Image Splicing & Deepfake Detection

Submitted in partial fulfilment of the requirements of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Priyanka Khadse (Roll no. 31)

Tanmay Kulkarni (Roll no. 36)

Sanskriti Shevgaonkar (Roll no. 62)

Omkar Shinde (Roll no. 63)

Name of the Mentor

Prof. Uma Ade

Department of Computer Engineering

Watumull Institute of Electronics Engineering and Computer Technology

Ulhasnagar-421003

University of Mumbai

(AY 2022-23)

CERTIFICATE

This is to certify that the Project entitled “Image Splicing & Deepfake detection” is a bonafide work of

Priyanka Khadse (Roll no. 31)

Tanmay Kulkarni (Roll no. 36)

Sanskriti Shevgaonkar (Roll no. 62)

Omkar Shinde (Roll no. 63)

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Computer Engineering**”.

(Prof. Uma Ade)

Mentor

(Prof. Rahul Jinturkar)

Head of Department

(Prof. Sunita Sharma)

Principal

PROJECT APPROVAL

This Project entitled “Image Splicing & Deepfake detection” by

Priyanka Khadse (Roll no. 31)

Tanmay Kulkarni (Roll no. 36)

Sanskriti Shevgaonkar (Roll no. 62)

Omkar Shinde (Roll no. 63)

is approved for the degree of **Bachelor of Engineering** in **Computer Engineering**.

Examiners

Prof. Uma Ade

(Internal Examiner Name & Sign)

(External Examiner name & Sign)

Date:

Place:

ABSTRACT

The boom of digital images coupled with the development of approachable image manipulation software has made image tampering easier than ever. As a result, there is massive increase in number of forged or falsified images that represent incorrect or false information. This project presents a novel approach to detect copy move and splicing image forgery using a Convolutional Neural Network (CNN) with three different models i.e. ELA (Error Level Analysis), VGG16 and VGG19. The proposed method applies the pre-processing technique to obtain the images at a particular compression rate. These images are then utilized to train the model and further the images are classified as authentic or forged. The project also presents the experimental results of the proposed method and performance evaluation in terms of accuracy.

The deep learning algorithms so powerful that creating a indistinguishable human synthesized video popularly called as deep fakes have became very simple. Scenarios where these realistic face swapped deep fakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily envisioned. In this work, we propose a novel method for multi-modal deepfake detection with minimum resources. The proposed solution is designed with a Siamese network-based deepfake model with invariant of constructive loss and triplet loss. Contrastive loss uses the trained network's output for a positive example and calculates its distance to an instance of the same class and contrasts it with the range to negative samples. The triplet loss was computed by positioning the baseline that minimizes the distance to positive samples but maximizes the distance to negative samples. The technique has been evaluated against publicly available datasets: DFDC, FaceForensics++, and Celeb-DF (v2) and shows promising results in detecting manipulations during self and cross-testing.

ACKNOWLEDGEMENT

We would like to extend our sincere and heartfelt gratitude to our professor Mrs. Uma Ade who has helped us in this endeavour and has always been very cooperative and without his help, cooperation, guidance and encouragement, the project couldn't have been what it evolved to be.

We extend our heartfelt thanks to our faculty for their guidance and constant supervision, as well as, for providing us the necessary information regarding the project. We must thanks to our classmates for their timely help& support for compilation of this project.

Last but not least, we would like to thank all those who helped us (directly or indirect) towards the completion of this project within a limited time frame.

Priyanka Khadse (Roll no. 31)

Tanmay Kulkarni (Roll no. 36)

Sanskriti Shevgaonkar (Roll no. 62)

Omkar Shinde (Roll no. 63)

(B.E. Sem VIII)

TABLE OF CONTENTS

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

The development of user-friendly image manipulation software that is available at reasonable prices, has made the manipulation of such content easier than ever. In particular, some of these images are tampered with in such a way that it is impossible to detect even by humans. Three of the most common manipulations techniques are Copy-move: a specific region from the image is copy pasted within the same image. Splicing: a region from an authentic image is copied into a different image. Removal: an image region is removed and the removed part is then in-painted. In recent researches many deep learning techniques are used to detect forgery.

Deepfake is a technique for human image synthesis based on neural network tools like GAN (Generative Adversarial Network) or Auto Encoders etc. These tools super impose target images onto source videos using a deep learning technique and create a realistic looking deep fake video. We can use the limitation of the deep fake creation tools as a powerful way to distinguish between the pristine and deep fake videos. During the creation of the deep fake the current deep fake creation tools leaves some distinguishable artifacts in the frames which may not be visible to the human being but the trained neural networks can spot the changes.

1.2 Objectives

- To develop, deploy and manage the training dataset and keep adding more sets to improve the efficiency of the system.
- To study the working of CNN algorithm and trying to improve the performance of the algorithm for image splicing.
- To discover the distorted truth of the deep fakes.
- To provide a easy to use system for used to upload the video and distinguish whether the video is real or fake.
- To develop a user-friendly GUI i.e a simple UI/UX where user can easily add the image and check the results.

1.3 Motivation

Due to the huge development of technology, the usage of the digital image has been expanding day by day in our daily lives. Because of this forgery of the digital image has turned out to be increasingly straightforward and indiscoverable. At present technology where anything can be controlled or changed with the assistance of modern technology had started to disintegrate the authentic of images, counterfeiting and forgeries with the move to the Mega pixels, which gives a new way for forgery. To Overcome this we will design a system which will help in detecting the forged Images using CNN. the number of fake videos and their degrees of realism has been increasing due to availability of the editing tools, the high demand on domain expertise. Spreading of the Deep fakes over the social media platforms have become very common leading to spamming and peculating wrong information over the platform. To overcome such a situation, Deep fake detection is very important. So, we describe a new deep learning-based method that can effectively distinguish AI generated fake videos (Deep Fake Videos) from real videos. It’s incredibly important to develop technology that can spot fakes, so that the deep fakes can be identified and prevented from spreading over the internet.

Chapter 2

LITERATURE SURVEY

2.1. Survey of Existing System

Research Paper	Description	Comments
A Robust Copy Move Forgery Classification Using End to End Convolution Neural Network	A new CNN model which gives accuracy of 93-95%. Pooling and convolution layers give better accuracy.	-End to End CNN model -Pixel level localization.
Multimedia Forensic: An Approach for Splicing Detection based on Deep Visual Features	-Inception module based CNN network -accuracy 98.76 % 97.92% -split into 2 modules	-Gives high accuracy as global features of visual data are captured
Copy-Move Forgery Detection using Residuals and Convolutional Neural Network Framework	-use SDMFR and LFR -accuracy of 95.97 % CoMoFoD and 94.36% BOSSBase -6 convolutional layers with ReLU	- Not robust where no post processing operation has been applied.
DeepVision: Deepfakes detection using human eye blinking pattern:	Uses Long-term Recurrent Convolution Network (LRCN) Accuracy-87.5%	Detection by Eye Blinking describes a new method for detecting the deepfakes by the eye blinking as a crucial parameter leading to classification of the videos as deepfake or pristine.
Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos	Normal accuracy 97.69% Capsule network using noise accuracy 99.23%	Capsule networks to detect forged images and videos uses a method that uses a capsule network to detect forged, manipulated images and videos in different scenarios, like replay attack detection and computer-generated video detection.

2.2. Limitation of Existing Systems

Disadvantages of Movement Copy-move forgery detection

- Computation time is high.
- High Probability of false matches.
- Calculations of Zernike moment is complex

Disadvantages of Splicing forgery recognition

- For low quality image as size of block decrease so decreases efficiency
- Cannot deal with JPEG compression
- Average accuracy is less than other methods which are based on wavelet
- Unable to detect forgeries with scaling and heavy JPEG conversion

Disadvantages of existing deepfake system

- Sometimes video is original, audio voiceover is done according to lips syncing, which cannot be detected by existing system.

2.3. Problem Statement

The most common type of digital image forgery are copy move forgery and image splicing that manipulate the images in a way that is hard to be perceived by human perceptual system. We have proposed a new image forgery detection method based on deep learning technique, which utilizes a convolutional neural network (CNN) to automatically learn hierarchical representations from the input RGB colour images. Recent advances in deep learning have led to a dramatic increase in the realism of fake content and the accessibility in which it can be created. Already in the history there are many examples where the deepfakes are used as powerful way to create political tension[14], fake terrorism events, revenge porn, blackmail peoples etc. So it becomes very important to detect these deepfake and avoid the percolation of deepfake through social media platforms. We have taken a step forward in detecting the deep fakes using Siamese and end-to-end training paradigms.

2.4. Scope

The scope of the project is to provide the user a user-friendly application which would help to identify the tampered image and the original image using CNN method. The scope of the 13 project also includes the learning different ways to improve the efficiency of the algorithm and have a deep knowledge of the algorithm. There are many tools available for creating the deep fakes, but for deep fake detection there is hardly any tool available. Our approach for detecting the deep fakes will be great contribution in avoiding the percolation of the deep fakes over the world wide web. We will be providing a web-based platform for the user to upload the video and classify it as fake or real. This project can be scaled up from developing a web-based platform to a browser plugin for automatic deep fake detections. Even big application like WhatsApp, Facebook can integrate this project with their application for easy pre-detection of deep fakes before sending to another user. A description of the software with Size of input, bounds on input, input validation, input dependency, i/o state diagram, Major inputs, and outputs are described without regard to implementation detail.

