Fake Face Detection in Social Media Videos using Deep Learning: A Comprehensive Analysis and Robust Framework

A thesis

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

Mehrab Rabbi	**160104104**
Minhajul Islam	**170104001**
Yusha Abdullah	**180104012**
Faysal Mahmud	**180204075**

Supervised by

Ms. Tahsin Aziz

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

May 2023

CANDIDATES' DECLARATION

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

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

Mehrab Rabbi
160104104

Minhajul Islam
170104001

Yusha Abdullah
180104012

Faysal Mahmud
180204075

CERTIFICATION

This thesis titled, **"Fake Face Detection in Social Media Videos using Deep Learning: A Comprehensive Analysis and Robust Framework"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in May 2023.

Group Members:

Mehrab Rabbi	**160104104**
Minhajul Islam	**170104001**
Yusha Abdullah	**180104012**
Faysal Mahmud	**180204075**

Ms. Tahsin Aziz
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

We would like to express gratitude to Ms. Tahsin Aziz, an assistant professor in the department of Computer Science and Engineering at Ahsanullah University of Science and Technology, for her patient supervision, passionate encouragement, and helpful evaluation of our thesis work. Additionally, we appreciate the assistance and talks provided by Professor Dr. Md. Shahriar Mahbub, the respected Head of the Computer Science and Engineering department at Ahsanullah University of Science and Technology, as well as all of our respected teachers, lab assistants, and friends. Last but not the least, we want to thank our parents for their unwavering support and inspiration throughout our education.

Dhaka
May 2023

Mehrab Rabbi

Minhajul Islam

Yusha Abdullah

Faysal Mahmud

ABSTRACT

Deepfake technology is widely used, which has led to serious worries about the authenticity of digital media, making the need for trustworthy deepfake face recognition techniques more urgent than ever. This study employs a resource-effective and transparent cost-sensitive deep learning method to effectively detect deepfake faces in videos. In order to create a reliable deepfake detection system, four pre-trained Convolutional Neural Network (CNN) models: XceptionNet, InceptionResNetV2, EfficientNetV2S, and EfficientNetV2M were used. FaceForensics++ and CelebDf-V2 as benchmark datasets were used to assess the performance of our method. To efficiently process video data, key frame extraction was used as a feature extraction technique. Our main contribution is to show the model's adaptability and effectiveness in correctly identifying deepfake faces in videos. Furthermore, a cost-sensitive neural network method was applied to solve the dataset imbalance issue that arises frequently in deepfake detection. The XceptionNet model on the CelebDf-V2 dataset gave the proposed methodology a 98% accuracy, which was the highest possible whereas, the InceptionResNetV2 model, achieves an accuracy of 94% on the FaceForensics++ dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

As a result of the increase in processing power, deep learning techniques have gotten so powerful that it is now quite simple to create "deepfakes," or artificial videos that impersonate humans. It's not hard to see scenarios in which these plausible face swap deepfakes are employed to carry out various forms of revenge, instigate political crises, demand money, or blackmail individuals. An automated technique for spotting replacement and replication deepfakes is presented in this article. We are deploying it to combat artificial intelligence. In our method, frame-level characteristics are extracted which is called key-frame extraction from videos, and these features are then used to train a baseline Convolutional Neural Network (CNN) and some pre-trained models such as XceptionNet, EfficientNetV2S, EfficientNetV2M, and InceptionResNetV2 to classify whether the video is deepfake or real video. In order to simulate real-world situations and enhance the model's performance on real-world data, we test our approach on a significant number of datasets including FaceForensic++ [1], and Celeb-DF [2]. We also show how our system may generate competitive results using a very simple and dependable manner.

1.2 Deepfake

The terminology "deepfake" is derived from the underlying artificial intelligence (AI) technology known as "deep learning." To create bogus media that seems realistic, deep learning algorithms are utilized to swap faces in videos and digital content. These algorithms train themselves in how to solve issues when given large amounts of data. Although there are various ways to make deepfakes, the most popular one makes use of face-swapping autoencoders in deep neural networks with autoencoders. A series of video clips of the person you

want to insert in the target must come first, followed by a target video to serve as the foundation for the deepfake. Generative Adversarial Networks (GANs), another type of machine learning, are incorporated into the process. GANs identify and fix any deepfake problems over the course of several rounds, making it more challenging for deepfake detectors to identify them. In order to "learn" how to create fresh instances that closely resemble the real thing, GANs are also frequently utilized as a popular technique for the production of deepfakes.

1.3 How Deepfake Works?

Autoencoders or generative adversarial networks are the two ANN kinds that deep fake software developers most frequently employ (GANs). In order to duplicate desired data sets, autoencoders learn to replicate the vast amounts of data that are fed to them, primary images of faces and expressions. However, they are rarely exact replicas. On the other hand, GANs have a more intelligent system that consists of a generator and a discriminator. The first copies learned data into deepfakes that must then deceive the second. The discriminator assesses the efficacy of the generator's output by contrasting it with actual photographs. Of course, the most convincing deep fakes completely replicate human behavior. So, how is this technology used to create deep fakes? To accurately alter facial characteristics and expressions or layer one face over another, the algorithms powering apps like Reface and DeepFaceLab continuously learn from the data passing through them. The program essentially functions as a face-manipulating video editor. You may age someone up or down, edit oneself into videos, and more with several programs, some of which are more sophisticated than others. However, there are still issues with the technology. Although creating deep fakes may be more complex than creating fake live videos, they can still be easily distinguished from the real thing.

1.4 Motivation

Deepfake video detection tasks have a very limited dataset availability and are often imbalanced. A pipeline to solve the problem of data imbalance and extracting key frames from videos has been proposed in this study. The main idea behind key frame extraction is to reduce the volume of data and computation time in a video while maintaining its essential content and context, whereas cost-sensitive methods are used in situations where the goal is to give more importance or bias toward classes that may have a higher cost associated with misclassification. There are several approaches that use key frame extraction [3] and cost-sensitive neural networks [4] - [5] for various detection tasks. In this study, various

pre-trained CNN-based models were assessed to identify deepfake faces in videos.

There is a significant risk involved as we do not understand how these sophisticated neural networks generate their predictions. Users undoubtedly rely on the models as they hardly try to understand what is happening in the model's backend. The ability to understand how a model produces any kind of decision would be great. To analyze the weights and inner workings of a neural network as well as to explain the output of the pre-trained models used in the proposed methodology, we used a variety of Gradient-based Explainable AI tools, including SmoothGrad, GradCAM, GradCAM++, and Faster Score-CAM. The models in this study were also trained using two separate imbalanced datasets.

