**Table of Contents**

[Deep Fake Video Detection Using InceptionResnet-V2 3](#_Toc151330623)

[1. Problem Introduction 3](#_Toc151330624)

[3. Dataset Used 6](#_Toc151330625)

[3.1 Why DFDC Dataset 6](#_Toc151330626)

[4. Implementation Approaches 7](#_Toc151330627)

[5. Evaluation Metrics 7](#_Toc151330628)

[6. Applied Algorithms 7](#_Toc151330629)

[6.1 Convolutional Neural Network (CNN) 7](#_Toc151330630)

[7. Flow of Applying CNN on DFDC Dataset 8](#_Toc151330631)

[7.1 Loading the Kaggle Dataset 9](#_Toc151330632)

[7.2 Processing the Dataset 9](#_Toc151330633)

[7.3 Applying Image Data Augmentation 10](#_Toc151330634)

[7.4 Applying CNN to Dataset 10](#_Toc151330635)

[7.5 Fitting Model to Dataset 11](#_Toc151330636)

[7.6 Model Result Visualizations 12](#_Toc151330637)

[8. Technology Review 13](#_Toc151330638)

[8.1 Deepfake Video Creation and Detection 13](#_Toc151330639)

[8.2 Key Features: 16](#_Toc151330640)

[9. Conclusion 16](#_Toc151330641)

[10. References 17](#_Toc151330642)

**List of Figures**

[**Figure 1**: CNN Architecture ( Alzubaidi, L., Zhang, J., Humaidi) 3](#_Toc151330656)

[**Figure 2**: A Sample Diagram of CNN used for Deepfake Detection 8](#_Toc151330657)

[**Figure 3:** Importing Kaggle Dataset to Google Colab 9](#_Toc151330658)

[**Figure 4**: Applying Image Data Augmentation using Keras 10](#_Toc151330659)

[**Figure 5**: Visualization of the Neural network model 11](#_Toc151330660)

[**Figure 6**: Accuracy and Loss Graph of CNN 13](#_Toc151330661)

[**Figure 7**: Deepfake Generation using GAN 14](#_Toc151330662)

[**Figure 8**: A Sample Diagram of CNN used for Deepfake Detection 15](#_Toc151330663)

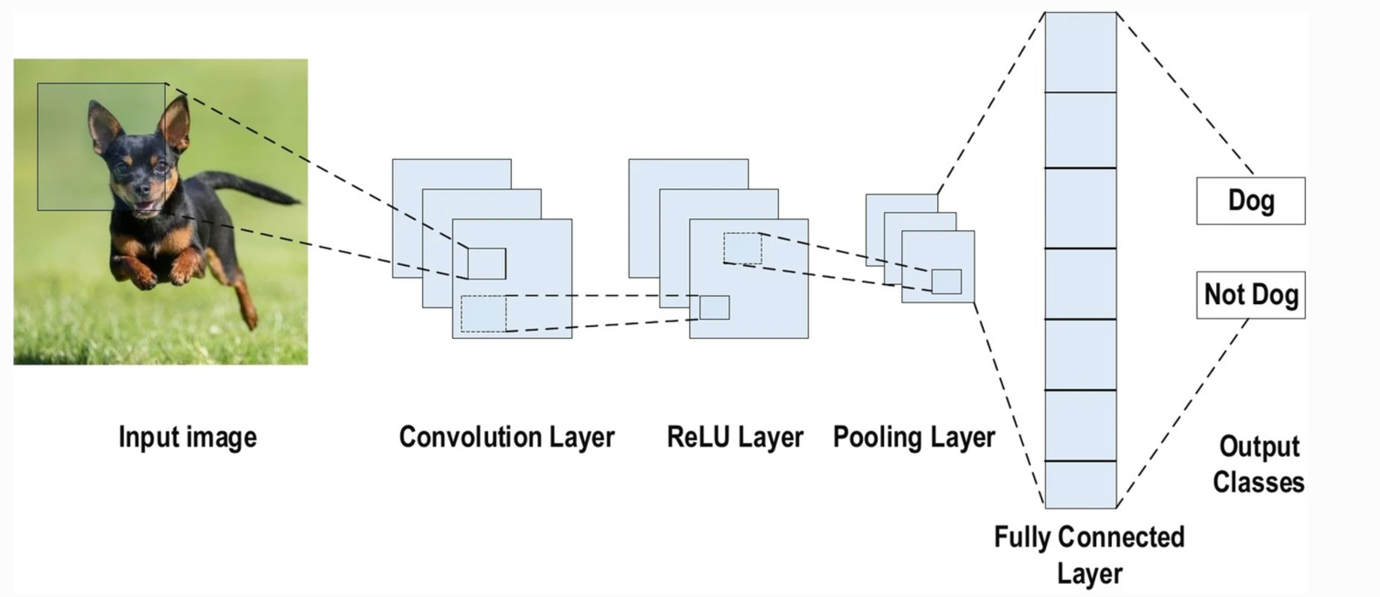
**List of Tables**

[**Table 1:** Configuration of Hyperparameters 11](#_Toc146813114)

