National College of Ireland

DeepFakeCNN: Deep Fake Image and Video detection using Convolutional Neural Networks

MSc Research Project
MSc in Cybersecurity

Darshan Siddaiah
Student ID: x22187456

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Darshan Siddaiah		
Student ID:	22187456		
Programme:	MSc in Cybersecurity	**Year:**	2024
Module:	Practicum		
Supervisor:	Mr. Vikas Sahni		
Submission Due Date:	16/09/2024		
Project Title:	DeepFakeCNN: Deepfake Image and video detection using Convolutional Neural Networks		

Word Count:7907 **Page Count:** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Darshan Siddaiah

Date: 16/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DeepFakeCNN: Deepfake Image and video detection using Convolutional Neural Networks

Darshan Siddaiah
22187456
MSc. Cybersecurity
National College of Ireland

Abstract

Organisations have significant challenges in dealing with social cybercrimes and safeguarding against the spread of manipulated media due to the emergence of deepfake technology. This study presents an advanced deep learning system, "DeepFakeCNN", built for the purpose of detecting and alerting users about the presence of deepfake images and videos. The DeepFakeCNN model uses a convolutional neural network to accurately distinguish between genuine and falsely created pictures. Additionally, it provides monitoring and analysis capabilities to security personnel. The methodology may be readily employed by several social media platforms to detect deepfake videos, pictures, reels, and other similar content. This includes popular communication services such as Microsoft Teams, Google Meet, and Meta's WhatsApp/Messenger. It immediately provides instant notifications when suspicious deepfakes are found. In this EfficientNetB7 a convolutional neural network demonstrated superior performance in the evaluation, achieving an impressive accuracy of 93.99%. This model demonstrated balanced performance by correctly distinguishing between genuine and fake images and videos, achieving a recall rate of 75.33%, a precision rate of 74.34% and F1 score of 74.83%.

Keywords: Deep Fakes Images, Image detection, Image Pre-Processing, Convolutional Neural Networks

1 Introduction

The continuous advancement of artificial intelligence (AI) technology poses significant challenges to cybersecurity and social integrity because of the emergence of deepfake technology. Deepfakes, which are artificially generated images, videos, sounds, or text produced using AI technology, are progressively improving in sophistication and are easier to make (Dudykevych, V., et. al., 2024). This presents a new opportunity for cyber attackers to manipulate images and videos in social media. This research project intends to investigate techniques for identifying and minimizing the risks associated with deepfake images and videos, specifically in relation to cybercrimes and social engineering, due to their increasing threat. The widespread and quick adoption of deepfake technology has detrimental effects on cybersecurity, identity theft, and the spread of false information. Developing strong approaches for recognizing and fighting deepfakes is crucial because to their increasing frequency and believability (Volkova, S.S., 2023). The objective of this project is to enhance individuals' and organizations' ability to safeguard themselves against the risks associated with deepfake images by comprehending the fundamental processes involved in deepfake creation and researching effective detection methods.

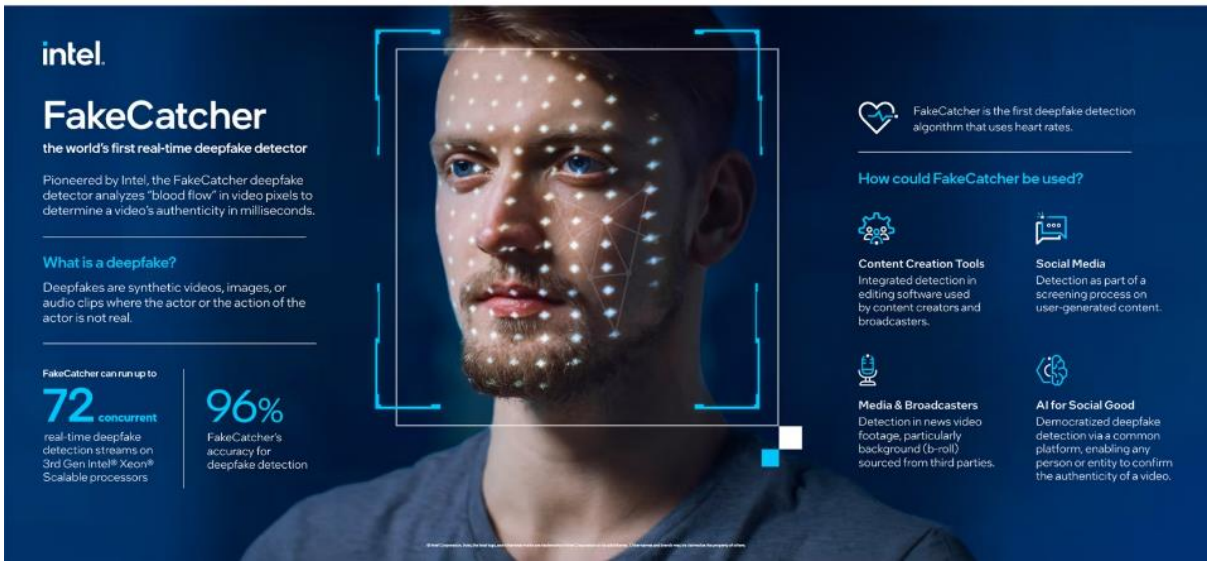


Figure 1: FakeCatcher, an Intel based fake images detection technique using the image processing and different computer vision solutions[1]

This work investigates the use of deep learning techniques in detecting and reducing the prevalence of altered photographs, commonly referred to as deepfakes, in the domains of cybersecurity and social contexts (Patel, Y., et. al., 2023). This research aims to improve cybersecurity practices and protect digital integrity in the face of growing AI technology by studying the core processes of deepfake development, assessing existing detection tools, and examining potential remedies.
This research aims to compare and train various CNN and RNN models for identifying deepfake images and videos, with a focus on improving accuracy and reducing false positives.

1.1 Research Question

The following is the proposed research question: "What are the most effective deep learning models for identifying and classifying deepfake images and videos, and how can these techniques be optimized to enhance accuracy and efficiency in cybersecurity?"

Solution to Address: Detection of deepfake images and videos with high accuracy, training and comparing different deep learning models.

1.2 Thesis Structure

The next section will provide a detailed discussion of previous research on deepfake detection, followed by an outline of the methods and techniques to be utilized in this study. The implementation chapter will cover the detailed process of model implementation, algorithm flow. The results and analysis chapter will present the findings across various cases and scenarios. Finally, the conclusion chapter will summarize the findings and propose for future work.

[1] https://www.intel.com/content/www/us/en/newsroom/news/intel-introduces-real-time-deepfake-detector.html

2 Literature Review

The proliferation of mobile camera technology and the rise of social media platforms (Adnan, S.R. and Abdulbaqi, H.A., 2022) for sharing have greatly facilitated the creation and dissemination of digital videos. Nevertheless, the prevalence of video modification and fabrication has diminished in recent years, thanks to the advancements in machine learning and computer vision techniques. This study employs the detection technique of comparing the regions of the produced face and their surrounding areas using the Convolutional Neural Network (CNN) Model (Adnan, S.R. and Abdulbaqi, H.A., 2022). The model was utilised on the DFDC dataset, consisting of 60 distinct clips for both actual and false videos. The methodology of this work consists of three stages. The first stage involves preprocessing, where each video is converted into frames and the face in each frame is detected and cropped using the Haar Cascade function. In the feature extraction stage, the ResNet-50 model is utilised for extracting features.
The fast advancement of deepfake production technology poses a severe challenge to the credibility of media content. The repercussions affecting specific persons and institutions might be severe. This study focuses on analysing the progressions of deep learning architectures, namely Convolutional Neural Networks (CNNs) and Transformers. They assess the efficacy of our newly created single model detectors in detecting deepfake content and conducting assessments across several datasets (Thing, V.L., 2023, July).