Chapter 3

PROJECT PROPOSAL

3.1. Proposed Methodology

The project is divided into 2 segments i.e.

1)Image Forgery 2)Video Forgery

For Image Forgery:

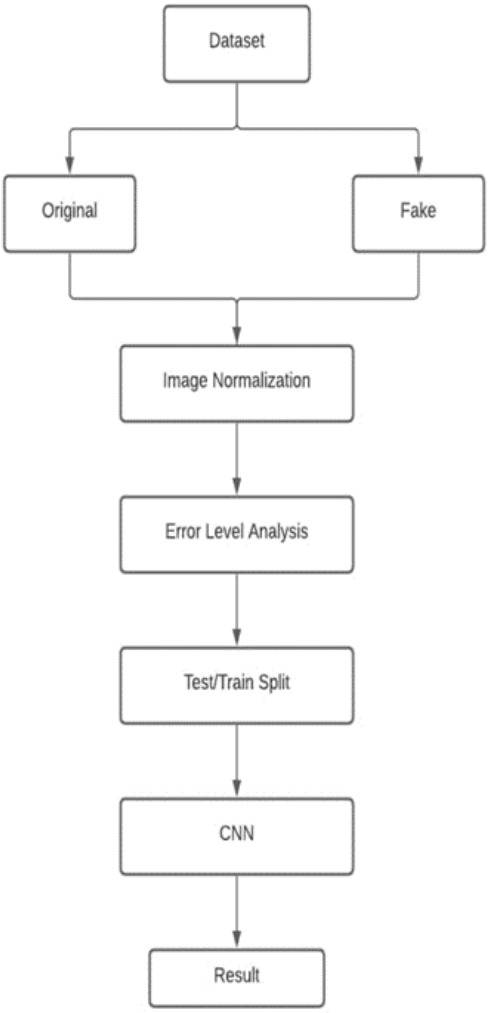


Fig 3.1 Block Diagram for Proposed Method of Image forgery detection

We have used Image Splicing technique for the Image forgery detection. First step is to divide the dataset into 2 categories: Original and Fake Images. Then we perform Data Pre-processing which involves normalization of images. The purpose of normalization is to make sure that all the images have similar data distribution. For normalization, the whole dataset is resized into 128*128 pixels. Then we perform Error Level Analysis, it is a technique for identifying images that have been forged by storing images at a certain quality level. This strategy is based on the fact that JPEG compression eliminates a lot of information about the brightness and colour of the original image.

$$ELA = O–R$$

The modified parts of the image in the forged ELA-generated image are brighter than the corresponding original components. Then finally we fit the data into the CNN architecture. CNN excels in terms of large-scale image classification. Convolutional layers, pooling layers, and fully connected layers are the three layers that make up CNN. Convolutional layer is used as feature extractor that learns feature representation. This is from the image that is input to CNN. Meanwhile, the Pooling layer shrinks the convolution layer's output map and prevents overfitting. The output of the convolution layer is divided into numerous small grids, and the maximum value of each grid is used to create a reduced image matrix. Even if the picture object is translating(shifting), the technique assures that the features retrieved are the same (shifting). Then, the fully connected layer will interpret these features and perform the high-level reasoning processes.

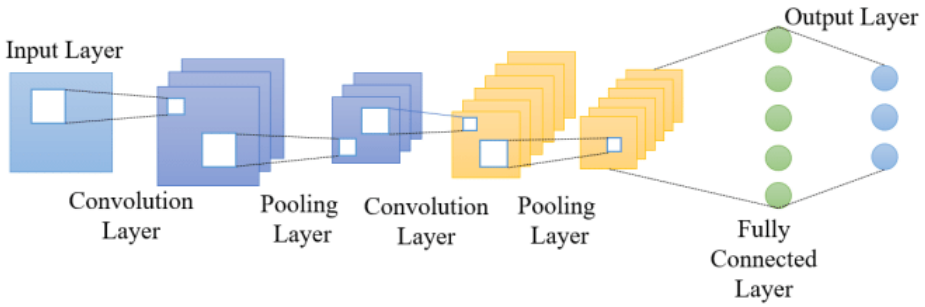
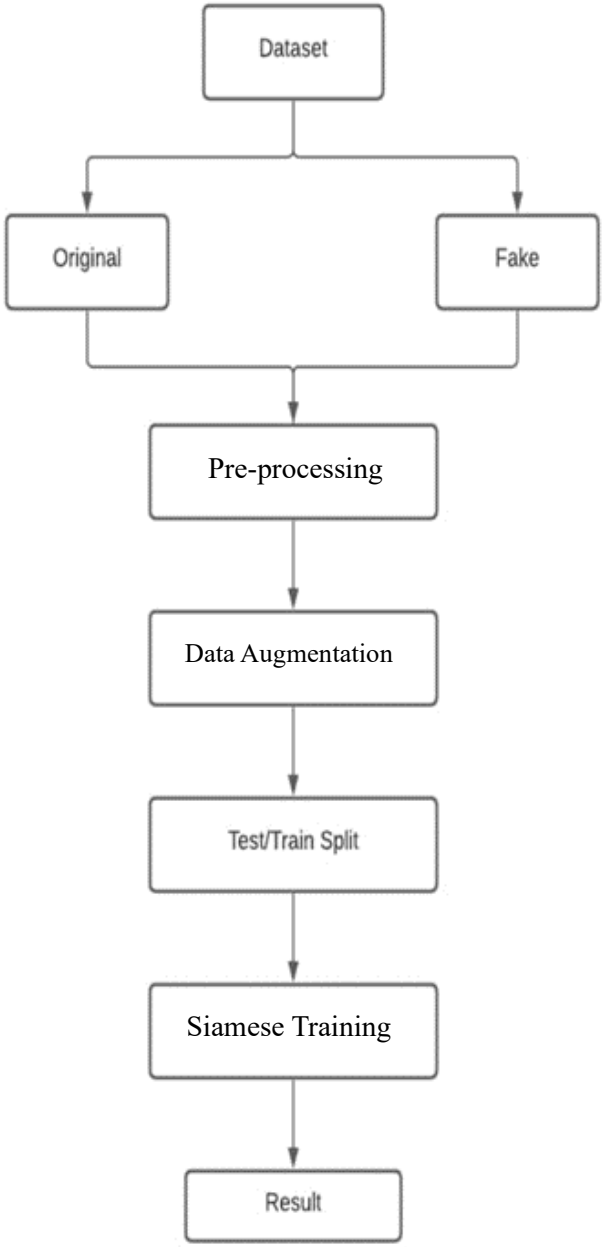


Fig 3.2 CNN Architecture

For Video Forgery:

We have used Deepfake detection method for finding out forgery in a video.

Fig 3.3 Block Diagram for Proposed Method of Image forgery detection



The proposed method in this research article aims to detect deepfakes in videos using machine learning techniques. The method uses three publicly available datasets (DFDC, FF++, and Celeb-DF v2) and employs a pre-processing step using Multitask Cascaded CNN (MTCNN) for face extraction and data augmentation. Transfer Learning with EfficientNet-B7, EfficientNetV2, and Vision Transformer base models is used, initialized with ImageNet pre-trained weights. The network training is performed using two techniques: end-to-end training and Siamese training. The paper proposes using an ensemble of models to improve prediction performance. The loss function for the end-to-end training method is calculated using the LogLoss.

$$L_L = \frac{-1}{N}\sum_{i=1}^{N}\left(y_i log(\sigma(\hat{y}_i)) + (1-y_i) log(1-\sigma(\hat{y}_i))\right)$$

Here yi is prediction score of the ith image while σ(.) is the sigmoid activation.

The loss function for Siamese training method is calculated using the triplet margin loss.

$$L_T = max(0, m + d(a,p) - d(a,n))$$

In the above equation, d(a, p) is the distance between anchor and positive image which is given by ||f(Ia)−f(Ip)||2, d(a, n) is the distance between anchor and negative image which is given by ||f(Ia) − f(In)||2, and m is the margin.

3.2. Details of Hardware & Software requirements

Software Requirements:

OS: Windows 7+, Android Studio 2010, SQL Server i.e SQLite of mobile, Python 3.0, Framework: PyTorch 1.4, Libraries: : OpenCV, Face-recognition.

