DEEPFAKE DETECTION THROUGH MOTION MAGNIFICATION INSPIRED FEATURE MANIPULATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Aydamir Mirzayev
September 2022

Deepfake Detection through Motion Magnification Inspired Feature Manipulation
By Aydamir Mirzayev
September 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Hamdi Dibeklioğlu(Advisor)

Ayşegül Dündar

Albert Ali Salah

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

DEEPFAKE DETECTION THROUGH MOTION MAGNIFICATION INSPIRED FEATURE MANIPULATION

Aydamir Mirzayev
M.S. in Computer Engineering
Advisor: Hamdi Dibeklioğlu
September 2022

Synthetically generated media content poses a significant threat to information security in the online domain. Manipulated videos and images of celebrities, politicians, and ordinary citizens, if aimed at misrepresentation, and defamation can cause significant damage to one's reputation. Early detection of such content is crucial to timely alleviation of further spread of questionable information. In the past years, a significant number of deepfake detection frameworks have proposed to utilize motion magnification as a preprocessing step aimed at revealing transitional inconsistencies relevant to the prediction outcome. However, such an approach is sub-optimal since often utilized motion manipulation approaches are optimized for a limited set of controlled motions and display significant visual artifacts when used outside of their domain. To this end, rather than apply motion magnification as a separate processing step, we propose to test trainable motion magnification-inspired feature manipulation units as an addition to a convolutional-LSTM classification network. In our approach, we aim to take the first step at understanding the use of magnification-like architectures in the task of video classification rather than aim at full integration. We test out results on the Celeb-DF dataset which is composed of more than five thousand synthetically generated videos generated using DeepFakes fake generation method. We treat manipulation unit as another network layer and test the performance of the network both with and without it. To ensure the consistency of our results we perform multiple experiments with the same configurations and report the average accuracy. In our experiments we observe an average 3% jump accuracy when the feature manipulation unit is incorporated into the network.

Keywords: Deepfake detection, computer vision, facial expression, motion magnification, video classification, deep learning, spatio-temporal analysis .

ÖZET

HAREKET BÜYÜTMESİNDEN ESİNLENEN ÖZNİTELİK MANİPÜLASYONU İLE DERİN SAHTELERİN TESPİTİ

Aydamir Mirzayev
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Hamdi Dibeklioğlu
Eylül 2022

Sentetik olarak oluşturulmuş medya içeriği, çevrimiçi alanda bilgi güvenliği için önemli bir tehdit oluşturmaktadır. Ünlülerin, politikacıların ve sıradan vatandaşların manipüle edilmiş videoları ve görüntüleri, yanlış beyan ve iftira amaçlıysa, kişinin itibarına ciddi zarar verebilir. Bu tür içeriğin erken tespiti, şüpheli bilgilerin hızlı yayılmasının zamanında önlenmesi için çok önemlidir. Geçtiğimiz yıllarda, önemli sayıda derin sahte algılama metodu, uzay-zaman tutarsızlıklarını ortaya çıkarma amacıyla, bir ön işleme adımı olarak, hareket büyütme teknolojisin kullanılabileceğini ileri sürdü. Fakat, sıklıkla kullanılan hareket büyütme metodları, sınırlı bir hareket kümesi için optimize edildiğinden ve etki alanlarının dışında kullanıldığında önemli görsel yapaylıklar gösterdiğinden, bu tür bir ön işleme yaklaşımı optimal değildir. Bu amaçla, hareket büyütmeyi ayrı bir işlem adımı olarak uygulamak yerine, bir CNN-LSTM sınıflandırma ağına, ek olarak eklenen, eğitilebilir, hareket büyütmeden esinlenen, öznitelik manipulasyonu unitesini test etmeyi öneriyoruz. Yaklaşımımızda, tam entegrasyondan ziyade, video sınıflandırma görevinde, hareket büyütme benzeri mimarilerin kullanımını anlamak için ilk adımları atmayı hedefliyoruz. Sonuçları DeepFakes sahte oluşturma yöntemini kullanarak oluşturulan, ve beş binden fazla sentetik video içeren Celeb-DF veri kümesinde test ediyoruz. Sonuçlarımızın tutarlılığını göstermek için aynı deneysel kurulumla birden fazla deney gerçekleştirerek ortalama doğruluğu sunuyoruz. Deneylerimizde, öznitelik manipulasyonu unitesini ağa dahil edildiğinde doğruluğun ortalama 3% artışlnı gözlemliyoruz.

Anahtar sözcükler: Derin sahte tespiti, bilgisayar görüşü, haraket büyütme, video sınıflandırması, derin öğrenme, uzay-zamansal analiz.

Acknowledgement

First and foremost it is my pleasure to express my deep gratitude to my graduate advisor, Asst. Prof. Dr. Hamdi Dibeklioğlu. It is through his help and insightful guidance that I was able to carry on through my graduate education and progress in my respective field. Dr. Dibeklioğlu was both a very insightful advisor and a thoughtful fellow thorough my education and for that, I will be forever grateful.

Next, I would like to thank Asst. Prof. Ayşegül Dündar and Prof. Albert Ali Salah for taking their valuable time to serve as members of my thesis committee and providing their perceptive feedback.

I would also like to thank my friends in Bilkent and outside of it for being there when I needed them and for never failing to provide emotional support and encouragement.

Last but not least, I would like to thank my family, my mother, father, and two wonderful sisters for never ceasing to support me and cheering me on in this laborious journey. In hopes of encouraging him, I would like to dedicate this thesis to my nephew Emil who wants to design video games as a computer scientist when he grows up.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Although recent in public popularity, face replacement and manipulation is an ever-growing sub-field of computer biometrics that has been around for a while now. Earlier approaches rely on signal processing and geometrical mesh computations to estimate the spatial dimensions of the target and the source faces and map one onto the other. However, the introduction of generative, and reconstructive networks, growing availability of large scale datasets and improvements in GPU technology have propelled the utilization of learning based approaches in the field.

In recent years, owing to the fast-paced advancement, increased availability of potent convolutional neural networks, and commercialization of the face-swapping technology, through mobile apps such as Face-App, there has been a surge in the amount of fake video and photo media content propagating through the internet. Such content, although more often created for entertainment purposes, has also been used to harm and taunt the reputation of targeted individuals, manipulate political debates, and influence elections. Although unlikely to fully prevent the spread of misinformation, timely detection of such manipulated content can significantly alleviate the intended harm caused by deepfake media. Automatic deepfake detection routines can also help to increase the speed of misinformation

labeling that has been employed by social networks such as Twitter and Facebook; potentially saving management resources, and preventing possible large-scale damages and scams.

In 2019 Facebook launched a deepfake detection challenge which brought in several scholars as competitors to develop methods for early detection of manipulated content. Although promising performance-wise, much of the top-scoring results relied on vanilla CNN networks and frame-based classification, while failing to thoroughly explore temporal relations in the dataset. Since then many new methods have emerged which tackled spatiotemporal cues through use of features such as optical flow [1, 2], color fluctuation [3, 4], behavioral patterns [5, 6], and more. One family of spatiotemporal analysis approaches relies on the use of motion magnification [7, 8, 9] as means of revealing temporal inconsistencies in the video and thus aiding the detection process. A common theme among such approaches is the use of motion magnification as a preprocessing step which although potentially effective is less than ideal as magnification networks are usually designed to work on a limited set of motions that might fall outside of the domain that is relevant to deepfake detection. In this work, instead of using magnification as preprocessing step, we are aiming to take a first step at understanding the effects of task-specific magnification like architectures on classification results. We borrow and modify the motion feature manipulation unit, and plug it into our deepfake classification network to measure relative improvements over the baseline network. We show that a magnification-inspired motion manipulator unit can lead to accuracy improvement on a deepfake classification network.

1.1 Research Question and the Motivation

Early motion magnification approaches propose to amplify subtle micro-motions such as trembling of steel structures, human micro-movements, fluctuations of a swing, etc., that would usually be challenging to spot with a human eye [7, 9]. Proposed methods use a combination of automatic motion grouping, tracking,

and human input to isolate the motions of interest in the frame. More often than not, such approaches assume the absence of larger motions in the recording. Liu *et al.* [7], for example, make use of user input to help correct for imperfections in layer segmentation. Similarly, Wu *et al.* [9] specifically focus on small fluctuations in the absence of the larger motions, and incorporate the user selected band of frequencies for an optimal result. Although such methods can operate without manual user adjustment, the authors specifically note that optimal results are achieved with a restricted band of motions and semi-automatic configuration.

Despite this seeming limited domain of applicability many deepfake detection frameworks, inspired by initial attempts in video spoofing detection literature [10, 11], propose to use magnification as preprocessing step [12, 13], with the goal of it enhancing prediction accuracy of convolutional networks. For example, learning based motion magnification model, proposed by Oh *et al.* [8] that is often utilized in deepfake detection literature as a preprocessing method, is trained to operate on a synthetically generated motion enhancement dataset and is only validated on a limited set of samples outside of that dataset. Moreover, it is specifically optimized to preserve texture and shape consistency in the reconstructed video recording. Such reconstruction is in opposition to the deepfake detection objective that benefits from the amplification of inconsistencies. And although many of the methods utilizing magnification as preprocessing do report accuracy improvements, the reasoning outlined above provides sufficient grounds to ask if such use of motion magnification is indeed optimal not only for video forensics but video classification in general.

It is not uncommon in deep learning to observe performance improvements on a classification task without being able to fully interpret the factors leading to that improvement. This is in part due to large modularity of network architectures and the complexity of the dataset at hand. We argue that interpretation of the accuracy improvement with a combination of motion magnification as preprocessing and a convolutional-LSTM classification is unreliable due to the high complexity of the system. The interpretability, which requires comprehensive experimental evaluations, is further complicated by the high computational and

memory cost associated with preprocessing the entire datasets with motion magnification. Considering that adjustment of motion magnification techniques to deepfake detection would require optimization across different frequency bands, magnification factors, and regions of interest, efficient and cost-effective estimation of such parameters seems challenging. We, therefore, argue that a much more tempered approach is needed if we want to understand the contribution of magnification-like architectures to deepfake detection and video classification in general.

Considering all of the points made above, we pose our main research question as follows: *"What type of magnification-like feature manipulation architectures can lead to performance improvement on a deepfake classification task?"*. We believe that answering this question can have following practical benefits:

- This might serve as a first step in understanding the applicability of learned, task-specific motion magnification-inspired feature manipulation in detection of manipulated video content. Such knowledge might prove useful in gauging the potential use of similar feature manipulation in classification tasks beyond deepfake classification.
- As the main contribution of any deepfake detection task, this method can introduce improvements to the literature on deepfake detection, therefore, contributing to the protection of individuals, and organizations from personal attacks and scams.