1.5 Contribution

The main goal of this thesis is to investigate DeepFake video detection and to progress the field by creating a deep learning-based methodology. The urgent need to stop the possible abuse of such technology and the growing worry over the modification of video content using deep learning techniques are the driving forces behind this research. The main objective of this study is an extensive examination of the most recent developments in deep learning techniques and architectures, with a focus on how well they may be used to identify DeepFake videos. To sum up, we have added the following contribution to this research:

- Conduct a comprehensive review of the literature on DeepFake videos and their detection techniques.
- Collect and preprocess a diverse dataset of DeepFake videos and their corresponding authentic videos.
- The key frame extraction approach was employed to reduce the amount of data and processing time in a video while keeping its important content.
- For an imbalanced dataset, we implemented the idea of a cost-sensitive neural network.
- The prediction of the trained models is explained by the Explainable AI techniques (XAI) including, SmoothGrad, GradCam, GradCam++, and Faster Score-Cam.
- Explore and implement state-of-the-art deep learning architectures, including some pre-trained models such as XceptionNet, EfficientNetV2S, EfficientNetV2M, and InceptionResNetV2 for DeepFake detection.

- Compare the performance of the implemented deep learning architectures on the collected dataset using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score.
- Perform a real-world case study to demonstrate the effectiveness of the developed DeepFake detection model on a variety of DeepFake videos.

1.6 Thesis Structure

Our thesis book consists of seven chapters based on our research work. Following, we will briefly discuss the basis of the chapters.

- **Chapter 1: Introduction**
  The thesis book's first chapter is this one. We will briefly outline the purpose, our aim, and the contribution of our thesis in this chapter.
- **Chapter 2: Literature Review**
  We will discuss some related work that has already been completed along with its methods, benefits, and drawbacks.
- **Chapter 3: Background Study**
  This chapter contains all the background information needed to understand our thesis topic. We'll talk about deep learning strategies and image-processing approaches.
- **Chapter 4: Dataset Acquisition**
  In this chapter, we will discuss the dataset and their frequencies along with face forgery types.
- **Chapter 5: Proposed Methodology**
  This chapter has covered our suggested approach for detecting deepfake faces from films using deep learning models.
- **Chapter 6: Experimental Result**
  We will describe the experimental findings in this chapter and explain the performance metrics for our proposed approach to performance evaluation.
- **Chapter 7: Result Analysis & Comparison:**
  In this chapter, we will compare the outcomes of previous research and our proposed approach and discuss which strategy is more effective.
- **Chapter 8: Conclusion**
  Finally, the concluding chapter will bring our study to a close. We will talk about the shortcomings of our work and potential future areas for development in this section.

1.7 Summary

This chapter provides a brief overview of DeepFake's functionality. Different DeepFake generating procedures and detection methods have been covered. We briefly outlined the purpose of our work and the driving force behind it. Finally, a brief description of our thesis' contribution and its structure was given.

Chapter 2

Literature Review on Deepfake Video Detection

2.1 Overview

Many different kinds of study have been done in the area of deepfake detection in recent years. Deepfake detection attracted researchers from a variety of fields, including computer vision, image processing, and deep learning. In order to identify the most beneficial and cutting-edge techniques that have been applied in recent articles, we analysed a few of the current studies. On various study pieces, we worked. We shall go into great detail about these papers and their operational methods in this chapter since they are relevant to our line of work.

2.2 Reviews of the related papers

In recent advancements in deepfake video detection, Lee and Kim [6] have introduced an innovative approach that focuses on extracting the rate of change between adjacent frames to discern whether a video has been manipulated or not. Leveraging datasets like Faceforensics++ [7] and DFDC [8], they meticulously extracted three hundred frames from each video, utilizing a deep neural network architecture that achieved an impressive accuracy of 97%, outperforming existing methods. However, it's worth noting that their method's effectiveness can be compromised when faced with manipulated frame images, which has led them to address this issue in ongoing research. For Deepfake Video Detection Xu et al. [9] proposed a method for constructing texture features and processing them with a feature selection method. This discriminant feature vector is then passed on to SVM for classification. DeepFake-TIMIT [10], Celeb-DF [11], FaceForensics++ [7], and DFDC [8] datasets

were used in this experiment. They have achieved the highest accuracy of 91.2% on C40 quality videos of the FaceForensics++ dataset. Though the texture details of the deepfake videos are insufficient, they think this method can work well if new deepfake videos with defective texture features are released in the future. Furthermore, Two-dimensional global discrete Cosine transforms (2D-GDCT) are used in the method that Kohli et al. [12] suggested to extract faces from a target video and convert them into the frequency domain. They used FaceForensics++ [7] and CelebDf-V2 [11] datasets in their study. Then, to identify fake facial images, a 3-layered frequency convolutional neural network (CNN) is used. At first, they split the target video into frames and further converted it into frequency, and then a convolutional neural network was trained to learn the facial image's frequency features. The highest accuracy from their proposed model is 86.08% and for average and maximum pooling technique is 85.24%. Their proposed model performed better in terms of detecting pristine faces. But in total accuracy, XceptionNet performed better. Lastly, Kim et al. [13] introduced two models on the FaceForensics++ dataset [7], with the first one being trained using XceptionNet and the second model requiring the weights of the first model for training. They employed Feature-based Representation Learning to identify common features across different deepfake videos, which they referred to as feature transfer learning. Notably, XceptionNet served as the backbone for their model, known as FReTAL, and it yielded an impressive accuracy of 86.97% in deepfake video detection. These pioneering approaches collectively represent significant strides in the ongoing battle against the proliferation of deepfake videos.

2.3 Summary

In this chapter, we looked over a few research articles and spoke about how they were put together. This literature review demonstrates that there have been numerous research publications published on the topic of deepfake video detection. Traditional classifiers were used by some of the researchers, while deep learning techniques were used by others. Using conventional methods, some works produced meaningful results while others did not. However, after reviewing these works, we may say that deep learning outperforms conventional classifiers due to their utilization of memory in the network and learning mechanism.

Chapter 3

Background Study

3.1 Overview

DeepFake videos are artificially produced videos made with deep learning algorithms that are intended to trick viewers by showing fake incidents or circumstances. They are becoming more and more prevalent and seriously endanger the security, privacy, and safety of the general people. Consequently, there is a rising need to provide efficient methods for identifying DeepFake movies. Convolutional neural networks (CNNs), in particular, have demonstrated encouraging outcomes in the detection of DeepFake videos. CNNs can be trained on datasets of real and DeepFake videos to learn the discriminative features that separate them. CNNs are capable of learning complicated features from photos and videos.

Overall, the background study emphasizes the significance of creating efficient methods for identifying DeepFake films and explores the function of CNNs in this process. Along with reviewing the shortcomings of current CNN designs, it also emphasizes recent developments in creating compact and effective architectures for DeepFake video detection.

3.2 CNN

Convolutional Neural Networks (CNNs) [14] are a class of deep neural networks that are highly adapted to the analysis of two-dimensional data, such as audio spectrograms and medical images. In a variety of tasks, including image classification, object detection, semantic segmentation, and others, CNNs have produced state-of-the-art results. A CNN is made up of several layers that, at a high level, convert an input image into an output prediction. An image is sent as a two-dimensional array of pixel values to the input layer, which is the first layer. The following layers are frequently convolutional layers that apply a set of

adjustable filters to the input. These filters, which are typically small (e.g., 3×3 or 5×5), are used for generating feature maps by applying them to the entire input. Through a technique known as backpropagation, the network itself learns the filters by adjusting the weights of the filters to reduce the error between the predicted output and the actual output.

