2022

Evaluating the Performance of Vision Transformer Architecture for Deepfake Image Classification

Devesan Govindasamy
Technological University Dublin

Recommended Citation

Govindasamy, D. (2022). Evaluating the Performance of Vision Transformer Architecture for Deepfake Image Classification . Technological University Dublin. DOI: 10.21427/8QA1-WS63

Evaluating the Performance of Vision Transformer Architecture for Deepfake Image Classification

Devesan Govindasamy

D20124946

A dissertation submitted in partial fulfilment of the requirements of Technological University Dublin for the degree of

M.Sc. in Computer Science (Data Science)

Jan 2022

DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computer Science (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation confirms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: **Devesan Govindasamy**

Date: **05/01/2022**

ABSTRACT

Deepfake classification has seen some impressive results lately, with the experimentation of various deep learning methodologies, researchers were able to design some state-of-the art techniques. This study attempts to use an existing technology “Transformers” in the field of Natural Language Processing (NLP) which has been a de-facto standard in text processing for the purposes of Computer Vision. Transformers use a mechanism called “self-attention”, which is different from CNN and LSTM. This study uses a novel technique that considers images as 16x16 words (Dosovitskiy et al., 2021) to train a deep neural network with “self-attention” blocks to detect deepfakes. It creates position embeddings of the image patches which can be passed to the Transformer block to classify the modified images from the CELEB-DF-v2 dataset. Furthermore, the difference between the mean accuracy of this model and an existing state-of-the-art detection technique that uses the Residual CNN network is compared for statistical significance. Both these models are compared on their performances mainly Accuracy and loss. This study shows the state-of-the-art results obtained using this novel technique.

The Vision Transformer based model achieved state-of-the-art performance with 97.07% accuracy when compared to the ResNet-18 model which achieved 91.78% accuracy.

Key words: Deep learning, Transformers, Vision-Transformers, Self-attention, ResNet, Transfer Learning.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr. Bianca Schoen Phelan for her assistance, support, motivation, and patience throughout this MSc dissertation.

I would also like to express my sincere gratitude to Dr. Emma Murphy and Dr. Luca Longo for advising on different aspects of the dissertation.

I would like to thank TPU Research Cloud (TRC) for providing access to free access to TPU's for training and running the Machine Learning Models.

Also, I would like to thank all the professors in Technological University Dublin, who guided and supported me throughout my study in the University.

TABLE OF CONTENTS

TABLE OF FIGURES:

TABLE OF TABLES & EQUATIONS

1 INTRODUCTION

1.1 Background

The term ‘Deepfake’ is used to describe synthetic media in which the person in an image/video is morphed to look like someone else using Deep Neural Networks. The impact of these deepfakes in this digital world is immense and has raised various concerns in the field of fake news and fraudulent activities, which becomes a perilous problem for the credibility of any information on the internet. To tackle this problem researchers from all over the world are trying to detect the deepfakes on the internet. There are various techniques to detect deepfake.

The proliferation of AI has also paved the way for the creation of more sophisticated deepfakes and requires advanced techniques to identify them. This led to a rapid increase in the circulation of deepfake images and videos on the internet which pose a severe risk to privacy.

Deepfakes started to get traction after 2017 when a mobile application named FaceApp was launched with the possibility of manipulating and realistically simulating photos and videos. This led to further innovation in the field of creating more deepfakes. Through the advancements of Deep Neural Networks (DNN) and Generative Adversarial Networks (GANs) framework many models were developed to generate superficially authentic simulations to at least trick the human observers.

Through the commercialization of Deepfake creation, even unskilled people can create deepfakes using some online tools like [1]. The AI firm Deeptrace has found over 15,000 deepfake videos online in Septemeber 2019, nearly doubling over nine months. Among all the videos, 96% of them were pornographic and 99% of those mapped faces from female celebrities on to porn stars[1]. Most of these deepfakes seem to be targeting women and celebrities. They are also used to sway people's opinion on real-world issues like the deepfake videos of Trump during the 2020 US presidential elections.

[1] https://deepfakesweb.com

The constant advancements in Deepfake generation were countered with similar advancements in detection techniques as well. Big tech companies and Governments have taken steps to counter the deepfakes. The first Deepfake Detection Challenge was kicked off last year backed by Microsoft, Facebook, and Amazon. It included research teams around the globe competing for supremacy in the deepfake detection game and innovating new techniques.

This study intends to contribute to the field of Media forensics by means of a new technique to identify deepfakes. It implements a new deepfake classifying technique using a Vision transformer model with a custom facial feature extractor in Python which should classify deepfakes by analysing the intra-frame irregularities like smudged/pixelated frames in the video with self-attention mechanism and the accuracy of the classification is compared to a state-of-the-art Residual CNN model for benchmarking purpose. This study also discusses ablation studies on the model with various parameters.

1.2 Research Project

Convolutional models are powerful tools that have been traditionally used for computer vision tasks. Some problems with them include the use of a pooling layer and its translational invariance. The relative position of distinct features is not encoded by CNN. To encode the combination of these features, large filters are necessary. For example, huge filters are required to encode the information "eyes above nose and mouth". Large receptive fields are required to track long-range dependencies within an image. Increasing the size of the convolution receptive field can increase the representational capacity of the network and improve the performance but doing so also loses the computational and statistical efficiency obtained by using the local convolutional structure.

To overcome this issue, a novel technique of Vision Transformer (ViT) is being slowly used to compete with the existing Convolutional models in Computer Vision. ViT was introduced by (Dosovitskiy et al., 2021) and has performed competitively to some of the state-of-theart Convolutional models in Computer Vision Tasks. This technique works with positional encodings of different patches of an image and passing it to a Transformer block with “selfattention” which can focus on the modified localities in the manipulated images. This could greatly reduce the computational cost of the model when compared to Convolutional models with large receptive fields.

The use of self-attention mechanisms has been around for a while in the field of computer vision, but they are added as an additional block to existing Convolutional blocks like Xception, ResNet, EfficientNet, DenseNet, Inception. But with the help of Vision transformers (ViT), it is possible to implement a pure transformer model without any convolutional block.

The research question is framed as follows:

“To what extent can the mean difference between the accuracy of a deepfake classification network be improved by using a Vision Transformer model when compared to using a traditional ResNet CNN model when a custom MTCNN face extraction input pipeline is used?”

1.3 Research Objectives

The research will be based on Computer vision and Transformers. It implements two different methods of deepfake classification namely a Vision Transformer model with selfattention and this method will be compared with a Residual CNN model with Resnet-18. Both methods will be implemented with the help of TensorFlow 2.0 and Python 3 and Google’s TPU cloud.

Several Deepfake datasets are available for research purposes, among which CELEB-DFv2 is used as it is one of the most diverse and high-resolution datasets. This dataset was able to beat some of the State-of-the-art models.

The research has various building blocks, each needs to be implemented separately and integrated together to achieve the results. This modular implementation gives the freedom to change different parts of the experiment without hassle.

- The research starts with the creation of a custom data-pipeline to accommodate the models that will be built on top of it.
- Model Schema is created so it could be added later to the training block of the code.
- Creating optimizers, loss function, Data Augmentation block.
- Integrating the model with the optimizer and loss function.
- Training the model with the dataset and compiling the results.
- Finally, both the models are evaluated using their accuracy, loss for their performances.

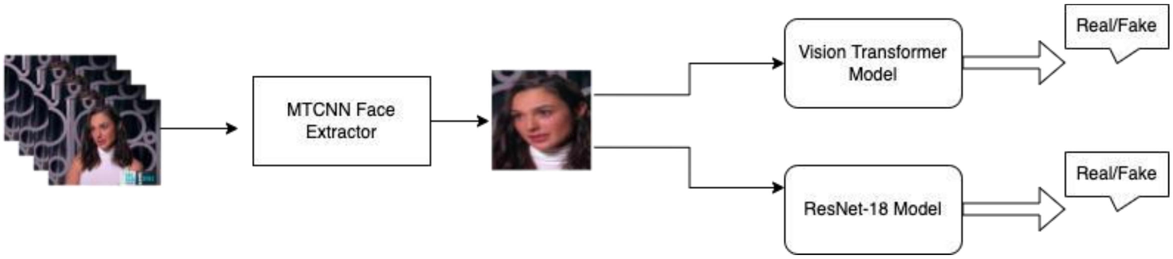

Figure 1.1 Experiment Design

1.4 Research Methodologies:

The research focuses on a purely Quantitative approach by designing an experiment and relying on its results. It follows an empirical research method as it involves gaining knowledge by observing the data and involves in defining the hypothesis test and prediction. Deductive reasoning will be applied for this research as the research starts with hypothesis testing, supporting data evidence is provided to test the hypothesis, and the conclusion is drawn based on the analysis.

A Vision transformer is built and is trained based on the faces extracted from the CELEBDF-v2 dataset using MTCNN model and is used to train both the models with real and fake images with some data augmentation. The models are then evaluated using the test images from the dataset and the results are compared, so this makes the research Deductive Research.

1.5 Scope and Limitations:

The aim of the research is to try and combat the increasing forgeries in the Media and help authenticate the images/videos on the internet. This technology's scope spreads wider than Media forensics and helps the moderation of deepfakes on the internet.

Limitations:

- The research focuses on identifying irregularities on images and needs to be trained to learn them and may or may not learn to identify them.
- Processing an image is a resource-intensive task and needs bigger clusters of GPUs, TPUs and longer time for training.
- Transformers perform better when they are pre-trained and fine-tuned on a specific task.
- This model cannot tackle all sorts of deepfakes and may fail in certain cases.

1.6 Document Outline:

The rest of the paper is organized as follows.

Chapter 2-Literature Review

This chapter is dedicated to the literature survey of the previous research papers and their proposals, this could help identify the growth of deepfake detection techniques over the years and formulate new designs and techniques to overcome their flaws and create a new model to perform better.

Chapter 3 – Design and Methodology

This chapter discusses the proposed methods for deepfake detection and provides the necessary background to the model's design and required resources. This chapter contains a detailed explanation of the dataset, model, and its evaluation.