Previous surveys have primarily concentrated on detecting deepfake images and videos. This study aims to provide readers with a comprehensive overview of the creation and detection of deepfakes, as well as their existing limits and potential directions for future research (Masood, M., et., al., 2023). Therefore, it is imperative to verify the authenticity of the digital photos. Deepfake pictures, a novel form of counterfeit photographs, are created using generative adversarial networks (GANs). These deepfake pictures pose a greater threat because of their highly realistic looks. This study examines various techniques for identifying deepfake images generated by Generative Adversarial Networks (GANs). (Remya Revi, K., Vidya, K.R. and Wilscy, M., 2021). The abstract (Mishra, A., et. al., 2024) explores the innovative field of DeepFakes, which combines deep learning with synthetic media in a novel way. The core of DeepFake creation relies on the use of Generative Adversarial Networks (GANs), namely the advanced techniques of face reenactment using DCGANs (Deep Convolutional GANs) and Autoencoders. The discussion explores the intricate balance between artistic liberty and ethical application, emphasising how DeepFakes, derived from sophisticated deep learning methods, reshape our understanding of artificial media, questioning the concept of truth in our more digitised society (Mishra, A., et. al., 2024). It emphasises the recent change in research focus towards developing methods to identify the origin of AI-generated pictures by providing supporting evidence. Robust, comprehensible, and transferable attribution techniques would ensure that malevolent users be held responsible for AI-fueled misinformation, provide plausible deniability to rightful users, and aid in safeguarding the intellectual property of deepfake technology (Bansal, N., et. al., 2023).

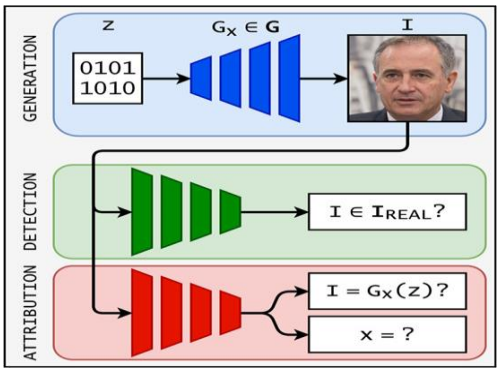


Figure 2: An example of (Bansal, N., et. al., 2023) to generate the fake images

Deep learning has been applied extensively in several domains, including computer vision, natural Deepfakes utilise advanced deep learning algorithms to generate counterfeit pictures that may be extremely difficult to differentiate from genuine ones. This paper examines the use of deep learning techniques for generating and identifying deepfake images, in response to the growing concern over personal privacy and security. Additionally, it suggests employing deep learning image enhancement methods to enhance the quality of the generated deepfakes (Khalil, H.A. and Maged, S.A., 2021). Although several approaches have been employed in the past to address the problem, the computational expenses remain substantial, and a truly efficient model has not yet been devised. Thus, they have introduced a novel model structure called DFN (Deep Fake Network), which incorporates the fundamental components of mobNet, a sequential arrangement of separable convolution and max-pooling layers with Swish as the activation function.

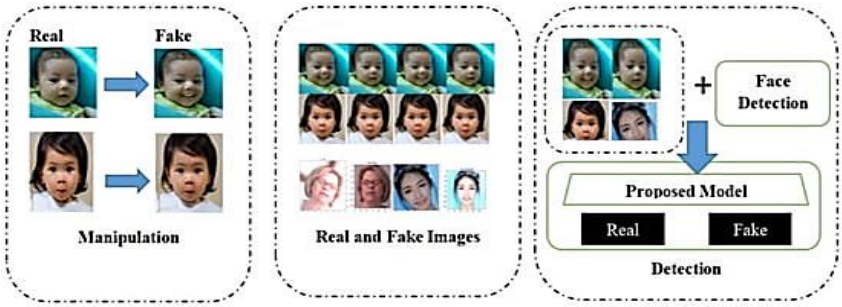


Figure 3: Deep Fake creation and detection model (Bansal, N., et. al., 2023)

The proposed strategy attained a 93.28% accuracy and a 91.03% precision when applied to this dataset. Furthermore, the training loss was 0.14, whereas the validation loss was 0.17. Additionally, various forms of face alterations have been addressed, leading to a model that is more resilient, adaptable, and efficient. This model is capable of detecting all types of facial alterations in videos (Bansal, N., et al., 2023). The instantaneous generation of intricate deep fakes, on the other hand, is causing more scepticism towards even real-time video conversations. Real-time detection of deepfakes presents unique obstacles in contrast to offline forensic analysis.

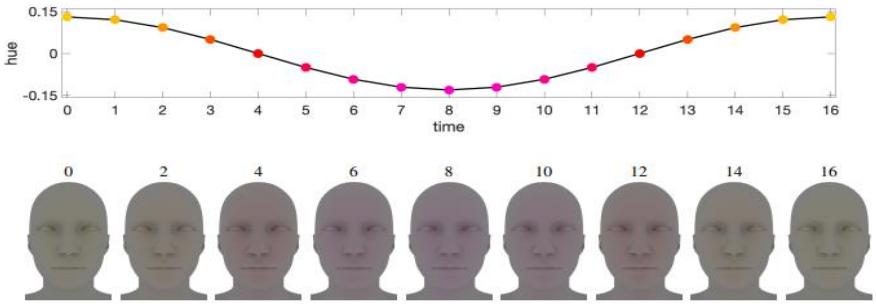


Figure 4: The top panel displays a visualisation of the changing hue of a uniformly coloured area light source, which is meant to simulate a computer screen. Presented here are nine representations of a three-dimensional model that is lighted with a unique colour from a light source at nine separate points in time. This simulation has a face with a uniform reflectance and an equal ratio of area-to-ambient light intensity. To enhance visualisation, the picture saturation was increased by 50% (Gerstner, C.R. and Farid, H., 2022).

The Fisherface algorithm is employed for face recognition by reducing the dimensionality of the face space using the Local Binary Patterns Histograms (LBPH) technique. Next, utilise Deep Belief Networks (DBN) in conjunction with Restricted Boltzmann Machines (RBM) to create a classifier specifically designed for detecting deep fake images or videos. The datasets utilised in this study are FFHQ, 100K-Faces DFFD, and CASIA-WebFace (Suganthi, S.T., et. al., 2022).

