National College of Ireland

Deepfake Detection using Bayesian Neural Networks

MSc Research Project
Data Analytics

Parag Suresh Joshi
Student ID: x19212071

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Parag Suresh Joshi
Student ID:	x19212071
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Noel Cosgrave
Submission Due Date:	16/12/2021
Project Title:	Deepfake Detection using Bayesian Neural Networks
Word Count:	6076
Page Count:	20

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:	
Date:	30th January 2022

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):	

Deepfake Detection using Bayesian Neural Networks

Parag Suresh Joshi
x19212071

Abstract

Deepfakes are images or videos that are manufactured by using artificial algorithms, image processing, and face swapping. Deepfakes, which use artificial intelligence (AI) to represent someone speaking or doing things that did not take place, have the potential to have a significant negative impact on society. The study reviews known deep learning approaches for Deepfake detection and seeks to present an additional way for Deepfake video identification using Bayesian Neural Networks. Deep learning is constantly advancing in terms of both producing and identifying deepfakes. A deepfake detection model built with an earlier dataset may become obsolete over time, necessitating the development of a new detection approach. Results of the research indicate that the model provides an average performance with an accuracy of 58 percent on an untrained dataset.

1 Introduction

Machine learning has grown at a breakneck pace over the last decade, and data science has embraced the technology in a variety of domains for a plethora of different applications. The advancement of machine learning technology sometimes leads to the data protection and security issues. With the advent of content sharing apps, it's more important than ever to ensure the data's integrity. Innovative solutions are being created using Machine Learning to automate the inspection of shared content and take relevant actions where necessary.

Deepfakes are a sort of "synthetic media" that includes images, sound, and video that appear to have been created using traditional methods but were actually created using deep learning algorithms. The term "Deepfake" is derived from the phrases "deep learning" and "fake," and refers to information generated by an artificial neural network. Any Deepfake content is usually created by artificial intelligence that appears to be genuine to a human. Deep learning algorithms are utilized in the majority of Deepfake content to swap faces in video and digital data to generate realistic-looking fake content.

1.1 Background

Deepfake technology may be used for a variety of creative and useful purposes. These include reanimation of historical individuals for teaching purposes, accurate video dubbing of foreign films[1], and virtually experimenting with clothes while shopping[2]. There

[1] https://variety.com/2019/biz/news/ai-dubbing-david-beckham-multilingual-1203309213/

[2] https://www.forbes.com/sites/forbestechcouncil/2019/05/21/gans-and-deepfakes-could-revolutionize-
?sh=581167cf3d17

are also various internet communities dedicated to making entertaining deepfake memes[3]. Various videos of Nicholas Cage in films in which he did not star, such as 'Fight Club' and 'The Matrix,' as well as footage of Jim Carrey playing Jack Nicholson in 'The Shining' surfaced on the internet.

Despite the technology's potential applications, deepfakes are infamous for their unethical and harmful aspects. One such example occurred at the end of 2017, when a Reddit user known as 'deepfakes' used deep learning to swap celebrities' faces into obscene movies and then put them on the internet. The finding sparked a media frenzy, and a slew of new deepfake videos sprung up as a result. BuzzFeed, in 2018, published a deepfake video of former President Barack Obama giving a speech which was created with the software (FakeApp) created by a Reddit user[4], and it aroused concerns about identity fraud, impersonation, and the propagation of misinformation on social networking sites. According to a Wall Street Journal story, audio Deepfake content was utilized to imitate the voice of an executive to try a USD 243K fraud transfer[5]. Deepfakes, like other synthetic media and fake news, have the more sinister effect of developing a zero-trust society, in which people are unable or unwilling to differentiate between truth and lie, making it imperative to recognize them before they have any long-term consequences.

Mirsky and Lee (2020) offer a brief summary of how Deepfakes are created using a variety of approaches in their research. Using autoencoders is one approach for creating any Deepfake content. An autoencoder is made up of two parts: an encoder that encodes source data into a lower-dimensional representation and a decoder that reconstructs the lower-dimensional representation back to the original data. To develop Deepfake content, first both the encoder and decoders for a source media content are constructed, then an encoder for a target media content is created, and the output of the target encoder is supplied as input to the source decoder. The output of the source decoder would be a media content having mixed features from the source and target media contents.

Deepfake material is now being created using a more advanced concept called generative adversarial networks (GANs), in which two artificial neural networks compete against each other to produce realistic-looking media content. These two artificial neural networks, dubbed the 'generator' and 'discriminator,' are effectively trained on the identical media data sets, with the generator being charged with creating fake material and the discriminator with detecting it. The generator improves the quality of the fake content based on the discriminator's feedback until the forgery is no longer detectable by the discriminator. As the generator improves its forged content, the discriminator improves its forgery detection, which further improves the generator's content, and eventually, realistic-looking pseudo media content is developed. Furthermore, because GANs can be trained effectively, all existing detection approaches can be configured in the discriminators, making Deepfake detection a challenging task.

1.2 Bayesian Paradigm

Jospin et al. (2020) discuss the notion of Bayesian Neural Networks in this research article, which highlights the limits of artificial neural networks and the answers that Bayesian Neural Networks give on the same. The article focuses on two major difficulties with traditional deep learning models, (A) Deep learning models can be over-confident in

[3] https://www.reddit.com/r/SFWdeepfakes/

[4] https://www.buzzfeed.com/craigsilverman/obama-jordan-peele-deepfake-video-debunk-buzzfeed

[5] https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11

their predictions. Even if the projections are incorrect, they are overconfident, and (B) The point estimate technique does not give adequate transparency on the production of output for out-of-training data points, and so the system's behavior is uncertain in such instances.

For these reasons, attempts have been made to overcome these restrictions using various strategies, one of which is the use of stochastic neural networks for uncertainty estimates. Stochastic neural networks are a form of artificial neural network that uses stochastic components to simulate various models and their probability distributions(Zhou; 2019). In these, numerous models are trained instead of a single model, and the overall output is projected depending on a specific level of aggregation of all the models. According to the research article(Jospin et al.; 2020), the fundamental rationale for this method is that it has been discovered that aggregating output from numerous average individual models provides better prediction than a single high-performance model. One of the advantages of employing these stochastic neural networks is that they offer an indication of the uncertainties connected with the process.

1.3 Research Question

RQ: "Can Bayesian Neural Networks provide a significant improvement in Deepfake detection over existing State of the Art Deepfake detection models?"

Sub RQ: Can the Precision and Recall of Deepfake detection models be improved by using credible intervals provided by Bayesian Neural Networks instead of point predictions provided by Artificial Neural Networks?