Hardware Requirements:

Intel Xeon E5 2637 (3.5GHz), RAM (16GB), Hard Disk (100GB), Graphic Card (5GB)

Chapter 4

PLANNING AND SCHEDULING

4.1. Planning and Formulation

1.Initial Discussion

2.Collecting Information

3.Problem Identification

4. Design of image splicing

5. Development of Image splicing

6. Design of DF detection model

7. Development of DF detection model

8. Implementation

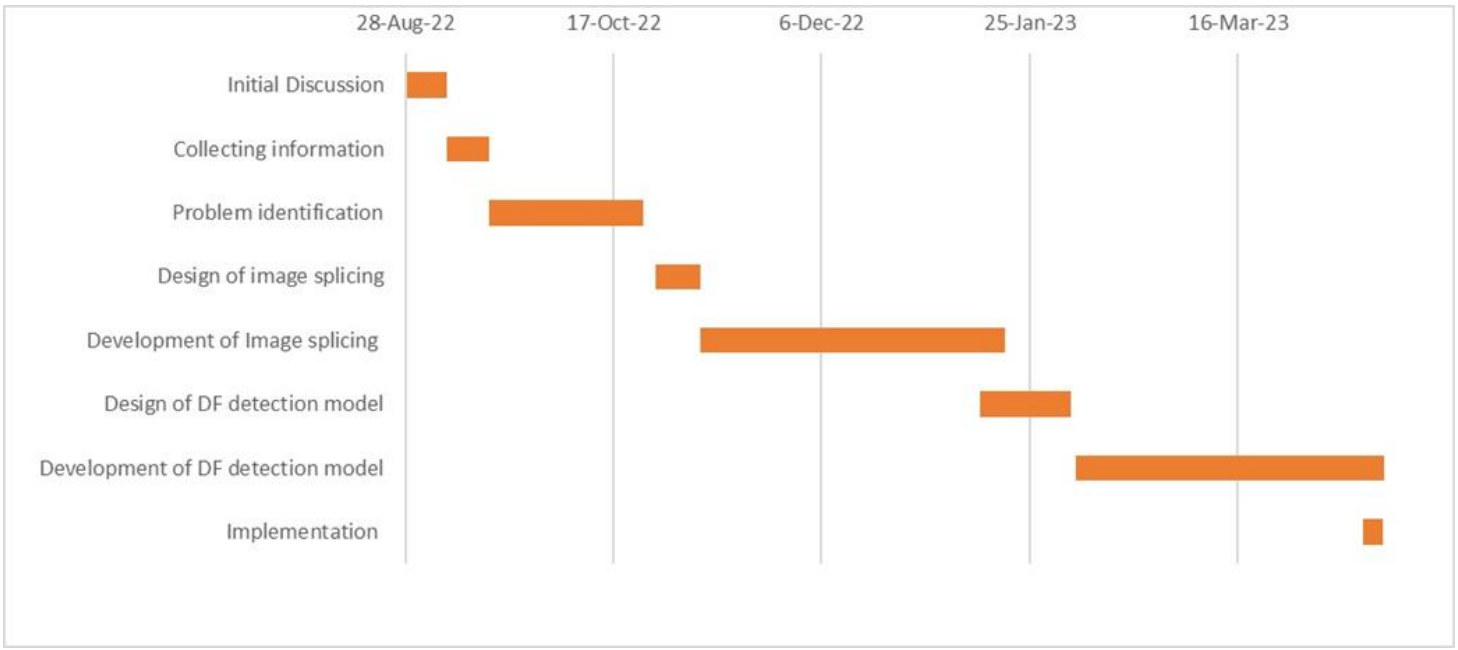


Fig 4.1 Gantt Chart

Chapter 5

System Design

5.1. Architecture/Framework

For Image Splicing:

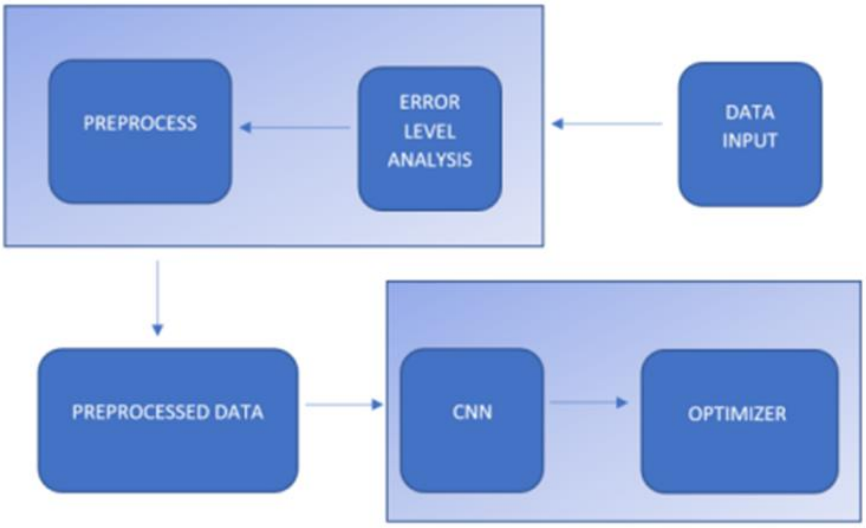


Fig 5.1 Architecture for image splicing

The above figure depicts the methodology followed for the image forgery detection. First of all the dataset is being collected from the CASIA-2.0 & NC2016 these two datasets.

In data pre-processing, it involves normalization of images. The purpose of normalization is to make sure that all the images have similar data distribution. For normalization, the whole dataset is resized into 128*128 pixels.

In the crucial step of Error level Analysis, we've to convert an image to ELA, so the pre-processed photos must first be resaved at a certain quality level. The image is whitened or brightened as a result of this technique. In order to resave the images, consider both genuine and falsified images that have been pre-processed and then resave photos at a specific compression level. Finally, Images that have been pre-processed and images that have been re-saved are compared to see how much of a difference there is. The modified parts of the image in the forged ELA generated image are brighter than the corresponding original components. Having ELA in the pre-processing tends to have a huge advantage in further processing of the neural network as the ELA converted image contains only non-redundant information and has similar intensity contrast as compared to the nearby pixels. Due to this reason, our neural network training optimizes in only 8-9 epochs with a learning rate of 0.0001.

The next step involves changing the image size. We want each RGB value to lie between 0 and 1 so we divide each cell value by 255.0 to normalize the training of the neural network so that it converges faster. After that each image is categorically labelled where 0 stands for authentic image and 1 stands for forged image. Subsequently, the images are divided into two sets – 80% of the images are taken for the training set and the remaining 20%for the validation set for the working of the convolutional neural network.

In CNN, convolutional layer is used as feature extractor that learns feature representation. This is from the image that is input to CNN. Meanwhile, the pooling layer shrinks the convolution layer's output map and prevents overfitting. In general, there are stacks of numerous convolutional and pooling layers before the fully connected layer that serve to extract a more abstract feature representation. Pooling layer usually decodes the image (such as 2x2) in the aggregation into a single unit. The first layer of the CNN is a convolutional layer with a kernel size of 5x5 and a total of 32 filters. The second layer CNN consists of a convolutional layer with a kernel size of 5x5 and 32 filters, as well as a max pooling layer with a size of 2x2. After that, the max pooling layer adds dropouts for 0:25 to avoid overfitting. The flatten layer changes the feature map by flattening it into the feature vector and passing it to a fully connected layer after images are trained from a fine-tuned pre-processed model. In the last dense layer of models, the fully connected layer is employed for pattern recognition, and the SoftMax activation function is used to convert the feature vector in a probabilistic manner. The training set is compared to the test set using SoftMax activation, and a probability distribution on actual and forged images is returned.

Optimization applied during the training is RMS Prop Optimizer in which the learning rate for each parameter is automatically adjusted without our intervention which helps in optimizing several parameters like number of features, number of training samples, target MSE, number of hidden layers, etc.

For the training of the neural network, two popular neural architectures have been utilized mainly VGG16 and VGG 19.

For Deepfake Detection:

In our experiments, we used three publicly available datasets: Deepfake Detection Challenge (DFDC), FaceForensics++ (FF++) and Celeb-DF (v2). DFDC dataset [20] was generated by Facebook AI for the Kaggle challenge. It contains over 470 GB of videos distributed as 19154 real and 100000 fake. FF++ dataset-utilizes about 500k frames from

1000 original videos downloaded from the internet with each video containing a minimum of 280 frames. The Celeb-DF (v2) is an enlarged version of the original CelebDF (v1) dataset. The dataset consists of 5639 deepfakes generated from 590 real videos downloaded from YouTube and has about 6011 videos in total.

For pre-processing we perform 2 steps: 1) Face Extraction: The Multitask Cascaded Convolutional Neural Network (MTCNN), as described in reference , was utilized for extracting faces from video frames. MTCNN comprises of three stages, namely, the Proposal Network (P-Net), the Refine Network (R-Net), and the Output Network (O-Net). P-Net is responsible for detecting faces across various resolutions, R-Net is utilized to remove overlapping detection boxes using non-max suppression, and O-Net is employed to obtain bounded faces using five facial landmarks. Ultimately, square colour images of size 224x224 are obtained as a result of this process. 2) Data Augmentation: We perform extra data augmentations offered by Augmentations library. We apply rgb-shift, random brightness/contrast, random gamma, hue saturation, horizontal flips, and jpeg-compression.

Transfer learning is a technique that uses knowledge gained from learning one task (source task) to improve the learning of another task (target task). In this work, three base models were used: EfficientNet-B7, EfficientNetV2, and Vision Transformer. These models were initialized with pretrained weights from the ImageNet dataset.