In answering the research question posed above, we avoid using motion magnification as preprocessing step, but rather device and test magnification-like, learnable feature manipulation unit, inspired by the architecture proposed by Oh *et al.* [8], and measure its contribution to the resulting classification accuracy. Instead of focusing on the visual reconstructive consistency of the network, which is outside the scope of our objective, we re-target the loss function to optimize for final classification accuracy in the network. Consequently, we do not enforce texture or shape consistency in generated frames which, as previously

stated, is in contradiction to our main objective. This prevents the overloading of the network with an expensive reconstructive load and instead reserves computational resources for accurate prediction. In addition, we replace the rigid and predefined scalar multiplication, with amplifying and dampening learnable multiplication array. We ensure smooth variability across the multiplication mask and add input-dependent variability. We believe that the following are the main contributions of our research:

- We propose and test the use a plug-in, magnification-inspired, learnable, dynamic feature manipulation unit for the task of deepfake detection. We display that it can yield accuracy improvements over the baseline network.
- We extend the existing feature manipulation unit by swapping scalar multiplication factor with a learnable, and input specific multiplication mask.
- Finally, we let the values across different sections of the mask vary smoothly while enforcing dampening beside amplification, which adds another dimension to learned feature manipulation.

Chapter 2

Related Work

In this section, we review existing literature on deepfake generation, detection, and use of motion magnification in video forensics. We first go over the evolution of fake generation methods from rule-based geometric approaches to learning based generative networks and talk over existing detection techniques. We then discuss motion magnification frameworks and their use in video and photo forensics.

2.1 Generation and Detection Methods

One of the factors that complicate manipulated content detection is the diversity and multiplicity of available fake generation methods which makes the development of source-agnostic detection methods challenging. Such techniques are not limited to face manipulations either, and expand well beyond the biometric domain, with works in the manipulation of scenery, viewpoint, motion, and everyday objects [14, 15, 16, 17, 18, 19]. Zhou *et al.* [16], for example, utilize two generative networks to design a person-to-person body motion transfer framework while Bansal *et al.* [18] use spatiotemporally constrained generative adversarial networks for appearance and motion transfer of, both, human faces and random

objects. Such approaches vary in goal and implementation but are relevant topics of past and present computer vision research. Similarly, the science of face manipulation is an ever-growing branch of such frameworks and has been around for more than two decades. While earlier approaches utilize computational geometry and linear algebra to compute mapping, reordering, and reconstruction of targeted video frames [20, 21, 22, 23, 24], more recent approaches increasingly rely on the use of generative, neural networks, and pixel-to-pixel transfer [25, 26, 27, 28, 29, 30]. In the context of face transfer and mapping, the photo or the video that is used as a canvas is referred to as the target, while the face that replaces the original is usually referred to as the source.

One of the earliest frameworks on face swapping by Blanz *et al.* [22] computes 3D and texture estimates of both target and source faces with a linear combination of existing face models while using standard graphics and rendering routines to match color and illumination. Authors first compute the boundaries of the domain of possible texture and 3D shapes by averaging a sample 200 faces and restricting any linear combination to lie withing few standard deviations from the mean. From a 3D estimate of the face, the final image is computed by estimating parameters such as pose, light direction and intensity, and color parameters. These parameters are estimated using an analysis-by-synthesis loop which seeks to minimize the distance to the original target image. Authors report a face swapping framework that is consistent across genders and a diverse set of face types. However, this approach and many similar ones that improve upon it [31, 32] require computation of 3D shape estimate of the face which could be computationally expensive. In later work by Bitouk *et al.* [21] authors eliminate the need for estimation of 3D geometry by ranking candidate source faces in their resemblance to the target face. The framework uses attributes such as pose, resolution, image blur, lighting and etc. to rank candidate replacement faces. Following this, the lighting and color properties of candidate's faces are adjusted to better blend into the target image. To adjust the candidate image to the target, they use quotient image formulation and alter low-frequency image content while high-frequency color information such as highlights and shadows remains

preserved. Although both previous works and many similar ones obtain adequate and visually convincing results, with the increased popularity of generative and encoder-decoder networks many recent approaches switch to learning-based solutions. Nirkin *et al.* [33] use a cascade of UNet and pix2pix structured subnetworks to first segment the source face and reenact it to match the position and pose of the target face and then fit and blend it to the target face. After initial detection and segmentation of the face region, the framework iteratively interpolates the source face to match the pose and position of the target face. To account for previously occluded regions of the now re-positioned face, the framework uses a face in-painting network while a blending network performs color adjustment to account for color and intensity differences.

Another family of face manipulation techniques aims to reenact or alter the existing face in the image or video rather than replace it. Such methods are often less likely to display artifacts such as illumination differences and blending inconsistencies and therefore can be more persuasive and dangerous adversarial tools. Much like face swapping methods, earlier face reanimation frameworks rely on signal processing and geometric computations. Bregler *et al.* [34] for example, propose a sequence manipulation framework, that position adjusts, and reorders video frames to manipulate the lip movements of the subject into creating an illusion of the subject uttering they never pronounced in the context of that video. The approach keeps subject-specific idiosyncrasies such as mimicking, particular eyebrow movements by phoneme learning. Authors apply the same logic to Visemes, which are visual counterparts of phonemes. Visemes are obtained from corresponding automatically segmented phonemes segments and later concatenated and blended into the background during synthesis. In another approach, Blaze *et al.* [20] model the face as a linear shape and texture combination of base faces. Authors control the likelihood of obtained face being a real face by computing the probability distribution of linear coefficients. Their method successfully models natural-looking faces and provides control over geometric and textural features of the face while preserving the realisticity of obtained faces. In an attempt to reduce the length of required training, in a transfer learning fashion,

Chan *et al.* [35] perform multidimensional morphable model transfer from a subject with a large video corpus to a subject with much smaller video corpus. The framework relies on initial facial correspondence matching by the user, during which the user places matching facial key points, then a flow matching algorithm selects images from the new corpus by minimizing the difference between optical flow of the original subject. Finally texture matching on the new corpus is performed to obtain final photo-realistic frame. Just as with swapping techniques, with the increased popularity of generative networks newer approaches have come to increasingly rely on neural network for image and video reanimation. In their end-to-end approach Xu *et al.* [36] leverage CycleGAN architecture to reanimate the original video to match the contex of target video.

Frameworks mentioned above and many others have sparked an unprecedented increase in the amount of manipulated content online. And although mostly innocuous, such methods present real threat to online security. Therefore, question of manipulated video detection is paramount to online security. In 2019 Facebook launched Deepfake Detection Challenge, along with a dataset of the same name [37]. The dataset contains 5214 samples of varying quality, background, race, lighting, etc. This further sparked active research interest in the detection of manipulated video content detection along with new datasets [38]. Deepfake detection literature is relatively young, with the first scientific publications emerging as recently as 2018. Initial frameworks closely followed photo and video face spoofing detection work [39, 40, 41, 42, 43, 44] by aiming to detect color and inconsistencies, temporal abnormalities, warping artifacts, and more.

Some of the approaches rely on the use of contextual information to detect object-level abnormalities in the video or image. Nirkin *et al.* [45] use identity level features from the swapped face and remaining regions of the head to deduct identity level discrepancies. To reveal such discrepancies author first separates the outer and inner regions of the face into two identity vectors. This is done by first detecting a larger bounding box containing the head region and then segmenting and separating the boundaries of the face. Segmented images are then trained for subject identification on two publicly available datasets. Authors then combine outputs from two networks to train a discrepancy detection network which uses

identity information obtained from each sub-network to decide if the two agree. Finally, the output of the discrepancy detection network is combined with the output of the generic face-swap detection network to produce the final output. Authors report state-of-the art performance on Celeb-DF [46] and FaceForensics++ [47] datasets. Some other works, utilize temporal contextual information instead [48, 49, 43]. Li *et al.* [43] utilize the fact many early deepfake generation methods focus on spatial realisticity of generated images but fail synthesize the temporally consistent behavior. More precisely the observe many deepfake videos lack natural human eyelid movements, with manipulated subjects blinking very rarely or not at all. Therefore to detect abnormal eye blinking behavior, authors first extract facial landmarks in each frame and normalize alignment and size of the face across frames. After face frames are aligned, eye region from each frame is cropped and fed to CNN network for extraction of object level features. Latly, these feature vectors are fed to the LSTM network for analysis of temporal behavior for the final prediction on the authenticity of the generated video. Authors find that the average blinking frequency for artificially generated videos is multiple times higher than that of original videos, and that setting a proper threshold value for frequency parameter, videos can be classified with state of the art accuracy using this method. Ciftci *et al.* [50] instead use an array of photoplethysmography signals extracted from cheeks and nose regions of the face. Authors argue that such signal do not preserve neither spatial nor temporal consistency in fake videos, and thus can be analyzed to detect deepfakes. Authors perform singal transformations to obtain a vector of features to feed to an SVM and CNN based classifiers.

Another family of solutions focuses on the analysis of micro-level temporal and spatial inconsistencies rather than analyzing global abnormalities. Guarnera *et al.* [51] employ convolutional traces pronounced in the local correlation of GAN-generated pixels to discriminate between natural and GAN-generated images. The authors use an expectation maximization algorithm over the image descriptors extracted from the images. The framework reports accuracy in the neighborhood of 90 percent on 6 different datasets. Zhao *et al.* [52] use a cascade of attention networks, textural feature enhancement, and feature aggravation to

reveal low-level spatial inconsistencies in the video frames of artificially generated videos. They argue that the discriminative inconsistencies between real and fake videos are better revealed at the local level and thus cannot be effectively captured by vanilla feed-forward networks. To resolve this discrepancy, at the first stage of the model, the authors generate several channels of attention maps that focus on separate regions of the face. Then to extract enhanced textural features, the authors utilize an enhancement block composed of local average pooling and densely connected convolutional block. Finally, Bilinear Attention Pooling is used to obtain the final feature map from texture and attention feature maps. Authors validate their approach on FaceForensics++, Celeb-DF, and DFDC datasets while reporting state-of-the-art performance. Hu *et al.* [53] try to estimate frame discrepancy by predicting future video frames using an autoregressive model and them comparing them with the actual frames from the video. Dang *et al.* [54] use plug-in attention network to a CNN backbone to attend to relevant regions of interest in the input frame and detect fakes. Beuve *et al.* [55] devise and test triplet loss with fixed positive and negative vectors tested with state-of-the-art convolutional backbone and attention block and report competitive performance on FaceForencis++ dataset. In [56] authors fuse frequency clues and convolutional RGB features to obtain a feature array representative of both domains. Frequency ques are obtained by first converting RGB space into YCbCr color space, then a Discrete Cosine Transform is performed on individual sections of the image, thirdly a new channels are formed from transformed sections of the image at a given frequency, and finally newly formed vectors of individual frequencies are concatenated together to form the final array. These transformed are then fed into a convolutional neural network and later fused with the features obtained from RGB network. Authors argue such processing enables the network to ignore the high-level semantic ques while preserving frequency information in the image, which positively contributes to the analysis of frequency-domain cues.