# Deep Fake Video Detection Using InceptionResnet-V2

## 1. Problem Introduction

The field of artificial intelligence has shown cutting edge performances in various fields in recent years. Benchmark performances have been shown by artificial intelligence algorithms in various fields such as image processing, text, audio, videos, etc. In image processing another significant area of applying artificial intelligence algorithms is the detection of deep fakes from videos.   
Deep fake technology has been emerged in recent times that applies affairs of another person to a video which can be manipulated for unfair purposes. This technology can harm persons’ lives by manipulating his face on any false video. For the detection of deep fakes from videos these results have applied the applications of deep learning algorithms which is a branch of artificial intelligence such as convolutional neural networks (CNNs). These are usually fairly large; for instance, if the input were 300 \* 300-pixel color images, the network would require 300 \* 300 \* 3 (RGB) = 270000 input nodes [1][2]. If one was to use a network consisting of fully connected layers, the number of weights in the network would explode. A CNN model is divided in to three layers: (1) Convolution layer, (2) Pooling layer, and (3) Fully connected layer as shown in Figure 1.



**Figure 1**: CNN Architecture ( Alzubaidi, L., Zhang, J., Humaidi)

Here we are listing few approaches for deepfake video detection that are already proposed.

Güera, et al. proposes a temporal-aware pipeline to automatically detect deepfake videos. The system uses a convolutional neural network (CNN) to extract frame-level features. These features are then used to train a recurrent neural network (RNN) that learns to classify if a video has been subject to manipulation or not. The paper presents accuracy metrics, such as precision, recall, and F1-score, to evaluate the RNN model’s performance and understanding the trade-offs between false positives and false negatives is crucial [3].

*Key Findings:* The paper Proposes a temporal-aware pipeline for deepfake detection, utilizes a CNN to extract frame-level features and employs an RNN for video-level classification.  
*Methodology:* Uses CNN for feature extraction and trains RNN for video classification.  
*Contribution:* Introduces a temporal-aware approach combining CNN and RNN for deepfake detection.

Next paper aims to solve this problem by proposing a model that analyses the frames of the videos using deep learning approach to detect inconsistencies in facial features, compression rate and discrepancies introduced in the videos while creating them. The model uses a convolutional neural network (VGG16 model) along with transfer learning to train the model that can catch these instilled errors in the deepfakes [4].

*Key Findings*: Proposes a model for deepfake detection based on frame analysis using VGG16 with transfer learning and identifies inconsistencies in facial features and compression rate.  
*Methodology*: Utilizes VGG16 with transfer learning for feature extraction. Focuses on analyzing frames for inconsistencies.   
*Contribution*: Introduces a deep learning approach to detect errors in facial features and compression rate.

Even though many existing deepfake detection approaches show promising results, the majority of them still suffer from a number of critical limitations. In general, poor generalization results have been obtained under unseen or new deepfake generation methods. Consequently, in this paper, the Authors propose a deepfake detection method called HCiT, which combines Convolutional Neural Network (CNN) with Vision Transformer (ViT). The HCiT hybrid architecture exploits the advantages of CNN to extract local information with the ViT's self-attention mechanism to improve the detection accuracy. In this hybrid architecture, the feature maps extracted from the CNN are feed into ViT model that determines whether a specific video is fake or real. The CViT model is trained using the binary cross entropy loss function. A mini-batch of 32 images are normalized using mean of [0.485 , 0.456 , 0.406 ] and standard deviation of [0.229 , 0.224 , 0.225 ]. The normalized face images are then augmented before being fed into the CViT model at each training iterations. Adam optimizer with a learning rate of 0.1e-3 and weight decay of 0.1e-6 is used for optimization. The model is trained for a total of 50 epochs. The learning rate decreases by a factor of 0.1 at each step size of 15 [5]. The authors designed and developed a Convolutional Vision Transformer (CViT) model to detect Deepfake videos using CNNs and Transformer. They called their model generalized because it can learn local and global features, has extensive data preprocessing, and uses a diverse dataset.

In the next paper, the authors trained a convolutional neural network (CNN) that demonstrates mentionable accuracy in detecting fake videos in low-resolution and short-time video data. State-of-the-art studies show that while success is achieved in detecting fake video on high-quality and long-term video equipment, the same performance is not observed with low and short-term solutions. The paper acknowledges some limitations of their model, such as the need for more data and more robust features. The paper also suggests some future work, such as improving the model’s generalization and robustness, incorporating temporal information, and exploring other deep learning techniques. Model predicts 94.93% accuracy in detecting fake videos for the DFDC dataset while the same is 93.2% for FaceForensics++ Dataset [6].