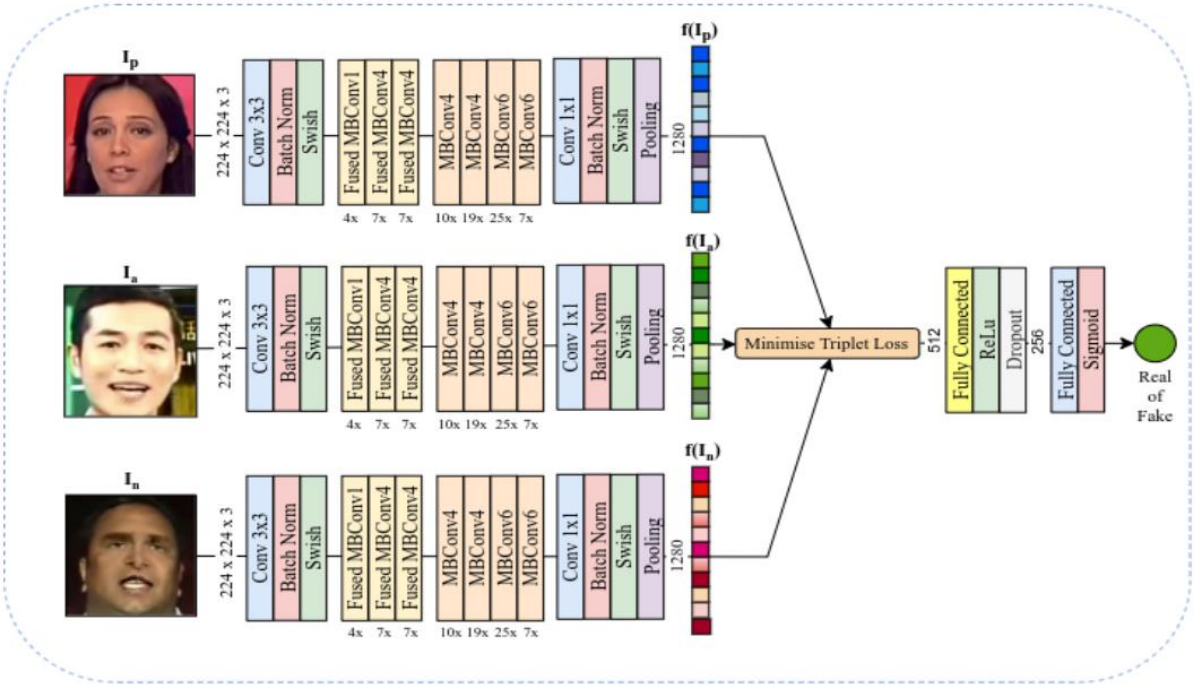


Fig 5.2 Siamese Network Architecture with EfficientNetV2.

The proposed approach involves training models using two techniques: end-to-end and siamese training. End-to-end directly learns the solution from the dataset while Siamese learns

similarities and differences between classes to generate embeddings. Ensembling is adopted to improve prediction performance.

1) End-to-End Training (EE): We pass an extracted frame from the video as input to the model which returns a prediction probability score yˆ. A predication score closer to 0 indicates a real face and score closer to 1 indicates a fake face.
2) Siamese Training (ST): To optimize the model, we use a ranking loss called Triplet margin loss. In a triplet network, the aim is to use distance comparisons to extract useful representations. To do so, the network uses triplet loss to cluster features of a same class together and features of another class are kept away in the feature space.

Finally, a classification network with output size of two (corresponding to the classes, real and fake) consisting of multilayer perceptions with two hidden layers of length 512 and 256 and which uses the end-to-end training method is applied on top of the network. Siamese training involves training the feature extractor using triplet loss and then fine-tuning the classification layers using log loss. The feature extractor is trained with 16 triplets (8 with real image anchor and 8 with fake image anchor) randomly selected from the training set, with a margin hyper-parameter of 1.2. After training the feature extractor, the classification layers are fine-tuned using the end-to-end training method with the same learning rate schedule, validation procedure, and number of iterations.

To justify the Siamese training concept as an addition to the traditional end-to-end technique, we used the t-SNE algorithm to generate a projection and understand if the features extracted by the feature extractor of Siamese training algorithm is discriminatory for the task or not. The projection obtained from faces in DFDC dataset using EfficientNetV2 as the base model and trained using Siamese method. We can see that all the real faces are clustered in the bottom half while the fake faces are clustered in the top region of the projection.

(a) Initial face embeddings (b) EfficientNetV2 ST

Fig 5.3 t-SNE plots of features obtained from faces of DFDC dataset

Chapter 6

Results and Discussion

6.1 Implementation Details

For Image Splicing:

We have chosen to work with three models to test the accuracy of our algorithm i.e., ELA, Vgg16 and Vgg19. We have mainly used two famous datasets -Cassia v2.0 and Media Forensics NC2016 for the training and testing part. For training part, we have used approx. 1000 images for every model.

a) ERROR LEVEL ANALYSIS

ELA is basically a forensic method which is helpful in identifying portions of an image having different level of compressions. In this process, ELA works by resaving the original input image at compression level of 95% and then comparing the difference with the original image. With successive resave operations been performed, they produce in- creased level of errors. After some sufficient number of resaves, it reaches its minimum error level producing a darker ELA. If suppose we observe any section of the image having different level of significant error level then there is a high possibility of digital modification. For the ELA training as shown in figure, we have used roughly 1000 images wherein 500 images were authentic and 500 images were tampered. We also provide two directories containing tampered and authentic images. We have used learning rate as 0.01 and no of epochs as 9. As an output we get training curve and confusion matrix. After this we can either choose to test images or train another model like VGG16.

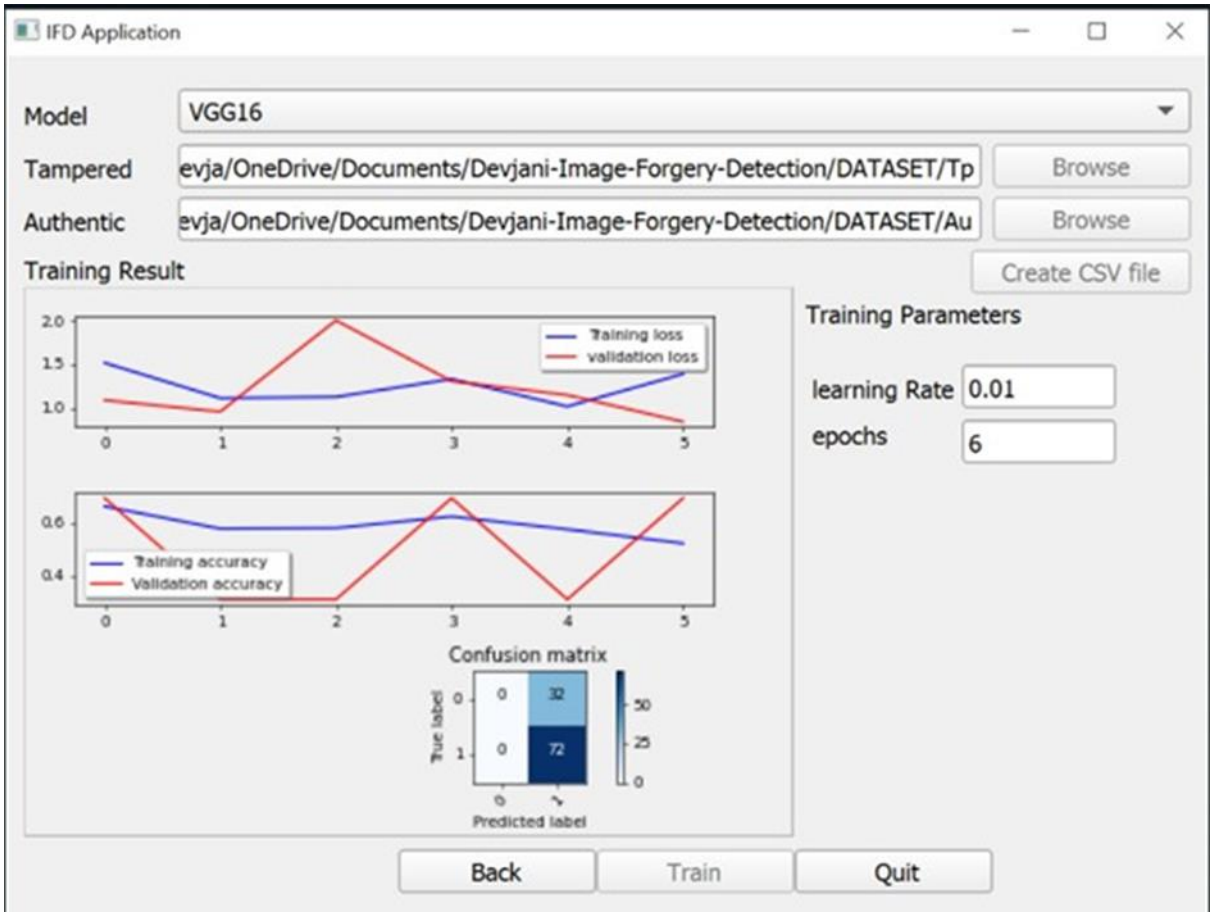


Fig 6.1 ELA Training Curve

For the ELA testing as shown in figure , we have used roughly 1000 images wherein 500 images were authentic and 500 images were tampered.

Below given image as shown in figure is an example of Authentic image of CASIA v2.0 dataset .We first select the model ELA using dropdown and browse for the image we have to test. Image properties are visible in LHS of the application main window. After that we click on “test” button to test for the accuracy of forged/not forged. We observe that for the authentic image it returns “Decision Not Forged” with an accuracy of 85.7%.

