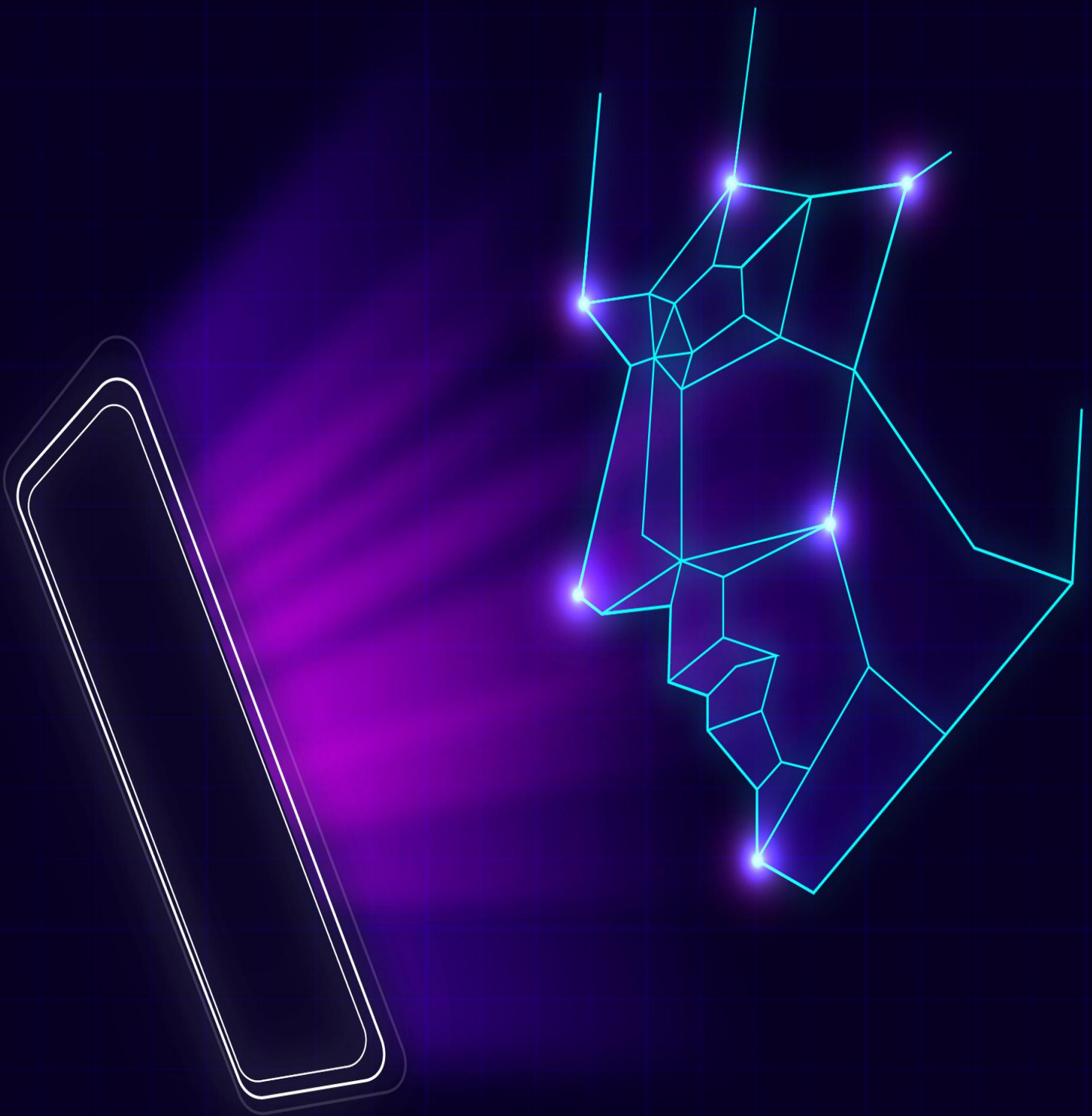


Deepfake Image detection using Anchored Pairwise Learning Approach

Srinath Sudharsan



Deepfake Image detection using Anchored Pairwise Learning Approach

by

Srinath Sudharsan

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

Track: Data Science & Technology
(Pattern Recognition & Bioinformatics)

at the Technische Universiteit Delft,
Faculty of Electrical Engineering, Mathematics and Computer Science,
to be defended publicly on August 24, 2020

Student number:	4770196	
Project duration:	January 02, 2020 - August 24, 2020	
Thesis committee:	Dr.ir. D M J Tax,	TU Delft, Supervisor
	Dr. H Hung,	TU Delft, Committee Member
	Dr.ir. U Gadiraju,	TU Delft, Committee Member

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Deep learning has enabled technologies that have been perceived as complex or impossible a few years ago. Deep learning models can be used to solve several complex problem statements thereby making it a prominent field of research ¹. With the advancements of Deep learning models, their application in domains have diversified. One prominent use-case is the generation of synthetic images or videos targeting unethical media manipulation [1, 2]. This raises concerns on implications caused in the society where there is loss of trust in digital media. The synthetic media generated using Deep learning models are called Deepfakes (a portmanteau of "Deep learning" and "fake"). Deepfake generation leverages powerful techniques using Deep learning and Artificial Intelligence (AI) to manipulate or generate visual content with a high potential prospect of deception to human perception. The application of deepfakes is primarily in face images or videos.

The use-case of synthetically generated media is a powerful tool to communicate information with false motives to public. However, synthetic media generation of faces is not new with Deep learning. Prior to deepfakes, synthetic face images and videos were generated using photo-editing softwares, primarily in the computer graphics domain [3]. The inclusion of Deep learning methods has only facilitated higher realism of synthetic media with lesser time computation required.

With synthetic media being made easier to generate using Deep learning models, appropriate detection methods should also be deployed. This thesis focuses on detecting deepfakes using Face Recognition Systems. Deepfake detection methods have been proposed and implemented since 2018. However, the fundamental challenge in deepfake detection is the capability of the model to learn discriminatory features between real and deepfake media (owing to high similarity and realism in synthetic media). Furthermore, the availability of real images or videos of target individuals is lesser in comparison with the number of deepfakes available thereby posing a challenge for the detection models to further learn the real vs. deepfake features.

In this thesis, a modification to traditional Siamese Network using Pairwise Learning approach is made called Anchored Pairwise Learning, to learn the similarity between a pair of input faces to address the challenges in deepfake detection with regard to availability of real faces per individuals class. Anchored Pairwise Learning combines two proposed methods called Anchor Siamese Network (ASN) and Anchor Face Detector Network (AFDN) to perform a binary classification of images belonging to real or deepfake class by anchoring one of the test input pairs. The reported best AUC performance of the method on a test set of real and deepfake images is 91.9% and 81% on an independent hold-out set. Additionally, a comparative study between the proposed method and a baseline traditional learning face recognition method is made and their results based on model performances are discussed. Furthermore, the proposed method is evaluated with other state-of-the-art deepfake detection methods from a generalization performance on the reserved independent hold-out set. The results conclude that the Pairwise learning techniques outperform traditional learning techniques in scenarios where the number of real images are lesser compared to the deepfake images.

¹<https://www.kdnuggets.com/2017/12/using-deep-learning-solve-real-world-problems.html>

Acknowledgements

It is with gratitude that I take this opportunity to thank those individuals who have been of immense help in the completion of my thesis work.

First and foremost, the entire thesis work is a result of the guidance of Dr. DMJ Tax, who is not only my daily supervisor, but a mentor as well. He was friendly and approachable, yet critical when it mattered most to bring out the best of the results. Needless to say that he was instrumental in furthering my research thinking. He has shown great compassion, during the thesis period and I appreciate the freedom that he gave me, to work my way through the thesis completion. Amongst the many leanings that I have had, one thing that strikes me most is his analytical and sequential approach to reduce any problem statement to simpler components. I truly enjoyed every moment of his inclusive and collaborative approach that aided me to complete the thesis.

Furthermore, I would like to extend my gratitude to Mr. Avinash Pancham, Cyber Forensics Evangelist who introduced me to the risks of Deepfake through a collaborative project on Deepfake creation. This acted as a fulcrum to the progress of my research into Deepfake detection. My thanks to Mr. Shreyas Srivatchan, my childhood friend who was instrumental in many brainstorming sessions in the conceptualization of the deepfake detection model used in this work.

My special thanks to my family whose moral support encouraged me to go through the thesis during difficult times. Certainly their constant motivation has been of immense help to focus on the assignment with a clear mindset. Also, I acknowledge with gratitude the support provided by my friends during this journey!

The cover page of the thesis report was designed using: <http://www.freepik.com>

Srinath Sudharsan
August 2020

Contents

Acknowledgements	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Challenges and Proposed Solution	3
1.2 Research Question	3
1.3 Research Scope	3
1.4 Contributions.	4
1.5 Report Outline	4
2 Background & Literature Survey	5
2.1 History of Deepfakes	5
2.2 Face Recognition System	5
2.3 Stages in Face Recognition systems	6
2.3.1 Face detection	6
2.3.2 Face alignment	7
2.3.3 Feature extraction	9
2.3.4 Feature matching.	10
2.4 Deepfake detection	13
2.4.1 Detection using holistic methods	13
2.4.2 Detection using feature-based methods	13
2.4.3 Detection using deep learning-based methods	16
2.5 Other Forgery Detection Methods	18
2.6 Summary of Detection Methods for Deepfakes	19
2.6.1 Intuition on real and deepfake data	19
2.6.2 Modification to Traditional Siamese Networks	19
3 Selection of Dataset	21
3.1 Publicly available datasets.	21
3.2 Comparison between the datasets	22
3.2.1 Based on number of real vs. deepfake videos.	22
3.2.1.1 Realism of deepfakes	22
3.2.1.2 Deepfake quality of bad samples	23
3.3 Celeb-DF v2 Data Generation	24
4 Experimental methodology	25
4.1 Dataset Split	25
4.2 Preprocessing.	26
4.2.1 Face detection	26
4.2.2 Face Alignment	27

4.3	Baseline Method	27
4.3.1	Additional Pre-processing	28
4.3.2	Feature Extraction of Eigenfaces.	29
4.3.3	Training	29
4.4	Anchor Siamese Network	30
4.4.1	Input-pair selection	31
4.4.2	Network Selection	32
4.4.3	Network Loss	33
4.4.4	Training	33
4.4.5	Anchor Face Detector Network	34
5	Results	36
5.1	Test Results of Baseline Method	36
5.2	Small experiment on ASN Method	38
5.3	Test Results using ASN method	39
5.4	Comparison of ASN method vs. Baseline Method	41
5.5	Generalization Performance.	42
5.5.1	Comparison with other deep learning models	43
6	Conclusion and Recommendations	46
6.1	Remarks.	46
6.2	Extended Applications.	47
A	Deepfake Creation	48
B	Data Generation	50
	Bibliography	53

List of Figures

1.1	Class of Face manipulation methods [4]	1
2.1	Stages in Face recognition systems [5]	6
2.2	MTCNN - stage based iteration approach for face detection [5]	7
2.3	68-point landmark descriptors (red dots) and five-point landmark descriptors (green dots)	8
2.4	Face alignment algorithm	8
2.5	DeepFace pipeline [6]	10
2.6	Generic representation of a Siamese Network	11
2.7	Triplet loss architecture representation on a Deep Network for a sample set of patches	12
2.8	Deepfake detection using BoW-based hand-crafted feature based approach [7]	14
2.9	Detecting tampered regions using Face Classification and Patch Triplet streams [8]	15
2.10	Proposed DeepFD pipeline using pairwise learning approach in [9]	17
2.11	Pairwise learning using Siamese Network on Contrastive Loss	17
2.12	Proposed flowchart using CFFN with triplet loss from [10]	18
2.13	SigNet architecture with layer information [11]	18
3.1	Visual low-quality of bad samples from UADF, DF-TIMIT, FaceForensics++ (FF-DF), DFD and DFDC datasets [12]	23
3.2	Sample frames from Celeb-DF v2 dataset [12]. First image denotes the real frame (source) and subsequent frames denote the deepfakes generated	24
4.1	Sample false positives detected by Viola-Jones detector by random inspection	27
4.2	Face alignment for sample face from the dataset	27
4.3	Baseline Method workflow	28
4.4	Sample face image after additional pre-processing operations	28
4.5	Obtained mean faces from rPCA	29
4.6	Cumulative variance vs rPCA components	30
4.7	Sample top eigenfaces	30
4.8	Training Stage of the proposed ASN using ResNet-50 architecture	31
4.9	Testing Stage of the proposed ASN using ResNet-50 architecture	31
4.10	Sample Input pair ordering	32
4.11	ResNet skip connection concept [13]	32
4.12	ResNet-50 architecture	33
4.13	maximum confidence score reported per anchor real class	34
4.14	Detected sample faces with bounding box by HoG detector	35
5.1	ROC on test set for Baseline method	36
5.2	Normalized Confusion Matrix of the baseline method	37
5.3	PR Curve for baseline method	37
5.4	Sample similarity scores on a small experiment set	38
5.5	ROC for varying anchor faces	39
5.6	Varying AUC vs. Anchor faces - trend	39

5.7	PR curve for 30 anchor faces	40
5.8	ROC for ASN method vs. Baseline method	41
5.9	ROC curve for 30 and 40 anchor faces on independent hold-out set	42
5.10	ROC performance for baseline and ASN method on independent hold-out set	42
5.11	ROC performance of SigNet with varying anchor faces on independent hold-out set	43
5.12	ROC performance of XceptionNet on independent hold-out set	44
5.13	ROC performance of ASN method with other methods on independent hold-out set	44
A.1	deep learning network models used for deepfake generation [14]	48
A.2	deepfake creation model using two encoder-decoder pairs [15]	49

List of Tables

2.1	Different embedding dimension reported in [16]	10
3.1	Details of datasets discussed in Chapter 3.1	22
3.2	Average Mask-SSIM score comparison for different datasets	23
4.1	Celeb-DF v2 dataset split for experiments (video reference)	25
4.2	Celeb-DF v2 dataset split for experiments (frame reference)	26
4.3	Viola-Jones feature detector vs. MTCNN	26
4.4	Summary of scores and confidence measures for sample faces from Fig 4.14	35
5.1	Summary of test results on the small experiment set	39
5.2	AUC score of random sampling vs. fixed anchor faces for 30 celebrities class	40
B.1	Deepfake generation table 1	51
B.2	Deepfake generation table 2	52

Introduction

In Digital Forensics, manipulation of synthetic media content has become ubiquitous and a critical problem to handle in the digital society. The images or videos that are seen on the internet or digital platforms have become questionable owing to its false content [17]. The generation of synthetic media is not a new problem. Prior to the introduction of deep learning and computer vision methods, synthetic media was generated using computer graphics and visualization techniques [18]. These techniques were primarily used to replace faces of existing persons with a likely similar face of another. The underlying motive maybe plenty such as defaming a person's public reputation, spreading false information, hoax content creation etc. The reason for using faces as a manipulation technique is that, faces play a focal role in the communication of information on digital platforms as the face of a celebrity or person can convey important information in its own right.

With the advent of computer vision aided deep learning methods, face tracking and synthetic media generation is made even easier [19]. The class of methods that focus on generating fake facial media is called Facial Manipulation. Rossler et al. [4] define facial manipulation methods into two categories - Facial expression manipulation and Facial identity manipulation (Refer Fig 1.1). Facial expression manipulation techniques are focused on transfer of facial expressions of source face to a destination face in real time with the use of commodity hardware. Facial identity manipulation are those techniques that replace faces of source person to the destination person's image or video [4]. This technique of synthetic media generation using deep learning methods is called Deepfake. Since the media form of deepfakes are images or videos, image classification techniques are widely used for detection (or classification). With the development of face recognition techniques, the deepfake detection tasks focus on inconsistencies noted in facial feature such as tampered regions of eyes, lip edges etc. [20]. The detection task is viewed in these cases as a binary image classi-

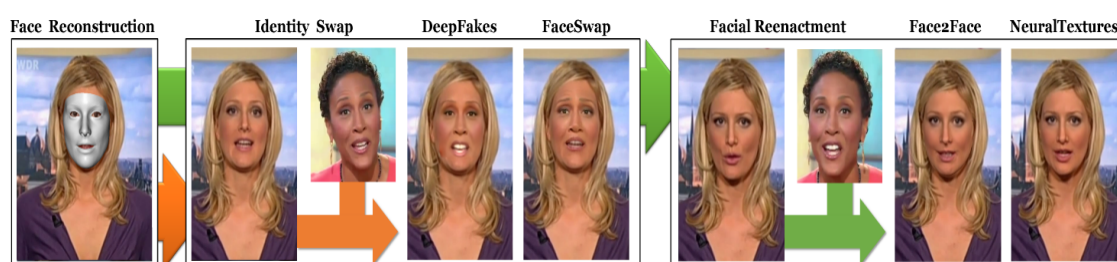


Figure 1.1: Class of Face manipulation methods [4]

fication approach in a supervised way (i.e., real vs. deepfake) [21, 22]. These class of techniques can also be called as *Traditional Learning Approaches* that use standard decomposition methods to learn representations of the data to classify into a binary label. To develop a good deepfake detection model using traditional learning approach, it would be crucial to train many real and deepfake images or videos of individuals with the help of GAN or Auto-encoder(AE) to achieve a good detection performance. However, this largely time-consuming and computationally intensive to perform. Additionally, in real-world setting, with the limited availability of real images or videos of individuals and many deepfakes as compared to the real, it becomes challenging for the trained model to learn discriminative features of the real class thereby leading to poor performance. As a result, the generalization¹ performance of the model may not be very promising.

To meet the challenges of generalization and poor availability of real class of images or videos, pairwise learning approaches are used to detect deepfakes. They are methods which learn the relationship between an input pair and evaluate based on similarity scores between the pairs of input [23]. Unlike traditional learning, the inputs to pairwise learning are pairs or tuples. They extract all possible pairwise preferences from the input classes and train the model to distinguish or find similarities. A metric loss (distance-based metric) computes the spatial distance between the input pairs in an embedding space and outputs a similarity or dissimilarity score which can be inferred by a classifier to detect real vs. deepfake. The most common applications of pairwise learning are in document-classification, search engine optimization tasks. The advantage of this approach is the use of less training data [24]. However, it is very crucial to ensure that all possible combination of pairs are learnt by the model. Hence, pair-selection and ordering from training data are important preprocessing steps in pairwise learning. From a generalization perspective, the features representation from pairwise learning are more discriminative and hence perform well on unseen data [25]. In the context of deepfakes, pairwise learning can be useful to learn similarity (or dissimilarity) between real-real, real-deepfake and deepfake-deepfake pairs, thereby enabling the model to be able to learn more representations [26].

With regard to the computational load of pairwise learning tasks, (especially if the training data is large) the load is high as it increases quadratically as opposed to traditional learning approaches owing to the input pairs at the training stage [27]. Hence, pairwise learning methods dealing with large datasets are often be used by limiting the number of training input pair combinations (as a hyper-parameter choice) to have better computational efficiency.

Since deepfake media are involved with forgery of faces, classical face detection and recognition techniques (called Face Recognition Systems) may be used to distinguish between real or deepfake faces. However, face recognition systems have some challenges with regard to being robust to intrusive factors such as pose, lighting conditions, illumination, facial expression, occlusions noted in the face images. Additionally, another challenge noted is being able to generate accurate feature representations learnt by the training model of choice. Lastly, the generalization of face recognition on unseen representation encoding (generated from test data) is a challenge. There are four classes of face recognition systems based on the feature extraction (representation) perspective [28]. They are:

- Holistic Methods - Use of global information extracted from the faces to perform recognition
- Feature-based Methods - Use of specific or similar local features extracted from the faces to perform recognition

¹<https://developers.google.com/machine-learning/crash-course/generalization/>

- Hybrid Methods - Combines holistic and feature-based methods in stages to perform recognition
- Deep learning-based Methods - Use deep learning models to extract highly discriminative features from a face and perform recognition

(A more detailed review on the stages of face recognition and the class of recognition systems can be found in Chapter 2.2 of this report)

1.1. Challenges and Proposed Solution

This thesis aims at addressing two important problems present in the detection of deepfakes. Firstly, the less availability of real images of target faces as compared to deepfake faces in a real world setting. To address this challenge, an Anchor Siamese Network (ASN) with an application of Anchor Face Detector Network (AFDN) is proposed in the testing stage of the model as one of the fixed input pairs, that leverages the limited availability of real faces. Together, the domain of learning approach is called Anchored Pairwise Learning. The primary purpose of the ASN with AFDN is to find the optimized number of real images per target face, required by the trained model to be able to detect a test face as real or deepfake using a pairwise learning approach. Therefore, the less availability of real faces will no longer be a hindrance to the performance of the model to detect deepfakes. Furthermore, with less real faces, the generalization capability of the model to detect unseen faces (real or deepfake) poses a challenge. Hence, the Anchored Pairwise Learning approach is used with the help of deep learning methods to address the generalization performance. Further details on the design choices and experimentation details of this approach are provided in Chapter 4.