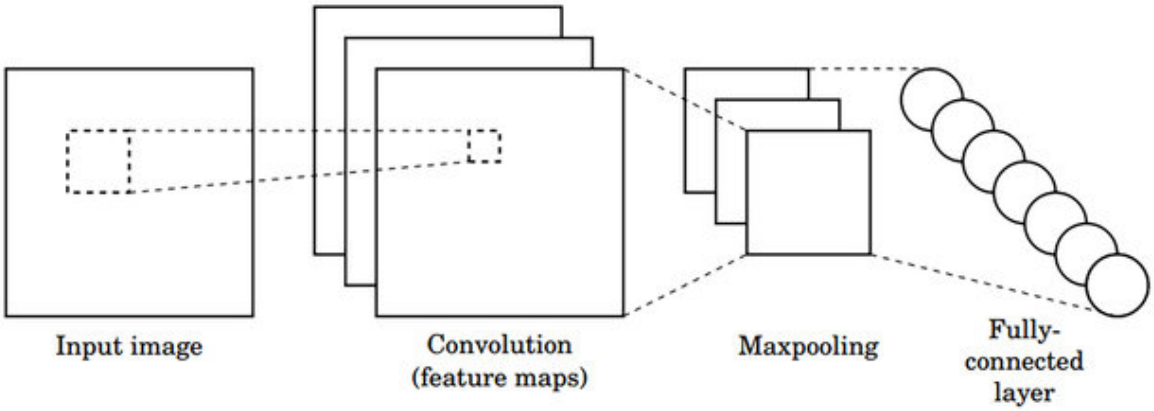


Figure 3.1: CNN Architecture

Following the convolutional layers, pooling layers are frequently used to reduce the output feature maps' dimensionality by reducing collections of related features into a single value. Lowering the number of parameters in the network, can help with training and prevent overfitting. One or more fully connected layers are then applied to the output of the convolutional and pooling layers, mapping the learned features to the final prediction. This last layer can generate a probability distribution over a group of classes, such as "cat", "dog", "bird" and so on, in image classification tasks. The ability of CNNs to learn hierarchical representations of an input image is one of their key features. Early convolutional layers pick up on basic properties like edges and corners, whereas subsequent layers pick up on more complex features like object parts and textures. Using this hierarchical representation, the network can recognize objects in the image regardless of where they are located, how they are oriented, or how big they are. For better performance and increased interpretability, CNNs have recently been integrated with other methodologies like transfer learning and attention processes.

3.3 XceptionNet

In 2016, Google researchers created the convolutional neural network (CNN) architecture known as XceptionNet [15]. Since the network is an extreme version of Google's Inception architecture, the name "Xception" was chosen to describe it. The main idea of XceptionNet is to use "depthwise separable convolutions" in place of the conventional Inception modules. Every channel of the input feature map is applied to the same set of filters in a typical

convolutional layer. A depthwise separable convolution, in contrast, combines the outputs of the depthwise convolution into a smaller set of channels using a 1x1 convolution. It is composed of two distinct layers: a depthwise convolution that applies a different set of filters to each input channel, and a pointwise convolution.

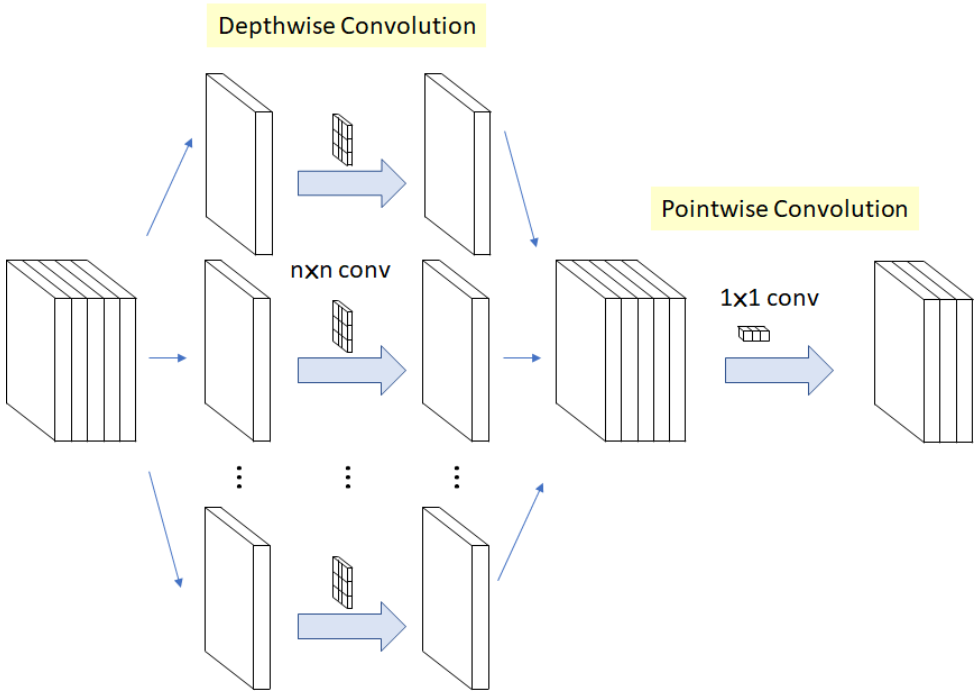


Figure 3.2: XceptionNet Architecture

The benefit of this method is that it drastically decreases the network's parameter number while still enabling the network to learn complex features. This increases the network's computational efficiency and makes the training simpler, especially on devices with limited resources. At the time of its release, XceptionNet had attained cutting-edge performance on a number of benchmarks for computer vision, including COCO object detection and ImageNet classification. Since then, it has gained popularity as a design for fine-tuning and transfer learning in computer vision research and applications. The increased depth and complexity of XceptionNet may make it more vulnerable to random initialization and hyperparameter settings than other architectures, which is a potential drawback. Nevertheless, this can be reduced by carefully choosing the initialization and regularization methods during training. Overall, the area of computer vision has benefited greatly from the use of XceptionNet, a strong and effective CNN architecture. Because of its creative application of depthwise separable convolutions, more research is being done on developing scalable and effective neural network designs.