Chapter 4 – Results, Evaluation and Discussion

This chapter discusses the detailed analysis of the results and output from each model. Suitable evaluation metrics have been obtained and the mean difference in accuracy is compared using a statistical test to reject or accept the null hypothesis.

Chapter 5 – Conclusion

This chapter summarizes the overall analysis and results obtained from the experiment conducted through this study and has also suggested the future scope for the research as an extension to this paper.

2 LITERATURE REVIEW AND RELATED WORKS

In this section, different techniques of deepfake creation and detection are explained starting from its early stages and to its recent ones along with some background. Both creation and detection have grown leaps and bounds over the years and became more sophisticated. This would help in designing the experiment for the research.

2.1 Deepfake Background:

Face manipulation has been around even before the emergence of deep learning. (Dale et al., 2011) introduced a face-swapping method based on a 3D multi-linear model for face tracking and warping. Later with the help of neural networks, (Zhmoginov et al., 2016) created a method to invert the low-dimensional face embeddings while producing highly realistic modified images. This technology was later implemented into a mobile application called FaceApp, the popularity of the app led to tremendous advancements in the creation of Modified videos and certainly the use of these fakes created using deep learning, hence "Deepfakes" became more common and were used in various criminal activities. These led to severe consequences in the Politics, Finance, the social life of many people. The threat included vengeful pornographic content made using modified faces of an individual mostly focused on women, hate comments from a person who never said it but were created using deepfake technology.

The US congress had two public hearings about deepfakes, and the topic had enough media coverage to create awareness about the digital provenance of any content. The congress also passed the first federal legislation which was later signed into law about the use of foreigndeepfake usage to influence the US elections. This became the first law to condemn the use of deepfake.

This year the US senate has passed an Deepfake Task force Act which plans to reduce the proliferation and impact of digital content forgeries, including by exploring how the adoption of a digital content provenance standard could assist with reducing the proliferation of digital content forgeries; develop mechanisms for content creators to cryptographically certify the authenticity of original media and non-deceptive manipulations and enable the public to validate the authenticity of original media and nondeceptive manipulations to establish digital content provenance.

This act allows individuals as well as tech companies to create standards and detection systems to identify digital forgery.

Tech giants have been offering grants and holding hackathons to moderate the flow of unauthenticated videos and images. With the availability of versatile and high quality Deepfake datasets, many detection models have been created. But with the evolution of the deeper Neural Networks the sophistication of the deepfakes have increased and so the detection models must also evolve accordingly.

The below gives a brief idea on the state-of-the-art deepfake creation and detection models.

2.2 Deepfake Creation:

Generative adversarial networks (GAN) and variational autoencoders (VAE) are a powerful tool for generating image content. However, early implementations produce images of low resolution that oftentimes exhibit blur, which allows to easily identify them as generated. (Karras et al.,2018) overcame this limitation by demonstrating the generation of high-resolution images of up to 1024 × 1024 pixels in the so-called ProGAN.

High-resolution deepfakes are mostly generated using Generative Adversarial Network (GAN) introduced by Goodfellow et. al. This technique has two generative models running simultaneously namely a Generator and a Discriminator. The generator creates the fake samples and will pass it to the discriminator which identifies whether the sample generated is fake or real. The discriminator is trained to identify the real domain images. This adversarial nature helps improve the generator model to a state where it can overcome the discriminator. The model keeps updating until the generator can generate a sample which can “fool” the discriminator. Discriminators are trained on the training samples so that they can identify the real samples. When the generator creates a sample which is close to the training sample only then it can overcome the discriminator.

Some of the notable deepfake generation techniques are Face2Face by (Thies et al., 2016) which generates a real-time facial reenactment of a monocular target video, FaceSwap by (Nirkin et al., 2019) which used an encoder-decoder to generate deepfakes. The method proposed by (Kim et al., 2018) extends these approaches by allowing to manipulate the 3D head position, head rotation, face expression, eye gaze, and eye blinking using a generative neural network. The method by (Bansal et al., 2018) is able to transfer video content from one domain to another, which can be applied to face-to-face scenarios. Some methods focus on changing certain facial attributes such as hair-colour or age in single images. The method by (Pumarola et al., 2018) can animate facial expressions in a convincing manner, given a single input image. These techniques were used to create some deepfake datasets like UADFV, DF-TIMIT, FF-DF, DFDC. These datasets helped create many detection algorithms.

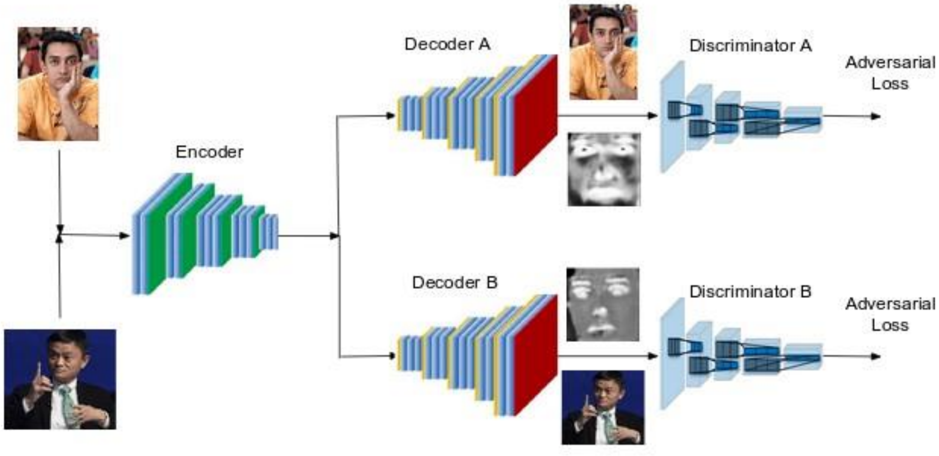

Figure 2.1 Generative Adversarial Network Architecture [2]

[2] https://imrahulr.github.io/deepfakes/gans/autoencoder/coe-cnds/2018/01/09/deepfakes.html

Some of the notable deepfake generation techniques are mentioned in the table below with their key features in generating the deepfakes.

Tools	Links	Key features
Face Swap-GAN	https://github.com/shaoanlu/faceswap-GAN	The auto-encoder architecture comprises of Adversarial Loss and perceptual loss (VGGface).
DFaker	https://github.com/dfaker/df	DSSIM loss function is used to reconstruct the face. Implemented based on Keras library.
DeepFace Lab	https://github.com/iperov/DeepFaceLab	Introduces new models with FaceSwap as base like H64, H128, LIAEF128, SAE. It supports multiple face extraction modes, e.g., S3FD, MTCNN, dlib
FaceShifter	https://lingzhili.com/FaceShifterPage	By utilizing and integrating the target attributes, high-fidelity face swapping can be achieved. This can be applied to any new pairs of face without requiring subject specific training
FSGAN	https://github.com/YuvalNirkin/fsgan	A face swapping and re-enactment model that may be used on pairs of faces without the need for training. Adapt to changes in both pose and expression.

Table 2.1 NOTABLE DEEPFAKE GENERATION TECHNIQUES

The dataset used in this dissertation, CELEB-DF-v2, was generated using an improved DeepFake synthesis algorithm. They were able to overcome some of the common issues in a deepfakes like Low resolution of synthesized faces which was improved to 256x256, Color mismatch which was reduced with the use of color transfer algorithm by (Erik Reinhard et. al., 2001) Temporal flickering which can be caught using naked eye was reduced by incorporating temporal correlations among the detected face landmarks. Specifically, the temporal sequence of the face landmarks is filtered using a Kalman smoothing algorithm to reduce imprecise variations of landmarks in each frame. There are in total 5, 639 DeepFake videos, corresponding to more than 2 million frames, with 712 real videos in the Celeb-DF dataset. The real source videos are based on publicly available YouTube video clips of 59 celebrities of diverse genders, ages, and ethnic groups (Li et al., 2020).

2.3 Deepfake Detection

Deepfake detection problems can be addressed as a binary classification problem with two classes real/fake (Rössler et al., 2019). They try to exploit the anomalies present in the images. These visual artifacts are mostly identified using Convolutional architectures and used to authenticate it. These detection algorithms use different Convolutional Neural Networks (CNN) for example InceptionV3, DenseNet, VGG16 (Dang, H. et al), Xception (Rössler et al., 2019).

The spatial anomalies are detected using convolutional architectures, but in cases of deepfake videos, some advanced detection systems use sequence learning techniques like RNN or LSTM to identify the temporal anomalies among the modified frames.

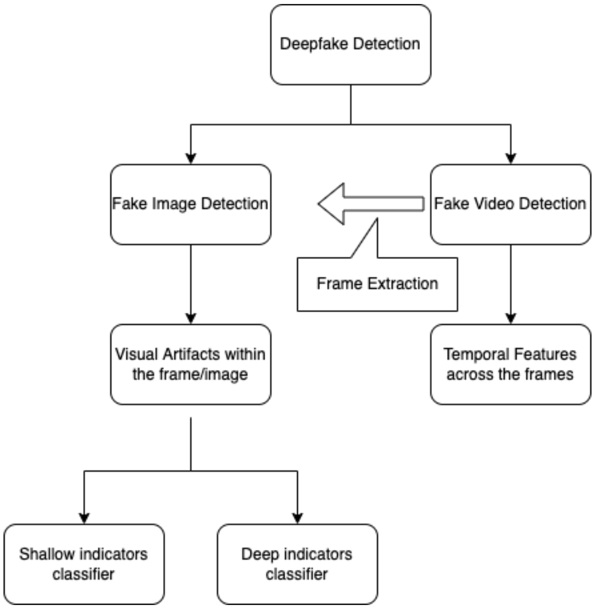
Deepfake Detection
Fake Image Detection
Fake Video Detection
Frame Extraction
Visual Artifacts within the frame/image
Temporal Features across the frames
Shallow indicators classifier
Deep indicators classifier

Figure 2.2 Deepfake Detection methods overview

The below explains the different detection systems with their drawbacks which helped in improving this research.

2.3.1 CNN-Based Detection Systems

Early Deepfake detection models used CNN models to identify shallow indicators in the images like the too-smooth skin, the color mismatch can evade detection (Huang et al., 2020, p. 1224) which were useful in identifying the deepfakes created in the early stages of generation like the FaceForensics++ which were of lower resolutions and had more visible flaws. Recent works (Verdoliva et al., 2020) have proved that deep architectures outperform the shallow networks by a large margin.