In conclusion the current literature focuses on both, creation of deep fakes and their detection. Both are still ongoing processes due to the constant advance in the field of deep learning, especially CNNs and GANs. The experts have taken an interest in various detection approaches wherein machine learning and computer vision can help in improving the reliability of the new media. From the reviewed papers, it can be noted that the CNN models such as ResNet-50, DFN along with other models have the capability to detect deepfakes. Considering the prospects of deepfake technologies' evolution and the challenges, this research will progress by employing a range of CNN and RNN models for detecting deepfake images and videos with the aim to contribute to the detection of fake and manipulated content in digital media.

2.1 Summary Table

All the above research is summarized in the table below.

Paper Name	Author Names	Dataset Used	Algorithms Used	Research Problem	Research Results
Deepfake video detection based on convolutional neural networks.	Adnan, S.R. and Abdulbaqi, H.A.,	DFDC dataset with different 60 clips for real and fake	Haar Cascade function with ResNet-50	The detection approach utilizes a Convolutional Neural Network (CNN) Model to compare the regions of the produced face with their surrounding areas.	With a detection accuracy of 98%, the Deepfake detection algorithm was able to identify the false face in the video.
Real-time advanced computational intelligence for deep fake video detection	Bansal, N., Aljrees, T., Yadav, D.P., Singh, K.U., Kumar, A., Verma, G.K. and Singh, T.	DFDC (Deep Fake Detection Challenge) dataset	The DFN (Deep Fake Network) uses XGBoost as a classifier to identify deepfake movies, and it contains the fundamental building elements of mobNet, like a linear stack of separable convolution, max-pooling layers activated by Swish.	Decreasing the computation and computational costs	The proposed strategy attained a 93.28% accuracy and a 91.03% precision when applied to this dataset. Furthermore, the training loss was 0.14, whereas the validation loss was 0.17. Additionally, various forms of face alterations have been addressed, enhancing the model's resilience, versatility, and efficiency, allowing it to detect all types of facial alterations in videos.
Detecting real-time deep-fake videos using active illumination	Gerstner, C.R. and Farid, H.	Simulates dataset using Mitsuba and real time dataset using different	a simple, dynamic, colored square on the display and then measuring the temporal impact	Real-time detection of deepfakes presents unique obstacles in contrast to offline forensic analysis.	The real-time measurement of deviations from the anticipated alteration in appearance can be employed to authenticate the identity of a

		skin tones for 15 users			participant in a video conversation.
Deepfakes-Generating Synthetic Images, and Detecting Artificially Generated Fake Visuals Using Deep Learning	Mishra, A., Bharwaj, A., Yadav, A.K., Batra, K. and Mishra, N.,	Synthtic Media	DCGANs (Deep Convolutional GANs) and Autoencoders	To convert arbitrary noise into highly realistic pictures, including subtle aspects like face emotions and lighting conditions via latent space interpolation.	These methods enable a generator to convert unpredictable noise into highly realistic pictures, capturing precise aspects like face expressions and lighting conditions by using latent space interpolation.
Deep fake detection and classification using error-level analysis and deep learning	Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A. and Alshehri, A.H.,	Dataset by Yonsei University's Computational Intelligence and Photography Lab	Error Level Analysis fed into Deep Neural Networks for the feature extraction with residual networks and K-NN.	An effective technique for distinguishing authentic from counterfeit information has become essential in the era of social media.	The suggested technique attained the utmost accuracy of 89.5% using Residual Network and K-Nearest Neighbor. The findings demonstrate the efficacy and resilience of the suggested methodology, therefore making it suitable for identifying deep fake pictures.
Deep Fake Detection Using Computer Vision-Based Deep Neural Network with Pairwise Learning	Saravana Ram, R., Vinoth Kumar, M., Al-shami, T.M., Masud, M., Aljuaid, H. and Abouhawwash, M.	Face2Face, FaceSwap, images using StyleGANs and DeepFake by kaggle	FC-DBNPL: Preprocessing using a Gabor filter-based Gaussian rule with the deep belief network classification algorithm known as paired learning.	An artificial intelligence-generated image or video created for the purpose of political manipulation, dissemination of false information, or pornography.	This proposed technique has significantly enhanced the accuracy of the detection rate by 98% across the datasets.

Table 1: Summary of different research done

3 Research Methodology

Deepfakes employ sophisticated deep learning methodologies to initially encode characteristics, followed by the reconstruction of pictures based on the encoded features. Autoencoders, a neural network design, are widely employed in deep learning for generating deepfakes. Given the widespread accessibility of deepfake creation programs, it is crucial for individuals to possess a fundamental comprehension of how to detect a deepfake. Indeed, firms such as Google, Amazon, and Meta have been aggressively promoting the analysis and comprehension of the distinguishing characteristics of deepfakes throughout the community. This research will focus on detecting deepfake images and videos on social media, aiming to integrate these detection techniques into a cybersecurity framework.

3.1 Dataset Acquisition

DFDC (DeepFakes Detection Challenge): AWS, Facebook, Microsoft, the Partnership on AI's Media Integrity Steering Committee, and academics have come together to build the Deepfake Detection Challenge (DFDC). The primary objective of this initiative is to expedite the advancement of novel techniques for identifying deepfake videos. As a result, an exclusive dataset for the challenge was distributed.

Figure 6: Examples showcasing the face swaps in the dataset
(Dolhansky, B., Howes, R., Pflaum, B., Baram, N. and Ferrer, C.C., 2019.)

This collection has both modified and real videos. The modified videos are the result of creating faces using various methods. The dataset was obtained from the following source[2]. The dataset will be managed strategically to maximize the output from these videos.

Figure 7: A sample of images showing the fakeness and actual image

3.2 Data Sampling

The holdout method will be used to create training and testing samples. There are essentially two types of data sampling techniques:

Holdout: The hold-out approach entails dividing the data into various segments, utilizing one segment for model training and the remaining segments for validation and testing purposes. It has the capability to be utilized for both the assessment and selection of models.

K-Fold Cross Validation: K-fold cross-validation involves partitioning the dataset into K-folds and using them to evaluate the model's performance when new data is introduced. K denotes the quantity of divisions in which the data sample is partitioned. For instance, when the value of k is determined to be 5, it might be referred to as 5-fold cross-validation. Every fold is utilized as a test set at a certain stage in the process.

[2] https://www.kaggle.com/c/deepfake-detection-challenge/data

3.3 Modelling

Deep learning is a very efficient and valuable method that has been extensively utilized in several domains, such as computer vision, machine vision, and natural language processing. Deepfakes employs advanced deep learning techniques to modify photos and videos of an individual in a manner that is indistinguishable from genuine content by human observers. Recently, several studies have been carried out to comprehend the functioning of deepfakes, and several deep learning-based methods have been proposed to identify and discern deepfake movies or pictures. This study presents a thorough examination of deepfake generation and identification techniques, including deep learning methodologies. Additionally, a thorough examination of various technologies and their applications in deepfake detection will be provided. This study will benefit researchers by including the latest cutting-edge techniques for identifying deepfake images and videos on social media. Moreover, the detailed presentation of current techniques and datasets used in this field will offer a comprehensive comparison with existing literature.