3.4 EfficientNet

Tan and Le proposed the EfficientNet which is a family of convolutional neural network (CNN) architectures in 2019 [16]. In order to obtain high accuracy while using a minimal number of computational resources, the main concept behind EfficientNet is to apply a compound scaling mechanism that balances the depth, width, and resolution of the network. There are multiple models in the EfficientNet family, ranging in size from the smallest, EfficientNetB0, to the largest, EfficientNetB7. With a series of convolutional layers and bottleneck layers that lower the network's computational complexity and parameter count, each model is constructed using a similar architecture. Each model, however, is scaled individually, with differences in depth, input resolution, and the number of layers.
To achieve the optimal balance between accuracy and efficiency, EfficientNet uses a compound scaling strategy that entails scaling the network's depth, width, and resolution all at once. In order to improve the detail and richness of the feature representation, the input image resolution is increased, the depth of the network is increased by adding more layers, the width of the network is expanded by increasing the number of channels in each layer, and the number of channels in each layer is increased.

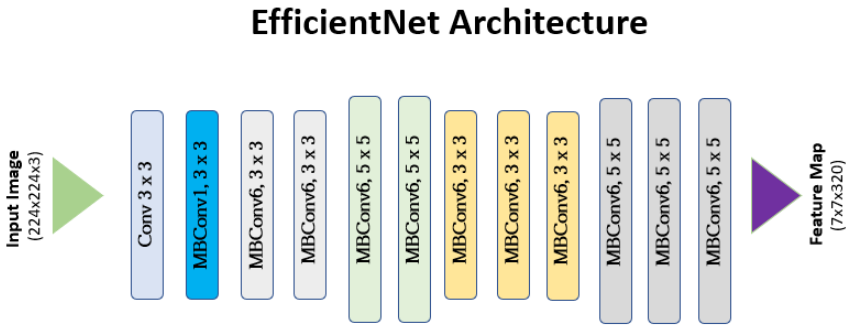


Figure 3.3: EfficientNet Architecture

In comparison to other state-of-the-art CNN architectures, EfficientNet has produced cutting-edge results on a number of computer vision benchmarks, such as the COCO object detection challenge and the ImageNet classification challenge. In particular, EfficientNetB0 outperforms ResNet50 on the ImageNet classification task despite using fewer parameters. The main benefit of EfficientNet is its efficiency, as the compound scaling mechanism permits great accuracy while reducing the network's computational expense and memory utilization. This makes it particularly beneficial for training on hardware with limited resources as well as applications that demand real-time or low-latency processing. Overall, the technology of computer vision has been greatly impacted by EfficientNet, a strong and effective CNN architecture. Its application of the compound scaling method has stimulated further

study into neural network architectures that are more effective and efficient, and many other CNN architectures have adopted its ideas.

3.4.1 EfficientNetV2S

EfficientNetV2S is a highly efficient convolutional neural network (CNN) architecture that aims to be both parameter and FLOP (floating-point operation) efficient. The EfficientNetV2 architecture, created by Google AI [17] in 2021, has been scaled down and made faster. Comprising MBConv and Fused-MBConv blocks, it achieves state-of-the-art results on tasks like ImageNet classification with just 22.2 million parameters and 3.9 billion FLOPs.It has been demonstrated that EfficientNetV2S outperforms other state-of-the-art models in terms of performance on a range of image classification tasks. Especially in situations where efficiency is a key requirement, such as mobile devices and embedded systems.

3.4.2 EfficientNetV2M

EfficientNetV2M is a highly efficient and powerful convolutional neural network(CNN) architecture designed by Google AI [17] for image classification. It features inverted residual blocks, mobile inverted bottleneck convolutions (MBConv blocks), fused mobile inverted bottleneck convolutions (Fused-MBConv blocks), and squeeze-and-excitation (SE) modules to optimize both accuracy and computational efficiency. The architecture is organized into stages with varying numbers of blocks, allowing for flexibility in model complexity. A fully connected layer and Global average pooling produce final predictions. Pre-trained on ImageNet, EfficientNetV2M is ideal for tasks like image recognition, object detection, and image segmentation, making it suitable for transfer learning.

3.4.3 InceptionResNetV2

InceptionResNetV2 is a convolutional neural network that fuses the strengths of Inception and ResNet architectures [18]. It combines dimension-reduction blocks, skips connections to address gradient vanishing problems, and Inception blocks to capture multi-scale characteristics. Global average pooling, fully connected layers, and a softmax output are used after feature extraction. It is well-known for its great accuracy and processing efficiency, making it a preferred option for many computer vision jobs. While the number of layers may vary, the architecture it normally employs is deep, demonstrating the versatility and efficiency of the system in complicated picture processing.

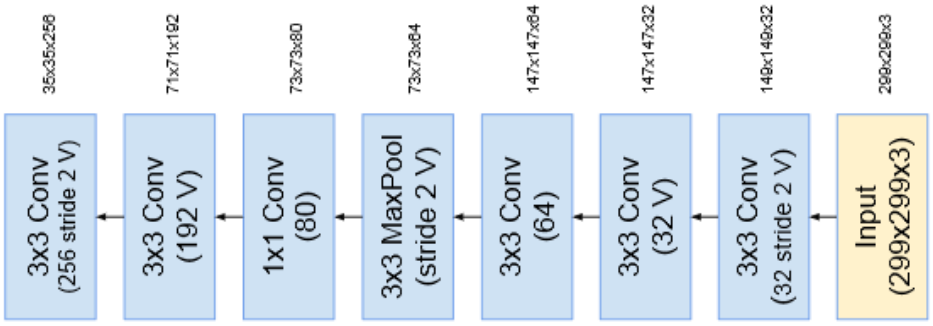


Figure 3.4: InceptionResNetV2 Architecture

3.4.4 Explainable Artificial Intelligence

Many Explainable Artificial Intelligence algorithms have already been developed for image categorization. Skin cancer identification and classification, however, have gotten less attention. Using SmoothGrad [19] and Faster Score-CAM [20], the interpretability of the proposed models was enhanced. In addition to reducing visual noise, SmoothGrad is compatible with a variety of sensitivity map techniques. After calculating the gradient for several samples surrounding the given sample and incorporating data from Gaussian noise, the average is determined. A quicker version of Score-CAM is referred to as Faster Score-CAM. In comparison to Score-CAM, it is more effective and clarifies the model better. Faster Score-CAM employs the channels with substantial fluctuations as mask pictures since multiple picture channels greatly influence how the final heat map is created.

3.5 Performance Metrics

To assess the effectiveness of the model, many indicators, frequently referred to as performance metrics, are employed. Performance measures are employed to assess the precision and potency of trained models. A model's effectiveness is assessed using its f1 score, recall, and precision.