These methods use face warping artifacts to help classify the deepfakes. Some artifacts like the eyes and teeth play a paramount role in deepfake detection but have limitations that the eyes need to be open, and the teeth should be visible, this severely limits the applicability of the models (Afchar et al., 2018, p. 3) (Matern et al., 2019) which can be seen in Figure 2.4. These algorithms dropped drastically in their performance when used on some of the most recent high quality deepfake datasets. For example, Li et al. put forward a detection method based on the face warping artifacts. This method achieved the AUC of 80.1% when testing on UADFV (Li et al., 2020) but dropped significantly to 56.9% when confronting with CelebDF.

Inconsistent head poses of a person on the deepfakes can also be used to identify deepfakes. As a result of face swapping, the landmark location of the fake faces can often deviate from those of the original faces. This discrepancy is exploited to classify deepfakes.

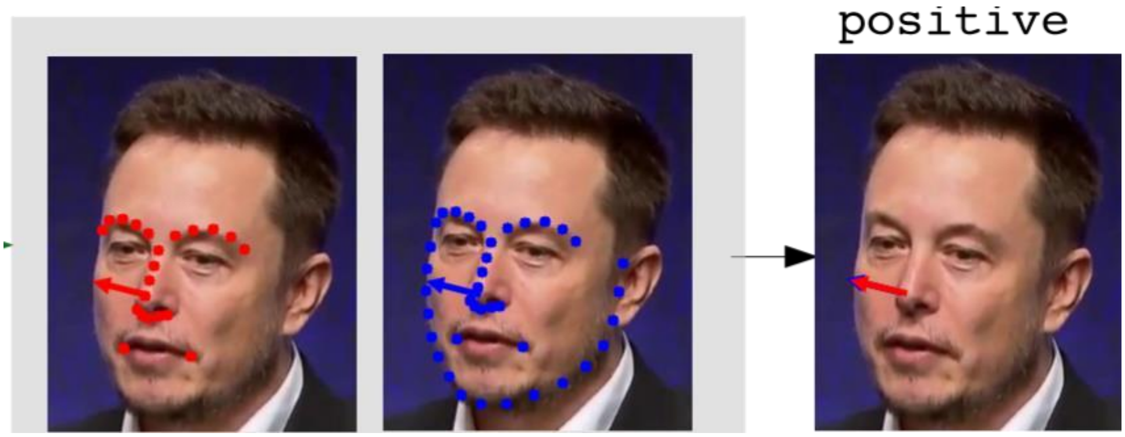

Figure 2.3 3D head poses deepfake classification. (Yang et al., 2019)

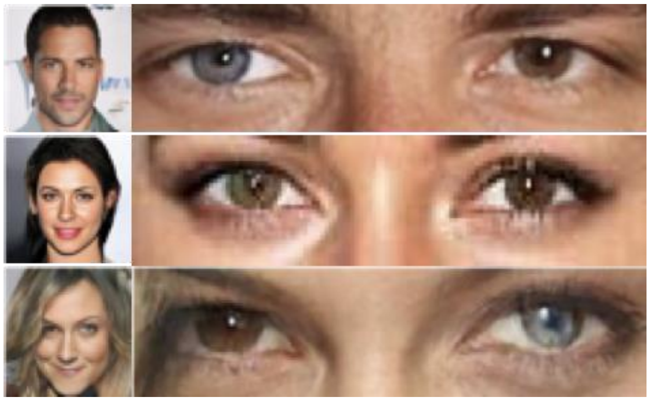

Figure 2.4 Samples of different methods display the difference between color of the left and right eye. (Matern et al., 2019)

One of the most robust deep convolutional architectures is the Xception network which allows a variety of models to be built on top of it as a backbone. Almost all existing algorithms have a similar pre-processing part which includes the extraction of the features from the images and using these features to train their models.

These models assume the deepfakes produced using the GANs algorithm are fewer resolution images (Li & Lyu, 2018) and can provide better results than High-Quality deepfakes (Korshunov & Marcel, 2019, p. 3). Using filters to reduce the feature size is not explored as some gaussian filters can inadvertently filter out the important visual artifacts in the frame and need to be finely tuned for this task or they may miss out on the fake frames and thereby wrongly classifying them (Masi et al., 2020, p. 671). For example, by removing, inserting, or cloning entire groups of frames one can completely change the meaning of a video. A simple frame-rate reduction was recently used to let Nancy Pelosi, Speaker of the U.S. House of Representatives, appear as drunk or confused (Verdoliva, 2020).

2.3.2 Sequence Learning Based Detection System

Research by (Guera & Delp, 2018, p. 2) used the Convolutional LSTM technique to identify facial anomalies like face warping artifacts, head position between inter-frames of a video. This is used because of the inconsistencies created while altering the original videos and we try to exploit that flaw. Sequence learning helped create a new technique to classify deepfake videos as opposed to earlier techniques using averaging of the frame level predictions to assign a prediction to the whole video. The use of LSTM architecture is resource intensive.

Other technique to analyse the spatiotemporal anomalies is the use of 3D convolutional architectures l which employs 3D filters that pick up the knowledge of spatiotemporal features from the videos, in contrast to 2D convolution, where the temporal domain is collapsed.

Deepfake detection algorithms have been fine tuned for detecting images over the years but the techniques to detect deepfake videos are still in the process of refinement. One of the most important problems that occur when detecting deep fake videos, is because for a video the prediction score of each frame is averaged to find the overall prediction score which in the controlled situation can affect the overall prediction score of the deepfake video (Montserrat et al., 2020).

Another method which addresses some of the common problems of classifying a video is by using Automatic Face-weighting (Montserrat et al.,2020). This technique can weigh the importance of the detected face in each frame as some frames may miss faces in real-time videos and can be weighed low and the overall classification score for the video can be calculated.

Sequence learning can also be used in finding the eye blinking patterns in the videos which can be used to classify deepfakes. This technique is one of the state-of-the-art techniques. One of the state-of-the-art detection techniques is using the biological signals done by (Ciftci & Demir, 2020) that can be extracted from the images like the blood vessels on the faces, heart rate from the videos which can be used for analyzing temporal consistencies and thereby classifying the videos.

The computational cost of processing each frame in the video is high, so some techniques use key-frame extraction to minimize the cost, but this trade-off is possible at the expense of missing out on the morphed frame (Mitra et al., 2021). The robustness of the models built thus far is very little. Each model focuses on certain flaws in the Generative Adversarial Networks (GANs) algorithms when subjected to different algorithms most models would fail (Kumar et al., 2020). The continuous battle of finding vulnerabilities on deepfake creation and detection techniques and exploiting them to improve the other makes this ever-evolving field. Carefully building a data-augmentation pipeline at the creation process can evade detections.

Comparing different models only based on accuracy seems unfair as some techniques use temporal anomalies and can be resource-intensive and have better accuracy than models using only visual artifacts (Jung et al., 2020, p. 83153).

Constant improvement in GAN Algorithm has overcome some of the state-of-the art deepfake dataset like FaceForensics++ (Fernandes & Jha, 2020, p. 232). If the background of the GAN algorithm used is not known then their weakness can't be exploited for detection (Deshmukh & Wankhade, 2020, p. 300). Increasing the robustness is a difficult task, since the training of the images is a resource and time-intensive task and adding more features to it would increase the model's training time (Lyu, 2020, p. 3).

2.4 Transformer Based Techniques for Image Classification:

The use of transformers has been de-facto in NLP tasks; it holds strong promises (Paul & Chen, 2021) toward a generic learning method that can be applied to various data modalities. Transformers were proposed by (Vaswani et al., 2017) for machine translation and have since become the state-of-the-art method in many NLP tasks by using contextualized embeddings obtained from self-attention. Large Transformer-based models are often pretrained on large corpora and then fine-tuned for the task at hand (Dosovitskiy et al., 2021).

Several attempts have been made to combine CNN-like architectures with self-attention

(Wang et al., 2018; Carion et al., 2020), with some replacing the convolutions entirely (Ramachandran et al., 2019; Wang et al., 2020a). The newer models, while theoretically efficient, have yet to be scaled efficiently on modern hardware accelerators due to the use of specialized attention patterns. As a result, in large-scale image recognition, traditional ResNet-like structures remain state-of-the-art (Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020).

In a naive application of self-attention to images, each pixel would have to pay attention to every other pixel. This does not scale to practical input sizes due to the quadratic cost in the amount of pixels. Many of these specific attention structures have shown to be effective.

2.4.1 Vision Transformer (ViT)

In computer vision research, there has recently been a rise in interest in Vision Transformers (ViTs). ViT's ideation is strongly grounded on introducing self-attention for images. As, self-Attention makes sense in spatial features. The ViT is a visual model based on a transformer's architecture, which was initially created for text-based operations. When used with a classification head, the ViT model depicts an input image as a series of image patches, like the sequence of word embeddings used when using transformers to text and predicts the label.

ViT exhibits an outstanding performance when pre-trained on large data and is fine-tuned for the required tasks. It can break the performance of a similar state-of-art CNN with 4x fewer computational resources (Dosovitskiy et al., 2021).

When it comes to NLP models, these transformers have a high success rate, and they're currently being used on photos for image recognition tasks. ViT separates the images into visual tokens, whereas CNN employs pixel arrays. The visual transformer separates a picture into fixed-size patches, embeds each one appropriately, and passes positional embedding to the transformer encoder as an input.

