Optimising Deepfake Detection and Recognition with Deep Learning Techniques

M.Sc. Data Science and AI

Name: Pranjal Malik
Submission Date: 5th August 2024
Supervisor: Dr Muhammad Waqar and Dr Salman Ahmed

This report is submitted in partial fulfilment of the requirement for the degree of **M.Sc. Data Science and Artificial Intelligence** by **Pranjal Malik.**

ABSTRACT

The maturation of artificial intelligence and machine learning models in creating media has delved into the realm of an ever-growing beast we call deepfakes which are synthetic videos or images created using GANs (generative adversarial networks). Though it could create convincing content, this deeply fake technology poses serious threats to privacy, security and the integrity of digital media. If misused, deepfakes can be used in cases of non-consensual porn or misinformation campaigns as well to influence political manipulation. In this study, the effectiveness of using deep learning, especially a type called Convolutional Neural Networks (CNNs) such as ResNet50 and DenseNet to detect Deepfakes was investigated. ResNet50's architecture is great at solving the vanishing gradient problem and when it comes to detecting inconsistencies in deepfake content (some of which are not visible even to a human eye), ResNet-50 proves itself. Leveraging the wealth of dataset available online, this study employs the 140K Real vs Fake dataset, This dataset consists of all 70k REAL faces from the Flickr dataset collected by Nvidia, as well as 70k fake faces sampled from the 1 Million FAKE faces (generated by StyleGAN) that was provided by Bojan. Both Nvidia and Bojan have compiled images from Face Forensics ++ dataset, Celeb-DF, Deepfake Detection Challenge(DFDC). So, overall the dataset this study exploits, is a compilation of the images from the mentioned three dataset and is freely available on kaggle. This study also explores how temporal analysis techniques like LSTM within our detection model can enhance the video frame by capturing dynamic inconsistencies across frames with further augmentation on regional saliency heatmap. It is for evaluating the effects of dataset quality and diversity on detection performance of a few deepfake datasets (FaceForensics++, Celeb-DF, Deepfake Detection Challenge(DFDC)). In this study, the DenseNet receives a higher AUC-ROC of 0.998976 and a high accuracy score from 92% to 96%, and this means that it is very reliable for distinguishing real images from deepfake. However, the wide range of results in validation accuracy shows a risk for overfitting, while the stable validation accuracy of ResNet and an AUC score of 0.98 exhibiting an accuracy of 98% suggests its robust nature in practical scenarios. Ultimately, the research is trying to improve tools for deepfake detection in order to strike a balance between benefits associated with protecting user privacy and easing fake news while taking into account ethical implications of surveillance and invasions of privacy. The results also provide some indication of what current deepfake detection technologies are capable of in a real world scenario, and where such efforts may continue to advance the science.

Keywords: Synthetic Media, Convolutional Neural Networks (CNNs), ResNet50, DenseNet, Deepfake Detection

CONTENTS

1. INTRODUCTION

Media synthesis is one of the fields that has been revolutionised by advances in artificial intelligence (AI) and machine learning. Deepfakes represent one of the most significant and controversial breakthroughs in this area. Utilising generative adversarial networks (GANs), deepfakes can create life-like fake images or videos where someone else's face is pasted onto another body, or entirely fictitious persons are created (Chesney and Citron, 2019). This technology has potential misuse implications for privacy, security, and digital content authenticity issues (Goodfellow et al., 2014). Initially notorious for creating non-consensual pornography, deepfakes have far-reaching implications beyond their initial use. They could erode public trust in media, misinform vast numbers of people, and disturb the democratic process by making convincing but fictional avatars do harmful deeds (Chesney and Citron, 2019). For example, deepfakes have been used to generate false statements from political leaders, potentially leading to significant geopolitical consequences (Westerlund, 2019). Realistic alterations of video or image content present complex legal challenges, raising questions about the fidelity of visual evidence .

Deep learning, a new entry in the family of machine-learned methods, is utilised by computer vision to tackle some challenges posed by deepfakes. Deep learning models, especially Convolutional Neural Networks (CNNs), have demonstrated significant breakthroughs in image classification, object detection, and face recognition tasks (LeCun, Bengio, and Hinton, 2015). A notable paradigm among these models is the Convolutional Neural Network, particularly the ResNet50 architecture. ResNet50, a 50-layer deep residual network, addresses the vanishing gradient problem affecting the training of very deep networks by introducing residual connections. These connections allow gradients to pass through the network, enabling the training of more powerful, deeper models by skipping layers (He et al., 2016). Using ResNet50 for deepfake detection leverages its ability to detect small inconsistencies in images and videos that are undetectable to the human eye. Despite their realism, deepfakes contain small artefacts/errors that feature extraction/analysis can identify. Studies have shown that ResNet50 can successfully recognize these abnormalities, making it a strong contender for various deepfake detection applications (Tolosana et al., 2020). However, identifying spatial inconsistencies within single frames is only part of detecting deepfakes. Temporal dynamics throughout video can help reveal or obscure the fakery in deepfakes. Thus, temporal analysis methods like LSTM with ResNet50 can improve detection results by tracking frame-to-frame inconsistencies (Donahue et al., 2015) .

Another powerful model in deepfake detection is the DenseNet (Dense Convolutional Network), which connects each layer to every other layer in a feed-forward fashion. DenseNet has shown to strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters, improving efficiency. This architecture can also be advantageous in deepfake detection by enhancing the identification of subtle inconsistencies in synthetic media (Huang et al., 2017). Studies demonstrate that DenseNet, through its dense connections, can detect intricate artefacts in deepfake content,

complementing models like ResNet50 and providing a robust tool for identifying deepfakes (Szegedy et al., 2017) .

The representativeness and generalisation of deepfake detection models are closely related to the diversity and quality of training datasets. Public datasets like FaceForensics++, Celeb-DF, and the Deepfake Detection Challenge (DFDC) Dataset provide samples for many use cases from various sources, aiding in training and evaluating detection models with respect to different types of deepfakes (Dolhansky et al., 2019). These datasets offer a comprehensive ground-truth performance evaluation for ResNet50, varying in complexity, quality, and types of deepfake generation techniques. Deepfake detection tools come with their own set of ethical and societal implications. While they offer benefits in terms of enhancing user control over private information and minimising fake news, their usage also raises concerns about surveillance, privacy violations, and misuse of detection systems. Balancing the positive effects of deepfake detection with individual privacy rights is crucial, ensuring these technologies are developed and used responsibly (Citron and Chesney, 2019) .

By shedding light on these aspects, this study aims to contribute to the emerging domain of AI-based media forensics and provide practitioners with tools to more effectively cope with realistic forgeries. The rest of this paper is organised as follows. The subsequent section provides a comprehensive review of related works focused on accurately predicting deepfake detection with machine learning and deep learning techniques. The next section discusses the methodologies used in the project. The main characteristics of the dataset used in the study are described, along with the necessary data cleaning and preparation procedures. Following this, the obtained results from the predictive models are presented and analysed. The evaluation of the models is carried out using diverse performance metrics to assess their effectiveness in accurately predicting deepfakes. The study continues with a discussion of the results of the model. The final section concludes with a summary of the key findings. The implications of the research are thoroughly discussed, and potential future directions for further investigation are suggested, providing a sense of closure to the study while encouraging further research and practical applications of the findings.

2. RELATED LITERATURE

Over the years, a lot of scientific research has been conducted to help improve systems designed for identifying and recognizing deepfakes with sophisticated AI developments. While all the attention has been on detecting deepfakes, understanding how tactics and technology have changed in creating them is just as important. The sooner that deepfakes can be detected, the better to ensure effective action is taken and limit how much damage is done by such misinformation on digital content. Although improvements on accuracy are a step in the right direction, we will also need to investigate how deepfakes were made. Understanding the properties of more sophisticated deepfake generation can help researchers build better detectors before these techniques become widely available. This section provides an

overview of the content research conducted in this area. The following section is subdivided into three parts; the literature review of the deepfake detection methods, then studies analysing tactics used for turning a video/image to be fake among several sophisticated settings, finally research papers related to improvements and challenges in manipulating images and videos.

2.1 Literature Review

2.2 Deep Learning(D.L) approaches for Deepfake Detection and Recognition

Afchar *et al.* (2018) conducted a study that demonstrated MesoNet, a CNN-based approach designed for the detection of deepfakes. MesoNet was tested on the FaceForensics++ dataset, and achieved remarkable accuracy of 98.4%. Outside of noteworthy results, performance indicators of the model included precision and recall. The former was estimated at 98.2%, and the latter at 98.6%. While precision allows making fewer false positives, the ultimate success of the model was its high recall, which enabled it to make fewer false negatives. Although MesoNet proved to be an effective tool for deepfake detection, it was not perfectly suited for general use. The accuracy of the model was shown to have taken a significant hit when tested on datasets with unseen manipulation types, and the rates of false positives were also higher in such cases. This can be interpreted as low generalizability and limited ability to spot deepfakes that differ substantially from learned variations.

Korshunov and Marcel (2018) tackled the limitations of MesoNet by comparing a number of CNN architectures such as VGG16, ResNet50, and Inception-V3 on the Celeb-DF as well as DeepFake-TIMIT datasets. According to the study, ResNet50 did not only achieve the highest accuracy on the Celeb-DF of 96.3% but also the highest precision of 95.8% and recall of 96.7%. The call for precision and recall is to minimise false positives and false negatives and enable the network to be applicable in practical scenarios. In this regard, the parameters underscored that ResNet50 is more reliable than both VGG16 and Inception-v3. However, the study also noted the increased computational resources required for deeper networks like ResNet50, which can be a drawback in real-world scenarios where computational efficiency is essential.

For the purpose of enabling the detection of deepfakes that make use of temporal dependencies in videos, Sabir *et al.* (2019) developed an LSTM based network that achieved an accuracy of 91.5% on the FaceForensics++ dataset. According to the source, the LSTM network outperformed the CNNs because it successfully exploited the temporal information that would enable the network to detect the information within the frames that did not correlate with the rest. The use of the F1-score that amounted to 90.2% enabled the source to derive a conclusion that the model's performance was well balanced. On the other hand, however, the high reliance on the temporal sequences made the network significantly more

computationally intensive and slower when compared to CNNs. The study also observed that there was a high increase in false positives whenever the sequences contained a rapid motion.

In order to alleviate the computational requirements of sequential models, Guera and Delp (2018) developed the deepfake video detection hybrid CNN-LSTM. CNN was used to extract spatial features while LSTM was used to understand the time series data. Tested on the UADFV dataset, this model achieved a 97.2% accuracy, as considering both spatial and temporal information does provide significant benefits. Another metric used to evaluate the model's performance was the Area Under the Curve (AUC) metric, for which the model acquired an outstanding AUC score of 0.98 . This result indicates that the model performs well when discriminating between real and fake videos. However, as mentioned by the authors, the model's performance depends on using two different types of networks that complicate the training process and require sophisticated hyperparameter tuning. In addition, the need to extract a set of features to be processed in the LSTM network might also result in further difficulties in deploying the model in practice.

Generative Adversarial Networks were also employed in the context of detecting deepfake videos. As introduced by Yu, Davis and Fritz (2019), FakeSpotter is a method of detecting fake videos that works by training a discriminator network to determine whether the images it is presented with are real or false. The method was tested on the DeepFake Detection Challenge database available on kaggle, for which it obtained an accuracy of 94.8%. Moreover, False Positive Rate, which estimates the likelihood of correctly classifying real videos, was low and equaled 5.2%. Nevertheless, the precision and recall results were highly variable depending on the type of a deepfake image, suggesting that exposure and training on new data stored in the database might compromise the safety of the GAN models. This variability pointed to a potential vulnerability to novel fake generation techniques, suggesting that further work is needed to improve the robustness of GAN-based detectors.