In recent research, the model was applied to the DFDC dataset containing 60 different real and fake clips. This study process goes through three stages. In the first stage, each video is converted into frames and the face in each frame is detected by pre-processing and then sliced ​​with the Haar-Cascade function; ResNet-50 is used in the feature extraction phase. As a feature extraction model, in the last step, the CNN classifier is used to determine whether the image is fake or real. In this method detection accuracy was achieved at 98% [7].

## 3. Dataset Used

Easy access of technology to anybody, it is easy to share and receive deepfake videos and images of people that have been shared online. A Deepfake is a type of Artificial Intelligence that uses machine learning at the base to create manipulated videos, images or sound of people that look realistic but in reality, they are fake. Deepfake technology can create a video of person doing or saying something that the person never really did or said. In modern time deepfake is a one of the serious issues and it is used frequently to manipulate faces of popular political leaders, Hollywood or Bollywood actors over porn images or videos. Also, deepfakes were used to create political tension between countries [8].

The dataset used in this report is a famous Kaggle competition dataset, known as deepfake detection challenge (DFDC). The dataset was used for the purpose to detect deepfakes in videos in 2020, and numerous benchmarks were attained as a result of application of various architectures. Although this competition is closed as of today, some data is still available for researchers to apply deepfake detection algorithms. More specifically, the dataset contains:

* A folder named “train sample videos”, comprising 401 training videos
* A folder named “test videos”, comprising of 400 testing videos
* The videos in both folders belong to 2 classes, i.e., real & fake
* A separate JSON file is also provided explicitly, which can be used to distinguish real & fake videos while extraction of frames [9].

For distinguishing real video frames from fake frames, appropriate approaches are applied after extracting the relevant information from the JSON file.

### 3.1 Why DFDC Dataset

The choice of DFDC dataset for deepfake detection may is based on its outstanding features in the field. DFDC is a well-known benchmark dataset specifically designed to test deepfake detection algorithms. Researchers and practitioners often choose established data sets to ensure that their models are tested on a range of robust samples.

The DFDC dataset contains both real and modified videos, making it suitable for training and evaluation of deepfake recognition. This is consistent with the project objective of developing a modification detection system. Furthermore, using a widely recognized dataset such as DFDC allows for better comparison with existing literature and facilitates the validation of developed models in real situations

## 4. Implementation Approaches

Overall, python programming language is used as a tool in this project. Python provides a wide range of pre-defined libraries and functions, which can be used for a variety of tasks such as machine learning, deep learning, data science, etc. Following python packages are used:

* NumPy – for performing computations on arrays
* Pandas – for data preprocessing, I/O, etc.
* TensorFlow – for deep learning
* Keras – a high-level API of TensorFlow used for accessing deep learning models such as CNN, LSTM, etc.
* Google Collaboratory, a utility provided by Google for accessing GPUs

## 5. Evaluation Metrics

For accessing the quality of deep learning model, classification evaluation metrics are used in this project, such as:

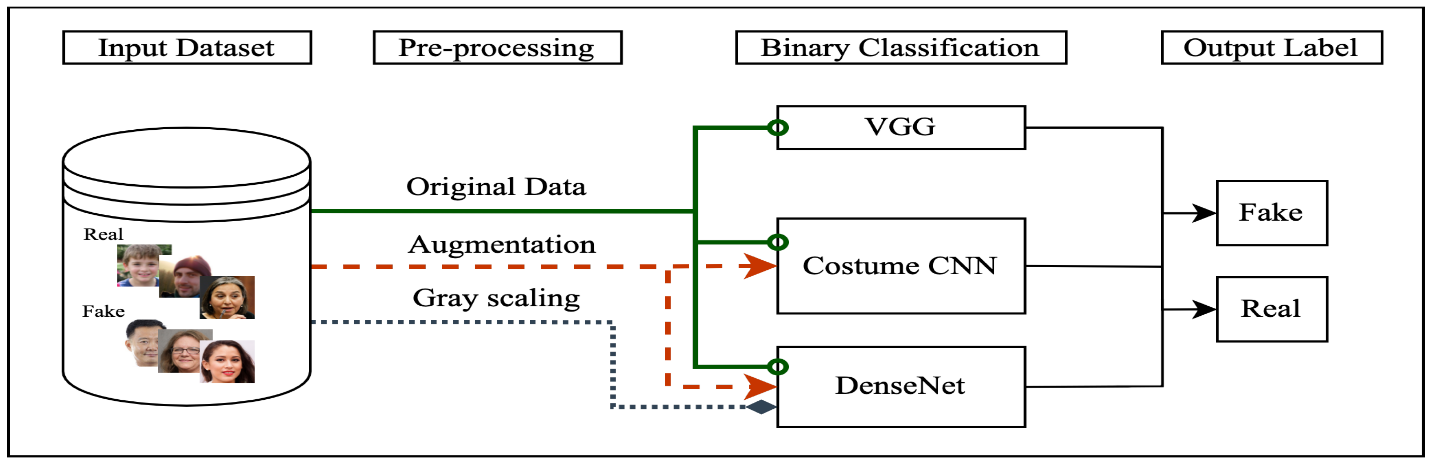
* Accuracy Score
* Accuracy Curve
* Loss Curve
* Precision and Recall
* Confusion Matrix

## 6. Applied Algorithms

Deep learning approach has been used in this research for detection of deep fakes from videos, More specifically Convolutional neural networks (CNNs).

