XAI – EMPOWERED ENSEMBLE DEEP LEARNING FOR DEEPFAKE DETECTION

A PROJECT REPORT

Submitted by

AMIDELA ANIL KUMAR [CB.EN.U4CSE20206]
DHEEPTHI PRIYANGHA S J [CB.EN.U4CSE20217]
PULLELA MEGHANA [CB.EN.U4CSE20239]
MUPPALLA DHEERAJ [CB.EN.U4CSE20241]

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112

MAY 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, COIMBATORE – 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **XAI – EMPOWERED ENSEMBLE DEEP LEARNING FOR DEEPFAKE DETECTION** submitted by Amidela Anil Kumar (CB.EN.U4CSE20206), Dheepthi Priyangha S J (CB.EN.U4CSE20217), Pullela Meghana(CB.EN.U4CSE20239), Muppalla Dheeraj (CB.EN.U4CSE20241) in partial fulfillment of the requirements for the award of Degree **Bachelor of Technology** in Computer Science and Engineering is a bonafide record of the work carried out under our guidance and supervision at the Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore.

Dr.Aarthi R
(Assistant Professor)
Department of CSE

Dr.Vidhya Balasubramanian
Chairperson
Department of CSE

Evaluated on:

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, the undersigned solemnly declare that the project report **XAI – EMPOWERED ENSEMBLE DEEP LEARNING FOR DEEPFAKE DETECTION** is based on our own work carried out during the course of our study under the supervision of Dr.Aarthi R, (Assistant Professor), Computer Science and Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement has been made wherever the findings of others has been cited.

Amidela Anil Kumar[CB.EN.U4CSE20206]-
Dheepthi Priyangha S J [CB.EN.U4CSE20217]-
Pullela Meghana[CB.EN.U4CSE20239]-
Muppalla Dheeraj[CB.EN.U4CSE20241]-

ABSTRACT

Deep learning-based approaches have shown remarkable progress in detecting deepfake images, which pose significant challenges to the authenticity of digital media. Recent technologies have evolved rapidly, allowing the creation of fake videos/images that are realistic. These artificial media pieces have provoked grave worries about the possibilities of their misuse in different areas like politics, journalism, and communication. Consequently, the demand for deepfake detection methods is increasing. In response to this growing threat, there is an increasing demand for robust deepfake detection methods. This paper proposes an ensemble-based approach for deepfake detection, leveraging deep learning techniques. Specifically, a combination of CNN models, including VGG, Xception, RegularizedConvNet, and RegularizedConvDenseNet, is employed, designed for image classification. Our approach integrates predictions from multiple models to enhance detection accuracy and reliability. Specifically, the base is formed by the ensemble of RegularizedConvNet and RegularizedConvDenseNet, which cooperate to prevent the spread of deepfakes and protect digital integrity.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our beloved Satguru **Sri Mata Amritanandamayi Devi**. This acknowledgment is intended to thank all those people involved directly or indirectly with our project. We express our thanks to **Dr. Vidhya Balasubramanian**, Chairperson, Department of Computer Science and Engineering and Principal, School of Computing, Amrita Vishwa Vidyapeetham, **Dr. C. Shunmuga Velayutham and Dr. N. Lalithamani,** Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guides, **Dr. Aarthi R and Dr. Sruthi C J** for their guidance, support and supervision. We feel extremely grateful to **Dr. S. Padmavathi, Dr. Raghesh Krishnan K, Dr. Aarthi R, Dr. Sruthi C J, Ms. Sujee R** for their feedback and encouragement which helped us to complete the project. We would also like to thank the entire members of the Department of Computer Science and Engineering. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project. Finally, we would like to thank all those who have helped, guided, and encouraged us directly or indirectly during the project work.

Amidela Anil Kumar [CB.EN.U4CSE20206]

Dheepthi Priyangha S J [CB.EN.U4CSE20217]

Pullela Meghana [CB.EN.U4CSE20239]

Muppalla Dheeraj [CB.EN.U4CSE20241]

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ROC	Receiver Operating Characteristic
AUC	Area under the ROC Curve
ROI	Region Of Interest
IDE	Integrated Development Environment
DFDC	DeepFake Detection Challenge
DARPA	Defence Advanced Research Projects Agency
FPR	False Positive Rate
TPR	True Positive Rate
XAI	Explainable Artificial Intelligence
LIME	Local Interpretable Model-agnostic Explanations

Chapter 1

INTRODUCTION

Deepfake is a kind of fake multimedia, which is produced by the fake content that is based on the existing materials and usually a person's media. In 2017, this name "deepfake" was first used by a Reddit user known as the user named deepfake (1). The fake content uses graphics, audio, and face-swapping tech based on Artificial Intelligence like generative adversarial networks (2). The deepfake operations in cybercrimes (3) include identity theft, cyber extortion, imposter scamming, fake news, instigate violence, financial fraud, cyberbullying, celebrity fake obscenity videos for blackmail, democratic elections, and many other applications. Uncovering deepfake media is a big deal, imposing high demand on digital forensics.

Recently, there has been progress in architectures, like GANs (Generative Adversarial Networks) (2), that has simplified deep fake generation. The architecture works on a source image and set of desired distortions, and then generates a believable manipulated image. While these GANs-generated DeepFakes have produced more realistic-looking output, they however do have visible flaws that a Convolutional Neural Network (CNN) is able to reveal. Through such models, the faces of persona and their facial movements are observed and images are produced by synthesis. Deepfake methods normally require a large amount of image and video data to train models to create photorealistic images and videos. As public figures such as celebrities and politicians may have a large number of videos and images available online, they are initial targets of deepfakes.

Depending on the intentions, deepfake may mainly be used for sabotaging organizations and targeting individuals. The cyber attacks using deepfake technology are now becoming a serious concern. The making of deepfakes is unethical; this is a major criminal offense. According to the Sensity report (4), the vast majority 96 percent of the deepfakes are extraordinary, which comes into focus. UK, USA, Canada, India, South Korea are among the countries affected by the deepfakes (4). In 2019, the chief executive officer was the target of cybercriminal attacks that led to telephone scammers stealing a sum of 243,000 dollars by transferring the amount into the bank account (5).

Effective and advanced tool is necessary to put the brakes on crime and prevent being allowed to pass.

Consequently, there is now greater significance to finding the truth in the digital world. Since most deep fakes are used for malicious purposes and are now easily produced by anyone using readily available tools, dealing with them is much more challenging (5). Deepfakes have been found using a variety of approaches so far. Since deep learning is also the foundation for the majority, a struggle has emerged between poor and excellent deep learning applications (6). Thus, in order to address this issue, the Defence Advanced Research Projects Agency (DARPA) of the United States initiated a media forensics research plan with the goal of creating techniques for detecting fraudulent digital information (7). Moreover, Facebook announced an AI-based deep fake detection challenge in association with Microsoft to stop the use of deep fakes to deceive viewers (8).

The researchers have been examining the fields of Machine Learning and Deep Learning (DL) for numerous years to try to find out how it can be used to identify deep fakes in audiovisual media. The ML algorithms use manual extraction of the non-functional features before the category prediction phase. Therefore, the operation of these systems hampers its progress if dealing with big volume of data. Though, DL algorithms automatically carry out these tasks that have a lot been of use in many other applications ranging from duplicitous videos among others. Convolutional neural network (CNN), which has proven to be successful as a few reasons: first, using state-of-the-art deep learning has led to automatically extracting low-level and high-level data from the database (9). Thus, an extension of these methods has gained the trust to be an appreciated resource for failure studies by scientists worldwide (9).

In this paper, deepfake identification is performed by developing an ensemble method of RegularizedConvNet and RegularizedConvDensinet by utilising the 140k Real and Fake Faces dataset. Ensemble of Deep neural networks outperform at image classification (10) and localization due to their strong prediction abilities. By incorporating several models with different architectures, ensembles can provide better predictability and stability. However, it can be difficult to predict which algorithm will achieve the maximum accuracy for a given prediction task and dataset.This is because certain deep learning methods may have poor overall prediction, but excel at classifying specific

subclasses. Ensembles can effectively utilize the combined strengths of various deep learning algorithms to overcome these limitations (11).

In this paper, a novel approach to deepfake detection is introduced by leveraging Explainable Artificial Intelligence (XAI) techniques, particularly the Local Interpretable Model-agnostic Explanations (LIME). Our methodology aims to identify the specific image features utilized by the model in identifying deepfakes, such as the regions corresponding to eyes, nose, mouth, and others. By employing LIME, it is possible to provide interpretable insights into which regions of an image are crucial for the model's decision-making process. This enables us to discern which parts of an image are potentially manipulated or deepfaked, thus enhancing the transparency and reliability of our deepfake detection system.

1.1 Impact of DeepFake on Society/Social Media:

DeepFake technology which uses AI to create highly convincing fake audio and video content, poses significant risks to society across various domains.

Misinformation: Deepfakes can be used to create realistic-looking video recordings/ images of public figures/celebrities/politicians saying or doing things they never did. This can be weaponized to spread false information, manipulate public opinion, and disrupt elections.

Identity Theft: Individuals can be targeted for identity theft, where their likeness is used in fraudulent activities, such as creating fake social media profiles, committing financial fraud, or engaging in cyberbullying.

Privacy Violations: Deep fake technology can be used to superimpose an individual's face onto explicit or inappropriate content, a practice is commonly known as "face-swapping". This constitutes a severe violation of privacy and can lead to problems like harassment, blackmail, and defamation.

Deterioration of Trust in social media: Deepfake can make it increasingly difficult for users to discern real from fake content, leading to the deterioration of trust in the authenticity of media shared on social media platforms. As deepfake technology advances and becomes more accessible, users may become more skeptical of the videos and images they encounter, leading to a decline in trust in the veracity of online media.

This erosion of trust can have far-reaching consequences for online discourse, media credibility, and interpersonal relationships.

Damage to Political Figures: Deepfake images and videos depicting political figures engaging in inappropriate behavior can damage their reputation. Even if the content is later proven false, the damage to the political figure's image may already be done, affecting public perception and potentially influencing election outcomes.

Crime and Fraud: Criminals could exploit deepfake technology to impersonate others in fraudulent activities, making it difficult for authorities to identify and prosecute the culprits.

Addressing the challenges posed by deepfake technology through deep learning architectures involves a comprehensive and adaptable approach. Deep learning offers powerful tools for analyzing and understanding complex patterns in multimedia data, making it well-suited for detecting and mitigating the effects of deepfake manipulation. One key aspect is feature extraction, where deep learning models can automatically learn and extract relevant features from images, videos, and audio recordings. By identifying subtle inconsistencies or artifacts indicative of deepfake manipulation, these features serve as valuable input for subsequent analysis and classification tasks. Additionally, adversarial training techniques can enhance model robustness against sophisticated deepfake attacks by exposing the model to adversarial examples during training, thereby improving its ability to detect and mitigate manipulated content. Ensemble methods, which combine multiple models or techniques, further enhance the effectiveness of deep learning architectures in combating deepfake technology. By leveraging the complementary strengths of different models or approaches, ensemble learning can improve overall performance and generalization, making it more challenging for adversaries to evade detection. Continual learning and adaptation are also essential, as deepfake techniques evolve over time. By continuously updating and refining deep learning models based on emerging threats and evolving manipulation techniques, researchers can ensure that their methods remain effective and resilient in the face of new challenges. Therefore, deep learning architectures provide a versatile and powerful framework for addressing the complex and evolving nature of deepfake technology, offering promising avenues for detecting, mitigating, and ultimately combating the spread of manipulated media content.