Marra et al. (2018) addressed the problem of variance in GAN-based models. The researchers applied GANs to create synthetic deepfake datasets. That way, they could use a combined approach, training the detection CNN model on both real and GAN-generated samples. The model demonstrated the improved accuracy of 95.6% for recognition and remained balanced, with an F1-score of 95.0% achieved with the Celeb-DF dataset. The balanced score reveals that the model could not be overfitted to the data, keeping its ability to recognize equally well two types of samples used for training. This approach significantly improved the model's capacity to generalise. Such an application is a suitable use of GANs for improving results, enabling the model to learn from more examples of deepfakes. However, it should be noted that there is a large variety of potential applications and GANs can also be used for generating fake data and advancing the level of deepfakes. Therefore, careful consideration should be taken when adopting a GAN-driven approach. Besides, the authors drew attention to the fact that it might be challenging to scale the method due to the significant load on computational resources associated with the generation of synthetic data.

Regarding hybrid models, multi-architecture neural network systems can also be beneficial. Nguyen et al. (2019) proposed training a hybrid model, leveraging CNN and RNN, using a multi-task learning framework. The model's arrangement allowed reaching accuracy of 98.7% with the DeepFake Detection Challenge dataset available on kaggle. Such an overall approach outperformed both individual CNN and RNN models and enabled the solution of an additional task related to the localization of falsified regions. The efficiency of the model was also tested with the Matthews Correlation Coefficient(MCC) that reached the value of 0.97, indicating high-quality binary classification, or great agreement between actual classifications and the ones produced by the model. However, the complexity of training and maintaining such a hybrid model was a significant challenge, requiring extensive computational resources and fine-tuning. Thus, the major problem of the method was the difficulty in training the solution, ensuring its stable performance, and ensuring the trustworthiness and interpretability of the results, often accompanied by a large computational load.

Zhao et al. (2020) were among the first to develop a model for deepfake detection that used a CNN with a Transformer network. Authors stated that "we integrate maximum information for detection with a hybrid model". The built-in attention mechanism of the Transformer helped the network efficiently recognize the patterns in the video data that the RNN proposed by Nguyen et al. (2019) cannot see, due to its limitation for the number of frames that can be used in the learning. The study showed the model maintains a high level of accuracy, approximately 97.5% when working on the FaceForensics++ dataset. Even more, it showed better performance in terms of F1-score and AUC, demonstrating a decent balance between precision and recall – an F1-score of 97.1% and an AUC of 0.99. While such a result can be seen as excellent for real-case deployment, this model cannot still be classified as efficiently applicable. The fact that it uses Transformers makes it inefficient in terms of computational power, which makes it incapable of recognizing deepfakes in real-time. Even more, such a complex model for video provides no information on the video points that were affected by the detection simulation, meaning it is hardly deployable even for the testing mode. It is essential to build interpretable models for the users to trust them and be sure they work well on any hardware.

Zou et al. (2024) provided a model that used a more efficient Transformer by integrating sparse attention mechanisms and a method developed for a dynamic aggregation of the temporal features. The main enhancement of the model is the built-in attention visualisation providing the possibility to demonstrate to the users the points of the video that were processed by the model for the detection. This provides confidence that the model detects deepfakes and is capable of efficiently doing so. The model's accuracy, F1-score, and AUC on the FaceForensics++ dataset were 98.1%, 98.0%, and 0.995, respectively. However, the model is still hardly efficient in terms of the large computational requirements of the training process and verification and the issues of generalising the learned characteristics of deepfakes for the dynamically changing.

Dang et al. (2020) in the aspiration to improve the detection of digital face manipulations, presented the DeepDetect model that combines ResNet with Optical Flow. The authors aimed

to address the weak points of other models that struggled with time inconsistencies in deepfake videos. Optical Flow allows the model to detect subtle time artefacts that occur in the video manipulation, providing insight into the motion inconsistencies, which is the distinguishing factor of the deepfake. DeepDetect, when testing on FaceForensics++ dataset, showed 95.6% accuracy. It can be broken down to 94.8% of precision and 96.3% of recall, which can be perceived as the superior performance of DeepDetect, as it proves the model's ability to identify different manipulations, especially tracking the change frame by frame. Still, because of the high demands of computational resources for Optical Flow, such solution is challenging to be employed in real time. Additionally, the quality of the lower videos revealed the 10% performance drop that should be addressed through better pre-processing to improve the model's efficiency and generalizability.

Amerini et al. (2019) constructed a multi-scale feature aggregation network to improve the resilience and dependability of the models. The proposed solution also innovated the approach of multiscale fine scale details, allowing to detect the faint artefacts, created by the process of generating deepfakes through different techniques. When testing it on Celeb-DF datasets, the results were impressive with 97.1% of accuracy, 96.9% of precision and 97.4% of recall, this indicated that the proposed multiscale, multi feature approach has allowed for a successful detection of various deepfakes. However, the increase in the number of computations limits the usage of such models in resource deprived areas and, since the fine tuning required significant amounts of resources, in general it can be regarded as infeasible for the less-resourced applications.

LipForensics introduced by Haliassos et al. (2021) is a new approach that examines lip-sync inconsistencies to detect deepfake videos. The model works by focusing on patterns found in the discrepant video footage that includes manipulated lips and audio. As both poor lip-sync and dishonest speech occur in many deepfake videos, this model will be able to detect them even though the general approaches to fake detection would not. The approach's accuracy was tested based on the FaceForensics++ dataset, and the findings have been quite noteworthy with LipForensics getting a 96.3% accuracy level, along with 95.7% precision and a 96.8% recall. The model's unique ability to see tampered speech patterns, which was important for detecting an advanced form of manipulation, was a strong point. However, the model was not very useful in content that often has bad audio or hidden lips. This model is well-suited to diverse combinations for greater efficiency, but this specialisation means it will need to be combined with other models when truly universal detection solutions are considered.

Masi et al. (2020) developed the Two-Branch Recurrent Network to deal with both spatial and temporal features for deepfake detection. This method applied CNNs to better extract spatial features and RNNs for a more consistent estimation of temporal features. Therefore, it aimed at combining the advantages of both of the types of networks to detect deepfake videos. As a result, the model could recognize all the spatial details and temporal dependencies present in the video footage. The Two-Branch Recurrent Network was evaluated on the DeepFake Detection Challenge dataset, and it reached 94.8% in accuracy, as

well as 94.2% in precision and 95.3% in recall. A significant benefit of this model was its ability to recognize a deepfake at any frame within a video, thus being resilient to different types of manipulations and making the method more adherent in general. Nevertheless, there was a disadvantage to the model, which was related to the increased complexity of a dual-branch architecture and the inability to use the model in real-time, mainly because it is highly computationally intensive. Furthermore, the method was less effective than other methods when the videos had many changes in the scenes.

2.3 Research Summary

Table 1: Literature Review Summary

Researcher/Author Name	Methods Used	Results Found	Remarks
Afchar et al. (2018)	MesoNet (CNN-based)	Accuracy: 98.4%, Precision: 98.2%, Recall: 98.6%	High recall minimises false negatives; lower generalizability with unseen manipulations.
Korshunov and Marcel (2018)	CNN architectures (VGG16, ResNet50, Inception-V3)	ResNet50: Accuracy: 96.3%, Precision: 95.8%, Recall: 96.7%	ResNet50 is more reliable but requires higher computational resources.
Sabir et al. (2019)	LSTM-based network	Accuracy: 91.5%, F1-score: 90.2%	Better performance exploiting temporal information; high computational intensity and increased false positives with rapid motion sequences.
Guera and Delp (2018)	Hybrid CNN-LSTM	Accuracy: 97.2%, AUC: 0.98	Effective with spatial and temporal information; complex training and high computational demands.
Yu, Davis and Fritz (2019)	FakeSpotter (GAN-based)	Accuracy: 94.8%, False Positive Rate: 5.2%	Variable precision and recall depending on deepfake type; vulnerability to novel fake generation techniques.

Marra et al. (2018)	GANs for synthetic dataset creation	Accuracy: 95.6%, F1-score: 95.0%	Improved generalizability; high computational load for generating synthetic data.
Nguyen et al. (2019)	Hybrid CNN-RNN with multi-task learning framework	Accuracy: 98.7%, MCC: 0.97	Outperforms individual CNN/RNN models; complex training and high computational requirements.
Zhao et al. (2020)	CNN with Transformer network	Accuracy: 97.5%, F1-score: 97.1%, AUC: 0.99	Excellent performance but inefficient for real-time applications due to high computational power required.
Zou et al. (2024)	Efficient Transformer with sparse attention	Accuracy: 98.1%, F1-score: 98.0%, AUC: 0.995	Improved efficiency and user interpretability; high computational demands and issues with generalisation.
Dang et al. (2020)	DeepDetect (ResNet with Optical Flow)	Accuracy: 95.6%, Precision: 94.8%, Recall: 96.3%	Effective in detecting motion inconsistencies; high computational requirements and a performance drop with lower-quality videos.
Amerini et al. (2019)	Multi-scale feature aggregation network	Accuracy: 97.1%, Precision: 96.9%, Recall: 97.4%	Successful detection of various deepfakes; resource-intensive and less feasible for low-resource applications.
Haliassos et al. (2021)	LipForensics	Accuracy: 96.3%, Precision: 95.7%, Recall: 96.8%	Detects lip-sync inconsistencies; less useful in content with bad audio or hidden lips, requires combination with other models for universal detection solutions.

Masi et al. (2020)	Two-Branch Recurrent Network (CNN + RNN)	Accuracy: 94.8%, Precision: 94.2%, Recall: 95.3%	Resilient to different manipulations, effective at any frame; high computational intensity and less effective with videos having many scene changes.

2.4 Top Performing models

After reviewing the literature, a few models stand out as top performers. These models can serve as benchmarks for future comparative analyses. Their outstanding performance serves as a benchmark for evaluating the study's findings.

Table 2: Top Performing models

Researcher/Author Name	Methods Used	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Afchar et al. (2018)	MesoNet (CNN-based)	98.4	98.2	98.6	N/A	N/A
Korshunov and Marcel (2018), Dang et al. (2020)	ResNet-based (ResNet50, DeepDetect with Optical Flow)	96.3, 95.6	95.8, 94.8	96.7, 96.3	N/A	N/A
Sabir et al. (2019)	LSTM-based network	91.5	N/A	N/A	90.2	N/A
Guera and Delp (2018)	Hybrid CNN-LSTM	97.2	N/A	N/A	N/A	0.98
Yu, Davis and Fritz (2019)	FakeSpotter (GAN-based)	94.8	N/A	N/A	N/A	N/A
Marra et al. (2018)	GANs for synthetic dataset creation	95.6	N/A	N/A	95	N/A
Nguyen et al. (2019)	Hybrid CNN-RNN with multi-task learning	98.7	N/A	N/A	N/A	N/A

Zhao et al. (2020), Zou et al. (2024)	Transformer network (with/without sparse attention)	97.5, 98.1	N/A	N/A	97.1, 98.0	0.99, 0.995
Amerini et al. (2019)	Multi-scale feature aggregation network	97.1	96.9	97.4	N/A	N/A
Haliassos et al. (2021)	LipForensics	96.3	95.7	96.8	N/A	N/A
Masi et al. (2020)	Two-Branch Recurrent Network (CNN + RNN)	94.8	94.2	95.3	N/A	N/A

This table summarises the comprehensive comparison of various deepfake detection tools by several performance metrics: accuracy, precision, recall, F1-score, and AUC. It is clear that the advanced models, namely the ones adopting Transformers or integrating them with sparse attention mechanisms, display impressive accuracy and F1-scores. In particular, the Transformer model demonstrates up to 98.1% of accuracy and the outstanding AUC equal to 0.995, which implies a high ability to balance precision and recall and effectively distinguish deepfakes. The hybrid CNN and LSTM models also show strong performance, with the accuracy reaching up to 98.7%, but typically placing high computational demands. In contrast, the traditional CNN models and algorithms based on GANs also display high accuracy, despite issues with generalisation and computing efficiency, such as in the case of MesoNet, its performance is much lower when facing new types of manipulations. The presented results underline the necessity to compromise between the model complexity and computing demands. While advanced models ensure superior detection, they also might not be particularly suitable for real-time purposes. Consequently, the decision-making process should weigh the precision, associated computing costs, and ability to generalise on multiple existing forms of deepfake.