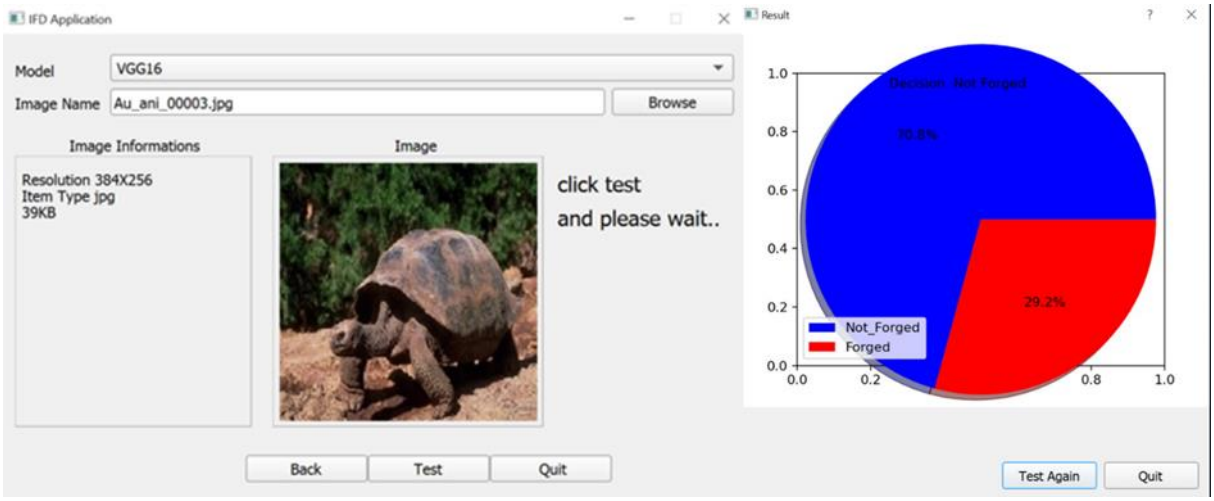


Fig 6.2 Result of Authentic Image – ELA

We test another image as shown in figure, but this time from tampered directory and observe the results. We observe that it detects the forged image with 100% accuracy.

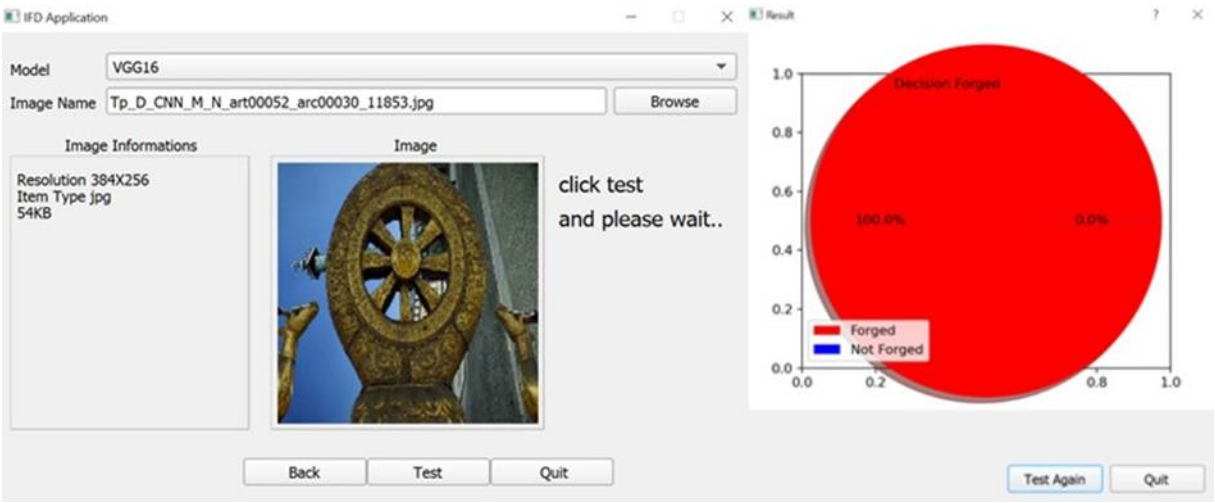


Fig6.3 Result of Tampered Image - ELA

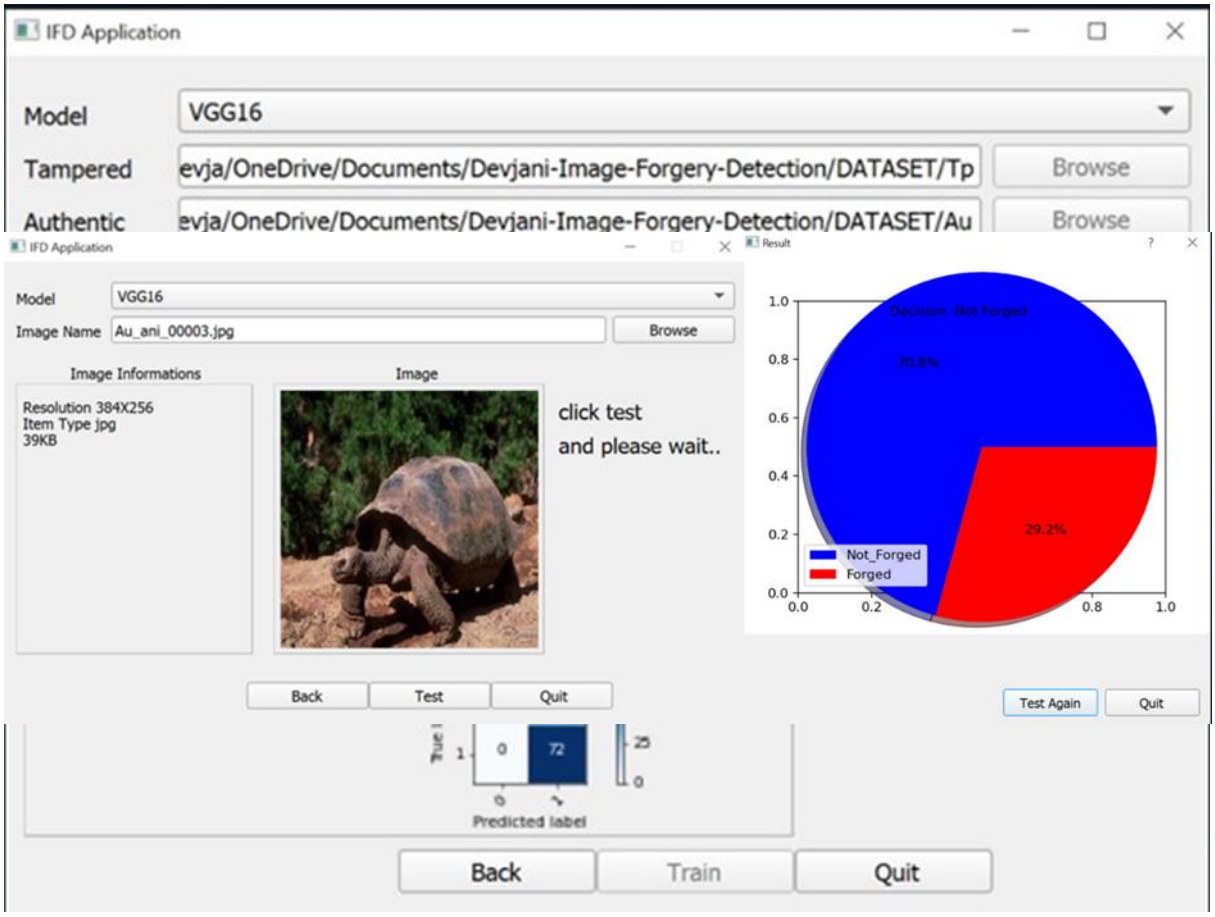
So as said above, the testing of ELA generates a satisfying accuracy of 70.6% on 1000 images tested so far. The accuracy of the model is 71.6%.

2. VGG16

The proposed CNN -VGG16 model first does some preprocessing (involving ELA) and then it passes the processed data to its CNN layers.

For the VGG16 training, we have used roughly 1000 images wherein 500 images were authentic and 500 images were tampered. We also provide two directories containing tampered and authentic images. We have used learning rate as 0.01 and no of epochs as 6. As an output we get training curve and confusion matrix as shown in figure. After this we can either choose to test images or train another model like VGG19.

Fig 6.4 VGG16 Training Curve



For testing purpose we have used a whooping set of 1000 images with 500 Authentic and 500 tampered images .The images were taken both from CASIA and NC2016 datasets to better test the efficiency of our work on complex images. The below figure shows the result of testing on Authentic image for VGG16 model. It gave “Decision Not Forged” with an accuracy of 70.8%.

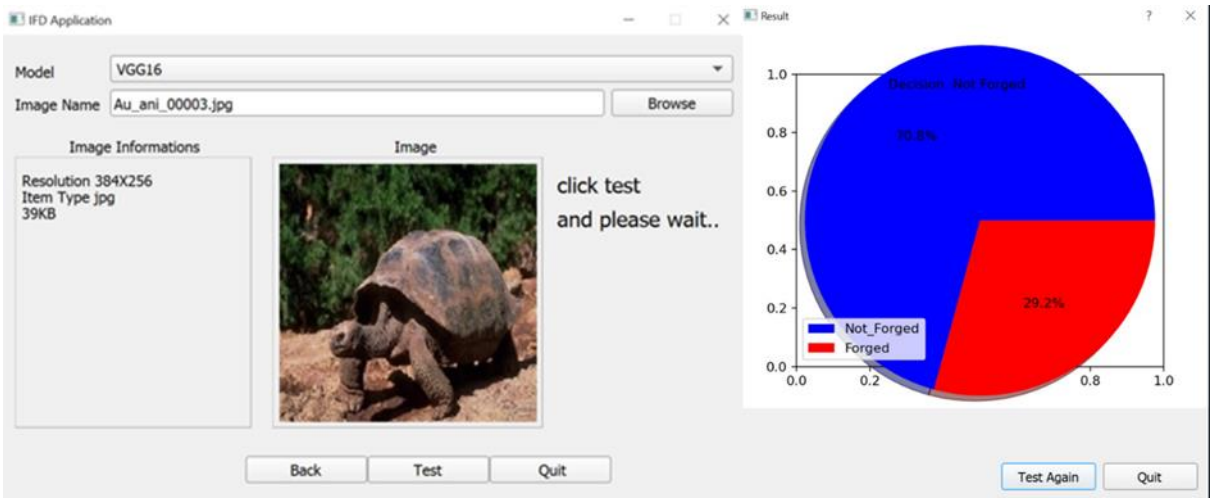


Fig 6.5 Result of Authentic Image - VGG16

The below figure shows the result of testing on Tampered image for VGG16 model. It gave “Decision Forged” with an accuracy of 100%.