With the focus of this thesis limited to challenges in deepfake detection, the generation of deepfakes is covered with the use of a benchmarked dataset containing real and deepfake media. There are several deepfake media datasets (video and image datasets) that are publicly available for research purposes. The dataset used for detecting deepfakes in this thesis is Celeb-DF v2 which was published in November 2019 [12]. Celeb-DF v2 dataset contains 590 real videos collected from YouTube with subjects ranging from different ages, ethnic backgrounds and genders and 5639 corresponding deepfake generated videos. The YouTube videos are of celebrities or people of popular acclaim in the society. A detailed overview of the dataset and choice of dataset is explained in Chapter 3.

1.2. Research Question

How is the proposed deep learning model capable of detecting deepfakes with lesser samples of training images (real images of target faces per class)? Furthermore, how does the performance of the model improve with varying the number of anchor images from the anchor face detector network? Lastly, how does the model perform on unseen faces (generalization performance being evaluated)?

1.3. Research Scope

The scope of the thesis is limited to deepfake media for detection. The reason for the choice of deepfakes is due to the realism of synthesized output media as compared to other facial identity manipulation techniques discussed above. Additionally, the scope of deepfake media is restricted to images. The deepfake images are not generated using creation methods, rather the images of real and deepfakes are obtained from Celeb-DF v2 dataset. However, an overview of the deepfake creation methods are discussed briefly in Chapter 2 to instantiate the implications of synthetic media generation and the realism of the output.

1.4. Contributions

With regard to deepfake detection methods, this thesis offers the following contributions:

1. A deep learning based face recognition technique using a modified Siamese network with ResNet-50 [13] architecture called ASN coupled with an application of AFDN to output a real face from the training set based on face detection methods with the purpose of anchoring one of the testing pairs during the testing phase of the model.
2. Demonstrate the performance of the proposed model on a hold-out test set to assess the generalization ability of the model (to evaluate the performance on unseen faces).
3. Highlight the significance of Anchored Pairwise Learning approach (a minor extension to Pairwise learning methods) over traditional learning through qualitative and quantitative result analysis.

1.5. Report Outline

Background knowledge and definitions pertaining to deepfake creation methods, an overview of face recognition systems and discussions on pairwise learning vs. traditional learning tasks that are used in this thesis have been explained in Chapter 2. It is combined with description of existing work in the deepfake detection field of media forensics. Chapter 3 contains the details related to the dataset used in this thesis with the choice considerations and other relevant deepfake dataset that are publicly available for deepfake detection. It is followed by description of methodology used to build deepfake detection models from dataset in Chapter 4. The performance of the models proposed with their qualitative and quantitative results analysis and methodology used to evaluate the performance are discussed in Chapter 5. Lastly, Chapter 6 concludes the thesis work with limitations and scope of further improvements.

2

Background & Literature Survey

This chapter includes background information required for a better understanding of the concepts used in this thesis. The thesis focuses on deepfake detection using face recognition methods. An overview of face recognition methods are discussed below.

2.1. History of Deepfakes

The initial implementation of facial identity manipulation is Face Swapping. Face swapping is a naive real-time face superimposition technique introduced as an entertainment tool on the popular social media platform Snapchat. With the advancement of deep learning models, face swapping was made more realistic with the help of deep learning thereby making it challenging for forensic models to detect fakes. The reason being, the deep learning models can preserve pose, facial expressions or features on the generated synthetic media [29].

Deepfake (portmanteau of "deep learning" and "fake"), is an artificial intelligence technique which uses superimposition of existing people's faces on a source face to create an existing or fictitious person's video sequence or image. The realism of deepfake media is very high owing to the use of deep learning models like auto-encoders and generative adversarial networks (GAN) that are applied widely in the computer vision domain to solve various face recognition or media-tracking problems [30–33]. The first instance of deepfake emerged in 2017, wherein synthetic media content of celebrities was generated using face swapping with actors from the adult entertainment industry¹. With this starting point having severe consequences, eventually, political campaigns or speeches were falsified using deepfakes [32, 34]. Deepfakes have also been used in rightful application such as voice-over generations in movies or re-shooting a movie scene without actors². However, the negative implications outweigh the positive use-cases of deepfakes. Over the years, many deep learning models have been used to create deepfakes [35–38].

2.2. Face Recognition System

Face recognition systems are computer vision techniques used to identify and verify faces of people from an image or video source. Face recognition is a specific case of object detection and verification method in computer vision. Hence, the traditional machine learning and deep learning techniques can be deployed to build a robust pipeline for identifying the face in an image or a video. With the use of these models, the face images are encoded with a feature representation (feature vector). The

¹<https://www.bbc.com/news/technology-42912529>

²<https://www.forbes.com/sites/bernardmarr/2019/07/22/the-best-and-scariest-examples-of-ai-enabled-deepfakes/>

learning face recognition algorithm iterates through this data to identify concurrent or similar patterns for classification. Face recognition systems have some challenges with regard to being robust to intrusive factors such as pose, lighting conditions, illumination, facial expression, occlusions. Lastly, the generalization capability of face recognition systems on unseen representation encoding (generated from test data) is a challenge. All face recognition systems follow a stage-based approach composed of five building blocks (Refer Fig 2.1). The first building block comprises of the face detection wherein the input image is detected for the facial regions. This is performed using pre-trained face detection methods that identify facial features in an image using computer vision techniques. In the second building block, face alignment is performed to orient the face image to a common orientation point (angle). The reference of orientation is fixated with one of the facial feature detected such as eyes, nose, ears, etc. The choice of face alignment is to align all face images on a common reference since face images may be rotated in natural settings based on different poses. Furthermore, in the feature extraction stage, the identified features that are representative to perform classification are used. The classification tasks are performed in the feature matching stage of the face recognition system.

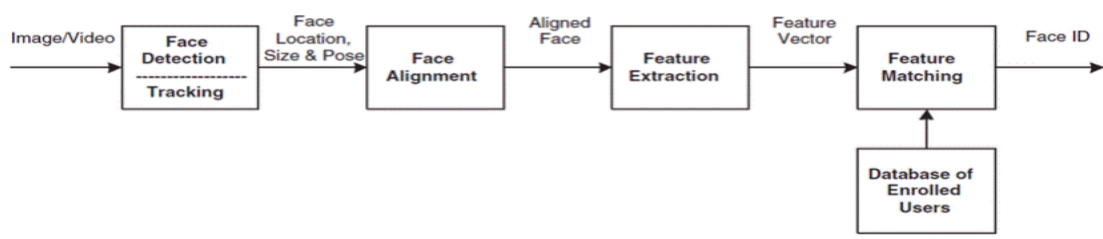


Figure 2.1: Stages in Face recognition systems [5]

2.3. Stages in Face Recognition systems

2.3.1. Face detection

Face detection is an initial step in facial recognition. For any face recognition method, identifying the face using detection methods aids the model to focus on the specific region of interest (here faces). The input to the face detection stage is the image or video containing the face. A face detector is used to find the position of the faces in a given image and the output is a set of coordinates locating the shape of the face and/or the facial landmark points (such as eyes, nose and other facial features). A very popular method for face detection was proposed called Viola-Jones algorithm in [39]. In this approach, the images are converted to grayscale. An outline box searches for Haar-like features within the image in a cascaded fashion. Haar-like features are a combinatory composition of edge features, line features and four-sided square features.

These features facilitate the detection algorithm to understand the nature of the image based on horizontal and vertical feature mapping. The Viola-Jones algorithm detects these Haar-features and maps them to the facial attributes (such as eye-brows, nose, mouth etc.). The best features are selected using Adaboost by sampling the face components from rest of the image using weak classifiers and cascade. The primary idea of this algorithm is the use of integral images for quick and real-time computation of Haar-like features, with the use of boosting to select the most relevant features (this process is called robust feature selection), and discarding of non-facial regions using cascading.

Another suggested approach for face detection is using Histogram of Oriented Gradients (HoG)

method. Here, a vectorized image patch consisting of a concatenation of HoGs from various sub-regions is fed to a linear Support Vector Machine (SVM) classifier to perform the classification task for detection. [40]. Face detection task can also be performed using deep learning. Multi-task Cascaded Convolutional Networks (MTCNN) [41] is a deep learning based face detection technique that leverages a cascaded architecture with three stages of CNNs to predict face and landmark location in a coarse-to-fine manner. MTCNN uses three networks namely, P-Net, R-Net and O-Net (Refer Fig 2.2). In the P-Net stage, the algorithm produces a candidate window sliding quickly through a shallow network of CNNs. The stage 2 containing the R-Net, refines the windows with the purpose of rejecting large number of non-facial windows present in the reference image with additional CNNs. Finally in the O-Net stage, deeper CNN layers are used to refine the result of the detected facial features in the image and further locate the five-point facial landmarks.

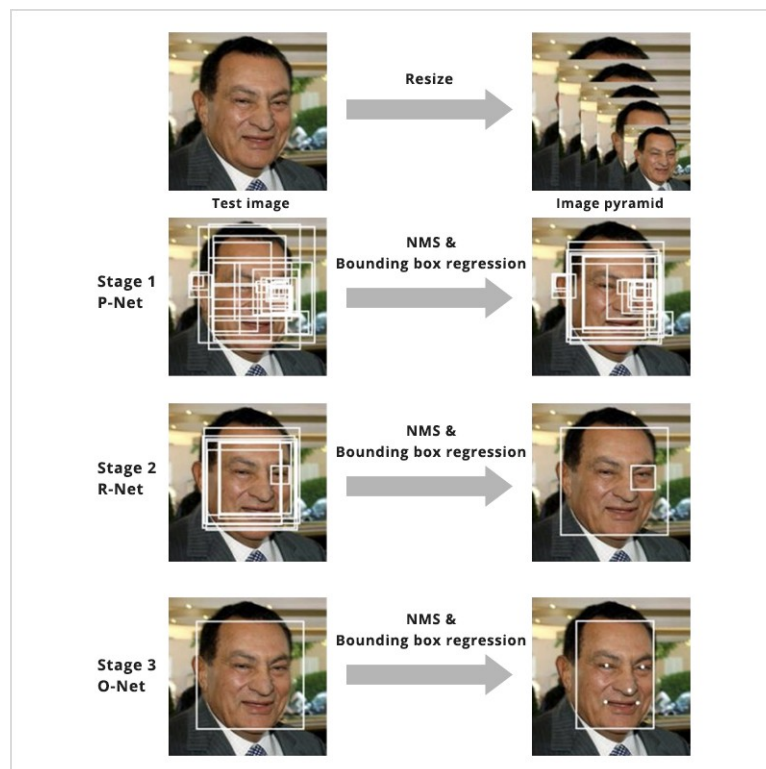


Figure 2.2: MTCNN - stage based iteration approach for face detection [5]

The output of the face detection methods yields facial regions and landmark point descriptors that are referenced facial region mappings trained over several faces. They are functionally known to generalize well on new face images. There are two popular facial landmark points descriptors that are widely used for face recognition tasks. They are five point facial landmark descriptors and 68 point facial landmark descriptors. The number of landmark points correspond to individual or a collective facial coordinate regions. Fig 2.3 illustrates the 68- and five point-facial landmark descriptors³.

2.3.2. Face alignment

Many face recognition algorithms depend on careful positioning of the face into a canonical pose, so the position of features relative to a fixed coordinate system can be examined. The facial landmark descriptors such as eyes and the lip region points can be centered and roughly oriented along

³Fig 2.3 Source: <https://multiforesee.com/wp-content/uploads/2018/03/WG2-Report-2017-1.pdf>

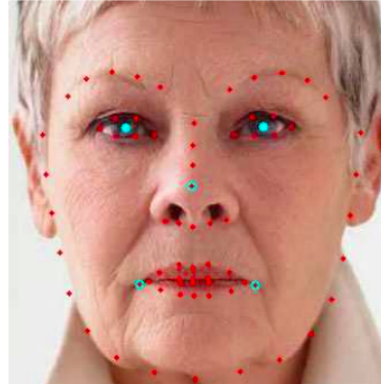


Figure 2.3: 68-point landmark descriptors (red dots) and five-point landmark descriptors (green dots)

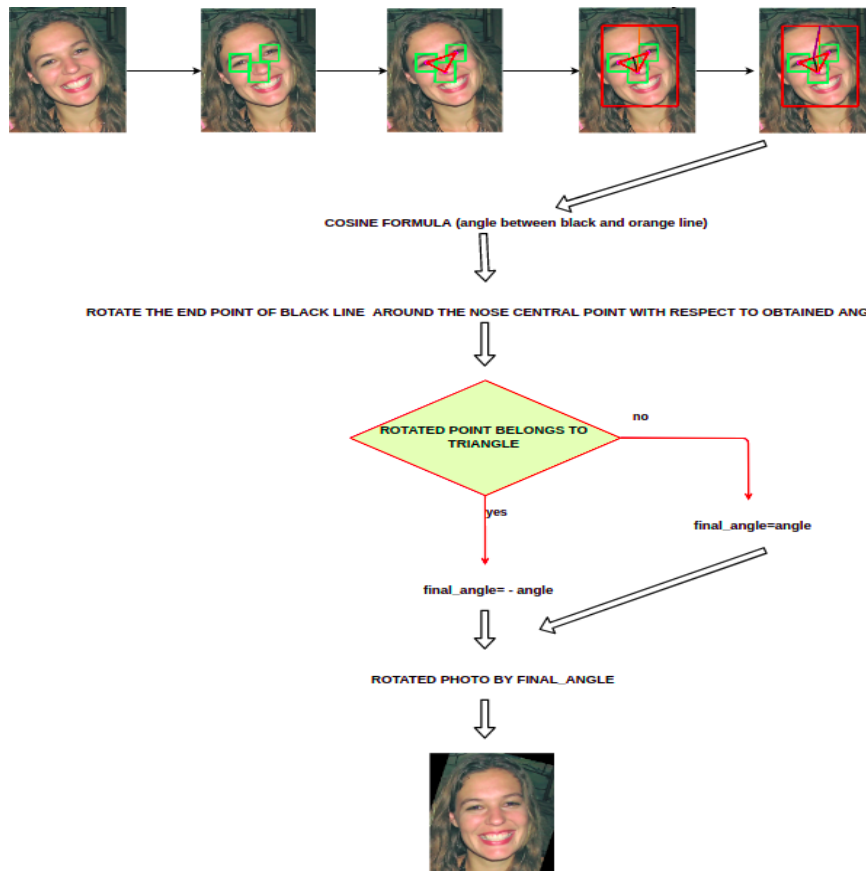


Figure 2.4: Face alignment algorithm

the same position of the image. The use of face alignment is to obtain a normalized representation of the face. The face alignment algorithm adopted in this thesis ⁴ uses Precise alignment algorithm (Fig 2.4). The facial landmark descriptors are obtained from face detection techniques. The choice of the number of descriptors is dependent on the use-case. The five point facial landmark points is obtained in this scenario. The central coordinates such as the eyes, nose are mapped and a triangle is made using the nose and eye coordinates to find the median of the triangle (with nose as the central point of reference). Further, the coordinates of the centre of a top side of a face rectangle (center of a top side) is obtained and a second triangle using the coordinates of the end of the median, nose

⁴Adaptation Source:<https://github.com/HikkaV/Precise-face-alignment.git>

coordinates and center of top side coordinates is made. The angle between the median and the side of the second triangle (between the black and orange lines in Fig 2.4) is found. With the obtained angle, the endpoint of the median around the nose coordinates with respect to the angle is rotated. If rotated point lies in the space of the second triangle (belongs to it), then the final angle (the angle we will use to rotate our image) equals to negative angle. Otherwise, final angle equals to the origin angle. Lastly, the whole image is rotated along the final angle.

2.3.3. Feature extraction

Feature Extraction in an integral part of the face recognition pipeline. There are four methods of feature extraction [28]. The features extracted from the face depends on the type of method that is used for extraction. In all the methods, a feature representation (or embedding) is generated as the output of feature extraction.

Holistic Methods

Holistic approach examines the face image input as a complete entity and extracts statistical information from the input data. These methods were a very favourable option in the field of face recognition in the 1990s. A few examples under this class of algorithms include Fisherfaces, Eigenfaces, Kernel methods and Independent Component Analysis(ICA). While Eigenfaces is an unsupervised approach aimed at dimensionality reduction, increasing interpretability, preserving maximum variability and decorrelating the features, Fisher Faces is a supervised dimensionality reduction technique that identifies the projection that best distinguishes the class by optimising the Fisher's Discriminant ratio. In [42] ICA is implemented for dimensionality reduction which aims to identify the projection that produces features not only uncorrelated but independent as well. In [43] a kernel trick is implemented on the LDA and PCA transforms to identify a better non-linear transform that yields Kernel LDA and Kernel PCA methods for good feature extraction. The disadvantages with holistic methods are its poor capability to learn subtle differences in features between two input faces.

Feature-based Methods

Feature-Based approaches are used to identify and extract facial landmarks (descriptors) such as the nose, mouth and the eyes to compute the distance between them and identify their geometrical characteristics. These approaches when subject to changes in pose and illumination, have been shown to remain unaffected. Despite this advantage, the identification of the landmarks has shown to be unreliable, thereby reducing the performance and efficiency of these approaches. In addition to this, feature-based approaches tend to disregard the texture of the face which could also be used to differentiate faces. A classical feature-based approach is Scale Invariant Feature Transform (SIFT) [44] which extracts distinctive invariant features from face images to perform reliable face matching, head pose matching with profile image feature technique [45]. Additionally, other methods include Speeded Up Robust Features (SURF) [46], Oriented FAST and Rotated BRIEF (ORB) feature detection method [47] which can also be used for face detection applications.

Hybrid Methods

Hybrid methods combine holistic and feature-based methods in a serial or parallel way to overcome the disadvantages of the individual methods. The computational costs incurred by using these methods are very high and hence, they are used only in specific use-cases as compared to holistic or feature-extraction methods with common use in 3D images.

Deep learning based Methods

The advancement in deep convolutional neural networks, has created many application in face

recognition recently. The primary reason for using deep learning methods is the discriminative feature learning tasks performed by Deep neural networks (DNN). The first of the face recognition algorithms, was proposed in 2014 by Facebook's research group [6] called DeepFace. In the work, the authors propose a four stage pipeline containing face detection, 3D alignment (for faces), generating representations and classification task. In Fig 2.5, the detection phase identifies facial points using landmark point detection method. Localization and Frontalization is performed for the faces using 3D alignment. The aligned faces are passed to a convolutional layer containing 32 filters (C1), a max pooling layer (M2) and sequential convolutional layers with 16 filters (C3) with three local connection layers (L4, L5 and L6). The functional use of L4, L5 and L6 are similar to the convolutional layers. However, weight sharing is not used. The reason being, in the aligned face specific areas contain different properties and hence possess more discriminative abilities in comparison to others. Lastly, two FCNN layers (F7 and F8) are used to generate a dense 4096-d face representation for each face input. The classification is performed from the generated face embeddings.

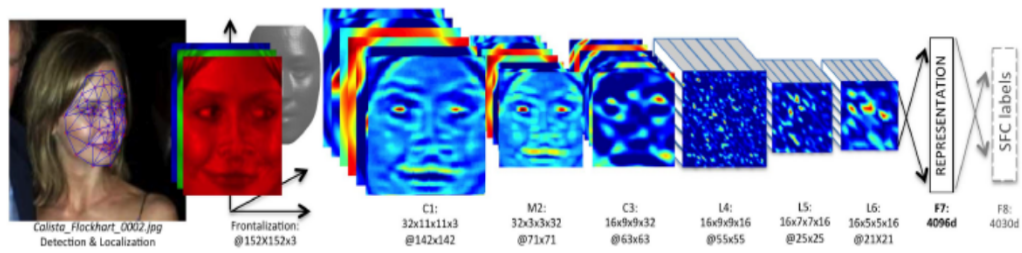


Figure 2.5: DeepFace pipeline [6]

In 2015, Google proposed FaceNet [16] that is a deep CNN model trained on pairwise loss called triplet loss (discussed in Section 2.3.4). In this research different feature representation dimensions were generated and the best representation was selected based on the classification accuracy. Table 2.1 summarizes the different representation dimensions tested with the corresponding accuracy scores for face classification task. It was concluded that 128-d representation provides better performance as compared to higher dimensional embeddings.

Table 2.1: Different embedding dimension reported in [16]

# Dimensions	Performance (Accuracy)
64	86.6% $\pm$ 1.7
128	87.9% $\pm$ 1.9
256	87.7% $\pm$ 1.9
512	85.6% $\pm$ 2.0

2.3.4. Feature matching

In feature matching, the output feature representations (embeddings) are tested on a validation or test set of face images. The approaches that can be used in feature matching are Traditional learning and Pairwise learning methods.

Traditional learning approach learns from different predetermined classes of data. A probabilistic or statistical classifier then predicts the class labels of the extracted feature vector. In this classification approach, single test data (faces) are evaluated by a model that is trained with single input data. The predictions made by the model can be used to perform a binary or a multi-class problem.

On the contrary, in pairwise learning, the model learns from similarities between data (faces) and computes a score based on a distance metric. The test image belongs to the class with the smallest distance. The loss function in pairwise learning are distance-based pair losses that are used to minimize or maximize the objective function (based on the use-case). The objective is to learn the extracted feature representations with greater distances to be different faces (representations). The classification tasks is primarily based on similarity score obtained from test pairs.

