An Analysis of Different Convolutional Approaches on Facial Manipulation for Deepfake Detection

A thesis

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

Farhan Sharukh Hasan	**170204066**
Md. Rahat Kader Khan	**170204074**
Hafez Md. Nayeem Uddin Khan	**170204091**
Shweta Bhattacharjee Porna	**170204111**

Supervised by

Mr. Shoeb Mohammad Shahriar

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

June 30, 2022

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mr. Shoeb Mohammad Shahriar, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Farhan Sharukh Hasan
170204066

Md. Rahat Kader Khan
170204074

Hafez Md. Nayeem Uddin Khan
170204091

Shweta Bhattacharjee Porna
170204111

CERTIFICATION

This thesis titled, **"An Analysis of Different Convolutional Approaches on Facial Manipulation for Deepfake Detection"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in June 30, 2022.

Group Members:

Farhan Sharukh Hasan	**170204066**
Md. Rahat Kader Khan	**170204074**
Hafez Md. Nayeem Uddin Khan	**170204091**
Shweta Bhattacharjee Porna	**170204111**

Mr. Shoeb Mohammad Shahriar
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Mohammad Shafiul Alam
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

First and foremost, we would like to express our profound gratitude to our supervisor, Mr. Shoeb Mohammad Shahriar, Assistant Professor, Department of CSE, AUST, for his unwavering support during the course of our thesis research and study as well as for his persistence, inspiration, passion, and vast knowledge. His guidance helped us in all the time of research and writing of this thesis. Our thesis study's mentor and supervisor were everything we could have hoped for. Besides our advisor, we would like to thank the faculties of the department for their support and help. We would like to thank our families for supporting us spiritually throughout our lives. Last but not the least, we are greatly thankful to the Almighty who granted us patience and motivation to complete our work to our hearts content.

Dhaka
June 30, 2022

Farhan Sharukh Hasan

Md. Rahat Kader Khan

Hafez Md. Nayeem Uddin Khan

Shweta Bhattacharjee Porna

ABSTRACT

In this era of misinformation, deepfake video is the most realistic fake info of all. It is a technique that uses a pre-trained generative adversarial network (GAN) to automatically replace the face of one person in a video with the face of another person. It has become nearly imperceptible to the naked eye due to recent development in hardware and software. It has now become a major source of concern for individuals all over the world, particularly on social networking sites like as Facebook, YouTube, Twitter etc. This recall for finding a better approach to detect deepfake videos. Over the years many convolutional approaches has been taken. Although there are only a few publically available dataset, those approaches were trained and tested on different datasets. Here we represent a comparative study on those various approaches. We begin by defining the term "Deepfake" and the motivation for researching this topic. Then we'll go over the databases that are publicly available, as well as the database that we're working with. Then we go over four convolutional techniques in details, including certain terms related to architectures. Following that, we showed related work on those architectures. Then, using the same dataset, we present a comparative study of those four convolutional techniques. Furthermore, we also discussed our long-term research strategy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the 19th century, photo manipulation was developed and soon applied to motion pictures. Although Deepfake technology has been developed by researchers at academic institutions beginning in the 1990s, lack of hardware power and advancement of technology restricted it's development. But today's advancement of hardwares make it possible to develop this technology even by amateurs in online communities. Nowadays, digital manipulation has been one of the most highlighted topic in this modern generation [20].

1.1 Definition

So, what is "Deepfake"? Formally, a deep learning approach to make fake content from original one is "Deepfake". "Deepfake" is referred to a deep learning based technique able to create fake videos by swapping the face of a person by the face of another person. It is typically used to spread false information. Media forensics has attracted a lot of attention in the last years in part due to the increasing concerns around Deepfakes. Deepfake technology is used in to create falsified content, replacing faces, speech, and manipulating emotions. It is used to digitally imitate an action by a person that the person did not commit. Deepfake can be used to defame persons by replacing their face by the face of a different person.

1.2 Deepfake in Old Age

While misleading video and image content has existed for as long as humanity, Deepfakes first appeared with the development of AI. However, Deepfake videos are based on technology that is more older than most people realize. In the 19th century, when the first cameras were beginning to be widely utilized in society, the concept of picture alteration initially

emerged. Soon, they would also be accustomed to video formats. Media content manipulation is therefore nothing new. Researchers at academic institutions developed Deepfake technology starting in the 1990s, and later, amateurs in online communities. It's very difficult to pinpoint the exact moment 'Deepfakes' were invented, as there are many different types of algorithms and AI developments being created similarly in different locations by different people. One may argue that the 1997 release of the Video Rewrite tool marked the beginning of Deepfake videos. One of the earliest programs of its sort, it was capable of editing existing videos of people speaking. The person in the original video was able to accurately lip-synch the words of the new track by adding an audio overlay to the original footage. Although "Deepfakes" have no one creator, one of their most essential aspects, generative adversarial networks (GANs), were developed in 2014 by Ian Goodfellow, a PhD graduate who would subsequently begin working at Apple. [21]

1.3 Todays's Deepfake

With the development of generative adversarial networks (GAN) in 2014, the field of synthetic picture and video synthesis gained significant pace. A Reddit user initially used the term "Deepfake" in 2017 to describe a technique for employing face swapping technology to produce fake pornography of famous people.

Figure 1.1: Progress of GANs over the last few years [1]

This anonymous user's technique gained popularity at the same time as TikTok and apps for anti-aging or facial rejuvenation, and shortly thereafter the first program that added any face to an existing video was released. The Internet was flooded with films for (mainly) hilarious purposes, including everyone from Bolsonaro as The Red Grasshopper to Cristina Kirchner as a Drag Queen from RuPaul's Race. The Wall Street Journal reports that the CEO

of an English company sent 220,000 euros to purchase software that imitated the voice of his German employer. This was the first significant Deepfake scam to go public in September. Children from all over the world are aware that they can mock their professors by playing photos on repeat in a virtual classroom, similar to what the Argentinian senator Esteban Bullrich did in Congress. Deepfakes cause more difficult issues. As seen during Argentina's 2019 presidential campaign, artificial intelligence is already utilized to generate large numbers of comments to place a good or service on e-commerce platforms. Our interaction with digital images was elevated by the pandemic. Seminars, job interviews, court appearances, and legislative sessions are helding in online. In our society's rituals and institutions, being "present" is becoming a less important prerequisite [22].

The current state of Deepfakes is discussed in one group of articles. The first piece presents a historical overview of the top 10 deep fakes currently popular on YouTube and analyzes viewer comments for linguistic responses.In authors view, the present articles are part of the very first wave of empirical work on the social impacts of Deepfakes. The second paper analyzes Reddit in 2018 to determine the atmosphere surrounding Deepfakes and then uses those findings to generate novel potential responses to problematic use cases (see Brooks). In the third paper (see Cochran and Napshin), students are polled about their knowledge of Deepfakes, their concerns about them, and how much control they feel platforms have over the technology. In the first study, researchers look at how young women rate their attractiveness both before and after being exposed to a Deepfaked image that combined their likeness with that of a celebrity. They showed that observing oneself within a Deepfake had favorable impacts contrary to conventional predictions, and they provided mechanisms for how Deepfakes affect self perception [23].

A Deepfake is not just any type of video editing, but rather the use of a particular technology. It is a deep learning process to make a fake record. This deep learning process is not just any type of machine learning, it is one of the most complicated and costly process in artificial intelligence.

1.4 Example of Deepfake

Earlier of year 2021, videos of famous Hollywood actor Tom Cruise started popping up on a social media named TikTok doing some surprisingly un-Tom-Cruise-like stuff. Despite the movie star hair, the eye-squinting and that trademark teeth-baring cackle, it wasn't really Cruise. The 10 videos, which were posted between February and June, featured an artificial intelligence-generated doppelganger meant to look and sound like him. The Deepfakes — a

combination of the terms "deep learning" and "fake" — were created by visual and AI effects artist Chris Umé with the help of a Cruise stand-in, actor Miles Fisher. In fig. 1.4 the left picture is the actor Miles Fisher and the right picture we can see Tom cruise photo using deep learning technology which seems real but it's actually fake [2].

Figure 1.2: Tom Cruise's fake photo generated using deep learning technology [2]

Here is another recent example, hackers posted a fake and heavily edited video of the Ukrainian president Volodymyr Zelenskyy on a Ukrainian news website on Wednesday, which was later debunked and taken down. The video was widely shared on social media. The video, a so-called Deepfake that lasted about a minute, appears to show a rendering of the Ukrainian president telling his soldiers to put down their weapons and give up the battle against Russia. Although the author of the Deepfake is still unknown, Ukrainian government officials have been issuing a warning for weeks about the likelihood that Russia may be spreading manipulated videos as part of its information warfare. Ukraine's military intelligence agency released a video this month about how state-sponsored Deepfakes could be used to sow panic and confusion. Although the lip-sync in the video is adequate, viewers quickly pointed out that Zelenskyy's accent was incorrect and that his head and voice did not appear to be genuine. The video was taken down from Facebook, YouTube, and Twitter, according to representatives of those companies, for breaking their rules. Meanwhile, the misleading video received a boost on Russian social media. Website for fact-checking "Using video forensics technologies and reverse image searching, it can identify that this video was computer-generated using still images from Zelenskyy's past press conferences," the verification service Verify said in a statement confirming this.

Figure 1.3: Zelenskyy Deepfake video goes viral, reflecting troubling new wave of disinformation [3]

The Queen's digitally generated doppelgänger will be speaking on Channel 4 while she delivers her customary address on the BBC and ITV. The BBC was informed by Buckingham Palace that it had no comment on the transmission. According to Channel 4, the purpose was to issue a "stark warning" on fake news in the digital age. Misinformation is frequently disseminated via Deepfake technology, which may be used to produce plausible but wholly fictitious video content. The Deepfake will attempt a popular dancing challenge on TikTok in the message. [24]

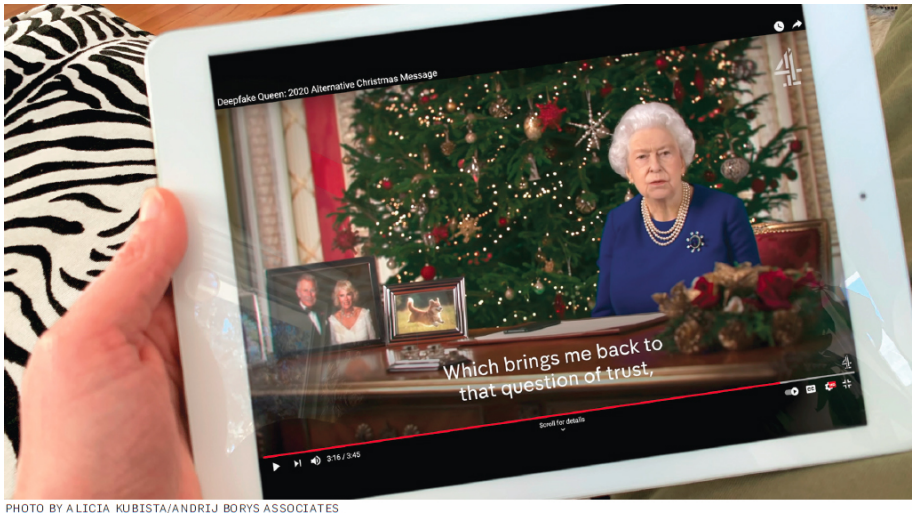

Figure 1.4: A Deepfake video from a December 25, 2020, posting "Deepfake Queen: 2020 Alternative Christmas Message [4] (source https://youtu.be/IvY-Abd2FfM)

1.5 Future Threat of Deepfake

Traditionally, the number and realism of facial manipulations have been limited by the lack of proper editing tools, the domain expertise required, and the complex and time-consuming process involved. Nowadays, it is becoming increasingly easy to a manipulate a real face of one person in an image or video, thanks to

- the accessibility to large-scale public data
- the evolution of deep learning techniques that eliminate many manual editing steps such as Autoencoders and Generative Adversarial Networks

For this open software and mobile application such as ZAO3 and FaceApp4 have been released opening the door to anyone to create fake images and videos, without any experience in the field needed. In response to those increasingly realistic manipulated content, large efforts are being carried out by the research community to design improved methods for face manipulation detection. Traditional fake detection methods in media forensics have been based on in-camera fingerprints, the analysis of the intrinsic fingerprints introduced by the camera device, both hardware and software. Manipulation of visual content has now become omnipresent, and one of the most critical topics in our digital society [25].

The negative effects of Deepfakes will require the efforts of many various parties, including platforms, journalists, and policymakers. Technical specialists should participate actively. Technical experts must pool their knowledge,their comprehension of how Deepfake technologies operate, their understanding of how the technology can be developed and used and focus their efforts in order to find solutions that permit the beneficial uses of synthetic media technologies while minimizing the negative effects. Technological specialists should be involved in creating, developing, and evaluating prospective technical solutions as well as working with legal, policy, and other stakeholders to achieve social solutions.

The Emergence of Deepfake Technology It is getting harder and harder to tell legitimate media from fake, thanks to modern digital technology. The introduction of Deepfakes, which are hyper-realistic videos that use artificial intelligence (AI) to show someone saying and doing things that never happened, is one of the most recent innovations contributing to the issue. Convincing Deepfakes can swiftly reach millions of individuals and have detrimental effects on our culture when combined with the reach and speed of social media. Despite the serious threat that Deepfakes pose to our society, political system, and economy, they can be stopped through legislation and regulation, corporate policies and voluntary action, education and training, as well as the development of technology for Deepfake detection, content authentication, and Deepfake prevention [26].

Chapter 2

Motivation

A viral video that appeared to show former US President Barack Obama cursing and calling President Donald Trump names but revealed the clip was actually fabricated using emerging video-editing technology. In this video we can see Fake Obama created using AI video tool - BBC News [5]. If you didn't know any better, you might think the video is real.

"We're entering an era in which our enemies can make it look like anyone is saying anything at any point in time even if they would never say those things," says former US President Barack Obama, his lips moving in perfect sync with his words as they become increasingly bizarre. So, for instance, they could have me say things like, I don't know, [Black Panther's] Killmonger was right! Or Ben Carson is in the sunken place! [27].

The man speaking is really Oscar-winning director Jordan Peele, not the former president, as the video soon makes clear. He is cautioning viewers not to believe everything they see online. The video featured the voice of director and actor Jordan Peele, whose voice had been added to an original Obama tape to create a "Deepfake"a recording of someone saying or doing something that never actually occurred. "This is a risky period, we need to be more cautious about what we trust from the internet. This technology, which is referred dubbed as "the future of false news," is already being utilized in contentious ways, such as inserting celebrity faces.

- Artificial intelligence was used to precisely model how Mr Obama moves his mouth when he speaks.
- Their technique allows them to put any words into their synthetic Barack Obama's mouth.

Figure 2.1: Fake Obama created using AI video tool - BBC News [5]

We must first understand how important this problem is to real world scenarios. **Let's see few applications where a solution to this problem can be very useful.**

1. "In the era of false news, Deepfake and propaganda based journalism train your mind to seek out the truth." The spread of misinformation through realistic images and videos has because it is used to propagate false information.

2. Deepfakes has demonstrated how computer graphics and visualization techniques can be used to defame someone by replacing their face by the face of a different person.

3. Such actions have made well-known famous celebrities, global leaders, internet sensations their targets.

4. With the development of hardware and software tools, fake identities are made similar to the original ones that provides false or harmful information online.

5. The amount of Deepfake content online is growing at a rapid rate. At the beginning of 2019 there were 7,964 Deepfake videos online, according to a report from startup Deeptrace; just nine months later, that figure had jumped to 14,678. It has no doubt continued to balloon since then [28].

6. People's trust in digital information is decreasing, and at the same time, it's getting harder to tell the difference between true and fake news.

7. Therefore, in order to resolve this issue, we require a method for identifying fake information coming from original sources and for stopping the online misuse of digital content.

Chapter 3

Database

Due to the widespread public concern over Deepfakes, this is one of the most popular face modification study areas nowadays. It involves swapping out one person's face for another person's in a film. In contrast to entire face synthesis manipulation, which operates at the image level, the objective of identity swap is to produce convincing phony videos. Since publicly available fake databases such as the UADFV database [29], up to the recent Celeb-DF and DFDC databases [6], [16] many visual improvements have been carried out, increasing the realism of fake videos. They are categorized in two generation. First and second generation. Three different databases are grouped in the first generation. Regarding the databases included in the 2nd generation, we highlight the recent Celeb-DF and DFDC databases released at the end of 2019. This database aims to provide fake videos of better visual qualities, similar to the popular videos that are shared on the Internet16, in comparison to previous databases that exhibit low visual quality with many visible artifacts. The database that are publicly available and categorized in 1st and 2nd generation are given below:

- 1st Generation
  - UADFV (2018) [29]
  - DeepfakeTIMIT (2018) [30]
  - DeepfakeDetection (2019) [31]

- 2nd Generation
  - FaceForensics++ (2019) [15]
  - Celeb-DF (2019) [6]
  - DFDC Preview (2019) [16]

Table 3.1: Basic information of various Deepfake datasets

Dataset	Real		Deepfake		Dataset Size	Release Date
	Video	Frame	Video	Frame		
UADFV	49	17.3k	49	17.3k	10gb	2018.11
DF-TIMIT:LQ DF-TIMIT:LQ	320	34.0k	320 320	34.0k 34.0k	230mb	2018.12
FF-DF	1000	509.9k	4,000	2,039.6k	9gb	2019.01
DFD	363	315.4k	3,068	2,242.7k	10gb	2019.09
DFDC	1,131	488.4k	4,113	1,783.3k	470gb	2019.10
Celeb-DF	590	225.4k	5,639	2,116.8k	9.45gb	2019.11

3.1 Existing Deepfake Datasets

Deepfake video is generated from generative adversarial network(GAN). This form of network need a very powerful computer. As a result, creating a Deepfake video is incredibly expensive, and just a few of them exist. Here's a quick rundown of the current publicly available dataset:

3.1.1 UADFV:

The UADFV database [29] has 49 real Youtube videos that were utilized to produce 49 false videos. via the FakeApp smartphone app, replacing the original face with the actor Nicolas Cage's.As a result, in all fake videos, only one identity is considered. Each video shows a single individual and has an average resolution of 294*500 pixels and a duration of 11.14 seconds. [19]

3.1.2 DF-TIMIT:

The Deepfake-TIMIT dataset includes 640 Deepfake videos generated with faceswap-GAN [32] and based on the Vid-TIMIT dataset [30]. DF-TIMIT-LQ and DF-TIMIT-HQ are two equal-sized subsets of the videos, with synthesized faces of 64 x 64 and 128 x 128 pixels, respectively, using faceswap.