1.2 Problem Definition

In the era of rapid technological advancements, the emergence of deepfake technology has raised significant concerns regarding the authenticity of digital media. Deepfake images and videos, generated using artificial intelligence algorithms, have the potential to deceive viewers by manipulating content in existing media or creating entirely fabricated visual narratives. As the proliferation of deepfake content poses serious threats to various domains, including journalism, politics, and personal privacy, the need for effective detection mechanisms becomes paramount. The primary objective of this project is to develop a robust deep-learning model capable of accurately identifying deepfake images/videos within a diverse dataset. Leveraging state-of-the-art deep learning architectures, including convolutional neural networks (CNNs) and transfer learning techniques, the model aims to learn discriminative features that distinguish between real and fake images/videos. Through rigorous evaluation using standard metrics such as accuracy, precision, recall, and F1-score, the performance of the model will be assessed on both validation and test sets to ensure its effectiveness in real-world scenarios. Additionally, ensemble learning techniques will be explored to further enhance classification performance by combining predictions from multiple individual models. The ultimate goal of this project is to provide a reliable solution for detecting deepfake images/videos, thereby mitigating the potential risks associated with the spread of misinformation and preserving the integrity of digital content across various online platforms and applications.

1.2.1 Problem UseCase:

1) Political Campaigns and Fact-Checking

During election campaigns, deepfake detection plays a pivotal role in safeguarding the integrity of the political process. Political campaigns can utilize deepfake detection to thwart the dissemination of maliciously fabricated videos, which have the potential to sway public opinion and damage candidates' reputations. Fact-checking organizations leverage deepfake detection as a critical tool to swiftly debunk false claims and disinformation, ensuring that accurate information prevails in the public discourse, fostering transparency, and preserving the integrity of the democratic process. By countering the

deceptive power of deepfakes, these measures contribute to the maintenance of fair and informed elections.

2) Media and Journalism Integrity:

Deepfake detection plays a crucial role in upholding the integrity of media and journalism. News agencies and media organizations rely on this technology to conduct authenticity verification checks on videos and images before publication. By doing so, they ensure that the content they disseminate is trustworthy, accurate, and free from manipulations that could mislead the public. Moreover, journalists themselves utilize deepfake detection tools to validate the credibility of their sources and interviewees. This proactive approach helps safeguard journalistic integrity by preventing the inadvertent inclusion of manipulated or deceptive content in news stories. In an era where misinformation can have far-reaching consequences, deepfake detection stands as a vital tool in maintaining the credibility and reliability of the media.

3) Legal and Law Enforcement:

Deepfake detection is pivotal in ensuring the integrity of legal proceedings and criminal investigations. In the realm of legal proceedings, legal professionals utilize deepfake detection to safeguard the authenticity of video evidence presented in court. This preventative measure helps prevent tampering or manipulation of crucial evidence, upholding the principles of justice and ensuring that legal verdicts are based on accurate and unaltered information. For law enforcement agencies, deepfake detection serves as a vital tool in criminal investigations. Investigators can analyze and verify videos that are related to criminal activities or threats, enhancing their ability to assess the veracity of evidence and witness statements. By leveraging deepfake detection, law enforcement agencies can maintain the highest standards of investigative rigor and objectivity, contributing to the pursuit of justice and the protection of communities from potential threats.

The development "XAI – Empowered Ensemble Deep Learning for Deepfake Detection" is aimed at creating a holistic solution to the growing challenge that deepfakes pose to humanity. This initiative exploits different deep learning techniques, mostly involving the convolutional neural network (CNN) structures that are developed optimally for image recognition. Analysis of dataset progress by using appropriate dataset manipulation methods, such as normalization and resizing is what the project does with

the purpose of making the input data ready for the model training. Critical modules include the training and adapting of VGG16 and Xception retraining as well as the creation of individualized CNN architectures. In addition, elaborating a hybrid model as a combination of the results from different individual models performs a better quality of deepfake detection in terms of effectiveness and verifiability. The project takes a careful look at the performance of these models employing a variety of metrics and concludes that the Ensemble approach already works beyond a reasonable doubt in producing a result with commendable accuracy and higher detection capacity. Besides, the project does the integration of Explainable Artificial Intelligence features, especially LIME, and the probing of the decisions of the ensemble model and the biases factors in the datasets. Lastly, the project intends to tackle AI-based applications geared towards mitigating deepfake, and fostering security, and trustworthiness in the digital world. The project goes beyond the mere tactic through its comprehensive and transparent methodology and pledges for improved resilience against the proliferation of deepfake manipulation and the authenticity of digital environments.

Chapter 2

LITERATURE SURVEY

Through this literature survey, the aim is to delve into the existing body of knowl- edge, and identify key theories, methodologies, findings, and gaps in research. By synthesizing and analyzing previous studies, the groundwork for our own re- search is laid, providing a comprehensive understanding of the subject's historical and current context. This literature survey serves as a vital framework upon which our own contributions and insights are built in subsequent chapters.

2.1 Deepfake detection methods

2.1.1 Deep fake detection and classification using error-level analysis and deep learning

The paper [12] suggests a solution that detects and classifies deep fake content in social media networks - an important measure to suppress the spreading of disinformation. It merges Error Level Analysis (ELA) with Convolutional Neural Networks (CNNs) for an effective feature extraction. ELA provides for the adjustment processes to spot any digital alterations, however, CNNs like GoogLeNet and ResNet18 are deep learning networks that use images as the independent variables. Classification is based on SVM and KNN, which work for the exact data segmentation. The results of the aforementioned framework are particularly noteworthy taking into consideration the high level of accuracy of 89.5% achieved by the algorithm, thus proving its effectiveness. Via image preprocessing, image detection, and advanced deep learning principle implementation, the system ensures the protection of society against misinformation and propaganda, thus relaxing the digital age security guarantee.

2.1.2 DeepVision: Deepfake Detection Using Human Eye Blinking Pattern

The paper "DeepVision: Deepfake Detection Using Human Eye Blinking Pattern" [33] provides a method for detecting deepfakes based on the analysis of blinking patterns, a typical single-frame human action that is difficult for deepfakes to convincingly imitate. This approach entails steps: preprocessing, eye detection and tracking, and abnormality detection through statistical methods. DeepVision's architecture includes target detection and eye tracking to make blinking pattern comparisons with its database of natural movements to finally allow deepfake accuracy testing. The dataset called FaceForensics++ includes deepfake and normal videos that has an accuracy rate of 87.5% in detection across a wide range of scenarios. Yet in cases of mental illness or if dopamine activity is the cause of eyelid movements, the algorithm's effectiveness may not be shown. DeepVision offers a good catch against human blink patterns which helps to form a unique idea about the existence of deepfake technology.

2.1.3 Capsule-forensics: using capsule networks to detect forged images and videos

The paper "Capsule-forensics: "Capsule Networks for Deepfake and Media Fake Detection" [34] is a conditional task that aims to address the media detection problem in the context of the development of media generation techniques, such as deepfake videos. It utilizes the first time an idea using a capsule network, invented initially for image processing tasks, to identify numerous kinds of falseness, including imitation attacks and computer-made pictures. Due to cross-entropy loss, the applied technique exceeds current methods, scoring high accuracy in face swapping and replay attack detection. The evaluation on FaceForensics and the dataset from Rahmouni et al. have proven good performance. In detail, besides a perfect accuracy of full-size test images, a relatively high detection accuracy also exists. Further undertakes are directed at the reinforcement of the method's resilience to tackle the multipronged attacks and the significant issues of the research field, promising an enhanced fight back against the bunch of media hoaxes.

2.1.4 DeepFake Image Detection by Omkar Salpekar

The paper "DeepFake Image Detection by Omkar Salpekar" [36] addresses the pressing need for robust DeepFake detection in the digital age, proposing a 2-phase learning approach using Siamese Neural Networks and CNNs. Phase 1 involves training a ResNet18-based CNN as a Siamese Network, termed the Common Fake Feature Network (CFFN), with triplet loss to learn feature-level distinctions between real and fake images. The CFFN is fine-tuned with regularized cross-entropy loss on a large dataset extracted from the DeepFake Detection Challenge (DFDC) dataset on Kaggle, comprising over 470GB of mp4 videos. A deliberately imbalanced dataset, with an 80-20 split between real and DeepFake images, ensures diverse detection capabilities. The model achieves impressive training and validation accuracies of 94% and 91%, respectively, demonstrating promising results in discerning between authentic and manipulated images.

2.1.5 DeepFake Detection Based on Discrepancies Between Faces and their Context

The paper [22] introduces a novel approach for detecting face swapping in images and videos by analyzing the disparities between manipulated facial regions and their surrounding context. It utilizes two deep learning networks: a Face Identification Network and a Context Recognition Network, to capture discrepancies between the manipulated face and its unaltered context. The method operates on the assumption that face manipulation techniques predominantly affect the internal facial features while leaving the outer context unchanged. By employing a two-stream neural network architecture, the method extracts high-level facial features and contextual information separately, which are then fused for classification. A large-scale dataset containing both DeepFake and genuine images is utilized for training and evaluation, including benchmark datasets like DFDC and Face Forensics++. The proposed method achieves promising results, outperforming existing techniques in DeepFake detection across various benchmark datasets. This research contributes to advancing the field of DeepFake detection by providing a robust and generalized approach that leverages disparities between manipulated faces and their surrounding context.

2.1.6 A Novel Deep Learning Approach for Deepfake Image Detection

The paper [35] addresses the rising threat of deepfake technology in cybercrimes, introducing a novel deepfake predictor (DFP) approach based on a hybrid of VGG16 and convolutional neural network architecture. Utilizing a deepfake dataset from Kaggle, comprising 1081 real and 960 fake face images, they developed the DFP model. This approach combines transfer learning techniques and deep learning-based neural network architecture, incorporating sequential input layers, VGG16 network layers, convolutional layers, max-pooling layers, dropout layers, flatten layers, and dense layers with specific activation functions. Hyperparameter tuning was conducted to optimize performance accuracy, and various metrics including loss, accuracy, precision, F1 score, specificity, and geometric mean were employed for evaluation. The DFP approach outperformed transfer learning techniques and other state-of-the-art studies as given in Figure 2.1, demonstrating superior performance in deepfake detection. Overall, the research highlights the effectiveness of the proposed methodology in addressing the pressing need for advanced deepfake detection methods.

Technique	Accuracy Score (%)	Loss Score	Precision Score (%)	F1 Score (%)	Specificity Score (%)	Geometric Mean Score (%)
NAS-Net	83	0.8	80	86	73	86
Xception	84	0.5	82	86	75	87
Mobile Net	88	0.4	86	89	84	88
VGG16	90	0.4	88	92	84	91
Proposed DFP	**94**	**0.2**	**95**	**94**	**94**	**94**

Figure 2.1: The comparative performance analysis of employed neural network techniques on unseen test data

2.2 Summary

The summary in Table 2.1 contains a concise overview of the key insights and findings gathered from the extensive literature survey conducted in this chapter.