In deep learning, Siamese networks are used for pairwise learning methods. A Siamese network is a twin network that uses shared weights in the training phase on a pair of distinct inputs and similar inputs to generate two high level embeddings. The shared weights between two networks guarantee that a similar input pair cannot be mapped by their respective networks to very different locations in a common feature space as both the networks are trained on the same loss function. It is important to note that the Siamese Network does not learn to make decisions on classification a face directly to the output of classes. But rather, the network learns a similarity function, learnt from two faces as input and expresses the similarity between the pairs. This way, the learnt representations (feature vectors) show higher discriminatory characteristics enabling the model to learn well. The output of Siamese Network is also a pair of output vector pairs called face embeddings. To compute the similarity the Euclidean distance of the two face embedding are computed and the classification decision is based on the threshold on the similarity scores (Fig 2.6 depicts a generic representation of a Siamese Network). In the training stage, the Siamese Network learns the discriminative features between two inputs (pairs) using the shared weight parameters. Additionally,

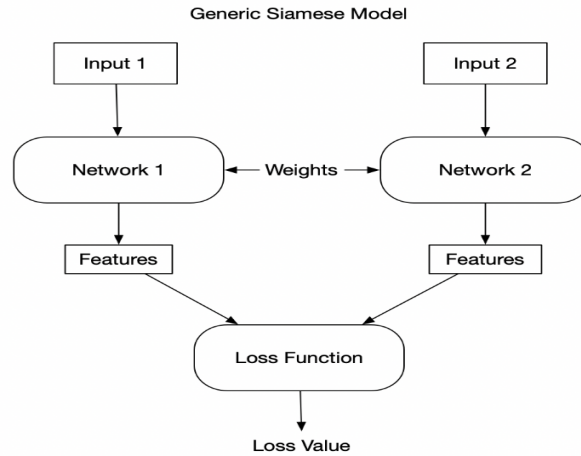


Figure 2.6: Generic representation of a Siamese Network

since the network is symmetric, when two distinct input pairs are trained, the top conjoning layer computes the same metric thereby unifying an embedding space [48]. The loss function used for Siamese networks is a distance-based deep metric loss. The first deep metric loss was proposed in [49] called Contrastive loss.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_i\}$ denote a training set. where x_i represents the data at the i^{th} index. Let $\mathbf{Y}$ represent the corresponding labels to train $\mathbf{X}$. $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_i\}$ where $\mathbf{y}_i$ represents the label for $\mathbf{x}_i$. The contrastive loss takes output feature representations from distinct and similar inputs from the trained model. Let $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$, be the elements of the selected input pair $\{\mathbf{x}_i, \mathbf{x}_j\}$. y_{ij}^{sim} is a binary label indicator based on the similarity measure for the pair $\{\mathbf{x}_i, \mathbf{x}_j\}$. Eq 2.1 defines the contrastive loss

function given by,

$$L_{\text{contrast}} = (1 - y_{ij}^{\text{sim}}) \frac{1}{2} (D_W)^2 + (y_{ij}^{\text{sim}}) \frac{1}{2} \{\max(0, m - D_W)\}^2 \quad (2.1)$$

where, $y_{ij}^{\text{sim}} \in \{0,1\}$. D_W is a distance measure between two transformed samples in a pair given its representations as $D_W = \|f(x_i) - f(x_j)\|$, where $f(\cdot)$ represents an embedding applied to the input image. This represents a learn-able distance function. m is the margin hyperparameter that denotes a hyper-sphere radius containing all similar samples. While training a model using contrastive loss, it is critical to pass all possible combinations of input pairs for better learn-ability.

Additionally, another popular deep metric loss uses a triplet embedding from trained networks is called triplet loss. The triplet loss was used by FaceNet model discussed in Chapter 2.3.3. In the training stage of FaceNet, the input samples are fed to the CNNs trained on a triplet loss. Triplet loss builds three patches $\mathbf{x}_i^a$ (anchor patch), $\mathbf{x}_j^p$ (positive patch where $\mathbf{x}_j$ is an image from the same class as $\mathbf{x}_i$) and $\mathbf{x}_k^n$ (negative patch where $\mathbf{x}_k$ is an image from a different class as $\mathbf{x}_i$). α is the margin hyperparameter between the positive and negative pairs. The loss enforces $\mathbf{x}_i^a$ is closer to all $\mathbf{x}_j^p$ than to $\mathbf{x}_k^n$. Eq. 2.2 is the loss function of triplet loss used to generate face embeddings (the output of the feature extraction process). The process described above is diagrammatically represented in Fig 2.7 for a sample set of anchor, positive and negative patches.

$$L_{\text{triplet}} = \sum_{i=1}^N \left[\left\| f(x_i^a) - f(x_j^p) \right\|^2 - \left\| f(x_i^a) - f(x_k^n) \right\|^2 + \alpha \right]_+ \quad (2.2)$$

where N is the size of triplets from training set. $+$ denotes:

$$[x]_+ = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else} \end{cases}$$

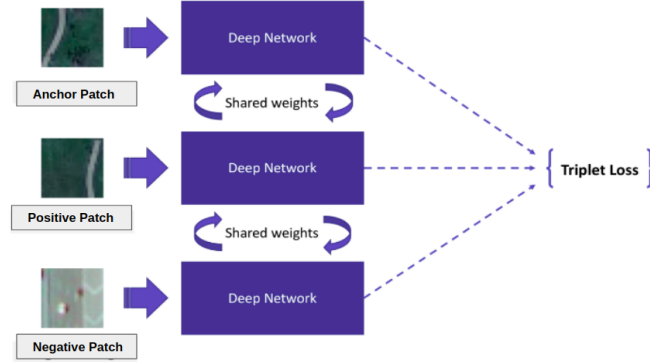


Figure 2.7: Triplet loss architecture representation on a Deep Network for a sample set of patches

The important aspect to consider when using pair-losses is the semantics of the pair combinations. It is critical to ensure that the model receives all possible combinations of input pairs for training. This helps in better feature representations thereby aiding in higher performance on a test set of unseen samples(faces).

With an introduction provided to classes of face recognition systems and the stages involved, the present focus is guided to the literature surrounding deepfake detection using face recognition methods.

2.4. Deepfake detection

In this section, a review of deepfake detection methods are discussed from literature. The aspects focus on the way the features are extracted using different approaches, or the specifications of the model choice used to detect the deepfake images.

2.4.1. Detection using holistic methods

Using pre-processing methods

In 2019, Xuan et al. proposed a novel method highlighting the importance of pre-processing in deepfake images [50]. The primary focus in the paper is to detect GAN generated images by exploiting the low-level features in the images. In the pre-processing stage, Gaussian blur and Gaussian noise are added to the images with the premise of removing low level pixel statistics. After pre-processing the images are trained on a traditional CNN network to classify real and deepfake images. The deepfake images are generated using DCGAN [51], WGAN-GP [52] and PGGAN [35] independently on CelebA-HQ [53] dataset (which was face database of real celebrity images). The demerit of this work is the low quality of deepfake faces generated using the different GANs above. The GAN generated faces are not realistic in comparison to the quality of the real faces that are used. However it is interesting to note that the dataset includes a standard set of real faces and different combinations of deepfake faces generated by each of the GANs. Thereby, this method showcases good performance on unseen faces in the testing stage (assessing the generalization capability of the model).

Using dimensionality reduction

An extension to Computer Vision tasks, researchers also experimented with the visual cues with guided audio artifacts to detect deepfake videos. This is performed using dimensionality reduction methods. An initial detection approach in deepfake detection is focused on detecting inconsistencies in visual-audio artifacts by Korshunov et al. [54]. The proposed work identifies inconsistencies noted in lip movements within real and deepfake videos. That is, a sync between lip movements and output audio speech is detected for variations. The audio features used are Mel Frequency Cepstral Coefficients (MFCCs). With the feature vectors, PCA is applied to reduce the dimensionality of the resultant. The PCA applied feature vectors are fed to a simple RNN with LSTM network (similar to [55]). Additionally, evaluations are performed based on Image Quality Measure (IQM) features. For this approach, PCA extracted features are classified using LDA and SVM (parallelly). The dataset used is DeepfakeTIMIT. The approach with IQM feature and SVM as classifier performed better than other comparative methods with 3.3% and 8.9% Equal Error Rate (EER) for the low-quality and high-quality videos in the dataset respectively.

2.4.2. Detection using feature-based methods

Using hand-crafted features

Hand-crafted features refer to properties derived using feature detection algorithms using the information present in the image. Prior to the introduction of deepfakes, synthetic face detection methods aided as pre-cursor to deepfake detection techniques. A popular technique to detect synthetic faces using hand-crafted features is proposed by Bharathi et al. [56] using Restricted Boltzmann Machine (RBM) to detect *digital retouching* (technique which detects modified artifacts in an image from source). This method uses patches that were selected from face images to learn the discriminative features. These features were used to classify the images as original or retouched (meaning real or synthetic faces). The approach is trained and tested on a privately collected dataset of real and synthetic media. The proposed algorithm achieves 87% classification accuracy to detect synthetic media to real.

With the detrimental impact of deepfakes on society, detection methods also focused on hand-crafted features of the deepfake and attempted to exploit the inconsistencies in the generation using feature-extraction methods. An initial experiment to detect deepfake and/or face swapped images is proposed by Zhang et al. [7] using Bag of Words method (BoW) approach (popularly used in Information Retrieval domain) to extract facial features from images which are learnt by Random Forest (RF), Multi-layer Perceptrons (MLP) and SVM. The architecture of the approach is diagrammatically represented in Fig 2.8. In the pre-processing stage, SURF are used as localized feature detectors (represented in green in the figure). It is designed to address the spatial distribution of gradient information in the neighbourhood of the image (localized setting). The interest points (local descriptors) are extracted as landmarks using a pre-defined grid approach (creating grids in images with rows and column layout) thereby creating a dense description of image contents. The SURF features are grouped using k-means clustering where each cluster center corresponds to a visual word which is plotted onto a histogram. The use of visual word is an encoding of the image features. The real images are extracted from LFW face database (proposed by Huang et al. in [57]) and the fake images are generated using synthetic generation methods. The drawback of this paper is with the realism of synthetic images that were generated.

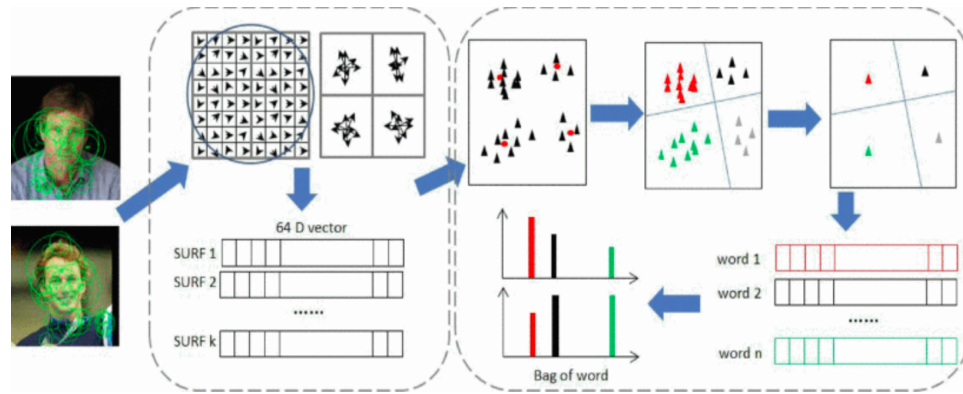


Figure 2.8: Deepfake detection using BoW-based hand-crafted feature based approach [7]

Using facial features

The deepfake detection methods shifted focus to learn facial features. The belief behind this choice is owing to the possibility of discriminatively learning facial features to distinguish between real and deepfakes. In 2019, Matern et al. [58] proposed a detection method using simple visual artifacts like color of eyes, inconsistencies around teeth and lip region are identified using illumination and geometry estimation techniques. The visual artifacts between real and deepfakes are trained and learnt by two classifiers - Logistic Regression and MLP. The dataset of real and deepfake faces are privately collected and the deepfake generation method is not disclosed by the authors. The AUC on the test set of the best performing model is 0.85. However, the demerit of this approach is with the evolution of deepfake generation techniques. The synthesized media has a more realistic quality output thereby overruling much of the inconsistencies that are observed, which is the fundamental hypothesis of this research.

Alternatively, Agarwal et al. [59], proposed a deepfake detection method focusing on specific individuals (political figures). As experimentation, a model to learn specific soft traits present in specific individual characterizations was trained. The hypothesis behind the paper was that "Deepfakes of specific individuals cannot be accurately reproduced by generative models". Facial features and expressions were extracted from real and deepfake videos and classified. The generalization performance was poor as the model failed to perform well on unseen individuals owing to its focus on

specific individual characteristic features.

Detection using steganalysis features

With more exploration into hand-crafted feature learning, detection methods for synthetic image detection are explored using steganalysis. Prior to the introduction of deepfakes, in 2018, Zhou et al. [8] proposed a method to detect tampered regions in synthetic images from real images using a two-stream pipeline using steganalysis⁵. Fig 2.9, showcases the model architecture of the two stream network. In the figure, the first stream (called Face Classification Stream), uses CNN model is trained and fed to a pre-trained InceptionV3 network [60] to classify whether a face was tampered or authentic (real). The CNN learns artifacts created for tampering (the artifacts included strong edges near lips, blurring at forehead of the face). A score is computed from the Face Classification Stream. In the second stream (called Patch Triplet Stream) image information is leveraged from in-camera processing (raw image formats) extracted using steganalysis (adapted from [61]). The information relates specifically to color information from Color Filter Array (CFA) patterns. This information passed to a training network trained on triplet loss and classified using SVM. A scoring function is also computed for the Patch Triplet Stream. The scores obtained from both the streams are aggregated to determine the overall classification performance of the proposed method. The dataset used for experiments is a privately collated dataset of real (authentic) and tampered images where the real images are collected from the internet. The two-stream pipeline method is an instrumental aid to build successive deepfake detection methods using steganalysis features and pairwise learning approaches.

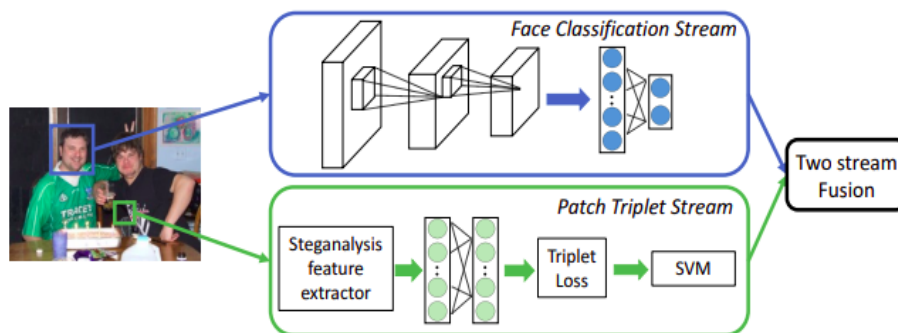


Figure 2.9: Detecting tampered regions using Face Classification and Patch Triplet streams [8]

Alternatively, research conducted by Rossler et al. [4] focused on exploiting hidden or low-level information in images. Two methods are proposed using hand-crafted features for detection. The first method, used a CNN that is trained on handcrafted steganalysis features from [62]. In the second method, a CNN is designed and trained to specifically suppress the high-level content of the image and extract low-level pixel statistical information using references from the works in [63]. Both the methods are evaluated on FaceForensics++ dataset containing real and deepfake videos. The dataset contains a ratio of one real face to one deepfake thereby not depicting the real-world setting of deepfakes being abundant in present state. Qian et al. [64], use a CNN architecture with a constrained first layer to perform High-Pass Filtering (HPF) of facial images with the premise of strengthening weak interceptor signals and reduce the impact of image content present. This therefore, suppresses the scene content and works on residual information. Similarly, Rao et al. in [65], used rich-model filters inspired from [66] and the network is used solely as a feature extractor. For classification, a SVM is used. An extension work is proposed by Bayar et al. [63], by using

⁵Steganalysis is a study of detecting messages hidden in images or signals using cryptography techniques

constrained filter weights that are learnt in the training phase with CNN which also performs the classification task with the complete architecture of Rao et al.'s work retained in this new experimentation. The behavior of rich-model CNN classifier are further replicated to extract steganalysis features with inspirations drawn from [66] by Cozzolino et al. [62]

With all the approaches discussed above, the steganalysis feature extraction method was a very useful approach to identify tampered regions of the image using filters. The primary focus of steganalysis-based methods are to capture low-level features and in some cases, discard global information from facial images. However, the challenge with using steganalysis features is it relies on heavy domain application of digital image processing techniques.

2.4.3. Detection using deep learning-based methods

Using traditional learning methods of CNNs

As part of the DeepFake Detection Challenge (DFDC) challenge for deepfake detection [67], two simple detection systems are proposed by Dolhansky et al. [67]. The first method uses a CNN model with six convolution layers with a fully-connected layer to detect low image manipulations using a frame-wise, face cropped approach called TamperNet. The second method used is an XceptionNet [68] that is trained on full-image frames. XceptionNet takes single input image in the training and testing stages (traditional learning). XceptionNet uses depth-wise separable convolution layers with residual connections. XceptionNet is adapted from InceptionV3 Network (also called InceptionNet). As part of the fine-tuning, the first 36 layers of the network is frozen. These layers corresponds to the first 4 blocks of the network. The last layers of the XceptionNet is replaced by a dense layer with two outputs, initialized randomly and trained for 10 epochs. The resultant network is trained. For optimization, ADAM with a learning rate of 0.001 with a batch-size of 64 is used. As for the input to XceptionNet, one frame is extracted from each real and deepfake video. The XceptionNet model is trained on FaceForensics++ dataset and evaluated on DFDC. The performance of the XceptionNet is with a 79.0% precision and 26.4% recall.

Afchar et al. [69] proposed two improvement methods to CNNs to aid in the detection of deepfake images that focus on mesoscopic properties of images. First, a CNN network comprises of a four convolutional layers with fully connected layer called Meso-4 is proposed. The second method, is a modification of Meso-4 which consists of an adaptation from the InceptionNet [70], called MesoInception-4. Their proposed approach is originally tested against DeepFakes using a private database, achieving a test accuracy of 98%.

Nguyen et al. [71], proposed a Capsule Network of CNNs that are trained to detect various tampered regions in deepfake images on the benchmarked FaceForensics++ dataset [4]. The authors extract frames as images from videos to create a face image dataset. The latent feature from the extracted images are learnt using VGG-19 pre-trained CNN architecture. The learnt features are fed to a capsule network of CNNs consisting of three primary capsules and two output capsules for classification of real and deepfake images respectively. A dynamic routing algorithm inspired from [72] is used to route the outputs of three CNN capsules to the two output capsules.

Using pairwise learning

The research on deepfake detection techniques using pairwise learning methods is presently very limited. In 2020, Hsu et al. [26] use a two-stage deep network pipeline using pairwise learning called DeepFD to perform a binary classification task of detecting real vs. deepfake images. The deepfake images are generated from real images extracted from CelebA dataset using four GAN variants: Least Squares GAN (LSGANs) [73], Deep Convolutional GAN (DCGAN) [51], Wasserstein GAN (WGAN)

[74] and WGAN-GP [52]. The primary objective of their research is to be able to detect deepfakes accurately irrespective of the type of GAN used for its generation. The authors hypothesize that each GAN-generated deepfake has specific learning characteristics while generating synthetic images. Hence, the authors explore this property with the use DeepFD. In the first stage of DeepFD, a novel feature extractor to extract common features between images called Common Fake Feature Network (CFFN) is defined. The CFFN uses a traditional Siamese network with DenseNet architecture trained on contrastive loss using transfer learning. The learnt features from CFFN were further passed to a second stage of vanilla CNN network (classification network) to classify real from deepfake images. Fig 2.11 details the CFFN and Classification tasks proposed in this work diagrammatically while Fig 2.10 showcases a schematic representation of the DeepFD pipeline. With regard to the choice of the

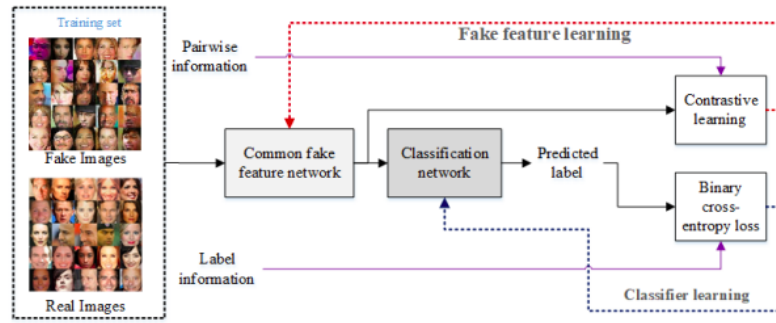


Figure 2.10: Proposed DeepFD pipeline using pairwise learning approach in [9]

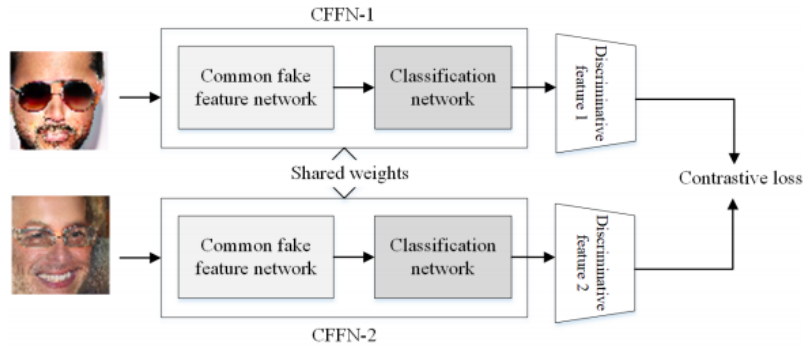


Figure 2.11: Pairwise learning using Siamese Network on Contrastive Loss

deepfakes provided as inputs in the Pairwise information stage, the order is randomized from the four different variants of GANs that are used for its generation. The drawback of this approach was that with new testing on new dataset, the two stage model required re-training to learn the common features. The GAN generated deepfake images produced low realism of deepfakes.