- **Accuracy:** This is the most basic performance metric that is used to measure the proportion of correctly classified examples. It is calculated as the ratio of the number

of correct predictions to the total number of predictions. The formula we applied to calculate the Accuracy:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3.1}$$

- **Precision:**
  This metric measures the proportion of true positives out of all the positive predictions. It is useful when the cost of a false positive is high.
  The formula we applied to calculate the Precision:

$$Precision = \frac{TP}{(TP + FP)} \tag{3.2}$$

- **Recal:**
  This metric measures the proportion of true positives out of all the actual positive examples. It is useful when the cost of a false negative is high.
  The formula we applied to calculate the Recall:

$$Recall = \frac{TP}{(TP + FN)} \tag{3.3}$$

- **F1 Score:**
  This metric is the harmonic mean of precision and recall and is used when both precision and recall are equally important.
  The formula we applied to calculate the f1 score:

$$F1Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{3.4}$$

Where,
True Positive (TP) = accurate positive classification
True Negative (TN) = accurate negative classification
False Positive (FP) = Positive class incorrectly classified
False Negative (FN) = Negative class incorrectly classified

3.6 Summary

This chapter provides a brief overview of each issue that is relevant to our thesis work, along with information on how they operate, their benefits and drawbacks, etc. We first attempted

to explain the fundamentals of image processing before moving on to a discussion of the fundamental CNN architecture and its various hyper-parameters.

Chapter 4

Proposed Methodology

4.1 Overview

The proposed method tries to create a reliable and effective model for identifying DeepFake videos. The methodology includes a number of processes, such as feature extraction from the data, model training, and testing. We have used a convolutional neural network to try and identify the deepfake video. We have attempted to train the proposed algorithm using five deep learning algorithms: Baseline CNN, XceptionNet, InceptionResNetV2, EfficientNetV2S, and EfficientNetV2M for classification utilizing the convolutional neural network step. We will first go over the proposed method for separating the feature. The identification of the deepfake video using a deep learning system will next be covered. Following that, we will introduce the suggested model and discuss each layer involved in identifying the deepfake video using CNN.

4.2 Our Working Approach for Deepfake Video Detection

Our proposed system of Deepfake Video detection using a deep learning algorithm consists of six stages:

- **Stage-1:** Dataset Acquisition
- **Stage-2:** Data Pre-processing
- **Stage-3:** Dataset Split
- **Stage-4:** Cost-sensitive
- **Stage-5:** Proposed Model

The proposed methodology is thoroughly described in this section. This is a flow chart of the suggested process in fig 4.1

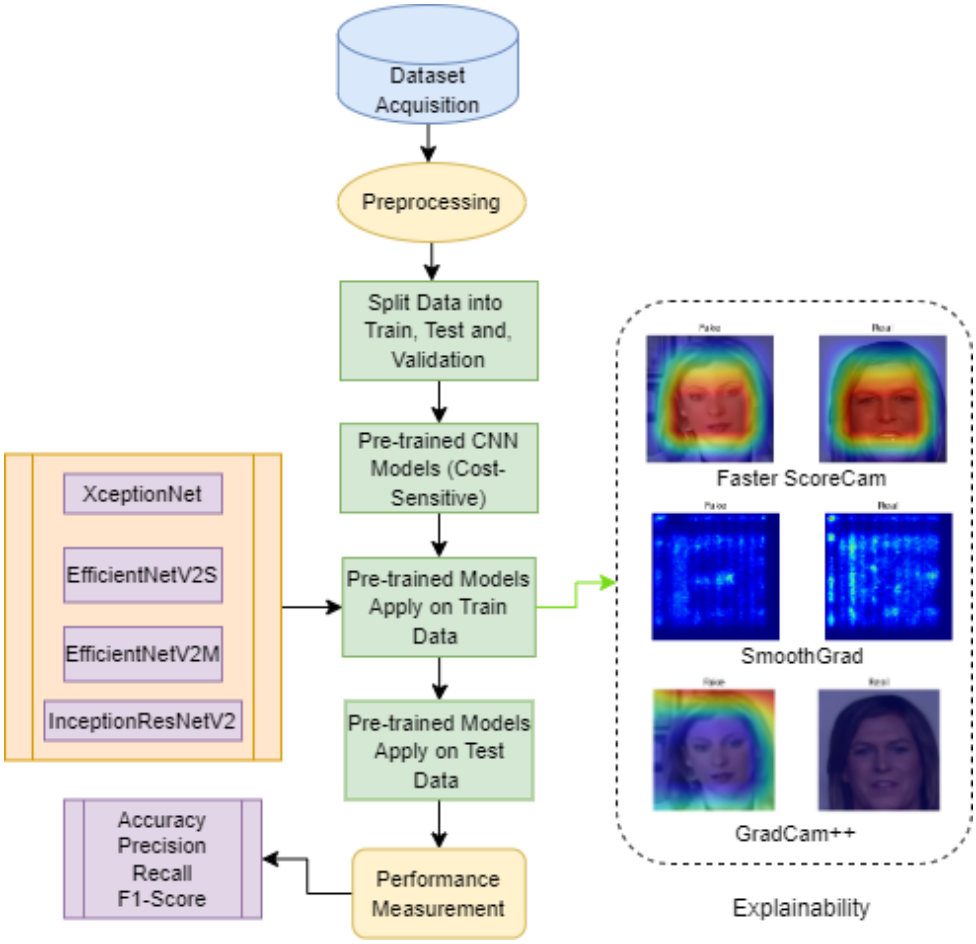


Figure 4.1: Methodology

4.2.1 Dataset Acquisition

There are a few publicly available databases that include both real and fake videos. The FaceForensics++ [7] dataset and Celeb-DF [11] were discovered the most frequently used sources of data, based on our research into previous studies. The frequency of the datasets is listed below in Table 4.1

Table 4.1: Information of Datasets

Name	Real	Fake	Forgery Methods
FaceForensics++	1000	4000	4
Celeb-DF	590	5639	1

4.2.2 Data Preprocessing

Firstly, it was determined that if the video was corrupted or not. The video file is immediately erased if it is discovered to be corrupted. Next, the Python "face recognition" package was used to extract faces from videos in order to find faces inside the video. It effectively filters out unnecessary background elements to extract just the faces. Each frame was resized to a constant 224×224 resolution during this process, and the video has 30 frames per second (fps) frame rate. The proposed approach used inter-frame differences during the key frame extraction step. The basic idea is simple: after loading the video, how much each pair of frames differs from one another was figured out. Then it was put in a local maximum detection process. Keyframes are specifically those for which the average inter-frame difference is at the local maximum. It is important to notice that noise was removed effectively and stopped the repeated extraction of frames from similar situations by smoothing the average difference values before completing the local maxima computation. Ultimately, the raw image data was transformed into a numpy array to further optimize computation time and improve training quality. The final dataset visualization is presented in Figure 4.1.

Figure 4.2: Preprocessed Data

4.2.3 Data Split Ratio

Training (80%), Testing (10%), and Validation (10%) were the three divisions of the dataset. To improve the distribution of data before distribution, the entire dataset was stratified. Dataset information is shown in Table 4.2 and Table 4.3.

Table 4.2: Data distribution of CelebDf-V2

Class	Train	Validation	Test
Fake	8995	1000	1111
Real	2383	265	294

Table 4.3: Data distribution of FaceForensics++

Class	Train	Validation	Test
Fake	6728	748	831
Real	3847	428	475

4.2.4 Cost-sensitive

It is crucial to prevent the machine learning model from developing a bias towards the dominant class in order to address the problem of class imbalance in the training data. To give the minority classes more weight throughout the training process, one typical strategy is to assign class weights to each class. Usually, the training dataset's distribution of class labels is used to determine these class weights. Particularly, each class is given a weight that is inversely proportionate to how frequently it appears in the data. As a result, classes that are less common are given greater weights while classes that are more prevalent are given lower weights.

