used to perform VIS with small degradation in performance when compared to the classical attention mechanisms. This suggests that FBQM can also help learning the segmentation masks in state-of-the-art VIS models while reducing the computational complexity of the attention mechanism.

5. CONCLUSION

In the first part of this thesis, we introduce a novel image alignment algorithm that we call VisIRNet. With VisIRNet, we demonstrate that it is possible to predict homography matrices necessary for image alignment task on non-aligned image pairs without explicitly knowing the intrinsic properties of the imaging system used. Our experimental results conducted on 5 multi- and single modality datasets namely, VEDAI, SkyData, MSCOCO, GoogleMap, and Google Earth show that our proposed algorithm performs state of the art results, when it is compared to other approaches.

Image alignment on image pairs taken by different onboard cameras on UAVs is a challenging and important topic for various aerial applications. When the images to be aligned are acquired by different modalities, the classic approaches, such as the SIFT and RANSAC combination, become limited on the alignment process. Deep learning based techniques can be more reliable in such situations as our experimental results demonstrate. LK based deep techniques recently shown promise, however, we demonstrate with our approach (VisIRNet) that without designing any LK based block, and by focusing only on the four corner points, we can sufficiently train deep architectures for image alignment.

Furthermore, in the second part of this thesis, extending the analysis to video domain, we study query-based online video instance segmentation first by evaluating the performance differences under different query parameterizations. On top of that, we introduced using Fourier Based Query Mixing (FBQM) in VIS models as a replacement of the traditional attention mechanisms and studied its performance. In particular, we replaced the traditional self-attention and cross-attention layers with computationally more efficient Fourier transform-based layers in four state-of-the-art VIS models including DVIS, CTVIS, GENVIS, and MinVIS, and we compared its performance to the attention based approach on YouTube-VIS 2019 dataset. Future directions may explore integrating the Fourier-based modules into other types of attention such as deformable attention, and

masked attention mechanisms.



REFERENCES

- [1] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Deep image homography estimation,” *CoRR*, vol. abs/1606.03798, 2016.
- [2] H. Le, F. Liu, S. Zhang, and A. Agarwala, “Deep homography estimation for dynamic scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [3] C.-H. Chang, C.-N. Chou, and E. Y. Chang, “Clkn: Cascaded lucas-kanade networks for image alignment,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [4] Y. Zhao, X. Huang, and Z. Zhang, “Deep lucas-kanade homography for multimodal image alignment,” *CoRR*, vol. abs/2104.11693, 2021.
- [5] S. Özer, M. Ege, and M. A. Özkanoglu, “Siamesefuse: A computationally efficient and a not-so-deep network to fuse visible and infrared images,” *Pattern Recognition*, vol. 129, p. 108712, 2022.
- [6] M. A. Özkanoglu and S. Ozer, “Infragan: A gan architecture to transfer visible images to infrared domain,” *Pattern Recognition Letters*, vol. 155, pp. 69–76, 2022.
- [7] S. Ozer, E. Ilhan, M. A. Ozkanoglu, and H. A. Cirpan, “Offloading deep learning powered vision tasks from uav to 5g edge server with denoising,” *IEEE Transactions on Vehicular Technology*, 2023.
- [8] S. Bhowmick, *The RGB rendering of visible wavelength lights (2019 02 28 14 47 31 UTC)*. 01 2017.
- [9] D. Carreres-Prieto, J. T. García, F. Cerdán-Cartagena, and J. Suardiaz-Muro, “Performing calibration of transmittance by single rgb-led within the visible spectrum,” *Sensors*, vol. 20, no. 12, 2020.
- [10] M. Magnusson, J. Sigurdsson, S. E. Armansson, M. O. Ulfarsson, H. Deborah, and J. R. Sveinsson, “Creating rgb images from hyperspectral images using a color matching function,” in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2045–2048, 2020.
- [11] C. Cucci, A. Casini, P. Marcello, and L. Stefani, “Extending hyper-spectral imaging from vis to nir spectral regions: A novel scanner for the in-depth analysis of polychrome surfaces,” *Proc SPIE*, vol. 8790, pp. 09–, 05 2013.
- [12] J. Delaney, M. Thoury, J. Zeibel, P. Ricciardi, K. Morales, and K. Dooley, “Visible and infrared imaging spectroscopy of paintings and improved reflectography,” *Heritage Science*, vol. 4, 03 2016.