An extension to [26] Zhuang et al. proposed [10] an approach where the CFFN Siamese network is modified with three residual units with different channels wherein the first residual unit consisted of two residual blocks with 96 channels each and the second and third residual units consisted of four and three residual blocks with 128 and 256 channels equivalently. The training loss used is triplet loss which takes pairs of anchor, positive and negative images to the CFFN. The classification layer is retained from the DeepFD approach. The detailed architecture diagram can be found in Fig 2.12. The dataset used in DeepFD was tested in this method and the best performance noted has a precision value of 0.98 and recall of 0.95 for LSGAN generated images (one of the several GANs used to generated deepfake images).

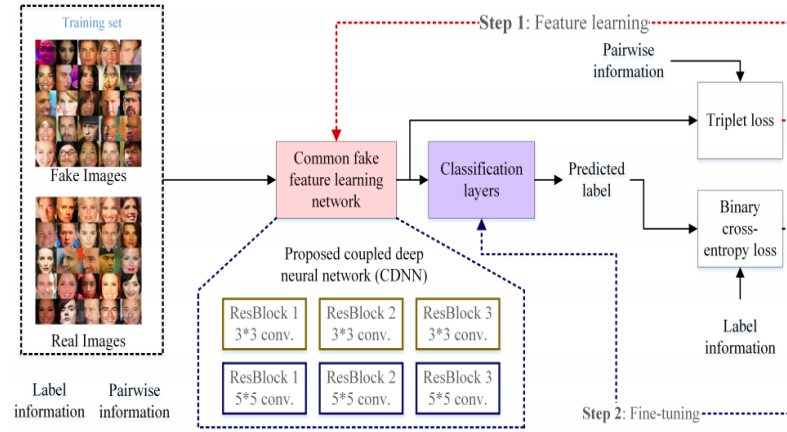


Figure 2.12: Proposed flowchart using CFFN with triplet loss from [10]

2.5. Other Forgery Detection Methods

In this section, the detection methods used for deepfakes is extended to image forgery with focus on signature verification and detection. The reason for discussing a popular forgery detector for signatures, is with regard to the close relatability of the detection task of images with faces. Furthermore, the discussed method here, SigNet is a top performing offline forgery detection method with extended applications to image forgery detections. The SigNet (proposed by Dey et al. [11]) was proposed in 2017 as a method to detect forgeries in signatures and to perform offline signature verification. The model uses CNN architecture that is inspired by ImageNet proposed by Krizhevsky et al. [75] for the image recognition tasks. Furthermore, SigNet uses a Siamese Network trained on Contrastive loss to learn the signature embeddings. The architecture of the SigNet proposed in [11] is detailed in Fig 2.13. As a preprocessing step, the SigNet model is trained on grayscale images. In the training, the SigNet outputs a 128-d embedding of the learnt features. Furthermore, the test inputs are both provided to the trained model in pairs where both the test images are randomly selected. The SigNet classifies the provided test images as similar or dissimilar pairs based on true and false labels. The model was trained and tested on four benchmarked signature datasets and showcased superior performance with accuracy between 85% to 100% on the four datasets.

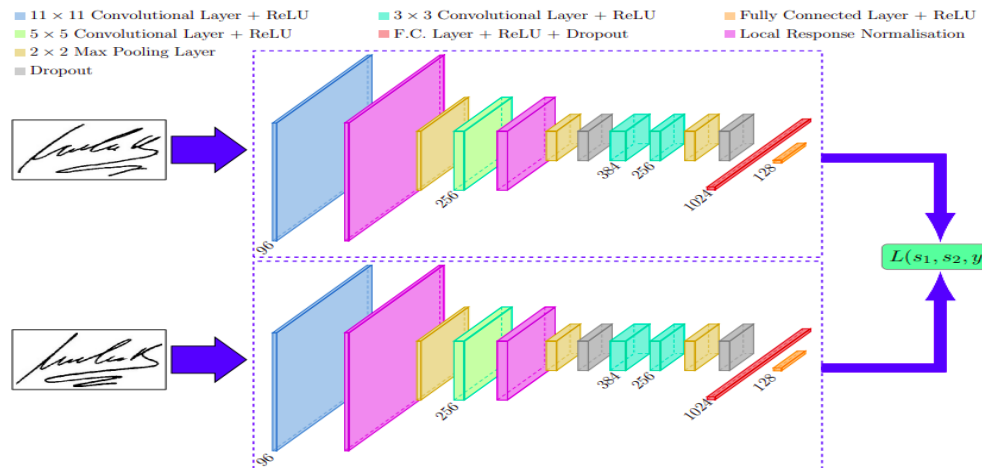


Figure 2.13: SigNet architecture with layer information [11]

The significantly good performance is attributed to the application of pairwise learning approach

used in the SigNet (with the incorporation of Siamese Network using contrastive loss). The primary goal of SigNet is to detect similarity between two provided signatures in the testing stage.

2.6. Summary of Detection Methods for Deepfakes

The existing deepfake detection methods that use facial recognition techniques to detect real vs deepfakes primarily use traditional classification techniques from the literature discussed in Section 2.4. It can be noted that the pairwise learning approaches have not been explored in detail for the detection purposes. Hence, in this thesis work, a detection method with primary focus on pairwise learning is proposed. The reason being, pairwise learning method learns features more discriminatively in comparison with traditional approaches owing to the similarity measure used between input pairs. Furthermore, the generalization ability of the models using pairwise approach, learn better and classify unseen images (faces) accurately as compared to traditional approaches. Lastly, the use of pairwise learning is more applicable in the scenario where the number of real samples (here faces) are lesser than the number of deepfakes available for a specific individual. From the extracts obtained from the literature on deepfake detection methods in current research, this thesis aims to cater to the lesser availability of real images of specific individuals in comparison to the deepfakes. As an addressal mechanism, the work leverages on the use of pairwise learning approach using a modified Siamese Network.

2.6.1. Intuition on real and deepfake data

In the detection methods that are discussed, most of the methods are adopted in the initial stages of deepfake progression. Hence, there was an insufficiency of deepfake generated data. Therefore, deepfake datasets are manually generated from real images or videos using different GAN variants⁶. There is no standardized dataset available containing real and deepfake collated media. With the progression of GAN variants and effective generation methods being introduced, the number of deepfake media focusing on specific individuals increased as compared to the number of real media available on the internet. Also, the application of deepfakes is extended to non-celebrity people (faces) with the introduction of powerful GANs such as StyleGAN [76], CycleGAN [77] that required lesser number of real images to generate realistic deepfakes. This led to an insufficiency in the number of real images or videos that were available for training and testing detection models. Additionally, companies such as Facebook, Microsoft and Google extended their support to the deepfake detection community by providing datasets containing lesser real images or videos as compared to the deepfakes to replicate the real world settings⁷. Therefore deepfake detection competitions and challenges are introduced. Chapter 3 of this report reviews state-of-the-art benchmarked datasets for deepfakes with design choices of dataset used in the experimentation phase of this project.

2.6.2. Modification to Traditional Siamese Networks

In this thesis, the use of pairwise learning is explored further in the detection of deepfakes. A modification is made to pairwise learning method with the help of anchoring (Anchored Pairwise learning). Siamese networks are models that are used for pairwise learning approaches. A variation to traditional Siamese Networks discussed in Section 2.3.4 is proposed called Anchor Siamese Networks (ASN). The modification is made in the testing phase of the Siamese Network where one of the test input pairs is anchored from an external test input selection network. A traditional Siamese Network takes the test input pairs using a sampling strategy from entire test set of data. However, in this approach, the anchored test image is selected from the trained set of images and compared with a counter test pair to compute the similarity score.

⁶A brief overview of deepfake creation (generation) methods are discussed in Appendix A of this report

⁷<https://venturebeat.com/2019/12/11/facebook-microsoft-and-others-launch-deepfake-detection-challenge/>

In the context of detecting deepfakes, the real face images from the training set is fed to an exclusive network called Anchor Face Detector Network called AFDN (discussed in Section 4.4.5) to select the best anchor images which is further used as one of the test inputs. The decision behind selecting the best anchor image can be relative to the problem statement. Furthermore, with the limited anchor images extracted from the training set, a classification task is performed. This way, limited number of training samples are used, thereby solving the problem of lesser real images available in a real-world setting (as in the case of deepfake detection problems). The loss used to train this network can be the same as the traditional Siamese Networks.

3

Selection of Dataset

The emergence of deepfakes and their applications used for incorrect purposes has led to concerns amongst the media forensic community. With the vision of developing good models to detect these deepfakes, there are several publicly available datasets containing deepfake videos and images. This chapter reviews benchmarked datasets that are made available for researchers to attempt to build detection models. Additionally, details of the dataset used in the thesis are also discussed.

3.1. Publicly available datasets

1. UADFV Dataset [78] - The UADFV dataset was the first deepfake benchmark dataset proposed. It contains a total of 98 videos with 49 real videos which were collated from YouTube and 49 deepfake videos which were created using FakeAPP (software used to generate deepfake videos).
2. Deepfake-TIMIT dataset [54] - The Deepfake TIMIT dataset is a database of videos which are generated using Faceswap-GAN¹. The dataset contains 16 manually selected pairs of people's videos from VidTIMIT dataset [79]. The pair selection was based on similar facial features such as hair color, hair style, facial hair etc. The total of 32 faces were trained on two models. The first model was a low-quality 64x64 dimension model and the second model was a high-quality 128x128 dimension model. For the low-quality model, about 200 frames of each person's video was trained and extracted at 4 frames per second (fps) from the source video. The high-quality model, about 400 frames of each person's video was trained and extracted at 8 fps from the source. The total number of deepfake videos that were generated were 620. Each real person's video contains around 20 deepfake videos (10 using the low-quality model and 10 with high-quality model).
3. FaceForensics ++ [4] - FaceForensics++ is a large scale forensics dataset that contains 1000 real videos that were manipulated using four manipulation techniques. They are deepfakes, face2face, faceswap and neural textures. The real videos were collated from YouTube of non-celebrity faces. The FaceForensics++ dataset was co-funded by Google. The deepfake videos were generated using modified faceswap algorithms. The deepfake videos were masked using post-processing for enhanced realism. The masking was performed using Poisson image editing technique [80]. The total number of real and deepfake videos amount to 1000 per class.

¹<https://github.com/shaoanlu/faceswap-GAN.git>

4. DFDC [67] - Facebook released a deepfake dataset as part of the deepfake detection challenge in 2019 called DFDC. The dataset contains 5000 videos (deepfake:real videos in a ratio of 1:0.28 respectively) that features two facial modification techniques (the details of the techniques were not disclosed). The data was gathered from collection campaigns with voluntary participants for the real videos using crowdsourcing.
5. Google's Deepfake Detection Dataset (DFD) [81] - Google in collaboration with Jigsaw created a dataset using multiple GANs to create 3000 deepfake videos. The method of generation was not disclosed by Google. The dataset contains 28 real videos of actors who volunteered via crowdsourcing. The 28 actors were deepfaked to create 3000 manipulated videos.
6. Celeb-DF v2 Dataset [12]- The Celeb-DF dataset is a celebrity face dataset containing real videos from YouTube interviews of famous celebrities. Real videos of 59 celebrities were collated with a total 590 real videos. For deepfake video synthesis, a custom deepfake algorithm was created to generate 5639 videos.

3.2. Comparison between the datasets

This section, compares the datasets discussed in Chapter 3.1 from different metrics to choose the ideal dataset used in this thesis work. The rationale is evaluated based on the following metrics:

3.2.1. Based on number of real vs. deepfake videos

Table 3.1 provides the details of the number of real videos to the deepfake videos. In real-world settings, the total number of real individual images or videos are limited and the possibilities of generating many deepfakes is very high. UADF, Deepfake-TIMIT (low-quality and high-quality) and FaceForensics++ datasets contain the same amount of real and deepfake videos which do not mimic the real world settings. Among, DFDC, DFD and Celeb-DF v2, the real-deepfake ratio of videos present in the dataset is the lowest for Celeb-DF v2.

Table 3.1: Details of datasets discussed in Chapter 3.1

Dataset	# of Real Videos	# of Deepfake Videos	Real-Deepfake ratio
UADFV	49	49	1:1
Deepfake-TIMIT (Low-Quality)	320	320	1:1
Deepfake-TIMIT (High-Quality)	320	320	1:1
FaceForensics++ (Deepfake)	1000	1000	1:1
DFDC	1131	4113	0.31:1
DFD	363	3068	0.11:1
Celeb-DF v2	590	5639	0.1:1

3.2.1.1. Realism of deepfakes

The visual quality of deepfake videos generated by Celeb-DF v2's deepfake algorithm provides higher realism than other datasets discussed in this chapter. The performance of visual quality is evaluated based on Mask-SSIM (introduced in [82]) as a quantitative metric. Mask-SSIM is an attribute of SSIM score that references comparisons quality of deepfake generated to the real source with head regions of the face (including face and hair). A low score implies the visual quality to be less realistic (meaning the deepfake generated contains distorted or tampered regions detectable by human perception). The Mask-SSIM score is computed frame-wise for each real and deepfake video and aggregated to make final conclusions between all datasets. The Mask-SSIM scores reported in Table 3.2 are normalized between [0,1].

Table 3.2: Average Mask-SSIM score comparison for different datasets

Dataset	Average Mask-SSIM score
UADFV	0.82
Deepfake-TIMIT (Low-Quality)	0.80
Deepfake-TIMIT (High-Quality)	0.80
FaceForensics++ (Deepfake)	0.82
DFDC	0.84
DFD	0.88
Celeb-DF v2	0.92

3.2.1.2. Deepfake quality of bad samples

Fig 3.1 shows a comparison of the low-quality of synthesized faces across different datasets. The faces in the figure represent the bad samples of faces generated by the respective deepfake algorithms.

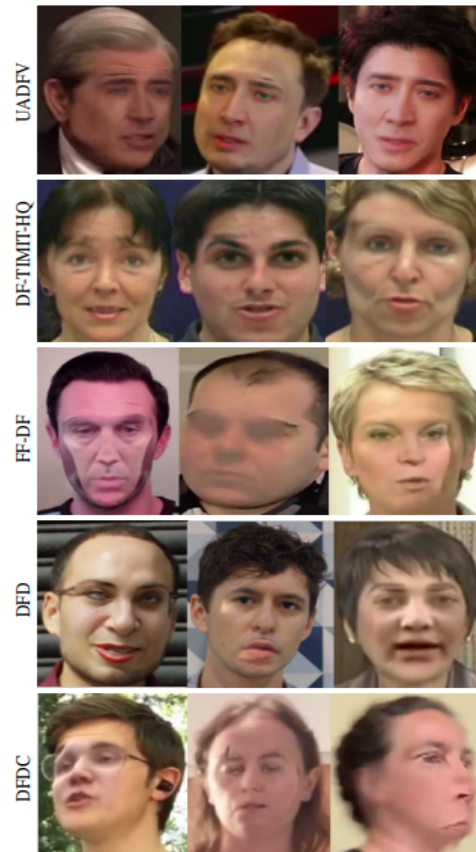


Figure 3.1: Visual low-quality of bad samples from UADF, DF-TIMIT, FaceForensics++ (FF-DF), DFD and DFDC datasets [12]

The low-quality of bad samples generated are corrected in the Celeb-DF v2 dataset's deepfake synthesizing algorithm called *Deepfake Maker Algorithm*. The algorithm learns the common features independent of the identity of the face to generate deepfakes (generation process explained in Chapter 3.3). The use of Deepfake Maker Algorithm produced high quality sample faces (Refer 3.2).



Figure 3.2: Sample frames from Celeb-DF v2 dataset [12]. First image denotes the real frame (source) and subsequent frames denote the deepfakes generated

3.3. Celeb-DF v2 Data Generation

The real videos from the Celeb-DF v2 dataset was collated from YouTube focusing on celebrities. As for the synthetic media generation, a *Deepfake Maker Algorithm* was implemented. The following are the steps of generation for the Deepfake Maker Algorithm proposed in [12]:

1. An input real video is detected for faces (frame-wise) and the corresponding facial landmark points are extracted.
2. The landmark points are used to perform face alignment adapted from [83].
3. The aligned faces are further cropped around the face region with the size of 256x256 pixels.
4. An AE network is trained on the cropped faces to generate synthetic faces using similar facial expressions on the destination frame.
 - The AE network contains two CNNs (encoder and decoder architecture). The encoder network converts face images into representations (feature vectors).
 - A single encoder network is used to learn all features regardless of the identity of the face in the source video (also called identity-independent features).
 - However, each identity has a dedicated decoder network that generates a synthetic face from the learnt features.
5. The deepfake faces are warped to the source faces and trimmed with a mask using facial landmark points.
6. Post-processing of the deepfake faces are performed using boundary smoothing techniques and the frames are concatenated into a video.

4

Experimental methodology

In this chapter, the experimental setup of the baseline model and a deep learning method (called Anchored Siamese Network) are discussed in detailed. All experiments are implemented in Python 3+ with the modules that enable face recognition (OpenCV, dlib), computation of linear algebra operations (Numpy, Scipy), implementation of machine learning and deep learning algorithms (scikit-learn, Keras with TensorFlow backend and PyTorch) and parallel computational task (DASK). Prior to understanding the details of the experimental setup, the strategy used to split the Celeb-DF v2 dataset is discussed.

4.1. Dataset Split

The Celeb-DF v2 dataset contains real and deepfake videos of 59 celebrities in total. For experimentation purposes, the dataset is split into three sets. They are training set, test set and independent hold-out set. (The detailed image-wise split of deepfake faces are discussed in Appendix B). The videos are split based on celebrities. The training and test sets are split among 50 celebrity videos ¹. The percentage allocation for the splitting is 80% for the training set and 20% for the test set. The remainder 9 celebrities are treated as an independent hold-out set (called hold-out set). The rationale of using a hold-out set is to effectively assess the generalization capability of the proposed methods to unseen faces. (Refer Table 4.1)

Table 4.1: Celeb-DF v2 dataset split for experiments (video reference)

Dataset	# Celebrities split	# Source Reference	Real Videos	Deepfake Videos
Training Set	40	id00 to id42	400	4308
Testing Set	10	id43 to id52	100	735
Hold-out Set	9	id53 to id61	90	596

All the videos in Celeb-DF v2 dataset are in range of 10-13 seconds with a constituent of 10 videos per celebrity (real class). The deepfake videos are generated based on swapping faces for each pair of 59 celebrities. In total each celebrity is varied from 6-15 different deepfake pairs. The choice of celebrities used for swapping is not disclosed by the authors of the dataset. The videos have a 30fps frame-rate. A total of 590 videos (across all celebrities) belong to the real class and 5639 videos (across all celebrities) belong to the deepfake class. Since the thesis focuses on detection of deepfake images, frame (images) extraction technique (using OpenCV) is implemented to extract images from each video. The training images are extracted at 3fps for each video for the real class

¹each celebrity's video is referenced with nomenclature of id##. id14, id15 and id36 are not present in the dataset

and 1fps for the deepfake class. The reason behind choosing 1fps for deepfake extracting deepfake training images as opposed to 3fps for real training images is due to the large availability of deepfake image possibilities in comparison to the lesser availability of real images. The real and deepfake test images are extracted at 3fps per video. The independent hold-out set is extracted at 1fps per video for real and deepfake class. The resultant real and deepfake image compositions for each dataset split is detailed in Table 4.2. The training, test and independent hold-out set composition is retained across all the models discussed in this chapter.

Table 4.2: Celeb-DF v2 dataset split for experiments (frame reference)

Dataset	# Real images (frames)	# Deepfake images(frames)
Training Set	15604	56698
Testing Set	3076	24420
Hold-out Set	1310	8823

4.2. Preprocessing

In the preprocessing stage, face detection and face alignment operations are performed to the extracted frames (images) from the videos of the dataset.

