DEEPFAKE AI

A MINOR PROJECT PROPOSAL REPORT
SUBMITTED TO THE DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING IN
PARTIAL FULFILLMENT OF THE
REQUIREMENTS

FOR THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING
In
Computer Science and Engineering

SUBMITTED BY

PRERIT MUJOO (2021A1R098)

SACHIN DEV (2021A1R102)

ASHU PAL (2021A1R116)

SUBMITTED TO: Mr. Navin Mani Upadhyay

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Jammu, India
2024

CANDIDATE'S DECLARATION

I, **Prerit Mujoo (2021a1r098), Sachin Dev (2021a1r102), Ashu Pal (2021a1r116),** hereby declare that the work which is being presented in the Mini Project Report entitled, "**Deepfake AI**" in the partial fulfillment of requirement for the award of degree of B.E. (CSE) andsubmitted in the CSE Dept. , Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me under the supervision of **Mr. Arslaan Manzoor Zargar**, (MIET) and **Ms. Harashleen Kour** (MIET).The matter presentedin this Mini Project Report has not been submitted in this or any other University / Institute for the award of B.E. Degree.

Signature of the Student *Dated*:

Prerit Mujoo

Sachin Dev

Ashu Pal

Computer Science and Engineering Department

Model Institute of Engineering and Technology (Autonomous)

Kot Bhalwal, Jammu, India

(NAAC "A" Grade Accredited)

Ref. No.: **Date:**

CERTIFICATE

Certified that this seminar report entitled **"Deepfake Detection."** is the bonafide work of "**Prerit Mujoo (2021a1r098), Sachin Dev (2021a1r102), Ashu Pal(2021a1r116) of 6th Semester, CSE, Model Institute of Engineering and Technology (Autonomous), Jammu"**, who carried out the seminar work under my supervision during May 2024.

Ms. Harashleen Kour	**Mr. Arslaan Manzoor Zargar**
Co-Coordinator	**Coordinator**
Mini Project Lab Incharge	**Mini Project Lab Incharge**
CSE, MIET	**CSE, MIET**

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to everyone who made the completion of this proposal possible. We are especially thankful to the Department of Computer Science and Engineering at the Model Institute of Engineering and Technology (MIET) for giving us the chance to explore our interests and ideas within the realm of engineering through this project.

Our sincere thanks go to Mr. Navin Mani Upadhyay, the Head of the Department of Computer Science and Engineering, for his support and guidance. Additionally, we deeplyappreciate the significant contributions of our faculty members, who provided valuable assistance during the research and feasibility study phases of the project.

We are incredibly grateful for the support and encouragement from everyone involved, which has inspired us throughout this journey. We appreciate their insightful guidance, constructive criticism, and friendly advice.

Submitted by:

Prerit Mujoo (2021A1R098)
Sachin Dev (2021A1R102)
Ashu Pal (2021A1R116)

ABSTRACT

In today's digital age, tremendous advances in artificial intelligence (AI) and machine learning have resulted in a new phenomenon known as "deepfakes." Deepfakes are synthetic media in which a person's face or voice is digitally altered to produce false content that appears genuine. While deepfakes can be used for fun, they also pose a huge threat to personal privacy, security, and misinformation.

This project's goal is to create a comprehensive deepfake detection system utilizing deep learning techniques. The suggested method uses a convolutional neural network (CNN) architecture to scan facial features and detect abnormalities or inconsistencies that could indicate a deepfake. The model is trained on a vast dataset of both actual and modified photos, guaranteeing that it can generalize to a variety of deepfake kinds and scenarios.

To improve the system's explainability and transparency, we include a Gradient-weighted Class Activation Mapping (Grad-CAM) module. Grad-CAM creates a heatmap that highlights the regions of the face that contribute the most to the model's prediction, giving users a visual representation of the detection process. This feature is critical for increasing trust and responsibility in the system's decision-making.

The project's implementation makes use of PyTorch, a prominent deep-learning framework, and Gradio, a user-friendly interface library. The system is intended to be scalable, efficient, and simple to deploy, making it appropriate for a variety of applications, including social media platforms, content moderation services, and digital forensics.

By building this deepfake identification method, we hope to contribute to ongoing efforts to prevent misinformation and protect individuals from the possible exploitation of deepfake technology. The use of explainability approaches improves the system's reliability and trustworthiness, paving the way for a more secure and transparent digital world.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION TO DEEP FAKE

1.1 Synopsis

Deep fake is a technique for human image synthesis based on neural networktools like GAN (Generative Adversarial Network) or Auto Encoders etc. These tools super impose target images onto source images using a deep learning technique and create a realistic looking deep fake image. These deep-fake image are so real that it becomes impossible to spot difference by the naked eyes. In this work, we describe a new deep learning-based method that can effectively distinguish Al-generated fake images from real images. We are using the limitation of the deep fake creation tools as a powerful wayto distinguish between the pristine and deep fake images. During the creationof the deep fake the current deep fake creation tools leaves some distinguishable artifacts in the frames which may not be visible to the humanbeing, but the trained neural networks can spot the changes. Deepfake creation tools leave distinctive artefacts in the resulting Deep Fake images, and we show that they can be effectively captured by Res-Next Convolution Neural Networks. Our system uses a Res-Next Convolution Neural Networksto extract frame-level features. These features are then used to train a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) to classify whether the image is subject to any kind of manipulation or not, ie, whether the image is deep fake or real image. We proposed to evaluate our method against a large set of deep fake images collected from multiplewebsites. We are tried to make the deep fake detection model perform better on real time data. To achieve this, we trained our model on combination of available datasets. So that our model can learn the features from different kind of images. We extracted an adequate number of videos from Face- Forensic++ [1] etc.

1.2 Project Idea

In the world of ever-growing social media platforms, Deepfakes are considered as the major threat of the Al. There are many Scenarios where these realistic face swapped deepfakes are used to create political distress, fake terrorism events, blackmail peoples are easily envisioned. It becomes very important to spot the difference between the deepfake and pristineimage. We are using Al to fight Al Deepfakes are created using tools like MTCNN [11] and ResNext [12], which using pre-trained neural networks like GAN or Auto encoders for these deepfakes creation. Our method uses a LSTM based artificial neural network to process the sequential temporal analysis of the image and pre-trained Res-Next CNN to extract the frame level features. ResNext Convolution neural network extracts the frame-level features, and these features are further used to train the Long Short-Term Memory based artificial Recurrent Neural Network to classify the image as Deepfake or real. To emulate the real time scenarios and make the model perform better on real time data, we trained our method with large amount ofbalanced and combination of various available dataset like FaceForensic++ [1], Deepfake detection challenge [2]. Further to make the ready to use for the customers, we have developed a front-end application where the user the user will upload the image. The image will be processed by the model and the output will be rendered back to the user with the classification of the image as deepfake or real and confidence of the model.