- [13] G. Fox, “The brewing industry and the opportunities for real-time quality analysis using infrared spectroscopy,” *Applied Sciences*, vol. 10, 01 2020.
- [14] R. Gade and T. B. Moeslund, “Thermal cameras and applications: a survey,” *Machine Vision and Applications*, vol. 25, pp. 245–262, Jan 2014.
- [15] A. A. Goshtasby, *Image Registration - Principles, Tools and Methods*. Advances in Computer Vision and Pattern Recognition, Springer, 2012.
- [16] F. Alam, S. Ur Rahman, A. Din, and F. Qayum, “Medical image registration: Classification, applications and issues,” *Journal of Postgraduate Medical Institute*, vol. 32, pp. 300–3007, 12 2018.
- [17] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes, “Medical image registration,” *Physics in Medicine and Biology*, vol. 46, p. R1, mar 2001.
- [18] M. I. McCartney, S. Zein-Sabatto, and M. Malkani, “Image registration for sequence of visual images captured by uav,” in *2009 IEEE Symposium on Computational Intelligence for Multimedia Signal and Vision Processing*, pp. 91–97, 2009.
- [19] S.-M. Huang, C.-C. Huang, and C.-C. Chou, “Image registration among uav image sequence and google satellite image under quality mismatch,” in *2012 12th International Conference on ITS Telecommunications*, pp. 311–315, 2012.
- [20] S. Chen, F. Yu, and X. Zhu, “Real-time registration in image stitching under the microscope,” in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 907–911, 2018.
- [21] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin, “An automatic image registration for applications in remote sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 9, pp. 2127–2137, 2005.
- [22] Z. li Song, S. Li, and T. F. George, “Remote sensing image registration approach based on a retrofitted sift algorithm and lissajous-curve trajectories,” *Opt. Express*, vol. 18, pp. 513–522, Jan 2010.
- [23] Y. F. LI Fuyu, “Summarization of sift-based remote sensing image registration techniques,” *Remote Sensing for Natural Resources*, vol. 28, no. 2, p. 14, 2016.
- [24] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin, “An automatic image registration for applications in remote sensing,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, pp. 2127 – 2137, 10 2005.
- [25] V. Mochalov, O. Grigorieva, D. Zhukov, A. Markov, and A. Saidov, “Remote sensing image processing based on modified fuzzy algorithm,” in *Artificial Intelligence and Bioinspired Computational Methods* (R. Silhavy, ed.), (Cham), pp. 563–572, Springer International Publishing, 2020.

- [26] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [27] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A review of yolo algorithm developments,” *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.
- [28] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.
- [29] K. Ying, Q. Zhong, W. Mao, Z. Wang, H. Chen, L. Y. Wu, Y. Liu, C. Fan, Y. Zhuge, and C. Shen, “CTVIS: Consistent Training for Online Video Instance Segmentation,” 2023.
- [30] L. Yang, Y. Fan, and N. Xu, “Video instance segmentation,” in *Proceedings of the IEEE/CVF ICCV*, pp. 5188–5197, 2019.
- [31] G. Bertasius and L. Torresani, “Classifying, segmenting, and tracking object instances in video with mask propagation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9739–9748, 2020.
- [32] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, “End-to-end video instance segmentation with transformers,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8741–8750, 2021.
- [33] D.-A. Huang, Z. Yu, and A. Anandkumar, “Minvis: A minimal video instance segmentation framework without video-based training,” 2022.
- [34] T. Zhang, X. Tian, Y. Wu, S. Ji, X. Wang, Y. Zhang, and P. Wan, “Dvis: Decoupled video instance segmentation framework,” *arXiv preprint arXiv:2306.03413*, 2023.
- [35] X. Deng, E. Liu, S. Li, Y. Duan, and M. Xu, “Interpretable Multi-Modal Image Registration Network Based on Disentangled Convolutional Sparse Coding,” *IEEE Transactions on Image Processing*, vol. 32, pp. 1078–1091, Jan. 2023.
- [36] Y. Luo, X. Wang, Y. Wu, and C. Shu, “Infrared and visible image homography estimation using multiscale generative adversarial network,” *Electronics*, vol. 12, no. 4, 2023.
- [37] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” *CoRR*, vol. abs/1506.02025, 2015.
- [38] P. Monasse, “Extraction of the Level Lines of a Bilinear Image,” *Image Processing On Line*, vol. 9, pp. 205–219, 2019. <https://doi.org/10.5201/ipol.2019.269>.
- [39] E. J. Kirkland, *Bilinear Interpolation*, pp. 261–263. Boston, MA: Springer US, 2010.