Several other approaches use frequency domain analysis to reveal artifacts resulting from uplconvolutional operations used by deepfake generation methods. Liu *et al.* [57] use changes resulting in the phase spectrum of up-sampled deepfake images to detect manipulated samples. The authors also note that low-level

texture information has more relevance to predictive outcome than higher level semantic information, and thus suppress higher-level information by contracting the convolutional neural network used in classification. Li *et al.* [56] on the other hand, use

2.2 Motion Magnification and Forensics

Although there is ample literature on selective signal amplification, the first attempt at a magnification of subtle motions in video recordings is proposed by Liu *et al.* [7]. The authors propose a user-aided motion tracking and amplification algorithm which performs semi-automatic clustering of pixels based on factors such as similarity of position, intensity, and motion characteristics. The aforementioned factors are evaluated over time and obtained cumulative similarity is then used to determine, and amplify selected clusters. Although reasonably effective cumulative grouping of pixels is time-intensive and requires long processing times. For this reason, in an attempt to arrive at a more efficient isolation and magnification of motions of interest, Wu *et al.* [9] propose to use temporal filtering of selected group of motions. The authors tackle this by first decomposing the video sequence into different spatial frequency bands, denoising, and then amplifying selected frequencies at any spatial location. The advantage of such an approach over [7] is that it does not rely on precise estimation and tracking of specific pixel motions and thus can yield more efficient processing. Following recent improvements in deep learning technology, Oh *et al.* [8] propose to utilize the power of convolutional networks for pixel-to-pixel motion magnification. Much like the approaches before it, they first extract valid spatiotemporal features and then amplify them by a selected amplification factor. Unlike the previous two approaches, authors rely on convolutional networks for deconstruction, spatial feature selection, and reconstruction of video frames. In addition, to train the network end-to-end they create a synthetic motion magnification dataset and utilize pixel difference as the estimate of relative motion.

Following significant improvement and increased availability of motion magnification routines, several approaches postulated that their use as a preprocessing step may help to reveal subtle inconsistencies in fake videos, thus aiding classification. Earliest such attempts are primary focused on the detection of face spoofing [10, 11, 58] while later transferring to deepfake detection. Fernandes *et al.* [59] propose to use motion magnified Neural Ordinary Differential Equations to estimate the heartbeat of the subject in the video with the goal of using inconsistencies in the heartbeat to estimate if the generated video is fake. In a similar approach, Qi *et al.* [12] perform motion magnification on non-overlapping regions of the face aided with the combination of temporal and spatial attention networks to reveal if the video in question is fake. A few other methods utilize motion magnification as a direct preprocessing step with the assertion that it has the potential to improve the accuracy of CNN-LSTM networks [60, 61]. Summary of existing works that utilize motion magnification for the detection of fake video content is outlined in Table 2.1.

One common feature among the methods described in Table 2.1 is the use of magnification as a preprocessing step. As per discussion in Section 1.1, we believe that such an approach is sub-optimal and an incremental and task-focused utilization of motion magnification-like units is required. To this end in this work, we focus on the experimental evaluation of magnification-like architectures in the task of motion magnification.

Table 2.1: Structural Summary of Existing Video Forensics Methods Utilizing Concept of Motion Magnification

Author	Overal Sctructure	Magnification Utilization	Description
Fei *et al.* [60]	CNN, LSTM	Prepossessing	*Use [8] to magnify video, extract spatial features with InceptionV3 [62] network, and feed the features to LSTM network.*
Das *et al.* [61]	SSIM, CNN, LSTM, HR	Prepossessing	*Use [9] to magnify input video, Then use frame similarity (SSIM), CNN-LSTM network, and heartbeat rhythms (HR) to make the final prediction.*
Qi *et al.* [12]	CNN, LSTM, Spatiotemporal attention	Preprocessing	*Use [9] to magnify video, separate frames into ROI and feed into temporal and spatial attention module, aggregate results with per-frame classification.*
Fernandes *et al.* [59]	Neural-ODE	Feature Extraction	*Use [9] as one of the feature extraction methods to feed to Neural Ordinary Differential Equations (Neural-ODE) model.*
Bharadwaj *et al.* [10]	LBP, HOOF	Preprocessing	*Use [9] to magnify input video, then extract Local Binary Pattenrs (LBP) and Histogram of Oriented Optical Flow (HOOF) for SVM and LDA classifiers.*
Raja *et al.* [11]	Fourier, CPI	Adapted Preprocessing	*Use phase adapted Eulerian magnification [9] on input video. Use magnified representation to compute and threshold cumulative phase information.*
Ge *et al.* [58]	CNN, LSTM, Temporal attention	Preprocessing	*Use [9] to magnify input videos, extract features using VGG-Face network, and feed it to LSTM unit enhanced by temporal attention.*

Chapter 3

Classification of Deepfake Videos

Here, we provide detailed information about the organization and inner workings of the proposed deepfake detection framework. We go over the structure and implementation of the motion magnification-inspired feature manipulation unit and its variations. The overall workflow of our framework is represented by Figure 3.1. As illustrated by Figure 3.1, frames are first aligned using facial landmarks, aligned frames are then segmented into sections of length K, passed through the CNN network for feature extraction, and then fed to the feature manipulation unit before making their way to the LSTM network. Predictions for individual segments are then aggregated using probability mean and majority aggregation methods. The backbone of the network consists of a small convolutional neural network meant for extraction of spatial features. In our approach, rather than aim to maximize accuracy on any given dataset, we aim to arrive at a magnification inspired feature manipulation architecture that would positively contribute to the prediction accuracy of the classification network. We compare the performance of baseline CNN-LSTM network to a feature manipulated CNN-LSTM architecture. We test several variations feature manipulation unit with features such as static multiplier, learned mask and Gaussian smoothed mask. We also perform multiple tests to verify the consistency in performance. The entire network is end-to-end trainable, while the aforementioned feature manipulation unit can be plugged in and out at the start of training.

The organization of our paper is as follows; in section Section 3.1 we describe the initial prepossessing of the input video stream where facial landmarks are used to isolate and normalize the face region in each frame. Later in Section 3.2 we briefly provide the details of the backbone convolutional network structure, along with the description of layer parameters. Finally, in section Section 3.3 we provide an overview of the temporal feature manipulation unit along with corresponding variations and modifications.

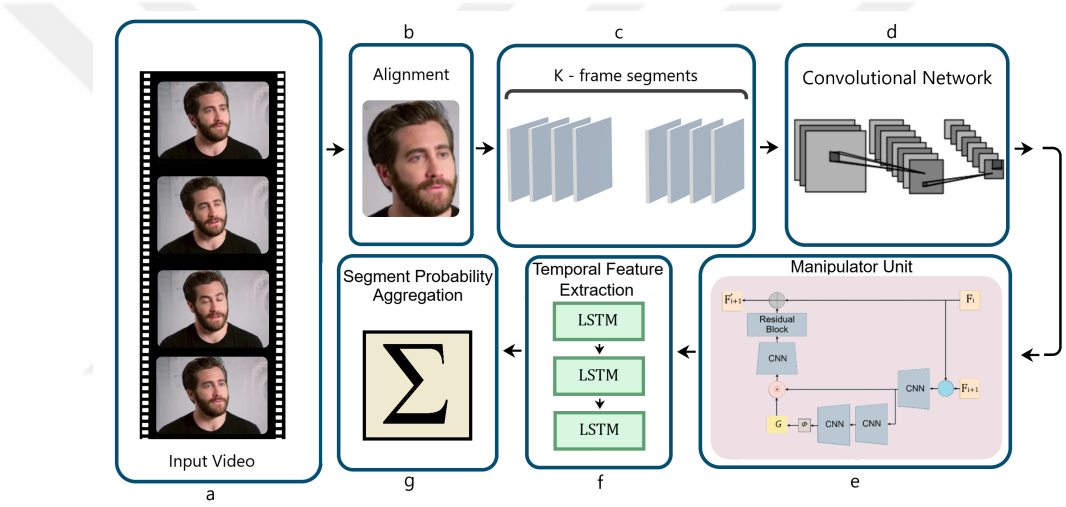

Figure 3.1: Overall workflow of the proposed framework. a) Frames are first aligned and cropped based on face landmarks. b) Aligned frames are segmented into sequences of equal length. c) Spatial features are extracted using CNN network. d) Magnification-Inspired Feature Manipulation. f) LSTM-DS network for final prediction of selected segments. g) Probability aggregation from individual segments.

3.1 Face Alignment

Contextual and diverse set of backgrounds, and large pose variations in the image complicate the task of deep neural networks and make it harder to learn latent representation crucial to detection of deepfake videos thus necessitating some of form of face normalization. However, due to the task at hand, the choice of face normalization is not as straightforward. That is, the objective of our network is to utilize spatio-temporal inconsistencies in the video for the classification of

deepfake videos. One possible option for normalizing faces in the frame and fitting them to a localized face model is face warping, however, such an approach is not ideal for our purposes since warping techniques inevitably introduce visual artifacts that are, in fact, very similar to those found in deepfake images. In fact, since many of the existing face swapping approaches use warping as a means of fitting the source face onto the target, the use of such preprocessing before detection might increase the similarity between two classes in the feature domain. For this reason, we instead opt for use of face rotation and centering to achieve a compromise between the preservation of original features and processing. More precisely we perform face cropping and alignment prior to feeding images to the model. We first extract facial landmarks using state of the art OpenFace landmark detection network [63, 64, 65] which, as illustrated in Figure 3.2, tracks a total of 68 facial landmarks across the boundary of the face, eyes, nose, and mouth.

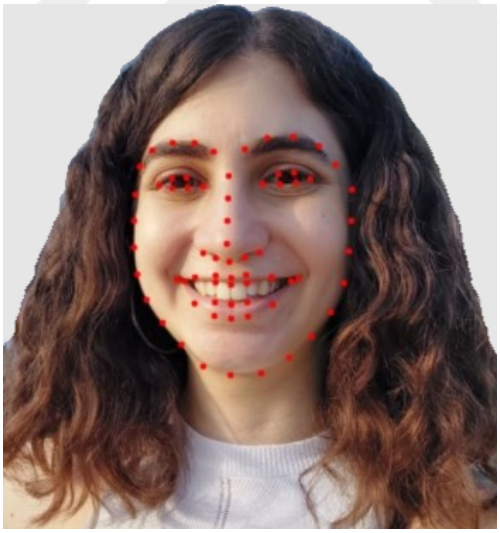

Figure 3.2: OpenFace landmarks.

Landmarks are then temporally smoothed using Running Median Smoothing algorithm. We then estimate the center of each eye by obtaining the mean of 6 respective eye landmarks for that eye. Two of the landmarks correspond to the right and left edges of the eye, and remaining four mark the points where pupil touches the eyelid. We then rotate images so that the line connecting centers is parallel to the horizontal axis. Finally we crop the images into squares such

that the midpoint of the horizontal line containing the center of the eyes of the subject is at the center of the image, and total distance between the eye centers is quarter of the image width. procedure is illustrated in figure Figure 3.3.