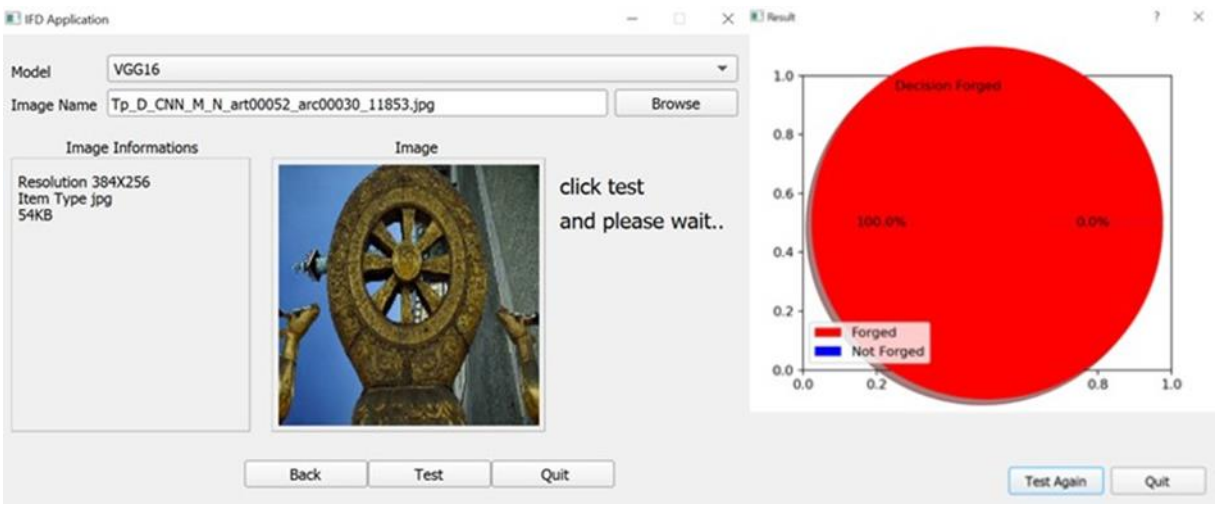


Fig 6.6 Result of Tampered Image - VGG16

As said above, this VGG16 model was tested for a total of 1000 images and the it was ob- served that it gave an accuracy of 71.6%, which is more than what we got for ELA model and thus demonstrates our true purpose of choosing VGG16 and VGG19 along with ELA. The accuracy of the model is 71.6%.

3. VGG19

For VGG19 model, we trained roughly 13000 images from NC2016 and CASIA dataset. After training we got a curve of training v/s validation loss and accuracy as shown in figure Training accuracy is 94.1379.

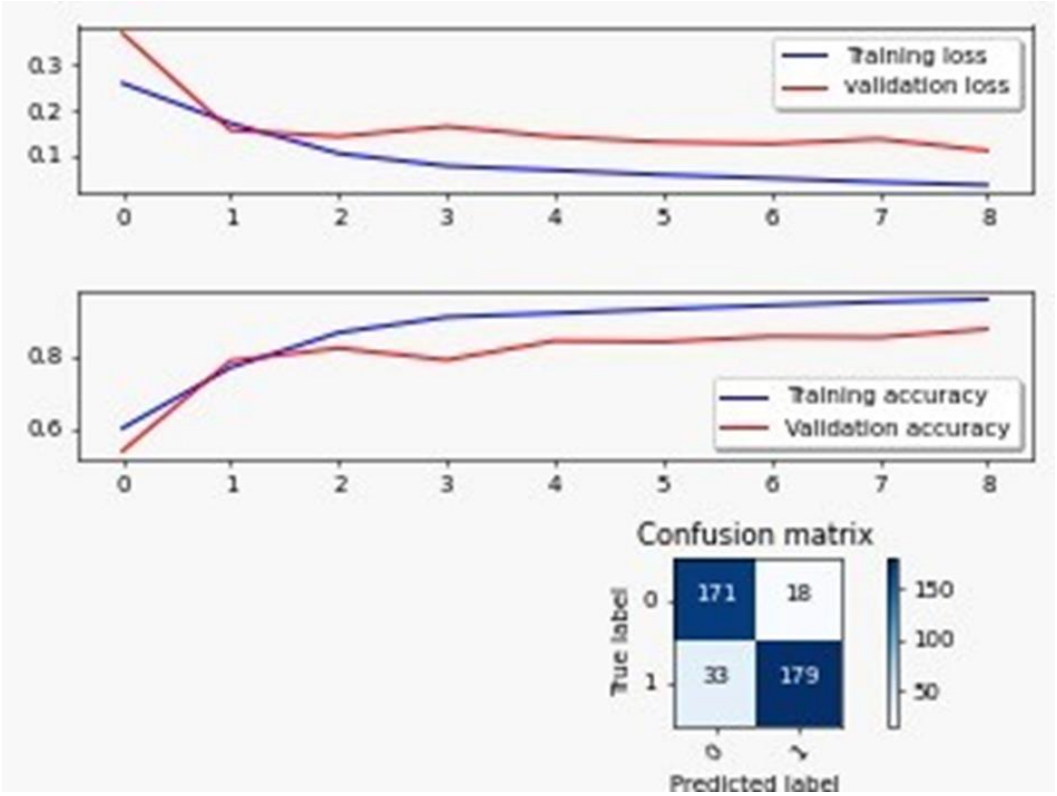


Fig 6.7 VGG19 Accuracy

After this we began our testing part on total of 1000 images with 500 authentic and 500 forged images.

The below figure shows the result of testing on Authentic image for VGG19 model. It gave “Decision Not Forged” with an accuracy of 100%.

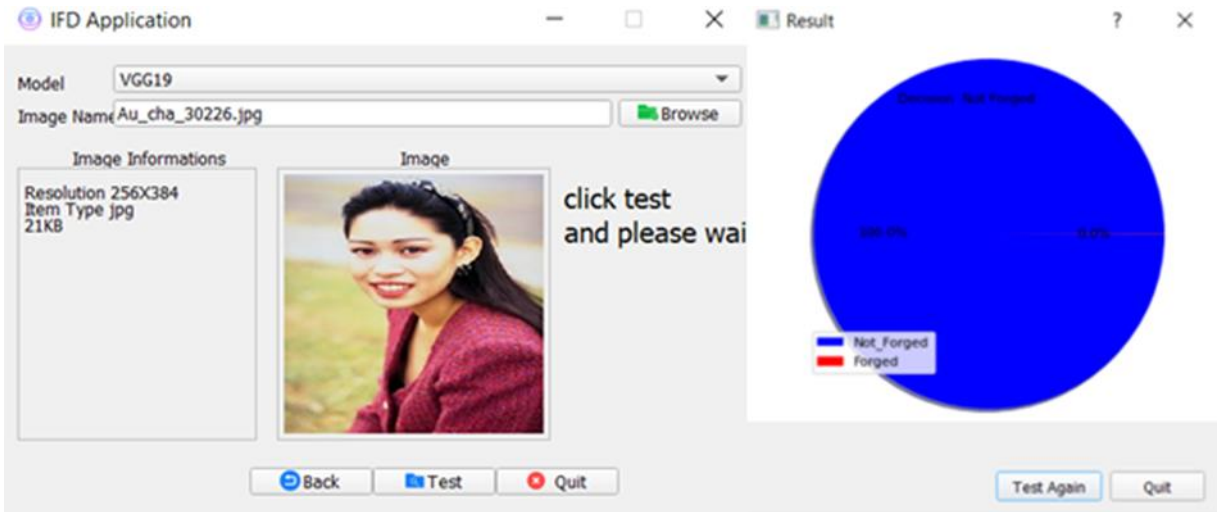


Fig 6.8 Result of Authentic Image - VGG19

The below figure shows the result of testing on Tampered image for VGG19 model. It gave “Decision Forged” with an accuracy of 100%. The accuracy of the model is 72.9%.