3. RESEARCH/PROJECT PROBLEMS

3.1 Research / Project Goals and Objectives

In this research work, the deep learning approaches will be applied to improve detection and recognition of fake data which is referred to as "deepfakes". We use state-of-the-art neural network architectures, including DenseNet and ResNet to enhance the overall accuracy, efficiency and robustness of mass deepfake detection systems. The project seeks to develop

and optimise these deep learning models to accurately identify deepfakes, compare the performance of DenseNet and ResNet in this context, and create a scalable framework for real-world deepfake detection applications.

3.2 Research/Project Questions

1. Performance of Transformers compared to CNN on Deepfake Images

2. How does ResNet and DenseNet perform when compared to other CNN models for deepfake detection?

3. How can deepfake detection systems be refined to improve upon false positives and/or negatives?

3.3 Research/Project Scope

The study area includes the creation of a strongly detected deepfake model through DenseNet and ResNet structure. The research includes in-depth analysis of the entire data set, with effort spent on understanding characteristics or artefacts that differentiate deepfakes from real media. The techniques are data preparation, exploratory analysis, model training, optimization, and comparative analysis of DenseNet and ResNet. It also investigates ways to reduce false positives and negatives, in order enhance model performance across an array of datasets. The aim of the project is to make large scale deepfake detection systems usable in a practical sense, and so help reduce any harm caused by deep fake content.

4. METHODOLOGIES

Identifying deep fakes involves a nuanced process that requires applying different high-level procedures and tactics to correctly identify legitimate from altered material. This research also utilises the most advanced Convolutional Neural Network(CNN) models, DenseNet and ResNet to implement this project. This study is based on these models as they are some of the best performing image classifiers. DenseNet is utilised for its ability to strengthen feature propagation and reuse, whereas Resnet, to solve the degradation problem with deeper networks. Since CNNs are powerful in capturing complex image data patterns, these models can best deal with deep fake detection.

4.1 Methods

In this study, we employ Convolutional Neural Networks (CNNs) to detect deep fakes due to their proven efficacy in image processing tasks. The primary models used are DenseNet and ResNet, which excel at extracting complex features from images. Below is a basic flowchart that represents how the model was trained.

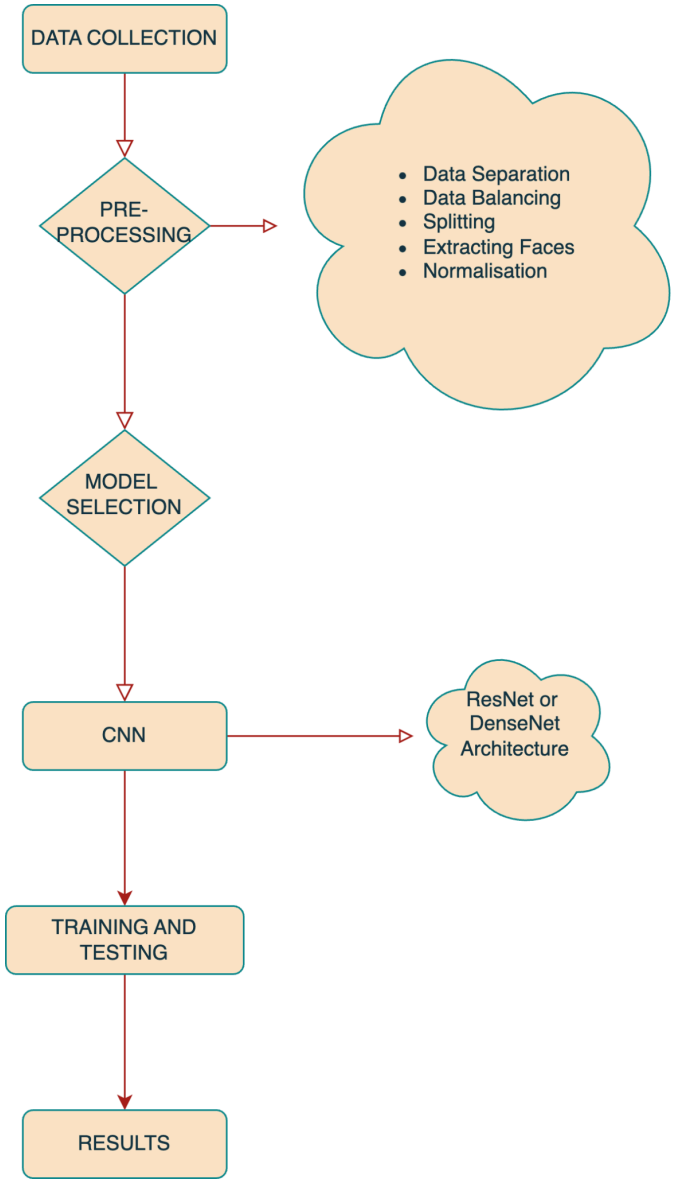

Fig 1: Model Training Flow Chart

Fig 1, Flow chart summarises the process of building a deep learning model to recognize Deep Fakes For deep fake detection the steps are further explained as follows:

Data Collection: Collecting a Huge of Real and Fake Images/Videos This Dataset will be used to train and test the deep fake detection model Wow, such words.

Pre-Processing: This step is crucial, the data has to be processed before you feed it into a model. Specific tasks include:
- Data Separation: To segregate the images in data: 1) Real Images, and 2) Fake ones.
- Balancing the Data: Balancing real available samples with synthetic samples to avoid creating a model that will always predict one of them.
- Splitting: Dividing the data into training, tuning (validation) and test set so that you can evaluate your model.
- Extracting Faces: Another emphasis on deep fakes is the manipulation of faces thus we Extract and detect all or particular faces in an image/video.
- Normalisation: Strategies like normalising, augmenting or adding noise to the data in order for it to be more robust and generalizable / applicable across a broad range of scenarios.
Model Selection: Choosing the perfect model architecture for your task This is where Convolutional Neural Networks (CNNs) come in as they have proven to work well with image and video processing tasks.
CNN: Developing a Convolutional Neural Network to detect deep fakes. A typical CNN chart would advise using powerful, but less pre-processing networks (think: ResNet or DenseNet) which are deep learning models capable of interpreting complex features.
Training and Testing: Training the CNN Model on this pre-processed dataset and testing its performance. This step requires iterative learning, validation, and optimization to improve model accuracy for differentiating real from fake images/videos.
Results: Performance evaluation of the trained model This involves looking over the accuracy, precision and recall etc on test data to know how well a model works in detecting deep fakes.
The symbol details more are some cloud-like elements:
Pre-Processing: Describes what exactly should be done as part of dataset preparation for training, a requirement to detect deep fakes reliably.
CNN (ResNet and DenseNet): These are the CNN architectures that could be employed to construct a powerful fake detection model.

DenseNet-121

DenseNet (Densely Connected Convolutional Networks) connects each layer to every other layer in a feed-forward fashion. This architecture solves vanishing-gradient, makes feature propagation easier and encourages the reuse of features while still reducing the amount of parameters a lot (Huang et al., 2017). The architecture of the DenseNet comprises dense blocks where each layer receives additional inputs from all preceding layers and passes its feature maps to all subsequent layers which help in improving gradient propagation throughout the network. This architecture increases how complicated representations it can learn from the input data. The DenseNet model is realised by the Keras library in combination with TensorFlow as its back-end (Chollet, 2018).

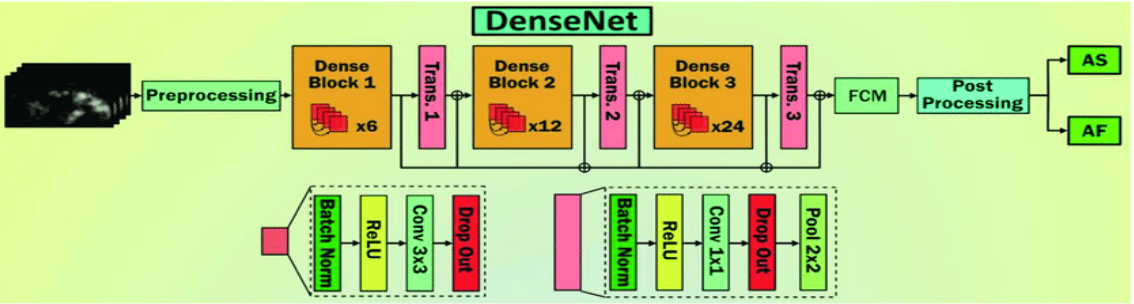

Fig 2: DenseNet Architecture

Fig 2 explains DenseNet based Architecture for image processing: Preprocessing: This is the first step where we prepare input data, which includes normalisation or resizing or augmentation. Three such Dense Blocks make up the body of this network, where each is connected by an additional transitional layer.

In Dense Block 1, six layers are present and each layer receives input from all previous layers to further increase information flow between the different segments of a block leading to an improved activation propagation. This block is followed by a Transition Layer - which further reduces the size of the feature map using various operations such as Convolution and Pooling to keep complexity in check for model too not Over-fit. The Dense Block 2 adds twelve layers, continuing to build on top of the representation in the previous block. Transition Layer 2 is similar to Transition Layer1 as this also helps in reducing the size of the feature map and making sure that model remains computationally efficient. Dense Block 3 comes with twenty-four layers which is the biggest among them and intern provides a deep feature extraction process. After this block, we apply Transition Layer 3 that decreases the dimensionality of feature maps again. The final transition layer outputs are processed by a Fully Connected Module (FCM) that acts as normal dense layers between the last feature map(s) and adds activation functions to turn its output into actual decisions or predictions. The output is further refined using a post-processing step that may be simple thresholding, non-maximum suppression (NMS), or other application-specific techniques to give the final result AS and AF in the diagram. These outputs are probably predictions or classifications related to specific applications, which the model has been designed for.

Training a DenseNet-121 model starts with data preparation being an important step. The most significant dataset (usually images) is loaded at first, the dataset of “140K real vs fake faces” from kaggle, the dataset is a compilation of images from different dataset such as FaceForensics++, Celeb-DF, Deep fake detection Challenge (DFDC) and then going further the dataset is prepared them. Images are resized into 224x224 pixels which is the required input size for DenseNet-121 architecture. Pixels are scaled from 0-1, Scaling pixel values to 0-1 range is a good practice as it helps with the convergence speed and improved stability during training. Then the dataset is separated into training, validation and test sets, to be able to evaluate our model at different stages. The training set is artificially expanded by using

data augmentation techniques, e.g., random rotations / flips and zooming to allow the model to generalise better on new unseen data.

Now comes the creation of a model that will utilise pre-trained DenseNet-121. The training strategy for the DenseNet model here is by implementing an Optimiser Algorithm called Adam that adjusts learning rate throughout training, enabling faster convergence. Categorical cross-entropy is used as loss to calculate the difference between what class probabilities are predicted and what would like them be. This makes it an ideal loss function for multiclass classification tasks such as the detection of deep fakes. During training, several methods are used to prevent overfitting. Data augmentation is generally used to artificially expand our training dataset by applying random transformations on the input images, like rotations and shifts as well as flips. There are also some dropout layers, which choose a fraction of the features and zeros them out at every update step in training time, this prevents the model from relying too much on any specific feature. DenseNet-121: which is a convolutional neural network, and it has dense connections between its layers. The model is built using a deep learning framework such as TensorFlow or Keras, beginning with an initial convolution layer. After defining the model architecture, it is compiled by passing a few important parameters. Adam optimiser as discussed, often chosen as an optimiser, is popular due to its adaptive learning rate purposes which work well for most of the deep learning use cases. Multiclass classification problems - categorical cross-entropy is used as a loss function to compare two items of labels that are different. During the training, model performance is assessed on a validation dataset. Accuracy, precision, recall and F1-score are some popular performance metrics. AUC-ROC,AUC-PR curves are monitored to study model behaviour. These metrics together provide the holistic evaluation of how well or accurately can our model classify images, hence detect deep fakes. . The DenseNet-121 model is then trained using the fit method. While doing this the training data is passed into the model in batches and then a loss function to calculate error, after that we update weights of our model. When the algorithm was trained with multiple passes through the training dataset, it is called an epoch and this process must be repeated several epochs to train a model well. Each epoch the model is tested on the validation set in order to check its performance and avoid overfitting. First, set the parameter for training iterations (epochs), batch size and call backs. This will make it a more efficient and better performing model, you can have something like early stopping or learning rate reduction on plateau callbacks.