The class weights are typically gathered into a dictionary or another format that can be conveniently incorporated into the model training procedure once they have been calculated. This dictionary is subsequently utilized as an input parameter throughout the model construction process in the majority of machine learning libraries and frameworks. The model is urged to pay more attention to the minority classes during training by adding these class weights, aiding in its learning of a more balanced representation of the data. This method helps models perform better and can be extremely useful when working with unbalanced datasets, such as those seen in tasks requiring fraud detection, uncommon event prediction, or medical diagnosis.

4.2.5 Model Trainig

In this study, we implemented four pre-trained CNN models (XceptionNet, EfficientNetV2S, EfficientNetV2M, and InceptionResNetV2). A 0.001 learning rate was applied for all the models. If the model consistently performs poorly for a set number of epochs, the ReduceLROnPlateau class was used to lower the learning rate. In each model, there is a batch size of 16 and the optimization technique has been implemented using the 'Adam' optimizer. The output of the base model was enhanced with the GlobalAveragePooling2D layer, followed by the ReLU activation function and a Dense layer. A dropout layer with a rate of 0.5 is em-

ployed to prevent overfitting. At the final dense layer of each model, the Softmax activation function was employed.

4.3 Summary

The proposed deepfake video detection methodology is explained in this chapter. With appropriate visuals and discussion, a detailed demonstration of a deepfake face detection from a video with the use of deep learning models is shown.

Chapter 5

Experimental Result Analysis

5.1 Overview

In this section, we will comprehensively describe the outcomes of our proposed methodology. We carried out the deepfake video detection using Baseline CNN, XceptionNet, InceptionResNetV2, EfficientNetV2S, and EfficientNetV2M models. Following we will do the performance evaluation process and compare the performance of these models. Furthermore, we also analyze our model with the existing model in terms of fake video detection.

5.2 Experimental Setup

We used the Jupyter Notebook and various Python packages such as Numpy, Pandas, OpenCV, etc for image processing. For the traditional classifiers, we used the Scikit-Learn. We used Python version 3.9 with Anaconda. For training and testing our model through CNN, we used Tensorflow and Keras framework. We used the dedicated GPU which is provided by Google Colab.

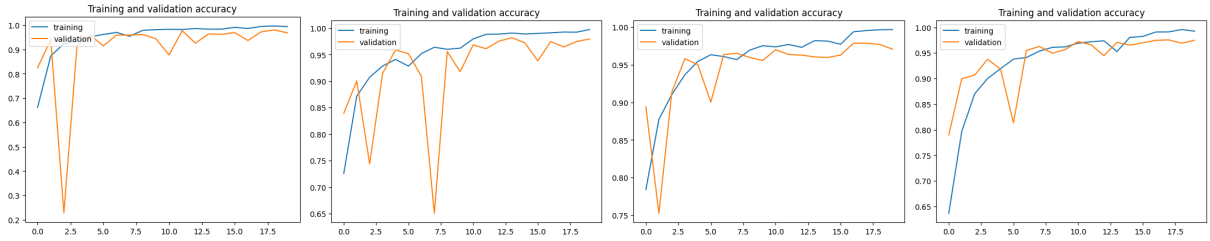
5.3 Experimental Result

Two distinct datasets were used for all four pre-trained models. All of these models have mostly correctly predicted the fake faces from videos. The results obtained from the datasets using the four pre-trained models are presented in Table 5.1 and 5.2. Moreover, the confusion matrix is shown in Figure 5.3 for the CelebDf-V2 dataset and Figure 5.9 for the FaceForensics++ dataset, and the model's explainability is shown in Figure 5.4 - Figure 5.6 for CelebDF-V2 dataset and Figure 5.10 - Figure 5.12 for FaceForensics++ dataset.

Table 5.1: Performance Metrics of Weighted Average on CelebDf-V2 Dataset

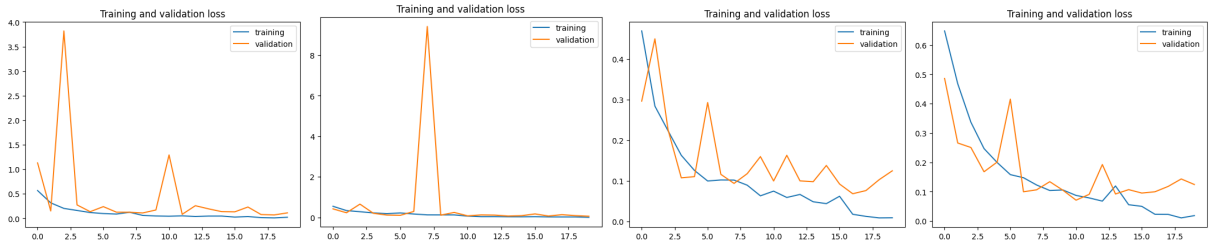
Model	Accuracy	Precision	Recall	F1-Score
XceptionNet	**98%**	0.98	0.98	0.98
EfficientNetV2S	97%	0.97	0.97	0.97
EfficientNetV2M	97%	0.97	0.97	0.97
InceptionResNetV2	97%	0.97	0.97	0.97

Table 5.1 shows that the models had great accuracy, with XceptionNet leading with 98% and EfficientNetV2S, EfficientNetV2M, and InceptionResNetV2 following very closely with 97% each. This suggests that the CelebDf-V2 Dataset performed exceptionally well overall.



(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

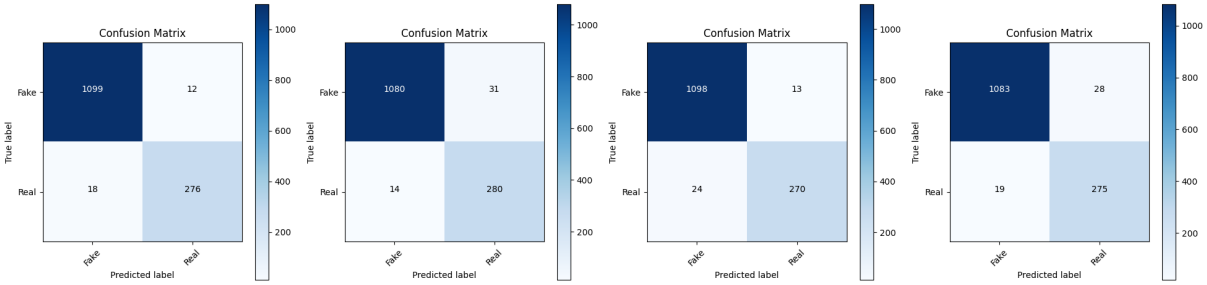
Figure 5.1: Training Accuracy Curve of All Pre-trained Models Based on CelebDF-V2 Dataset



(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

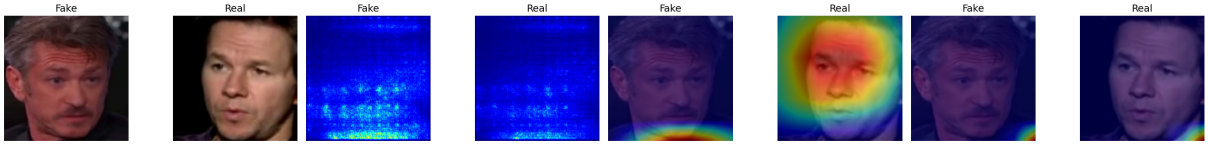
Figure 5.2: Training Loss Curve of All Pre-trained Models Based on CelebDF-V2 Dataset

From Figure 5.1 and Figure 5.2 we can observe all the training accuracy curves and loss curves of all pre-trained models and can see which model performs well.