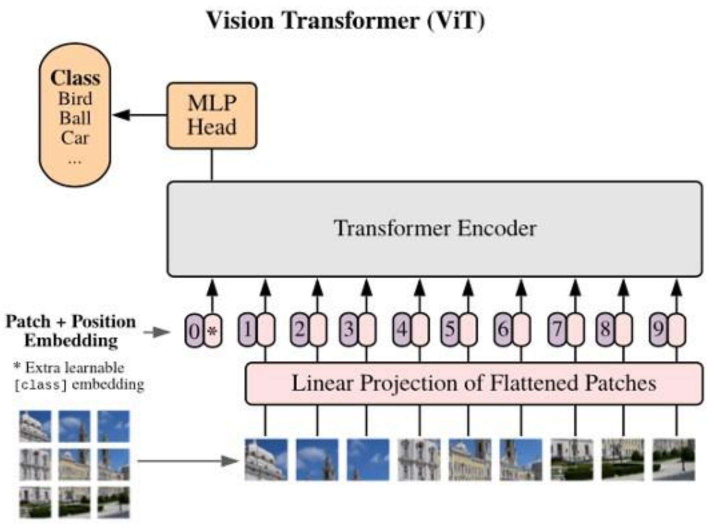
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 2.5 Vision Transformer Architecture- (Dosovitskiy et al., 2021)

Additionally, residual connections are provided after each block because they allow components to flow directly through the network without having to go through non-linear activations.

The MLP layer implements the classification head in the instance of image classification. At pre-training time, it uses one hidden layer and a single linear layer for fine-tuning.
In Computer Vision, the vision transformer model employs multi-head self-attention without the need for image-specific biases. The model divides the images into a series of positional embedding patches, which the transformer encoder processes. It does so to comprehend the image's local and global characteristics.

From the above literatures it has been hypothesized that the use of Vision Transformers for the task of Deepfake Classification could yield either competitive or better results when compared to a state-of-the-art Convolutional model namely ResNet model. To prove it an experiment has been designed and implemented in Chapter 3

3 DESIGN, METHODOLOGY, AND IMPLEMENTATION

This chapter will focus on the experiment used to determine whether the null hypothesis can be accepted or rejected. Two different models will be implemented for deepfake classification namely a state-of-the art ResNet CNN model, and a Vision Transformer based model. Ablation studies on both these models will be done. All the models will be tested on the CELEB-DF-v2 dataset. Data collection and preparation for conducting this experiment is described followed by the detailed explanation of the models for the purpose of the experiment.

The Accuracy score and loss of each model will be calculated and compared to check whether it can prove the null hypothesis or not.

3.1 Hypothesis:

NULL HYPOTHESIS: IF a custom MTCNN face extraction input pipeline is used to extract faces from the frames of a video and is used for classifying deepfake videos using a Vision Transformer, THEN the mean difference between the accuracy of the model and the ResNet CNN model will not be statistically significant (p-val > 0.05).

ALTERNATE HYPOTHESIS: IF a custom MTCNN face extraction input pipeline is used to extract faces from the frames of a video and is used for classifying deepfake videos using a Vision Transformer, THEN the mean difference between the accuracy of the model and the ResNet CNN model will be statistically significant (p-val < 0.05).

3.2 Data Collection and Understanding:

The deepfake dataset by (Yuezun Li et al.) contains videos of manipulated faces. The CelebDF-v2 dataset comprises 712 real videos and 5,639 DeepFake videos (corresponding to over two million video frames). The average length of all videos is approximately 13 seconds with the standard frame rate of 30 frame-per-second. The real videos are chosen from publicly available YouTube videos, corresponding to interviews of 59 celebrities with a diverse distribution in their genders, ages, and ethnic groups. The DeepFake videos are generated by swapping faces for each pair of the 59 subjects. The final videos are in MPEG 4.0 format.

Figure 3.1 Figure 3.2 Figure 3.3

Figure 3.1-3.3 Sample images from CELEB-DF-v2 dataset

3.3 Data Preparation:

Each video in the dataset is 13 seconds long with a standard 30 frame-per-second. The experiment was conducted with different framerates 5/15/25 per second to analyse the behaviour of the model.

The frame extraction task is run on a 96-core Intel Xeon CPU and takes around 30 minutes to complete.

The extracted frames are then passed to a face detection model which involves detecting the bounding box that contains the face in each image. A great bounding box should perfectly envelop the face without cutting out vital facial forms and characteristics or including more surrounding area than is required. This method would help reduce the background noise and focus on the modified faces which are the essential features when training the model. The MTCNN model is used for face detection. It is a 3 cascaded CNNs namely P-net, Rnet, O-net.

- **Pyramid Network (P-net):**

  The first step uses a picture pyramid made up of multiple scaled copies of the input image as its input. This gives the model many window sizes to pick from, allowing it to be scale invariant.

- **Refine Network:**

  The second stage is a CNN Refine Network(R-Net). Using non-maximum suppression, it further decreases the number of boxes and merges overlapping candidates (NMS).

- **Output Network:**

  In the third step, the Output Network does more of the same things as R-Net, but it adds the 5-point landmark of eyes, nose, and mouth to the final bounding box containing the recognized objects.

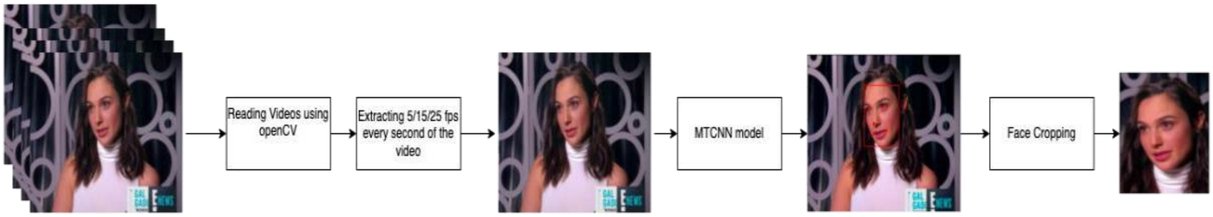
Reading Videos using openCV
Extracting 5/15/25 fps every second of the video
MTCNN model
Face Cropping

Figure 3.4 Data Input pipeline using MTCNN

The complete pipeline for the Multi-Task Cascaded Convolutional Neural Network explaining its working is shown in the Figure 3.5 for better understanding of the creation of bounding box around the faces in the frames

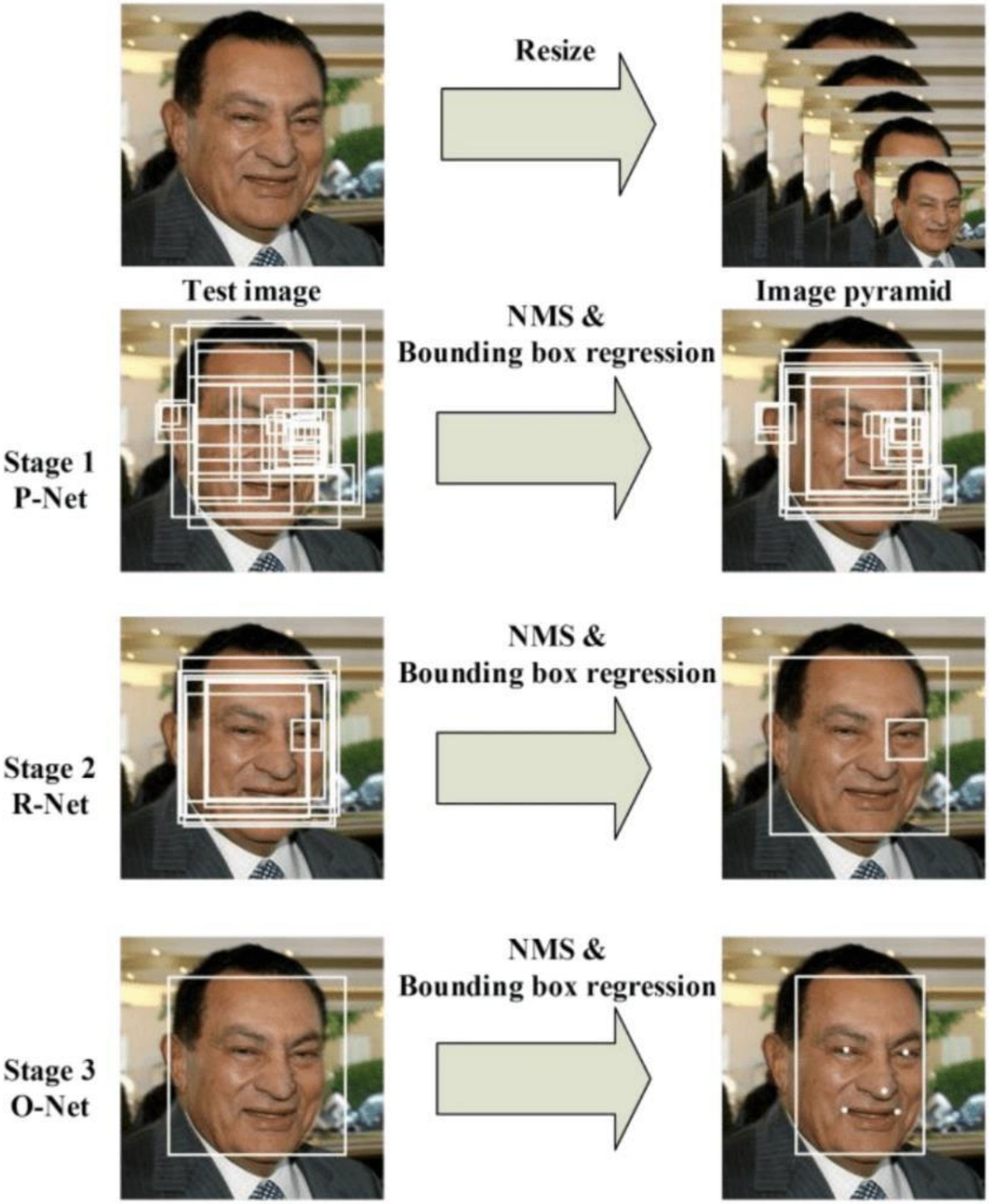

Figure 3.5 Pipeline for the Multi-Task Cascaded Convolutional Neural Network Taken, (Zhang, Zhang, Li, & Qiao, 2016)

3.4 Data Augmentation:

Computer vision models including Convolutional and Transformer models can perform better with few augmentations on the data. The frames extracted from the videos can have only a few variations as 25 frames per second may not have many variations. Though the variations would still be present, modifying them using data augmentation techniques can help the model to learn more robustly and reduce biases.

Two basic Augmentation techniques are used for 50% of the total input data namely image rotation and image flipping.