### 6.1 Convolutional Neural Network (CNN)

The employment of CNN in the detection of deep fakes is a crucial aspect that plays a significant role in combating this escalating threat. The advanced capabilities provided by CNN enable it to thoroughly analyze and comprehend intricate patterns within visual data, effectively distinguishing between authentic and manipulated images or videos. Through the utilization of multiple layers encompassing convolutional and pooling operations, key features are extracted from input data, permitting the identification of irregularities and artifacts associated with deep fakes. However, CNN surpasses basic image analysis as it also possesses the ability to recognize temporal connections within videos via sequential processing of frames. This functionality allows for proficient recognition of subtle alterations or distortions introduced by deep fake algorithms. By continuously training Convolutional Neural Network models on extensive datasets consisting of real and synthetic media samples, researchers can enhance the accuracy and versatility of the network. This makes CNN an invaluable tool in addressing the challenges posed by the misuse of AI-generated content. **Figure 1** shows the diagram of CNN that is used for deep fake detection.



**Figure 2**: A Sample Diagram of CNN used for Deepfake Detection

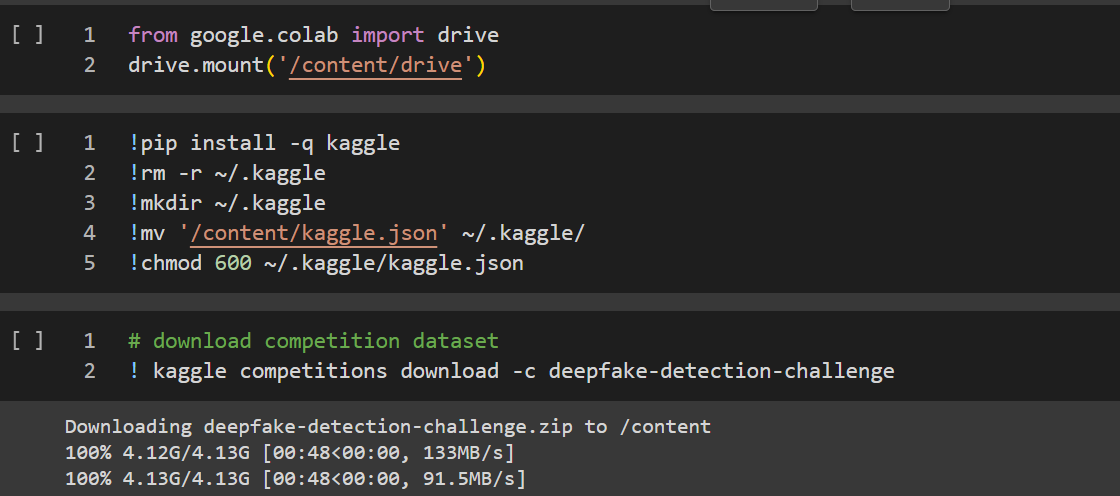
## 7. Flow of Applying CNN on DFDC Dataset

For applying the proposed architecture on DFDC dataset, a specific flow is followed, typically involved in a deep learning lifecycle for various problems such as image classification, segmentation, etc. This is done in multiple steps, i.e., from loading the dataset into memory, till the prediction on extracted frames from dataset and making predictions.

* Loading Dataset from Kaggle into Google Colab
* Extracting Sequenced Frames from Videos
* Placing Frames in a Particular Folder in Google Drive
* Applying Image Data Augmentation to Frames
* Initializing Deep Learning Model (CNN)
  + Making Layers of CNN (InceptionResNetV2)
  + Fine Tuning CNN
  + Applying an LSTM head to CNN
  + Making the Final Layer to Complete CNN Architecture
* Fitting the Model
* Calculating Accuracy & Graphs
  + Loss Graph
  + Accuracy Graph
* Evaluation and Testing

### 7.1 Loading the Kaggle Dataset

Originally, the dataset is placed at Kaggle, which needs to be imported into Google Colab working environment. For this purpose, a specific API command is taken from Kaggle dataset to be called in Google Colab. The dataset is loaded and extracted from Kaggle as follows:

****

**Figure 3:** Importing Kaggle Dataset to Google Colab

### 7.2 Processing the Dataset

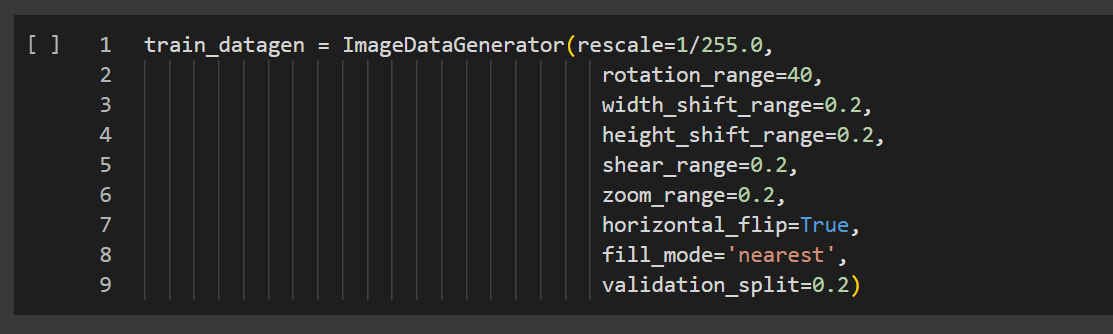
After loading the dataset, we are preprocessing video frames. The primary objective is to extract faces from videos, which are categorized as either 'REAL' or 'FAKE' based on information provided in the accompanying metadata.json file. This preprocessing step is necessary to prepare the face data needed to train the deepfake detection model.

Face Detection Loop: The script iterates through each video, utilizing OpenCV (cv2.VideoCapture) to open and process the video frames.  
The dlib library's face detector (dlib.get\_frontal\_face\_detector()) is employed to identify faces in each frame.

This is consistent with the broader goal of developing an efficient algorithm for distinguishing between real and manipulated video in the DFDC dataset. The labels help to perform the first phase of data preparation, ensuring that subsequent model training procedures are performed on well-organized and labeled facial data.

### 7.3 Applying Image Data Augmentation

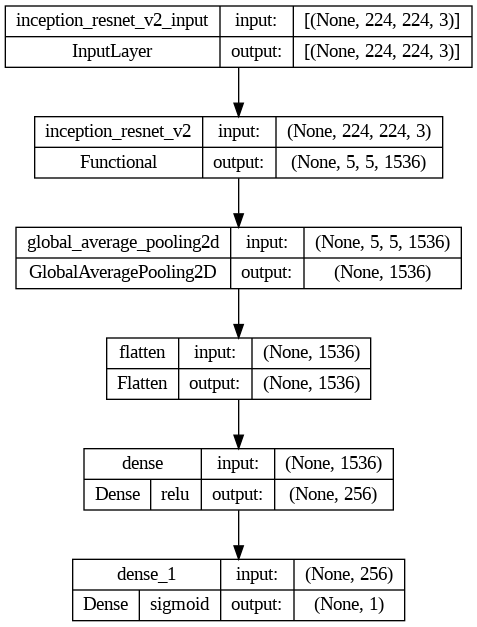
Data augmentation is a process in deep learning which is used to generate more images based on existing images for a smooth training process. This is done by applying multiple operations on images such as rotation, flipping, horizontal, vertical flipping, etc. The images are also scaled to a particular fraction, such as 1/255, keeping in view the 256 color channels in a colored image. Given code snap shows how this step is taken.



**Figure 4**: Applying Image Data Augmentation using Keras

### 7.4 Applying CNN to Dataset

The CNN architecture is applied to DFDC dataset by using a pre-trained CNN known as InceptionResNetV2. Pre-trained CNN is also fine-tuned on the dataset to learn feature representations. The pre-trained InceptionResNetV2 model is loaded with weights from ImageNet and configured to accept input images of dimensions (X, Y, 3). The layers of the base model are set to non-trainable to preserve the pre-trained weights. The Sequential model is then constructed, with the InceptionResNetV2 base model serving as the first layer. The subsequent layers include global average pooling, flattening, a dense layer with 256 neurons and ReLU activation, and a final dense layer with a sigmoid activation function, representing the binary classification output for deepfake detection. The code overall establishes a transfer learning approach, leveraging the InceptionResNetV2 architecture for feature extraction and adding trainable layers specific to the deepfake detection task.



**Figure 5**: Visualization of the Neural network model

### 7.5 Fitting Model to Dataset

After model initialization, it is fitted to the training dataset and validated on testing dataset using a specific number of hyperparameters such as optimizer, loss function, number of epochs, target matric, model call-backs, etc. The loss function, optimizer & target metric are used in model compilation. **Table 1** shows the overall configuration of hyperparameters used in the model.