The remainder of the paper is organized in the following format. Section 2 discusses and analyzes existing methods for detecting Deepfakes. Sections 3 and 4 of the paper explain the strategy and approach, which include a full description of the dataset collection, pre-processing phases, and data mining techniques employed.Sections 5 and 6 of the paper offer a description of the project's implementation, assessment methodologies, and outcomes. Section 7 concludes the research by offering an overall overview as well as suggestions for additional research.

2 Related Work

2.1 Deepfake Detection using traditional Machine Learning

Matern et al. (2019) present a comparison between Logistic Regression and Multi-layer Perceptron (MLP) performances for Deepfake detection. The idea behind the study was to look for any missing features between the eye and teeth areas of the videos and use them for efficient detection. The videos for this study were collected from YouTube, and only the frames with the eyes and mouth open were extracted. With Logistic Regression, they obtained an AUC of 0.78 and with MLP, they obtained an AUC of 0.85.

Instead of deep learning, Kharbat et al. (2019) employs machine learning techniques. On the area of deepfake detection, a technique based on SVM classifiers and a HOG feature point descriptor achieved a spectacular result. They demonstrate that combining machine learning algorithms with deep-learning approaches can yield impressive results for the task of detecting deepfakes.

Face recognition algorithms such as VGG and Facenet neural network, according to Korshunov and Marcel (2019), do not perform effectively on deepfake videos and fail to discriminate between original and manipulated video with a 95% failure rate. Lip-sync-based algorithms also failed to detect a disparity between speech and lip movement. However, the image-based technique using the SVM classifier has been determined to have an error rate of 8.97 percent on deepfake films. As a result, it is determined that image-based techniques have a better percentage of accuracy in detecting deepfake videos than the other methods.

2.2 Deep Learning based Deepfake detection

Stanciu and Ionescu (2021) explored if a video's temporal characteristics may be utilized to improve the performance of currently existing deepfake detection methods. Instead of feeding the model a totally aligned face and just selecting specific facial portions, they evaluated if particular facial sections provide more information regarding the validity of the video. The LSTM block was a two-layer, 256-layer LSTM that created a temporal descriptor for the sequence and was used for classification. The model provided a 13.46 percent increase in AUC for the CelebDF dataset, while it provided an almost faultless 99.95 percent AUC for the FF++ dataset.

Guera and Delp (2018) used a hybrid of CNN and LSTM to construct a temporal-aware system for Deepfake detection. This study expanded on an earlier notion of employing Long-term Recurrent Convolutional Networks (LRCN) to establish an end-to-end model for Visual Recognition. CNN was used to extract features, and the output was routed to LSTM, which was utilized to do temporal sequence analysis on the data. The raw data set was compiled from several sources, and the system was evaluated for correctness over numerous sub-sequences of video frames. For modified videos, an accuracy of 97 percent was achieved, demonstrating the efficacy of neural networks for Deepfake detection.

An proposition was made stating that detectors utilizing deep learning algorithms blindly do not perform well and that generative models perform better(Ciftci and Demir; 2019). They claimed that biological signals in images are neither spatially nor temporally conserved in manipulated video sequences, which has been a common argument in all research, and that this would be the core premise of detection of Deepfake videos. They used CNN to increase classification accuracy on a specialised data set, achieving 91 percent accuracy in Deepfake detection.

Using self-supervised decoupling networks, Zhang et al. (2021) suggested a method for detecting false videos and images even when they are compressed or of poor quality, particularly in social media (SSDN). Authenticity and compression features are employed to train the networks, and feature decoupling is accomplished via a self-supervised technique. The findings reveal that the suggested technique outperforms state-of-the-art deepfake detection methods for compression. SSDN achieves 91.80 percent accuracy in a Low-Quality setting, which is 1.4 percent higher than F3-net (state-of-the-art approaches). This approach is exceptional and is being examined for review since it considers compression and authenticity, which other methods do not, with substantially superior accuracy.

Suratkar et al. (2020) employed CNN architectures based on Transfer-Learning to increase the generalizability of Deepfake Detection. A method that use a CNN to capture attributes from each frame of a video clip in order to train a binary classifier that can efficiently discriminate between genuine and fake videos. The method is tested on a

large number of deepfake videos gathered from various datasets. CNN Base models included Xception, Inception v3, MobileNet, ResNet50, and others. Extra layers and hyperparameters such as dropout, decay rate, and so on were built on top of that for training and evaluated on the rescaled dataset. Except for the ResNets model, every other model performed well, with Inception v3 outperforming them all.

2.3 Deepfake detection on DFDC datasets

Mittal et al. (2020) attempted to construct an analogous model to improve detection accuracy by using the notion of Siamese Network Architecture(Hadsell et al.; 2006). They presented the hypothesis that previous systems only aim to target a single modality or a single feature, and that several modalities may be utilized to search for flaws in videos for effective Deepfake detection. In the study, they aimed to leverage the relationship between auditory and visual modalities in a video for the same purpose. To showcase this functionality, two Siamese Networks-based architectures, one for sound and another for visuals, were constructed. Although they reached an accuracy of 84.4 % while using model on the DFDC data set, they did not achieve great results on random in-the-wild videos.

Deepfake detection was accomplished at the University of California(Agarwal et al.; 2020) by utilizing CNN to identify discrepancies in images at pixel level caused by face warping. In order to conduct the research, a Siamese network was trained to identify discrepancies in camera information such as ISO, focal length, aperture size, and so on. The first network aims to identify any face modifications, while the second network attempts to detect low-level steganographic traits that indicate whether or not the image is consistent. The outcome is predicted using a mixture of the outputs of these two networks. They attained an accuracy of 82.4% for the Deepfake Detection Challenge (DFDC) data set after testing the algorithm on numerous data sets.

In an exploratory research, Hashmi et al. (2020) employed the Conv-LSTM Hybrid Framework for deepfake detection utilizing microscopic-typo assessment of video frames. CNN feature extractors employed a transfer learning technique, which begins with a pre-trained ResNet model and then passes it on to the LSTM network, which was chosen over GRU owing to memory restrictions. The training on the DFDC training set a took almost seven days.The model proved to be a heavyweight in terms of computing complexity. According to the evaluation results,the suggested model and network architecture set a new standard for detecting visual counterfeits,and the best categorization qualities were eyes, brows, head movement, and mouth movement.

Qi et al. (2020) present an hypothesizes that normal heartbeat rhythms found in real face videos would be disrupted or even completely broken in a DeepFake video by monitoring the minuscule periodic changes in skin color caused by blood pumping through the face(Yu et al.; 2019), making it a highly promising indicator for DeepFake detection. Using this concept, a DeepFake detection approach is suggested, which exposes DeepFakes by tracking pulse rhythms. They propose a motion-magnified spatial-temporal representation (MMSTR) to characterize the sequential signals of face videos, which provides significant discriminative characteristics for high accuracy DeepFake detection. Using this method, an accuracy of 64% was obtained on the DFDC preview dataset.