1.3 Motivation of the Project

The increasing sophistication of mobile camera technology and the ever- growing reach of social media and media sharing portals have made the creation and propagation of digital videos more convenient than ever before. Deep learning has given rise to technologies that would have been thought impossible only a handful of years ago. Modern generative models are one example of these, capable of synthesizing hyper realistic images, speech, music, and even video. These models have found use in a wide variety of applications, including making the world more accessible through text-to-speech, and helping generate training data for medical imaging. Like any trans-formative technology, this has created new challenges. So- called "deepfakes" produced by deep generative models that can manipulate video and audio clips. Since their first appearance in late 2017, many open-source deepfake generation methods and tools have emerged now, leading to a growing number of synthesized media clips. While many are likely intended to be humorous, others could be harmful to individuals and society. Until recently,the number of fake images 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. Just imagine a deep fake of our prime minister declaring war against neighboring countries, or a Deep fake of reputed celebrity abusing the fans. These types of the deep fakes will be terrible, and lead to threatening, misleading of common people. To overcome such a situation, Deep fake detection is very important. So, we describe a new deep learning- based method that can effectively distinguish Al generated fake images(Deep Fake Images) from real images. 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.

1.4 Problem Statement

Convincing manipulations of digital images and videos have been demonstrated for several decades using visual effects, 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. These so-called Al- synthesized media (popularly referred to as deep fakes). Creating the Deep Fakes using the Artificially intelligent tools are simple task But, when it comes to detection of these Deep Fakes, it is major challenge. Already in the history there are many examples where the deepfakes are used as powerful way to create political tension [14], fake terrorism events, 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 astep forward in detecting the deep fakes using LSTM based artificial Neural network.

1.5Objectives

1. Our project aims at discovering the distorted truth of the deep fakes.
2. Our project will reduce the Abuses' and misleading of the common peopleon the world wide web.
3. Our project will distinguish and classify the image as deepfake or pristine.
4. Provide an easy-to-use system for used to upload the image anddistinguish whether the image is real or fake.

1.6 Statement of Scope

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 image 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. 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 2
LITERATURE REVIEW

Face Warping Artifacts proposed a method to detect artifacts by comparing the generated face areas and their surrounding regions using a dedicated Convolutional Neural Network (CNN) model. The approach aimed to identify two types of face artifacts. The method is based on the observation that content deepfake algorithms can only generate images of limited resolutions, which then need to be further transformed to match the faces to be replaced in the source videos. However, their method did not consider the temporal analysis of the frames. Additionally, they used random noise in the training phase, which may not be the best option. Although the model performed well on their dataset, it may not perform as well on real-time data due to the noise introduced during training. To address these limitations, the proposed method should be trained on noiseless and real-time datasets.

Recurrent Neural Network (RNN) for deepfake detection used an approach that combines RNN for sequential processing of the frames along with an ImageNet pre-trained model. RNNs are well-suited for processing sequential data, such as video frames, as they can capture the temporal dependencies between frames. By incorporating an ImageNet pre-trained model, the method can leverage the knowledge gained from training on a large-scale image dataset, which can improve the performance of the deepfake detection task. The combination of RNN and pre-trained models allows for a more comprehensive analysis of the video frames, considering both the temporal and spatial features.

Limitations of Existing Methods

The existing methods have several limitations that need to be addressed:

- *Limited consideration of temporal information:* Some methods, such as Face Warping Artifacts, do not fully utilize the temporal information present in video frames. Deepfake detection can benefit from analyzing the consistency of facial features across multiple frames.

- *Reliance on pre-trained models:* While pre-trained models can provide valuable knowledge, they may not be optimized for the specific task of deepfake detection.

Developing methods to learn relevant features directly from the data can improve performance.

- *Lack of robustness to real-world scenarios:* Many methods are evaluated on datasets that may not accurately represent the challenges encountered in real-world scenarios, such as varying video quality, occlusions, and diverse facial expressions. Developing methods that are robust to these challenges is crucial for practical applications.

- *Limited generalisation:* Some methods may perform well on specific datasets but struggle to generalise to other datasets or real-time data. Designing methods that can generalise well across different datasets and scenarios is essential for their widespread adoption.

Proposed Approach

To address the limitations of existing methods and improve deepfake detection performance, the proposed approach combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) while incorporating temporal information and leveraging the advantages of transfer learning.

The proposed method consists of two main components:

- *Temporal Feature Extraction:* A CNN-based model is used to extract spatial features from individual frames. These features are then fed into an RNN, which processes the sequence of frames and captures the temporal dependencies between them. By considering the temporal information, the method can detect inconsistencies in facial features across multiple frames, which is crucial for identifying deepfakes.

- *Transfer Learning:* The CNN-based model is initialized with weights from a pre-trained model, such as VGG-16 or ResNet, which has been trained on a large-scale image dataset like ImageNet. This transfer learning approach allows the model to leverage the knowledge gained from the pre-trained model and adapt it to the specific task of deepfake detection. The RNN component is trained from scratch to learn the temporal patterns specific to the deepfake detection task.

By combining the strengths of CNNs and RNNs, the proposed method can effectively capture both spatial and temporal features from the video frames. The transfer learning approach helps to overcome the limitations of relying solely on pre-trained models and improves the generalization capabilities of the method.

To ensure robustness to real-world scenarios, the proposed method is trained on a diverse dataset that includes videos with varying quality, occlusions, and facial expressions. This diversity helps the model learn to handle the challenges encountered in practical applications.

CHAPTER 3
FEASIBILITY STUDY AND REQUIREMENT ANALYSIS

3.1 **Feasibility Study**

A feasibility study considers various constraints within which the system should be implemented and operated. In this stage the resource needed for implementation such as computing equipment, manpower and costs are estimated. The estimated are compared with available resources and a cost benefit analysis of the system is made.

The main objectives of the feasibility study are to determine whether theproject would be feasible in terms of the following categories:

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Schedule feasibility

3.1.1 Technical Feasibility

Since the android application uses software technologies and tools which are freely available and technical skills required can be easily manageable but requires a normal computing and the system server must be adequate and manageable in future. So, it is found that the hardware and software meet the need of the system. So, it's clear that the proposed project is technically feasible.

3.1.2 Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would gather from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses.

3.1.3 Operational Feasibility

Since the android application is interactive and data driven, the user need to be only a bit familiar with the software system backed with graphical explanations, which can easily be understood faster in timewith usage.

3.1.4 Schedule Feasibility

The dateline of a software system can be easily estimated if the properteam and achievable goals are formed.

3.2 Functional Requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification. The following are the functional requirements of our project:

3.2.1.1 Datasets (FaceForencis, vggFace2, DeepFake Detection Challenge(DFDC) etc)

3.2.1.2 Algorithmic Development & Architecture (MTCNN, ResNext..)

3.2.1.3 Model Training (Trained Model)

3.2.1.4 Security & Privacy

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

- The software should be efficiently designed to give reliable recognition of fake images and so that it can be used for more pragmatic purpose.
- The design is versatile and user-friendly.
- The application is fast, reliable and time saving.
- The system has universal adaptations.
- The system is compatible with future upgradation and easy integration.