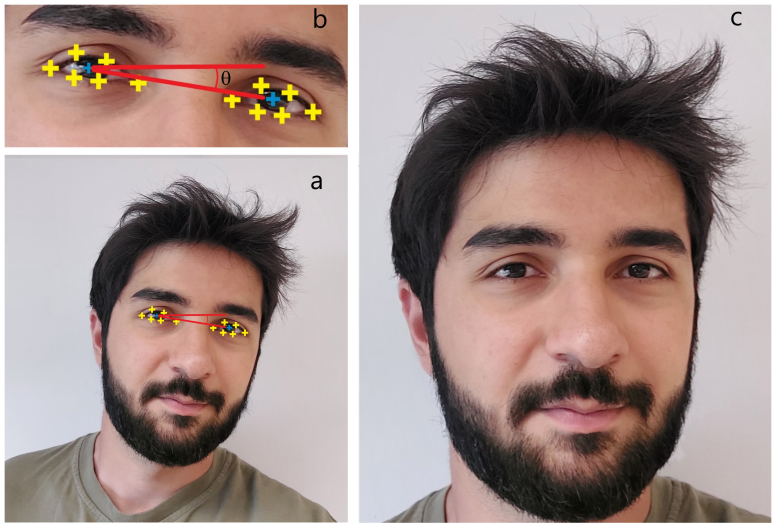

Figure 3.3: Overview of the face normalization routine. a) Detection of the center of the eye b) Computation of the angle with the horizontal axis c) Rotation, resizing and cropping of the final frame.

Finally images are resized to required lenght and width before being fed to the network. Additionally, input video streams are split into the segment of equal length, K, for training purposes, predictions for individual segments are later used for final classification which is described in Section 4.2.

3.2 Backbone Architecture

The main contribution of our work is centered around the magnification-inspired feature manipulation network which we test by plugging it in and out the main classification network. The baseline classification network consists of a small CNN comprised of three layers, two stacks of LSTM units, and a dense network for the final prediction. All the layers in the convolutional network are time-distributed, meaning that the K consecutive frames are fed to the network in sequence. This only changes when the feature manipulation unit is plugged into the network.

Since the feature manipulation network takes as input the difference between two features, the number of frames at the output is $K-1$, since the first frame has no reference frame. This relation and and simplified structure are illustrated in Figure 3.4. For clarity purposes the frames in the Figure 3.4 are illustrated as being fed to the network in parallel. In reality during training consecutive frames are fed in sequence and the gradients are computed based on the entire sequence.

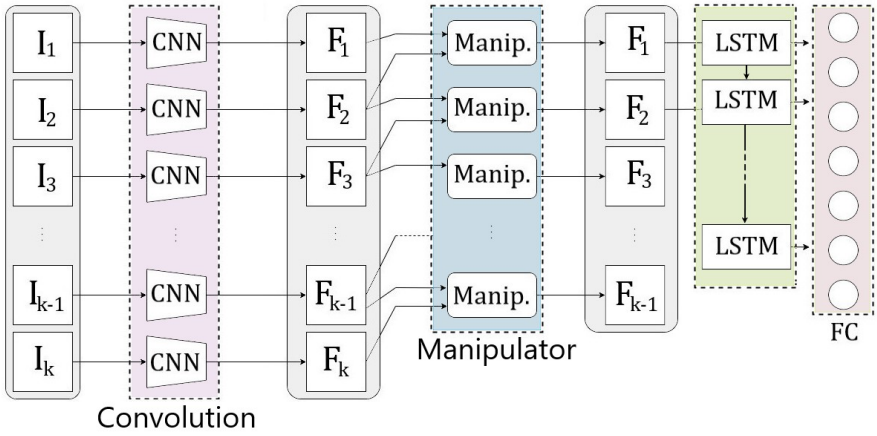

Figure 3.4: Simplified overview of the backbone architecture with the manipulator unit inserted in between.

3.3 Magnification-Inspired Attention Unit

In our work, we seek to utilize a magnification-inspired feature manipulator unit to enhance the prediction accuracy of the deepfake classification network. Proposed architecture borrows from existing motion magnification literature, and tries to re-target the task at hand to video classification. Unlike the original approach, we are not concerned with full-on motion magnification, and rather seek to utilize magnification-like feature manipulation architecture to obtain feature representations best suited for deepfake classification. Our approach borrows, and extends the feature manipulation unit proposed by Oh *et al.* [8], and sandwiches it between convolution and LSTM networks for spatiotemporal feature manipulation. As previously mentioned, instead of using magnification to preprocess the input videos we instead take an incremental approach in understanding the

contribution of learning-based magnification inspired architectures to the performance of the network.

The earliest attempt at motion magnification proposed by Liu *et al.* [7] relies on estimation and amplification of pixel-wise motions by tracking feature point trajectories. In light of the expensive computational cost associated with pixel-wise motion approximation, Wu *et al.* [9], instead, propose to use spatial decomposition, and temporal filtering to isolate the regions and frequency bands of interest for magnification. Oh *et al.* [8] extend this idea to learnable networks and propose to learn spatial features via convolutional networks; while modeling temporal manipulation as the distorted difference of consecutive frames.

By defining the video frame intensity $I(x, t)$ as both function of time t, and position x, and following the convention outlined by [8, 9], in its ideal form, the problem of motion magnification can be formulated by the Equation 3.1, where $\delta(\mathbf{x}, t)$ represent the displacement function, α represents the magnification factor, and $\tilde{I}(\mathbf{x}, t)$ represents the resulting magnified frame of interest.

$$\tilde{I}(\mathbf{x}, t) = f(\mathbf{x} + (1 + \alpha)\delta(\mathbf{x}, t)) \tag{3.1}$$

In Equation 3.1, above, the entire displacement function is magnified uniformly with no feature selection. For a targeted task at hand, however, we are more concerned with amplifying designated regions relevant to the learning task. To isolate translations in these regions, motion selector $\mathcal{J}(\cdot)$ is introduced as displayed by Equation 3.2.

$$\tilde{I}(\mathbf{x}, t) = f(\mathbf{x} + (1 + \alpha)\mathcal{J}(\delta(\mathbf{x}, t)) \tag{3.2}$$

Introduction of feature selector $\mathcal{J}(\cdot)$ into the equation, ensures that only relevant regions of interest are amplified, which finalizes the most basic formulation of selected motion magnification.

In practice, however, it is challenging for a single bank of filters, or a network, to obtain or learn feature representations diverse and powerful enough to reconstruct an accurate representation of motion with this basic relation. Thus, additional segmentation of the Equation 3.2, and further approximations are required. Effectively, when the ultimate task is photo-realistic but manipulated motion reconstruction, it is more feasible to perform displacement manipulation on the feature level rather than on the image level. This transfers the task of accurate reconstruction from feature selector network $J(\cdot)$ and delegates it further down the line to the reconstructive network. Such configuration is also advantageous for our problem since tasking the network with proper reconstruction is outside the scope of this paper and introduces redundant learning loads. For this reason it makes sense to perform motion manipulation on feature lever rather than image level. Thus, by annotating the feature vector corresponding to the frame I as F, we can rewrite the Equation 3.2 for feature manipulation in Equation 3.3.

$$\tilde{F}(\mathbf{x}, t) = f(\mathbf{x} + (1 + \alpha)\mathcal{J}(\delta_F(\mathbf{x}, t))) \tag{3.3}$$

Here in Equation 3.3, α, and $\mathcal{J}(\cdot)$, similarly, represent magnification factor and feature selector while $\delta_F(\mathbf{x}, t)$ represents the displacement for the feature vector F. Utilization of feature manipulation, is also displayed by Figure 3.4 where features obtained by deep convolutional neural network are fed sequentially to the feature manipulator unit.

Another computation that exerts a significant load on the learning network is the estimation of the displacement vector $\delta_F(\mathbf{x}, t)$, which requires isolation of object boundaries and tracking their motion. On-the-run estimation of the displacement vector is rather costly and prevents the network from running end-to-end. Because of this, [8] instead opt for the difference of the consecutive feature vectors as an approximation of displacement. This relation is expressed by Equation 3.4, where F_{i+1}, and F_i are consecutive feature vectors and Δ_{i+1} is their difference.

$$\Delta_{i+1} = (F_{i+1} - F_i) \tag{3.4}$$

Similar to how the original displacement features are processed by $J(\cdot)$ in equation 3.3, difference features are then passed through a selector network $\mathcal{L}(\cdot)$. The function of $\mathcal{L}(\cdot)$ within the Equation 3.4 is to further isolate relevant motions for the prediction just as $\mathcal{J}(\cdot)$ would in the original equation. Resulting selected displacement approximation then can be expressed by Equation 3.5, where D denotes the selected features of interest.

$$D_{i+1} = \mathcal{L}(F_{i+1} - F_i) \tag{3.5}$$

Utilizing relation defined by Equation 3.5 we can rewrite Equation 3.3 as selective amplification of difference vector defined by Equation 3.6, where F, and $F^{'}$ denote the original and manipulated feature frames respectively.

$$F^{'}_{i+1} = F_i + (1 + \alpha)D_{i+1} \tag{3.6}$$

To add further non-linearity to the amplified feature vector, convolutional network $\mathcal{H}(\cdot)$ is added to the manipulator unit, resulting in Equation 3.7.

$$F^{'}_{i+1} = F_i + \mathcal{H}((1 + \alpha)D_{i+1}) \tag{3.7}$$

Configuration expressed in Equation 3.7 closely follow the manipulator architecture proposed by [8] and is illustrated in Figure 3.5.

In this configuration the section b represents the residual layer that is used in the final processing of the feature vector before addition to the original feature vector. As expressed by Equation 3.7 and illustrated by Figure 3.5, amplification factor α is multiplied by all elements of the difference vector D which results in

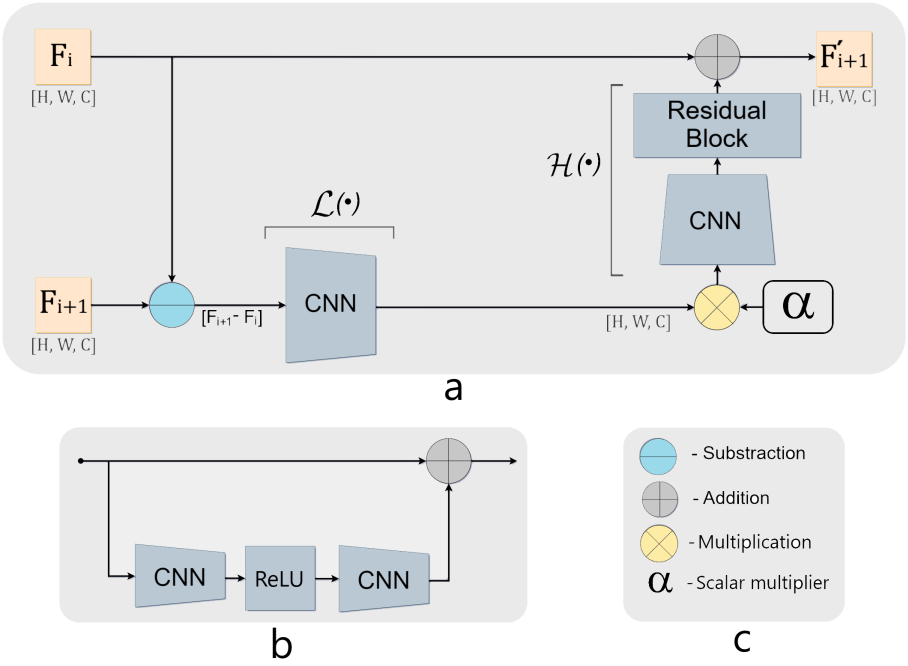

Figure 3.5: General architecture of manipulator unit with static multiplication factor a) Overall layout b) Residual block c) Operands and operations.