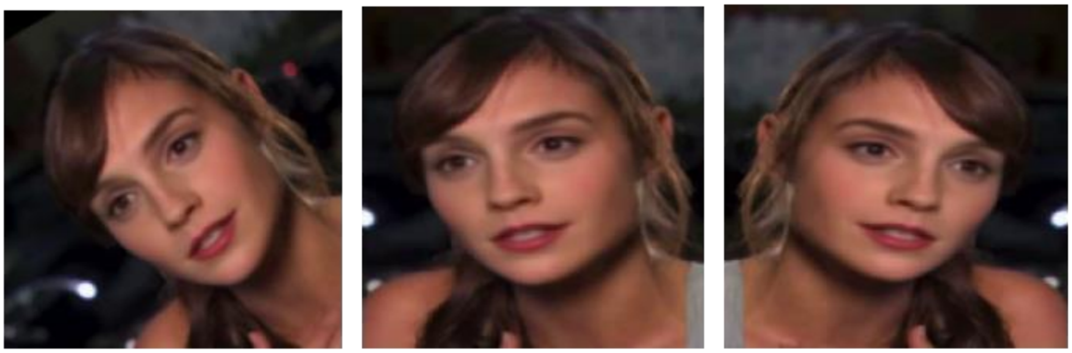

Figure 3.6 Augmented Input Images

The above-mentioned augmentations help the model to generalize the problem much better and can increase performance.

3.4 Experiment Setup:

The experiment is run on Google's TPU cloud with 8-core TPU v-2 type hardware accelerator that can deliver up to 180 teraflops and includes 64 GB of high-bandwidth memory and the models are created and trained using Python, TensorFlow 2, Flax, JAX

Flax is a high-performance neural network library and ecosystem for JAX that was developed by the Brain Team in Google. It contains the codebase for Neural Network API, Optimizers, Utilities all developed using JAX.

JAX is an automatic differentiation (AD) toolbox developed by a group of people at Google Brain and the open-source community. It aims to bring differentiable programming in NumPy-style onto TPUs which could make the matrix operations and differentiation operations that are essential for training a model run on TPU hardware. For example, normal numpy operations are performed on the CPU’s and can be accelerated using GPUs, but the TPU architecture is built totally different focusing only on ML models. So JAX numpy (jnp) can share the data among the cores and run at high speeds.

On the highest level JAX combines the previous projects XLA & Autograd to accelerate linear algebra-based projects.

XLA (Accelerated Linear Algebra) is a domain-specific linear algebra compiler that might potentially speed up TensorFlow models with no source code changes.

On the other side, Autograd supports automatic differentiation for a significant number of standard Python features. At any moment in time, it simplifies the derivative formulation of a compositional function.

3.5 ResNet-18 Model:

According to the universal approximation theorem (Kratsios, 2019), given enough capacity, it is known that a feedforward network with a single layer is sufficient to represent any function. However, this could lead to a massive network, and these massive networks are prone to overfitting the data. Therefore, there is a common trend in the research community that the network architecture needs to go deeper.

Since the introduction of AlexNet, state-of-the-art CNN architectures have gone deeper.

AlexNet has 5 layers (Liu & Deng, 2015), whereas VGG has 19 layers and Inceptionv1 has 22 layers (Szegedy et al., 2015). However, deep networks became harder to train because of the vanishing gradient problem. This is because the gradient is backpropagated to earlier layers of the deep network, repeated multiplication may make the gradient infinitely small. This results in the vanishing gradient problem and as the network goes deeper, it may start to degrade.

To overcome this issue Residual networks were introduced, these Resnet uses a concept called "identity shortcut connection" that skips one or more layers and is added to the forward network like the figure shown below.

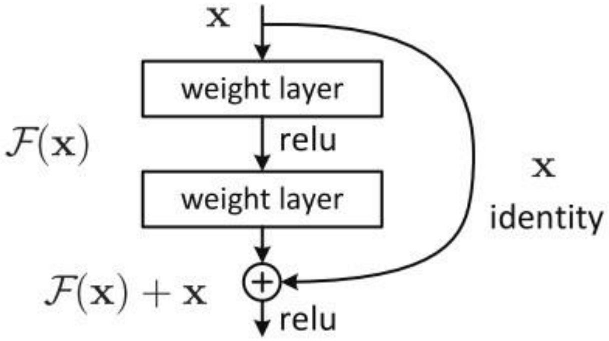
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.7 Residual Network Block

For the input x, F(x) is the output from the first activation relu function, in case of residual network an identity shortcut (x) which is the input will be later added to the network so that the model doesn't lose the data from the initial stages of the model and retain them.

3.5.1 Implementation:

Different versions of ResNet models include 18, 50, 101, 152 each with a greater number of layers than their predecessor with Residual Skip connections. In this research ResNet-18 has been implemented and the model architecture is shown in Figure 3.8. The 4 layers of convnet blocks each have filters of size 3x3 reduces the image size. In between the layers there are residual blocks which is used to connect the input of a layer directly to the output of a layer after skipping a few connections.

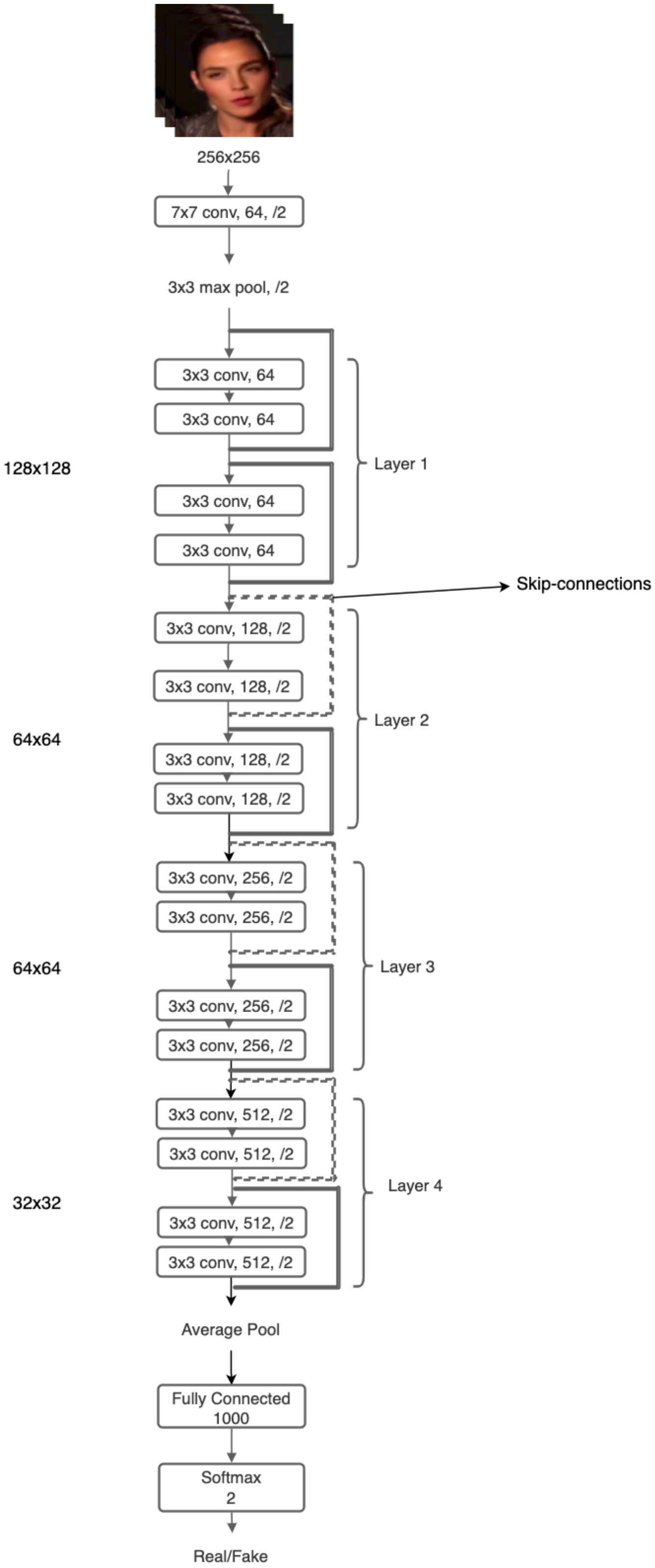

Figure 3.8 ResNet-18 Architecture

3.5.2 Hyperparameters Tuning:

Learning Rate

Learning rate is considered one of the most important hyperparameters for training deep neural networks as it essentially sets the pace for the model to learn. But choosing it can be quite hard, so instead of statically setting a learning rate, the research uses “Cosine decay learning rate”.

“Cosine decay learning rate” has the effect of starting with a large learning rate that is relatively rapidly decreased to a minimum value before being increased rapidly again. The resetting of the learning rate simulates a restart of the learning process, and the use of good weights as the restart's beginning point is known as a "warm restart," as opposed to a "cold restart," which uses a new set of small random values as a starting point.

$$\eta_t = \eta_{mini} + 1/2\ (\eta_{maxi} - \eta_{mini})(1+\cos(T_{cur}T_i\ \pi)) \qquad \ldots \text{ Equation 3.1}$$

where η_{mini} and η_{maxi} are ranges for the learning rate, and T_{cur} accounts for how many epochs have been performed since the last restart.

Cross-Entropy Loss:

As the model is used to classify binary classes i.e., real, and fake, “Cross Entropy loss” is used. It evaluates a classification model's output, which is a probability value between 0 and 1.

In addition, for mixed precision gradients, dynamic loss is employed as a scaling method. Gradient computations in float16 will cause numerical difficulties for many models since small/large gradients will be flushed to zero/infinity. Dynamic loss scaling is an algorithm that aims to find the largest scalar multiple for which the gradient does not overflow. This way the risk of underflow is minimized, and Adam Optimizer is used as optimizer.

Hyperparameters values are as follows:

Steps - 3000

Learning rate = 0.01, warmup steps = 9

Decay type: cosine

Batch size = 64

3.6 Vision Transformers (ViT):

The self-attention layer in ViT lets embedding information throughout the entire image. To reproduce the image's structure, the model additionally employs training data to represent the relative locations of image patches (Dosovitskiy et al., 2021).