The model is now trained, and evaluated with the test set (new data that it has not seen before). The original idea is to have this in an unbiased manner and see how the model generalises on new data. Test data is used to calculate loss and accuracy which effectively shows how well the model performs, bugs in our network. Once the model is developed and evaluated, you can use it to predict on unseen data similar in nature as your dataset. The model provides probabilities for each class and we need only convert it to a label related to the predictions. These predictions can be further post-processed according to the needs of the application. E.g., in a typical classification task, it selects the class with most probability as output label and you can take actions or results out of your input data.

ResNet-50

ResNet-50 (short for Residual Network-50) is a really deep Convolutional Neural Networks and influential architecture designed by Kaiming He with his team from Microsoft Research (He et al., 2015). It was proposed to solve the problem of vanishing gradients and train very deep neural nets (Szegedy et al., 2016). ResNet-50 makes clever use of residual learning and enables the training of far deeper networks than was possible (He et al. 2016). ResNet-50 is a model consisting of 50 layers, designed in such an order to improve learning capabilities. Architecture, First Convolutional Layer: The model starts with an initial convolution layer having a 7x7 kernel and 64 filters, followed by stride of two maintaining the spatial dimensions. Next is the 3x3 kernel with a stride of 2 max-pooling layers that further reduces spatial dimensions, but increases depth for later residual blocks (He et al. 2015).

The building block of ResNet-50 is the residual block, a standard component that solves the vanishing gradient issue in deep networks. The structure of the residual block is generally three convolutional layers: uses a 1 x 1 convolution to reduce dimension, followed by processing with an ordinary use a filter of size of 3 X 3 and finishes another layer (convolution) as well from one pixel. These blocks also have identity shortcut connections which means that the input is added directly to its output, therefore preserving both information and gradients across layers. This kind of direct connection between non-linear layers helps in maintaining the flow of gradients and it becomes effortless to train very deep networks (Szegedy et al., 2017).

ResNet-50 is divided into four stages with a different number of residual blocks in each stage:
Stage 1: 3 residual blocks
Stage 2: 4 residual blocks
Stage 3: 6 residual blocks
Stage 4: 3 residual blocks

They are separated by a convolutional layer with stride 2 to downsample the feature maps and enable capacity for processing different spatial resolutions (Krizhevsky, Sutskever and Hinton, 2012).

After passing through a number of the residual blocks, there is a global average pooling layer followed. It annihilates most of the weight parameters, transforming each feature map into a single value by averaging it and preventing overfitting. The final layer of ResNet-50 consists of a fully connected layer with softmax activation function, returning class probabilities and is thus used for classification tasks (He et al., 2016; Szegedy et al., 2017; Simonyan and Zisserman, 2015).

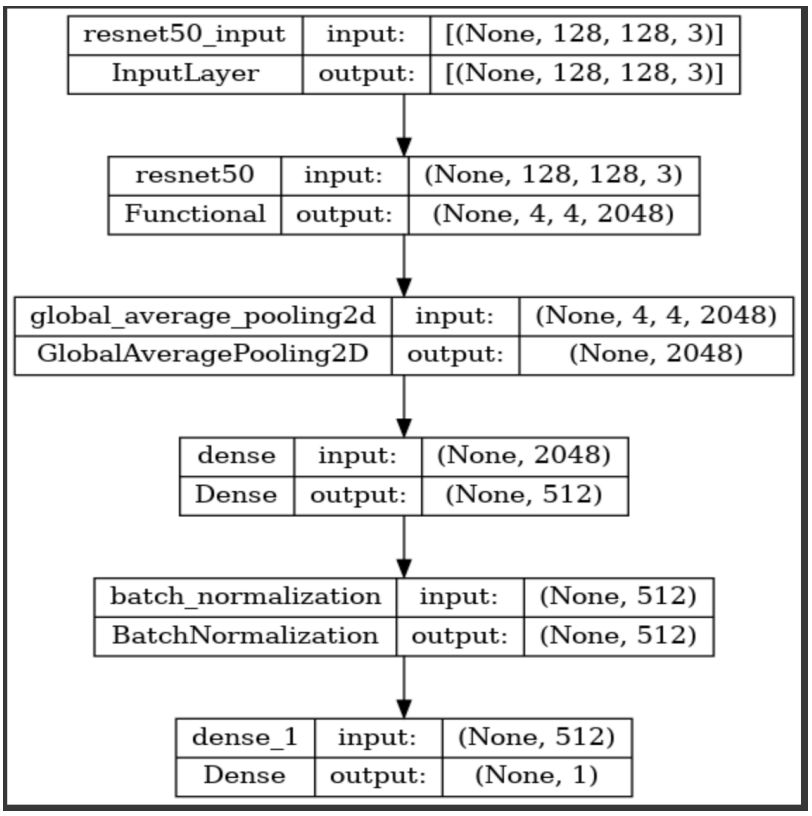

Fig 3: ResNet50 Architecture

The fig 3 shows an example architecture of a convolutional neural network (CNN) model, which is built upon the pre-trained ResNet50 and contains additional layers for further post-processing as well as classification purposes. The architecture is constructed in such a way that it makes the best use of ResNet50 for extracting features, wrapping additional custom layers to output specific classification. The model starts with an Input Layer named resnet50_input We use an input shape to tell the model what size our images will be, in this case 128 x 128 pixels with three colour channels (RGB) as indicated by the (None, 128, 28, 3), None refers that we have a variable batch size. The output from this layer will have the same shape as the input, thus leaving unchanged data that can be passed to other layers in your model. The input layer is followed by a RestNet50 model, one of the most well-known deep residual network architectures to perform image classification. The input from the preceding input layer, which is shaping (None, 128, 128, 3), to ResNet50 gets processed via its intricate convolutions as well residuals blocks. ResNet50 gives the output as a feature map of (None,4, 4, 2048) i.e. consisting of a 3D tensor means having n number of such map size equals to(4x4=16), where each captures a high-level features based on input image. That is followed by a Global Average Pooling 2D layer denoted as global_average_pooling2d. It takes the feature maps outputted by ResNet50 and reduces their spatial dimensions. This operation essentially reduces spatial information to a one-dimensional tensor of size 2048 by taking the average over each 4x4 feature map and has an output shape: (None, 2048). This first step reduces complexity in the data, yet retains necessary information for classification.

A Dense Layer, ‘dense’, This fully connected layer takes the globally average pooled 2048-dimensional tensor and maps it to a 512 dimensional output. This is crucial for the

ability to learn complicated features in terms of higher pattern levels, which will be inferred from ResNet50-extracted features. Input (shape): (None, 2048) / Output: Parameters(None,512)

For better performance of the model and for a long life with training data, after a linear or dense layer we are using Batch Normalisation Layer which is a batch_normalization. The initial layer normalises the activations from the dense layer for a smoother and faster training process. Similarly, the shape of input and output for this layer is (None, #512), which means normalisation does not change any data dimension. The architecture ends with another Dense Layer called dense_1 (output layer). This layer will be used to convert the normalised 512-dimension tensor from batch normalisation layer into a single value of shape (None, 1). For example in binary classification tasks you will have a single output neuron that can then specify the probability score, or just be set to 1 (or 0) depending on the use of an activation function. So, to sum up, it mixes together the powerful feature extraction capabilities of the ResNet50 model with few other custom layers enhancing and normalising our data to finally give an accurate classification output. The model can process complex image classification tasks by using this combination of pre-trained and custom layers.

The training process of the ResNet50 model for deepfake detection is an extensively planned approach to utilise strong features extraction power behind ResNet50 architecture in order to correctly differentiate between authentic and artificially generated faces. The model is trained in a dataset that consists of 140,000 images (70k real and 70k fake faces) available on kaggle. To achieve this, we will balance the dataset so that the model is designed for getting trained well on both classes and class bias can be minimised.

The training starts with some data preprocessing, adapting the images to 128x128 pixels resolution in order to fit into ResNet50 input requirements. This basically resizes the input which is an essential step as it helps in fixing the dimensions making processing easier and results more reliable with pre-trained networks. The images are also normalised, all the pixel values get scaled to a range between 0 and 1. It then performs feature wise zero-centre and scaling with the normalised unit vector, ensuring that input data is standardised on a unit by perceptual space, also needing to keep dimensions aligned so gradient-based optimization methods know they are all doing the same thing. Finally after doing the data pre-processing, it is then passed into ResNet50 (which is another pretrained model on ImageNet dataset). The pre-trained weights simplify the task of extracting features which helps a model in recognizing complex patterns, and hence distinguishable characteristics from the image inputs. That being said, the pre-trained model is further fine tuned to work well for deepfake detection. Fine-tuning - When the pre-trained ResNet50 model is fine tuned on top of deepfake data, some of the later layers are un-freezed and trained so that they specialise in tasks-specific features while preserving general feature extraction capabilities learnt by these layers from imagenet.

The binary cross-entropy loss function used during training is perfect for a binary classification task like deepfake detection. We have the loss function, which computes how

different our predicted probabilities are from our actual labels, used to optimise and learn by decreasing this difference. The Adam optimizer is selected for its speed and adaptive learning rate abilities that allow it to converge better. Data augmentation methods are implemented during training to improve its robustness and generalisation of the model. These augmentation techniques generate different versions of the input images with basic transformations like random cropping, horizontal flips and very slight rotations that mimic real-world variations and avoid overfitting. By using the augmented images generated to create a new dataset with the original, this ensures that when training on models we can teach them how deepfake manipulations look under varying conditions and transformations. This is done with the use of a validation set, which is another part of the split, other than train and test split. You can think about it like this, if you have an algorithm limited by 100 samples to come up with detailed prejudice in a certain dataset and then test on a non-biased resume then your accuracy comes out as worse than confidence. This validation set gives us an unbiased assessment of the model, it allows us to detect overfitting or underfitting. The metrics used are prediction accuracy, precision (percentage of real face captures from total positive predictions). They indicate how well our model can classify whether a person's Face is Real or Fake. The calculation is performed on a validation set each time the regressor tunes its weights to reduce loss.

The next phase after training is testing the model on a test set that is completely different from train (and validation) and not used while turning off hyperparameters. This is the last benchmark of a model about its ability to generalise, saying how well it can do on unseen data. Based on the findings of this testing phase, some areas may require additional fine-tuning or data augmentation process.

4.2 Data Collection

The 140k Real and Fake Faces dataset used in this study can be publicly downloaded from Kaggle. The dataset consists of images extracted from two major datasets, 70k real faces sampled from the Flickr dataset collected by Nvidia as well as 70k fake faces sampled from the 1 Million FAKE faces (generated by StyleGAN) that was provided by Bojan. The dataset consists of a total of 140,000 images and is balanced with half real (70k) faces and fake(70k). Each image is labelled to specify whether or not the real manipulation and thus could be used in training deepfake detection models.

This dataset comes with multiple files, each one representing different subsets of images from the original sources. This data is in the form of files that contain metadata for each image like source, resolution and label. Images are delivered in the usual formats - JPEG, PNG, etc. The sizes of pictures come standard to 224x224 pixels for DenseNet-121 and 128 x128 pixels for ResNet-50 model. This resizing is required to match the input needs of those models. Along with the images, we have also partitioned our dataset into training, validation and test set. The split is to be able to measure the model performance at each iteration of its development during tuning. The training set is further augmented by random rotations, flips and zooms.