3 Methodology

Because this project falls within the topic of data mining and data science, one of the most often used methodologies in this category, Knowledge Discovery in Databases (KDD), was chosen for this project. The rationale for selecting KDD over Cross-Industry Standard Data Mining (CRISP DM) is because CRISP DM normally ends with project deployment since it is meant to suit business applications, however with KDD this is not a mandatory step to finish which meshes perfectly for this project. Figure 2 gives information on the practices that would be undertaken in the KDD approach.

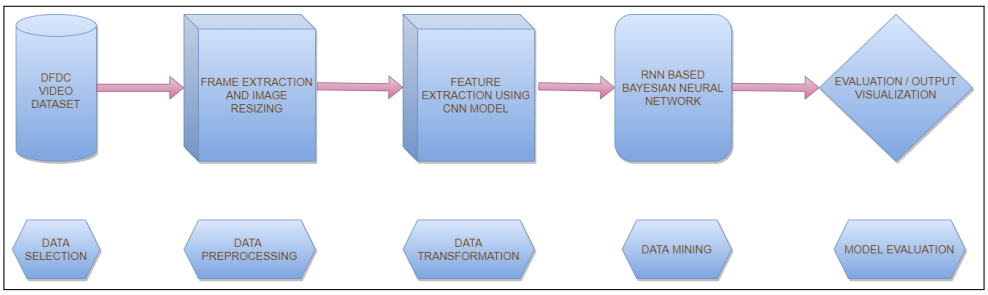


Figure 1: Methodology

3.1 Data Selection

In this research, the Deepfake Detection Challenge (DFDC) data set, which is available on kaggle, would be utilised to train the Bayesian Neural Network model. The Kaggle DFDC Dataset[6] consists of 400 Training and 400 Testing videos. The data collection was put together by Facebook, Microsoft, and AWS and is one of the most adaptable data sets for Deepfake detection available. Larger variations of the DFDC Dataset[7], referred to as DFDC Preview dataset (5K Videos) and DFDC Full dataset (124K Videos) have been made publicly available from Amazon S3 bucket.

Dolhansky et al. (2019) provide a brief description of the Deepfakes Detection Challenge (DFDC) Preview dataset, which consists of 5K videos with two facial modification algorithms applied to generate Deepfakes. For dataset creation, a small group of 66 people were picked from a pool of crowdsourced actors and split into 2 groups: training and testing. This was done to prevent face swaps between sets. To produce face swaps, two strategies were chosen (noted as methods A and B in the dataset), with the goal of portraying the true adversarial space of facial alteration. The videos contain a variety of lighting situations and head angles, and participants were free to record their videos with whichever background they wanted, resulting in aesthetically varying backgrounds. One important difference between this dataset and others is that actors have decided to participate in the construction of the dataset, which uses and changes their appearance.

[6] `https://www.kaggle.com/c/deepfake-detection-challenge/data`
[7] `https://ai.facebook.com/datasets/dfdc/`

3.2 Data Pre-processing

The selected dataset is then evaluated for class imbalance, resizing, normalization, data quality, and image dimensionality during the pre-processing stage. Exploratory Data Analysis is performed on the data for better model selection. For the videos in the DFDC dataset, each video is split into its respective frame and then resized to a pre-configured size.

3.3 Data Transformation

Data transformation is the process of converting data from one structure to another. It is important in data management and integration operations.
Feature extraction is done on the pre-processed output to generate patterns that will be used by the sequence processor. In this research, the Inception v3 model was adopted for feature extraction.

3.4 Data Mining

Finding connections, trends, and anomalies in a huge dataset and forecasting outcomes provides a wide understanding of data mining. The study proposes a design that is similar to the one used by Guera and Delp (2018). A combination of CNN and LSTM would be employed in the same way that it was in previous research. The above model will be used to develop a Bayesian Neural Network, with the weights and biases generated from a distribution rather than point values, as part of this research.

3.5 Model Evaluation

The distribution of positive and negative instances contained in the test set substantially influences the metrics produced by all relevant datasets addressing the task of Deepfake detection. As a result, it's difficult to quantify how any of the approaches tested on those datasets will fare in real-world production traffic. This is especially true when looking at the impact of false positives (FP) and the actions that come with them.

In their research, Dolhansky et al. (2019) propose an evaluation approach for Deepfake detection. A weighted precision for a deepfakes dataset can be considered as a very close estimate of the precision that would be computed by evaluating on an organic traffic dataset. Weighted Precision (wP) and standard recall (R) are defined as

$$wP = \frac{TP}{(TP + \alpha FP)}, \qquad R = \frac{TP}{(TP + FN)}$$

Where, TP indicates True Positive, FP indicates False Positive and FN indicates False Negative. Because there are few true negatives in the DFDC Preview dataset, any false positives will produce substantial variances in the wP metric and hence the wP results are computed with value of $\alpha = 100$.
Additionally, since false positives are strongly weighted, wP comes out to be a very small number. Hence, log(wP) is chosen for evaluation purposes, with values closer to zero indicating good performance.

4 Design Specification

The research proposes a design similar to the one implemented by Guera and Delp (2018). Figure 2 presents an overview of the model design.