3.1.3 DFD:

The Google/Jigsaw Deepfake detection dataset [31] has 3, 068 Deepfake videos generated based on 363 original videos of 28 consented individuals of various genders,ages and ethnic groups. The details of the synthesis algorithm are not disclosed, but it is likely to be an

improved implementation of the basic Deepfake maker algorithm. [6] implementation of the basic Deepfake maker algorithm.

3.1.4 DFDC:

Detection Challenge (DFDC) is a accelerate development of new ways to detect Deepfake videos. The DFDC has enabled experts from around the world to come together, benchmark their Deepfake detection models, try new approaches, and learn from each others' work.

The DFDC dataset consists of two versions: [33]

- Preview dataset
  - 5k videos
  - Featuring two facial modification algorithms
  - Associated research paper
- Full dataset
  - 124k videos
  - Featuring two facial modification algorithms
  - Associated research paper

One of the most recent public databases, the DFDC database [16], was released by Facebook in partnership with other companies and academic institutions including as Microsoft, Amazon, and the MIT. In this study, we use the DFDC preview dataset, which contains 1,131 genuine videos from 66 hired actors, ensuring realistic gender, skin tone and age variation.

In contrast to other popular databases, no publicly available data or data from social media sites were used to build this dataset. In terms of bogus videos, a total of 4,119 videos were made using two separate unidentified methods. Fake videos were created by switching persons with comparable face characteristics, such as skin tone, facial hair, glasses, and so on. It is crucial to emphasize that the DFDC database takes into account various acquisition scenarios, including indoor and outdoor , daytime, nighttime, etc. distances from the subject to the camera, and varied poses, among other factors.

3.1.5 Celeb-DF:

The Celeb-DF database [6] was created with the goal of generating fake videos of better visual quality compared with their original UADFV database. Celeb-DF dataset contains real and Deepfake synthesized videos having similar visual quality on par with those circulated online. The average length of all videos is approximate 13 seconds with the standard frame rate of 30 frame-per-second. The real videos are chosen from publicly available YouTube videos, interviews of 59 celebrities with a diverse distribution in their genders,ages and ethnic group. Regarding fake videos, a total of 795 videos were created using Deepfake technology, swapping faces for each pair of the 59 subjects. The final videos are in MPEG4.0 format.

Figure 3.1: Visual artifacts of Deepfake videos in existing datasets. Note some common types of visual artifacts in these video frames, including low-quality synthesized faces (row 1 col 1, row 3 col 2, row 5 col 3), visible splicing boundaries (row 3 col 1, row 4 col 2, row 5 col 2), color mismatch (row 5 col 1), visible parts of the original face (row 1 col 1, row 2 col 1, row 4 col 3) and inconsistent synthesized face orientations (row 3 col 3). This figure is best viewed in color [6]

We used FaceForensics++ (2019) [15] dataset where there are 1000 real videos and 1000 (FaceSwap), 1000 (Deepfakes), 1000 (Face2Face), 1000 (NeuralTexture) fake videos. This database is publicly available. A brief description of this databases is given below.

3.2 FaceForensics++

FaceForensics++ [15] is a forensics dataset consisting of 1000 original video sequences and 4000 fake videos. The manipulated videos are made using four automated face manipulation methods.

- Computer graphics-based approaches
  - Face2Face [34]
  - FaceSwap [35]
- Learning based approaches
  - Deepfake
  - NeuralTextures [36]

Each of the four techniques requires the input of source and target actor video pairs. The final output of each method is a video composed of generated images. Besides the manipulation output, authors also compute ground truth masks that indicate whether a pixel has been modified or not, which can be used to train forgery localization methods. To generate a large scale manipulation database, authors adapted state-of-the-art video editing methods to work fully automatically. The data has been sourced from 977 YouTube videos and all videos contain a track-able mostly frontal face without occlusions which enables automated tampering methods to generate realistic forgeries. As the authors provide binary masks the data can be used for image and video classification as well as segmentation. In addition, the authors provide 1000 Deepfake models to generate and augment new data.

In the following section, these methods are briefly described:

3.2.1 Face2Face

Face2Face [34] is a computer graphics based, facial reconstructing system that transfers the facial expressions from a source video to a target video while preserving the identity of the original person. This implementation is based on two video input streams, including a manual key frame selection. The first frames are used in order to obtain a temporary

face identity (3D model), and the expressions of the face are traced over the remaining frames. The rendered model is blended with the image and color correction is applied. The implementation is computationally lightweight and can be efficiently run on the CPU. [15]

Figure 3.2: Generated Video Frames from Face2Face.

3.2.2 FaceSwap

FaceSwap [35] is a graphics-based approach to transfer the face region from a source video to a target video. This model is back-projected to the target image by minimizing the difference between the projected shape and the localized landmarks using the textures of the input image.o process video database, the Face2Face approach to fully-automatically create reenactment manipulations were adapted. Each video was processed a preprocessing pass here, the first frames was used in order to obtain a temporary face identity (3D model), and track the expressions over the remaining frames. To process video database, the Face2Face approach to fully-automatically create reenactment manipulations were adapted. The frames with the left- and right-most angle of the face were automatically selected in order to select the keyframes required by the approach. Based on this identity reconstruction, the whole video to compute per frame the expression was tracked, rigid pose, and lighting parameters as done in the original implementation of Face2Face. The reenactment video outputs by transferring the source expression parameters of each frame to the target video was generated. [15]

Figure 3.3: Graphics based generated Video Frames from FaceSwap.

3.2.3 Deepfakes

The term Deepfakes has widely become a synonym for face replacement based on deep learning, but it is also the name of a specific manipulation method that was spread via online forums. The method is based on two autoencoders with a shared encoder that are trained to reconstruct training images of the source and the target face, respectively. To create a fake image, the trained encoder and decoder of the source face are applied to the target face. Deepfakes has become the synonym for face manipulations of all kind, however it origins to FakeApp [37]and faceswap github [38]. A face in a target sequence is replaced by a face that has been observed in a source video or image collection. The technique is built on two autoencoders that share an encoder and are trained to recreate training images of the source and target faces, respectively. The photos are cropped and aligned using a face detector. The trained encoder and decoder of the source face are applied to the target face to produce a fake image. Using Poisson image editing, the autoencoder output is then blended with the rest of the image.

Figure 3.4: Learning based generated Video Frames from Deepfakes.

3.2.4 NeuralTextures

NeuralTextures [36] uses the original video data to learn a neural texture of the target person, including a rendering network. We only modify the facial expressions corresponding to the mouth region, i.e., the eye region stays unchanged (otherwise the rendering network would need conditional input for the eye movement similar to Deep Video Portraits). The NeuralTextures-based rendering method used by Thies et al. [39] uses facial reenactment as an example. It uses the original video data to learn a neural texture of the target person, including a rendering network. This is trained with a photometric reconstruction loss in combination with an adversarial loss. a patch-based GAN-loss as used in Pix2Pix [40] is applied. The NeuralTextures approach relies on tracked geometry that is used during train and test times.

Figure 3.5: Learning based generated Video Frames from NeuralTextures.

Chapter 4

Background Studies

The chapter background study provides context to the information that we are going to discuss in this paper. Here we included some important topic we we conducted our research. The topic includes Neural Network (NN), Convolutional Neural Network (CNN), 3D ConvNet. We also briefly describe some well known architecture in convolution neutral network like XceptionNet [12], I3D [41] , Capsule Network [13] . Throughout our research we study them and experimented with them.

4.1 Neural Network (NN)

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Warren McCullough and Walter Pitts, two scholars from the University of Chicago who transferred to MIT in 1952 to become founding members of what is commonly referred to as the first cognitive science department, first proposed neural networks in 1944 [42].

Neural network interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. We can classify and cluster data using neural networks. They can be viewed as a layer of clustering and classification on top of the data you manage and store. They help in organizing unlabeled data into groups based on similarities between example inputs, and when given a labeled training set, they categorize data [7].

It is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. It can adapt to changing input. So the network generates the best possible result without needing to redesign the output criteria. The fundamental unit of a neural network is the “neuron”. Analogous to a

biological neuron, an artificial neuron is a computational unit that can receive some input, process it and propagate on some output downstream in the network.

fig. 4.1 shows a diagram of what one node might look like.

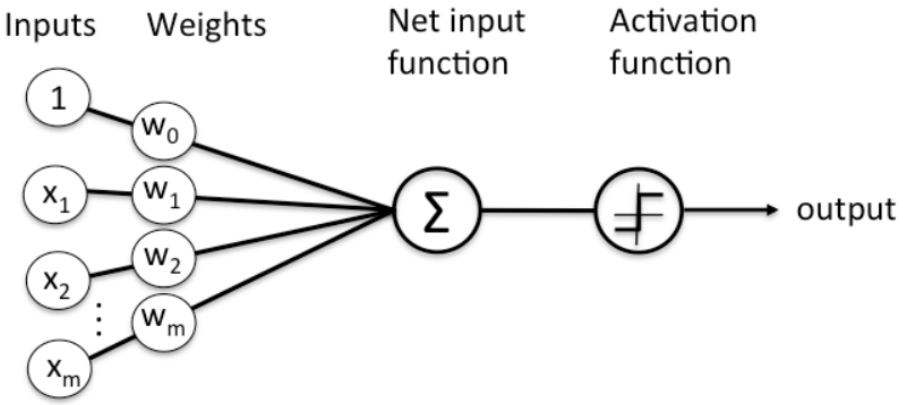

Figure 4.1: One node diagram [7]

Figure 4.2 illustrates a simple neural network. This network has three input neurons, two hidden layers with four neurons each, and one output neuron. [8]

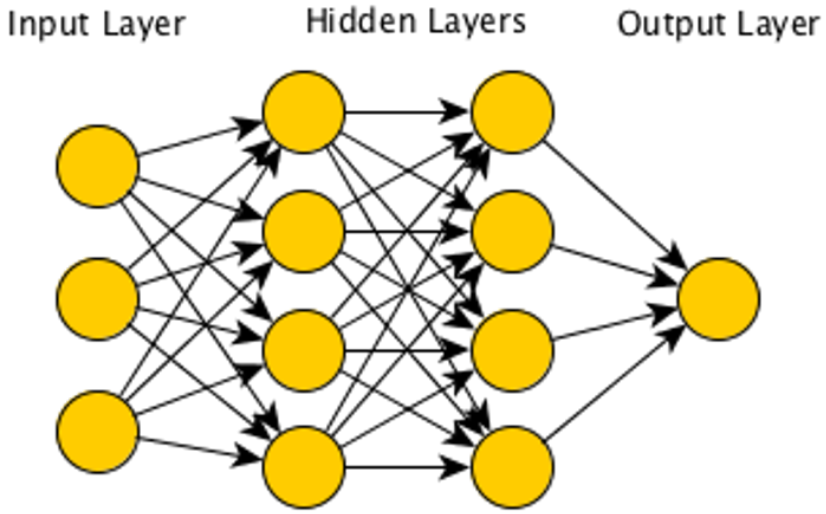

Figure 4.2: A simple neural network [8]

An input layer, a processing layer, and an output layer make up the three main parts. Various criteria may be used to weight the inputs. There are nodes and connections between these nodes in the processing layer, which is concealed from view, that are meant to be analogous to the neurons and synapses in an animal brain [43]. A neural network functions in a way similar to that of the human brain. In a neural network, a "neuron" is a mathematical function that gathers and classify data in accordance with a particular architecture. The network is quite similar to statistical techniques like regression analysis and curve fitting.

Layers of interconnected nodes make up a neural network. Every node is a perceptron, which resembles a multiple linear regression. The multiple linear regression signal is fed into a potentially nonlinear activation function via the perceptron.

Multi-Layered Perceptron: Perceptrons are organized in interconnected layers in a multi-layered perceptron (MLP). Input patterns are gathered by the input layer. Input patterns may map to classifications or output signals in the output layer.

The usage of neural networks is widespread, including applications in trading, business analytic, financial operations, corporate planning, and product maintenance. In business applications including forecasting and marketing research solutions, fraud detection, and risk assessment, neural networks have also become increasingly popular.

A neural network analyses price data to find trading opportunities based on the examination of the data. The networks are able to identify subtle nonlinear patterns and interdependencies that conventional technical analysis techniques cannot [43].

4.2 Convolutional Neural Network (CNN)

CNN is a particular type of feed-forward neural network in AI. It is widely used for image recognition. CNN represents the input data in the form of multidimensional arrays. It works well for a large number of labeled data. CNN extract the each and every portion of input image, which is known as receptive field. It assigns weights for each neuron based on the significant role of the receptive field. Convolutional Neural Networks [44] have been one of the most influential innovations in the field of computer vision. The peculiarity of a CNN lies in its filter layers, which comprise at least one folding layer. The input to CNN (such as an image) is routed through a series of layers to obtain a labeled output that can then be classified. They have performed a lot better than traditional computer vision and have produced state-of-the-art results. These neural networks have proven to be successful in many different real-life case studies and applications like:

- Image classification, object detection, segmentation, face recognition
- Self driving cars that leverage CNN based vision systems
- Classification of crystal structure using a convolutional neural network and many more, of course.

Figure 4.3. Illustrates a simple convolutional neural network. To understand it's success, we have to go back to 2012, the year in which Alex Krizhevsky used convolutional neural

networks to win that year's ImageNet Competition, reducing the classification error from 26% to 15%.

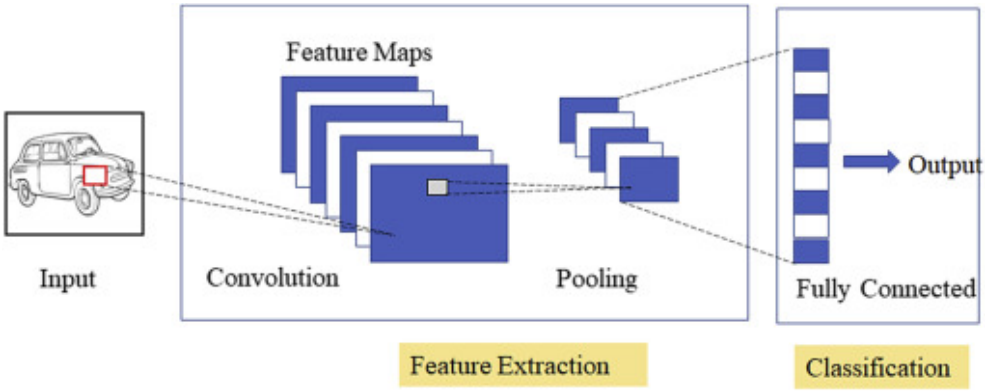

Figure 4.3: Convolutional neural network [9]

Convolutional neural networks utilize less pre-processing than other deep learning systems, which allows the network to learn filters that are generally created manually or produced in other systems.

Because CNNs are so independent of human effort, they have many advantages over other algorithms. Few people today would create a commercial application or take part in a competition in the field of computer vision without creating a CNN-based design. These architectures are now so common and well-liked. Deep Convolutional Neural Networks, commonly known as ConvNet, are widely used nowadays and are excellent at classifying images because they make use of spatial information. There are three types of primary layers in a convolutional neural network:

- convolutional layer
- pooling layer
- fully connected layer

The deep CNN architecture depicted in can be created by continually using these three layers. These layers are arranged in a specific order, starting with input layer and ending with output layer. Each of these layers has different parameters that can be optimized and performs a different task on the input data.

Input layer

A feature is a significant and distinctive attribute of an image. Input layer is the first layer of every CNN. It keeps the raw pixel value of the image. Pre-processing techniques can be used to feed input images to the CNN's input layer in order to increase accuracy. The

input image's composition would be (image height) x (image width) x (image depth). While grayscale images would have a depth of 1, RGB images would have a depth of 3.

Convolutional layer

The main component of CNN is the convolutional layer. It is made up of a number of learnable convolution kernels or filters. To create a feature map, these filters can learn feature representations of the input image. Each neuron in a feature map is connected to a set of neurons in the layer below it by a set of trainable weights. The input feature maps are first convolved with a learned kernel in order to produce a new feature, and the outcomes are then fed into a nonlinear activation function. By using various kernels, we will obtain various feature maps. Sigmoid, tanh, and ReLU are the common activation functions . By using many feature maps within the same convolutional layer, multiple features can be extracted at each location.

Pooling layer

By reducing the spatial resolution of the feature maps, the pooling layer is in charge of extracting prominent features. As a result, the computational performance is increased and spatial invariance to input distortions and translations is achieved. Normally, it is placed in-between convolutional layers. The moving step of kernels determines the size of feature maps in the pooling layer. Average pooling and max pooling are the two most common pooling operations. By stacking several convolutional layers and pooling layers, we can extract the high-level characteristics of inputs.

Fully connected layer

In a fully connected layer each neuron is connected to every neuron in the previous layer, and each connection has its own weight. As a result, it requires a lot of memory (weights) and computation (connections). This layer performs high level reasoning while flattening the input feature representation into a feature vector.