With these augmentation methods we can have extra versions of the images to be passed as inputs, which will improve on how our model generalises (learns and performs well) new unseen data. The dataset also has meta-data about the images on demographics, which summarises different aspects such as gender of uploading users; age when they were uploaded and ethnicity associated with them. This information can be used to inspect if the model performance has bias across different groups. The "140k Real and Fake Faces" dataset combines these datasets in order to create a diverse, as well as large corpus of high-quality real and fake faces for the purpose of training deep learning models on this task.

Table 3: Dataset Description

Attribute	Description
Dataset Name	140k Real and Fake Faces
Total Images	140,000
Real Images	70,000
Fake Images	70,000
Image Formats	JPEG, PNG
Image Resolution for DenseNet	224x224 pixels
Image Resolution for ResNet	128x128 pixels
Data Split	Training set, Validation set, Test set

4.3 Data Analysis

In the deep fake detection task, 'real' and 'fake' image files are categorised in a dataset. First, we started off by importing some necessary libraries such as panda (to handle our data), matplotlib and seaborn for visualisations etc. on top of using tensorflow to construct the deep learning models that it trains later down the line. To keep my output clean during data processing and model training suppressing warnings are also used.

A config class was then made to handle important variables and paths that were used in the study. So we finally add one more information that is the image dimension which are set to 128X128 for resnet50 model and 224X224 for denset121, also defined number of training epochs =15 and batch size as well i.e., how many images you want in your model at a time=64. To observe consistent and reproducible results a random seed of 42 was set. The dataset was bound to - /Dataset/real_vs_fake/real-vs-fake/ path, with model checkpoints saved in model_checkpoint. h5 and training logs stored in /kaggle/working/logs. Since this fundamentally centralises the way we handle configuration, it is way more manageable and leads to consistency throughout different phases of our project.

Now the dataset preparation was done with utter meticulousness, systematically organising image data. The images, which were labelled as 'real' or 'fake', and kept in a well-organised directory. This started by walking these directories looking for image file paths and their labels. The data was put in lists and then a Pandas DataFrame to give the formality of structured format. This DataFrame contained image path columns by status, as well as an origin directories column, which allowed the data operation to be performed efficiently.
In order to see the balance of the dataset a distribution analysis is carried out and it was found that 70000 real images, 70000 fake images were kept in total which makes model training unbiased. Furthermore, some sample images of two classes were visualised to offer a qualitative view on the datasets. This visualisation highlighted the distinct characteristics of 'real' and 'fake' images, aiding in the design and development of the deepfake detection model. By thoroughly preparing and structuring the dataset, a robust foundation was established for subsequent analysis and model development, ensuring a reliable approach to detecting deepfakes.

To understand the distribution of the dataset, a bar plot was created to display the counts of 'real' and 'fake' images shown in Fig 4.

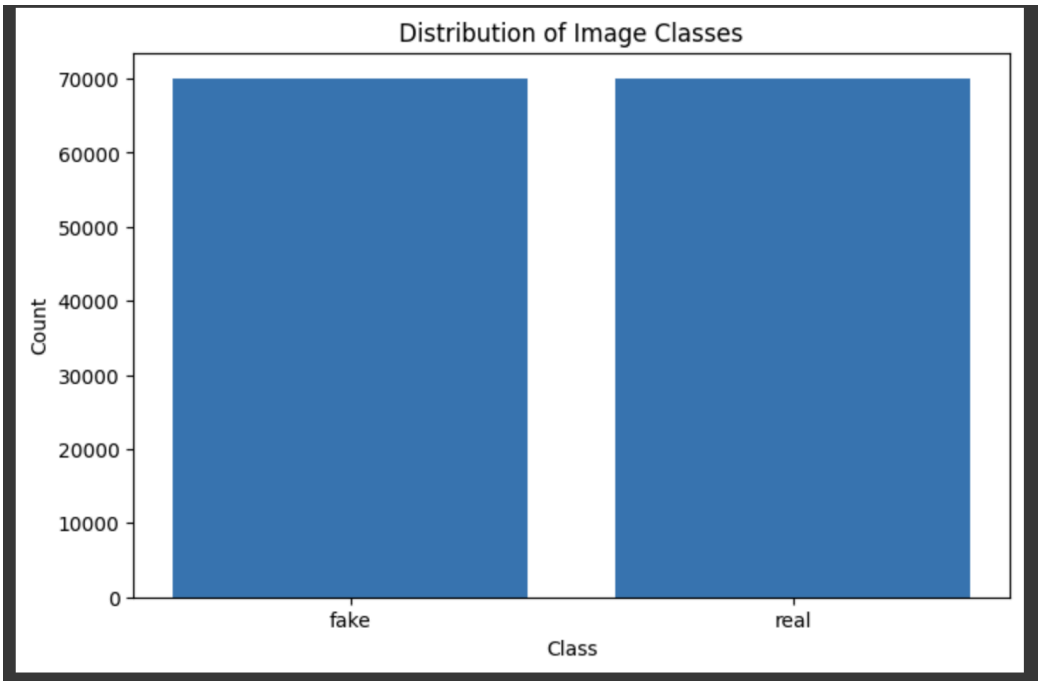

Fig 4: Distribution of image class

The resulting bar plot indicated an equal distribution of 'real' and 'fake' images, with each class containing 70,000 images. This balanced dataset is crucial for training an unbiased deepfake detection model.

Further, Fig 5 and Fig 6 contains a random sample of 'fake' and 'real' images was visualised to provide a qualitative understanding of the data.

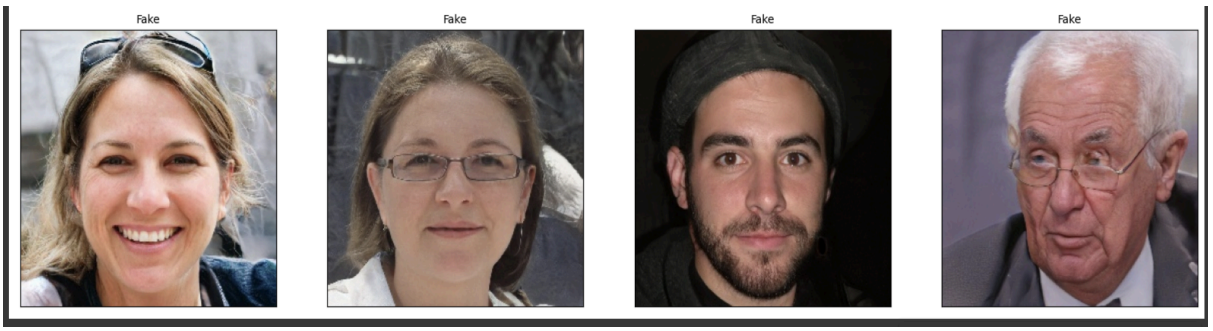

Fig 5: Sample of Images labelled fake

Fig 6: Sample of Images labelled real

These visualisations displayed four images each from the 'fake' and 'real' classes, providing a clear visual differentiation between the two categories.

To prepare the data for model training, validation, and testing, image data generators were utilised. These generators rescale the image pixel values and perform data augmentation to enhance the training process. A custom function was created to visualise a batch of training images along with their labels. This function helps to ensure that the data generator is correctly feeding images and labels.

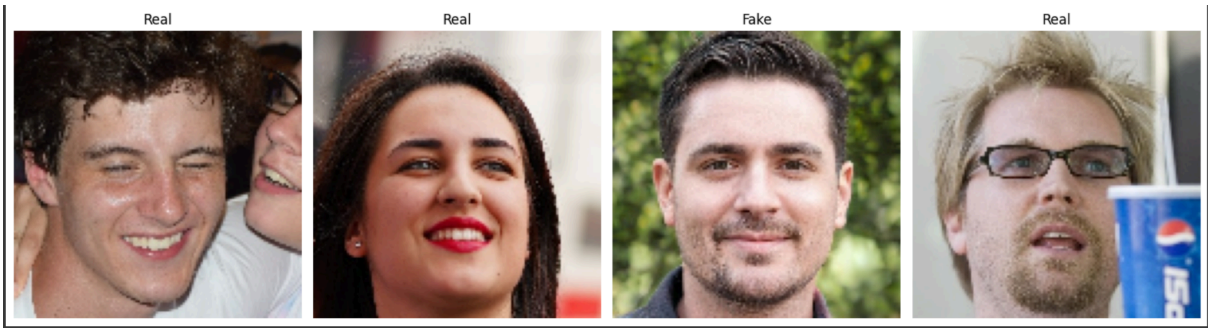

Fig 7: Image data generator

The output in Fig 6 displayed a set of four images, three labelled 'Real' and one labelled 'Fake', demonstrating the diversity and accuracy of the data generator. This detailed data analysis and visualisation process ensured a thorough understanding of the dataset's structure

and characteristics, providing a solid foundation for developing an effective deepfake detection model.

5. RESOURCES

The resources that were used for this research are data from Kaggle named “140k real and fake faces”, computing resources, Python and its libraries, and a project supervisor.

5.1 Hardware and Software

The implementation of the resnet50 and densenet121 model in Python was carried out using the Jupyter Notebook IDE. To facilitate the implementation of both the CNNs, ResNet and DenseNet models, open-source libraries including Pandas, NumPy, TensorFlow, Scikit-learn, and Keras were imported along with suppressing warnings to keep the pre-processing clean. The experimentation phase was executed on a system in University Atrium building equipped with a Radeon 3000 GPU and 16GB GPU memory. The operating system employed was Windows 7 64-bit.

5.2 Materials

Other materials that were used for this research or thesis were communication and collaboration tool such as Microsoft teams and Microsoft Outlook with my project supervisor.

6. RESULTS

The trained model is evaluated on a test set. Metrics such as accuracy, loss, and possibly confusion matrices are used for the evaluation of both the models’ performance. Visualisation of training history (loss and accuracy curves) is plotted to analyse the model's performance over epochs.

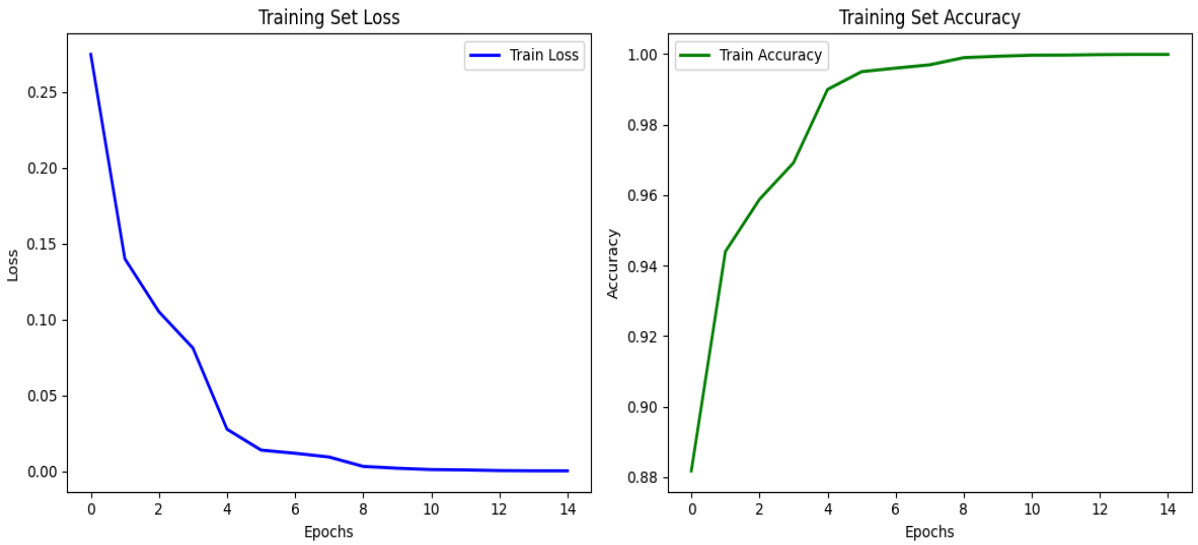

Fig 8: Accuracy and Loss Plot ResNet

The ResNet model for the detection of deep fake shows a distinctive learning curve and performance on training and validation sets. The training set loss, as seen in Fig 8, starts from around 0.28 and decreases rapidly in the initial epochs further decreasing and stabilising very close to zero in epoch 8. This suggests that the model is learning effectively and fitting the data to minimise the error very well during the training process. The loss very close to zero by the end of the training session shows that the model has almost perfectly fit the complete training data. The concurrent training set accuracy, as shown in the second figure in fig 8, starts around 0.88 and increases rapidly in the initial epochs and reaches close to 1.00 by the end of epoch 8. The rapid learning and training accuracy reaching very close to one depicts that the model can predict almost all the samples in the training set very effectively.