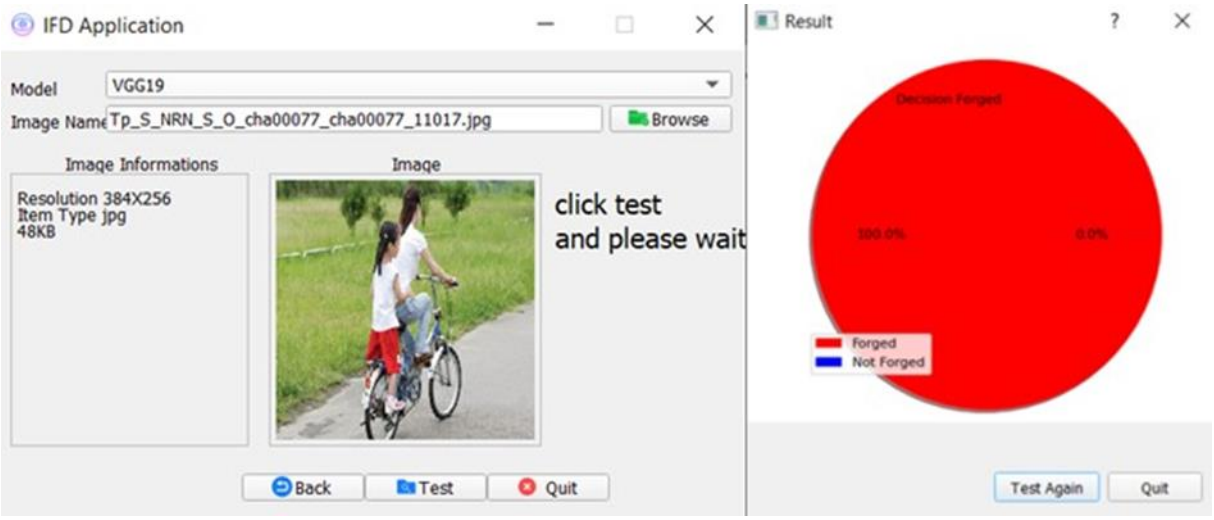


Fig 6.9 Result of Tampered Image - VGG19

It is observed by far that our algorithm is quite good at detecting the forged image as forged than real images as not forged. Also the accuracy of VGG16 is more than ELA and VGG19 .It is also observed that VGG19 performs better than VGG16 at the time of training as training/-validation accuracy is comparatively more as shown in the figure.

model	ELA	VGG 16	VGG 19
true positive	315	347	329
true negative	185	153	171
false positive	391	369	400
false negative	109	131	100
accuracy	70.6%	71.6%	72.9%

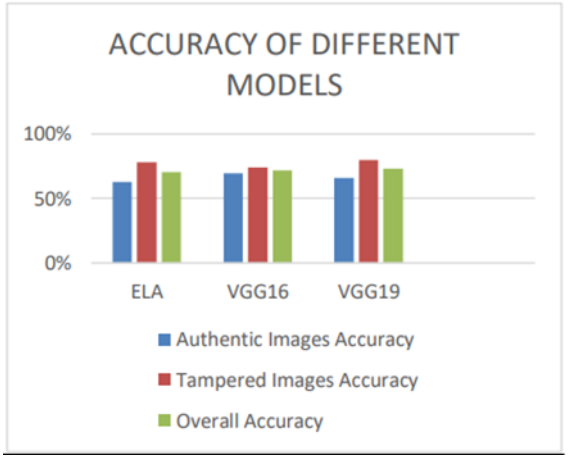


Fig 6.10 Accuracy table

For Deepfake Detection:

a) TimmV2:

Timm v2 uses EfficientNet as one of its model architectures for image classification tasks. EfficientNet is a family of models that were designed to achieve state-of-the-art performance while using fewer parameters and FLOPs (floating-point operations) compared to other models like ResNet or Inception.

Timm v2's implementation of EfficientNet uses a combination of depth, width, and resolution scaling to create a family of models that can be tuned to different resource constraints. The base architecture consists of a series of convolutional layers with different filter sizes, followed by global average pooling and a fully connected layer for classification.

Timm v2's implementation of EfficientNet also includes additional features like stochastic depth and drop connect regularization, which can help improve the model's performance and reduce overfitting.

To use EfficientNet in Timm v2, you can simply import the model from the library and load the pre-trained weights. Then, you can fine-tune the model on your own dataset using transfer learning techniques, or use it directly for inference on new images. Timm v2 also provides various tools for evaluating and visualizing the performance of the model.

Fig 6.11 Fake and real video detection on dfdc dataset using TimmV2.

b) Timm V2ST:

Timm V2st is an optimized variant of the Timm V2 library designed for efficient and accurate object detection and segmentation. It can handle both image and video data and is particularly useful for video-related tasks such as action

recognition and video captioning. Timm V2st includes several pre-trained models that use a combination of 2D and 3D convolutional layers to extract spatial and temporal features. It also includes feature fusion, adaptive feature pooling, and optimizations such as mixed precision training and model pruning to improve training and inference speed. Users can fine-tune pre-trained models on their own dataset and use the library's tools for data loading, evaluation, and visualization.

Fig 6.12 Timm V2st provides a flexible and powerful platform for spatiotemporal modeling in computer vision tasks.

Eg. Fake and real video detection on dfdc dataset using TimmV2ST.

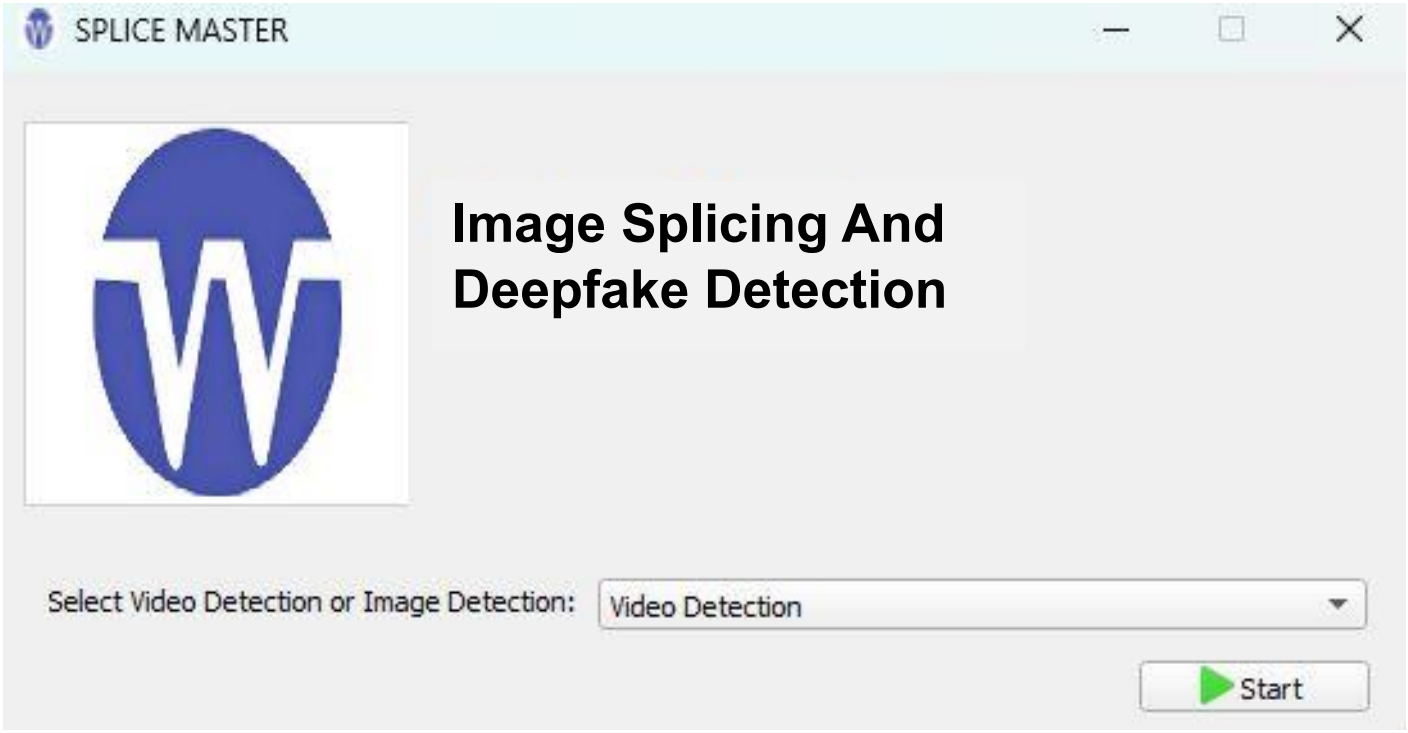
6.2 Result Analysis

Hence, we have implemented the image forgery detection using three models i.e. ELA, VGG16, and VGG19 with their respective accuracies as 70.6%, 71.6%, and 72.9%. We have worked with two most popular datasets for image recognition i.e. CASIA V2.0 and NC2016 datasets, for which it was observed that NC2016 is complex than CASIA V2.0 and hence forgery detection on NC2016 dataset results in less accuracy. We have also observed that VGG19 model is better than VGG16, and hence it is better than ELA model for detection of both authentic and forged images. It is also observed that the individual accuracy for tampered images detection is more than the individual accuracy of authentic images detection for all the three models.