uniform amplification of the vector across the spatial domain. In practice however, it could be desirable to use different multiplication factor in different sections of the displacement vector. In addition, with the original configuration, amplification factor is a constant and cannot be modified individually for each video, which complicates the optimization of the alpha value. Previous approaches utilizing motion magnification as prepossessing step, thus perform a series of full data experiment to arrive at optimal alpha value that maximizes total accuracy. It is clear such optimization is expensive in terms of cost and therefore dynamic optimization of multiplication factor is desirable. In addition, when the task at hand is manipulation of feature vector for the task of classification, in theory, it could be desirable to amplify some regions of the feature vector while dampening the others. To this end, diverging from the original manipulation unit, we propose to use point-wise, learnable, and dynamic multiplication-dampening array $\mathcal{A}(\cdot)$ in our implementation. Modified version of the resulting feature manipulation network is expressed by Equation 3.8, where $\mathcal{A}(\cdot)$ is a dynamic multiplication array function that takes as an input processed difference vector D and produces

multiplication factors for each pixel of the feature vector.

$$F'_{i+1} = F_i + \mathcal{H}(\mathcal{A}(D_{i+1}) \odot D_{i+1}) \tag{3.8}$$

This introduces additional flexibility to the network allowing it to discriminate spatially between displacement vector regions. In our experiments we compare the performance of multiplication mask with that of static multiplication factor. The architecture of obtained feature manipulation unit is described in Figure 3.6 where, similarly, section *b* represents residual processing of the final feature vector, and *c* contains denominations of individual functions.

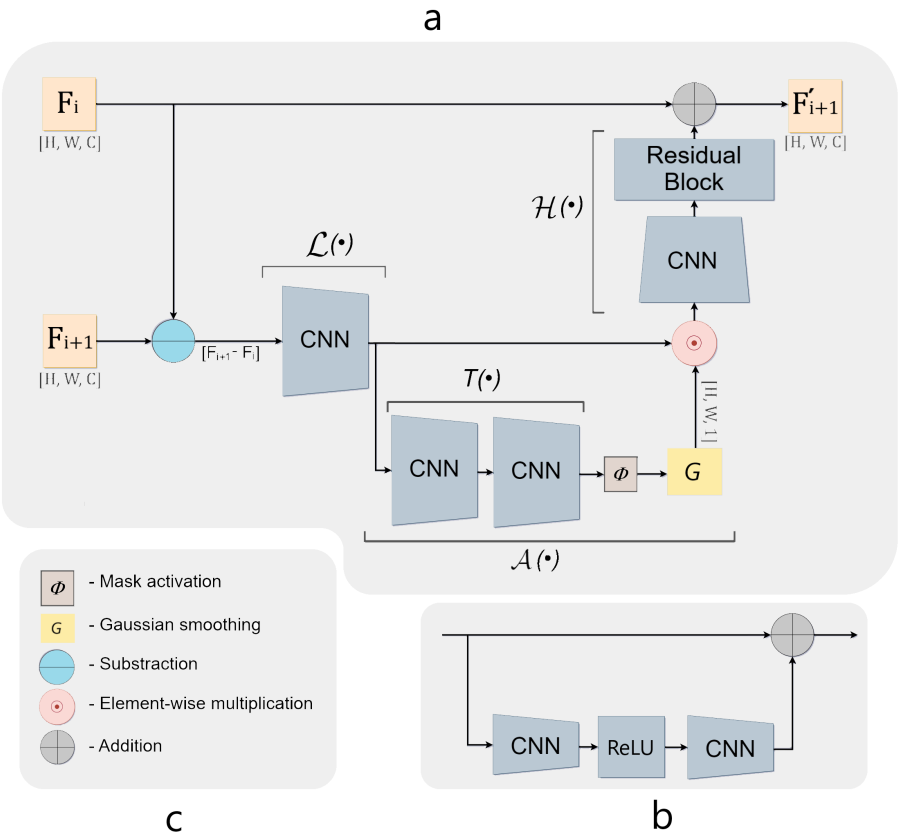

Figure 3.6: General architecture of manipulator unit with dynamic multiplication factor a) Overall layout b) Residual block c) Operands and operations.

It can be observed by comparing figures 3.6 and 3.5 that the dynamic

magnification-inspired feature manipulation unit has a lower convolutional section $A(\cdot)$ that is used to obtain dynamic multiplication mask that changes based on the input and corresponding difference vector. This enables us to have a more robust estimation of amplification factors and to distribute them based on input pixels. As previously indicated, by replacing static multiplication with dynamic multiplication vector we are allowing the feature manipulation unit to selectively dampen or amplify different regions of the feature vector. This is achieved by enforcing the mean of multiplication unit to unity by a custom-designed layer activation. Proposed feature multiplication function $\mathcal{A}(\cdot)$ work similar to spatial attention units. Firs the processed difference vector is passed through a series of convolutional layers. This is done to make the model dynamic, and vary the multiplication mask based on the input. Later the model is passed through mean enforcement function which as the name suggests enforces the mean of the multiplication array to unity. To achieve this, we perform Softmax-like activation, where first each element is exponentiated and then divided by total sum. This forces all the elements in the array to sum up to one much like probability vector. We then multiply the resulting array by its area to ensure that the average of all values in the array is one. This forces some sections of the mask to be attenuated while others are dampened. Finally the output of unity enforcement function is passed to spatial Gaussian smoothing to enable smooth spatial variability. Resulting relation is described by Equation 3.9, where, $T(\cdot)$ denotes the convolutional processing layers, $\Phi(\cdot)$ denotes unity enforcement function, and $G(\cdot)$ denotes the Gaussian smoothing.

$$\mathcal{A}(D_{i+1}) = \mathcal{G}(\Phi(T(D_{i+1})) \tag{3.9}$$

Detailed detailed description of the activation function described in Equation 3.10, where $\mathbf{X}$ denotes the input vector, H_x and W_x denote the dimensions of the input vector $\mathbf{X}$, $a_{i,j}$ denotes an individual element of $\mathbf{X}$ at location $[i, j]$, and $exp(\cdot)$ denotes the point-wise exponential function.

$$\Phi(\mathbf{X}) = \frac{exp(\mathbf{X})H_X W_X}{\sum_{i=1}^{H_X}\sum_{j=1}^{W_X} e^{a_{i,j}}} \tag{3.10}$$

Activation function described in Equation 3.10 ensures that only relevant sections of the difference feature vector are amplified while the ones that are irrelevant to the task are dampened. Contrary to the static multiplication his introduces additional flexibility to the network, since, as described above, when the task at had is classification rather than magnification, we are more concerned with isolating relevant features rather than amplifying the entire array. In this sense, the activation function $\Phi(\mathbf{X})$ performs as probability distribution of feature relevance and helps to amplify relevant ones.

Chapter 4

Experiments and Results

In the previous section, we described our method that utilizes an improved version of the motion magnification-inspired feature manipulator unit. The detection method proposed in this work aims to take the initial steps at understanding the applicability of learned motion manipulation in classification tasks. Rather than aim at maximizing performance on a given dataset we instead focus on exploring the types of feature manipulation architectures that constitute an improvement in the performance of the model. To this end, we perform a series of experiments with a standard baseline, and variations of feature manipulated architecture. The layout of this section is as follows; we first describe the dataset used for experimental evaluations, we then reveal prepossessing and network parameters, and we compare the performance of the model in the absence of feature manipulation, with the static multiplier, with a dynamic multiplier, and a Gaussian smoothed dynamic multiplier.

4.1 Dataset

For our experiments we utilize Celeb-DF deepfake detection dataset [66]. The dataset is comprised of 5,639 deepfake videos, sources for which serve 590 real

videos from 59 subjects. Resulting dataset has a total of 2 million fake frames. The dataset also contains an additional 300 real YouTube videos which are also used in training and prediction. Subject in the videos are sampled from a varied set of genders, race, and ages with 56.8% being male and 43.2 being female. Racial diversity of the dataset is less varied with 5.1% of the subjects being Asian, 6.8% being African American and remaining 88.1% being White. An example sample of subjects from the dataset is presented in the Figure 4.1.

Figure 4.1: Synthesized frame samples from the original dataset.

The videos have an average frame rate of 30 frames per second and an average length of 13 seconds. All real videos are obtained from YouTube and later manipulated by a modified version of DeepFake [67] face swapping method. Authors of the dataset perform several improvements to the original manipulation methods to obtain more realistic and consistent results. They first increase the resolution of the output from 64 by 64 pixels to 256 by 256 pixels by increasing the size and resolution of encoder-decoder filters. Authors also add color perturbations to the input during training to force the network to learn more color-consistent output, thus improving color and intensity uniformity between source and target faces. To improve the alignment of the source face onto the target, the authors also improve the estimation of face boundaries by increasing the number of points around the contours. Lastly, the authors use temporal Kalman filtering to introduce better temporal consistency in the output frames.

4.2 Framework Parameters

As described in Section 3.1, for training, we split the sequence of frames into non-overlapping sections of uniform length. During experiments, we set this constant

K at 32, with an image resolution of 256 by 256 pixels. Both parameters are picked as a compromise between computational efficiency and input resolution and are determined as a result of early experimentation. All of the experiments are carried out with `GeForce GTX 1080 Ti` and `GeForce RTX 2080 Ti` graphic processing units and `Tensorflow 2.3.0` deep learning library.

The overall layout of the network is illustrated in Figure 3.4 with a more detailed description of the parameters described in Table 4.1.

Table 4.1: Layer Architecture of the Proposed Framework

Layer	Sequence	Filters or Units	Activation	Kernel
Convolution	Yes	64	LeakReLU	3x3
Max Pool2D	Yes	-	Linear	2x2
Convolution	Yes	32	LeakReLU	3x3
Max Pool2D	Yes	-	Linear	2x2
Convolution	Yes	16	LeakReLU	3x3
Max Pool2D	Yes	-	Linear	2x2
Manipulator	Yes	-	-	-
Flatten	-	-	-	-
LSTM	Yes	256	Tanh	-
LSTM	No	256	Tanh	-
Dense	No	2	Linear	-

The entire Convolutional network that is illustrated in Figure 3.4 is time distributed. This means that each individual frame is fed sequentially to the convolutional neural network to extract spatial feature representations. The convolutional network is made up of three convolutional blocks. Each convolutional block is made up of a convolutional layer, followed by a 2D Max Pooling layer, a Leaky ReLU activation, and Batch Normalization followed by a Dropout layer. The Alpha parameter for the Leaky ReLU function is set at 0.4 while the dropout is fixed at 0.2. Kernel size for convolutional layers is 3 by 3 while the pool size for the pooling layer is 2 by 2. We use Gloroth Uniform Initializer for all of our convolutional layers on the backbone network and L2 regularization for LSTM and dense layers.

Layer parameters of the dynamic and Gaussian smoothed manipulator unit are outlined in Figure 3.6 with layer parameters outlined in Table 4.2.