The transformer encoder includes:

- MSP (Multi-Head Self Attention Layer): It concatenates all the attention outputs from the self-attention blocks to the right dimensions in a linear fashion. The several attention heads in an image aid in the training of local and global dependencies.
- MLP Layer (Multi-Layer Perceptron): This layer consists of two layers, each containing a Gaussian Error Linear Unit (GELU).
- LN (Layer Norm): It is added before each block and does not include any additional dependencies between the training photos. As a result, training time is reduced, and overall performance is improved.

$$\begin{aligned}\mathrm{GELU}(x) &= xP(X \leq x) = x\Phi(x) \\ &\approx 0.5x\left(1+\tanh\left[\sqrt{2/\pi}\left(x+0.044715x^3\right)\right]\right)\end{aligned}$$

Equation 3.2 GELU Formula

Figure 3.9 shows a high-level view of the model. A typical Transformer takes a 1D series of token embeddings as input. To handle 2D pictures, the image x is reshaped into a sequence of flattened 2D patches $x^{p} \in R^{N\times(P^{\wedge}2\ \cdot C)}$, where (H, W) is the original image's resolution, C is the number of channels, (P, P) is the resolution of each image patch and the number of patches produced is $N = H.W/P^{2}$ (Dosovitskiy et al., 2021). The patches are flattened and projected to D dimensions with a trainable linear projection because the Transformer utilizes a constant latent vector size D throughout all its layers. This is referred to as the patch embeddings, which is the outcome of this projection.

This transformer follows the similar architecture of (Dosovitskiy et al., 2021) and uses the pre-trained weights created by google after training with 14 million images from Imagenet21k dataset in their TPU clusters. This pre-trained transformer performs competitively to the state-of-the-art CNN models. Hence these weights are loaded as pre-trained weights and are then fine-tuned for the purposes of Deepfake detection using Transfer learning technique.

3.6.1 Implementation

The Vision Transformer model's architecture is explained in the following steps:

1. Split an image into patches which is kept as 16
2. The image patches are then flattened
3. Lower-dimensional linear embeddings are created from the afore-mentioned flattened image patches which can be referred as "Patch Embeddings"
4. Positional embeddings are then added to the image patches sequences in order for them to maintain their positional information which becomes crucial when identifying the irregularities on the frames.
5. An extra learnable class embedding is then prepended to the positional embeddings. This embedding is used to predict the input frame's category after being updated by self-attention.
6. The sequence is fed as an input to a state-of-the-art transformer encoder

7. An MLP head is just stacked on top of the learnable class embedding output from the transformer encoder.
8. Finally, Classification is performed.

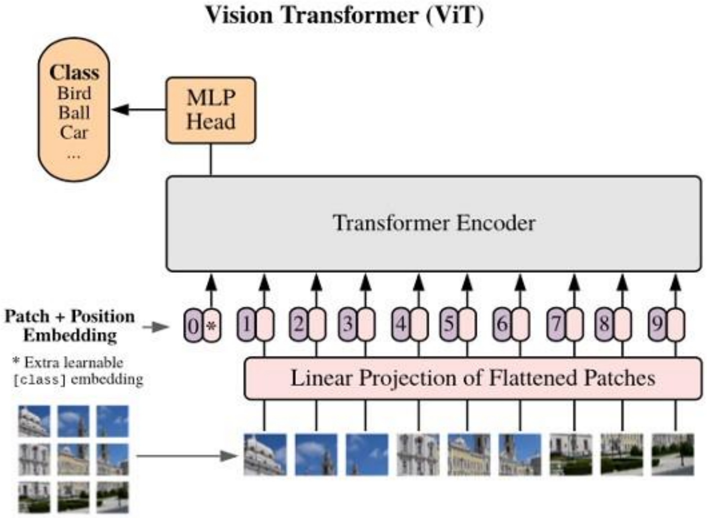
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 3.9 Vision Transformer (ViT) Architecture (Dosovitskiy et al., 2021)

The optimizer's choice, network depth, and dataset-specific hyperparameters all affect the performance of a vision transformer model. CNNs are less difficult to optimize than ViT.

The flattened patches are turned into a sequence of tokens with positional encoding that are then inputted into the transformer encoder. This positional embedding will help the transformer to learn the inductive bias for the task it is being trained for, it is always beneficial to help the learning process.

After then, the transformer uses the attention mechanism to generate a series of output tokens. A projector eventually connects the output tokens to the feature map. The latter enables the investigation potentially important pixel-level details and hence lowering the total number of tokens that needs to be examined and thereby lowering costs significantly. (Dosovitskiy et al., 2021).

The only change that is introduced when fine-tuning is to disregard the MLP layer and add a new D*K layer, where K is the number of classes in the dataset which is 2.

Transformer Encoder:

The transformer encoder module contains a Multi-head Self Attention (MSA) block and a Multi-layer Perceptron (MLP) block. The MSA block splits the input embeddings into multiple heads which is set as **12** so that each head can learn different levels of self-attention. All the 12 heads output from the MSA will be concatenated and passed through the MultiLayer Perceptron (MLP) with 3072 dimension and 12 layers.

Along with a Layer Norm, a Residual-skip connection is also used after every block to overcome the vanishing gradient problem.

A self-attention dropout is also set as 0.02, where elements are randomly dropped out of the SoftMax in the attention equation.

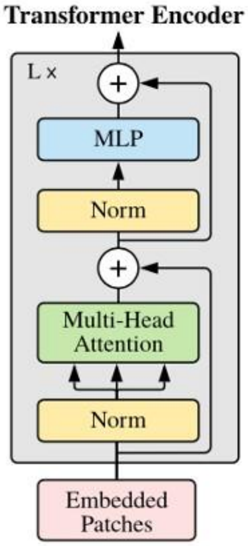

Figure 3.10 Transformer encoder (Dosovitskiy et al., 2021)

In Transformers, "attention distance" is equivalent to "receptive field size" in CNN’s. In the lower layers, average attention distance is very varied, with some heads focusing on large areas of the image while others focus to small regions at or near the query site (Dosovitskiy et al., 2021). All heads' attention distance rises as depth increases. Most heads attend broadly among tokens in the second half of the network.

For final detection, Softmax function is used to the MLP head output to create the weight values between 0 and 1.

3.6.2 Hyperparameters tuning

Like the ResNet-18 model, Cross Entropy Loss and Cosine Decay learning rate are utilized as the loss function and momentum optimizer is used for optimizing the model which were suggested by (Dosovitskiy et al., 2021) when pre-training the model.

Gradient descent does not exactly provide the direction in which the loss function is headed i.e., the derivative of the loss function. Therefore, it might not always be headed in the optimal direction. This is primarily because the earlier derivatives of the loss function act as a noise in the later stages of updating the weights. This causes slow training and convergence.

Using Momentum technique as the optimizer helps solve this issue of slow convergence.

The momentum approach extends the Gradient Descent method by providing a new variable V that represents velocity and a friction coefficient/smoothing constant β that helps in regulating the value of V and prevents overshooting the minima while also allowing faster convergence.

Hyperparameter values are as follows:

Base learning rate: 0.03

Batch size: 512

Steps: 3000

Decay type: cosine

Evaluate every: 100 steps

Patches: 16 x 16 Image
size: 384 x 384
Transformer:

- **Mlp dim: 3072**
- **Num heads: 12**
- **Num layers: 12**

Transformers work robustly when pre-trained with huge datasets (Paul & Chen, 2021) so this model uses Imagenet 21k weights as pre-trained. Imagenet 21k has over 14 million images which was used to train the model and those weights are loaded to the model and is then fine-tuned to achieve optimal performance. This transfer learning can help reduce the computational cost of the model tremendously.

3.7 Evaluation

The models are tested with the test split dataset from CELEB-DF-v2 dataset which contains 331 fake videos and 176 real videos with almost 1:2 ratio.

Accuracy is chosen as the metric to be used to compare the models as every frame in the model is modified and all of them are extracted and placed under their respective directory named with their class label.

The training loss of the model is also calculated as it could explain the models' ability to distinguish between the binary class and classify them.

An ablation study is done on the effect of the frames per second on the models' performance namely 5/15/25 frames per second. Three variations of the dataset are created with different frame rates and has been used to train the model and test it.

The results are logged using wandb, which centralizes all the results from different versions of the models and makes comparison easy.

T-test is performed on the results to figure out its statistical significance.

3.6 Summary

To conduct the experiment, the CELEB-DF-v2 dataset has been gathered and is preprocessed and frames are extracted which is then passed to a ResNet-18 model, and a Vision Transformer model which are set with suitable model hyper-parameters and trained. The Vision Transformer model used the Transfer Learning technique to reduce the training cost. In the case of the ResNet-18 and the Transformer model, Accuracy and loss metrics score are used for analyzing the performance of the model. The results obtained are discussed in Chapter 4.

4 RESULTS, EVALUATION AND DISCUSSION

The results of the experiment will be examined in this section and the hypothesis will be tested. For the hypothesis testing t-test will be used between the accuracy metrics obtained from the ResNet-18 model and Vision Transformer model. Furthermore, the strengths and weaknesses of the models are based on the findings while conducting the design.

4.1 Results:

The models that were implemented in Section 3, are successfully compiled, and trained. The loss and the accuracy of the model are evaluated and used to compare the ResNet model and the Vision Transformer model.

CELEB-DF-v2 datasets "Test" videos are used for testing the models. Both the models were passed the same dataset and the results below are inferred.
Models trained with only 5 frames per second have performed the best and even with higher 25 frames per second the models were able to achieve almost similar performances.

Input Variations	ViT-Accuracy	ResNet-18 Accuracy	ViT- loss	ResNet-18 loss
5fps	96.95%	91.78%	0.034	0.004
15fps	96.97%	89.41%	0.031	0.020
25 fps	97.07%	90.72%	0.032	0.010

Table 4.1 Results

Comparing State-of-the-arts with the Transformers Model and ResNet-18 model

Model Name	Accuracy
XceptionNet- Full Image- (Andreas Rössler et al.,2019)	74.5%
Conv-LSTM, Eye Blinking- (Jung et al.,2020)	87.5%
Meso-Net – (Afchar et al.,2019)	87.3%
Modified AlexNet – (Xie et al., 2020)	98.85%
ResNet-18 – This paper	**91.78%**
Vision Transformer – This paper	**97.07%**

Table 4.2 Comparing State-of the-art Deepfake Detection Models

4.2 Discussion:

Both the ViT and ResNet-18 models are trained with dataset created by extracting frames at different frame rates namely 5/5/25 fps. The trained model is then tested with the test split of the dataset with 1:2 ratio of real and fake images.