4.2.1. Face detection

To detect and crop the face regions from the real and deepfake images, two face detection methods are implemented in parallel and the best performing method based on better output of faces extracted is chosen. The two detection methods used are Viola-Jones's Haar Cascade feature detection method and MTCNN face detector. Both the methods are implemented using OpenCV and dlib face detection packages in Python. The detected faces are further cropped to 154x155 dimensions around the face region. Table 4.3 summarizes the total number of faces detected from the frames extracted from videos per dataset type. All the extracted frames contain one face with no occlusions or multiple faces in a single frame reference. Comparing the number of faces detected between Viola-Jones detector and MTCNN, the latter performs better in detecting faces from the total extracted frames. MTCNN produces an accuracy of 99.8% as compared to Viola-Jones detector method which reports a 85%. The possible reason for the failure to detect some faces from the extracted frames can be attributed to the quality of the source images in the dataset or the pose variations that the both the detectors failed to capture.

Table 4.3: Viola-Jones feature detector vs. MTCNN

Dataset	Total extracted frames	Viola-Jones detector	MTCNN
Training Set	72302	61055	71985
Test Set	27496	23602	27399
Hold-out Set	10133	8852	10086
Total	109931	93509	109884

Additionally, as a mechanism to identify false positives in the face detection between both the methods in discussion, a random inspection of the detected face images are performed. The inspection is performed manually and is restricted to 250 images per method. Viola-Jones detector method produced 31 false positive faces while there were none noted in faces detected by MTCNN from the 250 sample space (Fig 4.1 showcases some non-faces images as false positives detected by Viola-Jones detector) . Furthermore, an intuition from the results obtained in the work of Zhang et. al [41] conclusively provides sufficient rationale on the ideal face detector to be used in further experiments to be MTCNN. In the training set the final number of real faces are 15384 and deepfake faces are

56601. From the test set, there are 3017 real faces and 24382 deepfake faces. As for the independent hold-out set there are 1301 real faces and 8785 deepfake faces.

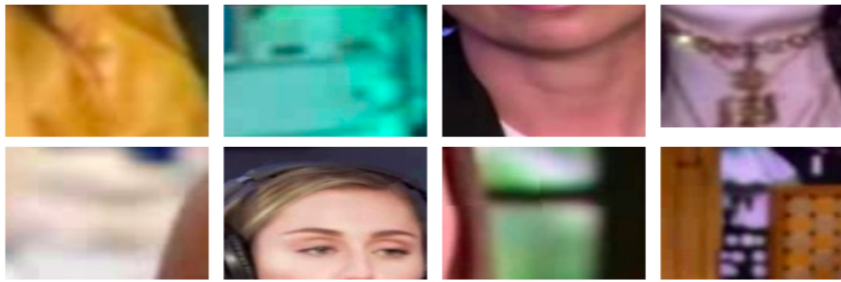


Figure 4.1: Sample false positives detected by Viola-Jones detector by random inspection

4.2.2. Face Alignment

The detected faces are referenced to a template using 2D transformation in the face alignment phase. The face alignment method used in the experiments is adapted from the Precise Alignment approach explained in Section 2.3.2. The facial features are five point facial landmark that are obtained from MTCNN face detector. The central coordinates is referenced using the nose coordinates. The reference central coordinates correspond to the eyes, nose and lip edges in the face. A sample face alignment procedure is diagrammatically represented in Fig 4.2 from a training face from the dataset, based on the Precise Algorithm.

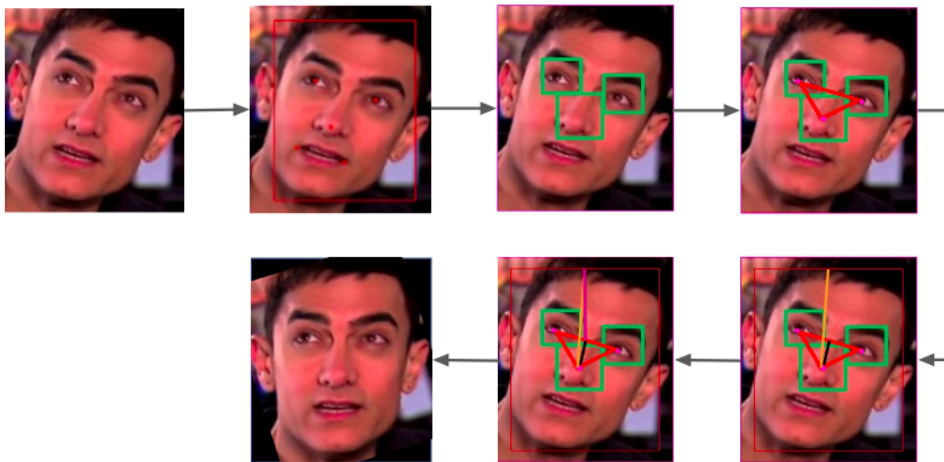


Figure 4.2: Face alignment for sample face from the dataset

4.3. Baseline Method

The baseline method implemented in this thesis is a holistic approach using Eigenfaces. Eigenface is an appearance based method ² with primary use in face representation, detection and recognition systems. Pentland et. al [84] proposed the eigenfaces in 1991 as a face recognition technique. Eigenfaces use Principal Component Analysis (PCA) as a dimensionality reduction technique. This method inputs the entire face and converts them to feature vectors, on which statistical operations

²Appearance based methods use no a priori knowledge on the data present in the image. Instead, they try through statistical analysis of the available dataset (either an image or image characteristics database) to extract the different variation modes of the database and provide a set of sub-classes which represents them best.

are performed to further classify or recognize faces (based on use-case). The eigenfaces detects and identifies faces with applications in real-time face recognition system used to track comparative characteristics between input faces. The method leverages the 2-D characteristic view of the faces by projecting faces onto an embedding space (called Eigen Face Space) to encode variations noted between faces. The Eigen Face Space constitutes eigen vectors (called eigenfaces) that a representative feature vectors. These features are further trained using a Logistic Regression classifier to predict deepfake or real faces on a test set. Fig 4.3 represents the workflow pipeline of the baseline method using Eigenfaces.

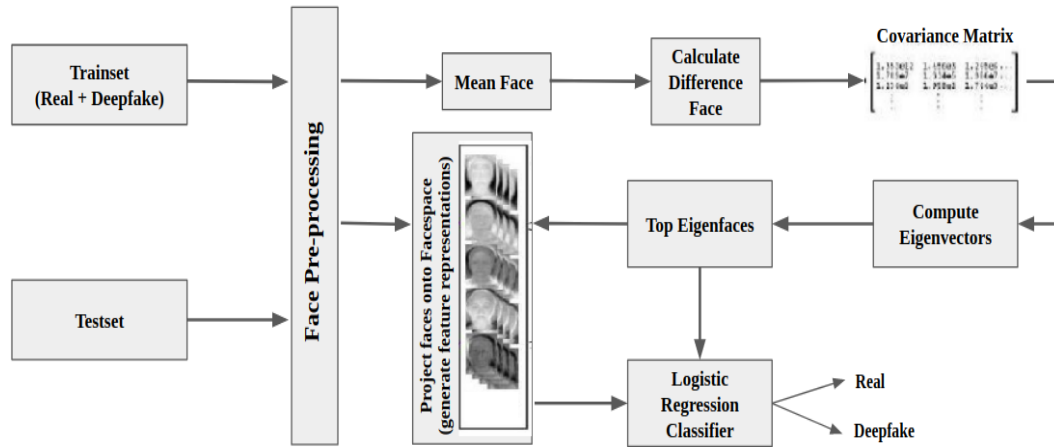


Figure 4.3: Baseline Method workflow

4.3.1. Additional Pre-processing

In extension to the preprocessing steps performed on the Celeb-DF v2 images, discussed in Section 4.2, the Eigenface approach requires additional steps to obtain the faces. The reason being, holistic methods (such as Eigenfaces) are sensitive to illumination, pose, scale sensitivity in the source images. Hence, the aligned faces obtained from the geometric normalization stage are further pre-processed for achieving good representations.

- RGB to Gray scale: Converts the input face present in 3-channel RGB to grayscale faces.



Figure 4.4: Sample face image after additional pre-processing operations

- Histogram Equalization: equalizes the histogram with regard to the unmasked regions in the face image. This balances the color and achieves maximum pixel depth of the face region in the cropped image.

A sample face image after performing additional pre-processing for the baseline method is represented in Fig 4.4.

4.3.2. Feature Extraction of Eigenfaces

A database containing real and deepfake training faces are collected (called training-DB) after performing additional pre-processing steps on the faces. The faces in the training-DB are flattened to a vector of length $(155 \times 154, 1) = (23870 \times 1)$. The resultant matrix T contains all the column vectors of the faces from the training-DB with a size $(23870 \times N)$ where N is the number of training faces in the training-DB. The training-DB contains a collection of 71985 thereby having a vector of size (23870×71985) .

Using classical PCA, the resultant covariance matrix obtained after computing the mean and subtracting the images from the mean, provides a large matrix thereby making PCA computationally intensive. Hence, an approximation to PCA is adopted in this work called Randomized PCA (rPCA) [85]. The reason for the choice of rPCA is for a covariance matrix that is large to fit in fast memory, rPCA requires only a constant number of passes over the data, as opposed to (k) passes for classical PCA approach.

rPCA computes the mean faces from the matrix T and subtracts each face vector from the mean (same steps as classical PCA). Fig 4.5 represents the mean faces generated using rPCA from T . Fur-

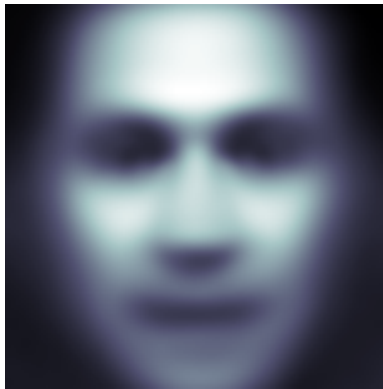


Figure 4.5: Obtained mean faces from rPCA

ther, to obtain the eigenfaces, a low-dimensional subspace (based on the assigned principal component) rPCA performs random sampling to capture the action of the covariance matrix. The matrix is then compressed either explicitly or implicitly to the subspace, and the reduced matrix is manipulated deterministically to obtain the desired low-rank factorization. This reduces the computational costs required for large covariance matrix.

Fig 4.6, represents the cumulative variance vs. the principal components. It can be noted that for 350 principal components a maximum variance is achieved. Hence, matrix T is reduced to (23870×350) . A Sample of the top 24 eigenfaces (from the selected 350 eigenfaces) are encoded back to face representations in Fig 4.7. The generated top eigenfaces constituting (23870×350) is the extracted feature vector (representation).

4.3.3. Training

With the extracted eigenfaces from the feature extraction stage, of the baseline method, a logistic regression classifier is trained on the eigenfaces and tested on a reserved test set to classify the real and deepfake images. The ratio of the class of real vs. deepfake faces are 1:3.5. Hence, to address the class imbalance in the training stage of the logistic regression classifier, the class weights are re-weighted in accordance to add priority in classification decision for the low real class of faces. As for the implementation of the logistic regression classifier, a grid search is performed to obtain

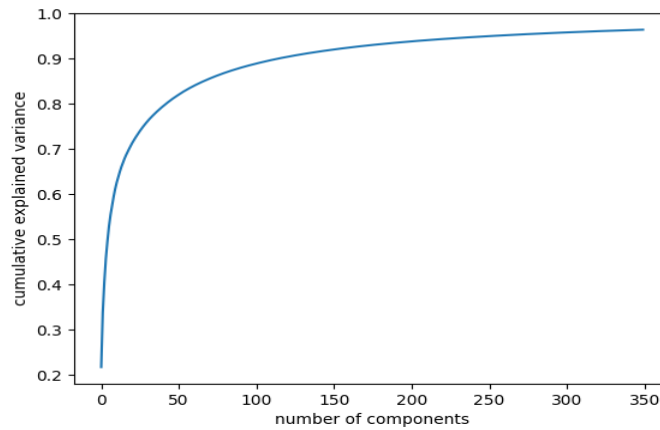


Figure 4.6: Cumulative variance vs rPCA components



Figure 4.7: Sample top eigenfaces

the optimal hyperparameters during training. The best regularizer was L2 regularization with an inverse regularization strength parameter (C) to be 10.

4.4. Anchor Siamese Network

The deep learning method proposed in this thesis uses a modified Siamese Network called Anchor Siamese Networks (ASNs) that uses pairwise learning approach to compute similarity between given input pair of images. From Fig 4.8, in the Training Stage, the input pairs of training images are chosen in combination of real-real, real-deepfake and deepfake-deepfake order. The input pairs are passed to a ResNet-50 ASN architecture to generate face representations. The ResNet-50 ASN architecture is a Siamese Network (modified only in the testing stage) using pre-trained ResNet-50 model. This model is trained on contrastive loss. The detection metric evaluates the similarity between input pairs on a common embedding space using a Euclidean distance between the pairs.

The modification is made to the ASN is in the Testing Stage of Fig 4.9. In a traditional sense, the Siamese Networks receives two varied test inputs as pairs for testing. However, in this work, one of test pairs is anchored from a Anchor Face Detector Network (AFDN) that contains the real images of all celebrities of the Celeb-DF v2 dataset split class-wise (discussed in detailed in Section 4.4.5).

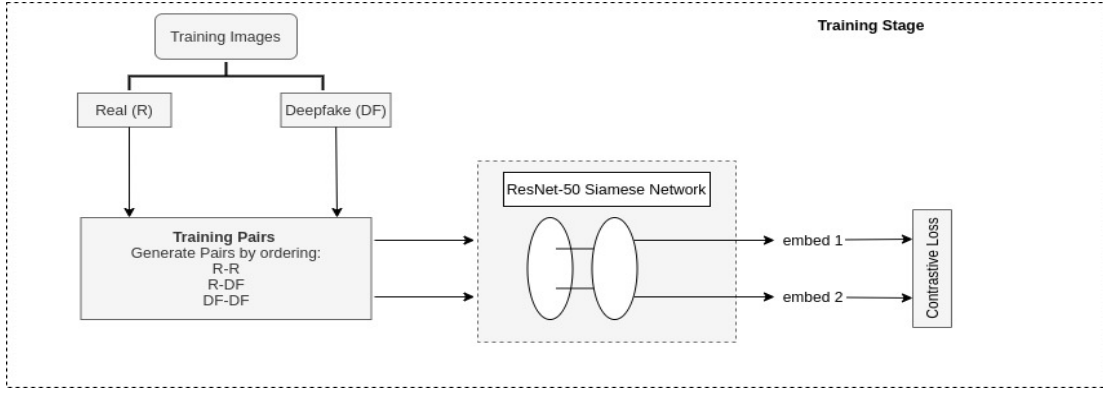


Figure 4.8: Training Stage of the proposed ASN using ResNet-50 architecture

The AFDN is a face detection database that outputs one real image per celebrity class based on the highest confidence score of face detection (using HoG descriptor method). The other input test pair, is obtained from the test set. A similarity score is computed across all test pairs generated and each input pair obtained from the test set is averaged across all anchor input pairs selected from the AFDN to make a decision on real or deepfake class.

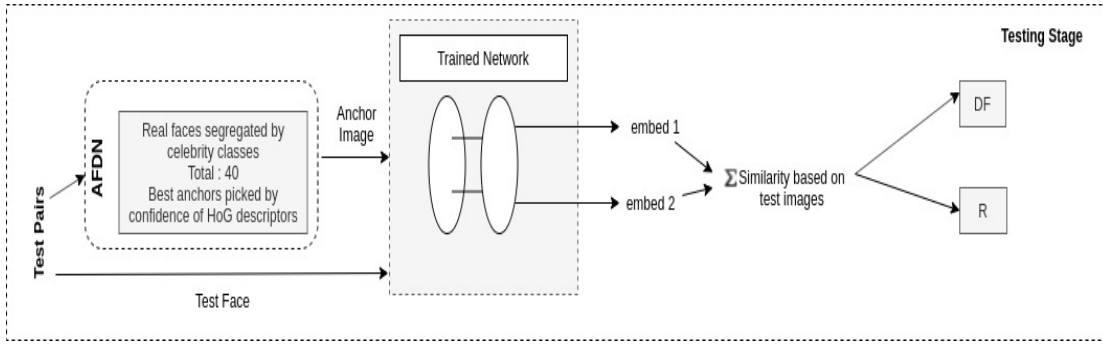


Figure 4.9: Testing Stage of the proposed ASN using ResNet-50 architecture

4.4.1. Input-pair selection

In the training phase, the choice of input-pairs is an important factor in pairwise learning approach. The pairs are ordered in sequences of real-real, real-deepfake, deepfake-deepfake. The reason being ordering of pairs is for the network to learn all possible combinations of discriminative inputs. Let the set of faces from the training set be defined as: $\mathbf{X}_{real} = [\mathbf{x}_{r=1}, \mathbf{x}_{r=2}, \dots, \mathbf{x}_{r=N_r}]$, denoting real faces where N_r refers to the number of training real faces and $\mathbf{X}_{dfake} = [\mathbf{x}_{d=1}, \mathbf{x}_{d=2}, \dots, \mathbf{x}_{d=N_{df}}]$ denoting deepfake faces where N_{df} refers to the number of training deepfake faces. The total number of training faces is denoted as $N_{Train} = N_r + N_{df}$. The associate label for each classes are denoted by $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$ where $\mathbf{y} = 0$ denotes real class and $\mathbf{y} = 1$ deepfake class.

The pairwise inputs are extracted from the training set $\mathbf{X}$ along with the corresponding labels $\mathbf{Y}$ using random sampling from each class following the order of first selecting a two faces forming a pair from $\mathbf{X}_{real}$, further selecting one face from $\mathbf{X}_{dfake}$ and one from $\mathbf{X}_{real}$ and finally two faces from $\mathbf{X}_{dfake}$ iteratively. Since there are lesser number of $\mathbf{X}_{real}$ faces in comparison with $\mathbf{X}_{dfake}$, the real faces are again selected upon exhaustion of all samples. Additionally, the total number of input-pairs that can be generated from this set is $C(N_{Train}, 2)$. Fig 4.10 displays a sample set of input pairs of all three possible combinations fed to the ASN during training stage.

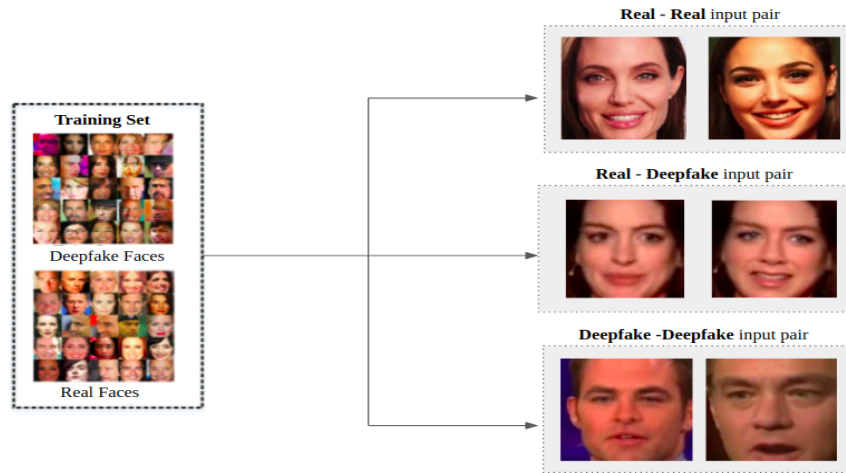


Figure 4.10: Sample Input pair ordering

4.4.2. Network Selection

The ASN uses a ResNet (Residual Networks) pre-trained architecture to learn the discriminate features from the input pairs. ResNet was first introduced by He et al. [13] in 2015 and was the winner of the ImageNet Challenge (2015). ResNet uses skip connection conceptualization that skips selective layers in the deep model as a method to address the vanishing gradient problem in dense networks [86]. The skip connections is a way of ensuring uninterrupted gradient flow from the initial to last layers of the network by providing an alternative pathway for gradient (in backpropagation). With the skip connection, the original input information is added to the output of the last stacked convolutional layer. The ResNet model stacks the residual blocks together thereby preserving gradient

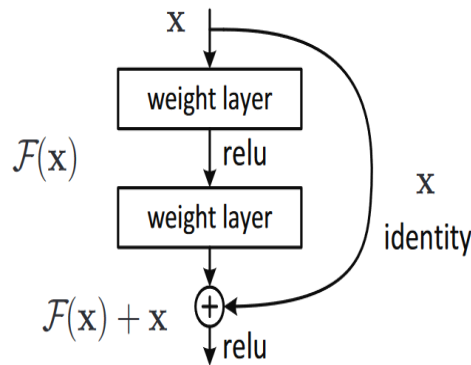


Figure 4.11: ResNet skip connection concept [13]

information. Therefore during backpropagation, the gradient is multiplied by one (identity function) and the resultant value is maintained in the previous layers. In Fig 4.11, let $H(\mathbf{x})$ be a underlying mapping of a set of stacked convolutional layers. $\mathbf{X}$ denotes the input faces to the first layers of the network. Multiple non-linear functions asymptotically approximate a residual function $F(\mathbf{X}) = H(\mathbf{X}) - \mathbf{X}$ thereby becoming $F(\mathbf{X}) + \mathbf{X}$. This is representative of element-wise addition thereby deriving an alternative name of "skip connections via addition" for residual networks.