3.3.2 Safety Requirements

- The Data integrity is preserved. Once the image is uploaded to the system. It is only processed by the algorithm. The videos are kept secured from the human interventions, as the uploaded image is not able for human manipulation.
- To extent the safety of the image uploaded by the user will be deleted after 30 min from the server.

3.3.3 Security Requirements

While uploading the image, the image will be encrypted using acertain symmetric encryption algorithm. On server also the image isin encrypted format only. The image is only decrypted from preprocessing till we get the output. After getting the output the imageis again encrypted.

CHAPTER 4
SYSTEM DESIGN

4.1 Use Case View

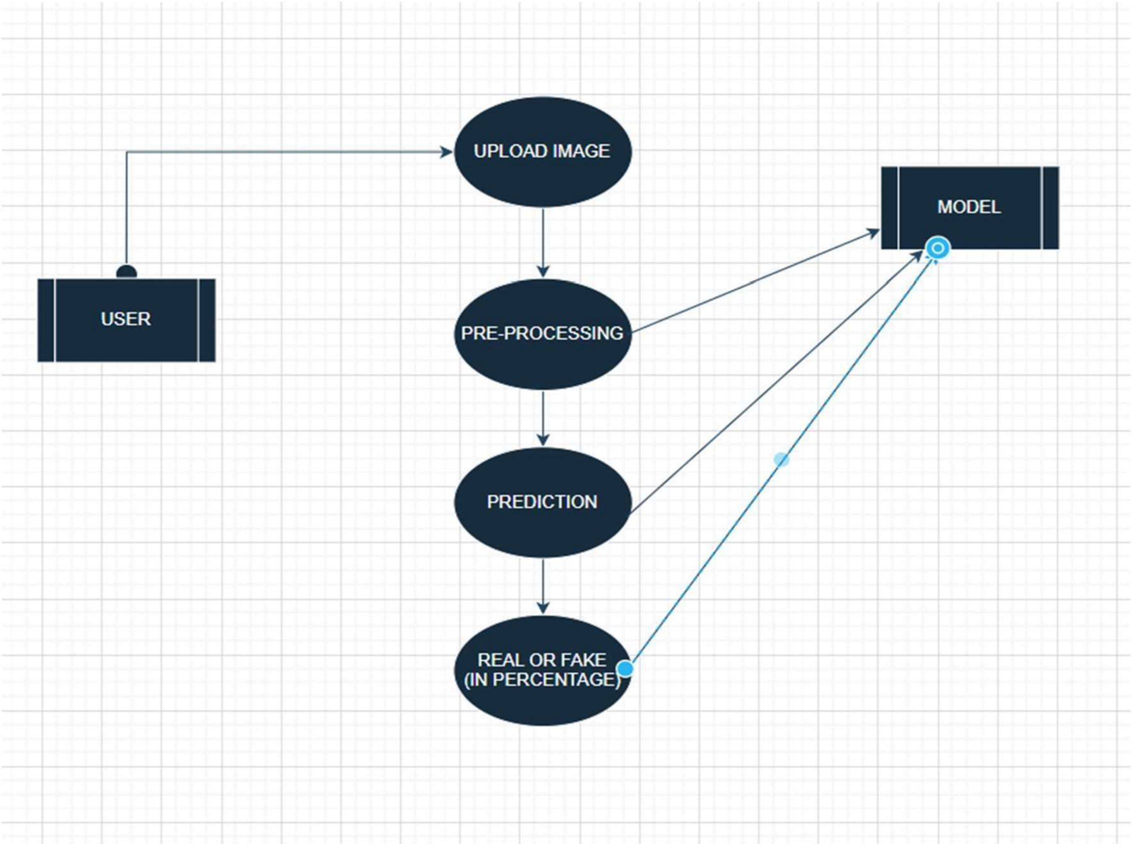


Figure 4.1: Use Case View

4.2DATA FLOW DIAGRAM

4.21 DFD Level 0

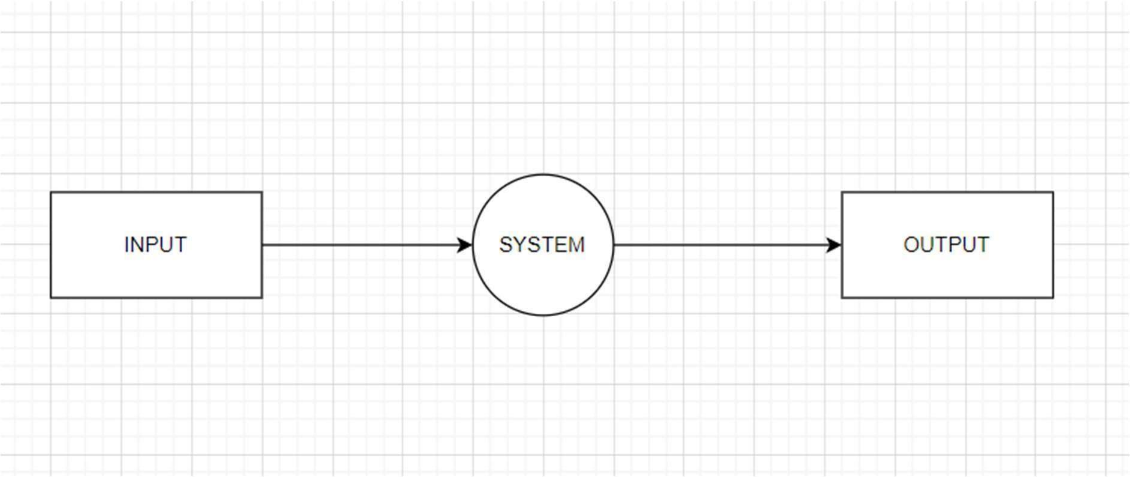


Figure 4.2: DFD Level 0

DFD level-0 indicates the basic flow of data in the system. In this System Input is given equal importance as that for Output.

Input: Here input to the system is uploading Image.

System: In system it shows all the details of the Image.

Output: Output of this system is it shows the fake Image or not.

Hence, the data flow diagram indicates the visualization of system with its input and output flow.

4.2.2 DFD Level 1

DFD Level 1 gives more in and out information about of the system. Where system gives more information of the procedure taking place.