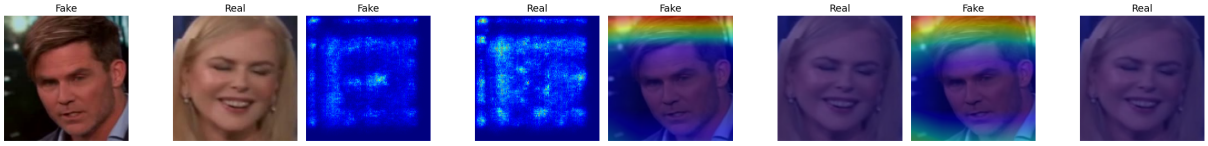
(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

Figure 5.3: Confusion Matrix of All Pre-trained Models Based on CelebDF-V2 Dataset



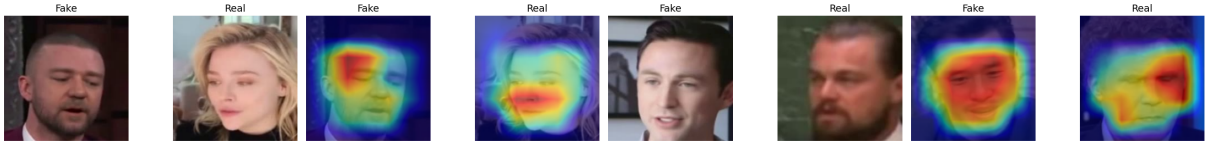
(a) Original Image (b) SmoothGrad (c) GradCam (d) Faster ScoreCam

Figure 5.4: Visualization of sample outputs how well the XceptionNet model can capture the faces on the CelebDF-V2 dataset.



(a) Original Image (b) SmoothGrad (c) GradCam (d) GradCam++

Figure 5.5: Visualization of sample outputs how well the InceptionResNetV2 model can capture the faces on the CelebDF-V2 dataset.



(a) Original Image (b) Faster ScoreCam (c) Original Image (d) Faster ScoreCam

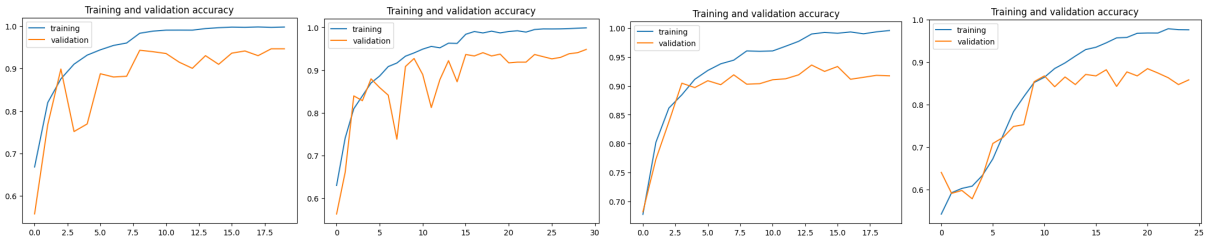
Figure 5.6: Visualization of sample outputs how well the EfficientNetV2S and EfficientNetV2M models can capture the faces on the CelebDF-V2 dataset.

Using Expainable AI it is determined how robust the model is. From Figure 5.4 to Figure 5.6 it has been proved that Expainable AI produces different results for different models. In the case of CelebDF-V2 EfficientNetV2S and EfficientNetV2M have shown greater possibilities in Faster ScoreCam. Explainable AI can detect the real and fake faces in this study which is our ultimate goal. It can be visualized what the model sees in terms of DeepFake Face Detection.

Table 5.2: Performance Metrics of Weighted Average on FaceForensics++ Dataset

Model	Accuracy	Precision	Recall	F1-Score
InceptionResNetV2	**94%**	0.94	0.94	0.94
XceptionNet	93%	0.93	0.93	0.93
EfficientNetV2S	92%	0.92	0.92	0.92
EfficientNetV2M	88%	0.89	0.88	0.88

Table 5.2 shows that the models also had high levels of accuracy, with InceptionResNetV2 resulting with 94%, XceptionNet coming in second with 93%, and EfficientNetV2S coming in third with 92%. It's important to note that these models did well overall on the FaceForensics++ Dataset, despite the fact that EfficientNetV2M had a slightly lower accuracy of 88%.



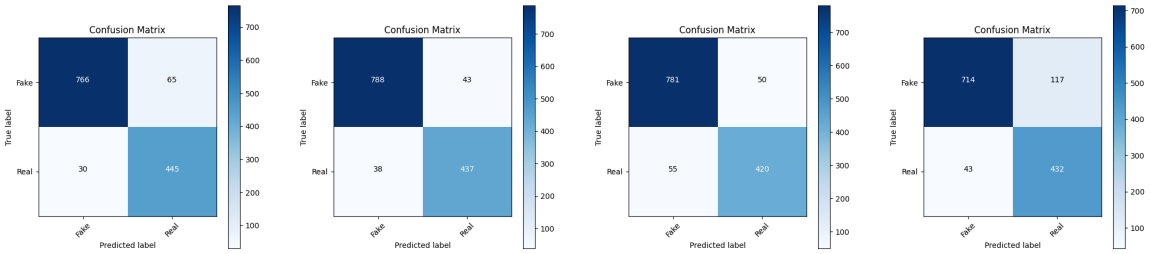
(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

Figure 5.7: Training Accuracy Curve of All Pre-trained Models Based on FaceForensics++ Dataset

(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

Figure 5.8: Training Loss Curve of All Pre-trained Models Based on FaceForensics++

From Figure 5.7 and Figure 5.8 we can observe all the training accuracy curves and loss curves of all pre-trained models and can see which model performs well.