Output layer

The output layer is the final layer of CNN. It is in charge of producing the output probabilities for each specified input class. The output probability is calculated using a softmax unit. Because it produces a high-performing probability distribution, Softmax is usually applied.

Each output class's probability adds up to one. The proper class will be the one with the biggest value.

4.2.1 Depth-wise Convolution and Depth-wise Separable Convolution

Standard convolution layer of a neural network involve input x output x width x height parameters, where width and height are width and height of filter. [9] Over-fitting is more likely when there are more parameters. People have frequently searched for various convolutions to prevent such situations. These categories include depth-wise convolution and depth-wise separable convolution.

4.3 Depth-wise Convolution

Normal CNN generally have two or three layers but deep CNN will have multiple hidden layers usually more than 5, which are used to extract more features and increase the accuracy of the prediction. In this convolution, we apply a 2-D depth filter at each depth level of input tensor. Lets understand this through an example. Suppose our input tensor is 3x8x8 (input channels x width x height). Filter is 3x3x3. [9] In a standard convolution we would directly convolve in depth dimension as fig. 4.4.

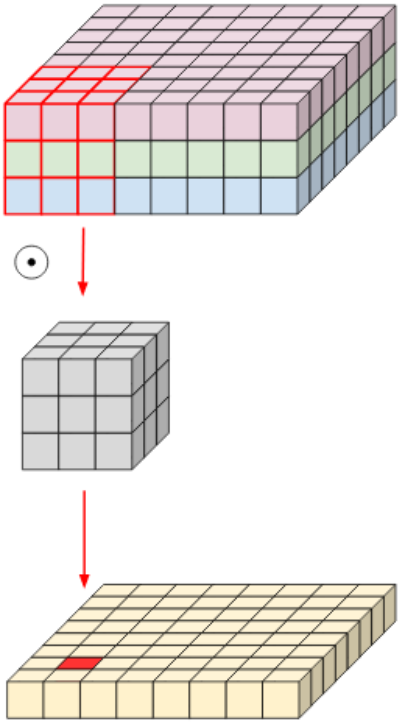

Figure 4.4: Normal convolution [9]

In depth-wise convolution, we use each filter channel only at one input channel. In the example, we have 3 channel filter and 3 channel image. What we do is to break the filter and image into three different channels and then convolve the corresponding image with corresponding channel and then stack them back fig. 4.5.

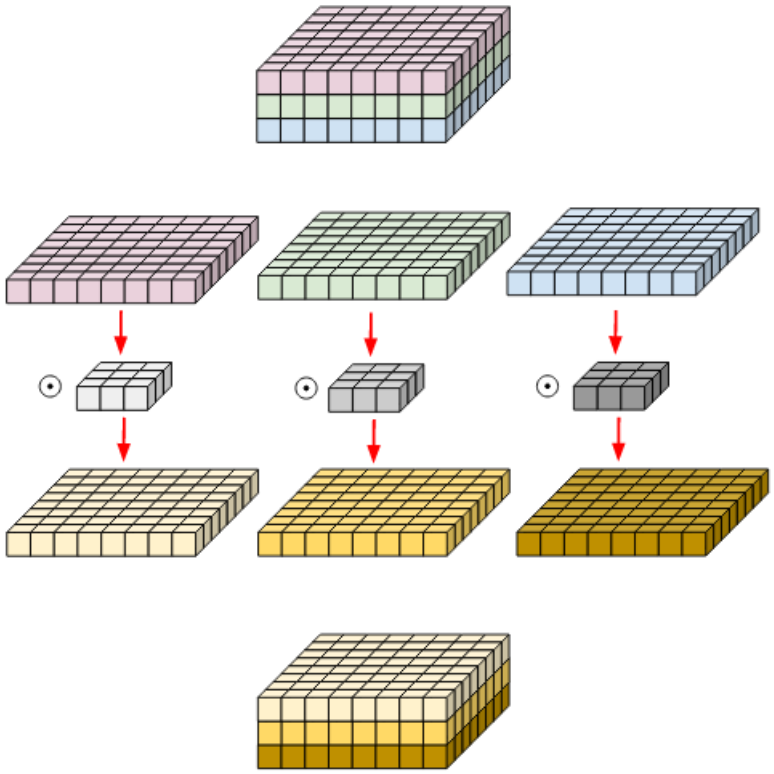

Figure 4.5: Depth-wise convolution. Filters and image have been broken into three different channels and then convolved separately and stacked thereafter [10]

Select a channel, set all of the filter's elements to zero aside from that channel, and then convolve to achieve the same result as with standard convolution. One filter will be required for each channel, making a total of three. Even though the parameters are the same, this convolution offers you three output channels with only one 3-channel filter, whereas standard convolution would require three 3-channel filters [10].

4.4 Depth-wise Separable Convolution

depth-wise separable convolutions or sometimes simply separable convolutions are variations of the standard convolution now the whole motivation for having a separable convolution is so that we can do something similar or same as convolution same as convolution but with lesser number of parameters . Depthwise Separable Convolution divides the computation into two phases, as opposed to standard convolution, which does the channel-wise and

spatially-wise processing in one step. A single convolutional filter is applied for each input channel during depth-wise convolution and the output of depth-wise convolution is then combined linearly using point-wise convolution. [11]. This convolution originated from the idea that depth and spatial dimension of a filter can be separated- thus the name separable. Let us take the example of Sobel filter, used in image processing to detect edges. We can separate the height and width dimension of these filters. Gx filter (see fig. 4.6) can be viewed as matrix product of [1 2 1] transpose with [-1 0 1]. [10]

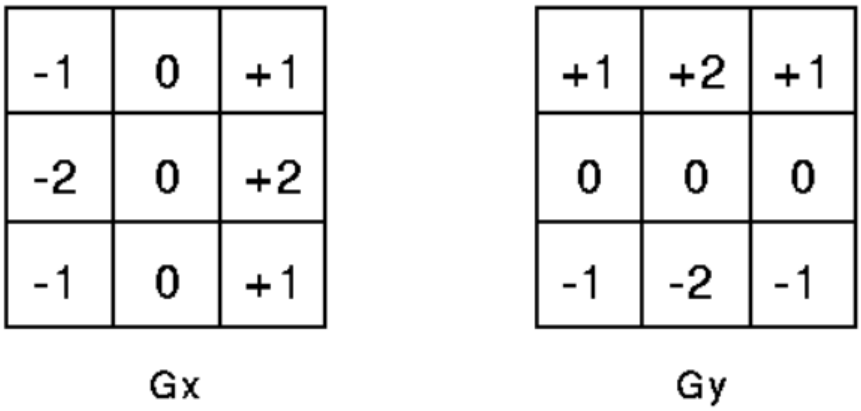

Figure 4.6: Sobel Filter. Gx for vertical edge, Gy for horzontal edge detection [11]

We notice that the filter had posed as something else. It appears to have nine parameters, however there are only six. The separation of its height and breadth dimensions has made this possible. We can do depth-wise convolution and then use a 1x1 filter to cover the depth dimension by using the same concept to separate the depth dimension from the horizontal (width x height) dimension fig. 4.7. Suppose we have 32 3x3 filter and the number of channels in the filter or the depth of the filter is 16. So when we convolve these 32 3x3 filters to 16 channel input what comes out is, dimension say if this is x and y same size input maybe x-2 and y-2 but the depth of this will be 32 because we have 32 input filters that we are multiplying. So let's break it down so that we have 3x3 matrices 16 of them sitting side by side or sitting separately. This operation can be viewed as first filter multiplies first channel, second filter multiplies second channel,third filter multiplies third channel second channel this one multiplies third and the last one last filter multiplies last channel and then we simply sum them all together right. We multiply with the first one and then sum to obtain the single channel output of one filter, this is the standard standard convolution operation. The number of parameter will be 32x3x3x16. Now the trick behind depth wise separable is that what we want to do now is we don't add these so we've multiplied the first channel with my first 3x3 value we've also multiplied my second channel with my second 3x3 value and so on. So instead of performing these additions that we have we don't add but we simply stack them.We will get an output that same size convolution but in we have 32 filters but each of them only have a 1 x 1 matrix weight so this is actually 1 x 1 for all

the 32 filters so in this way we get the same output and the number of the parameter will be 32x1x1x16. Number of parameter is less here. It turns out separable convolutions have similar performance as regular convolutions so it never hurts to use separable convolutions if we particularly want deeper neural network such that our training becomes much faster. These type of CNN's are widely used because of the following two reasons –

- They have lesser number of parameters to adjust as compared to the standard CNN's, which reduces overfitting
- They are computationally cheaper because of fewer computations which makes them suitable for mobile vision applications

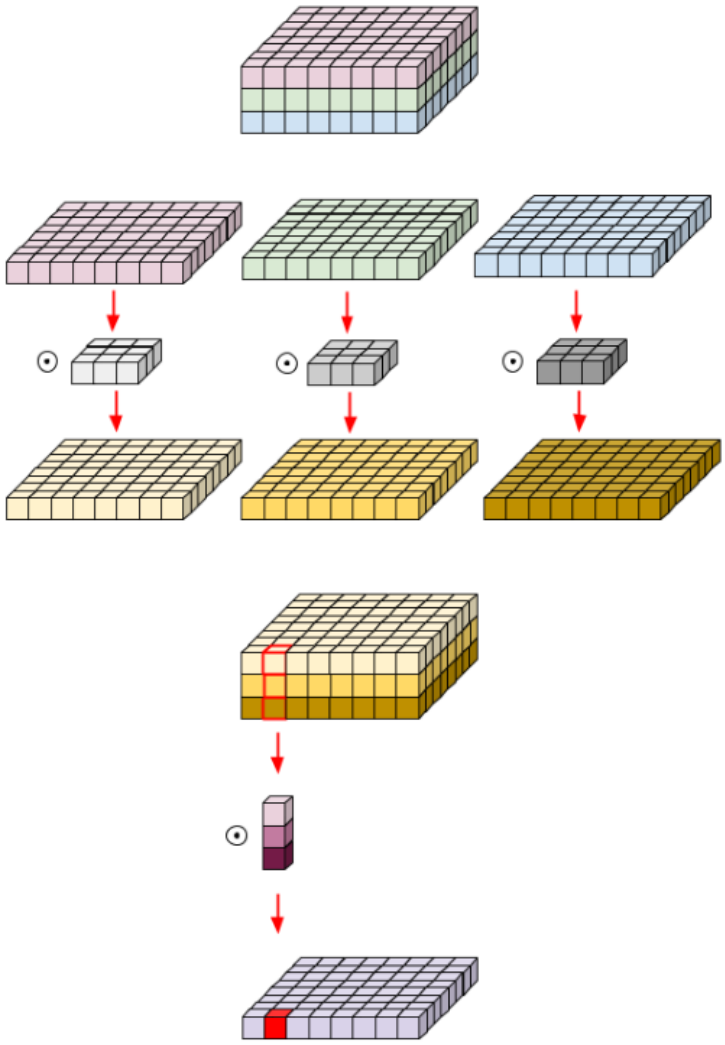

Figure 4.7: Depth-wise Separable Convolution [11]

4.5 XceptionNet

XceptionNet is a deep learning algorithm that is proposed by Francois Chollet (2017) [12]. The author propose a convolutional neural network architecture based entirely on depth-wise separable convolution layers. In effect, the author make the following hypothesis: that the mapping of cross-channels correlations and spatial correlations in the feature maps of convolutional neural networks can be entirely decoupled. Because this hypothesis is a stronger version of the hypothesis underlying the Inception architecture, they name their proposed architecture Xception, which stands for “Extreme Inception”. The author also experimented with their proposed model and compared with three other models.

XceptionNet is a traditional CNN trained on ImageNet based on separable convolutions with residual connections [15]. Xception, slightly outperforms Inception V3 on the ImageNet dataset (which Inception V3 was designed for) and significantly outperforms Inception V3 on a larger image classification dataset comprising 350 million images and 17,000 classes. Since the Xception architecture has the same number of parameters as Inception V3, the performance gains are not due to increased capacity but rather to a more efficient use of model parameters. Xception is a convolutional neural network that is 71 layers deep. We can load a pretrained version of the network trained on more than a million images from the ImageNet database and use to classify new images using the Xception model. Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. This observation leads them to propose a novel deep convolutional neural network architecture inspired by Inception [12]. The shape of a filter in a normal CNN layer is (output channels, input channels, k, k), where k denotes the filter size. Such a filter’s convolution operation evaluates spatial relations simultaneously in one channel and across many channels.Alternately, we could say that a CNN layer makes an effort to learn filters in a 3D space with two spatial dimensions—height and width—as well as a channel dimension. Therefore, the mapping between channel correlations and spatial correlations must be done using a CNN filter at the same time. Deep Learning with Depth-wise Separable Convolutions, breaks this duo way interpretation of channel correlations and spatial correlations from a single task into two separate operations. If you remember the Inception Net, it also does something very similar to decouple the cross-channel and spatial correlations, so this isn’t altogether new. The main difference is that the Xception Net, whose name stands for "Extreme Inception," makes a much stronger assumption about this decoupling [45].

The Xception architecture has 36 convolutional layers forming the feature extraction base of the network. The 36 convolutional layers are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. In short,

the Xception architecture is a linear stack of depthwise separable convolution layers with residual connections. This makes the architecture very easy to define and modify.It takes only 30 to 40 lines of code using a high level library such as Keras or TensorFlow-Slim, not unlike an architecture such as VGG-16, but rather unlike architectures such as Inception V2 or V3 which are far more complex to define. An open-source implementation of Xception using Keras and TensorFlow is provided as part of the Keras Applications module2 under the MIT license. In the below fig. 4.8 is the complete description of the specifications of the network proposed by Francois Chollet (2017) [12].

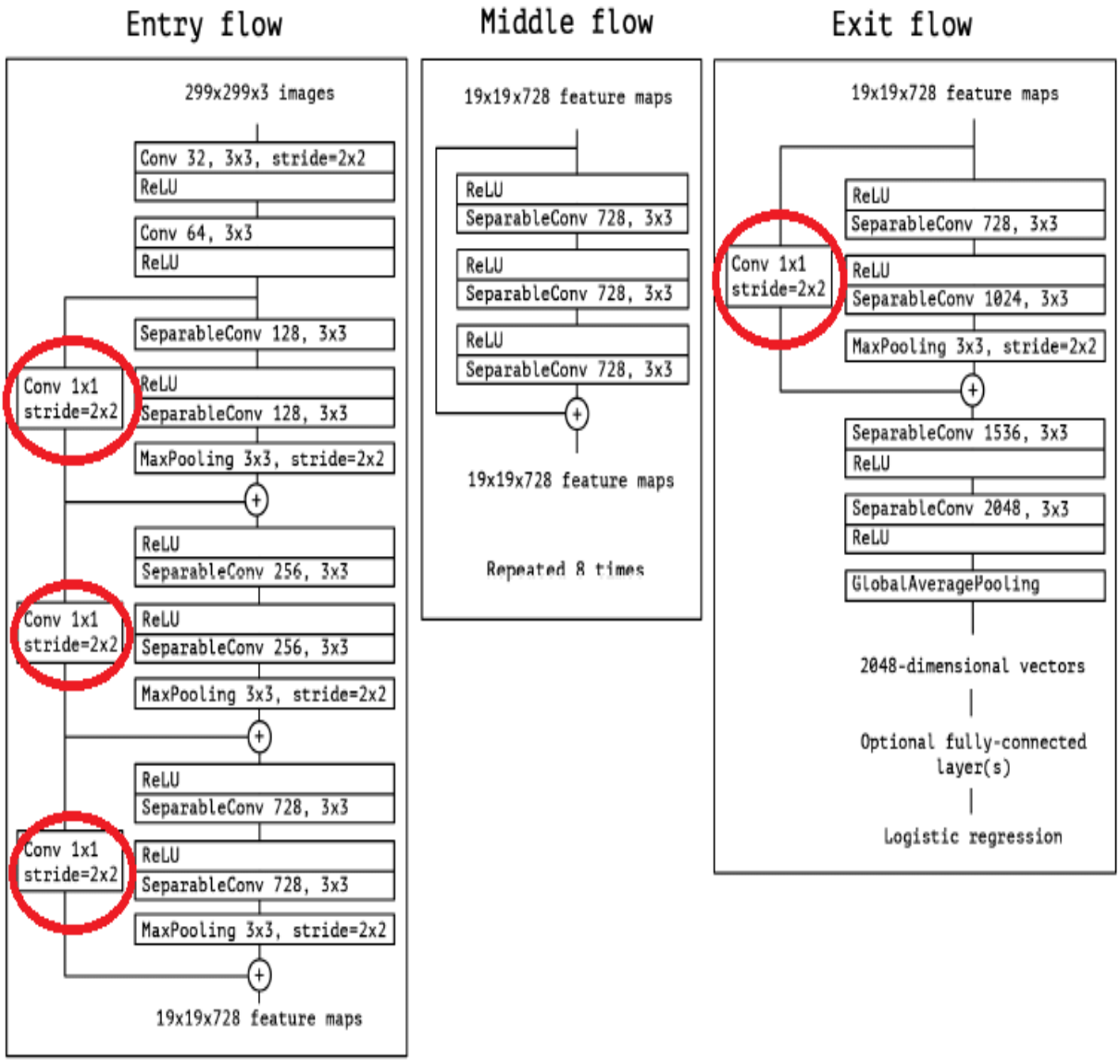

Figure 4.8: Xception Architecture [12]

4.6 Capsule Network

A capsule is a group of neurons whose activity vector represents the instantiating parameters of a specific type of entity such as an object or an object part. In other words a capsule is a small group of neurons that learns to detect a particular object with in a given region of image and outputs a vector whose length represents the estimated probability that the object is present, and whose orientation encodes the objects pose parameters. Similar to a regular neural network CapsNet is organized in multiple layers. But in the CapsNet the capsules in the lowest layers are called primary capsules and the capsules in the higher layer are called routing capsules that detects larger and more complex objects.