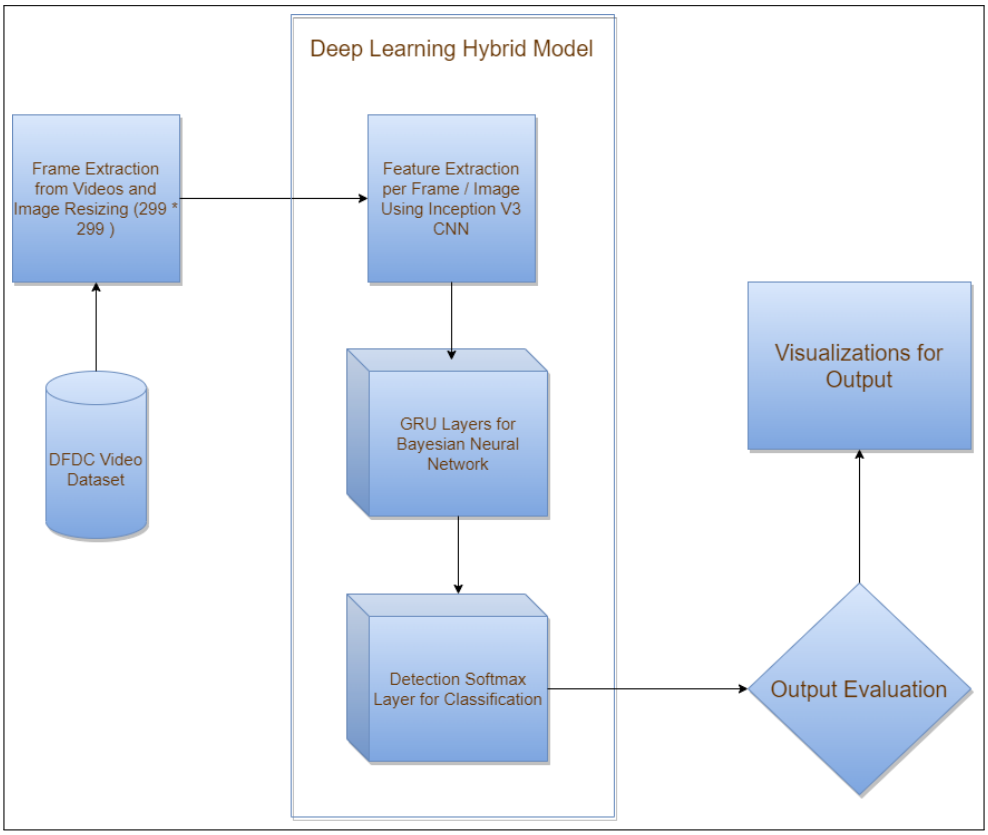


Figure 2: Design Overview

1. After reading the training videos from the dataset, each video is split into its respective frames and the resultant frame images are resized according to a specified configuration value in the code.

2. The resized images are then passed to the Inception V3 CNN Model[8] for feature extraction, which generates a 8x8x2048 size vector output for N (N=80 for this research) frames. The output from feature extraction is then concatenated into a single dimensionality vector which would be used as input to GRU Layer.

3. The GRU layer would be utilized for sequence processing. A 512-wide GRU layer, with a 0.3 dropout value, is used for temporal analysis of frames. L1 regularizer has been added to the GRU input layer to handle overfitting issues during model training.

4. The GRU layer is then connected to a 256-wide dense flipout layer which simulates the Bayesian paradigm of the Bayesian Neural Network. This layer configures the prior distribution for the weights and bias to be a multivariate normal distribution, the default value as recommended by keras guide[9]. Additionally, the benefit of

[8]Inception V3 Architecture: `https://cloud.google.com/tpu/docs/inception-v3-advanced`
[9]`https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseFlipout`

choosing a normal prior is that this results in a normal posterior distribution as the normal distribution is its own conjugate.

5. The final layer is a dense flipout layer with softmax activation which computes categorical probabilities for the input video sequence. Based on these probabilities, the lower and upper values of the 95 percent credible interval are derived. Based on these credible interval values, the input videos would then be classified as deepfake or real videos.

5 Implementation

This section goes through the processing of frames, feature extraction, and sequence processing in detail.

5.1 Preliminary EDA

Data analytics, it is claimed, is all about comprehending the data. Exploration of the videos in the dataset is the first step in implementation. The primary step was to load the dataset into Python and explore it, as well as to determine the type of files available in the dataset. The DFDC dataset includes video files as well as a metadata JSON file with the names of fake and actual videos labeled appropriately. All of the source code and model building operations for this study were carried out on the DFDC demo dataset, which comprises of 400 videos. The final model was then scaled up for the DFDC preview dataset, which contains 5000 videos. The remainder of this paper goes into the implementation specifics for both datasets.

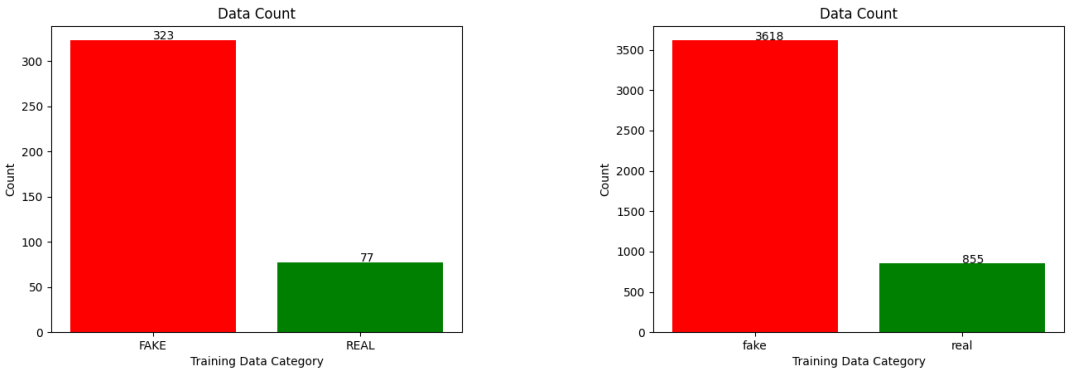
Usingn Python, the metadata JSON file is read and the count of each video category is plotted. Figure 2 shows the Real and Fake video counts for both the demo and preview datasets. It can be observed that the count of actual videos is considerably too low in comparison to the count of false videos in each of these datasets. The real-to-fake video ratio is nearly one-to-four, which may potentially induce a bias in our model. A decision to not handle this imbalance was made as upsampling would be quite a resource intensive process, while downsampling would result in possible loss of important data points. Moreover, it was observed that this balance did not affect the performance of the model for the test dataset, as explained in the evaluation section.

Following that, checks were performed to see whether any data was missing from the dataset, such as a video filename that was not contained in the metadata JSON. Because the JSON files for both datasets had all of the essential information with no gaps, no extra handling was required.

5.2 Frame Extraction and Resizing

The Decord Python library is used to explore videos and extract frames from these. Decord was chosen over opencv because it was proven to be twice as fast in certain cases[10]. Decord was used to read all of the videos in the training dataset and then separate them into individual frames. The frames were then turned into a numpy array, with each video consisting of a numpy array of N frames, and the output was processed

[10] `https://cv.gluon.ai/build/examples_action_recognition/decord_loader.html`



(a) 400 model EDA

(b) 5K model EDA

Figure 3: Category Count for dataset

for feature extraction.
Although each video has 300 frames, only a limited amount were chosen for two reasons: first, to minimize computing costs, and second, to reduce image duplication. To align with the model presented by Guera and Delp (2018) in their research, 80 frames per video were chosen from a population of 300 frames for this research. If all of the frames from each video are processed , the outcome is a slew of repeated images as there isn't much movement in the video in very short intervals. Since the demo dataset contains 400 movies, each of which is around 10 seconds long, there will be approximately 32000 extracted frames for processing following frame extraction. In the DFDC Preview dataset, the count will be considerably higher.

5.3 Feature Extraction

The most important and distinctive step undertaken in this project is filtering out irrelevant stuff and extracting essential features. It is a vital step since it removes the undesirable background from the frames and so essentially reduces the image size. The Inception v3 model is used for feature extraction.
On the ImageNet dataset, Inception v3 has been proven to achieve higher than 78.1 percent accuracy. Convolutions, average pooling, max pooling, concats, dropouts, and fully linked layers are some of the symmetric and asymmetric building elements used in the model (Szegedy et al.; 2016). Batchnorm is applied to activation inputs and is utilized extensively throughout the model. Softmax is used to determine loss. Figure 4 describes the Inception v3 architecture[11].