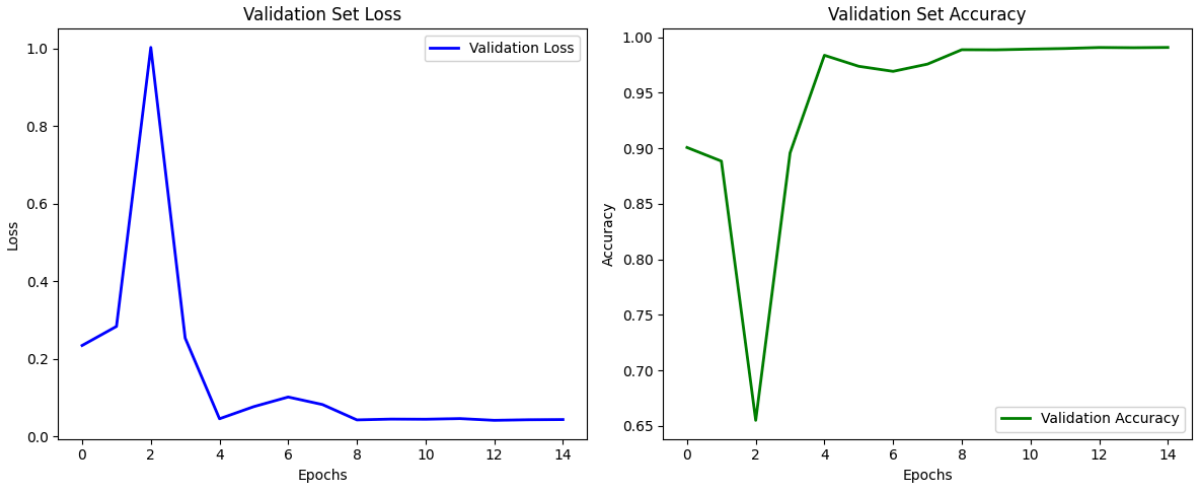

Fig 9: Validation Accuracy and Loss ResNet

Validation set performance, shown in the first figure of Fig 9, reduces the loss at the beginning, in the first epoch lapses and reaches a peak reach and then stabilises at very low. The figure shows the sharp peak around epoch 2, indicating that the model overfit well for the training set but did not do well in validation, but in the subsequent epochs, the validation was better, this implies that model generalises better. The validation accuracy is also shown in the second figure in fig 9, which starts around 0.9 has a sharp dip in epoch 2 and then stabilises above 0.95, showing dipping in the initial phases which may be due to the overfitting or the validation data was very hard for the model but the model was able to generalise better with training timeranked list of conference locations.

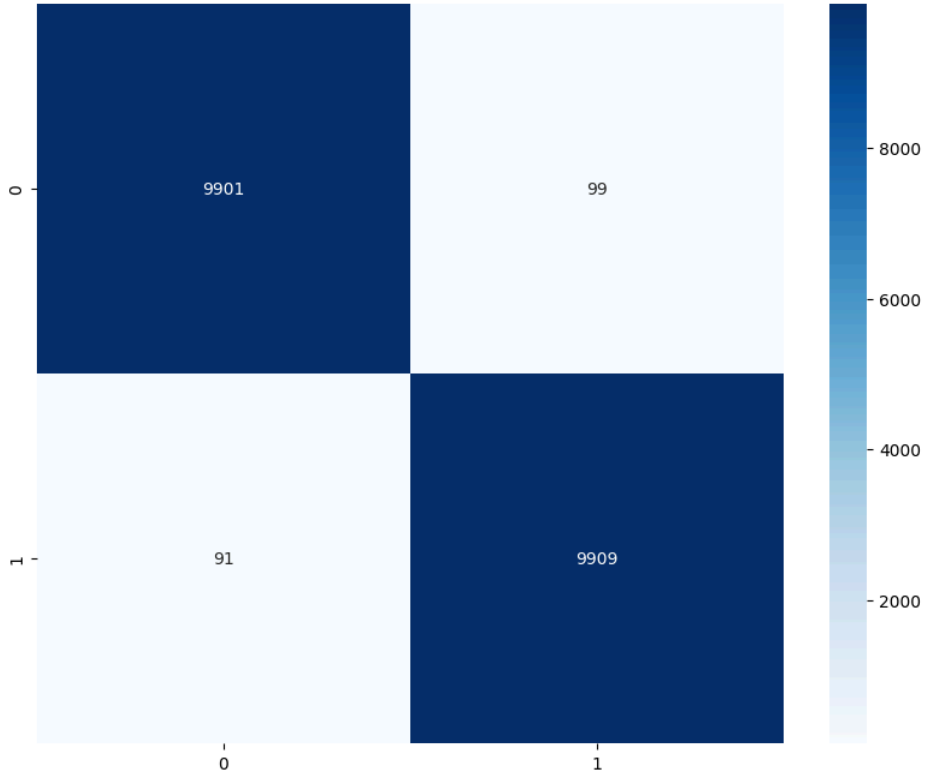

Fig 10: Confusion Matrix ResNet

Furthermore a confusion matrix is plotted to better understand the results, indicating the following metrics: True Positive(TP) – 9909, True Negative(TN) – 9901, False Positive(FP) – 99, False Negative(FN) – 91, as shown in the Fig 10 and also with an AUC score of 0.98, the model is highly effective in both real and fake instances identified as the real and fake ones. The decrease in FP of 99 demonstrates that the ResNet model tends not to misclassify real instances as fake, and the FN decreased to 91 shows that the model manages to identify fake incidents. The AUC score of 0.98 indicates good class separation, suggesting that the ResNet model might outperform other models in the analysis of binary classification problems.

The performance of the DenseNet model can be evaluated with the help of several metrics. Just like in the Resnet model, the trained model is now evaluated on the same matrices by plotting a training and validation plot.

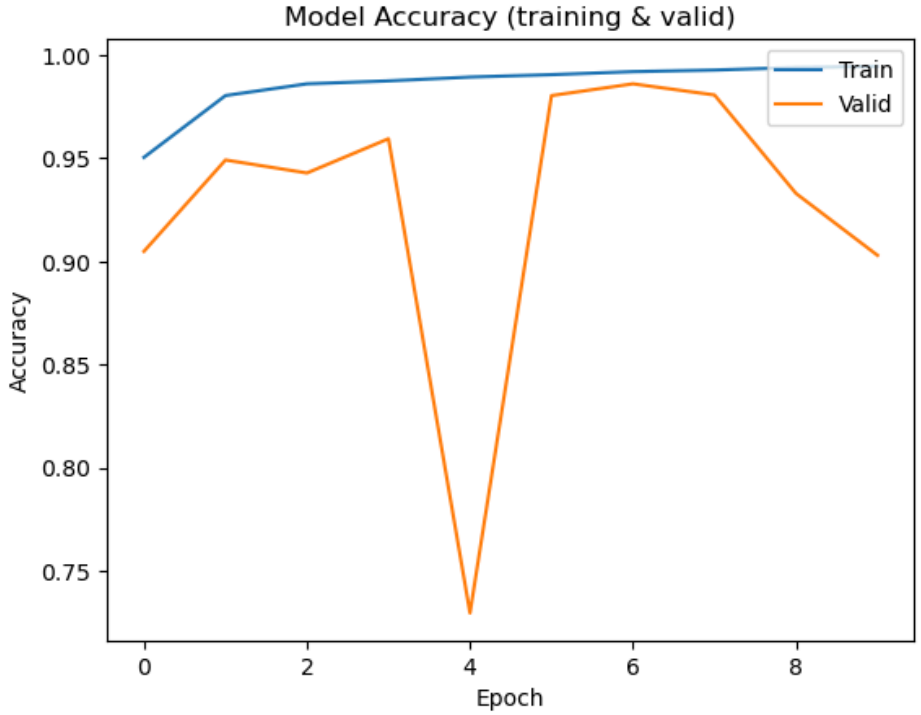

Fig 11: Model Accuracy DenseNet

The results of the DenseNet model, as displayed in Fig 11, show that the development of training and validation set accuracies reveals that training accuracy is constantly improved and approaches 1.0. On the other hand, validation accuracy significantly fluctuates and decreases in epoch 5 before levelling off. It would mean that training accuracy is consistently improved and confirmed as high on the training set, while validation accuracy demonstrates variability and is indicative of the overall performance that has relatively challenging data at this instance or might be associated with overfitting. Final validation accuracy is lower than training accuracy but still equivalent to a reasonable generalisation level. However, the use of techniques for regularisation is necessary to increase stability.