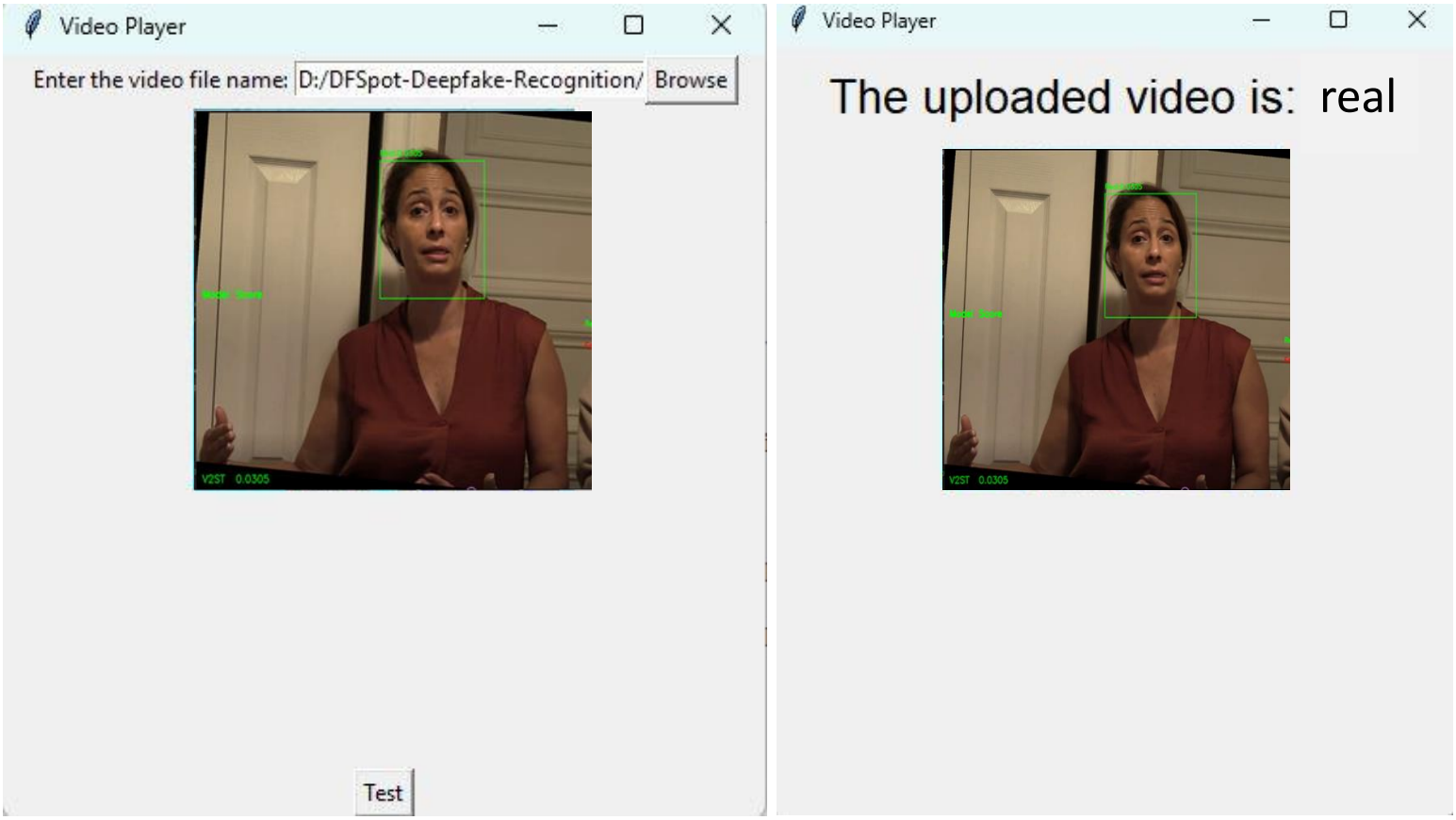
In deepfake Detection, the EfficientNetV2 ST model outperformed all other models with AUC scores of 0.9407 and 0.9764 on DFDC and FF++ datasets respectively. We can also observe that most of our Siamese trained models have more AUC score and low loss than their end-to-end trained counterpart, thus implying that Siamese trained models are more

robust. We find that our Vision Transformer ST model outperforms all other models when cross-tested on Celeb-DF (v2) dataset with an AUC score of 0.7866.

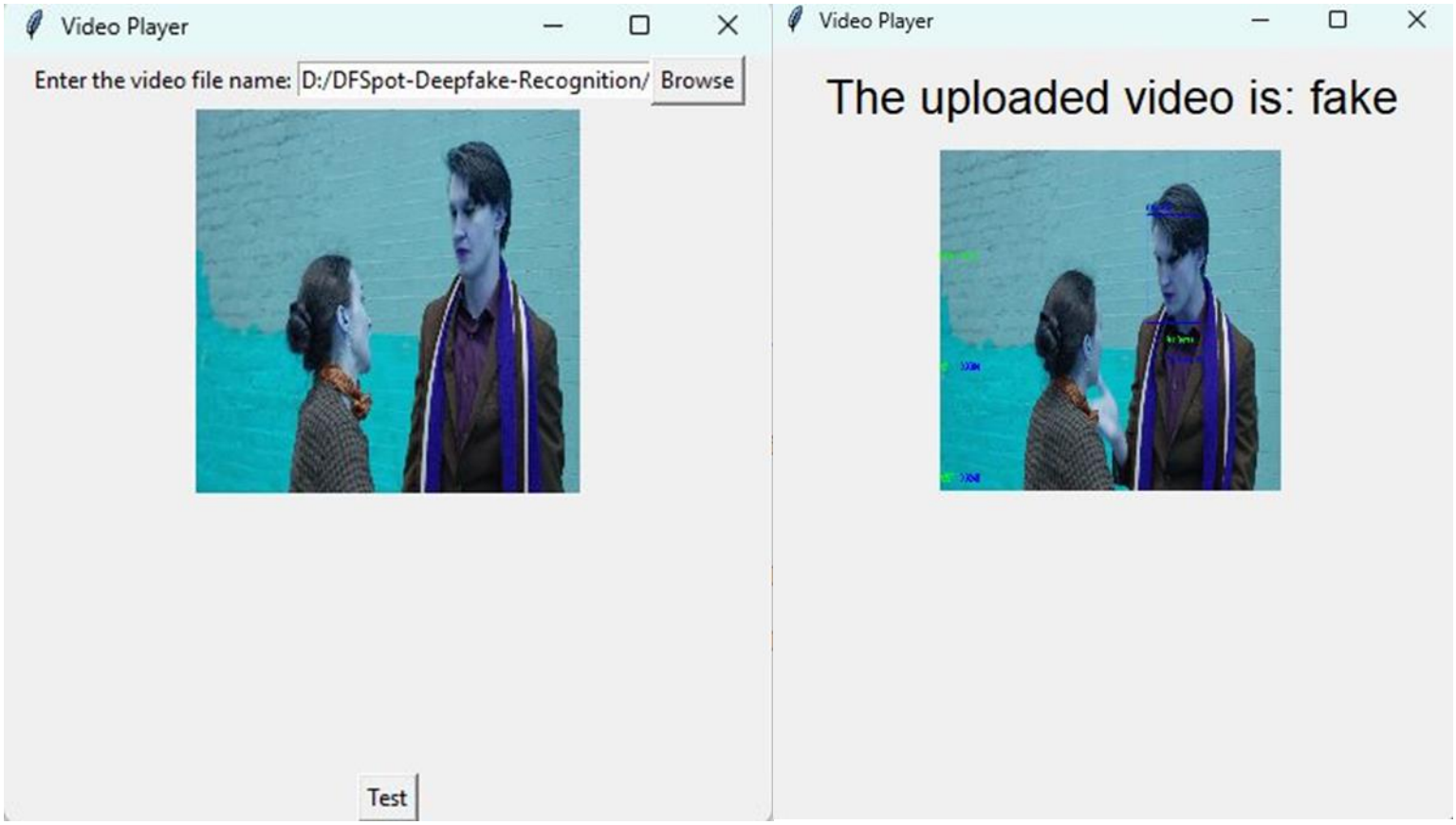
Results along with GUI



6.13 GUI Main Window



6.14 GUI Real Detection



6.15 GUI Fake Detection

Chapter 7

Conclusion

In this work we experimented with using a CNN in the image forgery detection task. More specifically, we used a CNN network to extract features from two datasets of varying difficulty, namely CASIA v2.0 and NC16. Furthermore, the extracted features were then used to train and test an SVM, achieving an accuracy of 96.82% and 84.89% on CASIA v2.0 and NC16 respectively.

In the age of digital media, detecting manipulated content has become more important than ever. Our solution for detecting deepfakes involves using a combination of models that are trained using siamese and end-to-end approaches. Our research shows that the siamese training method results in better performing models than those trained using end-to-end methods. Our EfficientNetV2 ST model performs similarly to state-of-the-art models on the FF++ dataset, but outperforms them on the DFDC dataset. By using an ensemble of models, our approach increases the overall reliability of deepfake detection. In the future, we plan to investigate different techniques for selecting frames and incorporating temporal information to further enhance the accuracy of deepfake detection.

Chapter 8

References

[1] Souradip Nath & Ruchira Naskar, “Automated image splicing detection using deep CNNlearned features and ANN-based classifier”, Springer-Verlag London Ltd., part of Springer Nature 2021- Jan 2021

[2] Sanjeev Kumar, Suneet K. Gupta ,”A Robust Copy Move Forgery Classification Using End to End Convolution Neural Network” ,8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) – June 2020

[3] Yuan Rao , Jiangqun Ni ,“Deep Learning Local Descriptor for Image Splicing Detection and Localization”, IEEE Access (Volume: 8) – January 2020.

[4] Eman I,Abd El-Latif,Ahmed Taha,Hala H Zayed,“A Passive Approach for Detecting Image Splicing using Deep Learning and Haar Wavelet Transform”, Arabian Journal for Science and Engineering, Volume 6 -December 2020.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets”, in Advances in Neural Information Processing Systems, pages 2672-2680, 2014.

[6] U. A. Ciftci, I. Demir and L. Yin, ”FakeCatcher: Detection of Synthetic Portrait Video using Biological Signals,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3009287.

[7] S. Tariq, S. Lee and S. Woo, ”A Convolutional LSTM based Residual Network for Deepfake Video Detection,” arXiv:2009.07480 [cs.CV], 2020.