For this research, the inception v3 model is implemented using keras module[12]. The parameters are configured as shown below.

- 'include_top' : Set to 'False' since the top layer is not intended to be the last layer. The output from Inception v3 would be used for sequence processing.
- 'weights' : Set to 'imagenet' to use pre-trained weights instead of recomputation of weights.

[11] `https://cloud.google.com/tpu/docs/inception-v3-advanced`
[12] `https://keras.io/api/applications/inceptionv3/`

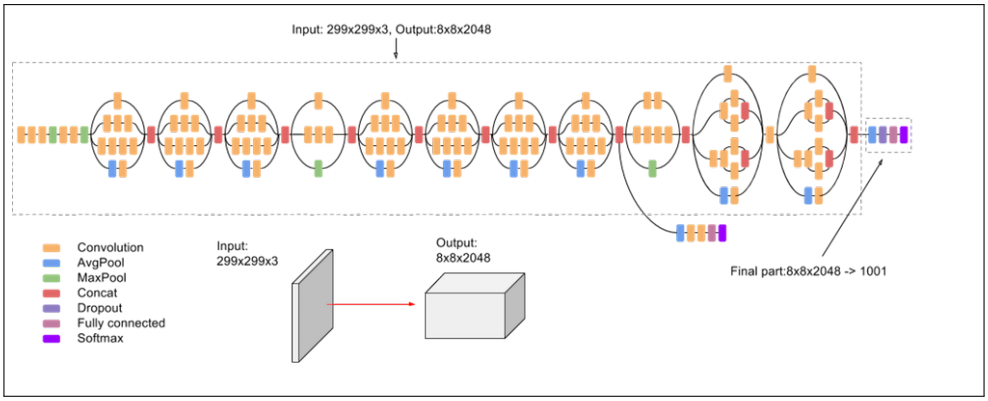


Figure 4: Inception v3 Architecture (adapted by google based on a model proposed by Szegedy, et. al(2016))

- 'input_tensor' : Set to 'None' as the inputs would only be used for Feature extraction purposes and will not be shared anywhere across the model.
- 'input_shape' : Set to (299,299,3) as suggested in the keras guide[13].
- 'pooling' : Set to 'max' to use global max pooling method for resize. Max pooling was observed to give better results than avg pooling in the research with a possible reason being that max pooling selects the brightest pixel during convolution which implicitly works better in images with a stark contrast between objects and their background.

Figure 5 shows the summary of the Feature Extraction model implemented in the system.

```
Model: "FeatureExtractor"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 299, 299, 3)]     0
_________________________________________________________________
tf.math.truediv (TFOpLambda) (None, 299, 299, 3)       0
_________________________________________________________________
tf.math.subtract (TFOpLambda (None, 299, 299, 3)       0
_________________________________________________________________
inception_v3 (Functional)    (None, 2048)              21802784
=================================================================
Total params: 21,802,784
Trainable params: 21,768,352
Non-trainable params: 34,432
_________________________________________________________________
```

Figure 5: Feature Extraction Model

[13] https://keras.io/api/applications/inceptionv3/

5.4 Sequence Processing

In this research, GRU was used to implement Sequence Processing. GRU is a kind of RNN that can maintain long-term reliability and accuracy. GRUs have been shown to be capable of supplementing CNN's feature extraction capabilities when used in layered settings. While CNNs can extract the key bits from them, GRUs can remember trends preferentially for a long period.Thus, when used for image classification, the RNN-CNN layered structure can potentially improvise over the traditional CNN classifier.
Figure 6 shows the summary of the Sequence Processing model implemented in the system.

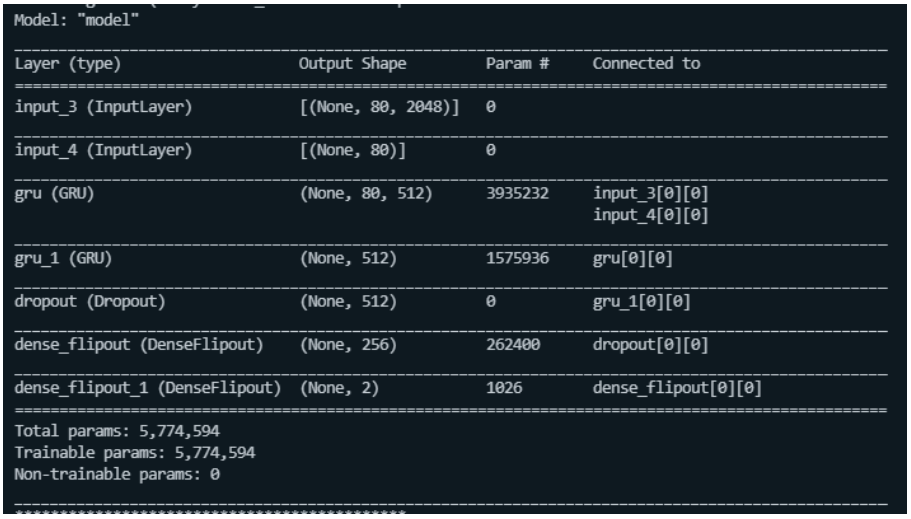


Figure 6: Sequence Processing Model

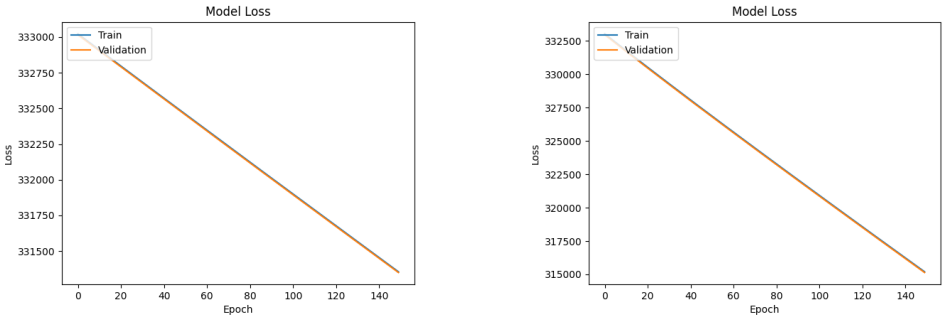
The GRU input layer is a 2048-wide layer which accepts a batch of N=80 frames per epoch. This layer performs temporal sequence analysis on the 80 frames to detect relationships between the feature extracted frames. L1 regularization is applied on this layer with lamba being 0.001 to avoid overfitting which was observed in the initial model building phase.
This is followed by another 512-wide GRU layer to create a more complex feature representation of the input. This will enable the model to extract more information. This is followed by a dropout layer with a rate of 0.3 to randomly shut off 30 percent neurons to further prevent overfitting.
The 256-wide dense flipout layer acts as the final layer which connects the GRU model to the output layer and handles the Bayesian paradigm by setting a prior distribution for the weights and bias instead of using singular point values.
The output layer is a dense flipout layer too, with a softmax activation function.