Same as shown:

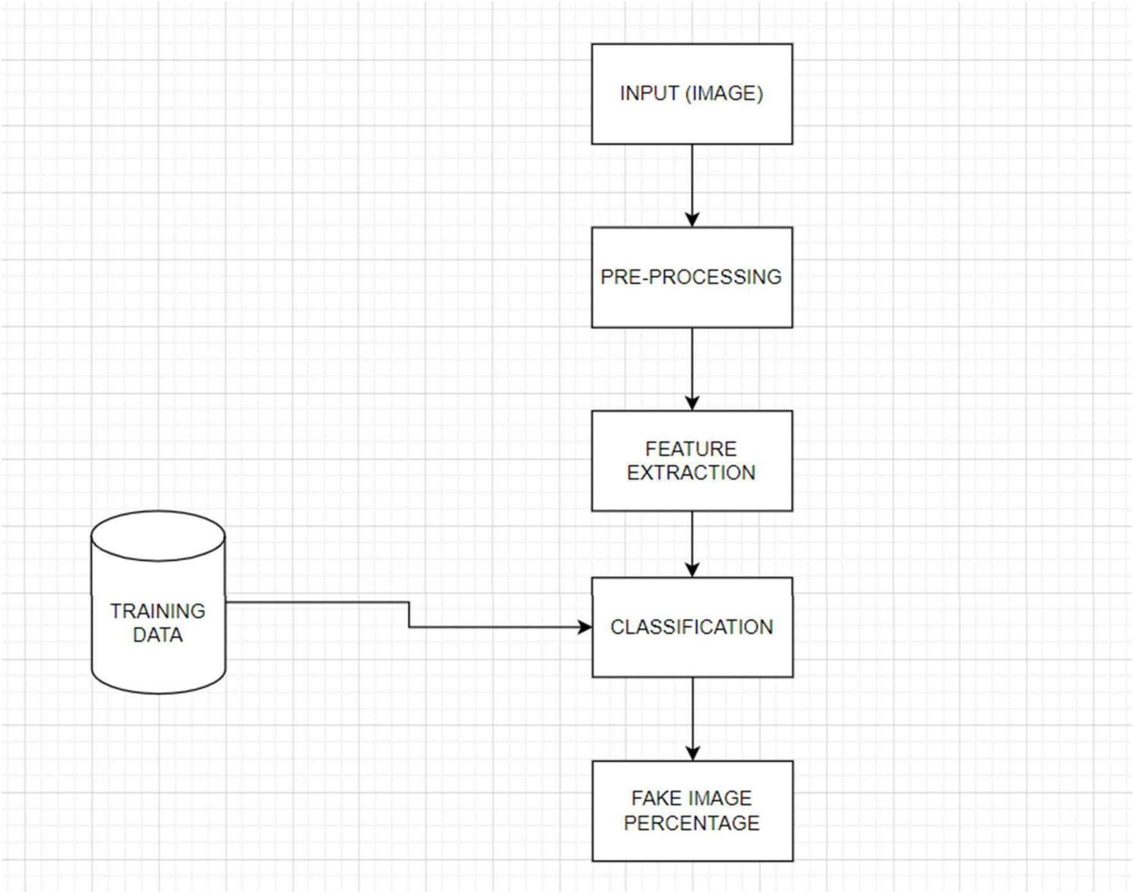


Figure 4.3 DFD: Level 1

4.2.3 DFD Level 2

DFD Level 2 enhances the functionality used by the user.