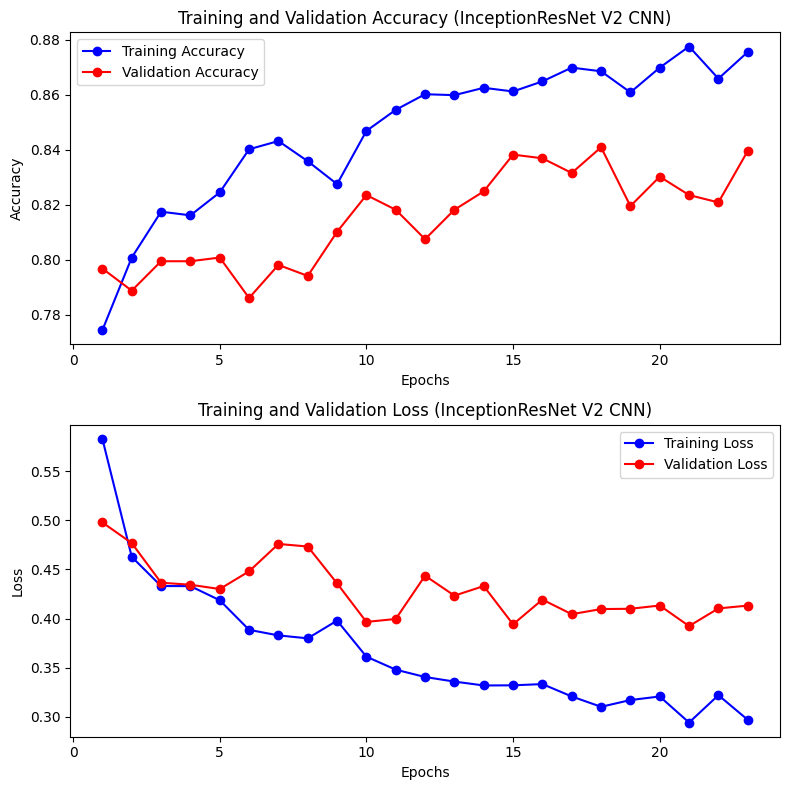
**Table 1:** Configuration of Hyperparameters

|  |  |  |
| --- | --- | --- |
| **Name of Hyperparameter** | **Description** | **Value** |
| Batch Size | The number of samples per gradient update during training. | 128 |
| Optimizer | Technique used to update learning rate while training | Adam |
| Learning Rate | The step size at which the optimizer adjusts the model weights during training. | 1e-5 |
| Loss Function | Function that calculates the predicted values with actual ones | Binary Cross Entropy (BCE) |
| Epochs | Number of times the training dataset is passed through the model while training | 50 |
|  |  |  |
| Callbacks | Metric that controls training | Early Stopping, Model Checkpoint, Reduce LR on Plateau |
| Number of Dense Layers | The number of dense (fully connected) layers in the model | 3 |
| Activation Among CNN & Dene Layers | The activation function used in the convolutional and dense layers. | ReLU |
| Activation on Final Layer | Same as above | Sigmoid |

### 7.6 Model Result Visualizations

After the training of model, results are calculated on the dataset using learning curves (accuracy & loss graphs), and the overall accuracy. The model has attained a testing accuracy of 83% on the testing dataset. Give below are accuracy & loss graphs of using this approach.