3.4.1 Convolutional Neural Networks based Transfer Learning

By leveraging the acquired characteristics from the initial work as a foundation, the model may expedite and enhance its learning process for the subsequent assignment. This can also aid in mitigating overfitting, as the model will have already acquired broad traits that are likely to be advantageous in the second assignment. A frequent occurrence observed in many deep neural networks trained on pictures is that in the initial layers of the network, the deep learning model attempts to learn low-level properties such as edge detection, color recognition, and fluctuations in intensities. These characteristics do not seem to be exclusive to a single dataset or job, since they may be used for any form of image processing, whether it is for spotting a lion or an automobile. In both instances, it is necessary to identify these fundamental characteristics. These traits are present independent of the specific cost function or picture dataset. Therefore, acquiring knowledge of these characteristics in the context of lion detection may be used to other tasks such as human detection.

Pre-trained Model: Begin with a model that has undergone prior training for a specific task utilizing an extensive dataset. This model has undergone frequent training on huge datasets, enabling it to identify generic traits and patterns that are important to a wide range of related occupations. The base model refers to the pre-trained model. The structure consists of many layers that have utilized the incoming data to acquire hierarchical feature representations.

Layer transfers: Within the pre-trained model, identify a collection of layers that effectively capture general information that is pertinent to both the current task and the prior one. Due to their inclination towards acquiring low-level information, these layers are commonly located in the uppermost part of the network.

Fine Tuning: Fine-tuning involves retraining certain layers of a model using the dataset provided by a new challenge. This method is referred to as fine-tuning. The objective is to retain the information acquired during pre-training while allowing the model to adjust its parameters to better align with the requirements of the present task.

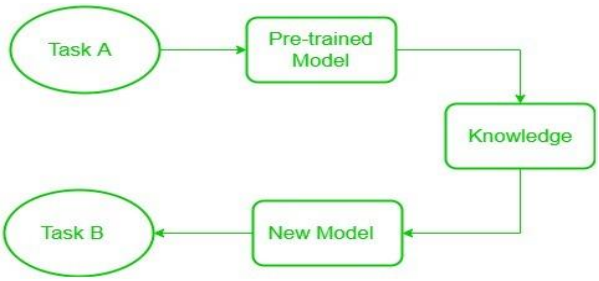


Figure 16: Transfer Learning[3]

[3] https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning

Different CNN models used in the research are as follows:

- **DenseNet121:** The DenseNet121 is a CNN that has dense connectivity between layers to increase the use of features as well as their dissemination across layers of the model. It has a network architecture which has 121 layers divided into dense blocks, transition layers and the final layer of global averaging followed by SoftMax classifier. They contain a sequence of convolutional layers linked densely, owing to which the flow of information is massive. Transition layers cover the spatial dimensions and the number of feature maps between the dense blocks, and often utilize batch normalization, 1x1 convolution and average pooling for dimensionality reduction as well as for down sampling.

- **InceptionResNetV2:** This is a new CNN architecture by Google that is the fusion of Inception and ResNet. It uses the inception module similar to in InceptionV3 where the convolution operations take place in parallel with the filter's different sizes and the max pooling allows assuming multiple scale features and hierarchical representation. Due to residual connections which are borrowed from the ResNet model, InceptionResNetV2 enables the exchange of information between the layers and thus reduces cases of gradient vanishing, a phenomenon that hampers the training of deep structures.

- **ResNET50:** ResNet50, the model that belongs to the family of ResNet from Microsoft Research. It is known for its ability to train deep neural networks because of its solution to vanishing gradient problem. It presents residual learning where shortcut connections are used, communications that would allow there to be direct interfaces between the layers.

- **VGG16:** VGG16 is an architecture of deep convolutional neural networks that is commonly used for image classification applications. The network is in fact built in 16 consecutive layers of artificial neurons which are all function to process the information of images at their level more incrementally of course to enhance the prediction result of the network.

- **EfficientNetB7:** EfficientNet is a convolutional neural network design and scaling approach that consistently scales all dimensions of depth, breadth, and resolution using a compound coefficient. The EfficientNet scaling approach differs from usual practice by evenly scaling network breadth, depth, and resolution using a predefined set of scaling coefficients, instead of random scales. EfficientNet uses a compound coefficient to systematically adjust the network's breadth, depth, and resolution in a consistent manner. The rationale for the compound scaling approach is based on the understanding that as the input picture is larger, the neural network requires additional layers to expand the receptive field and more channels to catch more intricate patterns present in the larger image.

3.4.2 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a form of Neural Network that utilizes the output from the previous step as input for the current phase. In conventional neural networks, each input and output are independent of one another. However, when it is necessary to anticipate the following word in a phrase, the preceding words are essential, therefore necessitating the retention of the prior words. As a result, Recurrent Neural Networks (RNN) were developed to address this problem by incorporating a Hidden Layer. The primary and paramount characteristic of RNN is its hidden state, which retains crucial information on a sequence. The state is commonly known as the Memory State because it retains information about the previous input to the network. The system employs identical settings for each input, executing a uniform operation on all inputs or hidden layers to generate the output. This feature simplifies the parameters, in contrast to other neural networks.

Different RNN models used in the research are as follows:

- **Gated Recurrent Unit (GRU):** GRU is an acronym for Gated Recurrent Unit and refers to a type of RNN that is a lot like LSTM, though not as complex. GRU has been written to address the issue of learning sequences since it is equipped with the ability to selectively

remember or forget specifics about the information as time passes. Although GRU is relatively lighter than LSTM, it has less number of parameters than LSTM and so it is less complex than LSTM but easier and more efficient to train.

- **LSTM:** Long Short-Term Memory Networks are a deep learning type of sequential neural network that has the astonishing ability to make information stay. It is a category of Recurrent Neural Network and well suits for resolving 'the vanishing gradient problem' which frequently challenges RNN. LSTM was formulated by Hochreiter and Schmidhuber to tackle the issue that was created by traditional RNNs and machine learning. LSTM Model can be implemented in Python and Keras and is suitable to use for this kind of recurrent problem.

4 Design Specification

The deepfake videos classification modelling is done based on a two-tier architecture. The first tier is the business layer where all data preprocessing and modelling is carried out, the second tier is the presentation tier where the results and insights gained are visualized for analysis. Figure 17 shows the design specification architecture for the deepfake videos sorting model.

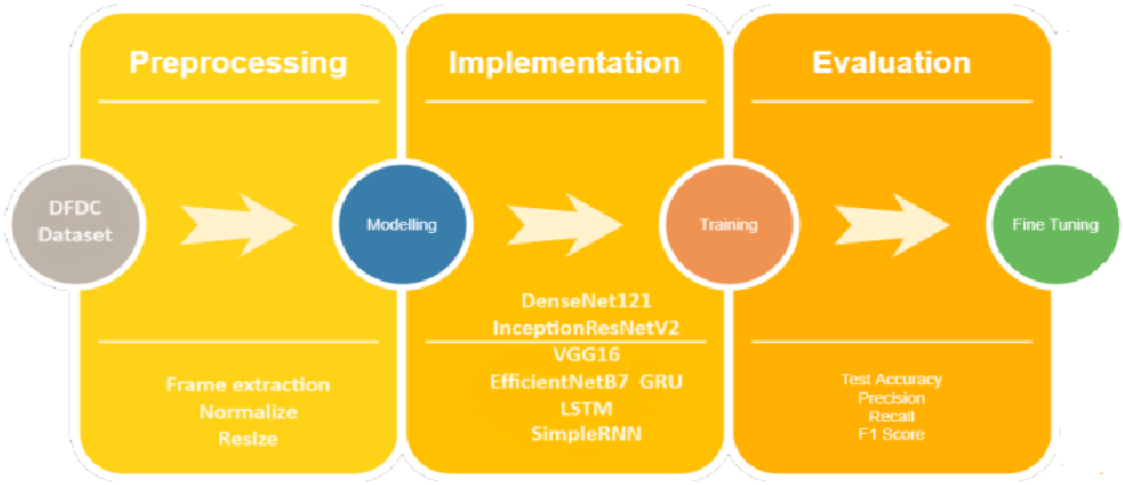


Figure 17: DeepFakes Detection using CNN and RNN

5 Implementation

This section will cover the implementation of the models, including the preprocessing of videos from the dataset before training, as well as the flow of the algorithms.