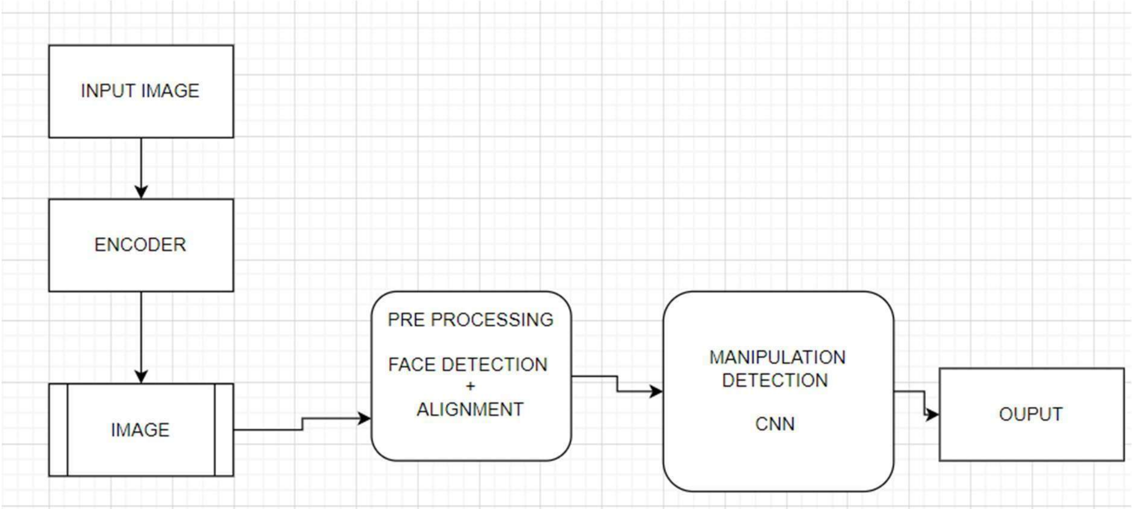


Figure 4.3: DFD Level 2

4.3 ACTIVITY DIAGRAM

4.3.1 Training Workflow

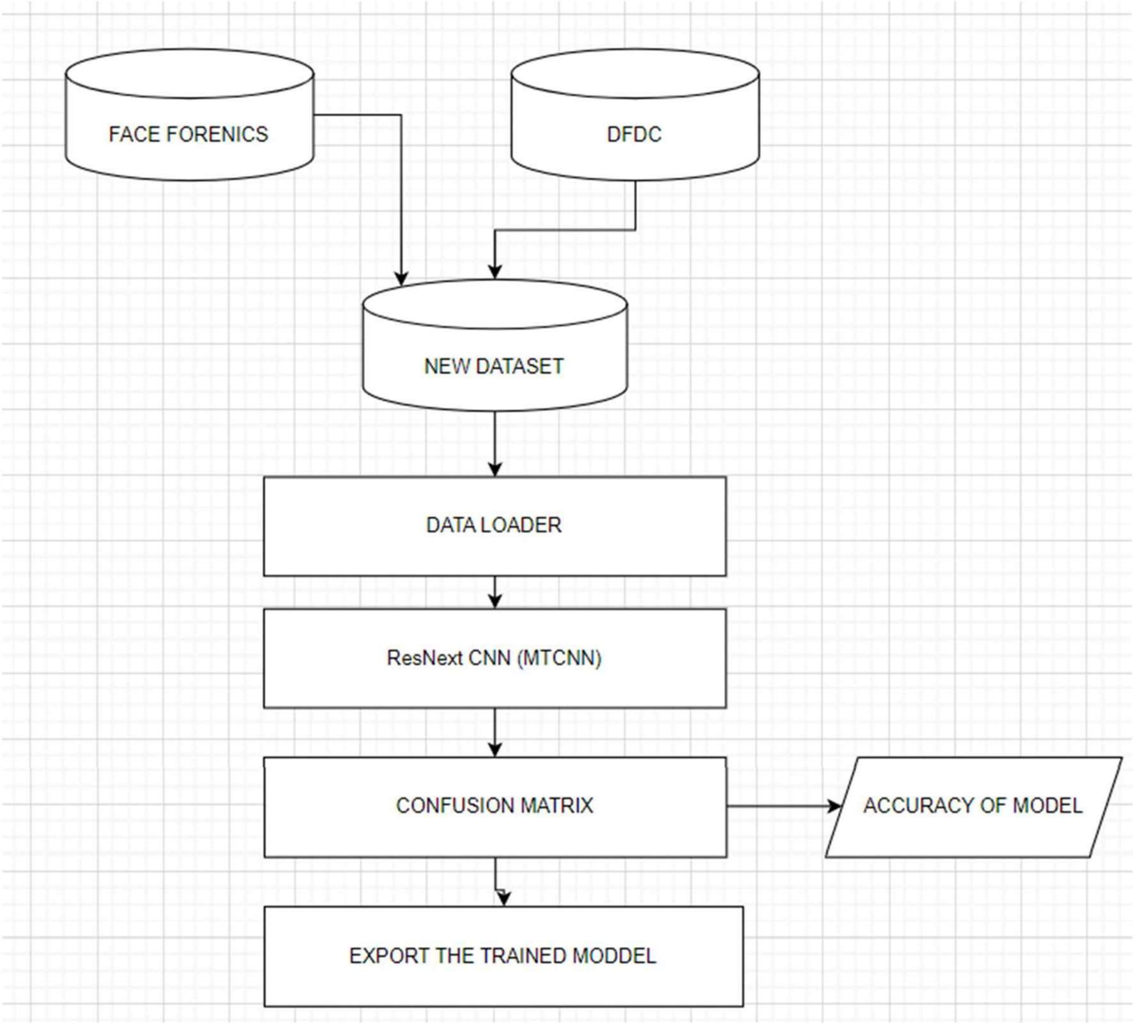


Figure 4.4: Training Workflow

4.3.2 TESTING WORKFLOW

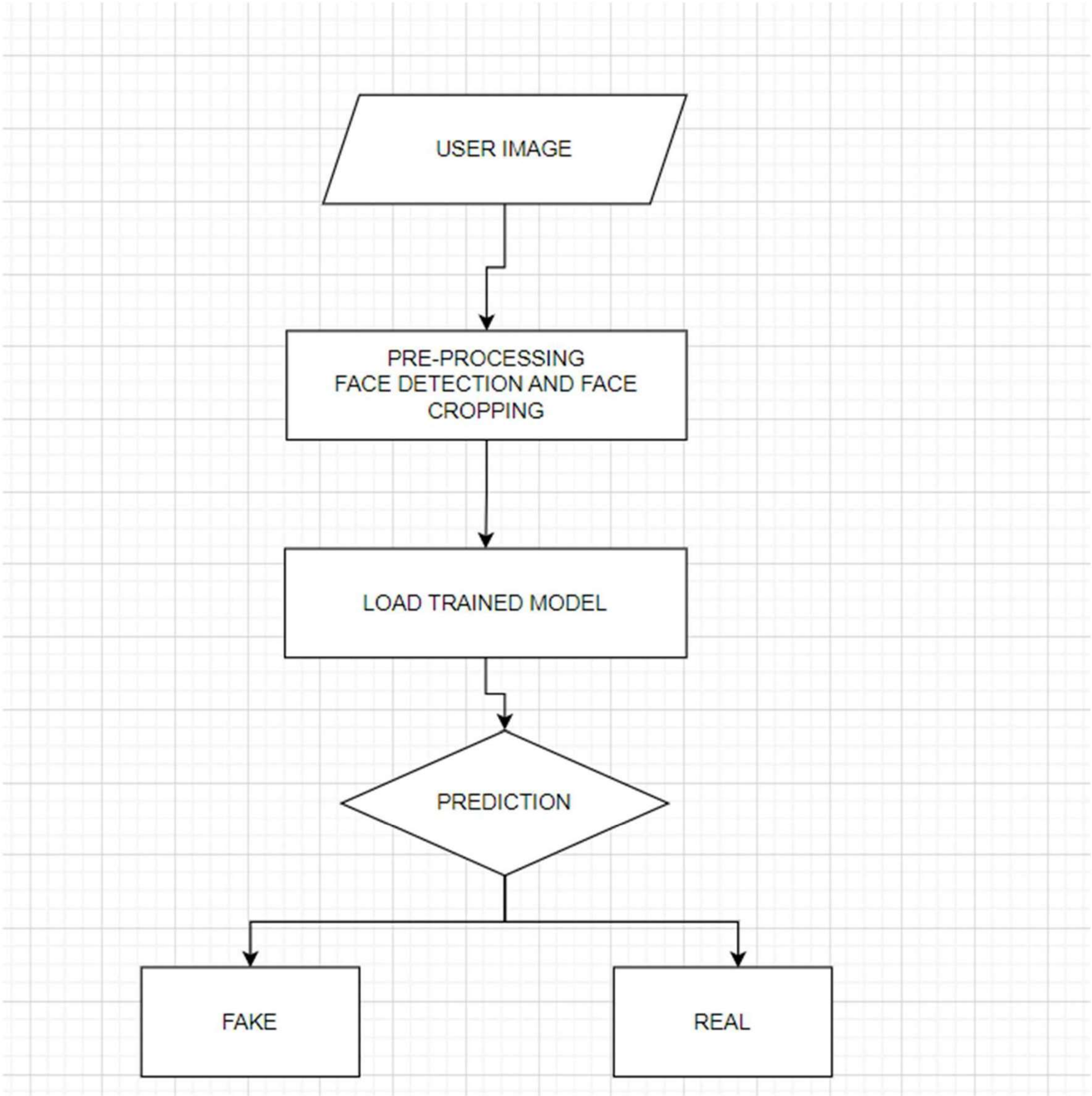


Figure 4.5: Testing Workflow

4.3 SEQUENCE DIAGRAM

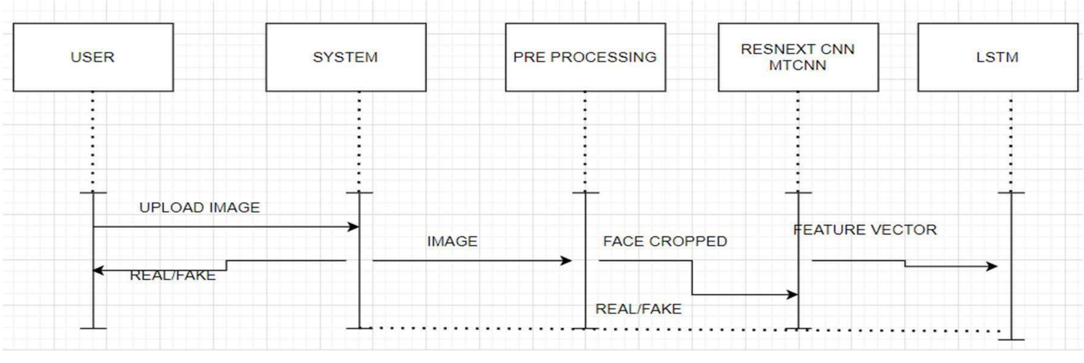


Figure 4.6: Sequence Diagram

CHAPTER 5

METHODOLOGY

5.1 System Architecture

In this system, we have trained our PyTorch deepfake detection model on equal number of real and fake images to avoid the bias in the model. The system architecture of the model is shown in the figure. In the development phase, we have taken a dataset, preprocessed the dataset and created a new processed dataset which only includes the face images.