Table 2.1: Summary of Deepfake Detection Methods

Methodology	**Dataset**	**Key Findings**
(12)Deep fake detection and classification using error-level analysis and deep learning.	Publicly accessible dataset compiled by Yonsei University's Computational Intel-ligence and Photography Lab.	The highest accuracy achieved by the proposed method is 89.5% using Residual Network (ResNet) and K-Nearest Neighbors (KNN).
(33)The eye blinking research paper involves proposing and developing a method called DeepVision to analyze significant changes in eye blinking patterns for detecting Deepfakes generated using the GANs model.	FaceForensics++	The accuracy of video level analysis is 87.5%.
		Continued on next page

Table 2.1 – continued from previous page

Methodology	Dataset	Key Findings
(34)The methodology used in the paper is based on the utilization of capsule networks to detect forged images and videos, including various types of spoofs such as replay attacks and computer-generated videos.	FaceForensics dataset, Deepfake Dataset	The accuracy of video level analysis is 83.33%.
(36)The methodology involves training a Siamese Neural Network using triplet loss on a ResNet18-based CNN to distinguish between real and DeepFake images, fine-tuning with regularized cross-entropy loss on an imbalanced dataset extracted from the DFDC.	DeepFake Detection Challenge (DFDC)	This model is capable of achieving high training and validation accuracy values of 94% and 91% respectively.

Continued on next page

Table 2.1 – continued from previous page

Methodology	Dataset	Key Findings
(35)This paper introduces a deep-fake predictor (DFP) approach based on a hybrid of VGG16 and convolutional neural network architecture.	Dataset taken from Kaggle, comprising 1081 real and 960 fake face images.	Among the comparison of models, proposed DFP achieves highest accuracy by 94% on unseen test data.
(11)Ensemble of 3 models (ConvNet, LeNet, EfficientNet) using the max voting method.	MIO TCD dataset	The ensemble accuracy - 92.77%. The proposed approach demonstrates promising results for real-world applications in traffic surveillance and management.
(12)Image preprocessing by resizing and Error Level Analysis, deep feature extraction using Convolutional Neural Networks for accurate classification.	Publicly accessible dataset compiled by Yonsei University's Computational Intelligence and Photography Lab	Highest accuracy of 89.5% via ResNet18 and KNN. Focused on developing algorithms to detect deep fakes in digital media, utilizing ML and DL techniques. Proposed a method for detecting and classifying deep fakes using error-level analysis (ELA) combined with deep learning techniques.

Continued on next page

Table 2.1 – continued from previous page

Methodology	Dataset	Key Findings
(14)Multi-attentional deepfake detection.	Celeb-DF dataset, FF++ dataset	AUC with Celeb-DF dataset is 67%. Demonstrated improved performance compared to traditional CNN-based methods by focusing attention on relevant facial regions. Showcased robustness to various manipulation techniques and environmental conditions.
(20)Dynamic face augmentation techniques.	Celeb-DF, Deepfake Detection Challenge Dataset, Face-forensic++	EffNet-B4 Face-Cutout - 95.44 (AUC%), Xception Face-Cutout - 95.66 (AUC%). Investigated the effectiveness of dynamic face augmentation, achieving a detection accuracy improvement of 10% on synthetic deepfakes. Dynamically altered facial features to enhance model generalization and robustness to unseen manipulation techniques.
(22)Discrepancy-based detection.	FF++, DFDC, Celeb-DF, DF-1.0	Accuracy: FF++ - 92.11%, DFDC - 65.76%, Celeb-DF - 63.27%, DF-1.0 - 62.46%. Improved the model's ability to discern subtle manipulation cues, enabling robust deepfake detection across various content types.

Continued on next page

Table 2.1 – continued from previous page

Methodology	Dataset	Key Findings
(24)Deepfake detection using DAG-FDD and DAW-FDD.	Celeb-DF, DFDC, DFD	Accuracy of Resnet-50 using DAG-FDD - 92.15%. Accuracy of Resnet-50 using DAW-FDD - 90.58%. The evaluation showed significant improvements in fairness metrics while maintaining detection accuracy.

The research papers collectively address the pressing challenge posed by deepfake technology, emphasizing the critical need for robust detection methods to mitigate its potential for spreading misinformation and facilitating cybercrimes. Each paper proposes a unique methodology for deepfake detection, often employing advanced deep learning techniques like convolutional neural networks (CNNs), ResNet18, and capsule networks. These methodologies aim to distinguish between real and fake media by analyzing various features, such as facial characteristics, blinking patterns, and discrepancies between manipulated regions and their contexts. Furthermore, the utilization of diverse datasets, including the DeepFake Detection Challenge (DFDC) and Face Forensics++, ensures the comprehensive training and evaluation of the detection models across different manipulation techniques and scenarios.

Evaluation of the proposed methods is conducted using standard metrics such as accuracy, precision, recall, F1 score, specificity, and geometric mean score. This rigorous evaluation provides insights into the effectiveness of the detection models in accurately identifying deepfake media. Some papers also explore the use of transfer learning techniques, such as fine-tuning pre-trained models like VGG16, Xception, and MobileNet, to enhance detection capabilities. Additionally, hybrid architectures combining different deep learning techniques and feature fusion approaches are highlighted for their promising results in improving detection accuracy.

Overall, the key learnings from these research papers underscore the importance of

innovative methodologies, comprehensive evaluation, and dataset diversity in advancing deepfake detection. By addressing research gaps and leveraging hybrid architectures and feature fusion approaches, the effectiveness and resilience of detection systems can be enhanced, ultimately contributing to the mitigation of the detrimental impacts of deepfake technology on society.

2.3 Software/Tools Requirements

2.3.1 Development Environment

Visual Studio Code is the recommended Integrated Development Environment (IDE) owing to its wide market adoption, a considerable amount of extensibility for customization, and a robust platform for Python proliferation.

Primary

- Google Colab
- Visual Studio Code

Core Libraries:

- TensorFlow (>= 2.16.1)
- Keras (>= 3.2.1)
- scikit-learn (>= 0.13.0)
- LIME

Optional Libraries: The system could use some more libraries to get those special functionalities granted that it is subject to further development. These can be installed additionally after the build phase but they must be interoperable.

2.3.2 Hardware Requirements

Processor

A GPU of higher power is recommended for efficient training, especially when dealing with a large collection of 140K images.

GPU

Please make use of two Nvidia Tesla T4 GPUs when training on big data as the training time will be significantly decreased.

Memory (RAM)

A minimum of 16GB RAM is recommended, starting at least 32GB or even more to avoid any hiccups, especially when dealing with large datasets and deep learning models.

Storage

Sufficient storage capacity is necessary to keep the training dataset (140k images) and probably pre-trained models that will be used. The particular filesystem attributes will vary depending on image resolution, format, etc.

Chapter 3

PROPOSED SYSTEM

The proposed work titled "Deepfake Detection Using Deep Learning Techniques" aims to address the escalating threat posed by deepfake technology by employing a multi-model deep learning approach. The methodology involves the utilization of diverse convolutional neural network (CNN) architectures tailored for image classification tasks. Specifically, RegularizedConvNet, VGG, RegularizedConvDenseNet, and Xception are employed, each with distinct configurations and regularization techniques. Data preprocessing involves resizing and normalization of images. Additionally, an ensemble method is implemented to amalgamate predictions from the individual models, enhancing the robustness and reliability of the detection system. By leveraging the strengths of multiple models and combining their predictions, the proposed methodology seeks to improve the accuracy and efficacy of deepfake detection, thus contributing to the ongoing efforts to combat the proliferation of misinformation and safeguard digital authenticity. The ensemble model stands out as the cornerstone of the proposed methodology for deepfake detection. While individual models like RegularizedConvNet, VGG, RegularizedConvDenseNet, and Xception each bring their own strengths and specialization to the table, the ensemble model capitalizes on the collective intelligence of these models to achieve superior detection performance.

3.1 Dataset Analysis

This study utilized the 140k Real and Fake Faces dataset, a compilation of 70,000 real faces from the Flickr dataset collected by Nvidia, and 70,000 fake faces sampled from the 1 Million FAKE faces generated by StyleGAN, as provided by Bojan. The dataset was aggregated by combining both sources and resizing all images to 256px. Additionally, the data was partitioned into training, validation, and test sets, with accompanying CSV files for convenience.

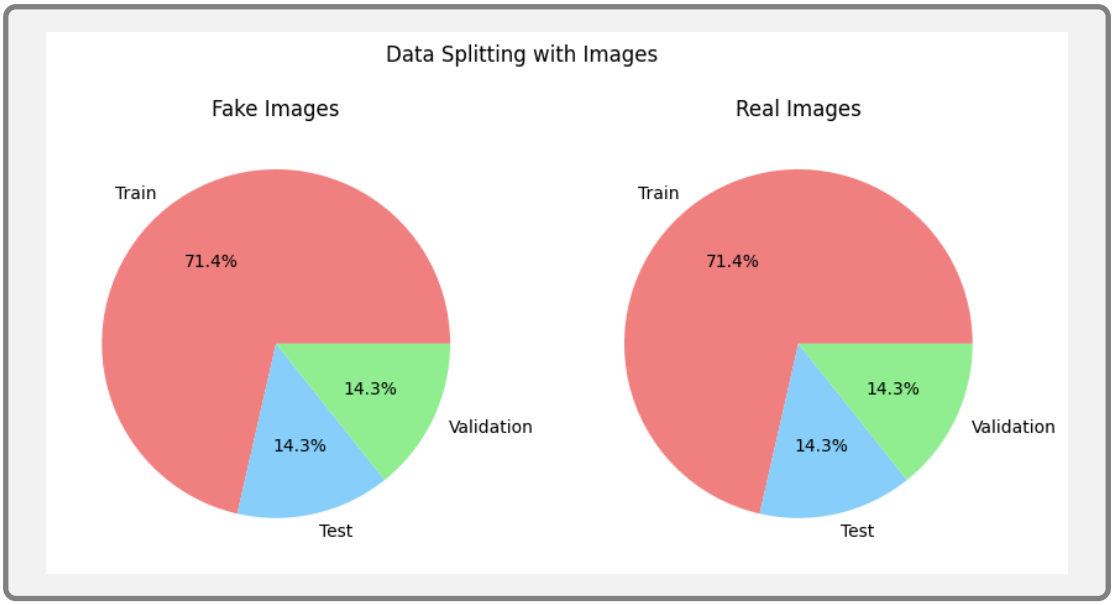

Figure 3.1: *Dataset splitting*

3.2 Methodology

3.2.1 Module 1: Model Training and Preprocessing

The process of training a deepfake detection model involves several crucial steps, including data preprocessing and model training. Data preprocessing plays a vital role in preparing the input data for training the deepfake detection model. Utilizing the ImageDataGenerator class from the Keras library to preprocess the data provides a range of powerful functionalities for data augmentation and normalization. By applying appropriate preprocessing techniques, the model's ability to generalize and improve its performance can be enhanced.

During the preprocessing stage, the input images are resized to a consistent shape to ensure compatibility with the deepfake detection model architecture. In our case, Resizing of is the images is done resolution of 128x128 pixels. Resizing the images not only standardizes the input dimensions but also helps reduce the computational complexity during training.

Furthermore, it is crucial to normalize the pixel values of the images to a common range for optimal training. By rescaling the pixel values to the range of [0, 1], ensures that the model can effectively learn from the data without being biased by variations in

pixel intensity.

Once the data has been preprocessed, data generators are created using the ImageDataGenerator class. These generators allow us to efficiently load the preprocessed data in batches during the model training process. The data generators provide an automatic and dynamic way of augmenting and preprocessing the images while feeding them to the model. This approach helps mitigate overfitting and enables the model to generalize well to unseen data.

The training data generator flows the training images in batches from the designated training folder. The test and validation data generators flow the respective images in batches from their respective folders. By segregating the data into separate sets, the model's performance can be evaluated on unseen data during testing and fine-tuned its parameters using the validation set.