"Capsule network" is not the new term as it was first introduced in 2011 by Hinton et al [46]. They argued that CNNs have limited applicability to the "inverse graphics" problem and introduced a more robust architecture comprising several "capsules." However, they initially faced the same problem faced by CNNs – the limited performance of hardware and the lack of effective algorithms, which prevented practical application of capsule networks. CNNs thus remained dominant in this research field.

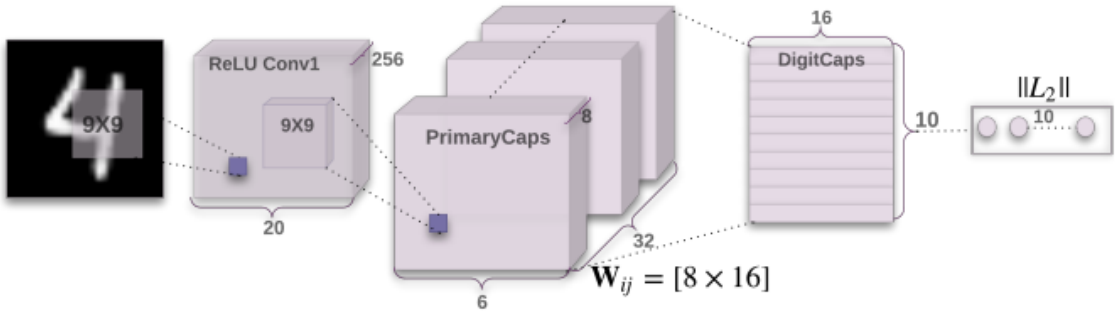

Figure 4.9: Capsule Network

A simple CapsNet architecture is shown in fig. 4.9. The architecture is shallow with only two convolutional layers and one fully connected layer. Conv1 has 256, 9×9 convolution kernels with a stride of 1 and ReLU activation. This layer converts pixel intensities to the activities of local feature detectors that are then used as inputs to the primary capsules. The primary capsules are the lowest level of multi-dimensional entities and, from an inverse graphics perspective, activating the primary capsules corresponds to inverting the rendering process. This is a very different type of computation than piecing instantiated parts together to make familiar wholes, which is what capsules are designed to be good at. The second layer (PrimaryCapsules) is a convolutional capsule layer with 32 channels of convolutional 8D capsules (i.e. each primary capsule contains 8 convolutional units with a 9×9 kernel and a stride of 2). Each primary capsule output sees the outputs of all 256×81 Conv1 units whose receptive fields overlap with the location of the center of the capsule. In total Primary

Capsules has [32 × 6 × 6] capsule outputs (each output is an 8D vector) and each capsule in the [6 × 6] grid is sharing their weights with each other. One can see PrimaryCapsules as a Convolution layer as its block non-linearity. The final Layer (DigitCaps) has one 16D capsule per digit class and each of these capsules receives input from all the capsules in the layer below.

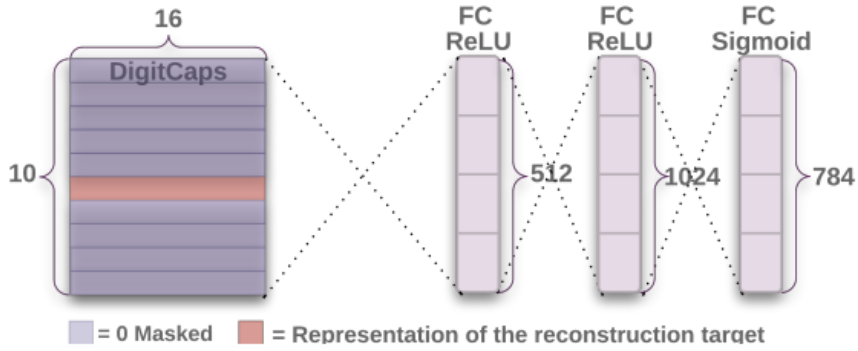

Figure 4.10: Capsul Network

There is routing only between two consecutive capsule layers (e.g. PrimaryCapsules and DigitCaps). Since Conv1 output is 1D, there is no orientation in its space to agree on. Therefore, no routing is used between Conv1 and PrimaryCapsules. All the routing logits are initialized to zero. Therefore, initially a capsule output is sent to all parent capsules (v0...v9) with equal probability.

In the below fig. 4.11 is the full architecture that is proposed by Huy H. Nguyen,Junichi Yamagishi (2019) [13] also experiment for Deepfake detection by Tolosana et al.(2020) [19]. We also experimented on this architecture. The capsule network includes several primary capsules and two output capsules ("real" and "fake"), as illustrated in fig. 4.11. There is no constraint on the number of primary capsules. Experiments demonstrated that a reasonably large number of primary capsules may improve network performance, but at the cost of more computation power. Three capsules are typically used for light networks (which require less memory and computation) and ten capsules are typically used for full ones (which require more memory and computation but provide better performance). While it is not necessary to use the same architecture for the primary capsules, we used the same design for all primary capsules to simplify the discussion. Each primary capsule is divided into three parts:

- a 2D convolutional part
- a statistical pooling layer and
- a 1D convolutional part

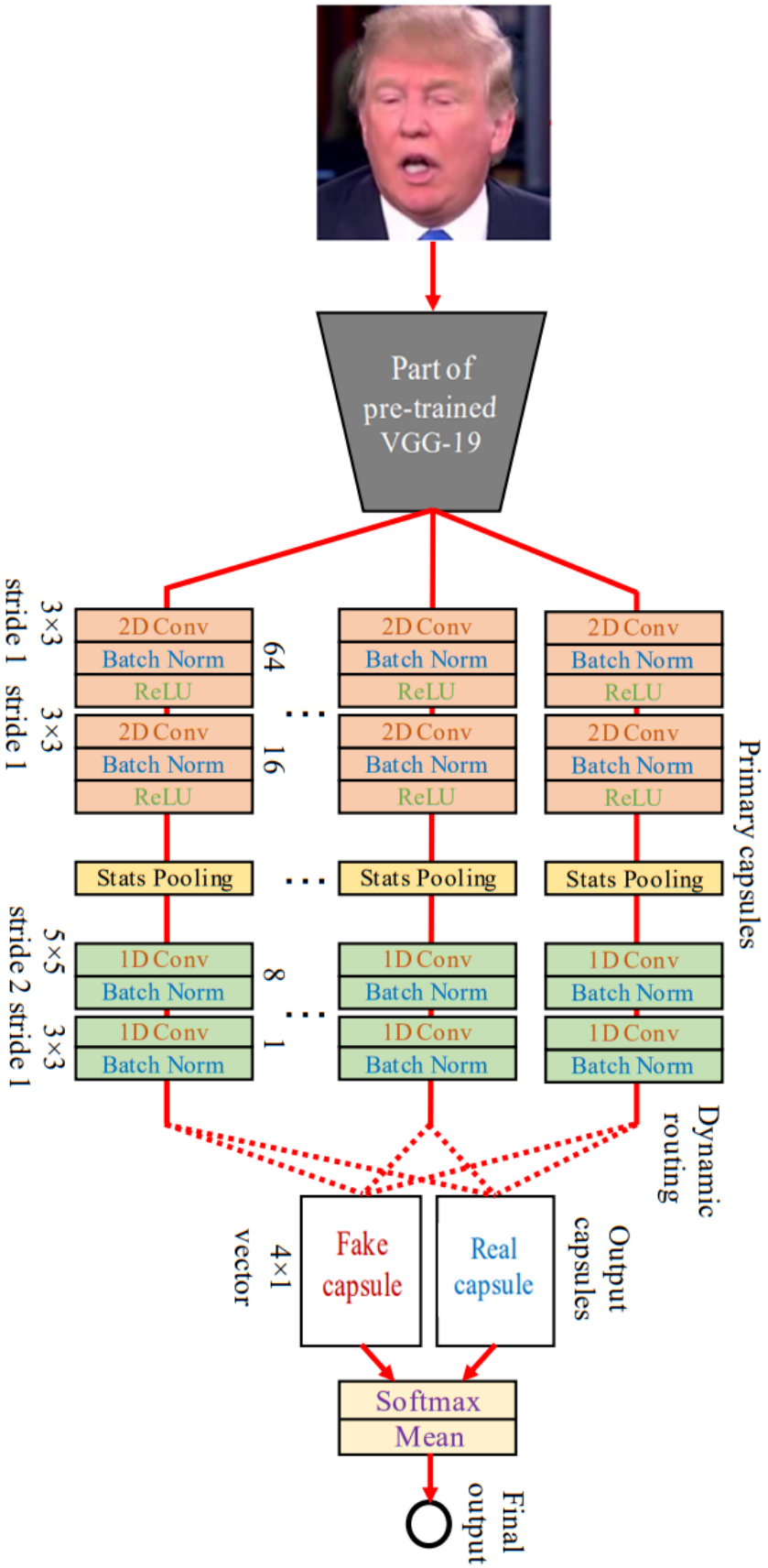

Figure 4.11: Capsule-Forensics architecture [13]

4.7 I3D

I3D incorporates sets of RGB frames as input. It replaces 2D convolutional layers of the original Inception model by 3D convolutions for spatio-temporal modeling and inflates pre-trained weights of the Inception model on ImageNet as its initial weights. Results showed that such inflation has the ability to improve 3D models. [17] Inflated 3D ConvNet (I3D) that is based on 2D ConvNet inflation: filters and pooling kernels of very deep image classification ConvNets are expanded into 3D, making it possible to learn seamless spatio-temporal feature extractors from video while leveraging successful ImageNet architecture designs and even their parameters.An I3D model based on Inceptionv1 obtains performance far exceeding the state-of-theart, after pre-training on Kinetics. [41]

The temporal component is one of the key distinctions between information contained in a single image and information contained in a movie. As a result, deep learning model designs have been improved to incorporate 3D processing to further process temporal data.

Researchers from DeepMind and the University of Oxford presented the I3D model in a paper titled "Quo Vadis, Action Recognition? The Kinetics Dataset and a New Model" [41]. The subject of this research is a new architecture that is presented alongside a comparison of earlier methods for action recognition in videos. In their method, all the filters and pooling kernels are inflated from a 2D architecture. They introduce a new dimension that must be taken into account by inflating them; in this example, that dimension is time. When inflated, filters that are square N x N in 2D models become cubic N x N x N. [14]

As can be seen by the diagram, the beginning of the network uses asymmetrical filters for max-pooling, maintaining time while pooling over the spatial dimension. It is not until later in the network that they run convolutions and pooling that includes the time dimension. The inception module is commonly used in 2D networks and is out of the scope of this article. In summary however, it is an approximation of an optimal local sparse structure. It also processes spatial (and time in this case) information at various scales and then aggregates the results. This module was motivated to allow the network to grow “wider” instead of “deeper”. The 1x1x1 convolution is used to reduce the number of input channels before the larger 3x3x3 convolutions, also making it less computationally expensive than the alternative. [14]

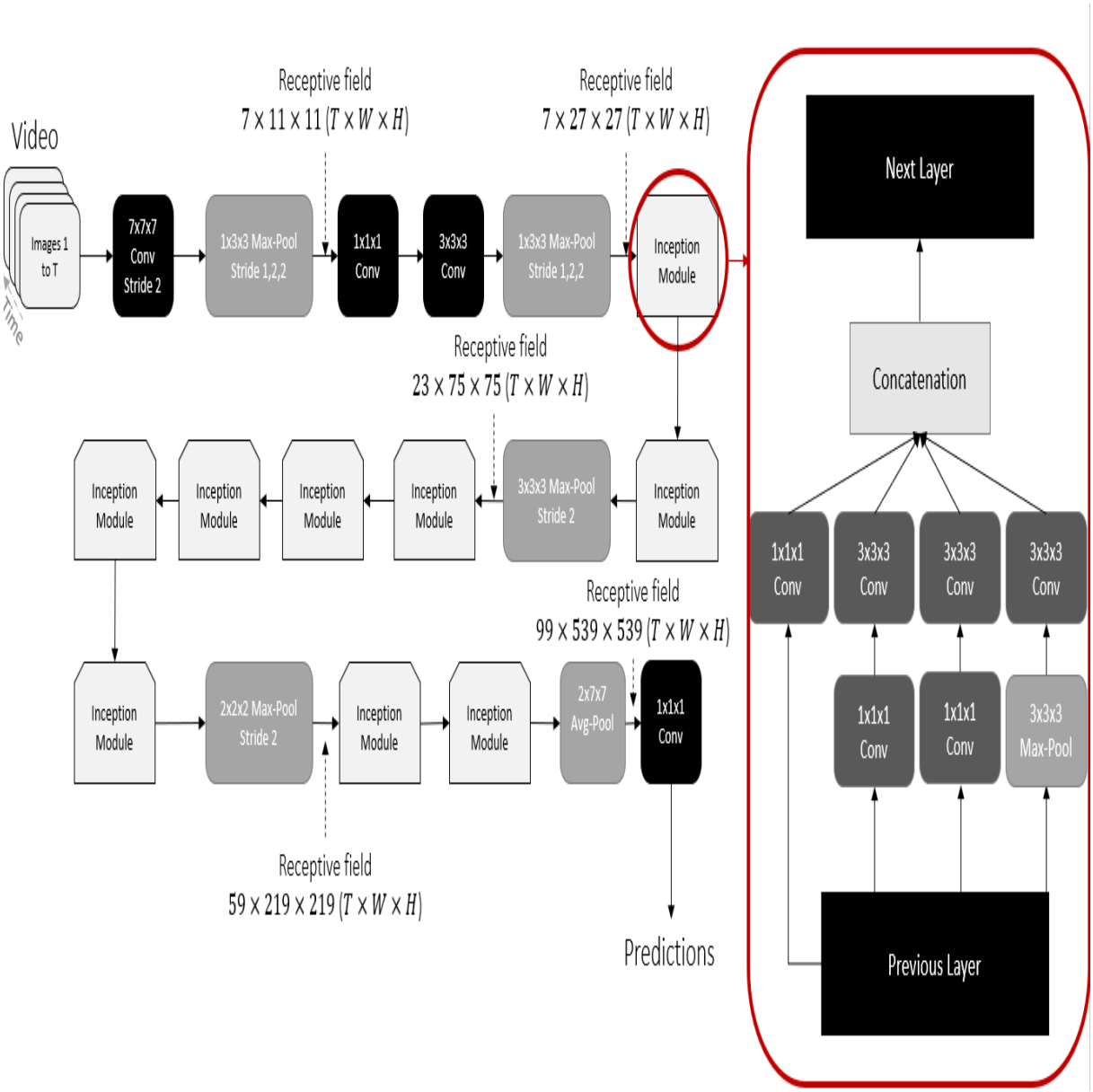

Figure 4.12: Image by author, adapted from Carreira and Zisserman (2017) [14]

Chapter 5

Related Works

Under the age of computer vision and deep learning technology, a new emerging techniques has introduced that anyone can make highly realistic but fake videos, images even can manipulates the voices. This technology is widely known as Deepfake Technology. Though it seems interesting techniques to make fake videos or image of something or some individuals but it could spread as misinformation via internet. Deepfake contents could be dangerous for individuals and for our communities, organizations, countries etc also. As Deepfake content creation involve a high level expertise with combination of several algorithms of deep learning, it seems almost real and difficult to differentiate. A wide range of articles have been examined to understand Deepfake technology more extensively. We have examined several articles to find some insights such as what is Deepfake, who are responsible for this, is there any benefits of Deepfake and what are the challenges of this technology. We have also examined several creation and detection techniques. Our study revealed that though Deepfake is a threat to our societies, proper measures and strict regulations could prevent this. A very recent survey has revisited image and video manipulation approaches and early detection efforts. [17]

- Generation of images and videos: Generative adversarial networks (GANs) [47] have enabled a set range of face manipulations including identity, facial attributes, and facial expressions.

- Deepfake detection: While some manipulation detection-methods are image-based, other approaches are video-based or combine audio-and-video detection. Some video-based approaches might outperform image-based ones in terms of performance, such approaches are only applicable to specific types of attacks. Image-based approaches are general-purpose detectors, for instance, the algorithms proposed by Fridrich and-Kodovsky [48] is applicable to both steganalysis and facial reenactment video detection.

- Adversarial training and detection: An adversarial example is one in which a machine learning model makes a false prediction as a instance with small, intentional feature perturbations. Adversarial examples are inputs to machine learning models can employ adversarial examples as inputs when the attacker intends for the model to fail. Adversary detection approaches attempt to verify the truthfulness of samples.