**7.6.1 Accuracy Graph of Model**



**Figure 6**: Accuracy and Loss Graph of CNN

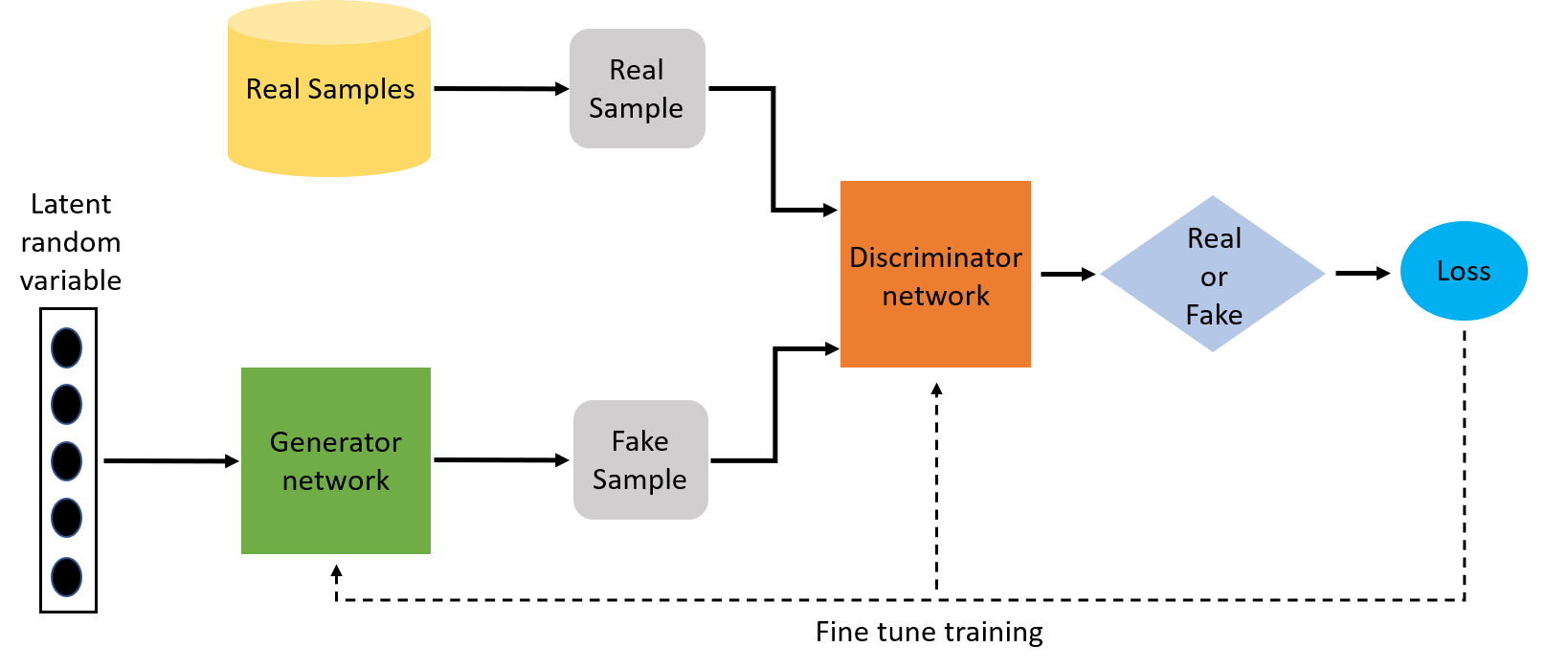
In above 2 graphs;

* The blue lines show the accuracy & loss on training dataset
* The red lines show the accuracy & loss on validation (testing) dataset

## 8. Technology Review

### 8.1 Deepfake Video Creation and Detection

There are lots of way to create a deepfake and tons of software are available on the internet that can create deepfakes without the need of any experience. These deepfakes are created using artificial intelligence and deep learning methods which rely on CNN called auto-encoder. Auto-encoder encodes the input by applying dimension reduction and image compression and then a decoder which reconstructs the image from the constructed representation by the encoder. Upgraded version of this method is using Generative Adversarial Networks (GANs) which is unsupervised deep learning algorithm [10].

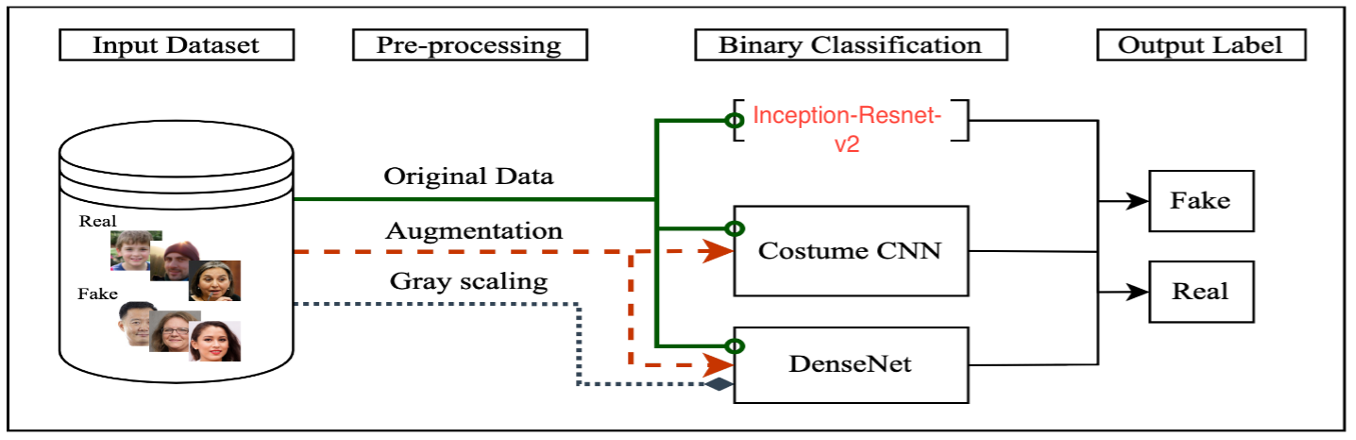


#### **Figure 7**: Deepfake Generation using GAN

The main part is to train the GAN model. The generator creates fakes samples and the discriminator find the difference between real and fakes. The generator is trained to create realistic outputs to deceive the discriminator, and the discriminator is trained to identify the fake inputs correctly.