In addition to addressing the vanishing gradient problem, another advantage of using skip connections is learning lower semantic information being passed from the initial layers to the dense layers due to skipping of some intermediary layers of the network. This was experimentally observed by

the authors in their research "It has been observed that in earlier layers the learned features correspond to lower semantic information that is extracted from the input. [13]". Therefore, this enables feature re-usability mechanism and ensures stabilized training convergence [87].

The ResNet architecture provide various pre-trained options with varying layers of residual networks. The choice of ResNet variant in this thesis is a ResNet-50 (a 50 layer residual network). The model was implemented as an ASN. The network was implemented using PyTorch's transfer learning module³. Therefore, the pre-trained network is trained entirely on the training set with pre-loaded parameters. In this experiment, no intermediary convolutional layers were frozen from the pre-trained network. Fig 4.12 illustrates a schematic representation of the ResNet-50 architecture with the associated layer information. The modification made from Fig 4.12 is the removal of

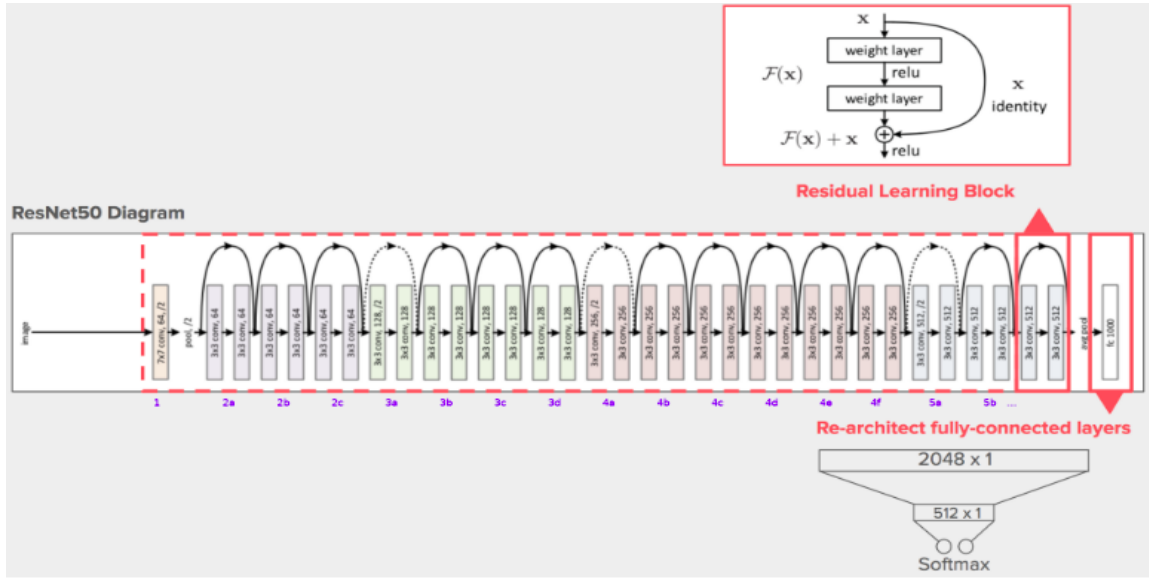


Figure 4.12: ResNet-50 architecture

the fully connected layer (FCNN) from the pre-trained model and adding an FCNN to accept 1000-dimensional feature embeddings (also called face embeddings).

4.4.3. Network Loss

The ASN uses a twin ResNet-50 pre-trained model for training on the input pairs. A deep metric loss is used for training the ASN. Deep metric losses explicitly align similar faces in a cluster (or group) onto a common embedding space and distance away dissimilar faces from each other. The penalty of the mappings during training from the input pairs are computed using the deep metric loss function. This adjusts the network weights thereby optimizing the network. The deep metric loss used in this experiment is Contrastive loss. Contrastive loss addresses a distance measure (Euclidean distance), between input pairs thereby pushing away dissimilar pairwise embeddings or clustering similar pairwise embeddings.

4.4.4. Training

The training phase of the deep learning model uses a ASN with ResNet-50 pre-trained model choice. The classification layer of the ResNet-50 model is replaced with a FCNN that generates 1000-d feature embedding for each trained face. The training set containing the faces of real and deepfake images are passed to the ASN in pairs following the sequence described in Section 4.4.1. The pre-

³<https://pytorch.org/docs/stable/torchvision/models.html>

trained ResNet-50 architecture takes input images of shape 224x224x3. Hence, a custom data-loader module (using PyTorch) is implemented to transform the faces to suit the input size of the ResNet-50 model of choice. During the training stage of the ResNet-50 model, the epochs are restricted to 271 with a batch size of 16. The pre-trained hyper-parameters are retained and fine tuning is not performed. The loss gradually decreases through 271 epochs. Therefore, a substantial decision to not fine-tune the model further is made.

4.4.5. Anchor Face Detector Network

In traditional approach of using Siamese Networks for classification, the test input pairs are randomly or pre-orderly selected from the test set. However, a slight modification is proposed in this work. During the testing stage, one of the pair of test pairs are anchored from the real faces of the training set and the similarity between the anchor face with a test face is computed to determine whether the test face belongs to a deepfake or real class (classification task). The purpose behind this modification is to efficiently account for the lesser availability of real faces in the dataset. To decide the best real anchor face to be inputted with a test face as a test pair, an AFDN is used. The

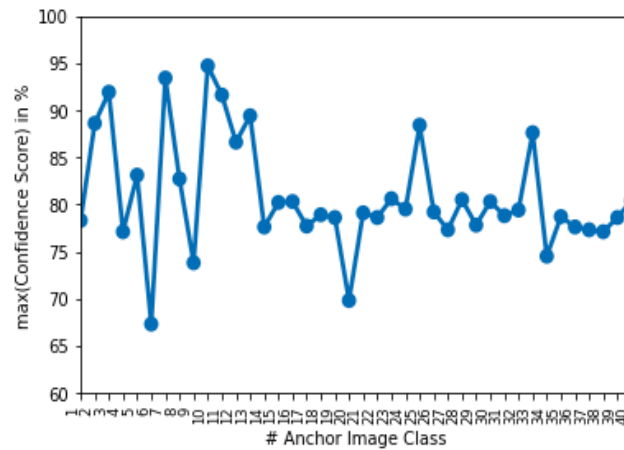


Figure 4.13: maximum confidence score reported per anchor real class

AFDN contains a Face Database of real faces from the training set of the dataset that is categorized based on the celebrity classes available. There are a total of 40 celebrities that are present in the training stage of the dataset from which the real and deepfake faces were extracted. All the real faces were categorized into the respective celebrity faces thereby amounting to 15384 faces categorized into 40 classes (based on celebrities). The output of the AFDN is one best real face per celebrity class.

To find the anchor real face per celebrity class, a pre-trained HoG descriptor approach is implemented from dlib to obtain the group of gradient directions from the face image. The design choice behind using HoG as a feature descriptor method is that the local object appearance and shape within a face can be described by the distribution of intensity gradients or edge directions. Additionally, HoG operates on localized cells of a face thereby upholding to in-variance of geometric or photo-metric transformation. This makes it suitable for face detection. Procedurally, every pixel in the face has gradients that are grouped into a number of orientation bins. The gradient magnitudes are further summed per sample thereby adding more weight to the higher contributing gradients (also called stronger gradients). Iteratively performing this operation on all the pixels provides a representation called descriptor characteristics. The representations are classified using a pre-trained linear SVM. The entire approach is available as a pre-trained function in dlib called (front_face_detector). The HoG + Linear SVM localizes the faces in an image. A confidence score

is reported for each face detection performed by HoG + Linear SVM approach. The face detection method returns a score based on the confidence of detecting the face with the identified descriptors (Fig 4.14 displays detected faces from the HoG detector and Table 4.4 summarizes the corresponding returned scores obtained for sample real anchor faces).

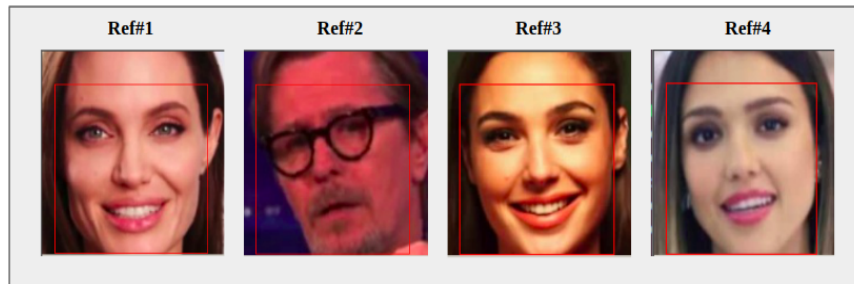


Figure 4.14: Detected sample faces with bounding box by HoG detector

A confidence percentage is computed from the obtained scores (the range of score are set be-

Table 4.4: Summary of scores and confidence measures for sample faces from Fig 4.14

Reference	Reported Score	Confidence in %
Ref1	3.218	91.9
Ref2	2.517	71.8
Ref3	2.991	85.4
Ref4	2.863	81.8

tween 0 and 3.5 as per dlib's HoG + Linear SVM predictor module). The confidence score defines the probability of the model's confidence in predicting the face with a bounding box. The face with the highest confidence measure is selected as the anchor face per celebrity class. In the event where there are two real faces per celebrity class having the same confidence score, a random picking between the two faces is considered as the output of the anchor real face.

This way, the AFDN provides 40 anchor real faces covering one face per celebrity class. These 40 anchor real faces are used as one of the test pairs along with all the test faces from the test set to a similarity score. Fig 4.13 showcases the maximum confidence scores obtained per anchor real class reported by the face detector method. The range of all the maximum confidence of face detection reported is between $\sim 67\%$ to 95% from the 40 celebrity class faces.

5

Results

This Chapter discusses the results obtained from the baseline method and ASN using AFDN, discussed in Chapter 4. The results are evaluated based on the performance of the models on the test set. Furthermore, this chapter reviews the generalization performance of the proposed ASN method on the Independent hold-out set and, a comparison with state-of-the-art deepfake detection method from literature is also made. Pre-trained models are used in this regard and the selected models are tested on the independent hold-out set.

5.1. Test Results of Baseline Method

The trained logistic regression model is tested on the Celeb-DF v2 collected test faces. The performance of the model is evaluated using AUC score from the ROC plot. The ROC curve is extended to a convex hull of the ROC curve points. An intuition is drawn from [88], where two existing classifiers from the ROC curves are compared to obtain a new classifier whose performance (in terms of its ROC) lies on a line connecting the performance of its two components. This is done by choosing one or other of the classifiers at random and using its result for extrapolation. This method of tuning is called Scott's Maximum Realizable ROC (MRROC). Scott's random combination method can be applied to each set of points along the ROC curve. So Scott's MRROC is the convex hull of the classifier's ROC. The MRROC is used in as the consistent ROC plots for all experiments in this thesis. Fig 5.1 showcases the ROC of the classification result on the test set.

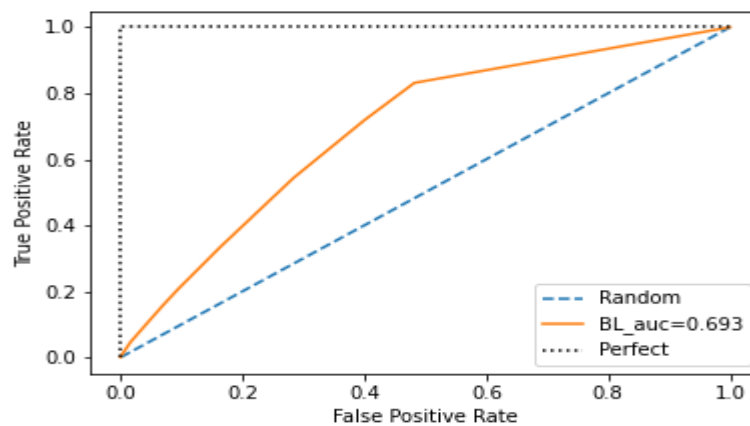


Figure 5.1: ROC on test set for Baseline method

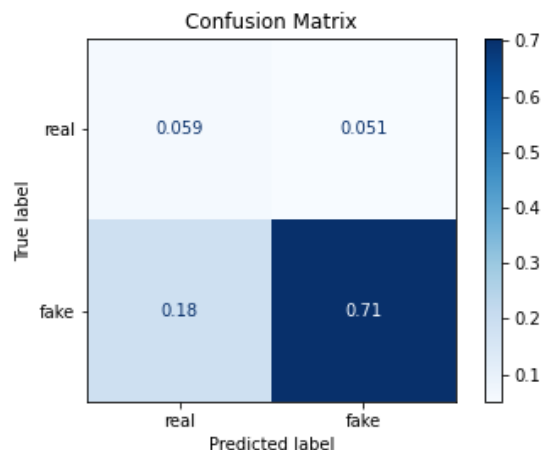


Figure 5.2: Normalized Confusion Matrix of the baseline method

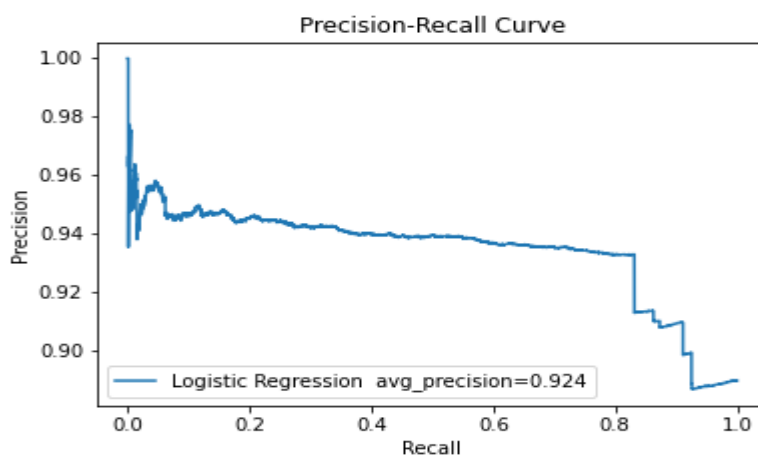


Figure 5.3: PR Curve for baseline method

The AUC obtained from the ROC curve is 69.3%. This implies that the baseline method showcases poor performance in making correct predictions of faces belonging to real or deepfake classes. This is observed from the low True Positive Rate (TPR) that defines the number of correct predictions made by the model to detect the output class labels (real or deepfake in this case). At the optimal threshold point of 0.64, the baseline method reports an accuracy of 76%. Fig 5.2 showcases the confusion matrix of the classifier performance. From the correctly classified labels, the model has a higher prediction rate on the deepfake classes, implying that the number of correctly predicted deepfakes is high (normalized score of 0.71). However, the model performs poorly in being able to make correct predictions from the real class (normalized score of 0.059).

To further understand the model's ability to make relevant predictions, the Precision-Recall curve (PR) is plotted across various thresholds. In the PR curve, the model is evaluated on the True Positive Rate (TPR) which implies the number of true positive predictions (real predictions) over the total predictions (correctly classified and incorrectly predicted real faces) with the total number of relevant correct predictions. The PR curve is plotted in Fig 5.3. The precision (also called TPR) is high for lower recall. With varying thresholds, the precision decreases non-monotonically with recall measure being significantly constant. The average precision for across all thresholds is 92.4%.

A primary reason accountable for the poor performance of the baseline method on the test set, is that Eigenfaces (or holistic class of face recognition methods) does not learn the discriminative features of the face attributes like eyes, nose etc. In the setting of deepfakes, the primary generation of deepfakes is to modify the facial attributes which Eigenfaces is agnostic to. With the application of rPCA, the features are reduced in dimension, however, the reduction is not made with regard to the facial attributes. Furthermore, the pose variations in the faces (despite performing face alignment) is still not accounted by the Eigenface algorithm, thereby leading to poor performance. Additionally, in the work proposed by Slavković et. al [89], an observation made is that Eigenfaces require faces to be present in a uniform background. In the case of a natural setting as found in the Celeb-DF v2 dataset, this is not applicable.

Overall, the choice of the baseline method using Eigenfaces is owing to its simplicity of implementation and further understand the performance of deepfake detection using a naive holistic face recognition system. From the derived results, the baseline method clearly performs poorly in being able to provide correct predictions of real and deepfake classes. On the contrary, a deep learning based method is further implemented to learn better feature discriminative characteristics present in a face using a pairwise learning approach.

5.2. Small experiment on ASN Method

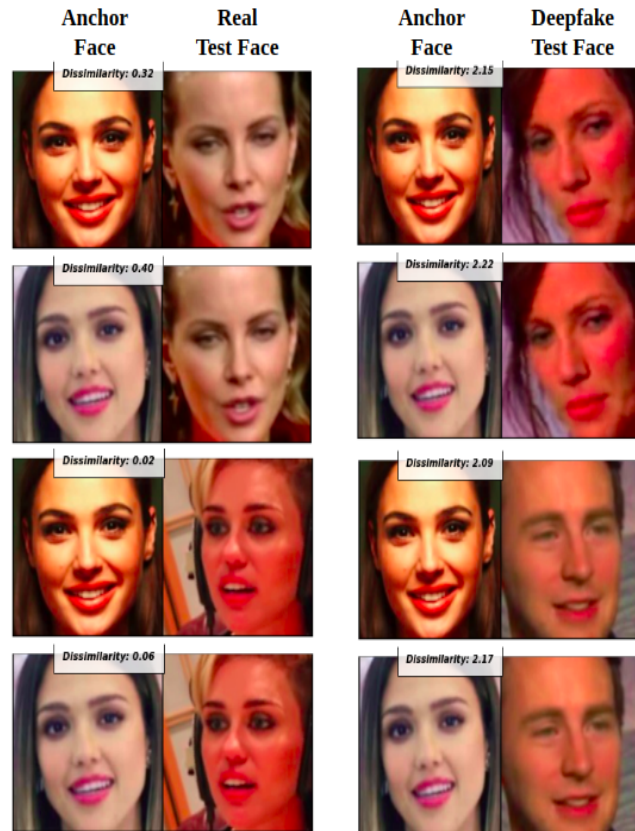


Figure 5.4: Sample similarity scores on a small experiment set

A sample experiment is conducted with four test faces randomly picked from the test set containing two real and two deepfake faces. From the AFDN, two best anchor faces are randomly selected to

Table 5.1: Summary of test results on the small experiment set

Anchor Face (AF)	Test Face (TF)	TF Type	Dissimilarity	Average Dissimilarity (Normalized)	Predicted label	Ground truth
AF1	TF1	Real	0.32	0.16	Real	Real
AF2			0.4			
AF1	TF2		0.02	0.01	Real	Real
AF2			0.06			
AF1	TF3	Deepfake	2.15	1	Deepfake	Deepfake
AF2			2.22			
AF1	TF2		2.09	0.97	Deepfake	Deepfake
AF2			2.17			

provide a visualization perspective of the testing stage. (Refer Fig 5.4). On the top-left the first anchor face (AF1) is selected from the anchor faces having maximum confidence score from the AFDN. AF1 is one of the input test pairs alongside a sample Real Test Face (TF1). A dissimilarity score of 0.32 is computed based on Euclidean distance between AF1 and TF1. Similarly, dissimilarity scores are computed for all test pairs in the figure. A higher dissimilarity score indicates lower similarity between input test pairs. The similarity scores are not normalized values between 0 and 1. The dissimilarity score is further normalized with the maximum for all test faces. Table 5.1 details the average dissimilarity scores per test face. The model predicts all test faces in the small experiment set.

5.3. Test Results using ASN method

The test set contains 27399 faces in total with 3076 real faces and 24323 deepfake faces. Each of the test image is paired along with one anchor real face extracted from the AFDN. This means that there are different similarity scores computed per test face from the test set. The distance measure used is Euclidean distance optimized with contrastive loss. The similarity (measure used is dissimilarity) scores per reference test face across the anchor faces are relatively normalized. The mean dissimilarity score is computed for each test face in the test set. The resultant mean dissimilarity score is thresholded to make a classification decision of belonging to real or deepfake class. The predicted labels are evaluated with the ground truth information of the test faces to determine the classification performance. As for the input pairs, since one of the pairs are obtained from the AFDN,

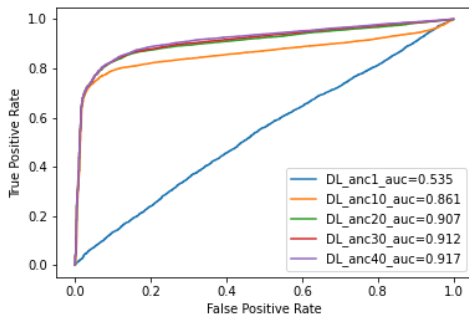


Figure 5.5: ROC for varying anchor faces

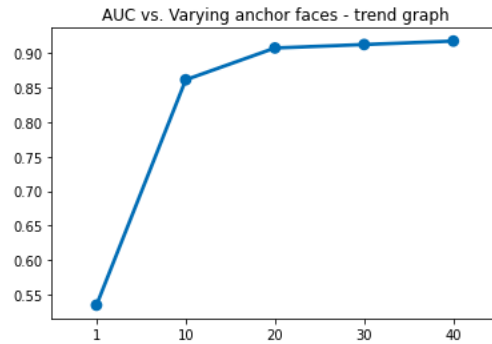


Figure 5.6: Varying AUC vs. Anchor faces - trend

identifying the optimal number of anchor face images required from the AFDN is crucial. Hence, different sets of experiments are performed on the test set by varying the number of anchor faces from the AFDN. The best performing model is evaluated based on highest the ROC-AUC metric. Fig

5.5 showcases the ROC plots for varying anchor faces selected from the AFDN.