We did a search on google scholar and found most cited and recent paper named “Deepfakes & Beyond (2020)” [25]. In “Deepfakes & Beyond (2020)” [25] paper there are total 17 method mentioned for Deepfakes detection. We worked with 4 of them. They are marked in Figure 5.1

Study	Method	Classifiers	Best Performance	Databases
Korshunov and Marcel (2018) [1]	Audio-Visual Features	PCA+RNN	**EER = 3.3%**	**DeepfakeTIMIT (LQ)**
		PCA+LDA, SVM	**EER = 8.9%**	**DeepfakeTIMIT (HQ)**
Matern *et al.* (2019) [89]	Visual Features	Logistic Regression MLP	AUC = 85.1%	Own
			AUC = 70.2%	*UADFV*
			AUC = 77.0%	*DeepfakeTIMIT (LQ)*
			AUC = 77.3%	*DeepfakeTIMIT (HQ)*
			AUC = 78.0%	*FF++ / DFD*
			AUC = 66.2%	*DFDC Preview*
			AUC = 55.1%	*Celeb-DF*
Yang *et al.* (2019) [90]	Head Pose Features	SVM	AUC = 89.0%	UADFV
			AUC = 55.1%	*DeepfakeTIMIT (LQ)*
			AUC = 53.2%	*DeepfakeTIMIT (HQ)*
			AUC = 47.3%	*FF++ / DFD*
			AUC = 55.9%	*DFDC Preview*
			AUC = 54.6%	*Celeb-DF*
Agarwal and Farid (2019) [91]	Head Pose and Facial Features	SVM	AUC = 96.3%	Own (FaceSwap, HQ)
Jung *et al.* (2020) [92]	Eye Blinking	Distance	Acc. = 87.5%	Own
Li *et al.* (2019) [26], [93]	Face Warping Features	CNN	AUC = 97.7%	UADFV
			AUC = 99.9%	**DeepfakeTIMIT (LQ)**
			AUC = 99.7%	**DeepfakeTIMIT (HQ)**
			AUC = 93.0%	*FF++ / DFD*
			AUC = 75.5%	*DFDC Preview*
			AUC = 64.6%	*Celeb-DF*
Afchar *et al.* (2018) [94]	Mesoscopic Features	CNN	Acc. = 98.4%	Own
			AUC = 84.3%	*UADFV*
			AUC = 87.8%	*DeepfakeTIMIT (LQ)*
			AUC = 68.4%	*DeepfakeTIMIT (HQ)*
			Acc. $\simeq$ 90.0%	FF++ (DeepFake, LQ)
			Acc. $\simeq$ 94.0%	FF++ (DeepFake, HQ)
			Acc. $\simeq$ 98.0%	FF++ (DeepFake, RAW)
			Acc. $\simeq$ 83.0%	FF++ (FaceSwap, LQ)
			Acc. $\simeq$ 93.0%	FF++ (FaceSwap, HQ)
			Acc. $\simeq$ 96.0%	FF++ (FaceSwap, RAW)
			AUC = 75.3%	*DFDC Preview*
			AUC = 54.8%	*Celeb-DF*
Zhou *et al.* (2018) [95]	Steganalysis Features + Deep Learning Features	CNN SVM	*AUC = 85.1%*	*UADFV*
			AUC = 83.5%	*DeepfakeTIMIT (LQ)*
			AUC = 73.5%	*DeepfakeTIMIT (HQ)*
			AUC = 70.1%	*FF++ / DFD*
			AUC = 61.4%	*DFDC Preview*
			AUC = 53.8%	*Celeb-DF*
Rössler *et al.* (2019) [12]	Mesoscopic Features Steganalysis Features Deep Learning Features	CNN	Acc. $\simeq$ 94.0%	FF++ (DeepFake, LQ)
			Acc. $\simeq$ 98.0%	**FF++ (DeepFake, HQ)**
			Acc. $\simeq$ 100.0%	**FF++ (DeepFake, RAW)**
			Acc. $\simeq$ 93.0%	FF++ (FaceSwap, LQ)
			Acc. $\simeq$ 97.0%	**FF++ (FaceSwap, HQ)**
			Acc. $\simeq$ 99.0%	**FF++ (FaceSwap, RAW)**
Nguyen *et al.* (2019) [96]	Deep Learning Features	AE + Multi-Task Learning	*AUC = 65.8%*	*UADFV*
			AUC = 62.2%	*DeepfakeTIMIT (LQ)*
			AUC = 55.3%	*DeepfakeTIMIT (HQ)*
			AUC = 76.3%	*FF++ / DFD*
			EER = 15.1%	FF++ (FaceSwap, HQ)
			AUC = 53.6%	*DFDC Preview*
			AUC = 54.3%	*Celeb-DF*
Nguyen *et al.* (2019) [97]	Deep Learning Features	Capsule Networks	*AUC = 61.3%*	*UADFV*
			AUC = 78.4%	*DeepfakeTIMIT (LQ)*
			AUC = 74.4%	*DeepfakeTIMIT (HQ)*
			AUC = 96.6%	*FF++ / DFD*
			AUC = 53.3%	*DFDC Preview*
			AUC = 57.5%	*Celeb-DF*
Dang *et al.* (2019) [17]	Deep Learning Features	CNN + Attention Mechanism	AUC = 99.4%	DFFD
			EER = 3.1%	
Dolhansky *et al.* (2019) [83]	Deep Learning Features	CNN	**Precision = 93.0%**	**DFDC Preview**
			Recall = 8.4%	
Wang and Dantcheva (2020) [98]	Deep Learning Features	3DCNN	**TCR = 95.13%**	**FF++ (DeepFake, LQ)**
			TCR = 92.25%	**FF++ (FaceSwap, LQ)**
Güera and Delp (2018) [99]	Image + Temporal Features	CNN + RNN	Acc. = 97.1%	Own
Sabir *et al.* (2019) [98]	Image + Temporal Features	CNN + RNN	**AUC = 96.9%**	**FF++ (DeepFake, LQ)**
			AUC = 96.3%	**FF++ (FaceSwap, LQ)**
Tolosana *et al.* (2020) [100]	Facial Regions Features	CNN	**AUC = 100.0%**	**UADFV**
			AUC = 99.4%	**FF++ (FaceSwap, HQ)**
			AUC = 91.0%	**DFDC Preview**
			AUC = 83.6%	**Celeb-DF**

Figure 5.1: Entire face synthesis: Comparison of different State-Of-The-Art detection approaches. The best results achieved for each public database are remarked in bold. Results in italics indicate that they were not provided in the original work. AUC = Area Under The Curve, ACC. = Accuracy, EER = Equal Error Rate.

5.1 FaceForensics++: Learning to Detect Manipulated Facial Images (2019)

FaceForensics++: Learning to Detect Manipulated Facial Images (2019) is paper proposed by Li et al. (2019) [15]. The rapid progress in synthetic image generation and manipulation has now come to a point where it raises significant concerns for the implications towards society. At best, this leads to a loss of trust in digital content, but could potentially cause further harm by spreading false information or fake news. This paper examines the realism of state-of-the-art image manipulations, and how difficult it is to detect them, either automatically or by humans. To standardize the evaluation of detection methods, they proposed an automated benchmark for facial manipulation detection. In particular, the benchmark is based on Deepfakes, Face2Face [34], FaceSwap [35] and NeuralTextures [36] as prominent representatives for facial manipulations at random compression level and size. The benchmark is publicly available and contains a hidden test set as well as a database of over 1.8 million manipulated images. This dataset is over an order of magnitude larger than comparable, publicly available, forgery datasets. Based on this data, the authors performed a thorough analysis of data-driven forgery detectors. They showed that the use of additional domain specific knowledge improves forgery detection to unprecedented accuracy, even in the presence of strong compression, and clearly outperforms human observers.

In this paper, author showed that we significantly surpass human observers by being able to detect such changes automatically and dependably. Author took advantage of recent developments in deep learning, specifically the ability to pick up extremely powerful image features using convolutional neural networks (CNNs). Author tackled the detection problem by training a neural network in a supervised fashion.

They presented an automated benchmark that takes into account the four manipulation methods in a realistic scenario, i.e., with random compression and random dimensions, as the field of digital media forensics lacks a benchmark for forgery detection. They evaluated both the ir forgery detection pipeline and the state-of-the-art detection techniques using this benchmark, which considers the restricted field of facial manipulation techniques. This paper makes the following contributions:

- an automated benchmark for facial manipulation detection under random compression for a standardized comparison, including a human baseline.
- a novel large-scale dataset of manipulated facial imagery composed of more than 1.8 million images from 1,000 videos with pristine (i.e., real) sources and target ground truth to enable supervised learning.
- an extensive evaluation of state-of-the-art hand-crafted and learned forgery detectors

in various scenarios.

- a state-of-the-art forgery detection method tailored to facial manipulations.

Compression and resizing techniques are well known for laundering manipulation traces from the data. These fundamental procedures are commonplace in real-world circumstances, such as when photographs and videos are posted to social media, one of the most significant application fields for forensic analysis. In order to cover such actual scenarios, our dataset is made up of wild videos that have been manupulated and compressed at various quality levels. Researchers may be able to benchmark their approaches and improve forgery detectors for facial imaging with the use of such a huge and varied dataset that is available.

Authors built a database containing more than 1.8 million images from 4000 fake videos, an order of magnitude larger than existing datasets.

Database: They used two computer graphics-based approaches (Face2Face [34] and FaceSwap [35]) and two learning based approaches (Deepfakes and NeuralTextures [36]). This four methods require source and target actor video pairs as input. The final output of each method is a video composed of generated images.

Postprocessing - Video Quality: To create a realistic setting for manipulated videos, author generated output videos with different quality levels, similar to the video processing of many social networks. Since raw videos are rarely found on the internet. So they compressed the videos using the H.264 codec, which is widely used by social networks or video sharing websites. They used a light compression denoted by HQ (constant rate quantization parameter equal to 23) to generate high quality videos which is visually nearly lossless. Using a quantization of 40 , Low quality videos (LQ) are produced.

Author considered fake detection as a binary classification problem per frame of the manipulated videos. The following sections show the results of manual and automatic counterfeit detection. For all experiments, we split the dataset into a fixed training, validation, and testing set, each consisting of 720,140 and 140 videos, respectively.

Forgery Detection of Human Observers: They did Forgery Detection of Human Observers with 204 participants, composed mostly of computer science students.

Evaluation: In the Figure below fig. 5.2, author showed the results of their study on all quality levels, showing a correlation between video quality and the ability to detect fakes.

Automatic Forgery Detection Methods: Author showed that the classification based on XceptionNet outperforms all other variants in detecting fakes.

Detection Based on Steganalysis Features: Author evaluated detection from steganalysis features.

Detection Based on Learned Features: For learned features detection, they evaluated five network architectures known from the literature to solve the classification task.

Forgery Detection of GAN-based Methods: This paper showed that all detection approaches achieve a lower accuracy on the GAN-based NeuralTextures approach.

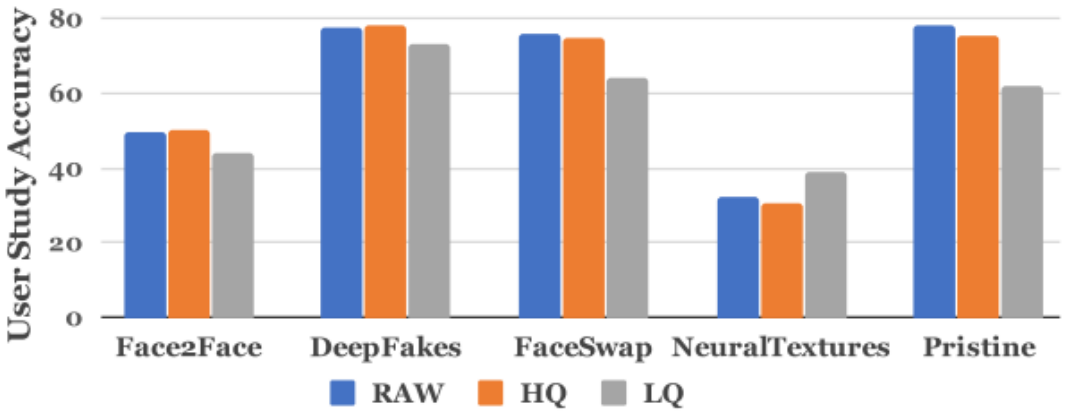

Figure 5.2: Forgery detection results of our user study with 204 participants. The accuracy is dependent on the video quality and results in a decreasing accuracy rate that is 68.69% in average on raw videos, 66.57% on high quality, and 58.73% on low quality videos. [15]

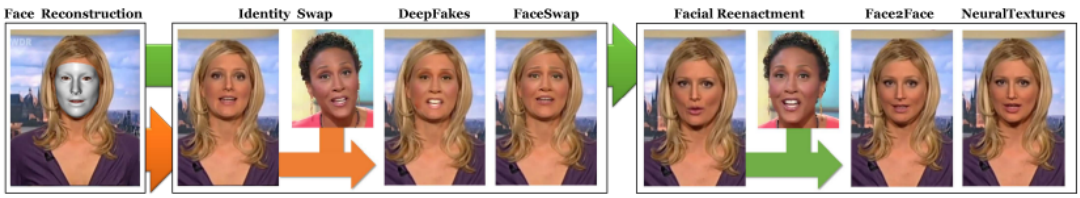

Figure 5.3: Advances in the digitization of human faces have become the basis for modern facial image editing tools. The editing tools can be split in two main categories: identity modification and expression modification. Aside from manually editing the face using tools such Photoshop, many automatic approaches have been proposed in the last few years. The most prominent and widespread identity editing technique is face swapping, which has gained significant popularity as lightweight systems are now capable of running on mobile phones. Additionally, facial reenactment techniques are now available, which alter the expressions of a person by transferring the expressions of a source person to the target. [15]

Proposed Method
XceptionNet (CNN + ImageNet)

Figure 5.4: Proposed Method

Used Acc. Method
Precision

Figure 5.5: Used Acc. Method

Used Database
DeepFakes (DF)
Face2Face (FF)
FaceSwap (FS)
NeuralTextures (NT)
Pristine Images (Real)

Figure 5.6: Used Database

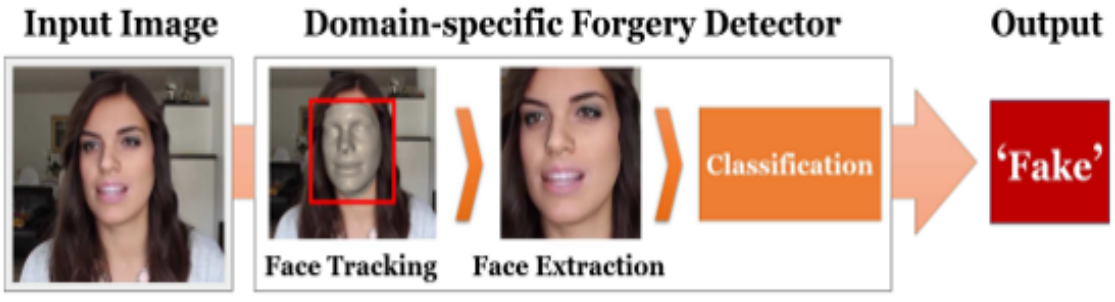

Figure 5.7: Our domain-specific forgery detection pipeline for facial manipulations: the input image is processed by a robust face tracking method; we use the information to extract the region of the image covered by the face; this region is fed into a learned classification network that outputs the prediction. [15]

Accuracies	DF	F2F	FS	NT	Real	Total
Xcept. Full Image	74.55	75.91	70.87	73.33	51.00	62.40
Steg. Features	73.64	73.72	68.93	63.33	34.00	51.80
Cozzolino *et al.*	85.45	67.88	73.79	78.00	34.40	55.20
Rahmouni *et al.*	85.45	64.23	56.31	60.07	50.00	58.10
Bayar and Stamm	84.55	73.72	82.52	70.67	46.20	61.60
MesoNet	87.27	56.20	61.17	40.67	**72.60**	66.00
XceptionNet	**96.36**	**86.86**	**90.29**	**80.67**	52.40	**70.10**

Figure 5.8: Binary detection accuracy of their baselines when trained on all four manipulation methods. Besides the naive full image XceptionNet, all methods are trained on a conservative crop (enlarged by a factor of 1.3) around the center of the tracked face. [15]

While current state-of-the-art facial image manipulation methods exhibit visually stunning results, author demonstrated that they can be detected by trained forgery detectors. It is particularly exciting that also the challenging case of low-quality video can be solved by learning-based approaches, when humans and hand-crafted features exhibit difficulties. To train detectors using domain-specific knowledge, author introduced a novel dataset of videos of manipulated faces that exceeds all existing publicly available forensic datasets by an order of magnitude. In this paper they focused on the influence of compression to the detectability of state-of-the-art manipulation methods,proposing a standardized benchmark for follow-up work. All of the image data, trained models, and benchmark are already being used by other researchers and are all freely accessible. In particular, transfer learning is of high interest in the forensic community. As new manipulation methods appear by the day, methods must be developed that are able to detect fakes with little to no training data. Their database is already used for this forensic transfer learning task, where knowledge of one source manipulation domain is transferred to another target domain, as shown by Cozzolino et al [49]. The dataset and benchmark became a stepping stone for future research in the field of digital media forensics, and in particular with a focus on facial forgeries.

5.2 The Deepfake Detection Challenge(DFDC) Preview Dataset (2019)

The Deepfake Detection Challenge (DFDC) Preview Dataset (2019) is paper proposed by Dolhansky et al. (2019) [16]. In this paper, the authors introduced a preview of the Deepfake Detection Challenge (DFDC) dataset consisting of 5K videos featuring two facial modification algorithms. A data collection campaign has been carried out where participating actors have entered into an agreement to the use and manipulation of their likenesses in their creation of the dataset. Diversity in several axes (gender, skin-tone, age etc.) has been considered and actors recorded videos with arbitrary backgrounds, thus bringing visual variability. Finally, a set of specific metrics to evaluate the performance have been defined and two existing models for detecting Deepfakes have been tested to provide a reference performance baseline. The DFDC dataset preview can be downloaded at:*here.*

The umbrella term 'Deepfakes' refers to a wide variety of methods for face swapping and manipulation, including methods that use state-of-the-art techniques from computer vision and deep learning, as well as other, simpler methods

The Deepfake Detection Challenge (DFDC) was announced in September 2019 [50] .It was a joint effort between industry, academia, and civil society organizations to invigorate the research related to the detection of facial manipulation. The challenge includes a dataset made up of a large number of videos with human faces and labels showing whether or not those faces were generated using facial manipulation methods. The dataset is made freely available to the public for the creation, testing, and analysis of technique for detecting videos with manipulated faces. All of the videos in the dataset were produced through agreements with hired actors. Developers who are interested in taking on this challenge must request for access and accept the DFDC's terms of service. As part of the development process for the DFDC dataset, author introduced in this paper a preview dataset consisting of around 5000 videos (original and manipulated). Author described the properties of the starter dataset and share relevant benchmark results using existing Deepfake detection methods.