Table 4.2: Layer Architecture of the Dynamic Manipulator Unit

Block	Layer Type	Filters	Activation	Kernel
$\mathcal{L}(\cdot)$	Conv2D	16	ReLU	3x3
$\mathcal{A}(\cdot)$	Conv2D	4	ReLU	3x3
	Conv2D	1	$\mathcal{G}(\Phi(\cdot))$	3x3
$\mathcal{H}(\cdot)$	Conv2D	16	Linear	3x3
	Conv2D	16	ReLU	3x3
	Conv2D	16	Linear	1x1

As can be observed from Table 4.2, we do not alter the number of filters of the convolutional layers of the manipulator unit and perform zero padding to preserve the original spatial dimensions. This results in the input and output dimensions of the unit being equal. This enables us to seamlessly integrate the manipulator unit after any of the convolutional layers without disturbing the original configuration of the baseline network. Implementation details of the feature manipulation unit with a static multiplication factor, illustrated in Figure 3.5, are identical to that of the dynamic feature manipulator. Layer parameters of the feature manipulation unit with static multiplier are also described in Table 4.2, with the only difference being the absence of the sub-unit $A(\cdot)$ in the second and third rows. The Gaussian filter $\mathcal{G}(\cdot)$ in block $\mathcal{A}(\cdot)$ of the dynamic feature manipulation unit has a kernel size of 3x3 and standard deviation of 1.

To evaluate the contribution of the manipulator unit to the prediction accuracy of the network we train the network without the manipulator unit. During these experiments, we keep the remaining training parameters unaltered, while only plugging and unplugging the manipulator unit. During training, we utilize Stochastic Gradient Descent with a learning rate of 0.0001 as an optimization function and Categorical Cross Entropy as a loss function, while randomly shuffling the samples at the start of each epoch. We use early stopping conditions for our training loop. During training, we conduct validation checks at the uniform intervals of 2000 samples and stop the training if the accuracy of the network does not increase for more than 5 intervals. We then record and report the model

with highest validation accuracy. However, for computational limit reasons we only perform validation checks after the first epoch. We perform a total of sever experiments with each configuration and report the average results.

As previously mentioned we subdivide the input video recordings into the segments of equal size. Later to obtain results on individual video recordings we utilize two distinct probability aggregation methods. The two probability aggregation methods are majority voting and total mean. The former, as the name suggests, obtains the final prediction for the recording based on the prediction from the majority of the samples. Equation 4.1 displays this relation.

$$\begin{aligned} P_v &= \frac{1}{|S_v|} \sum_{i \in S_v} d_i, \text{ where} \\ d_i &= \begin{cases} 1, & \text{if } P_{v,i} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4.1}$$

Here, P_v denotes the probability of the entire video recording, S_v denotes the subject of all video samples for the video v, and $P_{v,i}$ is the probability of an individual segment. The second method of final probability accumulation is average of total probability. This relation is expressed by Equation 4.2, and computes the probability of the final video being fake as the probability mean of individual segments of the video.

$$P_v = \frac{1}{|S_v|} \sum_{i \in S_v} P_{v,i} \tag{4.2}$$

Similarly, in Equaiton 4.2, P_v denotes the probability of the entire video, while S_v denotes the subset of all segments for that video.

4.3 Effect of Manipulator Unit

In the first set of experiments, we evaluate the contribution of the magnification-inspired feature manipulation unit to the prediction accuracy of the network. To compare the optimal contribution from the manipulation unit we use dynamic, and Gaussian smoothed configuration that is illustrated in Figure 3.6. As previously discussed, with this configuration multiplication mask is computed dynamically for each individual input while a unity enforcing function introduces both amplification and dampening to the mask. Finally, for spatial consistency, a Gaussian smoothing function is applied before actual multiplication. During the experiment, we first train and test the network without feature manipulation to obtain a benchmark on the performance. We then plug in the manipulator unit and repeat the same process to measure the contribution to the final accuracy. Each individual run is repeated seven times and average results are reported for individual segments of length 32 and entire videos. All the accuracy results are reported in Table 4.3.

Table 4.3: Effect of Gaussian Smoothed Dynamic Manipulator Unit on Accuracy

Manipulation	Aggregation	Train	Validation	Test
No	No aggr.	0.84	0.80	0.81
Yes	No aggr.	0.88	0.85	0.83
No	Voting	0.85	0.82	0.82
Yes	Voting	**0.91**	**0.89**	0.85
No	Mean	0.86	0.84	0.83
Yes	Mean	**0.91**	0.88	**0.86**

From the Table 4.3 we can observer that across results on individual segments and two probability aggregation methods results with feature manipulation unit display higher accuracy. We can also observe that in general results for two probability aggregation methods seem to agree on the final accuracy, with voting method displaying higher accuracy on validation set and mean of probability displaying higher accuracy with the test set. However, with both, the difference is very small and marginal. Results of this section display that, if properly configured, feature manipulation inspired from motion magnification literature can

yield performance improvements on general classification tasks. The high modularity of the feature manipulator unit allows us to conduct robust experiments and plug it in and out of the network as any other layer architecture, which gives us the chance to compare several configurations of the network. While in this section we primarily focus on the performance comparisons of the optimally performing, dynamic, and Gaussian smoothed feature manipulation unit to the baseline, in the following sections we also investigate the effects of non-dynamic, static multiplication and Gaussian smoothing individually.

4.4 Effect of Scalar Multiplication

The original feature manipulation unit proposed by Oh *et al.* [8], also illustrated in Figure 3.5, relies on the use of predefined magnification factor α. During our experiments, we determined that such a configuration is not ideal for our purposes since it is clear that feature selection by the preceding convolutional layers is not quite sufficient to obtain the best spatial representations of the features to be amplified. It could be argued that because our ultimate goal is not magnification, and subsequent reconstruction of selected features, our approach does not have to rely on amplification only. That is, much like with spatial attention networks it is of interest to amplify some regions of the feature map while dampening the others. In addition, while in the context of motion magnification the constant alpha serves the purpose of amplifying selected motions of interest by a given factor, thus serving as s controlling parameter of the output video, no such immediate function is required for the task of deepfake detection. For this reason, a constant value in the context of deepfake classification can only serve as an additional complexity in the network. Furthermore, optimization of such a constant would require brute-force search and be network-specific at best. To demonstrate this point we perform experimental evaluations of the results with static multiplication and report them in Table 4.4.

Table 4.4: Accuracy of Dynamic and Static Manipulator Units and α set at 2

Manipulation	Aggregation	Train	Validation	Test
Static	No aggr.	0.85	0.82	0.72
Dynamic	No aggr.	0.88	0.85	0.83
Static	Voting	0.88	0.86	0.75
Dynamic	Voting	0.91	0.89	0.85
Static	Mean	0.88	0.85	0.75
Dynamic	Mean	0.91	0.88	0.86

From the Table 4.4 we can observe that the reported average accuracy for the dynamic feature manipulator unit is much higher to that of static multiplier. Here as a static value we use 2 which is often used as the default magnification factor in motion magnification literature. As with previous experiments we perform a total of seven experiments with each configuration and report the average results. Higher accuracy achieved by the dynamic unit can be attributed to more complex feature selectivity and attention-like amplification and dampening. Inclusion of the a learnable and frame specific multiplication mask enables the networks to suppress the features that are common to the feature space of both, deepfake and real videos and that don't contribute the classification accuracy. When comparing the Table 4.3 and Table 4.4 we can see, in fact, that result with static multiplication factor are not significantly better than those of baseline network and which fails to justify added complexity and proves superiority of dynamic multiplier.

4.5 Effect of Gaussian Smoothing

In previous sections we discussed the use of dynamic multiplication mask and its effectiveness compared to static multiplication factor α. The optimal feature manipulation unit proposed in this research therefore is a cross between

magnification-inspired temporal feature manipulator and spatial attention. Following best practices in existing literature on spatial attention networks we apply Gaussian smoothing to our multiplication mask prior to multiplication with the feature map. This has the potential of achieving spatial consistency and smooth variability across the obtained mask thus simplifying the learning process. In addition, frame preprocessing and face alignment performed by this by our preprocessing algorithm means that face parts in the frame, although centered in the eye region move spatially depending on the initial pose of the subject. Smoother spatial variability, therefore, has the potential of simplifying the learning of such variations by the network and prevent overfitting to specific variations. To verify our initial assumption we perform experiment to measure quantitative contribution of the smoothing. Similar to all previous configuration a total of seven experiments are carried out with each configuration and average results from those runs are reported. Obtained results are presented in Table 4.5.

Table 4.5: Effect of Gaussian Smoothing on Accuracy

Manipulation	Aggregation	Train	Validation	Test
No Gauss.	No aggr.	0.85	0.82	0.80
Gaussian	No aggr.	0.88	0.85	0.83
No Gauss.	Voting	0.86	0.84	0.82
Gaussian	Voting	0.91	0.89	0.85
No Gauss.	Mean	0.87	0.85	0.82
Gaussian	Mean	0.91	0.88	0.86

We can observe from the results presented in Table 4.5 that in-fact Gaussian smoothing has a very significant contribution to the final accuracy of the network. Accuracy results in the absence of Gaussian smoothing, almost resemble those obtained with a static multiplication as presented in Table 4.4 factor confirming that spatial consistency plays a significant role in the isolation of relevant and irrelevant motions of interest. In is also a contributing factor in preventing the network to overfitting to specific to specific pixels and instead focusing on relevant regions of interest, which in turn, contributes to final accuracy.

4.6 Visualization of Multiplication Mask

By replacing the static multiplication factor with a dynamic mask we are enabling the network to attend differently to various sections of the image. Although we observe from the table 4.3 that dynamic mask indeed significantly contributes to classification accuracy it is also crucial to have a visual interpretation of such results. Essentially, in this section we aim to provide visual representation of the obtained magnification mask and interpret its regions of attenuation.