5.5 Model Performance

For model training, training data is split into 80 percent training data and 20 percent testing data, and training data was further divided into 80 percent training and 20 percent validation data. The Adam optimizer was used for optimization, with the learning rate set at 0.0001. The model was trained for 150 epochs, with early stopping enabled on

validation loss. The parameters for early stopping were configured as suggested in the keras official guide [14].

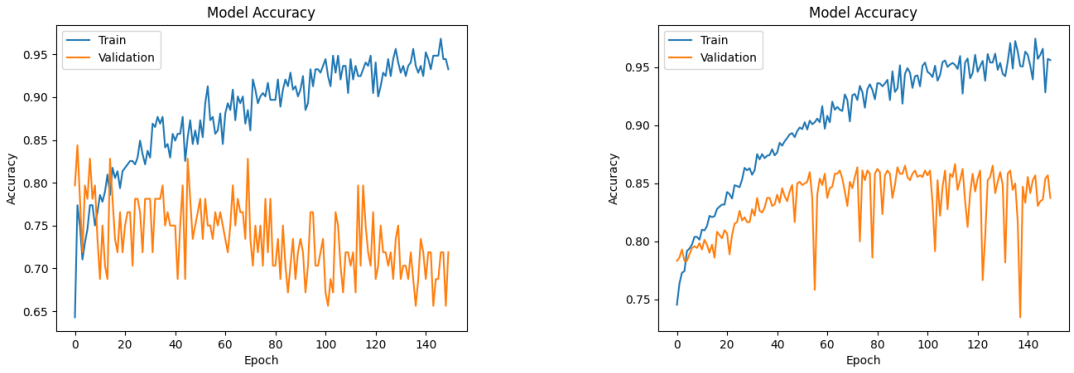
Figure 7 and Figure 8 provide details about the model loss and model accuracy values for both the DFDC demo and DFDC preview datasets.



(a) 400 model Loss (b) 5K model Loss

Figure 7: Model Loss

It can be seen in Figure 7 that these models show a linear reduction in loss. i.e. for a duration of 150 epochs, a loss of only around 1 percent is observed. However, it accuracy of these models for the same number of epochs is more than 90 percent for the training set of both DFDC datasets, while the validation accuracy sees a slight drop for both of these models.
A possible reason for the convergence issue of the loss might be that the number of epochs is significantly less and hence only a fraction of the entire loss function is observed in the Figure 7.



(a) 400 model Accuracy (b) 5K model Accuracy

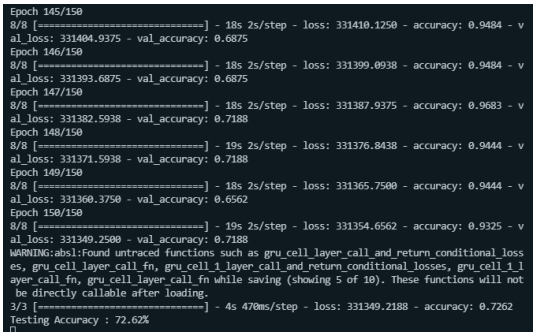
Figure 8: Model Accuracy

Although it can be seen in Figure 8 that the training accuracy is showing an upward trend as the number of epochs rises, the reason for training only 150 epochs is to avoid further reduction in the validation accuracy.

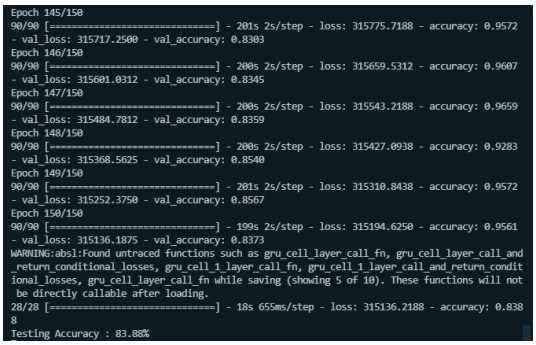
Figure 9 provides details about the model performance for both the datasets after 150 epochs. For DFDC demo dataset, the training accuracy was around 93 percent,

[14] `https://keras.io/api/callbacks/early_stopping/`

the validation accuracy was around 72 percent and the testing accuracy was around 73 percent. For the DFDC preview dataset, the training accuracy was around 96 percent, the validation accuracy was around 84 percent and the testing accuracy was around 84 percent.



(a) 400 model Perf



(b) 5K model Perf

Figure 9: Model Performance

The research aimed to initially create an effective model for the DFDC demo dataset which would then be scaled up for the DFDC preview dataset. Hyperparameter optimization of the DFDC demo model was attempted using keras autotuner, which could not be performed due to memory constraints and hence, manual hyperparameter tuning was performed resulting in the DFDC model referenced in Figure 6. It can be seen that the DFDC demo model didn't show any possible overfitting, and hence, the same model was used for DFDC preview dataset.

6 Evaluation

6.1 Data Category Details

Figure 6 shows the true count of the categories in the DFDC demo and the DFDC preview datasets.

(a) 400 model CM

(b) 5K model CM

Figure 10: Test Data Categories

It can be seen that the categories are balanced in the DFDC demo dataset, while the categories show an imbalance in the DFDC preview dataset. The model performance, especially evaluation on Recall values, need to be considered for the DFDC preview dataset. A higher Recall value for 'fake' category might not necessarily indicate good performance for the DFDC dataset as around 66 percent of the total amount of videos are 'fake'.

6.2 Model Evaluation

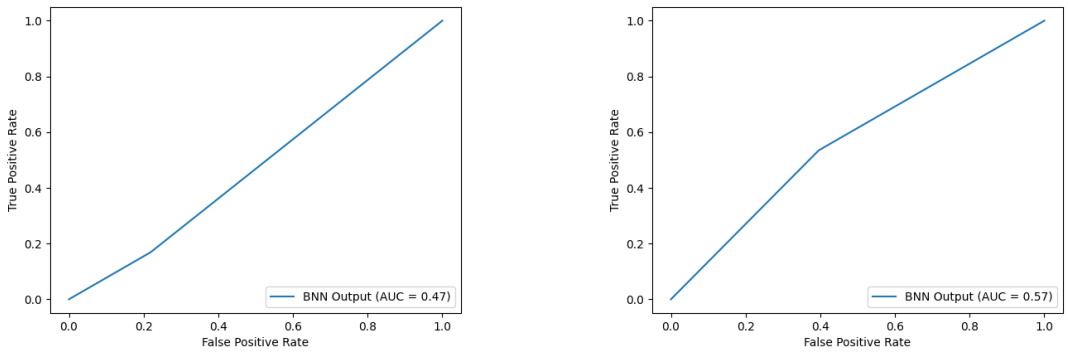
Table 1 shows the confusion matrix for the DFDC demo and DFDC preview datasets where '0' indicates 'Fake' category and '1' indicates 'Real' category. A noteworthy point here is that only predictions with values '0' and '1' are shown in the confusion matrix. 'Unsure' predictions, with value '2', are omitted from the confusion matrix.