In this preview dataset, 74 percent of the population is female, 26 percent of the population is male, and 68 percent of the population is Caucasian, 20 percent African-American, 9 percent east Asian, and 3 percent south Asian. In order to publish the final DFDC dataset,This dataset was not created using any publicly accessible data or data from social media platforms.

A small group of 66 people were selected from the pool of crowdsourced actors for this first version of the DFDC dataset and divided into a train and a test dataset. Such was done to prevent face swaps between cross-sets. To create face swaps, two techniques (designated as methods A and B in the dataset);with the intention of representing the real adversarial

space of facial manipulation, no further details of the employed methods are disclosed to the participants. Several face swaps between subjects were calculated. Featuring similar looks, where each look was inferred from attributes of the face (skin tone, facial hair, glasses,etc.) After a given pairwise model was trained on two identities, author swapped each identity onto the other's videos. Hereafter, they refered to the identity in the base video as the "target" identity, and the identity of the face swapped onto the video as the "swapped" identity. In this preview DFDC dataset, all base and target videos are provided as part of the training corpus.

Dataset	Ratio tampered:original	Total videos	Source	Participants Consent
Celeb-DF [4]	1 : 0.51	1203	YouTube	N
FaceForensics [8]	1 : 1.00	2008	YouTube	N
FaceForensics++ [9]	1 : 0.25	5000	YouTube	N
DeepFakeDetection [11] (part of FaceForensics++)	1 : 0.12	3363	Actors	Y
DFDC Preview Dataset	1 : 0.28	5214	Actors	Y

Figure 5.9: Specs of the most relevant Deepfake datasets in the literature. [16]

Deepfakes are now much less common in organic traffic (as compared to unaltered videos) than would be expected based on the ratios for the datasets in [16] It is plausible that x y if we assume that the ratio of Deepfake to unaltered videos is 1: x in organic traffic and 1: y in a Deepfakes dataset. It is important to construct measures that represent these variances even though it is impractical to create a dataset that replicates the statistics of organic traffic. As a very approximate approximation of the precision that would be derived by evaluating on a dataset representative of organic traffic, we can define a weighted precision for a Deepfakes dataset.Assuming the ratios of unaltered to tampered Assuming the ratios of unaltered to tampered videos differ between a test dataset and organic traffic by a factor of

$$\alpha = x/y \tag{5.1}$$

we define weighted precision wP and (standard) recall R as

$$wP = \frac{TP}{TP + \alpha FP}, R = \frac{TP}{TP + \alpha FN} \tag{5.2}$$

where TP, FP, and FN signify true positives, false positives and false negatives.

To derive an initial baseline, author measured the performance of three simple detection models. The first model was a frame-based model which we denote as TamperNet. TamperNet is a small DNN (6 convolutional layers plus a 1 fully connected layer) trained to detect low-level image manipulations, such as cut-and-pasted objects or the addition of digital text

to an image, and although it was not trained only on Deepfake images, it performs well in identifying digitally-altered images in general (including face swaps). The other two models are the XceptionNet face detection and full-image models, trained on the FaceForensics data set. For these models, one frame was sampled per second of video.

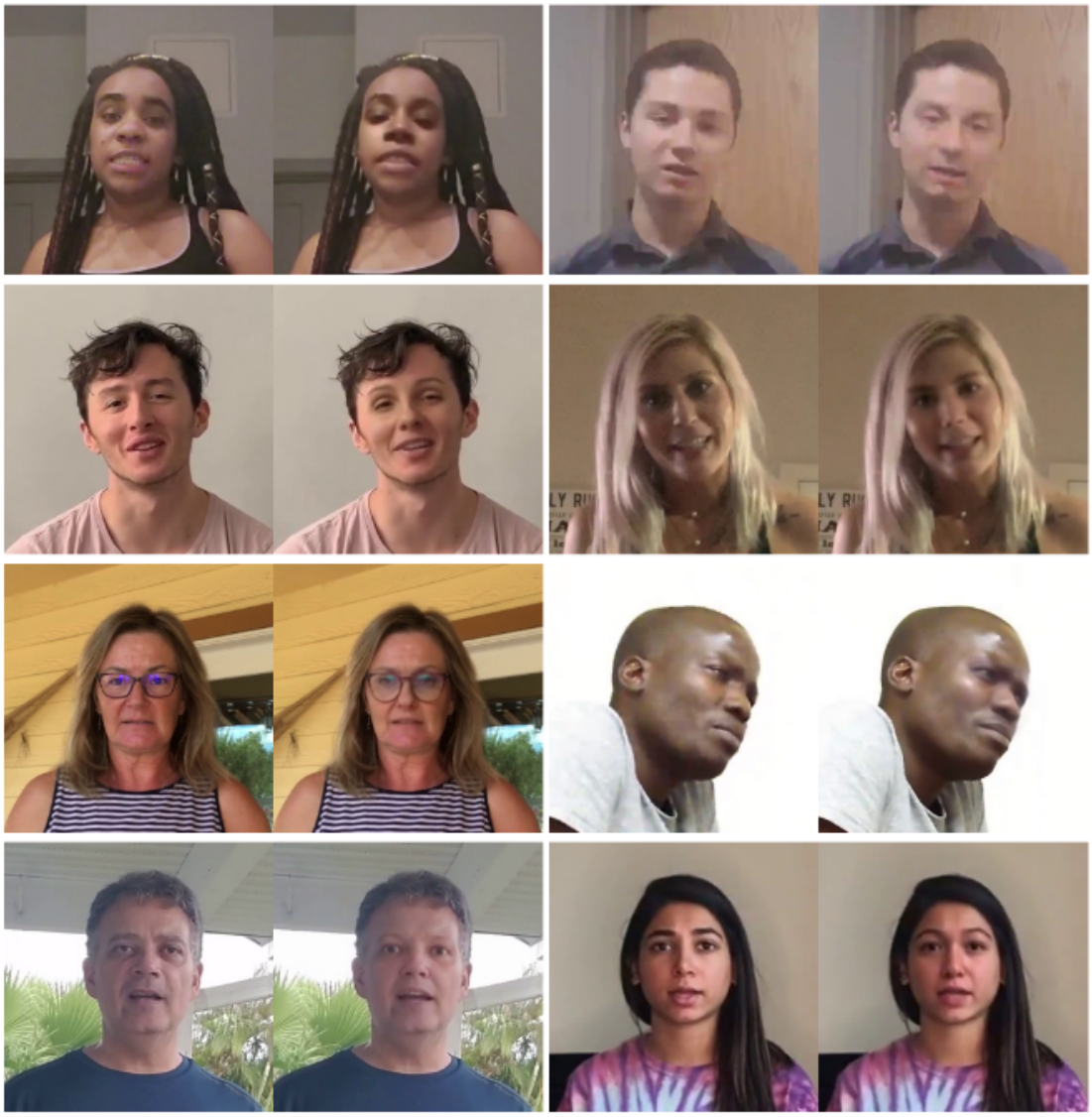

Figure 5.10: Some example face swaps from DFDC dataset. [16]

Proposed Method
TamperNet
XceptionNet Face Detection
XceptionNet Full Image Model

Figure 5.11: Proposed Method

Used Acc. Method
Precision
recall
log(weighted precision or WP)

Figure 5.12: Used Acc. Method

Used Database
DFDC (5214 videos)

Figure 5.13: Used Database

Method	Precision	Recall	log-WP
TamperNet	0.833	0.033	-3.044
XceptionNet (Face)	0.930	0.084	-2.140
XceptionNet (Full)	0.784	0.268	-3.352

Table 2. Video-level test metrics when optimizing for log(wP).

Figure 5.14: Video-level test metrics when optimizing for log(wP) [16]

Method	R=0.1	R=0.5	R=0.9
TamperNet	-2.796	-3.864	-4.041
XceptionNet (Face)	-1.999	-3.012	-4.081
XceptionNet (Full)	-3.293	-3.835	-4.081

Figure 5.15: Video-level log(wP) for various recall values [16]

5.3 A Video is Worth More Than 1000 Lies. Comparing 3DCNN Approaches for Detecting Deepfakes (2020)

A video is worth more than 1000 lies, comparing 3DCNN approaches for detecting Deepfakes (2020) is proposed by Wang and Dantcheva (2020) [17]. Manipulated images and videos have become increasingly realistic due to the tremendous progress of deep Convolutional Neural Networks (CNNs). This progress raises a number of social concerns related to the advent and spread of fake information and fake news. Such concerns necessitate the introduction of robust and reliable methods for fake image and video detection. Towards this in this work, the authors studied the ability of state of the art video CNNs including 3D ResNet, 3D ResNeXt, and I3D in detecting manipulated videos. The authors presented related experimental results on videos tampered by four manipulation techniques, as included in the FaceForensics++ dataset. They investigated three scenarios, where the networks are trained to detect

- All manipulated videos
- Separately each manipulation technique individually. Finally and deviating from previous works, they conducted cross-manipulation results, where they
- Detected the veracity of videos pertaining to manipulation-techniques not included in the train set.

The author distinguishes between two concerning situations: the first involves Deepfakes that are mistaken for the actual thing, while the second involves real videos that are mistakenly identified as fake, a situation known as "liar's dividend." Video evidence becomes highly questionable in light of these factors. Their research demonstrates the critical necessity for a deeper comprehension of manipulation techniques as well as the significance of creating algorithms that can successfully generalize to unknown manipulations. This work makes following two contributions:

- Author compared state-of-the art video based techniques in detecting Deepfakes. and their intuition is that current state of the art forgery detection techniques omit a pertinent clue, namely motion, by investigating only spatial information. author found out that generative models exhibit difficulties in preserving appearance throughout generated videos, as well as motion consistency. Hence, using 3DCNNs indeed outperforms state of the art image-based techniques was showed by the author.
- Author showed that such models trained on known manipulation techniques generalise poorly to tampering methods outside of the training set. Towards this, they

provided an evaluation, where train and test sets sets do not intersect with respect to manipulation techniques.

Author summarize for training setting.

- Raw data: It is interesting to note that the correct detection rates for all seven compared algorithms ranged between 97.03% and 99.26%. The highest score was obtained by XceptionNet.
- HQ: High quality compressed data was detected with rates ranging between 70.97% and 95.73% (XceptionNet).
- LQ: Intuitively low quality compressed data had the lowest detection rates with 55.98% to 81% (XceptionNet).They focused on the LQ-compression most challenging.It is a fact that reported detection rates applied to the analysis of a facial region 1.3 times the size of the cropped face. Lower accuracy was attained by analyzing the entire frame.

Database: The FaceForensics++ dataset [15] comprises of 1000 talking subjects, represented in 1000 real videos.Additionally, 4x1000 adversarial examples have been created using these 1000 real videos using the four manipulation schemes listed below:

- face-swap [35]
- Deepfakes
- face2face [34]
- neuraltextures [36]

Algorithms: Author selected three state-of-the-art 3D CNN methods.

- I3D incorporates sets of RGB frames as input. It replaces 2D convolutional layers of the original Inception model by 3D convolutions for spatiotemporal modeling and inflates pre-trained weights of the Inception model on ImageNet as its initial weights. Results showed that such inflation has the ability to improve 3D models.
- 3D ResNet and 3D ResNeXt are inspired by I3D, both extending initial 2D ResNet and 2D ResNeXt to spatio-temporal dimension for action recognition. Given the binary classification problem in this work, author replaced the prediction layer in all networks by a single neuron layer, which outputs one scalar value. All three networks have been pre-trained on the large-scale human action dataset Kinetics-400. author inherited the weights in the neural network models and further fine-tune the networks on the Faceforensics++ dataset in all their experiments.

Figure 5.16: Sample frames from the Faceforensics++ dataset From left to right: original source (large) and target (small) images, Deepfakes, face2face, faceswap, neuraltextures [17]

Author used PyTorch to implement models. The learning rate was 0.0003. For testing, author splitted each video into short trunks, each of temporal size of 250 frames. The final score assigned toeach test video is the average value of the scores of all trunks. The author conducted three studies using I3D, 3D ResNet, and 3D ResNext on the aforementioned manipulation techniques with the goals of training and detecting.

- All manipulation techniques
- Each manipulation technique separately
- Cross-manipulation techniques. To do this, author divided the training, testing, and validation sets in accordance with the Faceforensics++ dataset's protocol. In every trial, author presented the actual categorization rates (TCR).

Proposed Method
I3D
3D ResNet
3D ResNeXt

Figure 5.17: Proposed Method

Used Acc. Method
TRUE CLASSIFICATION RATE (TCR)

Figure 5.18: Used Acc. Method

Used Database
DeepFakes (DF)
Face2Face (FF)
FaceSwap (FS)
NeuralTextures (NT)

Figure 5.19: Used Database

In this paper, author compared three state-of-the-art 3D CNN four deep fake manipulation detection approaches. 3D ResNet and 3D ResNeXt were two of the three evaluated techniques. Additionally, author modified action recognition to create I3D. Experimental results showed that 3D video CNNs outperformed or performed at least similarly to image-based forgery detection algorithms. Further, a significant decrease in true classification rates in when detecting manipulated videos pertained to manipulation techniques not represented in the training set.

Train	Test	3D ResNet	3D ResNeXt	I3D
FS, DF, F2F	NT	64.29	68.57	66.79
FS, DF, NT	F2F	74.29	70.71	68.93
FS, F2F, NT	DF	75.36	75.00	72.50
F2F, NT, DF	FS	59.64	57.14	55.71

Figure 5.20: Detection of cross-manipulation methods, lq. true classification rates reported. df/Deepfakes, f2f/Face2face, fs/Face-swap, nt/Neuraltextures. [17]

5.4 Deepfakes Evolution: Analysis of Facial Regions and Fake Detection Performance (2020)

Deepfakes Evolution: Analysis of Facial Regions and Fake Detection Performance (2020) is proposed by Tolosana et al. (2020) [19]. Media forensics has attracted a lot of attention in the last years in part due to the increasing concerns around Deepfakes. Since the initial Deepfake databases from the 1st generation such as UADFV and FaceForensics++ up to the latest databases of the 2nd generation such as Celeb-DF and DFDC, many visual improvements have been carried out, making fake videos almost indistinguishable to the human eye.

An exhaustive analysis of both 1st and 2nd Deepfake generations using state-of-the-art fake detectors provided by the present study. Two different approaches are considered to detect fake videos:

- The traditional one followed in the literature and based on selecting the entire face as input to the fake detection system [6].
- A novel approach based on the selection of specific facial regions as input to the fake detection system.

The main contributions of this paper are as follow:

- An in-depth comparison in terms of performance among Identity Swap databases of the 1st and 2nd generation. In particular, two different state-of-the-art fake detectors are considered:
  1. Xception
  2. Capsule Network.
- An analysis of the discriminative power of the different facial regions between the 1st and 2nd generations, and also between fake detectors.

The analysis carried out in this study will benefit the research community for many different reasons:

- insights for the proposal of more robust fake detectors, e.g. through the fusion of different facial regions depending on the scenario:light conditions, pose variations, and distance from the camera.
- The improvement of the next generation of Deepfakes, focusing on the artifacts existing in specific facial regions.

Proposed evaluation framework of this paper

This evaluation framework are graphically summarises in fig. 5.25 . It comprises two main modules:

- facial region segmentation
- fake detection system

facial region segmentation: The segmentation of the entire face as input to the fake detection system and the segmentation of only particular facial regions are two separate approaches that are examined.

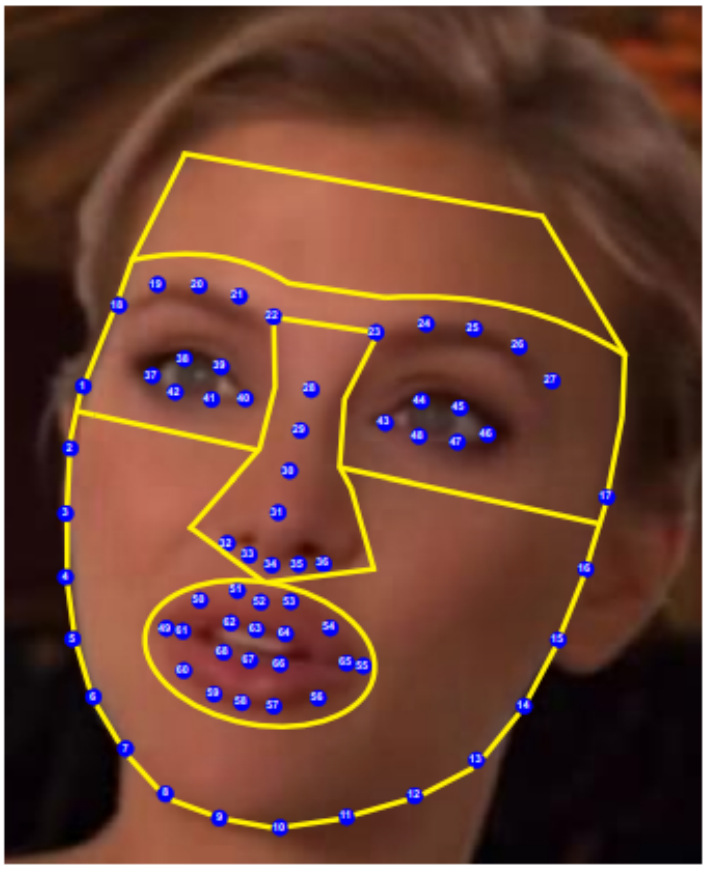

Figure 5.21: Example of the different facial regions (i.e., Eyes, Nose, Mouth and Rest) extracted using the 68 facial landmarks provided by OpenFace2 [18]

Fake Detection Systems: In this paper evaluation framework two different state-of-the-art fake detection approaches are considered :

- Xception :Xception [12] is a CNN architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions.