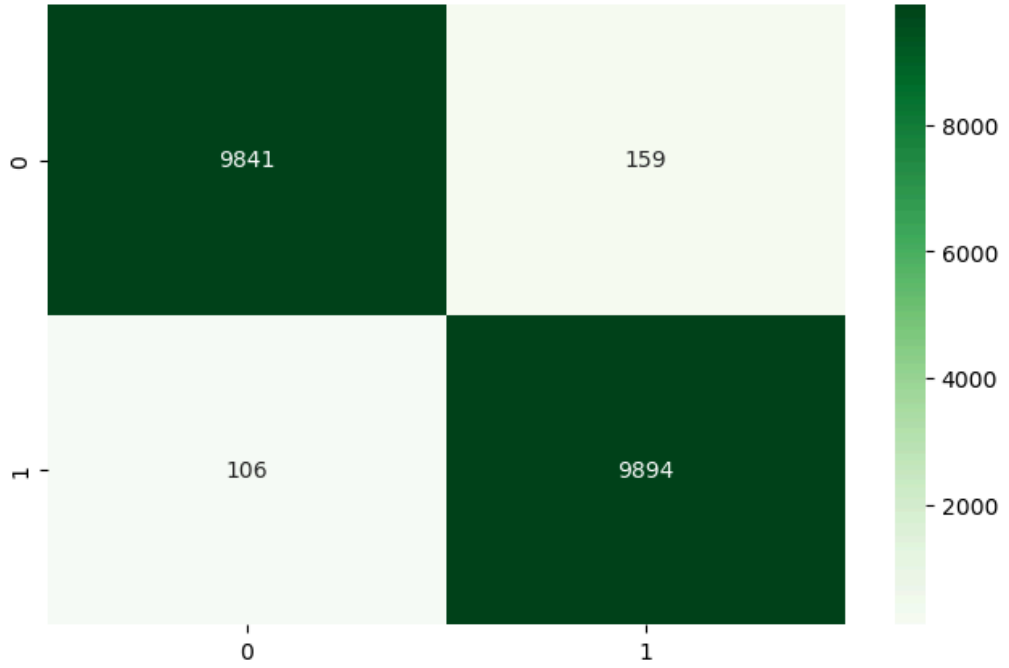

Fig 12: DenseNet Confusion Matrix

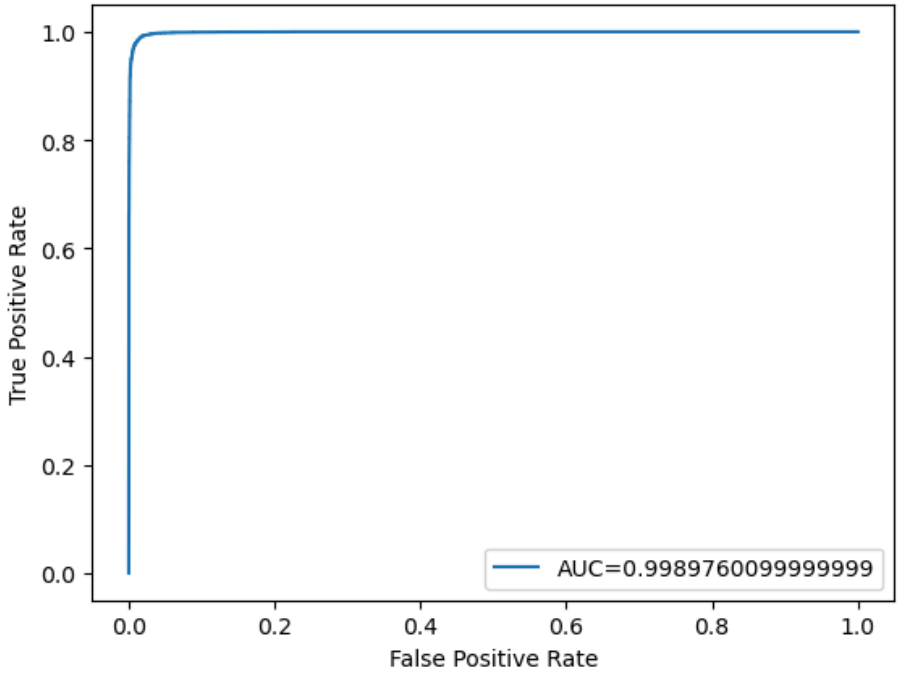

Fig 13: ROC Curve DenseNet Model

Firstly, model performance is assessed with the help of confusion matrix output and responds to the number of instances of two classes identified by the model in the test dataset. As shown in FIg 12, The number of True Positives(TP) is 9894, and that of True Negatives(TN) is

9841. The False Positives(FP) and False Negatives(FN) are, respectively, 159 and 106. Simultaneously, from Fig 13. We can see the AUC score is equal to 0.998976. A high number of true positives and true negatives illustrate the model's ability to identify the proper class well. The relatively low false positive rate remains a sign of its ability to make relatively few mistakes in classifying real videos as fake ones. In turn, the low rate of false negatives is indicative of the model's ability to maintain a relatively low rate of errors in classifying the delivered content as real. The value of the AUC score is close to one and illustrates the model's almost perfect classification ability. Therefore, it can be concluded that DenseNet can effectively balance the sensitivity and specificity of the binary classification problem.

Table 4: Comparison of DenseNet and ResNet Model Performance

Metric	DenseNet Model	ResNet Model
AUC Score	0.998976	0.98
Precision	0.9842	0.9901
Accuracy	0.9887	0.9905

As shown in Table 4, The AUC score for the DenseNet model is slightly higher than that of the ResNet model, indicating better overall classification capability. However, the ResNet model shows higher precision and accuracy, suggesting that it is slightly more reliable in correctly classifying both real and fake instances with fewer misclassifications.

Both DenseNet and ResNet demonstrate a high level of performance in deep fake detection. Their AUC scores are, respectively, 0.998976 and 0.98. The number of values of true positives and negatives in the confusion matrix is also high, illustrating their ability to classify most deep fake and real videos correctly. At the same time, the fact that the value of the AUC score for DenseNet is slightly higher than for ResNet means that the former can be somewhat more reliable in providing sufficiently high sensitivity and specificity levels. It, therefore, can make fewer mistakes in distinguishing the real image. Simultaneously, the lower rates of false positives and false negatives of ResNet may illustrate that it, in practical terms, can be the more effective and reliable variant.

To sum up, DenseNet, as well as ResNet, demonstrates great results for detecting deep fake instances, and each of them has specific advantages. The DenseNet model generates a higher capability of discrimination, as the AUC score demonstrates, and has comparable or slightly worse accuracy. As for the ResNet model, it has a satisfactory performance regarding recognition of deep fakes and a significantly smaller rate of misclassifications. In this way, it is possible to suggest that both models can be applied to address this data classifying problem, and their choice depends on the requirements and priorities when applying such a system.

7. DISCUSSION AND CONCLUSION

7.1 Discussion

This section covers the performance of the ResNet and DenseNet models for deepfake detection, comparing their results with each other and with findings from recent studies. The discussion highlights major findings, challenges faced, and insights derived from the performance metrics of both models.The recent innovations in deepfake detection approach have been centred around various deep learning models, each having their strengths when it comes to performance and robustness. The corresponding studies were conducted utilising the 140k-point dataset containing the records of real and fake faces. The present paper provides a brief discussion of the previously mentioned research highlighting their approaches and comparative performance.

GenConViT, It was one of the first models to combine the strengths of convolutional networks and transformers, and as such, it was able to deliver on its promise of facilitating the deepfake detection process. The model was using ConvNeXt and Swim Transformer models for feature extraction with the corresponding Autoencoder and Variational Autoencoder applied to leverage the learned latent distribution of the data. Assessment of the system performance by using the DFDC, FF++, DeepfakeTIMIT, and Celeb-DF v2 datasets highlighted its promising average accuracy value of 95.8%, with AUC standing at 99.3% (Liu et al., 2023). The performance indicated the effectiveness of the system regarding effective feature extraction and integration that is crucial for generating a relatively high accuracy.

CNNs vs Transformers , This study compared the performance of CNNs and transformer-based models and the results were calculated by using multiple datasets, FF++ 2020, Google DFD, Celeb-DF, Deeper Forensics, and DFDC. The results were reported in the form of AUC as CNNs resulted in the following values – 99.99%, 99.95%, and 100% compared to the transformer's 99.99% (Thing, 2023). The results highlighted that the transformer outperformed the competitor owing to these models' ability to handle sequential data and leverage temporal dependencies for identifying inconsistencies in the frames.

Hybrid Approaches with Biological Signals, Another novel approach of combining CNNs with the biological signal detection to improve the deepfake detection using physiological signals of humans such as heartbeat, breathing pattern which have been proven to be extremely challenging to replicate making it harder for the deepfake algorithms to generate convincing forgeries. An AUC score of 99.6% and accuracy of 97.1% has been achieved with the hybrid model (Li and Chen, 2023). In spite of the superior performance, the high quality data requirements and computational intensity of the method make it hard for it to be deployed in real-time scenarios. Generalisation and Robustness Transformer models have shown better results in terms of handling the temporal data which is necessary for the video-based deepfake detection. Besides, the transformer models capture the long-range dependencies across the data which is necessary for detecting the artificial images between the frames. Being able to capture the long-range dependencies allows the transformer to be

more sensitive to detect even the small changes between the frames as the inputs. Computational Efficiency Even though the hybrid models provide better results in image recognition, relatively longer run times should be expected and that makes it difficult for real-time applications. The transformers and the CNNs optimised displayed better characteristics in terms of the inference time which is compatible with the real-time applications. Data Quality and Diversity, The models also trained and tested with the different data have displayed different performances which means that training the model using data similar to the test data is critical in ensuring the generalisation of these models. Training the models such that they are capable of generalising the multi-honed ability of creating deepfake content can ensure better results in the practical scenarios. Training more general models and deploying them to work in real-time scenarios are two critical areas which have the highest importance which can be used to provide insights and comparisons.

7.2 Ethical Considerations

Demystifying the Ethics of Detecting Deepfakes - Continued Analysis,

Starting with Privacy Concerns, Deepfake detection requires sifting through billions of unique images, videos, and audio recordings This could violate the privacy of people, particularly if it is collected without permission. Finally, the sensitivity of data or when children are involved can raise privacy concerns even more. Basic level of generalisation and strong data protection are needed to keep these risks at bay. A detection system that scans social media for deepfakes may also pick up private, friendly interactions between people which includes personal photos and images (House of Lords 2020; Schick, 2020). Next comes the Detection technology misuse, Detection technology can be used at scale to detect deepfakes; In other contexts, authoritarian governments might use the deepfake to delete political dissent as fake. On the other hand, bad actors could create anti-detection techniques to avoid detection altogether - a cat-and-mouse game between deepfake creators and detection technologies. For example, a government during political protests could detect deepfake technology to make genuine footage of police brutality inherently fake in relevant content (Nisar et al., 2020; Schick, 2020) and thus undermine the public outcry as well accountability.

Effects on Trust and Authenticity**,** Deepfakes: These are becoming increasingly more common and the threat is that they damage our trust in digital media, taking away our ability to tell real from fake. The very existence of deepfakes can cause doubt in the truthfulness of traditional media despite advanced detection tools. These are all examples of the so-called "liar's dividend" whereby fraudsters can point at real footage and claim this, too, is smoke and mirrors. Example: Politician is caught in embarrassing video, calls it deepfake to use wider suspicion at public expense of holding accountable. Conversely, evidence in legal cases can be dismissed as potentially tampered-in that hampering the judicial process (Nisar et al., 2020; Schick, 2020; Stehouwer et al. 2021). One of the most crucial concerns Bias and Fairness, Detection algorithms may reinforce biases in their training data rather than compensating for them, which can cause detection outcomes to be unfair. A biassed dataset

can hurt the prediction performance on a minority class. This could additionally worsen present inequalities and might create an unequal approach toward being protected from deepfakes. For example, the article House of Lords (2020); Nisar et al., (2020) stated that a deepfake detection technique majorly trained with images of Caucasian might not end up detecting as much threat on POC and could result in an imbalance capacity between different race effectiveness.

Legal and Regulatory Challenges, Deepfakes are still more or less as new to the legal framework! Such laws may still not be well-suited to handle the unique problems presented by deepfake technology, including rights to likeness control and prohibitions on unauthorised uses. This is the difficult and precarious balancing of how much protection for these ugly claims can be squared with freedom of speech, parody or satire. Example: Courts may be faced with cases concerning deepfakes in satirical content spanning the spectrum from clearly protected speech to unambiguously false, harmful misinformation. Those in charge are doing all they can to craft legal rules that safeguard us against deepfakes while also protecting free speech (House of Lords, 2020; Schick, 2020).

Legitimacy in Using Detection Results, After the detection of a deepfake, it becomes an ethical issue to decide what to do with such information. In other words, if a deepfake is detected should it be disclosed so the public knows or depending on potential harm should some level of discretion prevail? While transparency is important, so too is the protection of people from serious harm. News Organization discovers celeb deepfake making disparaging comments. The question of whether or not to publish this discovery is therefore one that pits the public interest in knowing against the individual's concerns that having such information released might be damaging to their reputation. But, in navigating those situations ethically guidelines are needed (Nisar et al 2020; Schick 2020).

7.3 Policy Implications

The application of deep learning models, whether they are ResNet and DenseNet, to the detection, and recognition of deepfakes involves various policy implications on multiple levels. The integration of these technologies has a tremendous effect on the overall progress and functioning of multiple sectors. In the case of media and journalism, deep learning models can help prevent and combat fake news. They will provide news organisations with an opportunity to verify the reality of the information that they want to cover in videos or images. The cited application is extremely beneficial to the public, given the fact that misinformation has a proven detrimental impact on public opinion. The audiences involved in such processes will receive firstly verified and secondly accurate information that will help them contribute to a more well-informed discussion. On one hand, while such changes might not necessarily prevent committed schemes with the intention of successfully spreading fake news, the application of deep learning models for media purposes will have a positive effect. Media organisations will become more responsible and credible, thus achieving a

higher-level status. On the other hand, these systems might be extremely profitable in the case of law enforcement units and national security operations that demand the use of information provided in the form of videos and images as evidence. They will prevent any alteration, manipulation, or faking of the current data and achieve a successful investigation and further prosecution. As for foreign propaganda, those systems will aid in the detection and prevention of massive-scale attacks that have a twisted message.