The data generators enable efficient handling of large datasets by loading a batch of images into memory at a time, reducing memory requirements. Additionally, the generators can perform real-time data augmentation techniques such as random rotations, flips, and zooms, enriching the dataset and improving the model's ability to generalize.

3.2.2 Module 2: Implementation of base models

3.2.2.1 VGG16

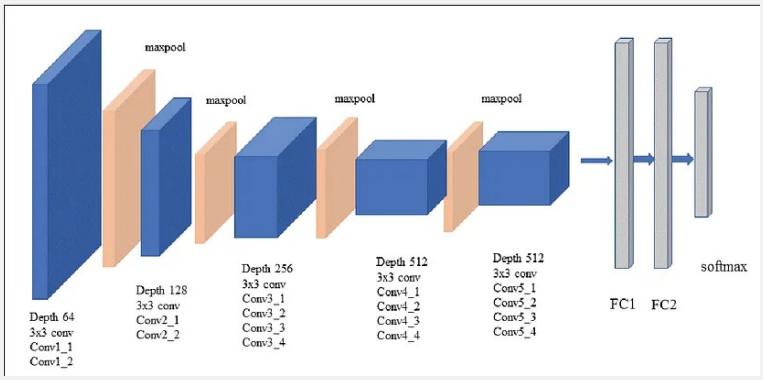

Figure 3.2: VGG16

Employing transfer learning using the VGG16 model pre-trained on ImageNet, the model was customized according to the specific requirements.

Base Model Modification:

The top layer is exclude in the VGG16 model (include_top=False) to enable customization for binary classification. The input shape was set to (128, 128, 3), indicating that the model accepts images of 128x128 pixels with 3 color channels (RGB). The base, comprising VGG16 convolutional layers, was set to non-trainable (trainable=False) to utilize the pre-learned weights effectively without altering them.

Custom Layers Addition:

A Flatten layer to transform the 2D feature maps into a 1D vector,is necessary for feeding into subsequent dense layers. Two Dense layers were added: the first with 256 units utilizing ReLU activation for non-linearity, and the second as a single-node output layer using the sigmoid activation function, suitable for binary classification.

Model Compilation:

The model was compiled using the Adam optimizer, chosen for its efficiency in handling sparse gradients on noisy problems. For the binary classification task, the binary cross-entropy loss function is utilized. The accuracy metric was chosen to evaluate the model's performance during both training and testing.

Model Summary:

The `model.summary()` method provides an overview of the architecture, essential for verifying the network's layout. It consists of 18 layers from VGG16 (including convolutional and pooling layers) and 3 additional custom layers (flattened and two Dense layers).

3.2.2.2 Xception:

Library Import and Base Model Setup:

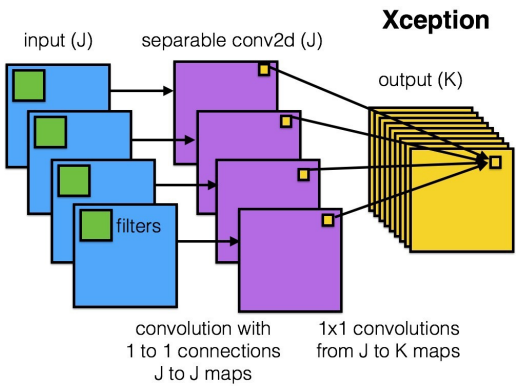

Figure 3.3: Xception

The code starts by importing necessary libraries from TensorFlow Keras, including Xception for loading the pre-trained model and Dense for creating custom layers. The base model is initialized by loading the pre-trained Xception model with weights trained on the ImageNet dataset. The top layer is excluded to allow for further customization.

Freezing Base Model Layers:

Following the setup of the base model, the layers of the base model are frozen to prevent them from being updated during training. This ensures that only the custom layers added later will be trained.

Custom Layers Addition:

Custom layers are added after the base model's output. A Global Average Pooling layer is applied to reduce the spatial dimensions of the output. Then, a dense layer with 128 units and a ReLU activation function is added for non-linearity. A dropout layer with a dropout rate of 0.5 is added to prevent overfitting. Finally, a dense layer with 1 unit and a sigmoid activation function is added as the output layer for binary classification.

Model Compilation:

The model is compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric for training.

Callbacks:

Three callbacks are defined to monitor the training process and adjust the learning rate accordingly:

1. **EarlyStopping:** This callback monitors the validation loss and stops training when the loss stops improving for a specified number of epochs.

2. **ReduceLROnPlateau:** This callback monitors the validation loss and reduces the learning rate by a factor of 0.2 when the loss plateaus.

3. **ModelCheckpoint:** This callback saves the best model based on validation accuracy during training.

3.2.2.3 RegularizedConvNet

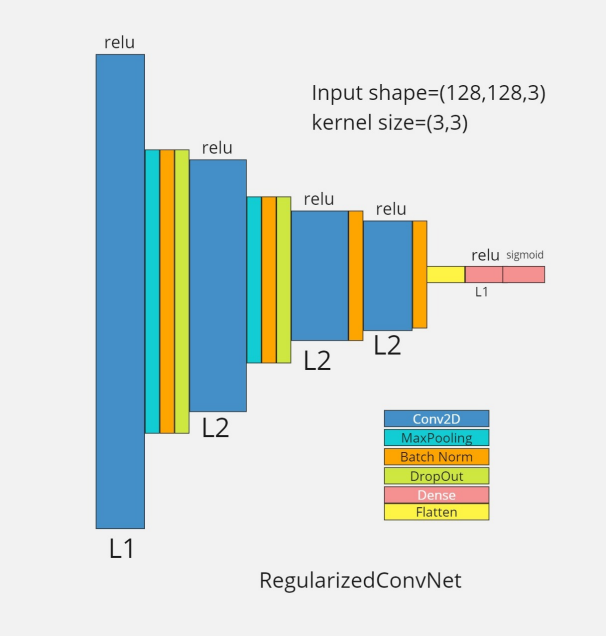

Figure 3.4: RegularizedConvNet

Importing Libraries: The first section of the code imports necessary libraries including TensorFlow, Keras, and additional libraries for regularization.

Defining Input Shape: The input shape for the model is specified as a 3-channel RGB image of size 128x128x3.

Model Architecture:

The model is defined using the Sequential model class from Keras. It consists of several convolutional and pooling layers followed by some fully connected layers.

Convolutional Blocks:

First Convolutional Block: A convolutional layer with a kernel size of 3x3, ReLU activation function, and L2 regularization with a strength of 0.001 is used. The output of this layer is passed through a max pooling layer with a pool size of 2x2 to reduce spatial dimensions. A batch normalization layer is applied to normalize the activations of the previous layer.

Second Convolutional Block: Another convolutional layer with a kernel size of 3x3, ReLU activation function, and L2 regularization with a strength of 0.001 is used. The output of this layer is again passed through a max pooling layer with a pool size of 2x2. Another batch normalization layer is applied to normalize the activations.

Third Convolutional Block: Another convolutional layer with a kernel size of 3x3, ReLU activation function, and L2 regularization with a strength of 0.001 is used. The output of this layer is passed through a batch normalization layer. Finally, a flattened layer is used to flatten the output into a 1D array to prepare it for the fully connected layers.

Layers for the RegularizedConvNet

There are 13 layers in the model:

1. Input layer (shape=(128, 128, 3)) 2. Convolutional layer (32 filters, 3x3 kernel, stride 1, padding 1, activation "relu", kernel regularization=L2(0.001)) 3. Max pooling layer (pool size=2, stride=2, padding=0) 4. Batch normalization layer 5. Dropout layer (rate=0.3) 6. Convolutional layer (64 filters, 3x3 kernel, stride 1, padding 1, activation "relu", kernel regularization=L2(0.001)) 7. Max pooling layer (pool size=2, stride=2, padding=0) 8. Batch normalization layer 9. Dropout layer (rate=0.25) 10. Convolutional layer (128 filters, 3x3 kernel, stride 1, padding 1, activation "relu", kernel regularization=L2(0.001)) 11. Batch normalization layer 12. Dropout layer (rate=0.25) 13. Fully connected layer (128 units, activation "relu", kernel regularization=L1(0.01))

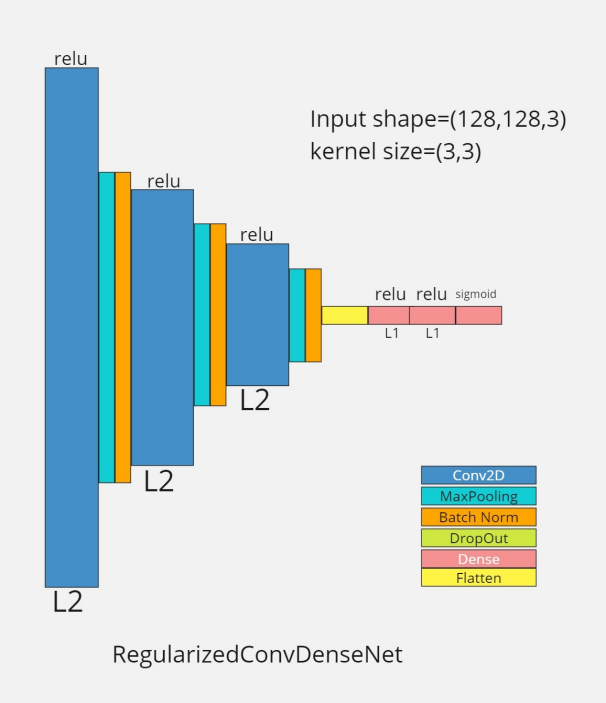

Figure 3.5: RegularizedConvDenseNet

3.2.2.4 RegularizedConvDenseNet:

Library Import and Input Definition:

The code begins by importing necessary libraries such as TensorFlow, Keras, and modules for image preprocessing. It then defines the input shape of the model, which is an RGB image of size 128x128x3.

Convolutional Layers Setup:

The convolutional layers are defined with L2 regularization to prevent overfitting. The first layer has 64 filters, a 3x3 kernel size, ReLU activation, and L2 regularization with a strength of 0.001. Similarly, the second convolutional layer has 64 filters, a 3x3 kernel size, ReLU activation, and L2 regularization with the same strength.

Pooling and Normalization:

Max pooling layers with a pool size of 2x2 are applied after each pair of convolutional layers to reduce spatial dimensions. Batch normalization layers are added after the first and second convolutional layers to stabilize and accelerate the training process.

Flattening and Dense Layers:

The output of the convolutional and pooling layers is flattened into a 1D array using the flattening layer. Then, two dense layers with ReLU activation and L1 regularization are added. The first dense layer has 128 units, and the second dense layer has 64 units, both with L1 regularization strength of 0.001.

Output Layer:

The output layer consists of a single unit with a sigmoid activation function for binary classification.

Model Compilation:

The model is compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric.

3.2.3 Module 3: Implementation of Ensemble of Regularized CNNs

To enhance the performance and robustness of our deepfake detection system, Ensemble model is developed that combines the predictions from multiple individual models. Ensemble models have proven to be effective in improving accuracy and generalization by leveraging the strengths of different models. In the ensemble approach, two deep learning models, RegularizedConvNet and RegularizedConvDenseNet, are selected and their complementary characteristics are leveraged to achieve superior detection capabilities.

RegularizedConvNet RegularizedConvNet is a convolutional neural network (CNN) architecture specifically designed for deepfake detection. It consists of multiple convolutional blocks, incorporating regularization techniques such as L2 regularization, batch normalization, and dropout layers. The model takes input images of size 128x128x3 RGB and aims to learn discriminative features for distinguishing between genuine and manipulated faces. RegularizedConvNet is compiled using the Adam optimizer, binary cross-entropy loss function, and accuracy as the evaluation metric.