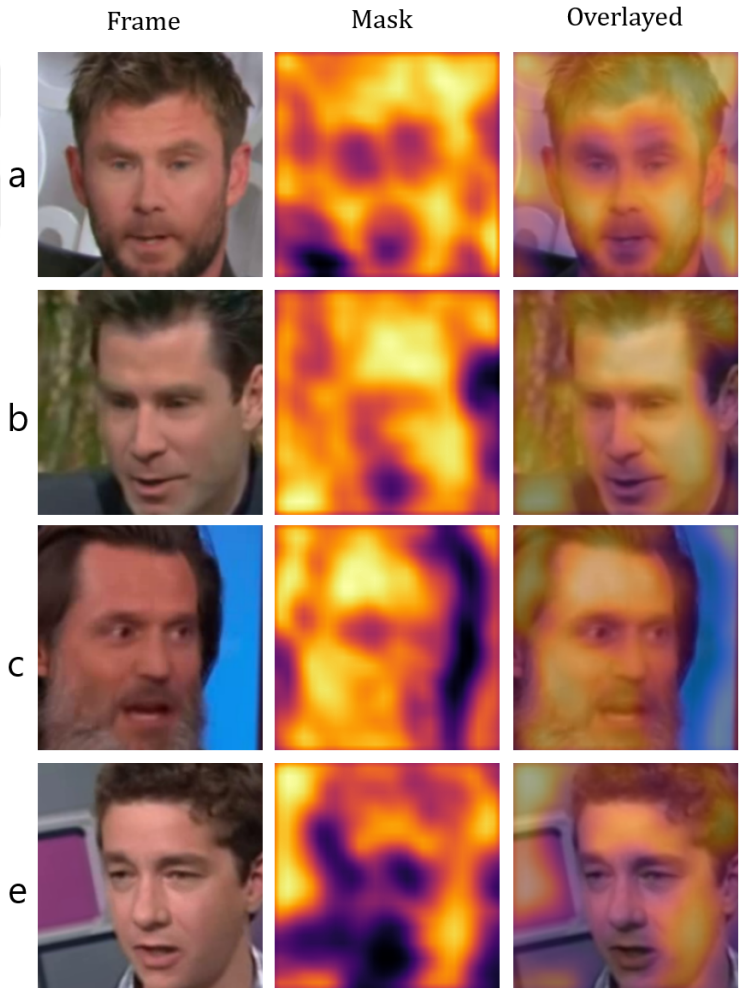

Figure 4.2: Values of the multiplication mask $\mathcal{A}(\cdot)$ represented along with frames that generate them. Original frame is displayed on the left, multiplication mask in the middle and overlayed picture of the two is to the right.

Our initial expectation would be to observe the regions containing warping

artifacts and transfer seams to see the most attenuation. In our task, such regions coincide with the boundaries of the face and the lower hairline. Attenuation of such regions would mean that most relevant artifacts occur at the boundaries where source face meets the target face. Such a result would be expected for several reasons; due to the nature of DeepFake generation method, the estimation of the face mask around the target and source face is inherently imperfect resulting in inconsistent seams and boundaries. In addition, there is often a color mismatch between source and target faces which results in color artifacts around the seams. Lastly, estimation of face landmarks is highly irregular and even in the presence of temporal filtering yields rippling. All of these factors, although addressed to a degree by the generation method, result in color, shape and temporal artifacts around the bounds of the merged face. Thus in our visual observations to expect the model to focus on these sections of the frame.

To observe and interpret the results, we plot the sample input frames from the dataset along with up-scaled multiplication mask, and the image overlayed with the mask, in Figure 4.2, where brighter colors represent higher intensities. As can be observed from the Figure 4.2, per initial expectation, the network indeed focuses on boundaries of the face which are expressed with a brighter color in the mid column. An interesting observation is that the eye regions and the mouth regions are less attenuated, this can be explained by the fact that these regions usually lie inside of the boundaries that represent the seams of the transferred face. An important observation to make is that not all multiplication masks display equal visual interpretability. Boundaries of different face regions of the subject (a) in Figure 4.2 are more clearly defined those of the subject (b). In Figure 4.3 we also plot a close-up shot of the multiplication mask. As can be observed from the figure values range somewhere in between 0.2 and 1.6 proving that indeed both dampening and amplification is performed by the mask. Observation of the multiplication mask provides us with an intuitive understanding of the processed involved in the magnification-like architecture. The network selectively amplifies sections of the difference vector most relevant to our predictive outcome. In that sense the magnification mask acts as a spatial attention network expanding the range and variability of the network. This is in stark contrast to multiplication

with a static constant which is rigid and does not vary between samples. This rigidity, as observed by the results leads to lower predictive accuracy.

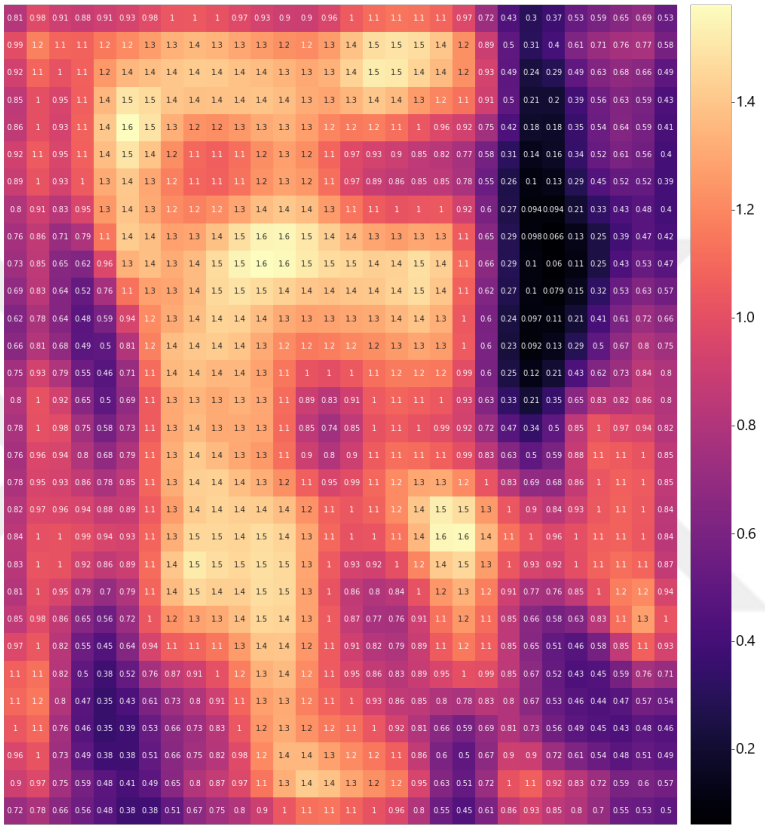

Figure 4.3: Close-up visualization of the multiplication mask ($\mathcal{A}(\cdot)$) values with rounded quantities printed on the pixels and a heatmap to the right. Values can be observed fluctuating between 0 to 1.6, which ensures both amplification and dampening.

4.7 Discussion of Preliminary Observations

Before settling on the final architecture of the network, several preliminary experiments are carried out. These experiments have influenced further architectural details of the detection framework and formed the foundations of initial decision-making. To understand the effects of different types of face normalization on the performance of the network, we have constructed a simple image classification using an Imagenet pre-trained convolutional network, and trained it on frames

extracted from the videos of the Celeb-DF [46] dataset. In addition visual patters in these videos have also been analyzed.

We have tried three forms of preprocessing in initial experiments; In the first form of preprocessing faces are simply cropped using OpenFace [63] face detection library. For the second preprocessing, face landmarks are detected, 2D mesh is estimated, and the face is warped to a pre-defined normalized face mask. Finally, in the third type of processing, which is also described in Section 3.1, images are rotated, resized and cropped such that the eyes of the subject are always at the center of the image. In these experiments we have observed the following; Although the use of standard cropping eliminates a good chunk of surrounding scenery from the image, a lot of facial movement remains. This means that at any frame transition the face in the video fluctuates significantly in the 2D plane and is not centered at any one location. This can be explained by the imperfections in face landmark detection but also by the innate instability in the type of processing utilized. Additionally, we have also observed that the network struggles significantly with learning meaningful representations and obtaining statistically significant accuracy. To account for large fluctuations of the face in the 2D domain, in the second experiments, we resort to the use of face warping as processing routine. Based on visual observations, this almost completely eliminates spatial fluctuations of the face and ensures that all the individual face regions are at predefined locations of the image. However, this type of preprocessing introduces a problem of warping artifacts. A lot of face swapping frameworks use some of form of warping during the face transfer from the source to target. Such artifacts prove useful features in the detection of fakes later on during training. However, by introducing warping into the preprocessing routine, both, for real and fake samples, we suppress such artifacts as discriminant features. Warping also removes sections of the face beyond the seams of the inner face region which coincides with the seams of face swap where a lot of spatiotemporal artifacts manifest themselves. As a result, the ability to visually differentiate between fake and real videos is also reduced. To find a compromise between two approaches we instead opt to use a less aggressive normalization method that is discussed in Section 3.1. With the third method, to minimize the motion of the face in the 2D axis

we detect and center the eyes of the subject at two pre-defined locations in the image, in parallel to the horizontal axis. If the eyes of the subject are at an angle to the horizontal axis, the image is first rotated and only then cropped. This does not fully eliminate 2D fluctuations, however it ensures that the input face is more localized than standard cropping, while warping artifacts are not introduced and the seams of the inner face are preserved. Indeed, initial experiments shown better visual interpretability and higher classification accuracy.

Another initial challenge in the research has been related to design characteristics of the magnification inspired feature manipulation unit and improvements made upon it. Due to convolutional building blocks of the magnification inspired feature manipulation unit described in in Figure 3.5, the number of parameters increases exponentially with the the dimensions of the input. To avoid the curse of dimensionality, earlier on we have experimented with the use of separated convolutions, however, with such configurations the feature manipulation unit has been struggling to learn meaningful representations of the input and the weights of the unit were changing only slightly. By recording the gradients of the network at each iteration, the problem of dying gradients have been observed. For this reason, it has been decided to bring back conventional convolutions and instead reduce the channel dimensions of the input to the manipulation unit and add pooling layers in between convolutional layers of the network.

One significant modification introduced to the magnification inspired feature manipulation unit is the use of dynamic multiplication mask that both amplifies and dampens selected features. Activation function $\Phi(\cdot)$ illustrated in Figure 3.6 and expressed by Equation 3.10 has been carefully designed to amplify relevant sections of the frame while dampening the others. In earlier forms of the network, the values of the mask have been fluctuating in the free range without any restriction of the values. However, we have observed that, often times, the values of the mask converge to values that are very close to zero. And this in turn results in dampening of the entire feature map. This phenomenon is illustrated by the Figure 4.4.

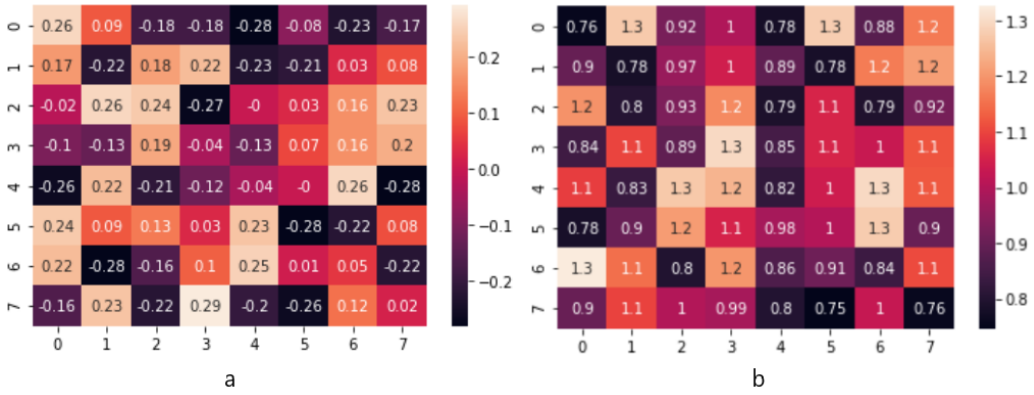

Figure 4.4: Comparison of the mask values with and without activation function $\Phi(\cdot)$. It can be observed from the image that the values without activation (a) converge to subzero values thus dampening the entire array, while (b) activation function helps to restore the values to a desired range.