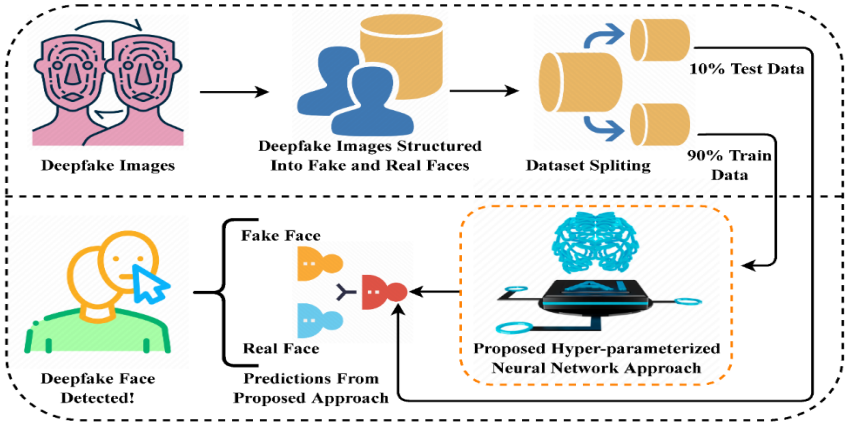


Figure 17: DeepFakes Detection using Convolutional Neural Networks Framework

5.1 Video Pre-processing steps

5.1.1 Frame Extraction

This was the first phase in our preprocessing pipeline, and it required extracting frames from the video files. Through the process of reading and processing each video, individual frames were captured at predetermined intervals. With videos, as opposed to photos, consisting of several frames that can reveal additional information over time, this was an extremely important consideration. Collecting these frames enabled the transformation of temporal data into a spatial format that can be utilized by Convolutional Neural Networks (CNNs). All the extracted frames were then classified as either "fake" or "real" based on the metadata that was provided in the JSON file that corresponded to each frame. The labelling was necessary for supervised learning, which allowed the models to acquire the ability to differentiate between false and real frames by learning the distinguishing characteristics.

5.1.2 Normalization

It was time to normalise the frames after extraction. Image pixel values are often normalised to a standard range, usually between 0 and 1, as part of common preprocessing procedures. To achieve this, the pixel values are divided by 255, which represents the maximum value for an 8-bit image. Training times and convergence rates are both improved by normalisation. In addition to making the neural network more stable and effective, it makes sure that the pixel values are all on the same scale, which stops the model from favouring greater pixel values.

5.1.3 Data Augmentation

Data augmentation techniques were employed to enhance the models' resilience and generalization capabilities. This process involves generating new training samples from existing data through various transformations. Several augmentation methods were utilized, including horizontal flipping, zooming, and rotating. Rotation was applied to ensure the model's independence from the orientation of faces in the frames. Zooming improved the model's ability to detect false features across different scales by exposing it to a range of magnifications. Horizontal flipping prevented the model from becoming biased towards either side of the face. These augmentation strategies significantly diversified the training data, which improved the models' generalization to new data and reduced the likelihood of overfitting.

Algorithm Flow

Step 1 – Data collection and preprocessing: Gathering a varied dataset comprising authentic as well as fake photographs and videos. Data preprocessing is essential to ensure uniform picture dimensions, color channels, and file formats. Additionally, expanding the dataset enhances diversity and resilience.

Step 2 – Extraction of features: Obtaining prominent characteristics from both authentic and fake images and videos. Refinement of the pre-trained models using the acquired dataset to customise them for the detection job.

Step 3 – Architecture of the model: A deep learning model architecture must be developed specifically for detecting deepfake content. Design considerations should include Siamese Networks, Generative Adversarial Networks (GANs), and Convolutional Neural Networks (CNNs). It is crucial to incorporate layers for feature extraction, information fusion, and classification.

Step 4 – Sampling Sets: The dataset must be divided into training, validation, and testing subsets. The training set will be used to train the model, while the validation set will be employed to evaluate performance and adjust hyperparameters to prevent overfitting. Finally, the testing set will be used to assess the model's ability to generalize.

Step 5 – Model Identification: The trained model should be used to analyze unfamiliar images and videos to detect instances of deepfake manipulation. The learned feature extraction layers will be employed to derive features from the input data. Based on these extracted features, the model will determine whether the input is genuine or fake. A predetermined threshold should be established for classification to control the rate of false positives and false negatives.

Post-processing refers to the stage in a process where additional actions or modifications are made to a product or data after it has undergone the initial processing. Post-processing methods should be utilized to enhance the accuracy and quality of detection results. Additionally, methodologies like temporal consistency analysis should be applied to identify and detect deepfake videos.

Step 6 – Enhancing the Model: The model needs to enhance the overall detection accuracy by implementing voting systems i.e. training different models on the same dataset and their predictions are combined in some way to produce a final prediction or ensemble approaches of combining multiple models to improve the overall performance.

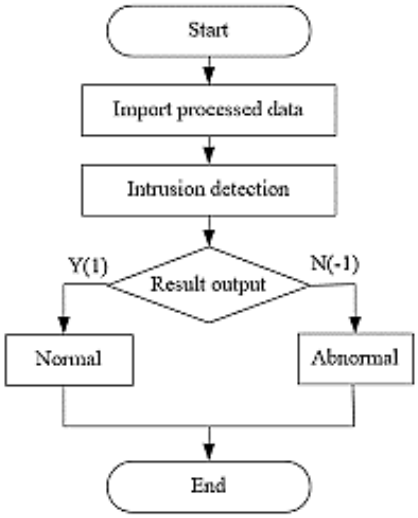


Figure 18: DeepFakeCNN implementation in the cybersecurity

6 Evaluation

The performance of each model was thoroughly assessed using multiple essential indicators on the validation set, specifically designed for our deepfake detection assignment. The selected metrics for evaluation are:

Precision – The percentage of deepfake frames that were detected as such out of all the frames that the model predicted as deepfake. This is important for our deepfake detection objective. With high precision, the model effectively minimizes the inaccurate labelling of genuine frames as deepfake, indicating a low false positive rate.

$$Precision = \frac{TP}{TP + FP}$$

TP - True Positive
FP - False Positive

Recall – Here, recall measures how well the model can spot all of the real deepfake frames in the dataset. With a low false negative rate and a high recall value, the model successfully captures most of the deepfake instances.

$$Recall = \frac{TP}{TP+FN}$$

F1 score – By combining precision and recall into a single statistic, the F1 Score strikes a good compromise between the two. It shows how well the model identifies deepfake frames overall, taking false positives and false negatives into account, for our deepfake detection.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

ROC AUC – One way to evaluate a model's performance in detecting deepfake frames is by looking at its Receiver Operating Characteristic Area Under the Curve, or ROC AUC. As a rule, a ROC AUC number closer to 1 implies that the model does a far better job of discriminating between the two classes than random guessing does, whereas a value closer to 0.5 indicates the opposite.