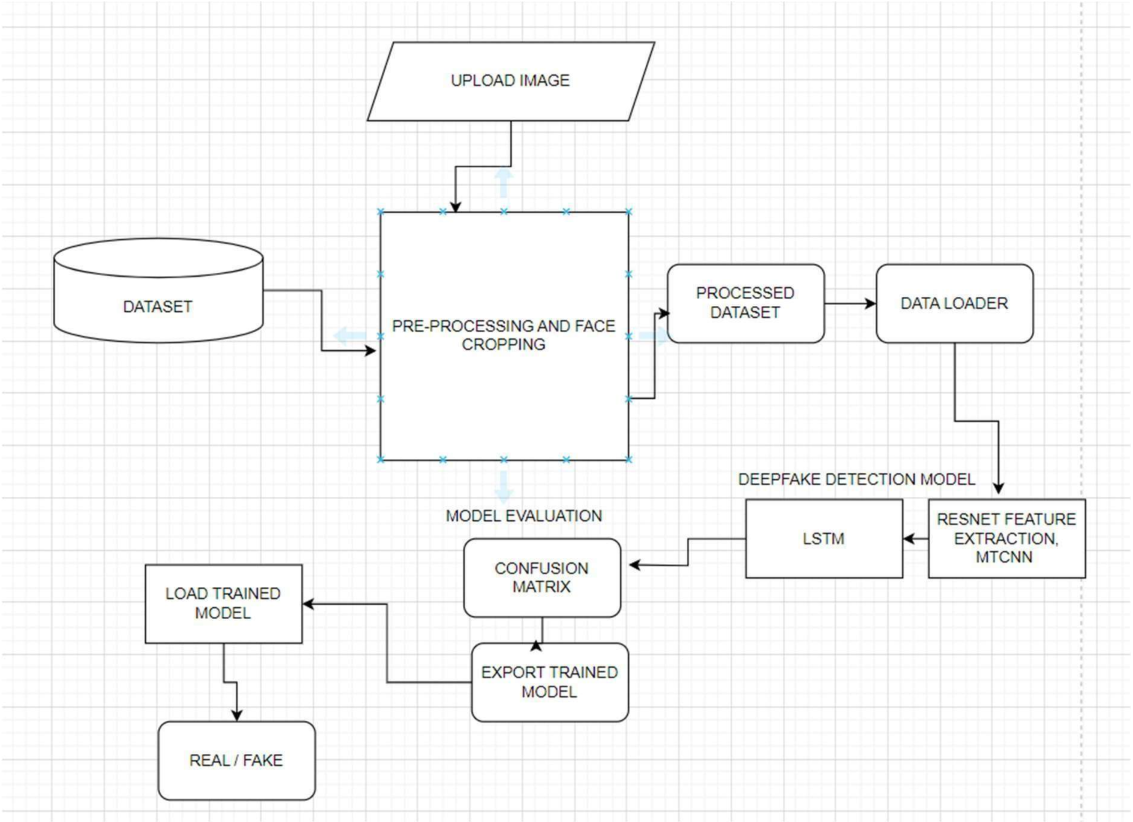


Figure 5.1: System Architecture

5.2 CREATING DEEPFAKE IMAGES

To detect the deepfake images it is very important to understand the creation process of the deepfake. The majority of the tools including the GAN and autoencoders takes a source image and target image as input. These tools detect the face in the image and replace the source face with target face on each frame. Then the replaced frames are then combined using different pre- trained models. These models also enhance the quality of image my removing the left-over traces by the deepfake creation model. Which resultin creation of a deepfake looks realistic in nature. We have also used the sameapproach to detect the deepfakes. Deepfakes created using the pretrained neural networks models are very realistic that it is almost impossible to spot the difference by the naked eyes. But the deepfakes creation tools leaves some of the traces or artifacts in the images which may not be noticeable by the naked eyes. The motive of this paper to identify these unnoticeable tracesand distinguishable artifacts of these images and classified it as deepfake or real image.

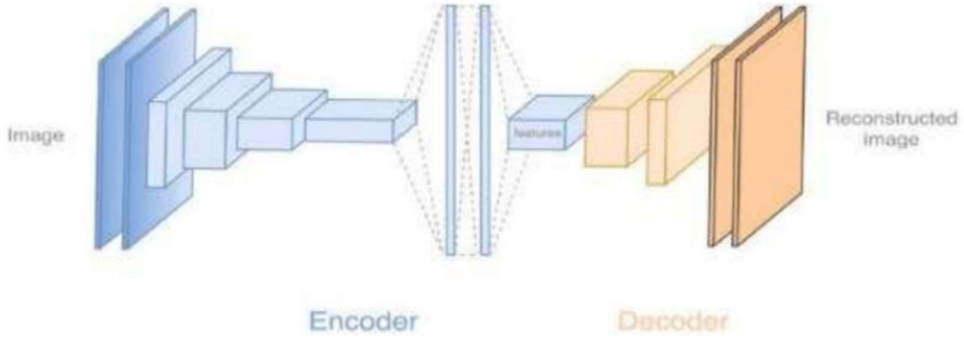


Figure 5.2: Deepfake Generation.

5.3 ARCHITECTURAL DESIGN

5.3.1 MODULE 1 Dataset Gathering

For making the model efficient for real time prediction. We have gathered the data from different available datasets like FaceForensic++(FF) [1], Deepfake detection challenge (DFDC) [2], and Celeb-DF [3]. Further we have mixed the dataset the collected datasets and created our own new dataset, to accurate and real time detection on different kind of images . To avoid the training bias ofthe model we have considered 50% Real and 50% fake images . Deep fake detection challenge (DFDC) dataset [3] consist of certain blurred images, as blurred deepfake are out of scope for this paper. We preprocessed the DFDC dataset and removed the blurred images from the dataset by running a python script. After preprocessing of the DFDC dataset, we have taken 1500 Real and 1500 Fake images from the DFDC dataset. 1000 Real and 1000 Fake videos from the FaceForensic++(FF) [1] dataset and 500 Real and 500 Fake images from the Celeb-DF [3] dataset. Which makes our total dataset consisting of 3000 Real, 3000 fake images and 6000 images in total. Figure 2 depicts the distribution of the datasets.