Early experimentation on the type of input processing and feature manipulation techniques leads to more robust architecture that is able to learn valid representations of the input image and amplify and dampen relevant transitional features in the image.

Chapter 5

Conclusion

In our research we have investigated the use of magnification inspired feature manipulation unit for the task of deepfake classification. There is a significant amount of literature that investigates the use of motion magnification as a preprocessing step for deepfake detection. However, we argue that, when the task at hand is deepfake classification, such an approach is sub-optimal for several reasons. First, motion magnification routines utilized by a lot of deepfake detection methods are optimized for a very narrow set of motions and lack applicability to motions outside of that domain. If the objective of the preprocessing is to reveal and amplify temporal inconsistencies in a deepfake video then there is no guarantee that using motion magnification will focus on these precise motions. Second, far from revealing desired inconsistencies in the video, many of the magnification approaches aim to preserve and enhance temporal and spatial consistency in the output, magnified, recording which is likely to dampen the features relevant to deepfake classification. It is therefore important to understand the applicability of motion magnification-like architectures to deepfake classification in a more incremental fashion. For this reason, in our research, instead of utilizing motion magnification as a preprocessing step, we aim to take a first step at understanding the effect of motion magnification inspired network architectures to the classification accuracy of convolutional neural networks. As a starting point for our research we use existing learning-based motion magnification unit proposed by

Oh *et al.* [8]. However, unlike the original network, we are not concerned with the reconstruction of the resulting video frame but only with manipulation of feature vector of concern. To understand the contribution of the unit we test several variations of the network. In our experimentation we find that using dynamic, learnable, and Gaussian smoothed multiplication mask work achieves optimal accuracy on the dataset at hand. We also show that values of the multiplication mask posses spatial interpretability by visualizing the mask, and observing attenuation at regions such as outer boundaries of the face which would normally contain merging artifacts. To prove the consistency of the results we conduct seven runs with each experimental setup and report the average accuracy. We test the results on publicly available Eye-Q deepfake detection dataset. Overall we observe that the inclusion of motion magnification-inspired feature manipulation unit in the network positively contributes to the final accuracy of the network with the exception of using static multiplication factor. Such a result is not completely surprising because while the dynamic multiplication mask manages to learn input specific amplification of the features vector, static multiplication factor does not constitute any meaningful constant for the task at hand. Therefore, the multiplication factor of two used in our experimentation has a very little relevance to the classification performance of the network. In addition, because our task at hand is not the magnification of the input video stream we do not always benefit from amplifying the feature vector. To the contrary, since some of the spatial regions bear no relevance to the classification of the video, it could be beneficial to dampen such regions instead of amplifying them. Thus by altering the activation function to active both dampening and amplification we add another dimension of flexibility to feature manipulation network. In this research, by experimenting with different configurations of magnification-inspired feature manipulation unit, we take the first step at understanding use of magnification-like architectures in classification, and manage to display potential applicability to the task of deepfake detection.

As future work applicability of aforementioned architecture to more types of backbone convolutional architectures can be tested. In addition performance on other types of deepfake detection architectures can be measured as well. The

method can also be tested on classification tasks outside of deepfake classification to understand contribution of magnification-like architectures to the task of classification in general. Because the magnification-inspired feature manipulation unit is highly modular and can be inserted at any section of the convolutional network, the performance improvements from inserting the unit into multiple sections of the network can be measured.

As a closing remark, although we display reasonable improvement with the addition of magnification-like feature manipulation unit, we believe that further experimentation needs to be carried out to understand magnification architectures and their use if video classification.

Bibliography

[1] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[2] R. Caldelli, L. Galteri, I. Amerini, and A. Del Bimbo, "Optical flow based cnn for detection of unlearnt deepfake manipulations," *Pattern Recognition Letters*, vol. 146, pp. 31–37, 2021.

[3] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92, 2019.

[4] M. Mao and J. Yang, "Exposing deepfake with pixel-wise ar and ppg correlation from faint signals," *arXiv preprint arXiv:2110.15561*, 2021.

[5] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 2823–2832, Association for Computing Machinery, 2020.

[6] P. Gupta, K. Chugh, A. Dhall, and R. Subramanian, "The eyes know it: Fakeet- an eye-tracking database to understand deepfake perception," in *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, (New York, NY, USA), p. 519–527, Association for Computing Machinery, 2020.

[7] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson, "Motion magnification," *ACM Trans. Graph.*, vol. 24, p. 519–526, jul 2005.

[8] T.-H. Oh, R. Jaroensri, C. Kim, M. Elgharib, F. Durand, W. T. Freeman, and W. Matusik, "Learning-based video motion magnification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[9] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Trans. Graph.*, vol. 31, jul 2012.

[10] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.

[11] K. B. Raja, R. Raghavendra, and C. Busch, "Video presentation attack detection in visible spectrum iris recognition using magnified phase information," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2048–2056, 2015.

[12] H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, W. Feng, Y. Liu, and J. Zhao, *DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms*, p. 4318–4327. New York, NY, USA: Association for Computing Machinery, 2020.

[13] J. Fei, Z. Xia, P. Yu, and F. Xiao, "Exposing ai-generated videos with motion magnification," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30789–30802, 2021.

[14] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-gan: Unpaired video-to-video translation," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 647–655, 2019.

[15] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, "Space-time neural irradiance fields for free-viewpoint video," in *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pp. 9421–9431, June 2021.

[16] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. Berg, "Dance dance generation: Motion transfer for internet videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[17] A. Siarohin, S. Lathuiliere, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-gan: Unsupervised video retargeting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[19] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson, "Motion magnification," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 519–526, 2005.

[20] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194, 1999.

[21] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: automatically replacing faces in photographs," in *ACM SIGGRAPH 2008 papers*, pp. 1–8, 2008.

[22] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel, "Exchanging faces in images," in *Computer Graphics Forum*, vol. 23, pp. 669–676, Wiley Online Library, 2004.

[23] J. Zhu, L. V. Gool, and S. C. Hoi, "Unsupervised face alignment by robust nonrigid mapping," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1265–1272, 2009.

[24] J.-K. Chou and C.-K. Yang, "Simulation of face/hairstyle swapping in photographs with skin texture synthesis," *Multimedia tools and applications*, vol. 63, no. 3, pp. 729–756, 2013.

[25] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4834–4844, June 2021.

[26] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia*, 2021.

[27] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[28] I. Perov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, M. Dpfks, C. S. Facenheim, L. RP, J. Jiang, *et al.*, "Deepfacelab: Integrated, flexible and extensible face-swapping framework," *arXiv preprint arXiv:2005.05535*, 2020.

[29] R. Chen, X. Chen, B. Ni, and Y. Ge, "Simswap: An efficient framework for high fidelity face swapping," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 2003–2011, Association for Computing Machinery, 2020.

[30] R. Natsume, T. Yatagawa, and S. Morishima, "Rsgan: face swapping and editing using face and hair representation in latent spaces," *arXiv preprint arXiv:1804.03447*, 2018.

[31] L. Zhang, S. Wang, and D. Samaras, "Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable model," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 209–216 vol. 2, 2005.

[32] J. R. Tena, R. S. Smith, M. Hamouz, J. Kittler, A. Hilton, and J. Illingworth, "2d face pose normalisation using a 3d morphable model," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 51–56, IEEE, 2007.

[33] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[34] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 353–360, 1997.

[35] Y.-J. Chang and T. Ezzat, "Transferable videorealistic speech animation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 143–151, 2005.

[36] R. Xu, Z. Zhou, W. Zhang, and Y. Yu, "Face transfer with generative adversarial network," *arXiv preprint arXiv:1710.06090*, 2017.

[37] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," *arXiv preprint arXiv:1910.08854*, 2019.

[38] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "Wilddeepfake: A challenging real-world dataset for deepfake detection," in *Proceedings of the 28th ACM international conference on multimedia*, pp. 2382–2390, 2020.

[39] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[40] G. B. de Souza, D. F. da Silva Santos, R. G. Pires, A. N. Marana, and J. P. Papa, "Deep texture features for robust face spoofing detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 12, pp. 1397–1401, 2017.

[41] Q.-T. Phan, D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale, "Face spoofing detection using ldp-top," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 404–408, 2016.

[42] S. McCloskey and M. Albright, "Detecting gan-generated imagery using color cues," *arXiv preprint arXiv:1812.08247*, 2018.

[43] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, 2018.

[44] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, 2018.

[45] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[46] P. S. H. Q. Yuezun Li, Xin Yang and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, 2020.

[47] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.

[48] Y. Xu, R. Zhang, C. Yang, Y. Zhang, Z. Yang, and J. Liu, "New advances in remote heart rate estimation and its application to deepfake detection," in *2021 International Conference on Culture-oriented Science Technology (ICCST)*, pp. 387–392, 2021.

[49] T. Jung, S. Kim, and K. Kim, "Deepvision: Deepfakes detection using human eye blinking pattern," *IEEE Access*, vol. 8, pp. 83144–83154, 2020.

[50] U. A. Ciftci, I. Demir, and L. Yin, "Fakecatcher: Detection of synthetic portrait videos using biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[51] L. Guarnera, O. Giudice, and S. Battiato, "Deepfake detection by analyzing convolutional traces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[52] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2185–2194, June 2021.

[53] J. Hu, X. Liao, J. Liang, W. Zhou, and Z. Qin, "Finfer: Frame inference-based deepfake detection for high-visual-quality videos," *IEEE Trans. Pattern Anal. Mach. Intell*, pp. 1–9, 2022.

[54] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[55] N. Beuve, W. Hamidouche, and O. Deforges, "Dmyt: Dummy triplet loss for deepfake detection," in *Proceedings of the 1st Workshop on Synthetic Multimedia - Audiovisual Deepfake Generation and Detection*, ADGD '21, (New York, NY, USA), p. 17–24, Association for Computing Machinery, 2021.

[56] J. Li, H. Xie, L. Yu, X. Gao, and Y. Zhang, "Discriminative feature mining based on frequency information and metric learning for face forgery detection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[57] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 772–781, June 2021.

[58] H. Ge, X. Tu, W. Ai, Y. Luo, Z. Ma, and M. Xie, "Face anti-spoofing by the enhancement of temporal motion," in *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pp. 106–111, 2020.

[59] S. Fernandes, S. Raj, E. Ortiz, I. Vintila, M. Salter, G. Urosevic, and S. Jha, "Predicting heart rate variations of deepfake videos using neural ode," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[60] J. Fei, Z. Xia, P. Yu, and F. Xiao, "Exposing ai-generated videos with motion magnification," *Multimedia Tools and Applications*, pp. 1–14, 2020.

[61] R. Das, G. Negi, and A. F. Smeaton, "Detecting deepfake videos using euler video magnification," *arXiv preprint arXiv:2101.11563*, 2021.

[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[63] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pp. 59–66, 2018.

[64] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L.-P. Morency, "Convolutional experts constrained local model for 3d facial landmark detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[65] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

[66] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[67] "Deepfakes/faceswap: Deepfakes Software for all.." Accessed Jun. 07, 2022 [Online].