In this project, video data was used to develop and evaluate various deep learning models for detecting deepfakes. The dataset comprised video files, each labeled as either fake or real, with these labels detailed in the accompanying metadata JSON file. Frames extracted from these videos were then employed for training and evaluation purposes.

There were three columns in the JSON file that was included in the dataset:

Label: Indicates whether the video is a fabrication or a genuine one.

Split: It is a parameter that indicates whether the video is a part of the training set, or the validation set.

Original: When referring to fraudulent videos, this term refers to the original video file.

In a summary, JSON file contains 400 rows with 3 columns in which train_sample_videos contain 401 videos and test_videos contain 400 videos.

6.1 Case Study 1: Models Using CNN based Transfer Learning methods

Transfer learning was employed to evaluate various pre-trained Convolutional Neural Network (CNN) models for deepfake detection. The models assessed included **ResNET50, DenseNet121, InceptionResNetV2, VGG16, and EfficientNetB7**. By leveraging their pre-trained weights on the ImageNet dataset, the models' learned features were adapted for deepfake detection. The top layers of each model were replaced with new fully connected layers, allowing the models to classify "real" versus "fake" while retaining the robust feature extraction capabilities of the pre-trained networks. This fine-tuning approach enabled the models to adapt to the specific characteristics of deepfake videos. These measures allowed for a thorough assessment of how well each model detected deepfake frames. How successfully each model avoided mislabeling actual frames while accurately detecting deepfake frames was shown by the findings. To find the best model for the deepfake detection challenge, this thorough examination was conducted.

Model	Accuarcy	Precision	Recall	F1 score	ROC AUC
VGG16	0.805	0.7433	0.9266	0.8249	0.4833
ResNET50	0.75	0.75	1	0.8571	0.5
DenseNet121	0.75	0.75	1	0.8571	0.5
EfficientNetB7	0.9377	0.7434	0.7533	0.7483	0.4866
InceptionNetV2	0.75	0.75	1	0.8571	0.5

Table 2: Analysis of different transfer learning models on the deepfakes detection

With an accuracy of 93.99%, EfficientNetB7 stood out as the best model in the evaluation. By accurately identifying both actual and false videos, this model displayed balanced performance with a recall of 75.33% and a precision of 74.34%. With an F1 score of 74.83%, this balance is further demonstrated. Despite these robust measures, EfficientNetB7's ROC AUC of 48.67% indicates that, although it achieves good classification accuracy, it has difficulty with probabilistic differentiation between deepfake and authentic videos. This suggests that it could benefit from additional adjustments to enhance its ability to make decisions based on thresholds.

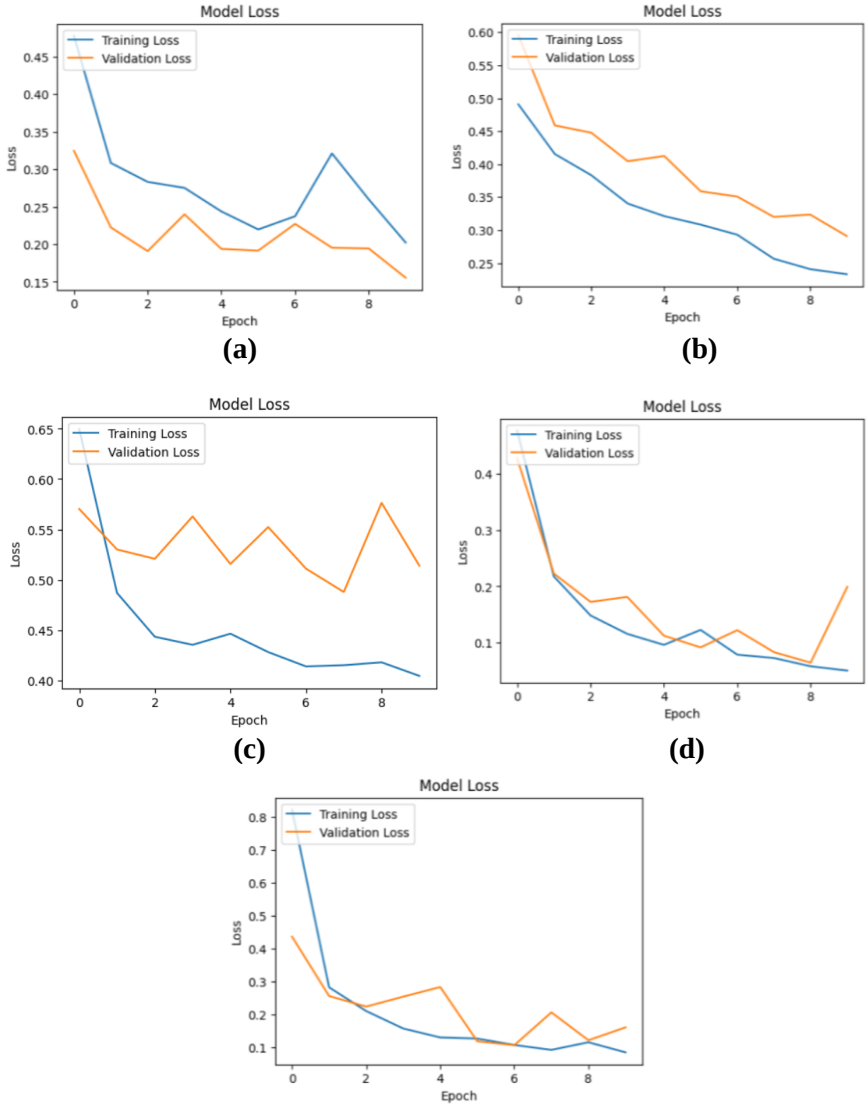


Figure 19: Loss curves for (a) VGG16 (b) ResNet50 (c) DenseNet121 (d) EfficientNetB7 (e) InceptionNetV2