Table 1: Confusion Matrix

	400 Model		**5K Model**	
	0 (Predicted)	**1 (Predicted)**	**0 (Predicted)**	**1 (Predicted)**
0 (Actual)	127	37	291	190
1 (Actual)	133	26	122	143

For the DFDC Demo dataset, of the 200 fake videos, 127 videos were classified correctly, 37 videos were incorrectly classified. Of the 200 real videos, only 26 videos were classified correctly, while 133 videos were incorrectly classified. The remainder of the videos were classified as 'Unsure'.
For the DFDC Preview dataset, of the 501 fake videos, 291 videos were classified correctly, 190 videos were incorrectly classified. Of the 276 real videos, only 143 videos were classified correctly, while 122 videos were incorrectly classified. The remainder of the videos were classified as 'Unsure'.

Figure 11 shows the ROC Curve for the DFDC demo and DFDC preview datasets.



(a) 400 model ROC

(b) 5K model ROC

Figure 11: ROC

For the DFDC Demo dataset, the Area Under Curve (AUC) is 0.47 which indicates that the model has poor accuracy and isn't performing well for the test data.

For the DFDC Preview dataset, the AUC is slightly better at value 0.57, which indicates that the model is performing better after being trained on a larger dataset. This trend indicates that on the DFDC full dataset, which contains 124K videos, the model is likely to perform even better.

Table 2 shows the evalution metrics for both the models.

Table 2: Evaluation Metrics

	400 Model Metrics		**5K Model Metrics**	
	0 (Fake)	**1 (Real)**	**0 (Fake)**	**1 (Real)**
Precision	0.49	0.41	0.70	0.43
Recall	0.77	0.16	0.6	0.54
F1-Score	0.6	0.23	0.65	0.48
Accuracy	0.47		0.58	
log-wP	-3.41		-4.19	

For the DFDC demo dataset, it can be observed that the Precision and Recall values for the 'Fake' category are 0.49 and 0.77 respectively, while the Precision and Recall values for the 'Real' category are 0.41 and 0.16 respectively. This indicates that the model is performing significantly better for Fake category while it is performing poorly for Real category.
For the DFDC preview dataset, it can be observed that the Precision and Recall values for the 'Fake' category are 0.70 and 0.60 respectively, while the Precision and Recall values for the 'Real' category are 0.43 and 0.54 respectively. This indicates that the model is performing slightly better for Fake category while it is giving average performance for Real category.
This trend indicates that the model is likely to give good Precision and Recall values when trained with a larger dataset.

Below is an output snippet for the credible intervals of a few sample predictions and their predicted categories. The prediction output value of '0' indicates the prediction as 'Fake', output value '1' indicates the prediction as 'Real',while an output value of '2' indicates the system is 'Unsure' of the category.

```
************************************
Predicting for 342 of 400 Test data.
Actual Class for Video : 0
CI for Fake (0) - [92.56, 43.32 ]
CI for Real (1) - [56.68, 7.44 ]
Prediction Based on Threshold : 2
************************************
Predicting for 343 of 400 Test data.
Actual Class for Video : 1
CI for Fake (0) - [91.09, 49.55 ]
CI for Real (1) - [50.45, 8.91 ]
Prediction Based on Threshold : 2
************************************
Predicting for 344 of 400 Test data.
Actual Class for Video : 1
CI for Fake (0) - [92.51, 51.24 ]
CI for Real (1) - [48.76, 7.49 ]
Prediction Based on Threshold : 0
************************************
Predicting for 345 of 400 Test data.
Actual Class for Video : 1
CI for Fake (0) - [99.72, 92.65 ]
CI for Real (1) - [7.35, 0.28 ]
Prediction Based on Threshold : 0
************************************
Predicting for 346 of 400 Test data.
Actual Class for Video : 0
CI for Fake (0) - [14.02, 1.05 ]
CI for Real (1) - [98.95, 85.98 ]
Prediction Based on Threshold : 1
************************************
Predicting for 347 of 400 Test data.
Actual Class for Video : 1
CI for Fake (0) - [37.08, 8.28 ]
CI for Real (1) - [91.72, 62.92 ]
Prediction Based on Threshold : 1
************************************
Predicting for 348 of 400 Test data.
Actual Class for Video : 0
CI for Fake (0) - [96.20, 65.71 ]
CI for Real (1) - [34.29, 3.80 ]
Prediction Based on Threshold : 0
************************************
```

400 model Inferencing output

```
************************************
Predicting for 628 of 777 Test data.
Actual Class for Video : 1
CI for Fake (0) - [48.85, 13.60 ]
CI for Real (1) - [86.40, 51.15 ]
Prediction Based on Threshold : 1
************************************
Predicting for 629 of 777 Test data.
Actual Class for Video : 1
CI for Fake (0) - [58.72, 15.09 ]
CI for Real (1) - [84.91, 41.28]
Prediction Based on Threshold : 2
************************************
Predicting for 630 of 777 Test data.
Actual Class for Video : 0
CI for Fake (0) - [10.22, 1.11 ]
CI for Real (1) - [98.89, 89.78 ]
Prediction Based on Threshold : 1
************************************
Predicting for 631 of 777 Test data.
Actual Class for Video : 0
CI for Fake (0) - [64.79, 24.57]
CI for Real (1) - [75.43, 35.21 ]
Prediction Based on Threshold : 2
************************************
Predicting for 632 of 777 Test data.
Actual Class for Video : 0
CI for Fake (0) - [20.73, 1.64 ]
CI for Real (1) - [98.36, 79.27 ]
Prediction Based on Threshold : 1
************************************
Predicting for 633 of 777 Test data.
Actual Class for Video : 0
CI for Fake (0) - [99.76, 95.17 ]
CI for Real (1) - [4.83, 0.24 ]
Prediction Based on Threshold : 0
************************************
Predicting for 634 of 777 Test data.
Actual Class for Video : 0
CI for Fake (0) - [35.83, 1.84 ]
CI for Real (1) - [98.16, 64.17]
Prediction Based on Threshold : 1
************************************
```

5K model Inferencing output

The snippet shows credible interval values for both 'Fake' and 'Real' categories, which are derived based on the central 95 percent (2.5 - 97.5%) quantile values of the prediction from the Bayesian Neural Network.
For the system to confidently classify any given video into a specific category, 2 conditions must be satisfied.