From the ROC-AUC scores, the model with 40 anchor face (one from each celebrity class) performs the best with a reported AUC of 91.7%. When one anchor face (from all the faces) is chosen the model performs poorly (AUC 53%). Hence, when one anchor face is selected, the performance is near random thereby implying that the model does not predict the real and deepfake classes accuracy irrespective of varying the thresholds. The decision to choose the optimal number of anchor faces lies between 30 and 40 anchor faces. The reason being, considering computational time vs. performance trade-off, the AUC score with 30 anchor faces reports 91.2% as opposed to 40 anchor faces reporting 91.7% but is achieved with additional resources and time computation. Hence, the optimized value chosen in this work is 30 anchor faces from the AFDN.

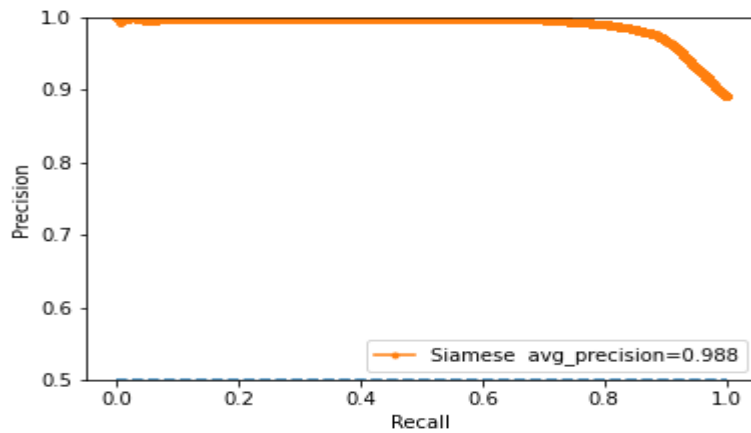


Figure 5.7: PR curve for 30 anchor faces

Table 5.2: AUC score of random sampling vs. fixed anchor faces for 30 celebrities class

Type	Reported AUC in %
Random Iteration #1	78.4
Random Iteration #2	81.8
Random Iteration #3	86.2
Random Iteration #4	76.9
Random Iteration #5	88.2
Anchor Faces from AFDN	91.2

Furthermore, the faces from the AFDN returns the face with the highest confidence per celebrity class thereby making it fixed every time. A comparison is made to determine the performance of the method by taking one random samples from 30 celebrity classes. The design choice of having 30 anchor faces with random sampling is made choosing the optimized value from Fig 5.5. Table 5.2 summarizes the results of the comparison of AUC score with 5 iterations of random sampling from 30 celebrity class with 30 anchor faces obtained from the maximum confidence of the AFDN.

The average AUC across five random iterations is 82.3% as opposed to selecting anchor faces based on a confidence score. Additionally, selecting a random face from each celebrity class shows variance in the reported AUC score thereby making the method unreliable to interpret the results. The attributed reason to lower AUC scores is also because the anchor faces randomly selected from each

celebrity class have a low confidence score reported by the HoG detector. Hence, it is conclusive to discard randomization as an option for choosing the one of test anchor pairs while testing the proposed model.

The PR curve for the ASN using the optimal anchor faces (30 faces) as one of the inputs to the test pairs is plotted in Fig 5.7. The ASN makes accurate true positive predictions that is reflected by the constant rate of Precision value being 1 for decreasing thresholds and increasing recall value scale. The attributed reasoning of having a constant high precision is due to the pairwise learning capability of the model. The model has learnt discriminative features from input test pairs passed thereby being able to distinguish the underlying real and deepfake class faces correctly. The reported average precision across all thresholds is 98.7% signifying a very high accuracy in predicting real and deepfake faces for different recall values.

5.4. Comparison of ASN method vs. Baseline Method

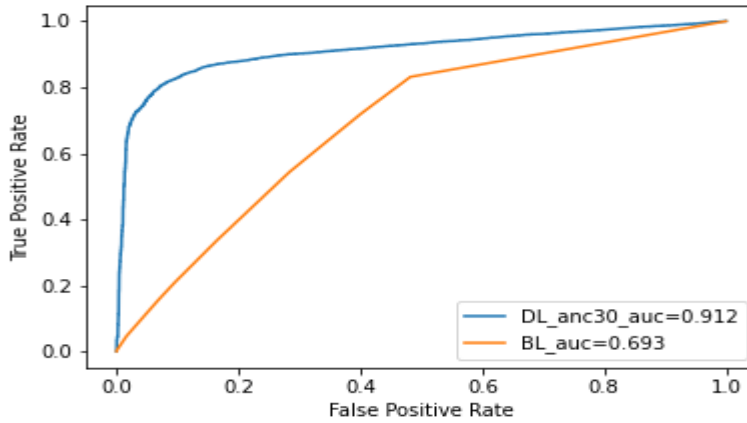


Figure 5.8: ROC for ASN method vs. Baseline method

In this section, a comparison on the test performance between the baseline method (using Eigen-faces) and Deep learning method using ASN is made. Both the methods are trained and tested on the same dataset. Fig 5.8 plots the ROC curves for the best results obtained between the Deep learning and the Baseline method. Comparing the AUC scores for both the models, the ASN method has a higher AUC curve (tending towards near perfect) signifying that the method shows higher separability between the real and deepfake class as opposed to the lower performance of AUC in the baseline method. In conclusion the ASN method performs better compared to the baseline method.

Conclusively, there are some interesting analysis that can be made from the results obtained. Firstly, from the results obtained, with lesser real faces in the test input pair, the ASN is able to predict real and deepfake classes more accurately than providing many real faces. This resolves the issue of having lesser real faces in real world setting as opposed to the increasing deepfakes. Furthermore, the use of similarity between faces helps to a model to learn more distinguishably the features present in a face. With regard to the generated feature representations, the ASN method generates higher discriminative feature representations. This is reflective of the high test performance.

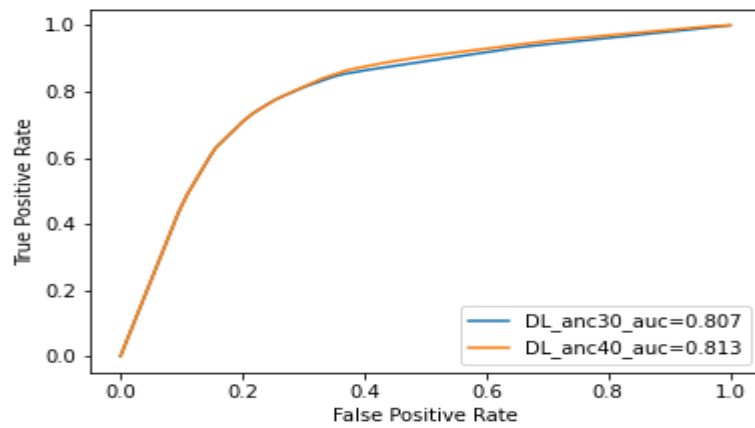


Figure 5.9: ROC curve for 30 and 40 anchor faces on independent hold-out set

5.5. Generalization Performance

The performance of the baseline and ASN method along with selected state-of-the-art deepfake detection methods are compared on an independent hold-out set to assess the generalization capability. The best performing model is evaluated based on the capability to predict real or deepfakes appropriately on unseen faces. The independent hold-out set as discussed in Section 4.2 contains real and respective deepfakes of 9 celebrities. This amounts to a total of 10086 pre-processed faces with 1301 real and 8785 deepfake faces. The comparative study on the generalization performance of the Baseline method (Eigenfaces) vs. ASN method is performed.

As for the ASN method, an additional choice of choosing the optimal number of anchor faces from the AFDN is also made. The reason behind this experimentation is to evaluate the consistency of the number of anchor faces needed between the testing phase and the generalization evaluation phase. In recollection from the test results obtained in Section 5.3, it is conclusive to use 30 anchor faces from the AFDN to achieve optimal model performance. In this experiment, the same 30 anchor faces in addition to 40 anchor faces from the AFDN is compared with ROC curves plotted on the independent hold-out dataset. Figure 5.9 contains the ROC curve performance of the ASN method with 30 and 40 anchor faces. Hence, with the final selection of 30 anchor faces on the generaliza-

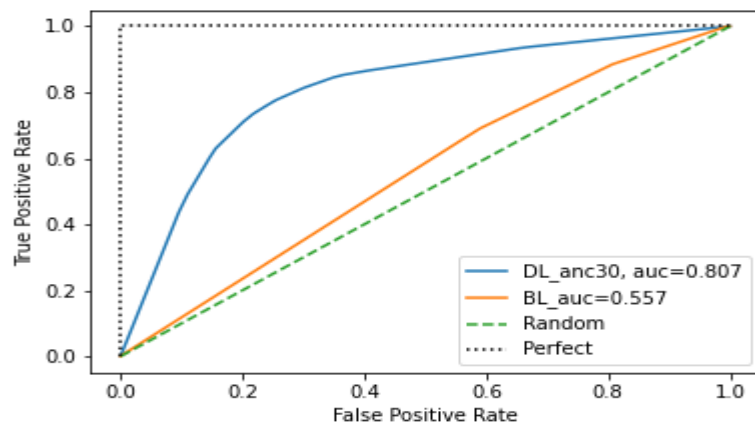


Figure 5.10: ROC performance for baseline and ASN method on independent hold-out set

tion evaluation, the baseline method is compared with the ASN method. (Refer Fig 5.10). From the figure, it can be inferred that the performance of the model with 30 and 40 anchor faces are considerably equivalent. Considering the computation efficiency, anchor faces of 30 is the optimal choice (the same optimal parameter from results obtained in the testing phase).

The baseline method shows poor performance on unseen faces by reporting an AUC of 55.7%. The resultant model at all possible thresholds fails to predict true positives for the real and deepfake class of faces accurately. However, the ASN method, reports an AUC of 80.7% with chosen 30 anchor faces from the AFDN. In comparison with the test performance of the ASN method (with an AUC of 91.2%) there is a borderline overfitting observed. However, the model is nevertheless able to distinguish real and deepfake classes of faces accurately with the current AUC. A contributonal reason to the overfitting in this scenario is owing to the possibility of the model's ability to make predictions with inclusion of mild noise in the data (faces). In comparison, with both the methods, the pairwise learning method using ASN method with AFDN predicts reflects higher performance despite having lesser real faces as an extrapolation to real-world settings.

5.5.1. Comparison with other deep learning models

Using SigNet and ASN

The generalization performance of the ASN method using pairwise learning is compared with two pre-existing forgery detection methods. The first model used for comparison is a modified SigNet model (discussed in Section 2.5). In this thesis, the SigNet is modified to learn face embeddings by retaining the ImageNet architecture of CNNs using the concept of Anchored Pairwise learning. The training set face input pairs are trained on the SigNet model. The modification is made in the testing stage where the AFDN is used for the anchoring of the network (similar to ASN ResNet-50, but with the architecture here being SigNet). The goal is to anchor one of the face input test pairs to convert the testing to a binary classification of real vs. deepfake faces. The optimized anchor faces from the AFDN is obtained from the ROC AUC results in Fig 5.11. The model optimally performs with

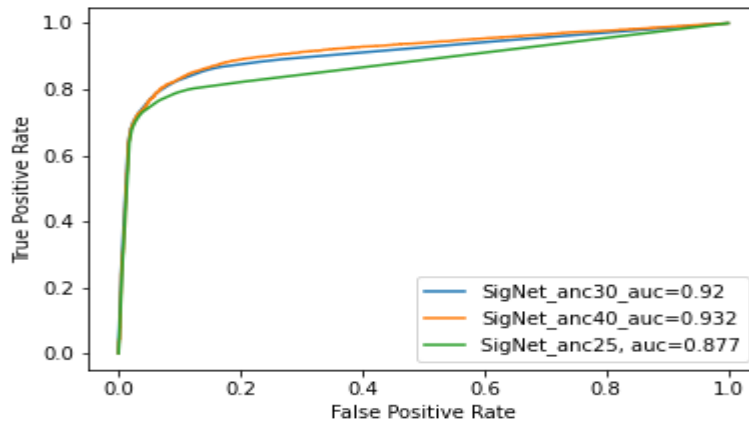


Figure 5.11: ROC performance of SigNet with varying anchor faces on independent hold-out set

30 anchor faces selected from the AFDN the modified SigNet model is tested on the Independent hold-out set to detect deepfakes thereby evaluating the generalization performance. The reported AUC performance of the SigNet model on the independent hold-out set is 92%. Interestingly, SigNet model using 25 anchor faces performs better than the deep learning method that uses ASN ResNet-50 model (with a reported AUC of 80.7%).

Using XceptionNet

Furthermore, to better understand the use of pairwise learning vs. traditional learning method, a deep learning model is used to evaluate the generalization performance. The model of choice used for this task is a pre-trained XceptionNet model (discussed in Section 2.4.3). The XceptionNet is an extension of the InceptionNet architecture with a replacement of inception modules with depth-wise separable convolutions. The inputs provided to this network are obtained from the pre-processing stage discussed in Section 4.2. In the evaluation stage, the Celeb-DF v2's independent hold-out set is used. Fig 5.12 showcases the ROC performance of the model. XceptionNet performs poorly on the independent hold-out set with an AUC score of $\sim 67\%$ in being able to detect deep-fakes from real faces.

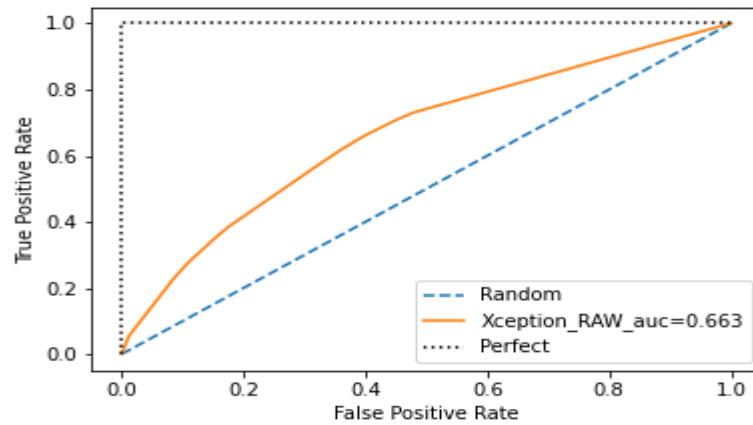


Figure 5.12: ROC performance of XceptionNet on independent hold-out set

Results Summary

In comparison to the traditional methods vs. pairwise learning methods used in this work, the following inferences can be drawn (Refer Fig 5.13):

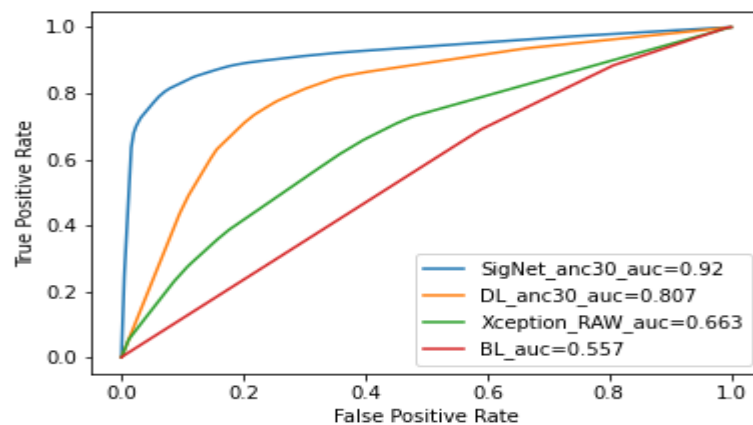


Figure 5.13: ROC performance of ASN method with other methods on independent hold-out set

- The SigNet model using Anchored Pairwise Learning performs the best with a AUC of 92%. The interesting aspect is, the model generates a 128-d embedding as opposed to the ASN

method with ResNet-50 model using 1000-d embedding. Comparing SigNet model with the proposed ASN method using ResNet-50 architecture the SigNet still outperforms with lower dimensional embedding. This proves that better model performance (with high AUC scores) can be achieved on lower dimensional embeddings of the features (as a feature representation). This way, computational intensity can be reduced by using low dimensional feature representations.

- The modification of SigNet using AFDN is attributed to a promising result of using only 30 anchor faces from the AFDN as opposed to using the entire anchor faces thereby addressing the key aspect of the research question of having lesser availability of real faces in real world setting as compared to the number of deepfakes. This proves that the use of ASN (the modification suggested in this work) performs well with the use of Anchored Pairwise Learning methods.
- Comparing traditional vs. pairwise learning methods - The traditional methods used in this work, namely, the Baseline method using Eigenfaces and the XceptionNet model perform poorly in comparison to the models that use pairwise learning technique (Anchored Pairwise Learning) being SigNet and ResNet50 both using ASN. This is conclusive that pairwise learning techniques showcase superior generalization performance with lesser real faces in the dataset.

6

Conclusion and Recommendations

The advent of deepfakes and its falsified applications has led to loss of trust in people. Furthermore, with robust and highly capable deep learning models, the availability of deepfakes of specific target individuals has increased. This has eventually created a huge imbalance of real vs. deepfake media of a specified targeted individual. In order to combat such issues, a deepfake detection method called ASN is proposed that leverages learning in pairs (pairwise learning), to qualitatively distinguish representative features between a real and a deepfake face. As opposed to traditional Siamese Networks (discussed in Section 2.3.4 of this report), the modification in the testing stage is made to the Siamese Network to utilize pairwise learning approach to generate discriminative features (face embeddings) with the help of a deep metric learning (contrastive loss). ASN uses an application network called AFDN to generate anchored faces to detect and classify images (real vs deepfake).

A baseline method using Eigenfaces is implemented as a holistic face recognition approach to evaluate the qualitative performance of detecting deepfakes against the ASN method. With the ASN method reporting AUC scores of 91.2% (on the test set) and 80.7% (on an independent hold-set), it is conclusive to prove that the approach using pairwise learning with anchored test input of limited real faces is a good performing model in the deepfake detection domain. Furthermore, the ASN method is evaluated with SigNet model (with a modification made using ASN) and a traditional learning model (XceptionNet) on the generalization set (independent hold-out set). Conclusively, methods using pairwise learning technique outperform traditional learning approaches.

6.1. Remarks

Regarding Anchored Pairwise Learning

The use of Anchored Pairwise Learning approach which is a modification made to Pairwise learning methods is a combination of ASN and AFDN to output anchored optimal faces to be used as one input pair to the testing stage. The application of this approach reports good detection performance achieved as opposed to traditional learning methods. This is attributed to the similarity based detection setting wherein the input pairs are learnt based on relative similarity between them. However, using ASN for extended multi-class problems will result in larger computation times owing to the increase in the number of anchor images as the number of classes increases. This might also lead to a lower confidence measure while making predictions.

Regarding AFDN

While the AFDN proves to be an efficient method to obtain desired anchor faces, the system heavily

relies on HoG descriptors. The choice to use HoG descriptors was made because they provide an efficient and robust mechanism to obtain localised features in sophisticated face recognition tasks. However, there is no existing research which validates the efficiency and performance of such descriptors in deepfake detection problems. Therefore, further research in this direction will certainly lead to improved performance of deepfake detection networks.

Regarding Network Architecture Optimization

With regard to the model architecture, the ResNet-50 performs relatively poorly on the independent hold-out set as compared to the test set. This may be a scenario of borderline overfitting. Further work to fine-tune the hyperparameters of model the model can be incorporated to maybe overcome this issue. Additionally, other variations of ResNet models maybe used to train the faces. Furthermore, ResNet-50 generates 1000-d feature embedding of input faces. An extensive study can be conducted on reducing the dimensionality of feature embeddings to provide equivalent or superior performance in comparison with the existing results. Additionally, an interesting result to note would be the superior performance of the SigNet model (which also uses pairwise learning) that generates lesser dimensional feature embedding to produce better AUC scores. Research can further be conducted to understand the relationship between optimized feature representations to attain better AUC scores.

6.2. Extended Applications

Within Forensics Domain

This thesis, proposes a deepfake detection method using limited real faces with the help of ASN and AFDN on images (faces). Further research can be explored into deepfake videos with the use of temporal information with ASN and AFDN (or an equivalent anchoring network). Additionally, the use of ASN can be incorporated in other Media Forensics domains, like fraud predictions on forgeries of signature (for verification tasks) or determining the genuinity of paintings (from counterfeit painting). The reason being, challenge of lesser number of real samples is also prevalent in the above use-cases. Therefore, the proposed work may significantly contribute in improving these domains.