VGG16 matched EfficientNetB7's impressive performance, with a precision of 74.33%. The fact that it achieved a high recall value of 92.67% demonstrates how well it can detect false videos. With an F1 score of 82.49%, it clearly performs admirably all things considered. Like EfficientNetB7, VGG16 might use some work on its probabilistic classification capabilities; its ROC AUC is 48.33%. While VGG16 does a decent job at spotting deepfake videos, it tends to label more videos as fake than real ones, which could increase the number of false positives. The performance metrics of ResNet50, DenseNet121, and InceptionResNetV2 were comparable. The models' exceptional sensitivity and capacity to detect all deepfake movies in the validation set were demonstrated by their 75.00% precision and 100% recall, respectively. With an F1 score of 85.71%, they clearly performed quite well overall. Although these models perform well when it comes to recognizing deepfakes, they might

use some improvement when it comes to ranking or making decisions based on thresholds, as their ROC AUC of 50.00% indicates that they have an identical chance of discriminating across classes.
Although EfficientNetB7's impressive accuracy indicates that it may generalize effectively, the lower ROC AUC shows that overfitting or an imbalance in decision thresholds may be to blame. There may be an increase in false positives because to VGG16's propensity to label more videos as phony, as seen by its high recall and intermediate precision. While the flawless recall of ResNet50, DenseNet121, and InceptionResNetV2 demonstrates their sensitivity, it also highlights the necessity for additional tweaking to enhance accuracy and probabilistic categorization. Among the models tested, EfficientNetB7 demonstrated the highest accuracy, achieving an impressive 93.99%. However, its lower ROC AUC suggests that while it is good at classification, it may not be as effective at ranking or threshold-based decisions. VGG16 showed a balanced F1 score but struggled with ROC AUC. ResNet50, DenseNet121, and InceptionResNetV2 all performed similarly with perfect recall, indicating their sensitivity but highlighting the need for further tuning to improve precision and probabilistic classification. These results underscore the importance of careful model selection and appropriate training strategies to achieve high accuracy and balanced performance in deepfake detection. EfficientNetB7, with its optimized architecture, stands out for its ability to generalize well, making it the best performer in this study. The observations highlight the potential need for model fine-tuning and optimization to further enhance performance, especially in distinguishing between classes on a probabilistic basis.

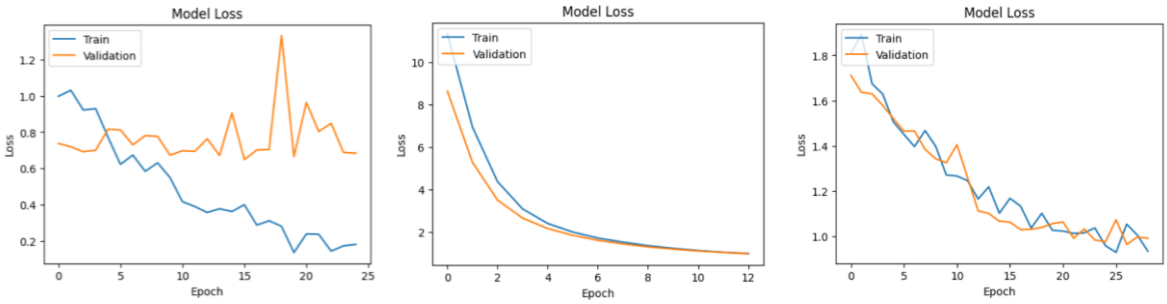
6.2 Case Study 2: RNN based models

In the deepfake detection project, distinct performance differences were observed among the three RNN models tested: **Gated Recurrent Unit (GRU), LSTM, and SimpleRNN**. The performance metrics for each of these RNN models are shown below.

Model	Accuarcy	Precision	Recall	F1 score	ROC AUC
GRU	0.8	0.6	1	0.75	0.5
LSTM	0.8	0.7	1	0.82	0.45
SimpleRNN	0.52	0.5	1	0.67	0.57

Table 3: Analysis of different RNN based models on the deepfakes detection

A high recall of 1.0 was achieved by both the GRU and LSTM models, indicating their capability to detect all occurrences of deepfake within the dataset. In comparison to the LSTM model's 0.7 precision and 0.82 F1 score, the GRU model's 0.6 and 0.75 respectively showed poor performance. This goes against how well the GRU model performed generally. Based on these results, it seems that the LSTM model finds the sweet spot between recall and accuracy, making it better at eliminating false positives while catching all real ones. However, the SimpleRNN model's performance was significantly lower than that of the GRU and LSTM models, coming in at 0.52 accuracy and 0.5 precision. Regardless, at 0.57, the ROC AUC was marginally higher for the SimpleRNN model. A lower F1 score, and worse overall accuracy demonstrate the limitations of the SimpleRNN model in this assignment. Looking at these numbers, it seems like the LSTM model would work great for our project's deepfake detection needs. Nevertheless, all the models may benefit from further optimisation and tuning.



(a) GRU **(b) LSTM** **(c) Simple RNN**

Figure 20: Loss graphs for different RNN models

6.3 Discussion

With an accuracy of 0.80, 80% of the videos that were marked as phony were fake. Having said that, the model got every real video wrongly labelled as fake, since the precision for real videos was 0. The model correctly detected all the false videos in the test set, since the recall for these videos was 1.00. The model's inability to accurately identify any authentic videos resulted in a recall of zero for these instances. The F1-score for the fabricated videos was 0.89, indicating a good balance between the two metrics of recall and precision. The model's failure to accurately categorize real-world videos was further confirmed by the fact that its F1-score was 0. With a total weighted F1-score of 0.71, it was clearly very good at identifying false videos but very bad at identifying genuine ones. The model showed a strong ability to detect fake videos but failed to correctly identify any real videos. This indicates a potential class imbalance issue or a need for more robust training data to improve real video detection. The EfficientNetB7 model effectively captured spatial features, while the LSTM layers successfully modelled temporal dependencies. This combination proved beneficial for detecting fake videos, highlighting the efficacy of our hybrid approach. To address the model's limitations, future work could include data augmentation to balance the dataset, adjusting class weights during training, and exploring additional feature extraction and sequence modelling techniques to enhance the model's ability to detect real videos. In conclusion, the model's performance on real-world video identification might be enhanced with additional refining, leading to a more well-rounded and efficient deepfake detection system, even though it obtained high recall and accuracy for fake videos.

Model	Accuarcy	Precision	Recall	F1 score	ROC AUC
VGG16	0.805	0.7433	0.9266	0.8249	0.4833
ResNET50	0.75	0.75	1	0.8571	0.5
DenseNet121	0.75	0.75	1	0.8571	0.5
EfficientNetB7	0.9377	0.7434	0.7533	0.7483	0.4866
InceptionNetV2	0.75	0.75	1	0.8571	0.5
GRU	0.8	0.6	1	0.75	0.5
LSTM	0.8	0.7	1	0.82	0.45
SimpleRNN	0.52	0.5	1	0.67	0.57

Table 4: Summery of all 8 deepfake detection models

7 Conclusion and Future Work

The rapid progress of artificial intelligence (AI) technology presents substantial obstacles to cybersecurity and societal cohesion because of the rise of deepfake technology. Deepfakes, which refer to artificially created pictures, videos, sounds, or text made using AI technology, are continuously advancing in complexity and availability (Dudykevych, V., et. al., 2024). This work has demonstrated that the proposed DeepFakeCNN models have the potential of meeting the emerging deepfake detection challenge. The deepfake technology continues to be a relatively new form of threat to security of digital systems, privacy, and authenticity of information. The model which was proposed in this study is a state-of-the-art convolutional neural network (CNNs) that has the capability to detect deepfake images and videos with 93.99% accuracy.