The results of the test dataset are logged on wandb dashboard and the best accuracy of the model is noted from the graph and added to the Table 4.1. The loss of the model at that accuracy is also noted to understand how well the model can distinguish between real and modified images and classify them.

From Table 4.1, it can be observed that the ViT model outperformed the ResNet-18 model in terms of accuracy. The ViT model with 25fps as input achieved 97.07% accuracy whereas ResNet-18 model achieved 91.78% accuracy with 5fps data.

WANDB Graphs:

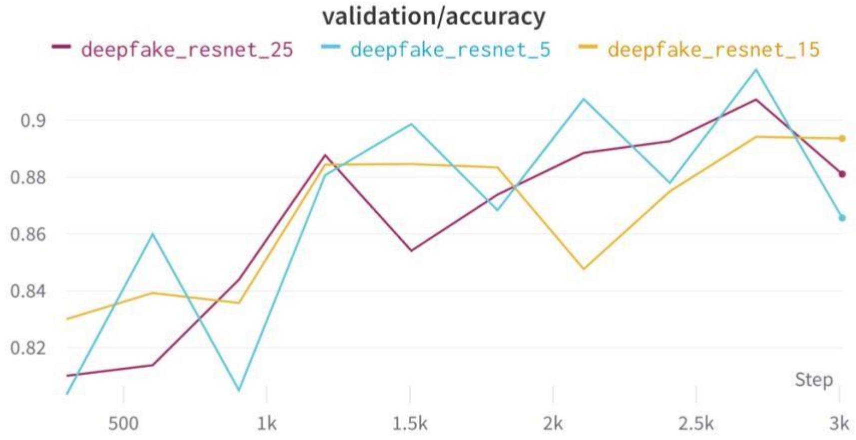

Figure 4.1 ResNet-18 Accuracy

From the Figure 4.1, it can be observed that the ResNet-18 model had a steep increase in the accuracy curve in case of 5fps dataset, for larger datasets the accuracy increases slowly compared to the smaller 5fps dataset. The highest accuracy is also achieved when the model was trained using 5fps dataset. This shows the Convolutional ResNet model can perform better with comparatively smaller data size.

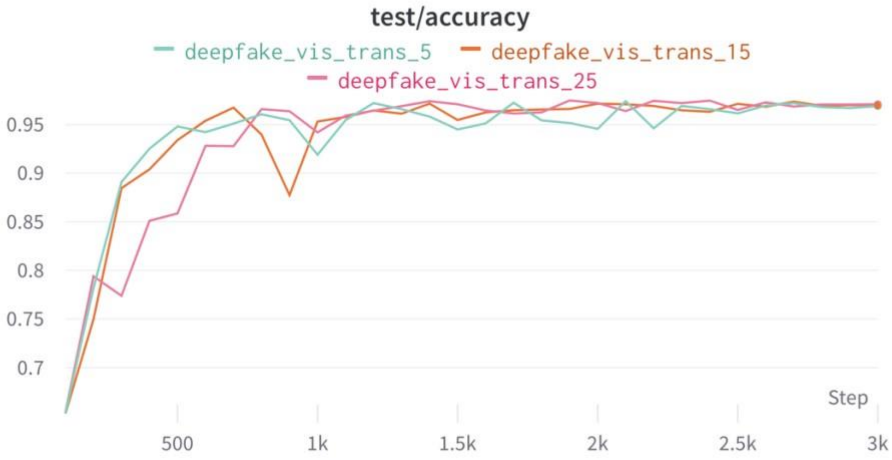

Figure s.2 Vision Transformers Accuracy

From the Figure 4.2, it can be observed that regardless of the dataset size the ViT model performed almost similarly providing consistent results. The results start to converge and reach a flat surface after 2000 steps. All the variations achieve around 96% with minimal difference.

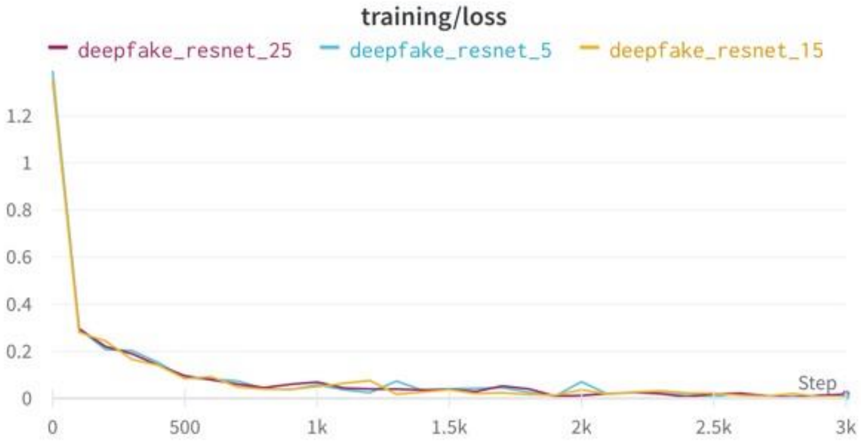

Figure 4.3 ResNet-18 Loss

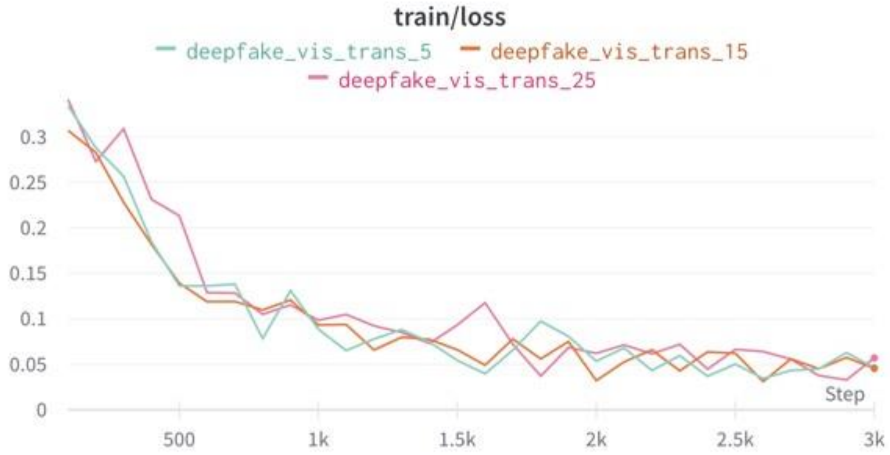

Figure 4.4 ViT- Loss

The models' training loss can explain the models learning of the model over time. The more exponential decay in the ResNet-18 model is the result of Cosine loss learning rate which sets the learning rate high and decreases it gradually. The ViT has much shallow curve compared to ResNet-18, this implies that the model's learning rate is adjusted optimally.

The use of the Vision Transformer to compete with the Convolutional Network helped the network use "attention" to focus on crucial facial features in the frames and reduce the computational considerably and provide better performance as the positional embeddings helped the model learn about the different features in a face and the significance of their position, this was not previously possible in Convolutional Network. The Transformer Network also can work more robustly than the Convolutional Model.

4.3 Statistical Evaluation:

T-test is done to statistically evaluate the models. This statistical result will be used to validate the hypothesis, whether it is significant or not.

The p-val of the t-test is 0.012 < 0.05, which implies that the mean difference of the accuracy of a Vision Transformer model and the ResNet-18 model is statistically significant. So, null hypothesis is rejected, and alternate hypothesis is accepted. This proves that “IF a custom MTCNN face extraction input pipeline is used to extract faces from the frames of a video and is used for classifying deepfake videos using a Vision Transformer, THEN the mean difference between the accuracy of the model and the ResNet CNN model will be statistically significant (p-val < 0.5).

5 CONCLUSION

5.1 Research Overview

The main objective of this research is to investigate the use of Vision Transformer for the Deepfake classification task. A Vision transformer model and a ResNet-18 CNN model are created. These models are trained and evaluated using the CELEB-DF-v2 dataset. The accuracy of the model is used to find if the Transformers performance is statistically significant using t-test.

5.2 Problem Definition

Many state-of-the-art Deepfake classification networks were discussed in Section 2, almost all of them use Convolutional networks as a part of their architecture. But CNN does not encode the relative position of different features. To encode the combination of these features, large filters are necessary. To track long-range dependencies within an image, large receptive fields are required. While increasing the size of the convolution kernels increases the network's representational capacity, it simultaneously reduces the computational and statistical efficiency gained by employing local convolutional structure. Vision Transformer architecture is used to solve the positional encoding problem without increasing the computational cost and thereby can provide better results compared to Convolutional Architectures.

5.3 Design/Experimentation, Evaluation & Results

To prove the hypothesis, two models are created namely a Vision Transformer based model and a Convolutional model with residual connections ResNet-18.

The models are created using TensorFlow 2, python and Flax on a Google Cloud TPU-v2 with 8 cores. Flax is a codebase created by the Brains of Google to utilize the full power of TPU's, it created models which can run on TPU architecture and exponentially decrease the training time of the models.

CELEB-DF-v2 dataset is used to train and evaluate the models, it is chosen as it has high quality versatile deepfake videos. These videos are pre-processed using a custom pipeline with MTCNN model to extract the faces from each frame of the video. The CELEB-DF-v2 dataset videos have modifications only on their faces. So, the MTCNN model extracts the required faces among the background noises and the faces are stored as images in the two directories real and fake. This pre-processing is done with one changing parameter namely the frames per second. Each video in the dataset is around 13 second long with 30fps. So, the number of frames extracted every second is kept as 5, 15, 20 fps accordingly. The different sizes of the dataset are used to train the model and the results are inferred.