- For a configured threshold value, the upper limit of the credible interval is over the threshold value.
- The credible intervals of the two categories should be mutually exclusive

If both the conditions are not met, the model will classify the output category as unsure.

For this research, the threshold value has been configured as '60'. It was observed that the model gave best performance at this value, but can be subjected for a large dataset.

In the snippet, it can be observed for both the models,in cases where the credible intervals are overlapping, the system is predicting a value of '2' indicating it is unsure of category for that video. Predictions '343' and '629' of DFDC Demo and DFDC Preview datasets demonstrate the same respectively.

6.3 Discussion

Following the evaluation of the model using the metrics recommended in the research article(Dolhansky et al.; 2019), Table 3 compares the Model performance of the DFDC preview dataset to the current performances presented in the research.

Table 3: Model Performance

Method	**Precision**	**Recall**	**log-wP**
TamperNet	0.833	0.033	-3.044
XceptionNet (Face)	0.930	0.084	-2.140
XceptionNet (Full)	0.784	0.268	-3.352
BNN Model(Research Model)	0.7	0.6	-4.19

It can be observed that the BNN Model lags behind in Precision as compared to other models. While it shows a good Recall value as compared to other models, this is due to the below scenarios.

1. The biased nature of data where 66 percent of the data is 'fake' and
2. The model has a high number of True Positives and False Positives, which indicates that most of predictions were classified as 'Fake'.

Moreover, the value for log of weighted precision is desired to be close to zero. For the research model, we have the largest value for log-wP, which indicates that the model has demonstrated inferior performance as compared to existing State of the Art Deepfake detection models.

Additionally, as per the results of the evaluation, the DFDC preview dataset performed significantly better than the DFDC demo dataset on the same model. As a result, it can be anticipated that given a dataset with more training data, such as the DFDC complete dataset, the model will perform even better. This Hypothesis can undoubtedly be verified on a system capable of processing 124K videos.

Lastly, the confusion matrix shows a larger count of False Positives and False Negatives for the DFDC preview dataset model. The ideal scenario was to minimize these quantities as a part of a secondary component of this research, but, it did not come to fruition. The BNN model still confidently predicted the videos in wrong categories, possibly due to bias of data. This assumption can be validated with training the model after handling the imbalance in the data.

7 Conclusion and Future Work

The research's primary objective was to evaluate the performance of Bayesian Neural networks for deepfake identification when compared to other state-of-the-art models. The

DFDC Preview dataset was used to train the Bayesian Neural Network, which was designed as a hybrid deep learning model that combined the Inception v3 Model for feature extraction with an RNN-based Sequence processor for temporal analysis. The model was tested against two existing models published in the DFDC preview data set publication, and the Bayesian Neural Network was shown to perform poorly when compared to the 'TamperNet' and 'XceptionNet' models. Despite achieving a high training accuracy of approximately 96 percent, the bayesian model could only attain a test dataset accuracy of 58 percent. Further research is necessary to improve on this.

The research model could not be trained efficiently owing to system restrictions. The model may be modified for greater performance as part of subsequent study, and the DFDC complete dataset, which contains 124K movies, can be utilized to train the model. To obtain a highly optimized model, additional tuning of hyperparameters can be accomplished using keras tuner.

Furthermore, additional work can be directed into the treatment of videos flagged as unsure by the model.

References

Agarwal, S., Farid, H., El-Gaaly, T. and Lim, S.-N. (2020). Detecting deep-fake videos from appearance and behavior, *2020 IEEE International Workshop on Information Forensics and Security (WIFS)* pp. 1–6.

Ciftci, U. A. and Demir, I. (2019). Fakecatcher: Detection of synthetic portrait videos using biological signals, *CoRR* **abs/1901.02212**.

Dolhansky, B., Howes, R., Pflaum, B., Baram, N. and Canton-Ferrer, C. (2019). The deepfake detection challenge (DFDC) preview dataset, *CoRR* **abs/1910.08854**.

Guera, D. and Delp, E. J. (2018). Deepfake video detection using recurrent neural networks, *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* **46**(1): 1–6.

Hadsell, R., Chopra, S. and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* **2**: 1735–1742.

Hashmi, M. F., Ashish, B. K. K., Keskar, A. G., Bokde, N. D., Yoon, J. H. and Geem, Z. W. (2020). An exploratory analysis on visual counterfeits using conv-lstm hybrid architectures, *IEEE Access* **8**: 101293–101308.

Jospin, L. V., Buntine, W. L., Boussaid, F., Laga, H. and Bennamoun, M. (2020). Hands-on bayesian neural networks - a tutorial for deep learning users, *ArXiv* **abs/2007.06823**.

Kharbat, F. F., Elamsy, T., Mahmoud, A. and Abdullah, R. (2019). Image feature detectors for deepfake video detection, *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)* pp. 1–4.

Korshunov, P. and Marcel, S. (2019). Vulnerability assessment and detection of deepfake videos, *Conference: 2019 International Conference on Biometrics (ICB)* pp. 1–6.

Matern, F., Riess, C. and Stamminger, M. (2019). Exploiting visual artifacts to expose deepfakes and face manipulations, *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)* pp. 83–92.

Mirsky, Y. and Lee, W. (2020). The creation and detection of deepfakes: A survey, *Association for Computing Machinery* **54**(1).

Mittal, T., Bhattacharya, U., Chandra, R., Bera, A. and Manocha, D. (2020). Emotions don't lie: An audio-visual deepfake detection method using affective cues, *Proceedings of the 28th ACM International Conference on Multimedia,* .

Qi, H., Guo, Q., Juefei-Xu, F., Xie, X., Ma, L., Feng, W., Liu, Y. and Zhao, J. (2020). Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms, *Proceedings of the 28th ACM International Conference on Multimedia* p. 4318–4327.

Stanciu, D.-C. and Ionescu, B. (2021). Deepfake video detection with facial features and long-short term memory deep networks, *2021 International Symposium on Signals, Circuits and Systems (ISSCS)* pp. 1–4.

Suratkar, S., Johnson, E., Variyambat, K., Panchal, M. and Kazi, F. (2020). Employing transfer-learning based cnn architectures to enhance the generalizability of deepfake detection, *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* pp. 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016). Rethinking the inception architecture for computer vision, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 2818–2826.

Yu, Z., Peng, W., Li, X., Hong, X. and Zhao, G. (2019). Remote heart rate measurement from highly compressed facial videos: An end-to-end deep learning solution with video enhancement, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* pp. 151–160.

Zhang, J., Ni, J. and Xie, H. (2021). Deepfake videos detection using self-supervised decoupling network, *2021 IEEE International Conference on Multimedia and Expo (ICME)* pp. 1–6.

Zhou, Z.-H. (2019). Ensemble methods: foundations and algorithms, *Chapman and Hall/CRC* **7**.