RegularizedConvDenseNet RegularizedConvDenseNet is another CNN architecture

tailored for deepfake detection. It comprises convolutional layers with L2 regularization, max-pooling layers, batch normalization layers, and a combination of dense layers with L1 regularization. The model takes input images of size 128x128x3 RGB and aims to capture relevant features for distinguishing between genuine and manipulated faces. Similar to RegularizedConvNet, RegularizedConvDenseNet is compiled using the Adam optimizer, binary cross-entropy loss function, and accuracy as the evaluation metric.

Ensemble Prediction

In the ensemble model, predictions are obtained from both RegularizedConvNet and RegularizedConvDenseNet and averaged to produce the final prediction. By taking into account the collective insights of both models, the aim is to improve the reliability and performance of deepfake detection. This ensemble prediction technique allows us to make more informed decisions and effectively identify manipulated videos or images. By combining the predictions of RegularizedConvNet and RegularizedConvDenseNet through ensemble averaging, the complementary strengths of these models can be leveraged to achieve improved performance in deepfake detection. The ensemble model serves as a powerful tool in combating deepfake manipulation and enhancing the overall security and trustworthiness of multimedia content.

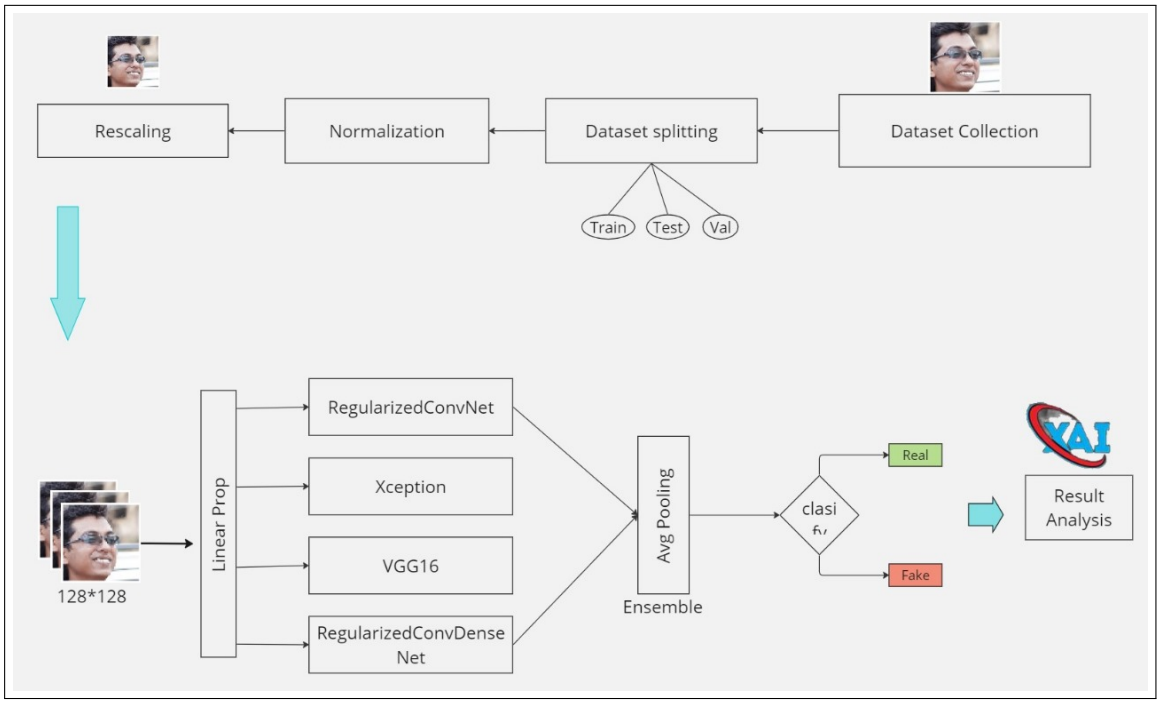

Figure 3.6: *Architecture Diagram for the Methodology*

Chapter 4

IMPLEMENTATION AND TESTING

This chapter contains a detailed account of the practical implementation of the methodologies and frameworks outlined in the preceding chapters. The cutting edge of our deepfake detection system is in a robust ensemble learning technique that it uses. This strategy leverages the complementary strengths of two powerful convolutional neural network (CNN) architectures: RegularizedConvNet and RegularizedConvDenseNet. Image-wise model analysis with each model gives out individual deepfake probability scores ($P(\text{deepfake})$) in the range of 0 (fake) to 1 (real).

The ensemble prediction is calculated using the following formula:

$$\text{Ensemble Prediction} = \frac{P(\text{deepfake\_RegularizedConvNet}) + P(\text{deepfake\_RegularizedConvDenseNet})}{2}. \tag{4.1}$$

Thus, this group method leads to a higher degree of generality and the correct classification accuracy in a broader variety of deepfake as the ensemble approach of different deepfake models reduces the error of choices and prevents the overfitting of the decisions.

4.1 Testing

The system outputs the prediction score of image Figure 4.1

Predicted Likelihood/Probability: Ensemble Net generates a probability score for every image before the output. This score usually varies between a range of 0-1 as greater values closer to 1 indicate a higher chance of this being a deepfake.

Actual Label: Since this is the testing period having labeled data and that actual label associated with each image ie. genuine or deepfake is known.

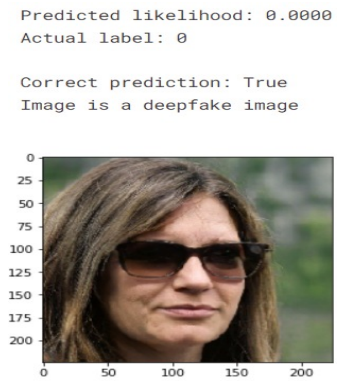

Figure 4.1: Testing

Correct Prediction: Through the comparison of the predicted probability with the preset threshold (e.g., 0.5), a system then decides whether an outcome is correct or wrong, i.e., it matches or does not match the actual label.

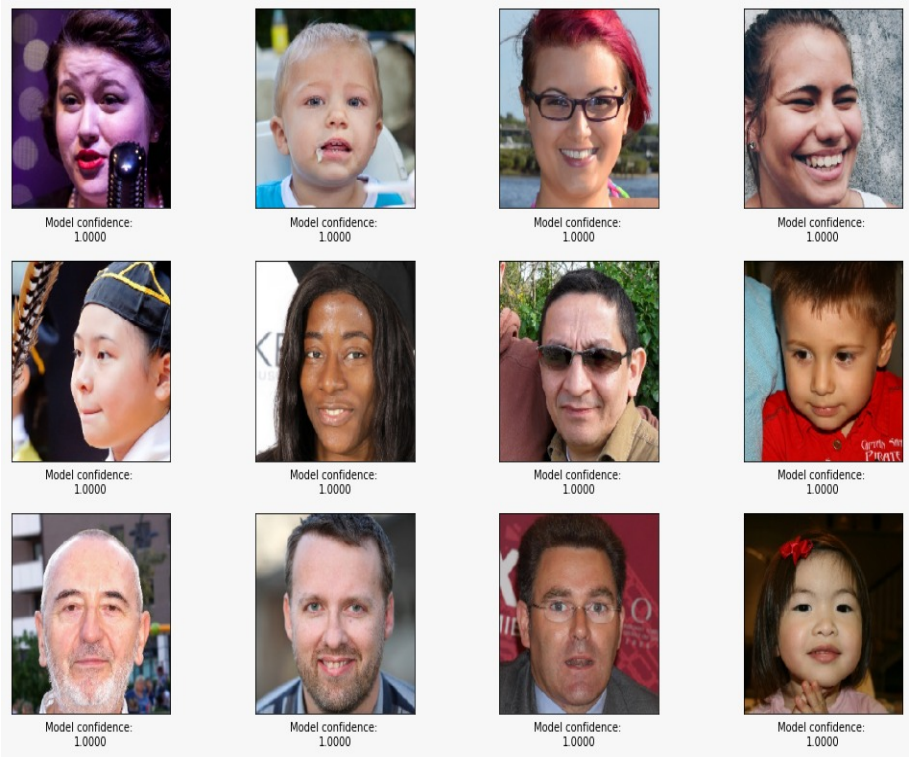

Figure 4.2: Real Images Classified Correctly

Some Real images with Label[1] are passed through the ensemble model. The system outputs the probability score of the real images where the classification is perfectly correct(score=1.000) referring to the same in Figure 4.2

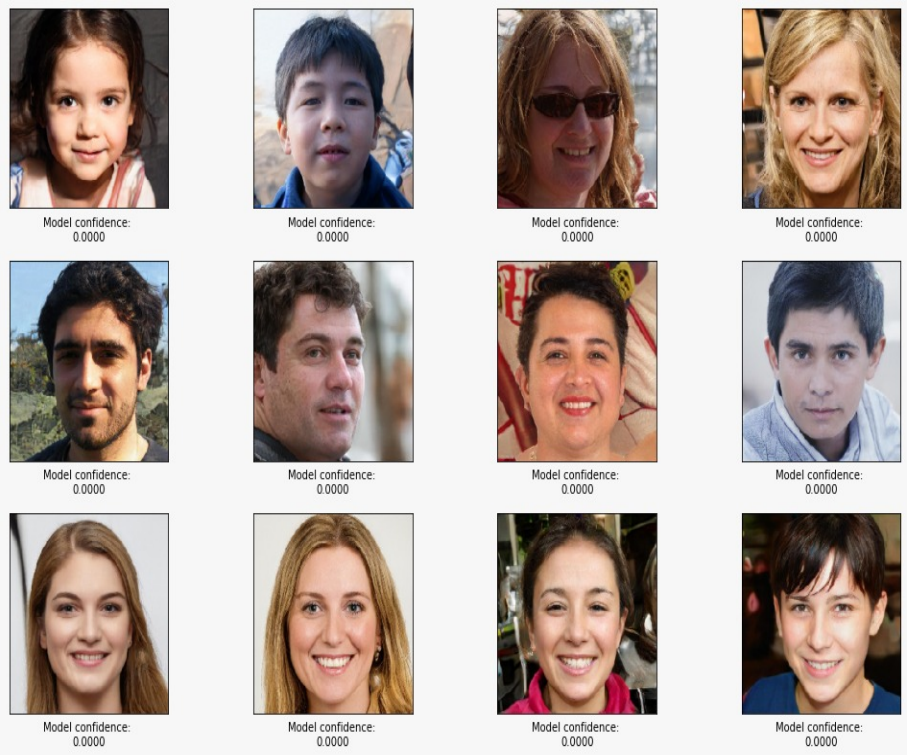

Figure 4.3: Fake Images Classified Correctly

Figure 4.4: Misclassified Images

Similarly, some fake images are passed through the ensemble model where the fake classification is also perfectly good with a score of 0, the same can be seen in Figure 4.3. The model fails for some images as shown in Figure 4.4; some fake images of the label[0] are classified as real with a probability score (>0.5). Model failure for the cases may be due to unclear data like the face is wrapping with background or unevenness of light e.t.c

Chapter 5

RESULTS AND DISCUSSION

In this chapter, the findings and analysis derived from the model outputs are discussed, offering insights and interpretations into the results obtained.