The pre-processed datasets are used to train the models with some data augmentation like image rotation and image flipping. This augmented data is loaded using TensorFlow datasets so the results can be replicated later. The hyperparameters of the models are tuned and are explained in Section 3. The models are trained for around 3000 steps and the CELEB-DF-v2 test data is used to evaluate the models with the same data pre-processing. Accuracy and the loss of the models are logged during training and testing.

A t-test is later done on the results of the models to evaluate their statistical significance and its result is used to either accept or refute the hypothesis.

5.4 Contributions and Impact

The Vision Transformer implementation takes one step forward towards the use of Vision Transformer for various image classification tasks and may in future become the de-facto architecture in computer vision. This research uses the experimental Flax codebase to run the models on Google's TPU cloud with 8-core TPU v-2 type hardware that can deliver up to 180 teraflops and includes 64 GB of high-bandwidth memory.

The use of TPU has not become more common among the Machine learning community and has only some features and many bugs which can be overcome through many contributions from the research community

5.5 Future Works and Recommendations:

The Vision Transformer performed competitively to some of the state-of-the-art deepfake classification models. But further improvement on this can be done. For example, robustness of the transformer model can take not only an image but rather feature vectors can also be passed as an input, these feature vectors can be obtained from previous Convolutional layers and thereby utilizing the benefits of both Convolutional and Transformer architectures.

This research utilized CELEB-DF-v2 dataset which had face manipulations done in all frames of a video and only one face is present in every video. In the real world, such ideal conditions may not always be met. So, the problem definition should be widened to include all possibilities such as multiple faces in a frame, only a portion of the video is manipulated which could drastically affect this model's performance.

The use of a transformer block makes this architecture versatile; it is possible to add multiple transformers and create “cross-attention” among the transformers when compared to just using “self-attention”. This “cross-attention” could also help in creating architectures that could find the temporal anomalies in a video and can classify videos much more efficiently compared to existing techniques.

Models can be created using JAX and can be added to the FLAX codebase to increase the use of TPU hardware and train experimental models much faster and efficiently with variations.

BIBLIOGRAPHY:

Montserrat, D. M., Hao, H., Yarlagadda, S. K., Baireddy, S., Shao, R., Horvath, J., Bartusiak, E., Yang, J., Guera, D., Zhu, F., & Delp, E. J. (2020). Deepfakes Detection with Automatic Face Weighting. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2851–2859. https://doi.org/10.1109/CVPRW50498.2020.00342

Jung, T., Kim, S., & Kim, K. (2020). DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. *IEEE Access*, *8*, 83144–83154. https://doi.org/10.1109/ACCESS.2020.298 8660

Kumar, A., Bhavsar, A., & Verma, R. (2020). Detecting Deepfakes with Metric Learning. *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, 1–6. https://doi.org/10.1109/IWBF49977.2020. 9107962

Matern, F., Riess, C., & Stamminger, M. (2019). Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 83–92. https://doi.org/10.1109/WACVW.2019.00 020

Yang, X., Li, Y., & Lyu, S. (2019). Exposing Deep Fakes Using Inconsistent Head Poses. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8261–8265. https://doi.org/10.1109/ICASSP.2019.868 3164

Li, Yuezun & Lyu, Siwei. (2018). Exposing DeepFake Videos By Detecting Face Warping Artifacts.

Ciftci, U. A., & Demir, I. (2020). FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. https://doi.org/10.1109/TPAMI.2020.3009 287

Lyu, S. (2020). Deepfake Detection: Current Challenges and Next Steps. *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. https://doi.org/10.1109/icmew46912.2020. 9105991

Akhtar, Z., & Dasgupta, D. (2019). A Comparative Evaluation of Local Feature Descriptors for DeepFakes Detection. *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–5. https://doi.org/10.1109/HST47167.2019.9 033005

Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. (2019). *Deep Learning for Deepfakes Creation and Detection*.

Karandikar, A. (2020). Deepfake Video Detection Using Convolutional Neural Network. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), 1311–1315. https://doi.org/10.30534/ijatcse/2020/6292 2020

Huang, Y., Juefei-Xu, F., Wang, R., Guo, Q., Ma, L., Xie, X., Li, J., Miao, W., Liu, Y., & Pu, G. (2020). FakePolisher: Making DeepFakes More Detection-Evasive by Shallow Reconstruction. *Proceedings of the 28th ACM International Conference on Multimedia*, 1217–1226. https://doi.org/10.1145/3394171.3413732

Fernandes, S. L., & Jha, S. K. (2020). Adversarial Attack on Deepfake Detection Using RL Based Texture Patches. *Computer Vision – ECCV 2020 Workshops*, 220–235. https://doi.org/10.1007/978-3- 030-66415-2_14

Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: a Compact Facial Video Forgery Detection Network. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–7. https://doi.org/10.1109/wifs.2018.8630761

Deshmukh, A., & Wankhade, S. B. (2020). Deepfake Detection Approaches Using Deep Learning: A Systematic Review. *Intelligent Computing and Networking*, 293–302. https://doi.org/10.1007/978-981- 15-7421-4_27

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

Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020). Emotions Don't Lie. *Proceedings of the 28th ACM International Conference on Multimedia*, 2823–2832. https://doi.org/10.1145/3394171.3413570

Korshunov, P., & Marcel, S. (2019). Vulnerability assessment and detection of Deepfake videos. *2019 International Conference on Biometrics (ICB)*, 1–6. https://doi.org/10.1109/icb45273.2019.898 7375

Mitra, A., Mohanty, S. P., Corcoran, P., & Kougianos, E. (2021). A Machine Learning Based Approach for Deepfake Detection in Social Media Through Key Video Frame Extraction. *SN Computer Science*, *2*(2), 98. https://doi.org/10.1007/s42979-021- 00495-x

Guera, D., & Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 1–6. https://doi.org/10.1109/avss.2018.8639163

Guarnera, L., Giudice, O., & Battiato, S. (2020). DeepFake Detection by Analyzing Convolutional Traces. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2841–2850. https://doi.org/10.1109/CVPRW50498.2020.00341

Verdoliva, L. (2020). Media Forensics and DeepFakes: An Overview. *IEEE Journal of Selected Topics in Signal Processing*, *14*(5), 910–932. https://doi.org/10.1109/JSTSP.2020.30021 01

Wan, X., Ren, F., & Yong, D. (2019). Using Inception-Resnet V2 for Face-based Age Recognition in Scenic Spots. *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 159–163. https://doi.org/10.1109/CCIS48116.2019.9073696

Chintha, A., Thai, B., Sohrawardi, S. J., Bhatt, K., Hickerson, A., Wright, M., & Ptucha, R. (2020). Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection. *IEEE Journal of Selected Topics in Signal Processing*, *14*(5), 1024–1037. https://doi.org/10.1109/JSTSP.2020.29991 85

Jiang, L., Li, R., Wu, W., Qian, C., & Loy, C. C. (2020). DeeperForensics-1.0: A LargeScale Dataset for Real-World Face Forgery Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2886–2895. https://doi.org/10.1109/CVPR42600.2020. 00296

Li, L., Bao, J., Yang, H., Chen, D., & Wen, F. (2020). Advancing High Fidelity Identity Swapping for Forgery Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5073– 5082. https://doi.org/10.1109/CVPR42600.2020.00512

Guo, Z., Yang, G., Chen, J., & Sun, X. (2021). Fake face detection via adaptive manipulation traces extraction network. *Computer Vision and Image Understanding*, *204*, 103170. https://doi.org/10.1016/j.cviu.2021.103170

Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., & Holz, T. (2020). Leveraging Frequency Analysis for Deep Fake Image Recognition. *International Conference on Machine Learning*, 3247– 3258. http://proceedings.mlr.press/v119/frank20a.html

Caldwell, M., Andrews, J. T. A., Tanay, T., & Griffin, L. D. (2020). AI-enabled future crime. *Crime Science*, *9*(1), 14. https://doi.org/10.1186/s40163-020- 00123-8

Carlini, N., & Farid, H. (2020). Evading Deepfake-Image Detectors with White- and BlackBox Attacks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*
Workshops (CVPRW), 2804–2813. https://doi.org/10.1109/CVPRW50498.202 0.00337

Maksutov, A. A., Morozov, V. O., Lavrenov, A. A., & Smirnov, A. S. (2020). Methods of Deepfake Detection Based on Machine Learning. *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 408–411. St. Petersburg and Moscow, Russia: IEEE. https://doi.org/10.1109/EIConRus49466.2020.90390577

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In International Conference on Learning Representations, 2018.

Shome, D., Kar, T., Mohanty, S., Tiwari, P., Muhammad, K., AlTameem, A., … Saudagar, A. (2021). COVID-Transformer: Interpretable COVID-19 Detection Using Vision Transformer for Healthcare. International Journal of Environmental Research and Public Health, 18(21), 11086. https://doi.org/10.3390/ijerph182111086

Zhmoginov, A. and Sandler, M., 2016. Inverting face embeddings with convolutional neural networks. arXiv preprint arXiv:1606.04189.

Dale, K., Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W. and Pfister, H., 2011, December. Video face replacement. In Proceedings of the 2011 SIGGRAPH Asia Conference (pp. 1-10).

J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and ¨ M. Nießner. Face2Face: Realtime Face Capture and Reenactment of RGB Videos. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2016.

Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7184–7193.

H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Perez, C. Richardt, M. Zoll ´ ofer ¨ , and C. Theobalt. Deep Video Portraits. ACM Transactions on Graphics (TOG), 37(4):163, 2018.

A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh. RecycleGAN: Unsupervised Video Retargeting. In ECCV, 2018.

A. Pumarola, A. Agudo, A. Martinez, A. Sanfeliu, and F. Moreno-Noguer. GANimation: Anatomically-aware Facial Animation from a Single Image. In Proceedings of the European Conference on Computer Vision (ECCV), 2018.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. IEEE Computer graphics and applications, 2001

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 1–11. https://doi.org/10.1109/ICCV.2019.00009

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3204–3213. Seattle, WA, USA: IEEE. https://doi.org/10.1109/CVPR42600.2020.00327

Paul, S., & Chen, P.-Y. (2021). Vision Transformers are Robust Learners. ArXiv:2105.07581 [Cs]. Retrieved from http://arxiv.org/abs/2105.07581

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 1–11. https://doi.org/10.1109/ICCV.2019.00009