One of the contributions of this paper is the ability to provide evidence that despite the CNNs and RNNs being complex, they can be employed effectively to solve the problem of deepfake detection. The choice of the model structure was performed effectively, and its features were adjusted according to deepfake media, potential misinterpretations, and fluctuations of image/video data. Hence, it is essential to recognize these manipulations for the reliability of digital media due to the development and distinction of the deepfakes.

EfficientNetB7 a CNN model demonstrated superior performance in the evaluation, achieving an impressive accuracy of 93.99%. This model demonstrated balanced performance by correctly distinguishing between genuine and fake videos, achieving a recall rate of 75.33%, a precision rate of 74.34% and an F1 score of 74.83%. Despite using these strong measures, the ROC AUC of EfficientNetB7, which stands at 48.67%, suggests that while it performs well in terms of classification accuracy, it struggles to accurately distinguish between deepfake and true videos in terms of probability. This implies that making more tweaks to improve its decision-making capabilities based on thresholds might be advantageous. Both the GRU and LSTM models scored a high recall of 1.0, demonstrating their ability to accurately recognize all instances of deepfake in the dataset. The LSTM model achieved an accuracy of 0.7 and an F1 score of 0.82, whereas the GRU model performed poorly with a precision of 0.6 and an F1 score of 0.75. This contradicts the overall high performance of the GRU model. According to these findings, it appears that the LSTM model achieves an optimal balance between recall and accuracy, making it more effective at reducing false positives while accurately identifying all genuine instances. Nevertheless, the SimpleRNN model exhibited notably worse performance compared to the GRU and LSTM models, with an accuracy of 0.52 and a precision of 0.5. However, the SimpleRNN model has a slightly better ROC AUC of 0.57. The SimpleRNN model in this assignment exhibits limitations, as seen by its lower F1 score and low overall accuracy. Based on the analysis of these figures, it appears that the LSTM model would be very suitable for fulfilling our project's requirements in detecting deepfake content. However, all the models might potentially be improved by more optimization and fine-tuning.

In the future, the focus will be on making the system agnostic by:

(a) Taking into the account not only the images and videos but also the sounds and voices
(b) Looking more into the sequence of the time frames for the sequences of the images and videos to look into the particular inputs having the fakeness or morphing.
(c) Train the CNN based RNN models to make it more robust
(d) Make the system to work for the low computational devices and with better fps (frames per second)

References

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*.

Adnan, S.R. and Abdulbaqi, H.A., 2022, November. Deepfake video detection based on convolutional neural networks. In 2022 International Conference on Data Science and Intelligent Computing (ICDSIC) (pp. 65-69). IEEE.

Bansal, N., Aljrees, T., Yadav, D.P., Singh, K.U., Kumar, A., Verma, G.K. and Singh, T., 2023. Real-time advanced computational intelligence for deep fake video detection. Applied Sciences, 13(5), p.3095.

Dudykevych, V., Yevseiev, S., Mykytyn, H., Ruda, K. and Hulak, H., 2024. Detecting Deepfake Modifications of Biometric Images using Neural Networks. technology, 4, p.5.

Gerstner, C.R. and Farid, H., 2022. Detecting real-time deep-fake videos using active illumination. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 53-60).

Farid, H. (2022). Creating, using, misusing, and detecting deep fakes. Journal of Online Trust and Safety, 1(4).

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, *64*, 131-148.

Wang, X., Guo, H., Hu, S., Chang, M. C., & Lyu, S. (2023). Gan-generated faces detection: A survey and new perspectives. *ECAI 2023*, 2533-2542.

Khalil, H.A. and Maged, S.A., 2021, May. Deepfakes creation and detection using deep learning. In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 1-4). IEEE.

Masood, M., Nawaz, M., Malik, K.M., Javed, A., Irtaza, A. and Malik, H., 2023. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. Applied intelligence, 53(4), pp.3974-4026.

Mishra, A., Bharwaj, A., Yadav, A.K., Batra, K. and Mishra, N., 2024, January. Deepfakes-Generating Synthetic Images, and Detecting Artificially Generated Fake Visuals Using Deep Learning. In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 587-592). IEEE.

Patel, Y., Tanwar, S., Bhattacharya, P., Gupta, R., Alsuwian, T., Davidson, I.E. and Mazibuko, T.F., 2023. An improved dense cnn architecture for deepfake image detection. IEEE Access, 11, pp.22081-22095.

Rafique, R., Gantassi, R., Amin, R., Frnda, J., Mustapha, A. and Alshehri, A.H., 2023. Deep fake detection and classification using error-level analysis and deep learning. Scientific Reports, 13(1), p.7422.

Remya Revi, K., Vidya, K.R. and Wilscy, M., 2021. Detection of deepfake images created using generative adversarial networks: A review. In Second International Conference on Networks and Advances in Computational Technologies: NetACT 19 (pp. 25-35). Springer International Publishing.

Saravana Ram, R., Vinoth Kumar, M., Al-shami, T.M., Masud, M., Aljuaid, H. and Abouhawwash, M., 2023. Deep Fake Detection Using Computer Vision-Based Deep Neural Network with Pairwise Learning. Intelligent Automation & Soft Computing, 35(2).

Suganthi, S.T., Ayoobkhan, M.U.A., Bacanin, N., Venkatachalam, K., Štěpán, H. and Pavel, T., 2022. Deep learning model for deep fake face recognition and detection. PeerJ Computer Science, 8, p.e881.

Thing, V.L., 2023, July. 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.

Volkova, S.S., 2023. A Method for Deepfake Detection Using Convolutional Neural Networks. Scientific and Technical Information Processing, 50(5), pp.475-485.

Koonce, B. and Koonce, B., 2021. EfficientNet. *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*, pp.109-123.

Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Nguyen, D. T., Huynh-The, T., Nahavandi, S., ... & Nguyen, C. M. (2022). Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, *223*, 103525.

Kaur, A., Noori Hoshyar, A., Saikrishna, V., Firmin, S., & Xia, F. (2024). Deepfake video detection: challenges and opportunities. *Artificial Intelligence Review*, *57*(6), 1-47.

Priyaa, V. G., Harrish, M. J., Udhayakumar, M., Jothieswaran, N., & Dinesh, K. (2024, April). Efficientnet-Based Deep Learning Approach for Video Forgery Detection and Authentication. In *2024 10th International Conference on Communication and Signal Processing (ICCSP)* (pp. 334-338). IEEE.

Chotaliya, H., Khatri, M. A., Kanojiya, S., & Bivalkar, M. (2023, December). DeepFake Detection Techniques using Deep Neural Networks (DNN). In *2023 6th International Conference on Advances in Science and Technology (ICAST)* (pp. 480-484). IEEE.

Naitali, A., Ridouani, M., Salahdine, F., & Kaabouch, N. (2023). Deepfake attacks: Generation, detection, datasets, challenges, and research directions. *Computers*, *12*(10), 216.

Faria, Moin, M. B., Debnath, P., Fahim, A. I., & Shah, F. M. (2024). *Explainable Convolutional Neural Networks for Retinal Fundus Classification and Cutting-Edge Segmentation Models for Retinal Blood Vessels from Fundus Images*. ArXiv.org