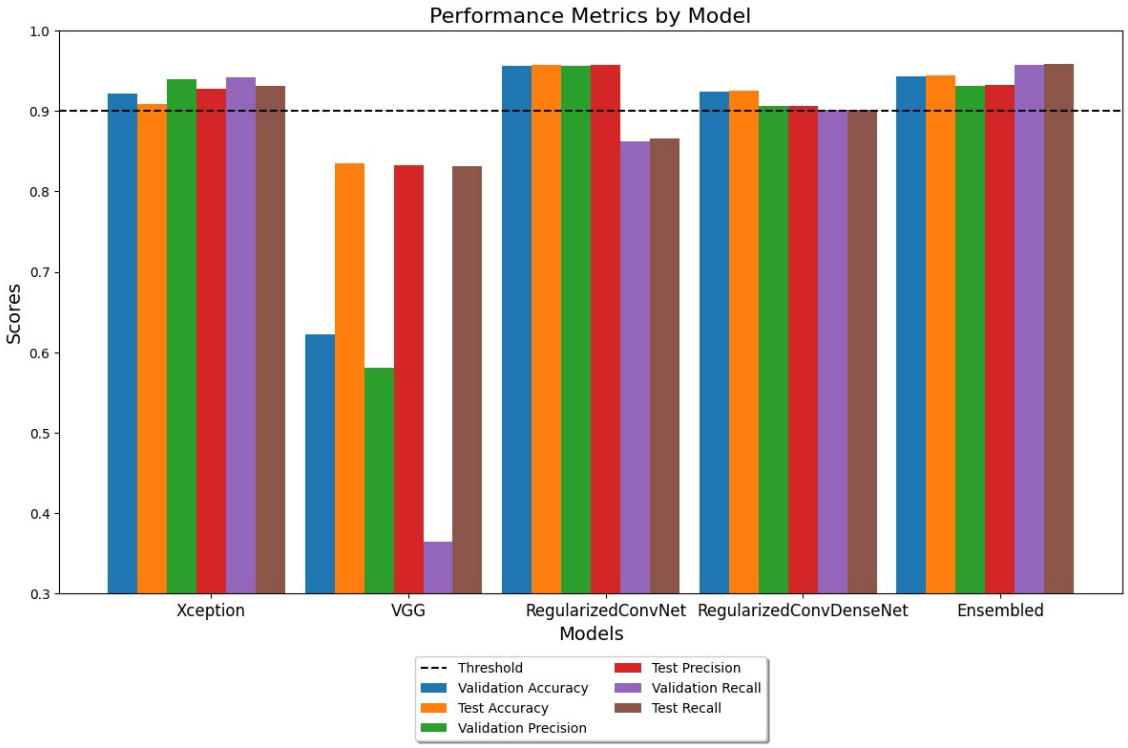

Figure 5.1: Performance Metrics

Shown is the bar graph 5.1 of the performance of several models related to the metrics groups. The research will thus disassemble the strong and weak aspects of the networks, with a specifying target of identifying a triumph of an ensemble method over regularized convolutional models.

In classification tasks, accuracy and precision are two main metrics used to evaluate the performance of a model. Accuracy measures the overall correctness of the model's predictions, while precision measures the proportion of true positive predictions out of all positive predictions made by the Ensemble model.

5.1 Confusion matrix

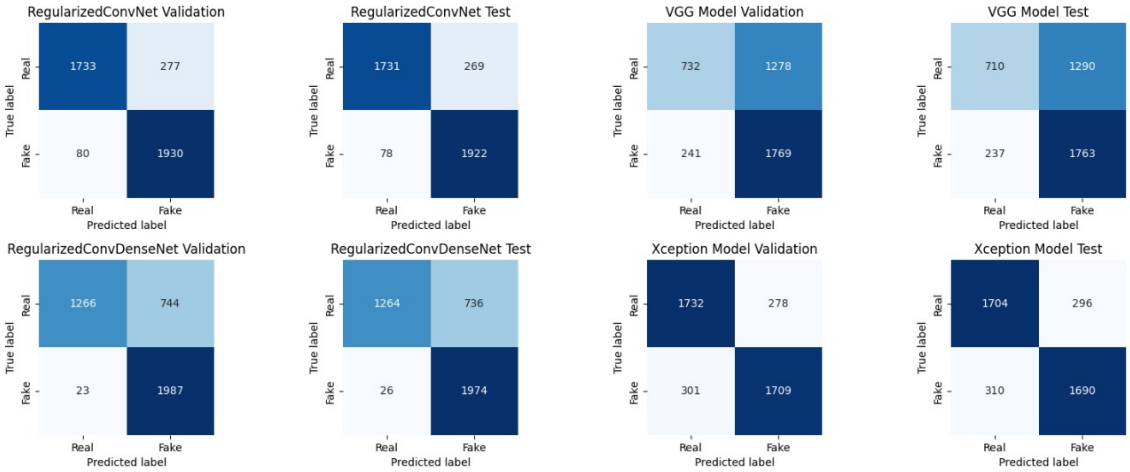

Figure 5.2: *Confusion metrics of RegularizedConvNet;RegularizedConvDenseNet;VGG;Xception model*

The Confusion metrics in Figure 5.2 make it clear that the VGG model outperforms the others. When examining the RegularizedConvNet confusion matrix, all True positives and False Negatives are fewer, indicating that the model performs well for all classifications. Comparing RegularizedConvDenseNet and Xception, Xception classifies True Negatives well compared to RegularizedConvDenseNet, but for False Negatives, Xception outperforms. Both True Positives and False Negatives are considered in the ensemble model. So, by ensemble averaging RegularizedConvNet and RegularizedConvDenseNet, normalization of True Negatives and False Positives will be done, resulting in better numbers.

5.2 Accuracy

Accuracy (Acc) is defined as the ratio of the number of correctly predicted instances to the total number of instances in the dataset:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5.1}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

5.3 Precision

Precision (P) is defined as the ratio of true positives to the sum of true positives and false positives:

$$P = \frac{TP}{TP + FP}. \tag{5.2}$$

Through the investigation of the metrics, the Xception model emerges as the best with high and improved quality of validation accuracy (0.92175) and test precision. Nevertheless, the test accuracy of this method is declining somewhat compared to the more accurate ensemble model. Contrarily, VGG models exact the weakest performance in the majority of metrics with the test accuracy being highly hit by accuracy (0.835). Therefore, the VGG model can be unstable regarding unseen data.

These key results show that the integrated model, while combing the outcomes of RegularizedConvNet and RegularizedConvDensinet, eventually turns out a top performer. It shows that Ensemble performs better than Individual models with 0.94475 accuracy, 0.9324 precision, and 0.959 recall in the test results. Because it performs better in validation accuracy than RegularizedConvNet, it goes ahead to exceed the other models in validation recall (0.957). The achievement highlights the diverse model's ensemble model competencies combining their strong sides. For achieving promising learning outcomes and facilitating meaningful cognitive improvements within a shorter timeframe, these new procedures hold considerable significance.

The ensemble approach, utilizing the strengths of different models, creates a more meaningful predictor. Therefore, this approach offers considerable advantages, which is one of the reasons why the ensemble model outperforms other methods. The ensemble approach involves obtaining predictions from each individual model and then averaging the errors produced by the models. This allows the model to achieve more reliable and accurate overall predictions, increasing the effectiveness of the models and eliminating the variance factor, a source of instability in models. Therefore, what ultimately matters is that if the individual models have different forms of biases, averaging their predictions helps to not only avoid these biases but also rely solely on the ensemble model, meaning the final results won't be influenced by the biases either. This technique is

referred to as "Bias Reduction". As a consequence, there is an enhancement of generalization, resulting in the model being less likely to overfit to the training data and performing well on unseen data.

Models	Metrics					
	Val. Accuracy	Test Acc.	Val. Precision	Test Precision	Val. Recall	Test Recall
Xception	0.92175	0.908	0.93909	0.92812	0.9415	0.9315
VGG	0.62214	0.835	0.58057	0.83267	0.36418	0.8315
RCN	0.95587	0.95688	0.95587	0.95688	0.86219	0.8655
RCDN	0.92425	0.92475	0.90579	0.90626	0.9015	0.902
Ensembled	0.943	0.94475	0.93093	0.93242	0.957	0.959

Table 5.1: *Model Metrics ;RCN(RegularizedConvNet) and RCDN(RegularizedConvNet)*

5.4 ROC Curve and AUC

In classification tasks, the Receiver Operating Characteristic (ROC) and Area under the ROC Curve (AUC) are important metrics used to evaluate the performance of a model in binary classification. The ROC curve is a graphical representation of the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. TPR is also known as sensitivity or recall, and FPR is the complement of specificity. The ROC curve shows the trade-off between sensitivity and specificity. AUC represents the area under the ROC curve. It provides an aggregate measure of performance across all possible classification thresholds. AUC ranges from 0 to 1, where a higher AUC value indicates better performance of the model.

Let's denote the true positive rate (TPR) as

$$TPR = \frac{TP}{TP + FN} \tag{5.3}$$

and the false positive rate (FPR) as

$$FPR = \frac{FP}{FP + TN} \tag{5.4}$$

where TP is the number of true positives, FN is the number of false negatives, FP is

the number of false positives, and TN is the number of true negatives.

The ROC curve is plotted using TPR (sensitivity) and FPR (1 - specificity) values for different threshold settings. AUC is calculated as the area under the ROC curve.

Let's denote the true positive rate (TPR) as

$$TPR = \frac{TP}{TP + FN} \tag{5.5}$$

and the false positive rate (FPR) as

$$FPR = \frac{FP}{FP + TN} \tag{5.6}$$

where TP is the number of true positives, FN is the number of false negatives, FP is the number of false positives, and TN is the number of true negatives.

The ROC curve is plotted using TPR (sensitivity) and FPR (1 - specificity) values for different threshold settings. AUC is calculated as the area under the ROC curve.

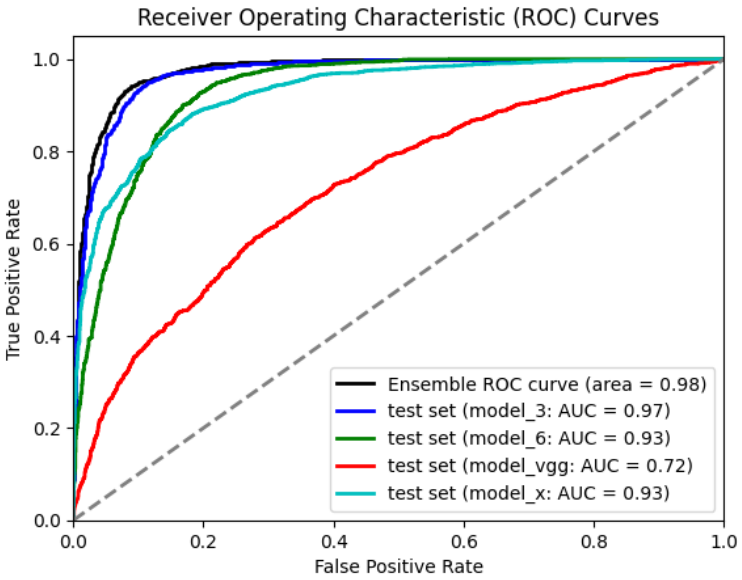

Figure 5.3: ROC Curves:*RegularizedConvNet* (model_3), *RegularizedConvDenseNet* (model_6), *VGG16* (model_vgg), and *Xception* (model_x) models.