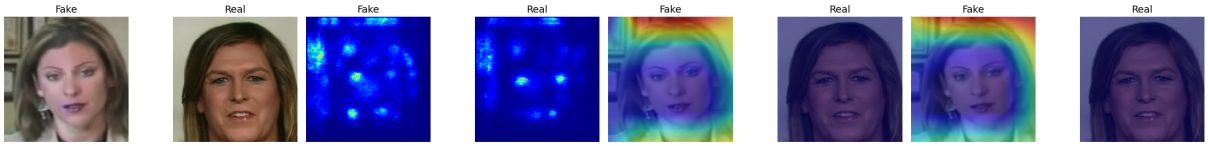


(a) XceptionNet (b) InceptionResNetV2 (c) EfficientNetV2S (d) EfficientNetV2M

Figure 5.9: Confusion Matrix of All Pre-trained Models Based on FaceForensics++ Dataset

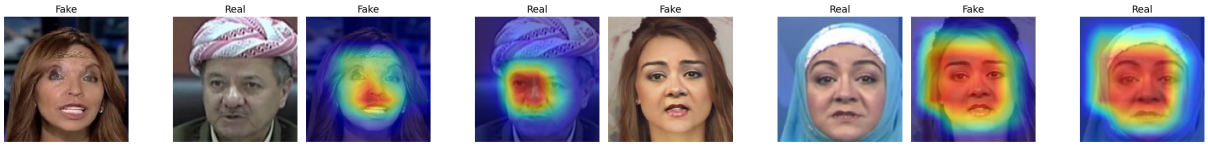
(a) Original Image (b) SmoothGrad (c) GradCam (d) Faster ScoreCam

Figure 5.10: Visualization of sample outputs how well XceptionNet model can capture the faces on FaceForensics++ dataset



(a) Original Image (b) SmoothGrad (c) GradCam (d) GradCam++

Figure 5.11: Visualization of sample outputs how well InceptionResNetV2 model can capture the faces on FaceForensics++ dataset



(a) Original Image (b) Faster ScoreCam (c) Original Image (d) Faster ScoreCam

Figure 5.12: Visualization of sample outputs how well EfficientNetV2S and EfficientNetV2M model can capture the faces on FaceForensics++ dataset: First row represents the EfficientNetV2S and second row represents EfficientNetV2S model

For the Faceforensics++ dataset, the Expainable AI has proven much more potential in the case of GradCam, GradCam++, and Faster ScoreCam. All the models for this dataset have more robustness than the CelebDF-V2 dataset which is shown in Figure 5.10 to Figure 5.12. Explainable AI can also detect real and fake faces from the Faceforensics++ dataset. It can be visualized more accurately what the model sees in terms of DeepFake Face Detection in this particular case.

5.4 Summary

In this section, we provided a comprehensive analysis of many models utilizing performance metrics such as the confusion matrix, loss curve, accuracy curve, and explainable AI. The evaluation of these measures allowed for an extensive understanding of the model's advantages and disadvantages, permitting well-informed judgments on the model's applicability for practical applications.

Chapter 6

Result Analysis & Comparison

6.1 Overview

In this section, we compared the performance of the existing models to the proposed models. Though we used two datasets to measure the performance of the proposed model, we will take into consideration the results of the FaceForensics++ dataset.

6.1.1 Performance Comparison

In this part, we compared our result with some of earlier research. With an excellent accuracy rate of 98%, our proposed deepfake face identification algorithm exceeds earlier state-of-the-art techniques. The outcome proves how effective our strategy was. This accuracy, in particular, performs better than a number of important previous studies in the field. A Deep Neural Network architecture developed by Lee and Kim [6] obtained 97% accuracy. In their study, Xu et al. [9] reached an accuracy of 91.2%. While average and maximum pooling strategies were able to achieve an accuracy of 85.24%, the proposed method of Kohli et al. [12] produced a maximum accuracy of 86.08%. Two different models were introduced by Kim et al. [13], and their proposed model successfully detected deepfake videos with an accuracy of 86.97%.

6.2 Summary

In this section, we compared the performance of our proposed deepfake face identification models with existing ones, primarily using the FaceForensics++ dataset. Our models achieved a remarkable accuracy rate of 98%, surpassing earlier state-of-the-art techniques.

For context, previous studies achieved accuracies ranging from 86.08% to 97%, showcasing the superior performance of our approach.

Chapter 7

Counclusion

7.1 Advantages and Limitations

The deepfake video detection method used in this study relies on extracting key frames as a resource-effective strategy. Our research has shown that it is possible to build reliable models for deepfake video detection even with limited resources and its findings are encouraging. The interpretability of our models has been improved by our usage of Explainable AI techniques, offering important insights into their decision-making processes. The lack of high-quality video datasets for training and evaluation was one of the main issues we faced, which made it difficult to fully explore the potential of the models. However, more deepfake video datasets are expected to be released in the future, which will considerably increase the accuracy of our research and the overall efficiency of deepfake detection systems. Our models will have the chance to improve their comprehension of deepfake patterns and characteristics as these datasets become available.

7.2 Future Works

We believe that Explainable AI applications will develop further, possibly producing even better outcomes in terms of model interpretability and decision explanation. This innovation will not only improve our current detection method but also increase our understanding of the methods behind deepfake video production and detection. Furthermore, the field of deepfake detection is dynamic, with continuous advancements in AI and machine learning. Future research in this field may present cutting-edge models and methodologies that are superior to the capabilities of existing systems, ultimately resulting in more dependable and robust deepfake detection systems. These developments will be necessary to maintain the integrity of online material and keep up with the deepfake technology field.

References

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

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

[3] A. Mitra, S. P. Mohanty, P. Corcoran, and E. Kougianos, "A machine learning based approach for deepfake detection in social media through key video frame extraction," *SN Computer Science*, vol. 2, pp. 1–18, 2021.

[4] M. Tanvir Rouf Shawon, G. Shahariar Shibli, F. Ahmed, and S. K. Saha Joy, "Explainable cost-sensitive deep neural networks for brain tumor detection from brain mri images considering data imbalance," *arXiv e-prints*, pp. arXiv–2308, 2023.

[5] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on knowledge and data engineering*, vol. 18, no. 1, pp. 63–77, 2005.

[6] G. Lee and M. Kim, "Deepfake detection using the rate of change between frames based on computer vision," *Sensors*, vol. 21, no. 21, p. 7367, 2021.

[7] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

[8] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," *arXiv preprint arXiv:2006.07397*, 2020.

[9] B. Xu, J. Liu, J. Liang, W. Lu, and Y. Zhang, "Deepfake videos detection based on texture features.," *Computers, Materials & Continua*, vol. 68, no. 1, 2021.

[10] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.

[11] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df (v2): a new dataset for deepfake forensics [j]," *arXiv preprint arXiv*, 2019.

[12] A. Kohli and A. Gupta, "Detecting deepfake, faceswap and face2face facial forgeries using frequency cnn," *Multimedia Tools and Applications*, vol. 80, pp. 18461–18478, 2021.

[13] M. Kim, S. Tariq, and S. S. Woo, "Fretal: Generalizing deepfake detection using knowledge distillation and representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1001–1012, 2021.

[14] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017.

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

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[17] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," 2021.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[20] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.