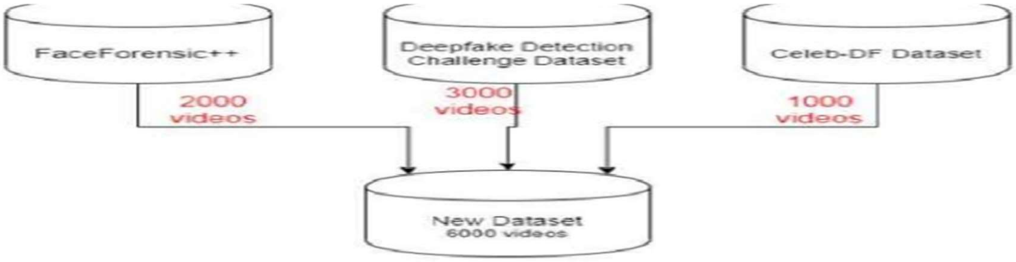
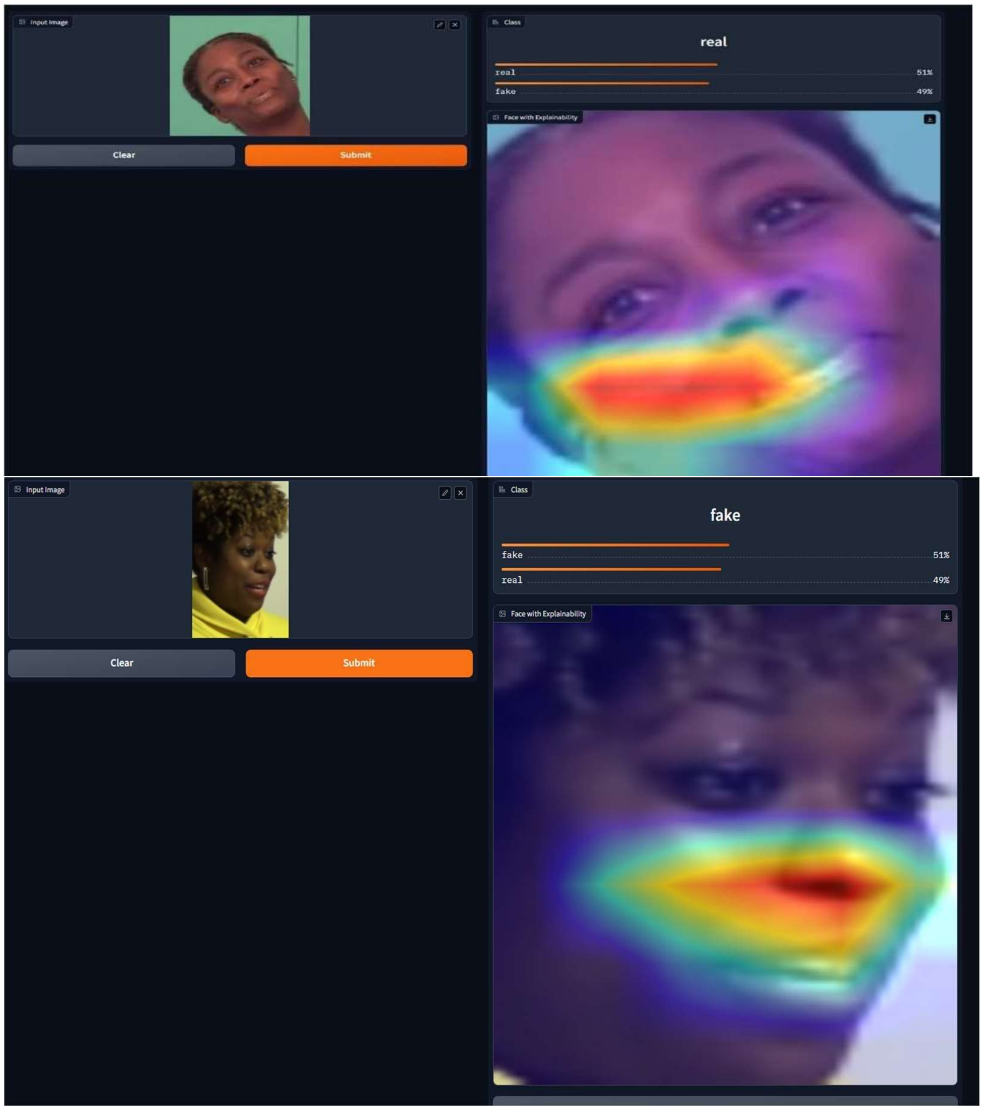


Figure 5.3: Dataset

5.3.2 MODULE 2 : PRE_PROCESSING

In this step, the videos are preprocessed and all the unrequired and noise is removed from images. Only the required portion of the imagesie., face is detected and cropped. The first steps in the preprocessing of the images are to split the video into frames. After splitting the images into frames, the face is detected in each of the frames and the frame is cropped along the face. Later the cropped frame is again converted to a new image by combining each frame of the image. The process is followed for each image which leads to creation of processed dataset containing face only image. The frame that does not contain the face is ignored while preprocessing. To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each image. So, based on our Graphic Processing Unit (GPU) computational power in experimental environment we have selected 150 frathes as the threshold value. While saving the frames to the new dataset we have only saved the first 150 frames of the video to the new video. To demonstrate the proper use of Long Short-Term Memory (LSTM) we have considered the frames in the sequential manner ie, first 150frames and not randomly. The newly created image is saved at frame rate of 30 fps and resolution of 112 x 112.



real

real 51%

fake 49%

fake

fake 51%

real 49%

Figure 5.4: Pre-processing

5.3.3 MODULE 3: DATA SPLIT:

The dataset is split into train and test dataset with a ratio of 70% train images and 30% test images. The train and test split are a balanced split i.e., 50% of the real and 50% of fake images in each split.

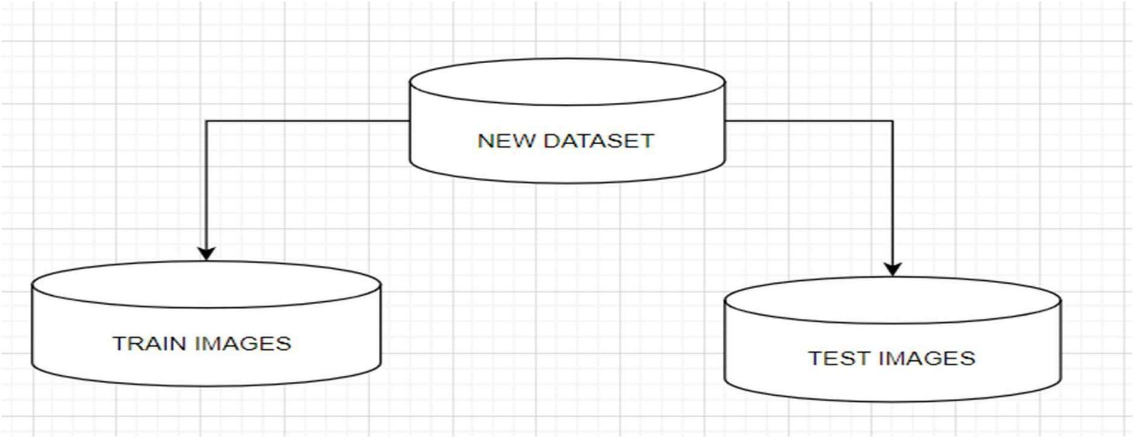


Figure 5.5: Train test split

5.4ALGORITHM DETAILS

5.4.1 Pre-processing

PyTorch is used in application like image recognition and language processing.