Examining the ROC curves shown in Figure 5.3 reveals a clear leader: the mixed method, ensemble model. The AUC now measures 0.98, which says much about its potential in the context of classifying either of the positive or negative classes. As a result, they can classify data points with high precision which is particularly critical in a variety of tasks such as detection of anomalies, fraud detection, security systems,

etc. In the niche of AUC scores, RegularizedConvNet claims the first place (0.97) and RegularizedConvDenseNet is close by with an AUC score of 0.93 which is also a celebrated figure exhibiting promising separation between classes. Xception matches other models in accuracy, with an AUC of 0.93. At second glance, however, perhaps because the separation between classes is not shown to be as significant as the figures of other models, it seems that there are some subtle distinctions. VGG is, representative of the other networks as well, much weaker than GPT and BERT because of its lowest AUC (0.72). ROC curve visualization of VGG shows a pretty bad one with probably the greatest overlapping lines, which indicates its obvious shortcomings in distinguishing between two categories.

The supremacy of the Ensemble model can be found in its cunning approach to the integration of the concatenation of tasks achieved through the additive regularization of the ConvNet and ConvDenseNet. Ensemble learning, which just happens to be a natural method, often results in better outcomes, the overall improvements being synergistic and rather than merely the difference between individual model predictions. This way of solving is usually an efficient and successive failure approach, which may provide a better result probably than the single model solution because it is more robust and reliable. Ensemble methods achieve this enhanced performance through two key mechanisms: Ensemble methods achieve this enhanced performance through two key mechanisms:

Reduced Variance: An ensemble of models reduces the variance in the respective results, moves the mean prediction closer to the actual value, and decreases its jitter, meaning that the prediction becomes smoother when more models are included in the ensemble. Therefore, their combination results in a more consistent and stable model as it cuts down the possible effect of random errors that may affect the internals of separate models.

Reduced Bias: If the base models are biased, their predictions will also be biased, and averaging them in the ensemble model will reduce the overall bias frequency in the final ensemble model. This can lead to a considerable generalization of the model, reducing the risk of the model fitting data too closely and achieving better performance on entirely new unseen data.

5.5 LIME - the Local Interpretable Model-Agnostic Explanations

There are several factors that contribute to the practicality of the ensemble model as an image classification tool. However, it is still helpful to analyze the image classification performance of the ensemble model. While it plays an important role in the development of financial markets, the opacity of its operation, also known as the mystery box issue, should also be considered. There is a potential risk of diminishing trust and utility in AI systems due to the lack of transparent reasoning. The Explainable Artificial Intelligence approach, specifically LIME, is employed to reveal which image features, such as eyes/mouth/cheeks/neck e.t.c regions, the model used.

Figure 5.4: LIME Analysis

LIME turns out to constitute one of the most useful and useful AI explainability tools for evidently expressing a complex ensemble model. Contrary to the conventional technique that comes with some predetermined frameworks, the LIME model provides a model-agnostic explanation that can easily be used in this kind of analysis without being restricted by the ensemble architecture. The LIME approach to explainability

centers around making clear about individual classifications. For an image it creates a neighborhood that is made up of several versions, that can be slightly different, with variations to the eye and mouth areas Figure 5.4 through ways like blurring or masking. Valued LIME then resorts to the ensemble model for predicting class labels to each altered image. LIME provides similar feature sets to the original image that have not been modified by changing the pixels in the eye and mouth areas. These components are then boosted visually, transforming into a form that can be comprehended by the public.

LIME used on classified samples extracted from the model's ensemble solves the case with the foregoing insights. The frequent addressing of eye and mouth offends in the explanations regards them as sources of information that the model relies on to make decisions. This may suggest that the task deals with the analysis of facial expressions or the identification of specific features contained within the aforementioned areas, thus. LIME is more impressive than that as in addition to providing explanations for the model's predictions, it develops a level of trust and understanding of how the model understands the image information. This transparency is important in the field of real-world architecture machine learning model deployment. Furthermore, LIME can be used to identify any bias in the given data that was used for the training. By targeting regularly, the inappropriate features in the eye or mouth regions, LIME could provide useful clues to the possible frames the model has learnt from training. This is necessary to delve deep and have this data cleaned.

Although LIME helps in getting some insights, it is also right to say that it has also some unreliable aspects. LIME translates individual predictions, whereas the latter delivers, though, the overall meaning of the whole model. As well, the generation of explanations for all the images may take a lot of computing resources. On the other hand, LIME is used for this analysis to assess the model's performance plus involve a deeper understanding of the model's decision-making process, normally relating to the eyes and the mouth areas. The AI techniques will give a new understanding of how the model predicts which builds trust and faith in the accuracy of the model, helps to identify biases, and ultimately results in building a more transparent and reliable model for classification tasks.

5.6 Video Frame level Analysis

A complete real video was passed through the xception, vgg16, regularizedConvnet, regularizedConvDenseNet, ensemble model(RegularizedConvnet + RegularizedConvDenseNet).Firstly the video was passed through a face detector to the face alone.Region Of Interest (ROI) is Face. Then the video with the face alone is passed frame by frame to the model. Graphs with frame number vs probability scores are plotted to analyze

Frame-level analysis of videos involves testing the ensemble model strength for the video classification with a frame-level analysis.

5.5 shows the frame number vs Probability scores of the frame for a REAL video.

Observing that all the models (Xception, Vgg, RegularizedConvNet, RegularizedConvDenseNet, and Ensemble) gave probabilities (>0.8), which is a good classification. Additionally, Vgg model gave more probabilities for the frames compared to all other models.

5.5 shows the frame number vs Probability scores of the frame for FAKE video. Observing that all models (Xception, Vgg, RegularizedConvNet, RegularizedConvDenseNet, and Ensemble) partially failed to give the correct classification. It can be observed that up to half of the frame (>350), the classification was done perfectly, but the models failed to classify after frame 350. This is due to the face not being fully detected after those frames. Model Xception has many fluctuations with the classification. Vgg model totally failed with video classification. RegularizedConvNet and Ensemble partially failed with the classification

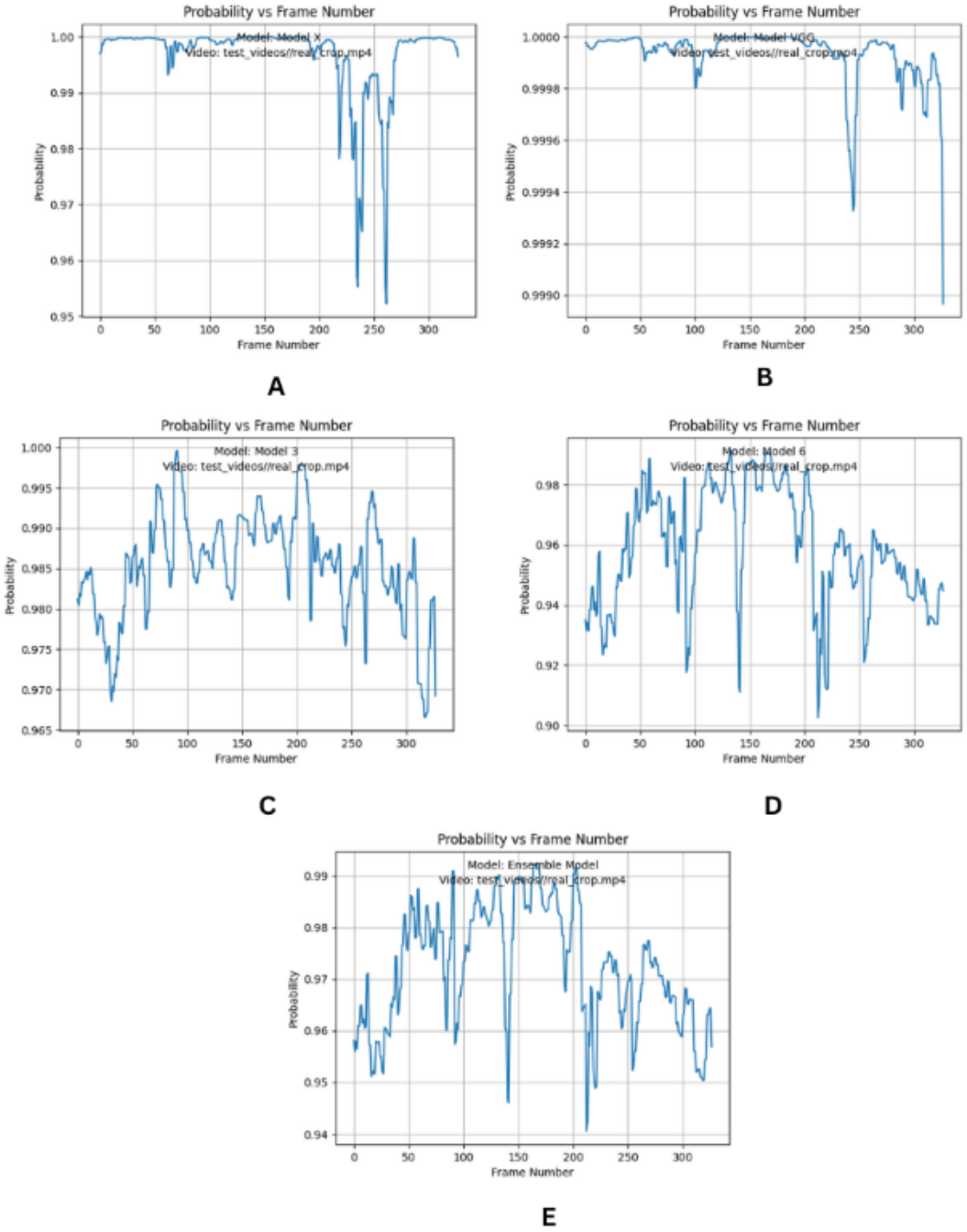

Figure 5.5: Frame level analysis for a REAL video **A**.*Xception* (model_x),**B**.*VGG16* (model_vgg), **C**.*RegularizedConvNet* (model_3), **D**.*RegularizedConvDenseNet* (model_6), and **E**.*Ensemble* models.