- [40] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014.
- [41] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery : A small target detection benchmark,” *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.
- [42] J. A. Rice, *Mathematical Statistics and Data Analysis*. Belmont, CA: Duxbury Press, 3rd ed., 2007.
- [43] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [44] X. Wang, I. Misra, Z. Zeng, R. Girdhar, and T. Darrell, “Videocutler: Surprisingly simple unsupervised video instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22755–22764, 2024.
- [45] R. Panero Martinez, I. Schiopu, B. Cornelis, and A. Munteanu, “Real-time instance segmentation of traffic videos for embedded devices,” *Sensors*, vol. 21, no. 1, p. 275, 2021.
- [46] L. Mou and X. X. Zhu, “Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6699–6711, 2018.
- [47] C.-H. Tseng, C.-C. Hsieh, D.-J. Jwo, J.-H. Wu, R.-K. Sheu, and L.-C. Chen, “Person retrieval in video surveillance using deep learning-based instance segmentation,” *Journal of Sensors*, vol. 2021, no. 1, p. 9566628, 2021.
- [48] X. Zhu, J. Liang, and A. Hauptmann, “Msnet: A multilevel instance segmentation network for natural disaster damage assessment in aerial videos,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2023–2032, 2021.
- [49] Z. Gu, H. Chen, and Z. Xu, “Diffusioninst: Diffusion model for instance segmentation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2730–2734, IEEE, 2024.
- [50] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for universal image segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

- [51] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, “Transunet: Transformers make strong encoders for medical image segmentation,” *arXiv preprint arXiv:2102.04306*, 2021.
- [52] J. M. J. Valanarasu and V. M. Patel, “Unext: Mlp-based rapid medical image segmentation network,” in *International conference on medical image computing and computer-assisted intervention*, pp. 23–33, Springer, 2022.
- [53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [54] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, pp. 213–229, Springer, 2020.
- [55] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, “Mask dino: Towards a unified transformer-based framework for object detection and segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3041–3050, 2023.
- [56] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.
- [57] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, “Visual attention network,” *Computational Visual Media*, vol. 9, no. 4, pp. 733–752, 2023.
- [58] K. Ying, Q. Zhong, W. Mao, Z. Wang, H. Chen, L. Y. Wu, Y. Liu, C. Fan, Y. Zhuge, and C. Shen, “CTVIS: Consistent Training for Online Video Instance Segmentation,” 2023.
- [59] T. Zhang, X. Tian, Y. Wu, S. Ji, X. Wang, Y. Zhang, and P. Wan, “Dvis: Decoupled video instance segmentation framework,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1282–1291, 2023.
- [60] M. Heo, S. Hwang, J. Hyun, H. Kim, S. W. Oh, J.-Y. Lee, and S. J. Kim, “A generalized framework for video instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14623–14632, 2023.
- [61] D.-A. Huang, Z. Yu, and A. Anandkumar, “Minvis: A minimal video instance segmentation framework without video-based training,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 31265–31277, 2022.
- [62] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for universal image segmentation,” 2022.

- [63] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [65] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.
- [66] B. Cheng, A. Schwing, and A. Kirillov, “Per-pixel classification is not all you need for semantic segmentation,” *Advances in neural information processing systems*, vol. 34, pp. 17864–17875, 2021.
- [67] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, “Sipmask: Spatial information preservation for fast image and video instance segmentation,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 1–18, Springer, 2020.
- [68] S. Yang, Y. Fang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, “Crossover learning for fast online video instance segmentation,” in *proceedings of the IEEE/CVF international conference on computer vision*, pp. 8043–8052, 2021.
- [69] L. Ke, X. Li, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, “Prototypical cross-attention networks for multiple object tracking and segmentation,” *Advances in neural information processing systems*, vol. 34, pp. 1192–1203, 2021.
- [70] M. Li, S. Li, L. Li, and L. Zhang, “Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11215–11224, 2021.
- [71] Y. Fu, L. Yang, D. Liu, T. S. Huang, and H. Shi, “Compfeat: Comprehensive feature aggregation for video instance segmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1361–1369, 2021.
- [72] J. Wu, Q. Liu, Y. Jiang, S. Bai, A. Yuille, and X. Bai, “In defense of online models for video instance segmentation,” in *ECCV*, 2022.
- [73] Y. Zhou, T. Zhang, S. Ji, S. Yan, and X. Li, “Dvis-daq: Improving video segmentation via dynamic anchor queries,” *arXiv*, 2024.
- [74] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