﻿A deepfake detection is a system that will predict whether an image/video is a real image/video or it is a deepfake. Algorithms are used in the same way to create different deepfakes thus leaving some discrepancies during the editing process. Factors such as compression variations, lighting differences, and temporal contrasts such as lip and eye movements can be specifically targeted to train models to detect Deepfake videos [11]. Deepfakes can be used for various purposes as cybercrime, misinformation and creating tensions between countries. There are several approaches to detect deepfakes in videos. Audio-visual synchronization is the inconsistencies between the audio and the lip movement of the person that does not match. Other approach is Face Manipulation detection when the person in the video shows unnatural facial expressions or inconsistent head movement.

Among the proposed methods for Deepfake detection, Convolution Neural Networks (CNN) is a popular choice. CNNs have demonstrated good performance and scalability for applications involving image and video processing compared to supervised learning methods in artificial intelligence. CNN has a unique ability to extract features from images that can be used in various applications. With feature extraction by Convolutional Neural Network, other supervised learning tools can be used for Deepfake final classification to create a better and more accurate Deepfake detection model. By continuously training Convolutional Neural Network models on extensive datasets consisting of real and synthetic media samples, researchers can enhance the accuracy and versatility of the network. This makes CNN an invaluable tool in addressing the challenges posed by the misuse of AI-generated content. Figure 8 shows the diagram of CNN that is used for deep fake detection.



**Figure 8**: A Sample Diagram of CNN used for Deepfake Detection

### 8.2 Key Features:

**Feature Extraction:** CNNs excel at automatically learning hierarchical features from visual data, making them highly effective in extracting relevant patterns and information from images and video frames.

**Convolutional Layers:** These layers use convolutional operations to detect spatial patterns in the input data. They are crucial for capturing local features and hierarchies in visual information.

**Pooling Layers:** Pooling layers downsample the spatial dimensions of the data, reducing computational complexity while retaining important features.

**Fully Connected Layers:** These layers connect every neuron to every other neuron in the subsequent layer, enabling the network to learn global patterns and make predictions.

**Activation Functions:** Common activation functions, such as ReLU (Rectified Linear Unit), introduce non-linearity to the model, enabling it to learn complex relationships in the data.

## 9. Conclusion

In this report, the complete process of implementing the CNN based architecture is explained on DFDC dataset. DFDC (deep fake detection challenge) is a famous benchmark dataset which was held at Kaggle in 2020, with a purpose to extract a model for intelligent detection of deep fakes from videos. After a thorough examination of existing research studies, the convolutional architecture seems to be efficient for detection of deep fakes, since a video is composed of a sequence of frames. The proposed architecture is applied to the dataset by first extracting the frames from videos, plus placing them in a separate folder for loading and applying a deep learning model on the dataset. After model’s training on the dataset, it is found out that the model attains a good accuracy of 83% on the test dataset as the highest.

## 10. References

1. Santos, I., Castro, L., Rodriguez-Fernandez, N., Torrente-Patino, A., & Carballal, A. (2021). Artificial neural networks and deep learning in the visual arts: A review. Neural Computing and Applications, 33, 121-157.
2. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.
3. Güera, David, and Edward J. Delp. "Deepfake video detection using recurrent neural networks." 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, 2018.
4. Karandikar, Aarti, et al. "Deepfake video detection using convolutional neural network." International Journal of Advanced Trends in Computer Science and Engineering 9.2 (2020): 1311-1315.
5. B. Kaddar, S. A. Fezza, W. Hamidouche, Z. Akhtar and A. Hadid, "HCiT: Deepfake Video Detection Using a Hybrid Model of CNN features and Vision Transformer," 2021 International Conference on Visual Communications and Image Processing (VCIP), Munich, Germany, 2021, pp. 1-5, doi: 10.1109/VCIP53242.2021.9675402.
6. Rahman, Ashifur, et al. "Short And Low Resolution Deepfake Video Detection Using CNN." 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2022.
7. S. R. Adnan and H. A. Abdulbaqi, "Deepfake Video Detection Based on Convolutional Neural Networks," 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 2022, pp. 65-69, doi: 10.1109/ICDSIC56987.2022.10075830.
8. Nataraj, L., Mohammed, T. M., Chandrasekaran, S., Flenner, A., Bappy, J. H., Roy-Chowdhury, A. K., & Manjunath, B. S. (2019). **Detecting GAN generated fake images using co-occurrence matrices**. *arXiv preprint arXiv:1903.06836*.
9. <https://www.kaggle.com/competitions/deepfake-detection-challenge/data>
10. Karandikar, A., Deshpande, V., Singh, S., Nagbhidkar, S., & Agrawal, S. (2020). Deepfake video detection using convolutional neural network. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), 1311-1315.
11. Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE international workshop on information forensics and security (WIFS)* (pp. 1-7). IEEE.