- Capsule Network:It is based on the current Capsule Networks, which require less parameters to train than classic CNN and combine traditional CNN with Capsule Networks .

The experimental study design takes into account four public datasets. In particular, two first-generation databases (UADFV and FaceForensics++) and two recent second-generation databases (Celeb-DF and DFDC).

1st Generation

Database	*Real Videos*	*Fake Videos*
UADFV (2018) [5]	49 (Youtube)	49 (FakeApp)
FaceForensics++ (2019) [15]	1,000 (Youtube)	1,000 (FaceSwap)

2nd Generation

Database	*Real Videos*	*Fake Videos*
Celeb-DF (2019) [6]	408 (Youtube)	795 (DeepFake)
DFDC Preview (2019) [7]	1,131 (Actors)	4,119 (Unknown)

Figure 5.22: Identity swap publicly available databases of the 1st and 2nd generations considered in our experimental framework. [19]

All databases in the paper have been divided into non-overlapping datasets, development (80% of the identities) and evaluation (20% of the identities).

For the UADFV database, for example, only the final evaluation of the models was considered for all real and fake videos relating to the identity of Donald Trump.

For the FaceForensics++ database, author considered 860 development videos and 140 evaluation videos per class (real/fake). It is proposed in [15], selecting different identities in each dataset (one fake video is provided for each identity).

For the DFDC Preview database, the same experimental protocol was followed which is proposed in [16] as the authors already considered this concern.

Finally, for the Celeb-DF database, author considered real/fake videos of 40 and 19 different identities for the development and evaluation datasets, respectively. [19]

Proposed Method
Xception
Capsule Network

Figure 5.23: Proposed Method

Used Acc. Method
Area Under the Curve (AUC)

Figure 5.24: Used Acc. Method

Used Database
UADFV (49 videos)
FaceForensics++ (1000 videos)
Celeb-DF (890 videos)
DFDC (1131 videos)

Figure 5.25: Used Database

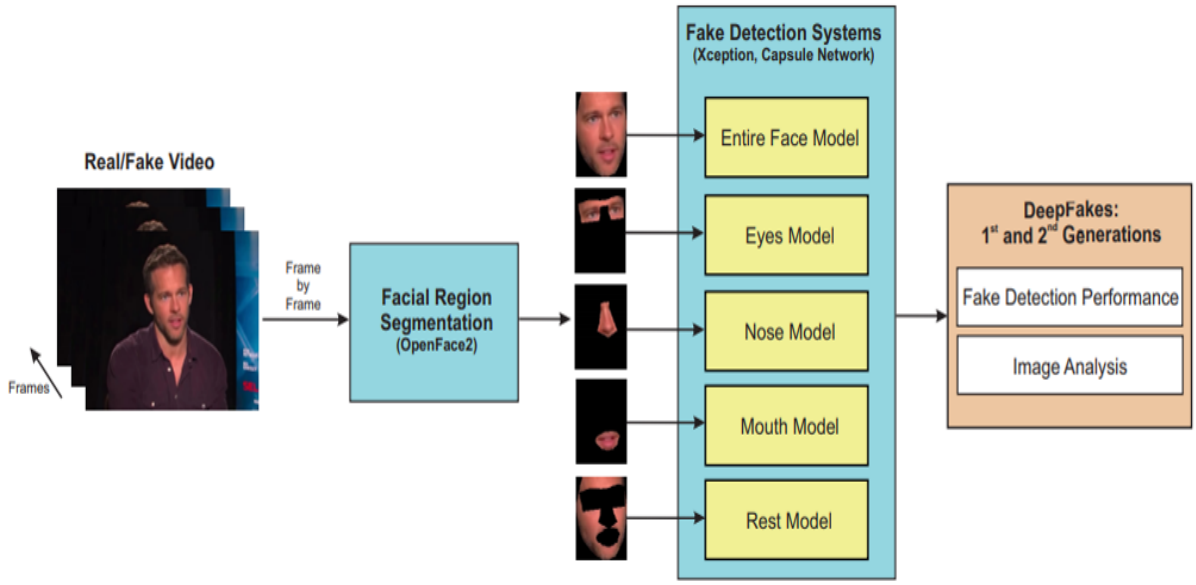

Figure 5.26: Architecture of our evaluation framework to analyse both facial regions and fake detection performance in Deepfake video databases of the 1st and 2nd generations. Two different approaches are studied: i) selecting the entire face as input to the fake detection system, and ii) selecting specific facial regions. [19]

Study	Method	Classifiers	AUC Results (%)			
			UADFV [5]	FF++ [15]	Celeb-DF [6]	DFDC [7]
Yang *et al.* [8]	Head Pose Features	SVM	89.0	*47.3*	*54.6*	*55.9*
Li *et al.* [6]	Face Warping Features	CNN	97.7	*93.0*	*64.6*	*75.5*
Afchar *et al.* [27]	Mesoscopic Features	CNN	*84.3*	84.7	*54.8*	*75.3*
Sabir *et al.* [13]	Image + Temporal Features	CNN + RNN	-	96.3	-	-
Dang *et al.* [19]	Deep Learning Features	CNN + Attention Mechanism	98.4	-	*71.2*	-
Present Study	Deep Learning Features	Xception [17]	**100**	99.4	**83.6**	**91.1**
		Capsule Network [22]	**100**	**99.5**	82.4	87.4

Figure 5.27: Comparison in terms of AUC (parcent) of different state-of-the-art fake detectors with the present study. The best results achieved for each database are remarked in bold. Results in italics indicate that the evaluated database was not used for training [19]

Two different approaches have been followed in this evaluation framework to detect fake videos:

- selecting the entire face as input to the fake detection system.
- selecting specific facial regions such as the eyes or nose, among others, as input to the fake detection system.

Regarding the effectiveness of fake detection, author focused to the very poor outcomes obtained in the most recent Deepfake video datasets of the 2nd generation with EER values of 20- 30%, as compared to the EER values of the 1st generation ranging from 1% to 3%. Additionally, author highlighted the significant improvements in facial area realism at the image level in Deepfakes of the 2nd generation, which achieved fake detection results between 24 and 44 percent EERs. These regions include the nose, mouth, and edge of the face. The analysis conducted by the researcher offers the research community helpful insights, e.g.:

- For the proposal of more robust fake detectors, e.g. through the fusion of different facial regions depending on the scenario: light conditions, pose variations and distance from the camera
- the improvement of the next generation of Deepfakes, focusing on the artifacts existing in specific facial regions.

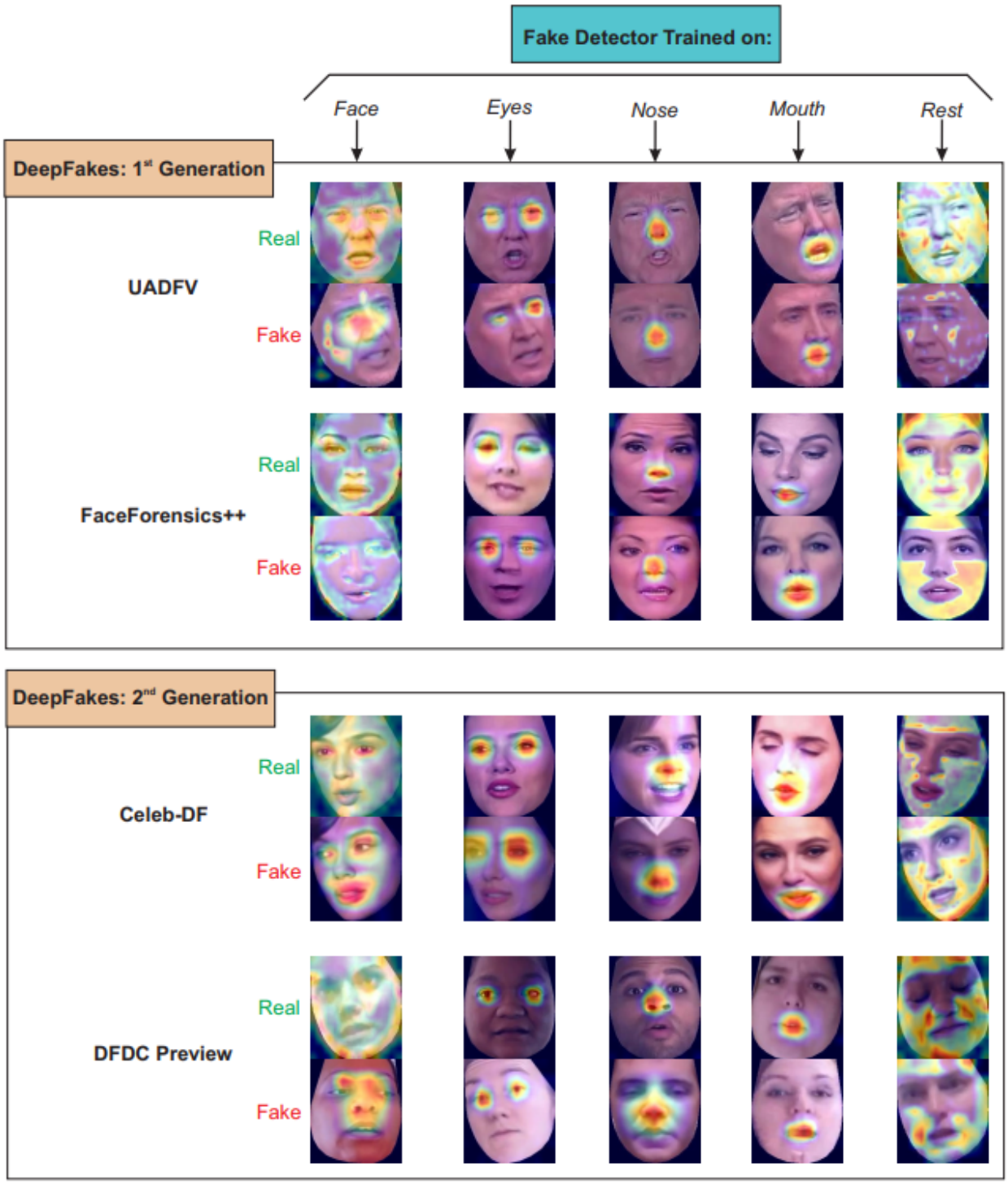

Figure 5.28: Real and fake image examples of the Deepfake video databases evaluated in the present paper with their corresponding Grad-CAM heatmaps, representing the facial features most useful for each fake detector (i.e., Face, Eyes, Nose, Mouth and Rest.) [19]

Chapter 6

Experiment and Result

In the previous chapter, we discussed briefly about the related work on deepfake detection where we explain four papers. In this chapter, we will explain about the experiments that we conducted during our research.

6.1 Experimental Setup

For conduction our experiment we used both tensorflow and pytorch. We did not use any IDE or software only notebook. But we also depended on online notebook called Google Colab [51], or "Colaboratory", allows us to write and execute Python in our browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Google Colab also provides TPU. TPU or Tensor Processing Unit is an AI accelerator application specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning, particularly using Google's own TensorFlow software [52]. Now Google Colab provide three services [53]. First one is limited resource where no monthly subscription is required called Colab. The second one is called Colab Pro where monthly payment is required. This package provide faster GPU's, more memory and a longer run time. The Final one is called Colab Pro+ where monthly subscription is also required but can get background execution feature. For our experiment we used the first one where we get:

- GPU Name: NVIDIA Tesla K80

- GPU memory: 12 GB
- Straight Run Time: 12 Hour
- Storage: 30 to 70 GB
- TPU: Not disclosed

We also used another online editor called Kaggle [54]. Kaggle [54] offers a no-setup, customizable, jupyter notebooks environment access GPUs at no cost to us and a huge repository of community published data & code. From Kaggle [54] notebook we can get

- GPU Name: NVIDIA Tesla P100
- GPU memory: 16GB
- Straight Run Time: 9 Hour
- Storage: 5 GB
- TPU: v3-8

We first experimented our code on online notebook so that we did not any package error problems. After confirming our code on online editor we train our models on full dataset. The specifications of the device that we used for training the model:

- Operating system: Windows 10 Home
- Processor: Intel(R) Core(TM) i5-10500 CPU @ 3.10GHz 3.10 GHz
- Installed RAM: 16.0 GB
- System type: 64-bit operating system, x64-based processor
- GPU Name: NVIDIA GeForce GTX 1660 SUPER
- GPU memory: 6 GB

6.2 Dataset Preprocessing

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format [55]. In our experiment, we used only video dataset named FaceForensics++ [15] which has 5000 video (1000 original video and 4000 fake video). For our experiment, we were needed to preprocessed our video dataset each time. Our video dataset preprocessing can be divided into following steps:

6.2.1 Extracting Frames from Videos

From each video in the dataset 350+ frame can be extracted. But we took only 10 frame per video because of hardware limitation. We took one frame per sec. As a result we can get 10 different pose of the subject in the frames. In Below is the example extracted frame from a video name “000”.

Figure 6.1: Extracted Frame from Original video "000"

Figure 6.2: Extracted Frame from Deepfake video "000"

Figure 6.3: Extracted Frame from Face2Face video "000"

Figure 6.4: Extracted Frame from Faceswap video "000"

Figure 6.5: Extracted Frame from NeuralTextures video "000"

We run a script in python to extract one frame per second. In below is the example code for real videos.

```
import os
import cv2
import glob
import math

folders = glob.glob('E:\\Thesis\\Database-NEW\\real')

frame_size = 10

videonames_list = []
for folder in folders:
    for f in glob.glob(folder+'/*.mp4'):
      videonames_list.append(f)
      print(f)
print('There are {} videos in Folder'.format(len(videonames_list)))


for i in range(0,len(videonames_list)):
  count = 0
  frame_count = 9
  cap = cv2.VideoCapture(videonames_list[i])
  frameRate = cap.get(5) #frame rate
  while(cap.isOpened()):
    frameId = cap.get(1) #current frame number
    ret, frame = cap.read()
    if (ret != True):
        break
    if (frameId % math.floor(frameRate) == 0):
        filename = 'E:\\Thesis\\Database-NEW\\real_frame\\'+ str(i) + '_' + str
   (count)+'.jpg'
        cv2.imwrite(filename, frame)
        count += 1
        if count > frame_count:
            break
  cap.release()
```

Listing 6.1: Videos to frames

6.2.2 Face Detection and Frame Cropping

For detecting deepfake we only need the face area of the frame. So, we crop the face area using face detection algorithm. In the below is the example of cropped face from the extracted frames.

Figure 6.6: Cropped image from Original frames

Figure 6.7: Cropped image from Deepfake frames

Figure 6.8: Cropped image from Face2Face frames

Figure 6.9: Cropped image from Faceswap frames

Figure 6.10: Cropped image from NeuralTexture frames

The face detection algorithm that we used is from OpenCV library called Cascade Classifier [56]. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. This algorithm is from Paul Viola and Michael Jones (2001) [57]. This function return four integer value or pixel value. Using those value we crop each frame and save it our destined location. We also resized the frame by 300 x 300. Below is the python script that we wrote to extract the cropped face from the frame.

```
import os
import cv2
from PIL import Image

image_path = "E:\Thesis\Database-NEW\real_frame"

count = 0
face_cascade = "D:\Py files\AAA - Thesis Codes\Face Detection\opencv-4.x\data\
    haarcascades\haarcascade_frontalface_alt.xml"
cascade = cv2.CascadeClassifier(face_cascade)

# Iterate through files
for f in [f for f in os.listdir(image_path) if os.path.isfile(os.path.join(
    image_path, f))]:

    img = cv2.imread(os.path.join(image_path, f))
    count += 1
    print(count)
    for i, face in enumerate(cascade.detectMultiScale(img, 1.25, 6)):
        x, y, w, h = face

        sub_face = img[y:y + h, x:x + w]

        if sub_face is None:
            continue
        else:
```

```
25              cv2.imwrite(os.path.join("E:\Thesis\Database-NEW\realcrop\", "{}{}.
    jpg".format(count, i)), sub_face)
26
27
28 f = r'E:\Thesis\Database-NEW\fake_crop'
29 for file in os.listdir(f):
30     f_img = f+"\"+file
31     img = Image.open(f_img)
32     img = img.resize((300,300))
33     img.save(f_img)
```

Listing 6.2: Function used for face detection and cropping the frame

6.2.3 Data Cleaning

After cropping the face we reviewed our dataset and found many cropped image without face or partial face in many frames. Although the detection method that we used is pretty well known but there is no method that can detect 100%. So, it detected some of the frame without face. We physically removed those frame from our dataset. We also found some frame with two or more faces. We also removed those frame as it would increase noise in our dataset.

6.2.4 Data Transformation

When we extracted frames from the videos, the size of each frame was 640 x 480. After cropping each frame we resize it according to the model input. Also our model took RGB frames so we did not convert our frame to any different type of image like gray scale or binary. In our dataset, We have two class.