- [75] J. Wu, Y. Jiang, S. Bai, W. Zhang, and X. Bai, “Seqformer: Sequential transformer for video instance segmentation,” in *European Conference on Computer Vision*, pp. 553–569, Springer, 2022.
- [76] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, “Stem-seg: Spatio-temporal embeddings for instance segmentation in videos,” in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 158–177, Springer, 2020.
- [77] H. Lin, R. Wu, S. Liu, J. Lu, and J. Jia, “Video instance segmentation with a propose-reduce paradigm,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1739–1748, 2021.
- [78] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim, “Video instance segmentation using inter-frame communication transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 13352–13363, 2021.
- [79] J. Wu, Y. Jiang, W. Zhang, X. Bai, and S. Bai, “Seqformer: a frustratingly simple model for video instance segmentation,” *arXiv preprint arXiv:2112.08275*, vol. 2, no. 3, p. 4, 2021.
- [80] S. Yang, X. Wang, Y. Li, Y. Fang, J. Fang, W. Liu, X. Zhao, and Y. Shan, “Temporally efficient vision transformer for video instance segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2885–2895, 2022.
- [81] M. Heo, S. Hwang, S. W. Oh, J.-Y. Lee, and S. J. Kim, “Vita: Video instance segmentation via object token association,” *Advances in neural information processing systems*, vol. 35, pp. 23109–23120, 2022.
- [82] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, “Mots: Multi-object tracking and segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7942–7951, 2019.
- [83] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2020.
- [84] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, “Vit-yolo: Transformer-based yolo for object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2799–2808, 2021.
- [85] P. Lv, L. Ma, Q. Li, and F. Du, “Shapeformer: a shape-enhanced vision transformer model for optical remote sensing image landslide detection,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 2681–2689, 2023.

- [86] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [87] K. Luo and I.-b. Lian, “Building a vision transformer-based damage severity classifier with ground-level imagery of homes affected by california wildfires,” *Fire*, vol. 7, no. 4, p. 133, 2024.
- [88] C.-F. R. Chen, Q. Fan, and R. Panda, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.
- [89] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, “Vision transformers for remote sensing image classification,” *Remote Sensing*, vol. 13, no. 3, p. 516, 2021.
- [90] Y. Guo, C. Wang, S. X. Yu, F. McKenna, and K. H. Law, “Adaln: a vision transformer for multidomain learning and predisaster building information extraction from images,” *Journal of Computing in Civil Engineering*, vol. 36, no. 5, p. 04022024, 2022.
- [91] R. Koshy and S. Elango, “Multimodal tweet classification in disaster response systems using transformer-based bidirectional attention model,” *Neural Computing and Applications*, vol. 35, no. 2, pp. 1607–1627, 2023.
- [92] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr, “Lavt: Language-aware vision transformer for referring image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18155–18165, 2022.
- [93] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, “Segmenter: Transformer for semantic segmentation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7262–7272, 2021.
- [94] R. Ghali, M. A. Akhloufi, M. Jmal, W. Souidene Mseddi, and R. Attia, “Wildfire segmentation using deep vision transformers,” *Remote Sensing*, vol. 13, no. 17, p. 3527, 2021.
- [95] J. Wu, Q. Liu, Y. Jiang, S. Bai, A. Yuille, and X. Bai, “In defense of online models for video instance segmentation,” in *European Conference on Computer Vision*, pp. 588–605, Springer, 2022.
- [96] G. Yenduri, G. Srivastava, P. K. R. Maddikunta, R. H. Jhaveri, W. Wang, A. V. Vasilakos, T. R. Gadekallu, *et al.*, “Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions,” *arXiv preprint arXiv:2305.10435*, 2023.

- [97] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, “Attention mechanisms in computer vision: A survey,” *Computational visual media*, vol. 8, no. 3, pp. 331–368, 2022.
- [98] D. Hu, “An introductory survey on attention mechanisms in nlp problems,” in *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*, pp. 432–448, Springer, 2020.
- [99] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [100] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [101] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.
- [102] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.
- [103] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [104] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, “Ota: Optimal transport assignment for object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 303–312, 2021.
- [105] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. H. Torr, and S. Bai, “Occluded video instance segmentation: A benchmark,” *International Journal of Computer Vision*, vol. 130, no. 8, pp. 2022–2039, 2022.
- [106] F. D. Keles, P. M. Wijewardena, and C. Hegde, “On the computational complexity of self-attention,” in *International conference on algorithmic learning theory*, pp. 597–619, PMLR, 2023.
- [107] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, “Linformer: Self-attention with linear complexity,” *Preprint, arXiv:2006.04768*, 2020.
- [108] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient transformers: A survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2022.
- [109] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, “Fnet: Mixing tokens with fourier transforms,” *arXiv preprint arXiv:2105.03824*, 2021.

- [110] P. Heckbert, “Fourier transforms and the fast fourier transform (fft) algorithm,” *Computer Graphics*, vol. 2, no. 1995, pp. 15–463, 1995.
- [111] M. Heo, S. Hwang, J. Hyun, H. Kim, S. W. Oh, J.-Y. Lee, and S. J. Kim, “A generalized framework for video instance segmentation,” in *CVPR*, 2023.
- [112] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” 2017.
- [113] W. L. Briggs and V. E. Henson, *The DFT: an owner’s manual for the discrete Fourier transform*. SIAM, 1995.