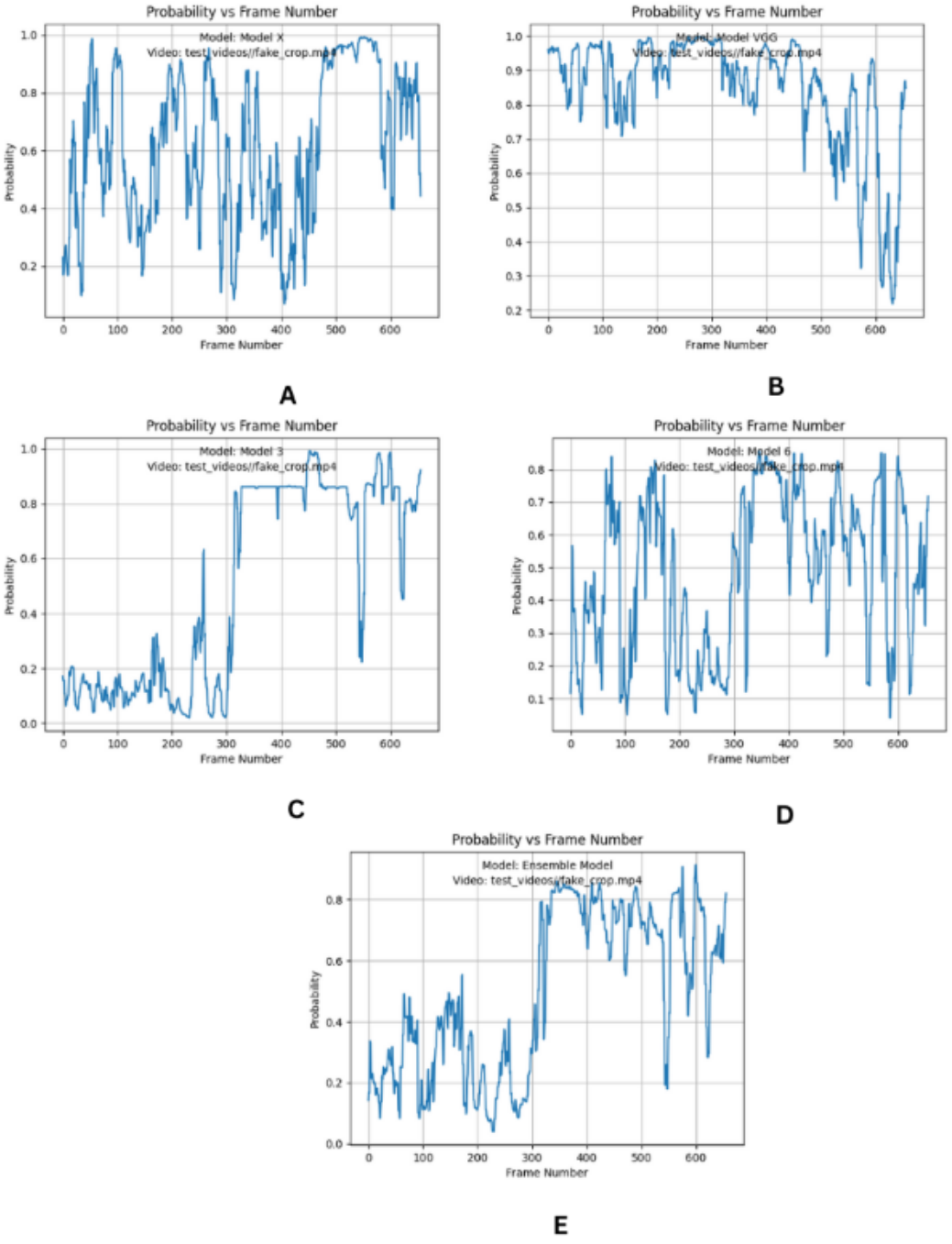

Figure 5.6: Frame level analysis for a FAKE video **A**.*Xception* (model_x),**B**.*VGG16* (model_vgg), **C**.*RegularizedConvNet* (model_3), **D**.*RegularizedConvDenseNet* (model_6), and **E**.*Ensemble* models.

Chapter 6

CONCLUSION

In this chapter, the key findings and implications drawn from the analysis of the model outputs are summarized. The efficacy of ensemble approaches stands out as a crucial factor highlighted by our investigation into the detection of deepfakes using deep learning models. The model ensemble proves to clearly outperform, especially with the AUC value at 0.98. On the other hand, this best-in-class metric of a model greatly improves its capability to distinguish between genuine and deepfake content compared to separate models. Nevertheless, evaluation does not only mean achieving raw performance. Gaining trust with deepfake detection models includes understanding their decision-making capabilities and the models that act behind them. This is where XAI methods (particularly LIME - Local Interpretable Model-Agnostic Explanations) come into focus. Utilizing LIME with the ensemble model will allow us to learn more about which image features vitally affect the decision of the classifier, such as facial features like eyes and mouth. Perhaps, this attention could be a failed attempt to convey the details associated with facial expressions that can be changed in deepfakes. LIME not only allows the interpretation of AI models but goes beyond that. Its clear demonstrations of the model's explanatory power for the team build trust and understanding of why the ensemble model works in this way. Additionally, LIME will reveal biases that exist within the training data, providing the opportunity to fix the nature of the model used for better accuracy. Overall, though deep learning serves as a means of falsification detection, ensembles constitute a strong instrument in deepfake detection. As shown by the ROC curves and the produced results, the ensemble model is effective at separating real and fake media. Also, LIME as an XAI method provides a very distinctive interpretable level, emphasizing transparency and enabling ongoing adjustments of the model. These three factors of high accuracy, explainability, and bias elimination are major contributors to the advancement of accurate and credible deepfake detection systems.

Chapter 7

FUTURE ENHANCEMENT

Apart from ensemble deep learning models as well as LIME, researchers should dig deeper into XAI (explainable AI) techniques such as SHAP and Integrated Gradients to further explore the model decision- processes and exposed general patterns of feature importance levels, which will in turn improve the transparency and dependability of the detection system. Also, training adversarial exposure into the development model will make the system stronger against the attacks of adversaries who are well prepared by the means. In the process of developing deepfake technology, it is of utmost importance to address the issue of audio deepfakes through research aiming at developing detection procedures for multimodal detection techniques. The immediate detection of deepfakes on streaming platforms provides a bunch of different hurdles, that require respective algorithms for quick processing and large-scale architectures. In addition, being fair and transparent with bias reduction in deepfake detection models should also be the primary focus to develop methods that can identify and fix biases within training data including promoting fairness and equity in model prediction across diverse demographic groups. Researchers can pioneer the field of deepfake detection by filling in these gaps in future work on this issue. Because of that, these detection systems will be more logical, reliable, and unbiased, and will be able to follow the alterations of artificially generated identity forgery.

REFERENCES

[1] Zobaed, S.; Rabby, F.; Hossain, I.; Hossain, E.; Hasan, S.; Karim, A.; Hasib, K.M. Deepfakes: Detecting forged and synthetic media content using machine learning. In Artificial Intelligence in Cyber Security: Impact and Implications; Springer: Berlin/Heidelberg, Germany, 2021; pp. 177–201.

[2] Thambawita, V.; Isaksen, J.L.; Hicks, S.A.; Ghouse, J.; Ahlberg, G.; Linneberg, A.; Grarup, N.; Ellervik, C.; Olesen, M.S.; Hansen, T.; et al. DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine. Sci. Rep. 2021, 11, 21869.

[3] Ahmed, M.F.B.; Miah, M.S.U.; Bhowmik, A.; Sulaiman, J.B. Awareness to Deepfake: A resistance mechanism to Deepfake. In Proceedings of the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, 4–5 July 2021; pp. 1–5.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[5] Naik, R. Deepfake Crimes: How Real and Dangerous They Are in 2021? 2021. Available online: https://cooltechzone.com/research/deepfake-crimes.

[6] Bracken, B. Deepfake Attacks Are About to Surge, Experts Warn|Threatpost. 2021. Available online: https://threatpost.com/deepfake-attacks-surge-experts-warn/165798/.

[7] Amin, R., Al Ghamdi, M. A., Almotiri, S. H. Alruily, M. Healthcare techniques through deep learning: Issues, challenges and opportunities. IEEE Access 9, 98523–98541 (2021).

[8] Turek, M.J. Defense Advanced Research Projects Agency. https://www.darpa.mil/program/media-forensics. Media Forensics (MediFor). Vol. 10 (2019).

[9] Schroepfer, Mike. "Creating a data set and a challenge for deepfakes." Facebook artificial intelligence 5 (2019).

[10] Nandan, K.V.P., Panda, M., Veni, S.: Handwritten digit recognition using ensemble learning. In: 2020 5th International Conference on Communi-cation and Electronics Systems (ICCES), pp. 10081013 (2020). https://doi.org/10.1109/ICCES48766.2020.9137933

[11] Asmitha, U., S. Roshan Tushar, V. Sowmya, and K. P. Soman. "Ensemble Deep Learning Models for Vehicle Classification in Motorized Traffic Analysis." InInternational Conference on Innovative Computing and Communications: Proceedings of ICICC 2022, Volume 2, pp. 185-192. Singapore: Springer Nature Singapore, 2022.

[12] Rafique, R., Gantassi, R., Amin, R. et al. Deep fake detection and classification using error-level analysis and deep learning. Sci Rep 13, 7422 (2023). https://doi.org/10.1038/s41598-023-34629-3

[13] Rana, M. S., Nobi, M. N., Murali, B., Sung, A. H. (2022). Deepfake detection: A systematic literature review. IEEE Access, 10, 25494-25513.

[14] Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., Yu, N. (2021). Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2185-2194).

[15] Zhou, Y., Lim, S. N. (2021). Joint audio-visual deepfake detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 14800-14809).

[16] Yu, P., Xia, Z., Fei, J., Lu, Y. (2021). A survey on deepfake video detection. Iet Biometrics, 10(6), 607-624.

[17] Seow, J. W., Lim, M. K., Phan, R. C., Liu, J. K. (2022). A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities. Neurocomputing, 513, 351-371.

[18] Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., Xia, W. (2021). Learning self-consistency for deepfake detection. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 15023-15033).

[19] Kwon, P., You, J., Nam, G., Park, S., Chae, G. (2021). Kodf: A large-scale Korean deepfake detection dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10744-10753).

[20] Das, S., Seferbekov, S., Datta, A., Islam, M. S., Amin, M. R. (2021). Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3776-3785).

[21] Ahmed, S. R. A., Sonuç, E. (2023). Deepfake detection using rationale-augmented convolutional neural network. Applied Nanoscience, 13(2).

[22] Nirkin, Y., Wolf, L., Keller, Y., Hassner, T. (2021). DeepFake detection based on discrepancies between faces and their context. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10), 6111-6121.

[23] Heidari, A., Jafari Navimipour, N., Dag, H., Unal, M. (2023). Deepfake detection using deep learning methods: A systematic and comprehensive review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, e1520.

[24] Ju, Y., Hu, S., Jia, S., Chen, G. H., Lyu, S. (2024). Improving fairness in deepfake detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 4655-4665).

[25] Wang, T., Chow, K. P. (2023, June). Noise-based deepfake detection via multi-head relative-interaction. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 12, pp. 14548-14556).

[26] Wang, T., Cheng, H., Chow, K. P., Nie, L. (2023). Deep convolutional pooling transformer for deepfake detection. ACM Transactions on Multimedia Computing, Communications and Applications, 19(6), 1-20.

[27] Cozzolino, D., Pianese, A., Nießner, M., Verdoliva, L. (2023). Audio-visual person-of-interest deepfake detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 943-952).

[28] Kamat, S., Agarwal, S., Darrell, T., Rohrbach, A. (2023). Revisiting generalizability in deepfake detection: Improving metrics and stabilizing transfer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 426-435).

[29] Li, C., Huang, Z., Paudel, D. P., Wang, Y., Shahbazi, M., Hong, X., Van Gool, L. (2023). A continual deepfake detection benchmark: Dataset, methods, and essentials. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1339-1349).

[30] Heo, Y. J., Yeo, W. H., Kim, B. G. (2023). Deepfake detection algorithm based on improved vision transformer. Applied Intelligence, 53(7), 7512-7527.

[31] Pu, J., Sarwar, Z., Abdullah, S. M., Rehman, A., Kim, Y., Bhattacharya, P., ... Viswanath, B. (2023, May). Deepfake text detection: Limitations and opportunities. In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 1613-1630). IEEE.

[32] Khochare, J., Joshi, C., Yenarkar, B., Suratkar, S., Kazi, F. (2021). A deep learning framework for audio deepfake detection. Arabian Journal for Science and Engineering, 1-12.

[33] T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," in IEEE Access, vol. 8, pp. 83144-83154, 2020, doi: 10.1109/ACCESS.2020.2988660.

[34] H. H. Nguyen, J. Yamagishi and I. Echizen, "Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 2307-2311, doi: 10.1109/ICASSP.2019.8682602.

[35] Raza, Ali, Kashif Munir, and Mubarak Almutairi. 2022. "A Novel Deep Learning Approach for Deepfake Image Detection" Applied Sciences 12, no. 19: 9820. https://doi.org/10.3390/app12199820

[36] Salpekar, Omkar. “DeepFake Image Detection.” (2020)