Apply ASN in other Pairwise Networks

The SigNet model is originally used in signature forgery detection and verification tasks. With the incorporation of ASN in SigNet and when applied to deepfake image (face) detection, in this thesis the model provides superior results. Such extrapolations of applying ASN (and AFDN) can be made to other state-of-the-art deep learning pairwise models to assess the performance in specific use cases.

Cross-domain applicability

The ASN can be extended in application outside of the Forensics domain. In specific, ASN can be used in computer vision domains that require 3D object localization tasks that use depth perception. 3D object localisation is a crucial task for many robotic applications. It involves finding the full state of a known object in 3D space. For example, a robotic manipulator (robotic arm) would need to know the exact location and shape of an object - say a water bottle in order to grasp it and place it somewhere else. The perception network used here can compare know objects with the perceived object and make a prediction based on a similarity metric, a process similar to ASN that is proposed in this thesis. By doing so, the entire 3D state of an object can be accurately deciphered. Another domain of application can be in video face recognition from still-to-video face recognition systems where the number of reference faces are limited in availability [90]. In such a scenario, the use of ASN with AFDN may be helpful in being able to output the optimized face to be referenced with the faces captured in unconstrained videos.

A

Deepfake Creation

This section reviews some of the state-of-the-art deepfake generation methods from literature. Deepfakes are created primarily from five basic network models [14].

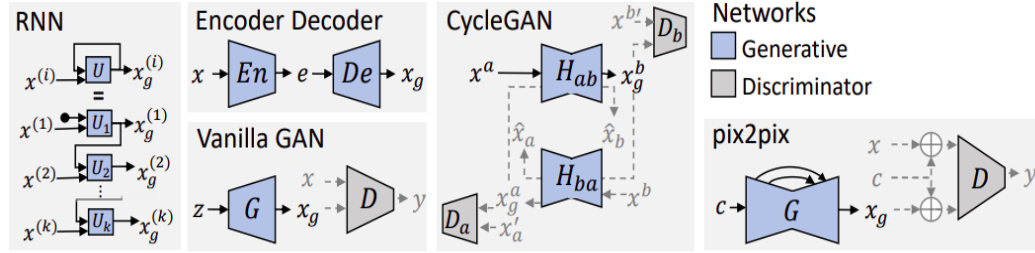


Figure A.1: deep learning network models used for deepfake generation [14]

1. Recurrent Neural Networks (RNN): RNNs handle sequential and variable length data. This is useful in deepfake video generation process using long short-term memory units (LSTM).
2. Encoder-Decoder Networks (autoencoders): The encoder-decoder network contains two networks, encoder and decoder. The first use of deepfake creation was attempted through FakeApp¹, which was developed by an anonymous Reddit user. The primary goal of FakeApp was to swap faces of users on celebrity faces and was viewed more as a creative game, using autoencoder-decoder networks. The autoencoder extracts the latent features of the face and reconstructs the face using a decoder. In order to apply face swapping between two images, two encoder-decoder pairs were used where each pair was trained on the image set, and the encoder's parameters were shared between two pairs. This facilitates the common encoder to detect similarity between source and destination images based on common facial feature orientations between faces (like eyes, nose, mouth etc.). The ideation and technique was adapted in the future works of DeepFaceLab², DFaker³ and DeepFake-tf⁴ which are computer vision aided softwares for plug-and-play deepfake generators Fig A.2 from [15] details the creation of face swapping images using autoencoder-decoder network.

¹<https://drive.google.com/file/d/1d9jCoPmPzjtBRtRyZh211GYzyNBDgiR/view>

²<https://github.com/iperov/DeepFaceLab.git>

³<https://github.com/dfaker/df.git>

⁴https://github.com/StromWine/DeepFake_tf.git

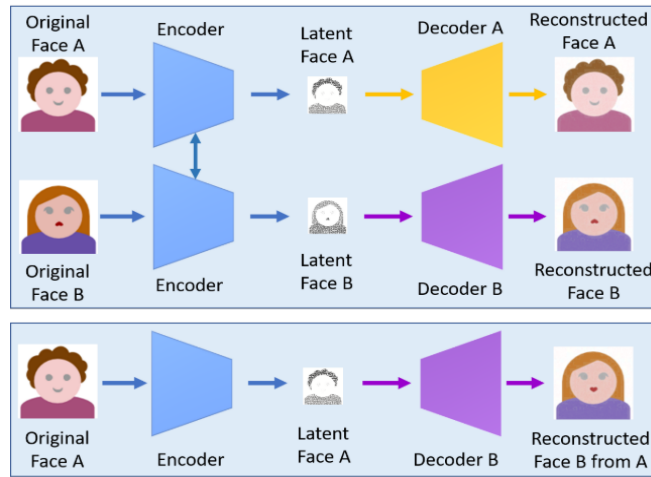


Figure A.2: deepfake creation model using two encoder-decoder pairs [15]

3. Generative Adversarial Networks (GAN): With the first introduction of GANs by Goodfellow et al. [38], the generation of deepfakes were made easier. A GAN consists of a generator (G) and discriminator (D) network where G creates synthetic samples of images and D learns to differentiate from real samples. An adversarial loss is used to train either of the networks. G learns to generate samples that are deviant from the original data distribution. Upon training, D is not used further and G generates fake data. The output of GANs are photo-realistic synthetic media. The initial GAN variant introduced for deepfake generation was an Image-to-Image translation technique - referred to as *pix2pix* [91]. This framework of this network learns the learn a loss function to train mapping apart from learning just the input to output mapping of image data. This generates improved higher resolution synthetic media with better fidelity. An improvement to *pix2pix* framework, CycleGAN was introduced [77]. CycleGAN uses two GAN networks which divide the losses into generator loss and discriminator loss. There is an additional cycle-consistency loss in the generator loss that ensures reasonable mapping between input and output images. The objective of CycleGAN is to handle better object transfiguration and style transfer in the output images thereby adding higher realism to image generation. The above discussed are some variations of GANs that were introduced for deepfake creation. There are more variants of deepfake GANs discussed in the survey paper in [92].

B

Data Generation

In this section, the deepfake image generation method and the dataset split is discussed. The three types of dataset used in this work are: Training set, Test set and Independent hold-out set. All three dataset comprises of two labels: Real and Deepfake faces. The Real faces are split based on the ID references representing videos of each celebrity corresponding to one specific ID. The IDs range from id00 - id61 (id14, id15 and id36 are not present in the dataset provided by the authors), amounting to 59 celebrities in total. The extracted frames from the videos are performed based on the ID references. id00 to id42 belong to the training set (real frames). Additionally, for the deepfakes, Tables B.1 and B.2 details the deepfakes and the associated source real video reference of which the frames are extracted.

As for the test set, id43 to id52 denote the celebrities used for testing. As for the generation of deepfakes for the test set, the synthesis of the source id reference overlaps with the training set. However, for the independent hold-out set, there is no overlap of the celebrities used in the training set used as reference to generate the deepfake. Therefore, the independent hold-out set is completely exclusive set of unseen images from the training set (for both real and deepfake media). The choice of the deepfakes and the rationale behind the choices are not revealed by the authors.

Basic Information of the Celeb-DF v2 dataset

The Celeb-DF dataset consists of 590 real and 5639 DeepFake videos. The average length of The real videos are chosen from YouTube videos, from interviews snippets given by 59 celebrities that cover a diversified distribution based on age, gender and ethnic backgrounds. 56.8% of the celebrities from real videos are men and 43.2% are women. Additionally, around 8.5% aged 60 years, 30.5% range between 50 and 60 years, 26.6% are between 41 and 49 years, 28.0% are between 31 and 39 years and 6.4% 30 years old. As for the ethnic background of the population, 5.1% are Asians, 6.8% are African-Americans and 88.1% are Caucasians.

ID reference	Deepfake ID reference	Dataset Type
id00	id01,id02,id03,id06,id09,id16,id17,id20,id21, id23,id26,id28	Training Set
id01	id00,id02,id03,id04,id06,id09,id16,id17,id20, id21,id23,id26,id30,id31,id35	Training Set
id02	id00,id01,id03,id04,id06,id09,id16,id17,id20, id21,id23,id26,id28,id30,id31,id35,id37	Training Set
id03	id00,id01,id02,id03,id04,id06,id09,id16,id17, id20,id21,id23,id26,id28,id30,id31,id35,id37	Training Set
id04	id00,id01,id02,id03,id06,id09,id20,id21,id23, id26,id28,id31,id35,id37	Training Set
id05	id19,id21,id25,id37,id38	Training Set
id06	id00,id01,id02,id03,id04,id09,id16,id17,id20, id21,id23,id26	Training Set
id07	id10,id11,id12,id13	Training Set
id08	id00,id01,id02,id03,id04,id05,id06,id07,id09	Training Set
id09	id00,id01,id02,id03,id04,id06,id16,id17,id20, id21,id23,id26,id30,id31,id35,id37	Training Set
id10	id07,id11,id12,id13	Training Set
id11	id07,id28,id30,id31,id35,id37	Training Set
id12	id07,id10,id31,id32,id33,id34,id35,id37,id38	Training Set
id13	id07,id11,id29,id30,id31,id32,id33,id34,id35,id37,id38	Training Set
id16	id00,id01,id02,id03,id06,id09,id17,id20,id21, id23,id26,id28,id30,id31,id35,id37	Training Set
id17	id00,id01,id02,id03,id06,id09,id16,id20,id21, id23,id26,id28,id30,id31,id35,id37	Training Set
id18	id24,id29,id32	Training Set
id19	id20,id21,id22,id23,id24,id25,id26,id27,id28	Training Set
id20	id00,id01,id04,id06,id09,id16,id17,id19,id21, id22,id23,id24,id25,id26,id27,id28,id29,id30	Training Set
id21	id00,id01,id02,id03,id04,id06,id09,id16,id17, id19,id20,id22,id23,id26,id27,id28,id29,id30, id31,id33,id34,id35,id37,id38	Training Set
id22	id19,id20,id21,id23,id24,id25,id26,id27,id28	Training Set
id23	id00,id01,id02,id03,id04,id06,id09,id16,id17, id19,id20,id21,id22,id24,id25,id26,id27,id28	Training Set
id24	id19,id20,id21,id22,id23,id25,id26,id27,id28	Training Set
id25	id19,id20,id21,id22,id23,id24,id26,id27,id28	Training Set
id26	id00,id01,id02,id03,id16,id17,id19,id20,id21, id22,id23,id24,id25,id26,id27,id28,id29,id30, id31,id32,id34,id35	Training Set
id27	id19,id20,id21,id22,id23,id24,id25,id26,id28	Training Set
id28	id00,id01,id02,id03,id04,id06,id09,id16,id17, id19,id20,id21,id22,id23,id24,id25,id26,id27, id29,id30,id31,id32,id33,id34,id35,id37,id38	Training Set
id29	id20,id21,id23,id26,id28,id30,id31,id32,id33, id34,id35,id37,id38	Training Set
id30	id01,id02,id03,id04,id09,id16,id17,id20,id21, id23,id26,id28,id29,id31,id32,id33,id34,id35	Training Set

Table B.1: Deepfake generation table 1

ID reference	Deepfake ID reference	Dataset Type
id31	id01,id02,id03,id04,id06,id09,id16,id17,id20, id21,id23,id26,id29,id30,id32,id34,id35,id37, id38	Training Set
id32	id20,id21,id23,id26,id29,id30,id31,id33,id34, id35,id37,id38	Training Set
id33	id20,id21,id23,id26,id28,id29,id30,id31,id32, id34,id35,id37,id38	Training Set
id34	id20,id21,id26,id29,id30,id31,id32,id33,id35, id37,id38	Training Set
id35	id01,id02,id03,id04,id06,id09,id16,id17,id20, id21,id23,id26,id29,id30,id31,id32,id33,id34, id37,id38	Training Set
id37	id01,id02,id03,id04,id06,id09,id16,id17,id20, id21,id23,id26,id28,id29,id30,id31,id32,id33, id34,id35	Training Set
id38	id20,id21,id23,id26,id28,id29,id31,id32,id33, id34,id35,id37	Training Set
id39	id40,id42,id43,id44,id45,id46,id47,id48	Training Set
id40	id39,id41,id42,id43,id44,id45,id46,id47, id48	Training Set
id41	id40,id42,id43,id44,id45,id46,id47,id48	Training Set
id42	id39,id40,id41,id43,id44,id45,id46,id47, id48	Training Set
id43	id39,id40,id41,id42,id44,id45,id46,id47, id48	Test Set
id44	id39,id40,id41,id42,id43,id45,id46,id47, id48	Test Set
id45	id39,id40,id41,id42,id43,id44,id46,id47, id48	Test Set
id46	id39,id40,id41,id42,id43,id44,id45,id47, id48	Test Set
id47	id39,id40,id41,id42,id43,id44,id45,id46, id48	Test Set
id48	id39,id40,id41,id42,id43,id44,id45,id46, id47	Test Set
id49	id48,id50,id51,id52,id53,id54,id55,id56, id57,id58	Test Set
id50	id49,id51,id52,id53,id54,id55,id56,id57	Test Set
id51	id49,id50,id53,id54,id55,id56,id57,id58	Test Set
id52	id49,id50,id53,id54,id55,id56,id57,id58	Test Set
id53	id49,id50,id51,id52,id54,id55,id56,id57, id58	Independent Hold-out
id54	id49,id50,id51,id52,id53,id55,id56,id57, id58	Independent Hold-out
id55	id49,id50,id51,id52,id53,id54,id56,id57, id58	Independent Hold-out
id56	id49,id50,id51,id52,id53,id54,id55,id57, id58	Independent Hold-out
id57	id49,id50,id51,id53,id54,id55,id56,id58	Independent Hold-out
id58	id49,id51,id52,id53,id54,id55,id56,id57	Independent Hold-out
id59	id50,id60,id61	Independent Hold-out
id60	id50,id59,id61	Independent Hold-out
id61	id50,id59,id60,id61	Independent Hold-out

Table B.2: Deepfake generation table 2

Bibliography

- [1] Carl Öhman. Introducing the pervert's dilemma: a contribution to the critique of deepfake pornography. *Ethics and Information Technology*, pages 1–8, 2019.
- [2] Philip Patterson, Lee Wilkins, and Chad Painter. *Media ethics: Issues and cases*. Rowman & Littlefield, 2018.
- [3] Peter K Doenges, Tolga K Capin, Fabio Lavagetto, Joern Ostermann, Igor S Pandzic, and Eric D Petajan. Mpeg-4: Audio/video and synthetic graphics/audio for mixed media. *Signal Processing: Image Communication*, 9(4):433–463, 1997.
- [4] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–11, 2019.
- [5] Anil K Jain and Stan Z Li. *Handbook of face recognition*, volume 1. Springer, 2011.
- [6] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [7] Ying Zhang, Lilei Zheng, and Vrizlynn LL Thing. Automated face swapping and its detection. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 15–19. IEEE, 2017.
- [8] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839. IEEE, 2017.
- [9] Chih-Chung Hsu, Chia-Yen Lee, and Yi-Xiu Zhuang. Learning to detect fake face images in the wild. In *2018 International Symposium on Computer, Consumer and Control (IS3C)*, pages 388–391. IEEE, 2018.
- [10] Yi-Xiu Zhuang and Chih-Chung Hsu. Detecting generated image based on a coupled network with two-step pairwise learning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3212–3216. IEEE, 2019.
- [11] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K Ghosh, Josep Lladós, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017.
- [12] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, United States, 2020.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [14] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *arXiv preprint arXiv:2004.11138*, 2020.
- [15] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection. *arXiv preprint arXiv:1909.11573*, 2019.
- [16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- [17] Cristian Vaccari and Andrew Chadwick. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media+ Society*, 6(1):2056305120903408, 2020.
- [18] Hany Farid. *Fake Photos*. MIT Press, 2019.
- [19] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library, 2018.
- [20] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7184–7193, 2019.
- [21] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 43–47, 2018.
- [22] Francesco Marra, Diego Gagnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 384–389. IEEE, 2018.
- [23] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: a survey. *Symmetry*, 11(9):1066, 2019.
- [24] Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *European Conference on Machine Learning*, pages 658–665. Springer, 2007.
- [25] Yuyang Wang, Roni Khardon, Dmitry Pechyony, and Rosie Jones. Generalization bounds for online learning algorithms with pairwise loss functions. In *Conference on Learning Theory*, pages 13–1, 2012.
- [26] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. Deep fake image detection based on pairwise learning. *Applied Sciences*, 10(1):370, 2020.
- [27] Vitalik Melnikov, Pritha Gupta, Bernd Frick, Daniel Kaimann, and Eyke Hüllermeier. Pairwise versus pointwise ranking: A case study. *Schedae Informaticae*, 25:73–83, 2016.
- [28] Daniel Sáez Trigueros, Li Meng, and Margaret Hartnett. Face recognition: from traditional to deep learning methods. *arXiv preprint arXiv:1811.00116*, 2018.
- [29] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3677–3685, 2017.

- [30] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [31] Yuwei Guo, Licheng Jiao, Shuang Wang, Shuo Wang, and Fang Liu. Fuzzy sparse autoencoder framework for single image per person face recognition. *IEEE transactions on cybernetics*, 48(8):2402–2415, 2017.
- [32] Ayush Tewari, Michael Zollhoefer, Florian Bernard, Pablo Garrido, Hyeonwoo Kim, Patrick Perez, and Christian Theobalt. High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [33] E Howcroft. How faking videos became easy and why that’s so scary, 2018.
- [34] Robert Chesney and Danielle Citron. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Aff.*, 98:147, 2019.
- [35] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [36] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. corr abs/1703.10593 (2017). *arXiv preprint arXiv:1703.10593*, 2017.
- [38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [39] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. IEEE, 2005.
- [41] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.
- [42] Marian Stewart Bartlett, Javier R Movellan, and Terrence J Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13(6):1450–1464, 2002.
- [43] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Fgr*, volume 2, page 215, 2002.
- [44] Tony Lindeberg. Scale invariant feature transform. 2012.
- [45] Divyarajsinh N Parmar and Brijesh B Mehta. Face recognition methods & applications. *arXiv preprint arXiv:1403.0485*, 2014.

- [46] Ana Cris Murillo, José Jesús Guerrero, and C Sagues. Surf features for efficient robot localization with omnidirectional images. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3901–3907. IEEE, 2007.
- [47] Jie Xu, Hua-wen Chang, Shuo Yang, and Minghui Wang. Fast feature-based video stabilization without accumulative global motion estimation. *IEEE Transactions on Consumer Electronics*, 58(3):993–999, 2012.
- [48] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.
- [49] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.
- [50] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. On the generalization of gan image forensics. In *Chinese Conference on Biometric Recognition*, pages 134–141. Springer, 2019.
- [51] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [52] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.
- [53] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [54] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.
- [55] Pavel Korshunov and Sébastien Marcel. Speaker inconsistency detection in tampered video. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2375–2379. IEEE, 2018.
- [56] Aparna Bharati, Richa Singh, Mayank Vatsa, and Kevin W Bowyer. Detecting facial retouching using supervised deep learning. *IEEE Transactions on Information Forensics and Security*, 11(9):1903–1913, 2016.
- [57] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2008.
- [58] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92. IEEE, 2019.
- [59] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–45, 2019.
- [60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

- [61] Miroslav Goljan and Jessica Fridrich. Cfa-aware features for steganalysis of color images. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090V. International Society for Optics and Photonics, 2015.
- [62] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pages 159–164, 2017.
- [63] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016.
- [64] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.
- [65] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2016.
- [66] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
- [67] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.
- [68] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
- [69] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.
- [70] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [71] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019.
- [72] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.
- [73] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.
- [74] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

- [75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [76] Dmitry Nikitko. Stylegan encoder for official tensorflow implementation, 2019.
- [77] Yongyi Lu, Yu-Wing Tai, and Chi-Keung Tang. Conditional cyclegan for attribute guided face image generation. *arXiv preprint arXiv:1705.09966*, 2017.
- [78] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.
- [79] Conrad Sanderson. The vidtimit database. Technical report, IDIAP, 2002.
- [80] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003.
- [81] Nick Dufour and A Gully. Contributing data to deepfake detection research. *Google AI Blog*, 2019.
- [82] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.
- [83] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.
- [84] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587, 1991.
- [85] Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2010.
- [86] Michael A Nielsen. *Neural networks and deep learning*, volume 2018. Determination press San Francisco, CA, 2015.
- [87] Nikolas Adaloglou. Intuitive explanation of skip connections in deep learning. <https://theaisummer.com/>, 2020. URL <https://theaisummer.com/skip-connections/>.
- [88] Martin JJ Scott, Mahesan Niranjan, and Richard W Prager. Realisable classifiers: Improving operating performance on variable cost problems. In *BMVC*, pages 1–10, 1998.
- [89] Marijeta Slavković and Dubravka Jevtić. Face recognition using eigenface approach. *Serbian Journal of electrical engineering*, 9(1):121–130, 2012.
- [90] Fania Mokhayeri and Eric Granger. Video face recognition using siamese networks with block-sparsity matching. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(2):133–144, 2019.
- [91] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [92] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *arXiv preprint arXiv:2001.00179*, 2020.