1. Real
2. Fake

The dataset that we used FaceForensics++ [15] did not has any csv file or json file for labeling. The video was divided in different folder. Therefore, labeling the dataset, we just assign the variable Real or Fake depending on the folder we took the frames from. After that we used a function called `to_categorical()` [58] before pass the data to our models. This function is from tensorflow library. It takes three arguments:

- label or y
- number of classes

- data type

And it returns a binary matrix representation of the input. The class axis is placed last. We also wrote a text document for labeling each video physically. Each row of the text document contained a video name and a space and the label. Here we used

- 0 for real video and
- 1 for fake video

6.3 Dataset Splitting

In our experiment, we used only video dataset named FaceForensics++ [15] which has 5,000 video (1,000 original video and 4,000 fake video). The fake videos were made using four different methods:

- Deepfake: 1,000 videos
- Face2Face: 1,000 videos
- FaceSwap: 1,000 videos
- NeuralTexture: 1000 videos

Now from 5,000 videos we only worked with 1,600 videos because of hardware power limitation. They are:

- Real: 8,000 videos from 000 to 799
- Fake:
  - Deepfake: 200 videos from 000 to 199
  - Face2Face: 200 videos from 200 to 399
  - Faceswap: 200 videos from 400 to 599
  - NeuralTexture: 200 videos from 600 to 799

Before cleaning the dataset we got about 1,600 x 10 = 16,000 frames. After cleaning we got about 7,803 real frame and 7805 fake frames total 15,608 frames. Then, we split the dataset where we kept 70% frames for training, 10% for validation and 20% for testing. So we got

- Train total 11,248 frames where
  - Real: 5,613
  - Fake: 5,635
- Validation total 1257 frames where
  - Real: 628
  - Fake: 629
- Test total 3,103 frames where
  - Real: 1,562
  - Fake: 1,541

6.4 Experiment

There were many convolutional approaches that is taken over the year for deepfake detection. By studying the survey paper "DeepFakes & Beyond (2020)" [25], we studied many unique approaches that were presented. Although there were many common convolutional architectures, each authors presented different approaches. Now based on the result on each paper, we choose to implement four of them. Among them two of the approaches follows same convolutional architectures but different training approaches. The four approaches are:

1. XceptionNet (Li et al.(2019) [15])
2. XceptionNet (Face) (Dolhansky et al.(2019) [16])
3. I3D (Wang and Dantcheva(2020) [17])
4. Capsule Network (Tolosana et al.(2020) [19])

The authors employed different datasets and various accuracy methods to test these four convolutional architectures. However, we used the same dataset and the same accuracy evaluation approach to implement those four methods for the comparison study. Thus, we carried out four experiments.

6.4.1 Experiment 1

In this experiment, we implemented a deepfake detection model technique describe in Li et al.(2019) [15]. Here we experimented XceptionNet as the authors described. We also followed the training method the authors had followed. Although the author used pytorch we used keras to train our model. As the authors had suggested, first we pre-trained the network initializing the weights with "imagenet" for 3 epochs on training dataset. After training 3 epochs we remove the top most layer and add similar top layer and again trained for 15 epochs. After that we just evaluated our model on both training dataset and test dataset. Below is the hyperparameters value that we used for our experiment.

- Library: Tensorflow, Keras
- Model: XceptionNet
- Learning Rate: 0.0002
- Optimizer: Adam optimizer
- Epoch: 3 then 15
- Batch Size: 32 then 16
- Activation Function: Softmax

6.4.2 Experiment 2

In this experiment, we implemented a deepfake detection model technique describe in Dolhansky et al.(2019) [16]. In the paper, authors experimented XceptionNet with full frames and with cropped face frames. They get the best result for XceptionNet (face) with their DFDC dataset. So in this experiment we only implemented the XceptionNet (face). The training method were very simple in this case. We used the model from keras library and define the initial weight with imageNet. The input size of the image was same as the experiment 1. Below is the hyperparameters value that we used for our experiment.

- Library: Tensorflow, Keras
- Model: XceptionNet (Face)
- Learning Rate: 0.0002
- Optimizer: Adam optimizer

- Epoch: 10
- Batch Size: 64
- Activation Function: Softmax

6.4.3 Experiment 3

In this experiment, we implemented a deepfake detection model technique describe in Wang and Dantcheva(2020) [17]. The authors experiment with nine methods. Among them I3D got the best result. Now the architecture of I3D was got from Joao and andrew(2018) [41]. Joao and andrew(2018) [41] created an open-source toolbox called MMAction2 [59] for video understanding based on PyTorch. So as the authors experimented their experiment via MMAction2 [59], we also conducted our experiment using MMAction2 [59]. Now, to train the I3D we needed to write a config file to define the basic hyperparameter values and the paths to the dataset. Below is the hyperparameters value that we used for our experiment.

- Library: Pytorch
- Model: I3D
- Learning Rate: 0.0001
- Optimizer: SDG (Gradient Descent With Momentum) optimizer
- Epoch: 20
- Batch Size: 16
- Activation Function: Softmax

6.4.4 Experiment 4

In this experiment, we implemented a deepfake detection model technique describe in Tolosana et al.(2020) [19]. Here the authors experimented seven methods where capsule network got the best result. The authors used the capsule network from Huy H. Nguyen, Junichi Yamagishi (2019) [13]. We also used the same architecture the authors had used. The authors of Junichi Yamagishi (2019) [13] shared their source code on GitHub. We experiment with the same code with some adjustment. Below is the hyperparameters value that we used for our experiment.

- Library: Pytorch
- Model: Capsule Network
- Learning Rate: 0.0001
- Optimizer: Adam optimizer
- Epoch: 25
- Batch Size: 50
- Activation Function: Softmax

6.5 Result

For all four experiment we used 1600 videos where we took 10 frames per video. We used 1152 videos for training, 128 videos for validation and 320 videos for testing. After each experiment we evaluated our model for both training and test dataset. We used accuracy method to evaluated our models. First we tried the hyperparameter values used by the authors of the papers. Then, we again tried tuning the hyperparameter values but got the result shown in table 6.1. The outcome allows us to compare the models for our comparative study. The table 6.1 shows that the XceptionNet(Face) got the best accuracy for test dataset. Although Capsule Network and XceptionNet got the perfect accuracy for training dataset, it fell short on the test dataset.

Experiment	**Model**	**Ours**	
		Train Accuracy	**Test Accuracy**
1	XceptionNet	0.99	0.7
2	XceptionNet (Face)	1	0.99
3	I3D	0.85	0.62
4	Capsule Network	0.99	0.8

Table 6.1: Results from the experiments

We also compared our result with the authors results although they used different evaluation methods. In the table 6.2 we can see the results of authors and ours. In XceptionNet ours results is not change with the authors. Here Li et al. (2019) [15] used about 1,000 frames randomly selected from each video whereas we used about 11,520 frames to train the model. So we expected the accuracy should have been increases but it did not. We think that it happened because in author's dataset the fake videos were made using only one deepfake method. But in our dataset there is four type of deepfake videos. In our second experiment where we implemented Dolhansky et al. (2019) [16], got higher result then the authors.

We got too perfect result for Xception(face). We think increasing the test data will decrease the accuracy. In I3D we got the lowest results and also we could not get the result shown in Wang and Dantcheva (2020) [17]. We think that it is because the authors used 5,000 videos for training but we used only 1,600 videos for training. We can say that it will improve with the increase of training dataset. In Capsule Network we also did not get the result as the authors. Although the authors used about 2,000 videos for training they used only one type of fake videos. But in our experiment we used four different type of fake videos. Therefore, we think that with the increase of fake videos type Capsule Network will not perform well.

Paper	**Model**	**Authors Result**			**Ours**
		Precision	**TCR**	**AUC**	**Accuracy**
Li et al.(2019) [15]	XceptionNet	70			70
Dolhansky et al.(2019) [16]	XceptionNet (Face)	93			99
Wang and Dantcheva(2020) [17]	I3D		87		62
Tolosana et al.(2020) [19]	Capsule Network			99.5	80

Table 6.2: Comparing results with authors

Chapter 7

Conclusion

In our thesis, we compare studies on the identification of deepfakes the future of false information. In order to research the suggested detection methods, we looked through survey articles especially the most recent survey paper "DeepFake and Beyond(2020)" [25]. In this paper the authors write about 16 different approaches for deepFake detection. We decided to use four strategies based on the authors' experimental findings. For a fair comparison, we attempted to examine those techniques using the same dataset.

For the comparative study of those four methods, we conducted four experiment.

- Experiment 1: XceptionNet (Li et al.(2019) [15])
- Experiment 2: XceptionNet (Face) (Dolhansky et al.(2019) [16])
- Experiment 3: I3D (Wang and Dantcheva(2020) [17])
- Experiment 4: Capsule Network (Tolosana et al.(2020) [19])

Each of the experiment was conducted as the authors had suggested, even the preprocessing of the dataset. First we tried the same hyperparameters value as the authors. Then we tried tunning the hyperparameters value to get the highest accuracy. But ended up with the accuracy show in table Table 6.1.

The models that we got from our experiment performs better for XceptionNet (Face) (Dolhansky et al.(2019) [16]). XceptionNet (Li et al. (2019) [15]) and Capsule Network (Tolosana et al.(2020) [19]) got a good result in training dataset. But despite of tuning the hyperparameter values the models fell short on test dataset. Therefore, we can conclude that XceptionNet that proposed by Li et al.(2019) [15] got the best accuracy.

Chapter 8

Future Work

The evolution of modern technology has brought us many blessing. But it has make crime more easy. With the internet in our hand it has become almost impossible to identify the truth. In today's world most concerning issue is misinformation. We can see many fact checker company is emerging. Even fact checking is becoming a common job for the media. Most of the fact checking is done by simple google search. For checking image manipulation there are few software. To check image origin there are google image search. But for video manipulation there is none. With the advancement of deepfake videos it is becoming impossible to know if it is fake or real. This facial identity manipulation technique first uses facial recognition to crop the face, then trains two auto-encoder and one shared auto-encoder for source and target. Then the target is run through the source auto-encoder and stuck to images using Poisson image editing. With the advancement of generative adversarial network (GAN) this technique will improve more. So the only way is to develop an AI system to verify the videos.

Keeping that in our mind, we want to keep working on deepfake in the coming days. The experiment that we had some limitation. We want to overcome them in future. For finding the best accuracy we will train our models on a large dataset. Future work will involve the consideration of additional deepfake manipulation-techniques. Further, we want to develop an improved dataset for deepfake with all kind of manipulation technique. We also want to find the best Convolutions Neural Network(CNN) approach to detect manipulated videos.

References

[1] "4.5 years of GAN progress on face generation. https://t.co/kiQkuYULMC https://t.co/S4aBsU536b https://t.co/8di6K6BxVC https://t.co/UEFhewds2M https://t.co/s6hKQz9gLz pic.twitter.com/F9Dkcfrq8l." https://twitter.com/goodfellow_ian/status/1084973596236144640?s=20, note = Accessed: 2022-06-24.

[2] "Edition.cnn.com." https://edition.cnn.com/2021/08/06/tech/tom-cruise-deepfake-tiktok-company/index.html. Accessed: 2022-06-18.

[3] "Deepfake video of zelenskyy could be 'tip of the iceberg' in info war, experts warn." https://rb.gy/a43woz. Accessed: 2022-06-20.

[4] "What to do about deepfakes." https://dl.acm.org/doi/fullHtml/10.1145/3447255?casa_token=BSgCNgeOoYYAAAAA:BVK-rak0PkL87k6H72YwbNKnThfDjnXFzb_IN8LXVSmsv-Z2IMAVqil3ANTSYP4KClae9DX3qhzD4IM, note = Accessed: 2022-06-18.

[5] "Fake Obama created using AI video tool ,bbc news." https://youtu.be/AmUC4m6w1wo. Accessed: 2022-06-20.

[6] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," *CoRR*, vol. abs/1909.12962, 2019.

[7] "A Beginner's Guide to Neural Networks and Deep Learning." https://wiki.pathmind.com/neural-network. Accessed: 2022-06-25.

[8] "Machine Learning at Condé Nast, Part 1: A Neural Network Primer." https://technology.condenast.com/story/a-neural-network-primer. Accessed: 2022-06-25.

[9] "Depth-wise Convolution and Depth-wise Separable Convolution." https://rb.gy/jih31o. Accessed: 2022-06-25.

[10] "Depth-wise Convolution." https://bit.ly/3ymZcv1. Accessed: 2022-06-25.

[11] "A Basic Introduction to Separable Convolutions." https://rb.gy/iku4tl. Accessed: 2022-06-25.

[12] Francois, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[13] J. Y. H.H. Nguyen and I. Echizen, "Use of a capsule network to detect fake images and videos," in *"arXiv preprint arXiv:1910.12467*, 2019.

[14] "Understanding the Backbone of Video Classification: The I3D Architecture." https://rb.gy/2ywbuw. Accessed: 2022-06-25.

[15] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971, 2019.

[16] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. Canton-Ferrer, "The deepfake detection challenge (DFDC) preview dataset," *CoRR*, vol. abs/1910.08854, 2019.

[17] Y. Wang and A. Dantcheva, "A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes," in *2020 15Th IEEE international conference on automatic face and gesture recognition (FG 2020)*, pp. 515–519, IEEE, 2020.

[18] 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, IEEE, 2018.

[19] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, "Deepfakes evolution: Analysis of facial regions and fake detection performance," in *International Conference on Pattern Recognition*, pp. 442–456, Springer, 2021.

[20] "Deepfake." https://en.wikipedia.org/wiki/Deepfake. Accessed: 2022-06-19.

[21] "Deepfake History: When Was Deepfake Technology Invented?." https://deepfakenow.com/deepfake-history-when-invented/, note = Accessed: 2022-06-24.

[22] "Deepfakes: the hacked reality." https://www.newagebd.net/article/116541/deepfakes-the-hacked-reality, note = Accessed: 2022-06-24.

[23] J. T. Hancock and J. N. Bailenson, "The social impact of deepfakes," 2021.

[24] "Deepfake queen to deliver Channel 4 Christmas message." https://www.bbc.com/news/technology-55424730, note = Accessed: 2022-06-22.

[25] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[26] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, no. 11, 2019.

[27] "This PSA About Fake News From Barack Obama Is Not What It Appears buzzfeed.news." https://rb.gy/iyk5wy. Accessed: 2022-06-20.

[28] "Deepfake de Tom Cruise: pas pour le premier venu." https://www.sciencepresse.qc.ca/actualite/2021/03/08/deepfake-tom-cruise-pour-premier-venu.

[29] Y. Li, M. Chang, and S. Lyu, "In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking," *CoRR*, vol. abs/1806.02877, 2018.

[30] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics* (M. Tistarelli and M. S. Nixon, eds.), (Berlin, Heidelberg), pp. 199–208, Springer Berlin Heidelberg, 2009.

[31] "Contributing Data to Deepfake Detection Research." https://rb.gy/f0ohmk, note = Accessed: 2022-06-20.

[32] "faceswap-GAN github." https://github.com/shaoanlu/faceswap-GAN. Accessed: 2022-06-18.

[33] "Deepfake Detection Challenge Dataset." https://ai.facebook.com/datasets/dfdc/, note = Accessed: 2022-06-17.

[34] "Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, Matthias Nießner "Face2Face: Real-time Face Capture and Reenactment of RGB Videos" 2016." https://arxiv.org/abs/2007.14808.

[35] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," *CoRR*, vol. abs/2007.14808, 2020.

[36] "Justus Thies, Michael Zollhöfer, Matthias Nießner "Deferred Neural Rendering: Image Synthesis using Neural Textures" 2019." https://arxiv.org/abs/1904.12356.

[37] "Mac OS X Web Browser Automation and Webapp Testing Made Simple.." https://www.fakeapp.com, note = Accessed: 2022-06-23.

[38] "MarekKowalski/FaceSwap: 3D face swapping implemented in Python." htpps://github.com/MarekKowalski/FaceSwap/, note = Accessed: 2022-06-19.

[39] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[41] A. Z. Joao Carreira, "Quo vadis, action recognition? a new model and the kinetics dataset," in *" in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 6299–6308, 2017.

[42] "Explained: Neural networks." https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414. Accessed: 2022-06-25.

[43] "Neural Network Definition." https://rb.gy/0oyeol. Accessed: 2022-06-25.

[44] "Convolutional Neural Networks in Python with Keras." https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python. Accessed: 2022-06-25.

[45] "Facial Landmarks Detection Using Xception Net." https://rb.gy/ud5llo. Accessed: 2022-06-25.

[46] A. K. G E Hinton and S. D. Wang, "Transforming auto-encoders," in *in International Conference on Artificial Neural Networks (ICANN)*, 2011.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[48] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[49] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "Forensictransfer: Weakly-supervised domain adaptation for forgery detection," *CoRR*, vol. abs/1812.02510, 2018.

[50] "M. Schroepfer. (2019) Creating a data set and a challenge for deepfakes facebook ai blog. [online]." https://ai.facebook.com/blog/deepfake-detection-challenge/. Accessed: 2022-06-21.

[51] "Google Colab." https://rb.gy/a5jban. Accessed: 2022-06-25.

[52] "Tensor Processing Unit." https://rb.gy/l3gamg. Accessed: 2022-06-24.

[53] "Google Colab Services." https://rb.gy/p13dfk. Accessed: 2022-06-25.

[54] "Kaggle." https://www.kaggle.com/. Accessed: 2022-06-2.

[55] "Data Preprocessing in Data Mining." https://rb.gy/dhmlar. Accessed: 2022-06-24.

[56] "Cascade Classifier." https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. Accessed: 2022-06-2.

[57] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pp. 1–1, 2001.

[58] "tf.keras.utils to-categorical TensorFlow Core v2.9.1." https://rb.gy/wahdyl. Accessed: 2022-06-25.

[59] "open-mmlab/mmaction2: OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark." https://github.com/open-mmlab/mmaction2. Accessed: 2022-06-24.

[60] "Source code for the experiment." https://drive.google.com/drive/folders/1BpWEPfZe8-beNb9J5d6kuOA1c8e4KBe5?usp=sharing. Accessed: 2022-06-25.

Appendix A

Resources

We have uploaded the resources and source on our google drive [60]

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Saturday 4th March, 2023 at 7:10pm.