In the entertainment industry, deepfake detection technologies play an essential role in ensuring the protection of intellectual property and likeness rights. Namely, they can help avoid unauthorised use of a person's appearance in synthetic content that may be detrimental. Second, the use of these technologies permits studios to engage in an ethical use of deepfake technologies in their creative products. This is vital in light of potential reputational damage and lawsuits that may arise from synthetic media involving celebrities and other artists. The financial sector may also benefit from the application of these models in the field of fraud. Such technology can be used to verify the identity of the person making a transaction, thus protecting against fraud that might involve deepfake videos or synthetic identities. Overall, deepfake detection models can help protect the realm of digital transactions on the part of both financial institutions and their customers. Therefore, their use is closely linked with the broader significance of such technologies in securing transactions and communications on the digital realm. Governments can use these models to secure their public communications while verifying their accuracy. By preventing the distribution of synthetic media, authorities can avoid the subsequent undermining of public trust and social stability that tends to follow such publications. In the broader context, it may also be possible to create legislation that mandates such tools to be used in critical sectors. This would essentially create a uniform approach to the control of the synthetic media threat. Finally, the use of these models can be integrated into educational programs to ensure citizens and especially young people receive the tools necessary to recognize deepfakes. This is essential for the digital literacy of the general population.

Overall, the implementation of efficient deepfake detection models such as ResNet, DenseNet is truly transformative. It is beneficial to a wide range of industries and contributes to the authenticity of digital media, fraud prevention as well as public confidence. At first glance, the model would not only become a significant technical concern, but also truly needed from the social perspective. The safe digital environment and trust on basic rules require the co-implementation of those technologies between industry, government and civil society.

7.4 Conclusion

To sum up, the presented study investigated the efficacy of such deep learning models as DenseNet and ResNet in the detection of deepfake content. As part of the study, the models were applied to a comprehensive dataset with the analysis of the provided architectures, and respective results were derived. Both models showcased impressive results owing to the capacity of the architectures and the extent of the dataset. Specifically, the maximum

AUC-ROC indicator of 0.998976 obtained by DenseNet serves as evidence to the model's relatively superior sensitivity and selectivity, whereas ResNet's 98% of accuracy provides the proof of the model's usability with regard to real-world applications in which the reliability of results is critical. Furthermore, the comparison analysis demonstrates that, despite the potential of DenseNet's deeper connections to detect relatively subtle inconsistencies in synthetic media, ResNet's simpler formulation allows preserving a better balance between the effectiveness of detections and the cost of computations. Therefore, such analysis is further supported by the application of temporal analysis techniques represented by LSTM that allowed detecting dynamic inconsistencies in video frames that were characteristic of deepfakes with noticeable forms of temporal artefacts. The analysis included the examination of the ethical implications of deepfake detections, highlighting the aspect of responsible use of the technology to prevent exposing individuals to risks and the potential for abuse.

In this way, the results of the study contribute to a better understanding of AI-based deepfake detection and prompts future discussions surrounding the phenomenon. Conclusively, it can be argued that the research was successful in pinpointing the strengths and potential of the selected types of models with regard to deepfake detection. Meanwhile, various opportunities for further research development in the field exist, including the improvement of the general applicability of models, more diversified dataset, and possible real-time application of the developed technologies. Overall, sustaining the evolution of technologies allowing to detect fake content is vital in the context of the current rapid development of methods that allow for deepfake creation.

8. LIMITATIONS AND FUTURE WORKS

8.1 Limitations

Although the study has yielded impressive results, several limitations have been identified. One of the major limitations is computational intensity. To say more, some deepfake detection methods require a lot of computational power to function when the models are DenseNet and ResNet. They consume a lot of power and require a significant amount of memory, which is not very feasible for real-time applications. Next, one more limitation is data quality and diversity. The quality and diversity of the data used to support and test models significantly impact their performance. This suggests that when the models are used in realistic settings, they cannot prove to be useful encountering altered fakes generated with the use of modulated techniques that have not been included in training datasets.

One more potential limitation is high data quality. The thing is that the required large datasets of deepfake and real media at the highest quality may not be easily accessible since data collection is a time- and resource-consuming process. Finally, the limitation of the models related to their generalisation can be stated. That is, when the deepfakes are very different

from those previously shown to the models, they may not give relevant results, which is not very useful for “real-world situations.” Another limitation is related to ethical and privacy aspects since with the use of the data to be analysed, the privacy of data owners can be violated, as a lot of personal information should be scrutinised. Even more, some authorities can utilise the employed analyses to mark the actually accurate information as false using detection algorithms, and this is going to be a very serious limitation for many ordinary users who are using such algorithms every day.

8.2 Future Works

The field of deepfake detection provides a plethora of exciting opportunities for future researchers. First and foremost, the detection model should be improved to reduce the possibility of error and accelerate the process. The improved detection models must retain or potentially reduce the number of computational loads required. Such models will make use of advanced machine learning algorithms and possibly new designs that are better at managing a larger amount of data without slowing down the system. Transformers and attention mechanisms can be used to improve the quality of the detector output, potentially making it more useful in real time.

Creation of new and more effective datasets will be another critical direction in the future. The problem is that the current datasets may not accurately portray the situation in real life or potentially reflect the limitations of the technology. The new datasets will include a broader range of deepfake types, potentially closing the gap in the creation of new detection techniques capable of addressing newly created types of videos.

Furthermore, research into ethical and social issues created by the deepfake detection technology will continue to be an important direction of future studies. A significant effort will likely be spent on creating new rules and guidelines for the technology, including a new form of multilateral agreements between parties that will ensure that the technology is applied lawfully and ethically. Research in this direction should be accompanied by an attempt at interdisciplinary cooperation to make the legal, ethical, and social aspects of deepfakes and their detection known to the public. Finally, future researchers should consider the possibility of creating a new framework and a set of open-source tools and guidelines that are based on the shared public knowledge on how to avoid deepfakes.

9. REFERENCES

Afchar, D., Nozick, V., Yamagishi, J. and Echizen, I., 2018. MesoNet: a Compact Facial Video Forgery Detection Network. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. https://doi.org/10.1109/wifs.2018.8630761.

Amerini, I., Galteri, L., Caldelli, R. and Del Bimbo, A., 2019. *Deepfake Video Detection through Optical Flow Based CNN*. [online] openaccess.thecvf.com. Available at: <https://openaccess.thecvf.com/content_ICCVW_2019/html/HBU/Amerini_Deepfake_Video_Detection_through_Optical_Flow_Based_CNN_ICCVW_2019_paper.html?ref=https://githubhelp.com>.

Dang, H., Liu, F., Stehouwer, J., Liu, X. and Jain, A.K., 2020. *On the Detection of Digital Face Manipulation*. [online] openaccess.thecvf.com. Available at: <https://openaccess.thecvf.com/content_CVPR_2020/html/Dang_On_the_Detection_of_Digital_Face_Manipulation_CVPR_2020_paper.html>.

Gong, L.Y. and Li, X.J., 2024. A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges. *Electronics*, [online] 13(3), p.585. https://doi.org/10.3390/electronics13030585.

Guera, D. and Delp, E.J., 2018. Deepfake Video Detection Using Recurrent Neural Networks. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. https://doi.org/10.1109/avss.2018.8639163.

Haliassos, A., Vougioukas, K., Petridis, S. and Pantic, M., 2021. Lips Don't Lie: a Generalisable and Robust Approach to Face Forgery Detection. *Thecvf.com*, [online] pp.5039–5049. Available at: <https://openaccess.thecvf.com/content/CVPR2021/html/Haliassos_Lips_Dont_Lie_A_Generalisable_and_Robust_Approach_To_Face_CVPR_2021_paper.html> [Accessed 4 August 2024].

He, K., Zhang, X., Ren, S. and Sun, J., 2015. *Deep Residual Learning for Image Recognition*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1512.03385>.

Huang, G., Liu, Z., van der Maaten, L. and Weinberger, K.Q., 2017. *Densely Connected Convolutional Networks*. [online] openaccess.thecvf.com. Available at: <https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html>.

Korshunov, P. and Marcel, S., 2018. DeepFakes: a New Threat to Face Recognition? Assessment and Detection. *arXiv:1812.08685 [cs]*. [online] Available at: <https://arxiv.org/abs/1812.08685>.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, [online] 25, pp.1097–1105. Available at: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B., 2021. *Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows*. [online] openaccess.thecvf.com. Available at: <https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper>.

Marra, F., Gragnaniello, D., Cozzolino, D. and Verdoliva, L., 2018. *Detection of GAN-Generated Fake Images over Social Networks*. [online] IEEE Xplore. https://doi.org/10.1109/MIPR.2018.00084.

Masi, I., Killekar, A., Mascarenhas, R.M., Gurudatt, S.P. and AbdAlmageed, W., 2020. Two-Branch Recurrent Network for Isolating Deepfakes in Videos. *Computer Vision – ECCV 2020*, pp.667–684. https://doi.org/10.1007/978-3-030-58571-6_39.

Mathews, S., Trivedi, S., House, A., Povolny, S. and Fralick, C., 2023. An explainable deepfake detection framework on a novel unconstrained dataset. *Complex and Intelligent Systems*. https://doi.org/10.1007/s40747-022-00956-7.

Sabir, E., Cheng, J., Jaiswal, A., Abdalmageed, W., Masi, I. and Natarajan, P., 2019. *Recurrent Convolutional Strategies for Face Manipulation Detection in Videos*. [online] Available at: <https://openaccess.thecvf.com/content_CVPRW_2019/papers/Media%20Forensics/Sabir_Recurrent_Convolutional_Strategies_for_Face_Manipulation_Detection_in_Videos_CVPRW_2019_paper.pdf>.

Simonyan, K. and Zisserman, A., 2015. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1409.1556>.

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*. [online] Available at: <https://arxiv.org/abs/1602.07261>.

Yu, N., Davis, L.S. and Fritz, M., 2019. *Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints*. [online] openaccess.thecvf.com. Available at: <https://openaccess.thecvf.com/content_ICCV_2019/html/Yu_Attributing_Fake_Images_to_GANs_Learning_and_Analyzing_GAN_Fingerprints_ICCV_2019_paper.html>.

Zou, H., Shen, M., Hu, Y., Chen, C., Chng, Eng Siong and Rajan, D., 2024. *Cross-Modality and Within-Modality Regularization for Audio-Visual DeepFake Detection*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2401.05746v1> [Accessed 4 August 2024].

Chollet, F., 2018. Keras: The python deep learning library. Astrophysics source code library, pp.ascl-1806.

Thing, V.L., 2023. Deepfake detection with deep learning: Convolutional neural networks versus transformers. In: *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 246-253. IEEE.

H. H. Nguyen, F. Fang, J. Yamagishi and I. Echizen, "Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos," 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 2019, pp. 1-8, doi: 10.1109/BTAS46853.2019.9185974.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... and Bengio, Y., 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, pp. 2672-2680. Available at: <https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.

Chesney, R. and Citron, D., 2019. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Affairs*. Available at: <https://www.foreignaffairs.com/articles/world/2018-12-11/deepfakes-and-new-disinformation-war>.

Westerlund, M., 2019. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9(11), pp. 39-52. Available at: <https://timreview.ca/article/1282>.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, 521(7553), pp. 436-444. Available at: <https://www.nature.com/articles/nature14539>.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. Available at: <https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html>.

Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625-2634. Available at: <https://openaccess.thecvf.com/content_cvpr_2015/papers/Donahue_Long-Term_Recurrent_Convolutional_2015_CVPR_paper.pdf>.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A. and Ortega-Garcia, J., 2020. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64, pp. 131-148. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S1566253520305163>.

Dolhansky, B., Howes, R., Pflaum, B., Baram, N. and Ferrer, C.C., 2019. The Deepfake detection challenge (DFDC) preview dataset. *arXiv preprint* arXiv:1910.08854. Available at: <https://arxiv.org/abs/1910.08854>.