OpenCV is a Python open-source library for computer vision in artificial intelligence, machine learning, facial recognition, etc.

Grad-CAM works by finding the final convolutional layer in the network and then examining the gradient information flowing into that layer.

5.4.2 MODEL DETAILS

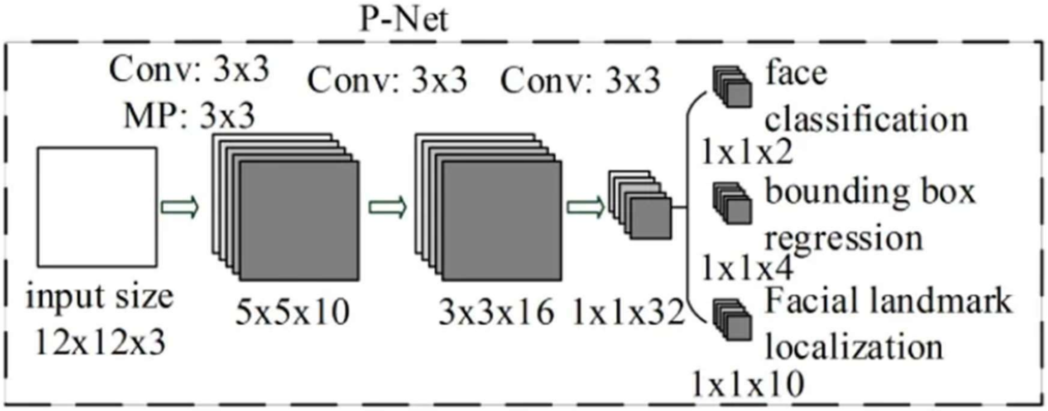
The Model consists of following Architectures:

MTCNN: Multi-task Cascaded Convolutional Networks (MTCNN) is a framework developed as a solution for both face detection andface alignment. The process consists of three stages of convolutional networks that are able to recognize faces and landmark location such as eyes, nose, and mouth.

STAGES OF MTCNN

STAGE 1

This first stage is a fully convolutional network (FCN). The differencebetween a CNN and a FCN is that a fully convolutional network does not use a dense layer as part of the architechture. This Proposal Network is used to obtain candidate windows and their bounding box regression vectors.

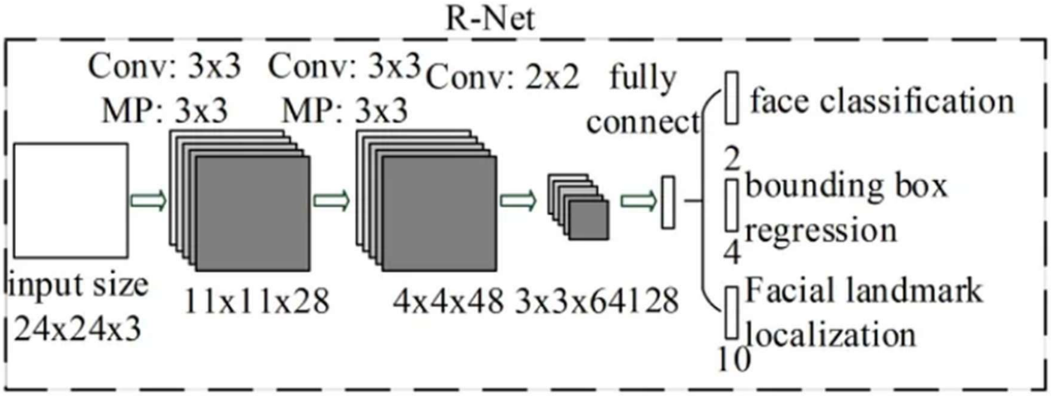


P-Net (from MTCNN paper)

STAGE 2

The Refine Network (R-Net)

All candidates from the P-Net are fed into the Refine Network. Noticethat this network is a CNN, not a FCN like the one before since there is a dense layer at the last stage of the network architecture. The R- Net further reduces the number of candidates, performs calibration with bounding box regression and employs non-maximum suppression (NMS) to merge overlapping candidates.

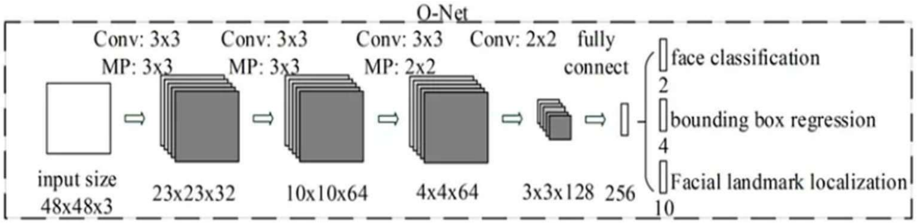


R-Net (from MTCNN paper)

STAGE 3

The Output Network (O-Net)

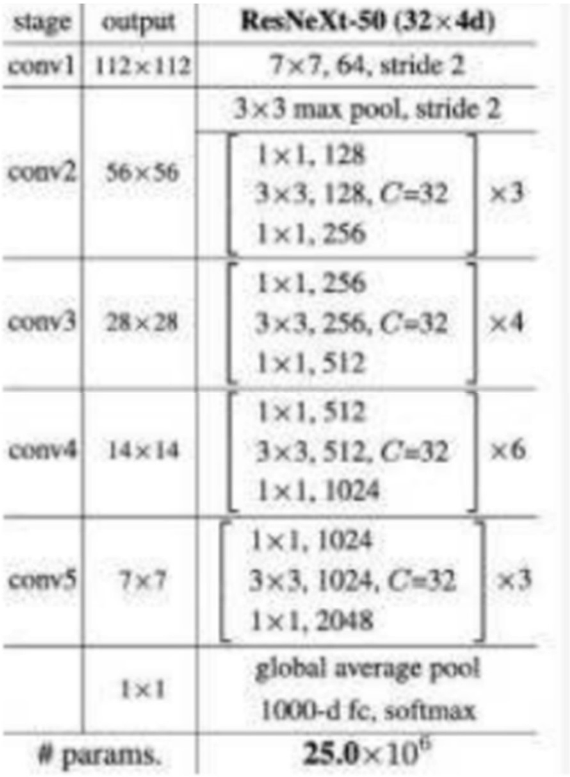
This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.



O-Net (from MTCNN paper)

ResNext CNN:

The pre-trained model of Residual Convolution Neural Network is used I model's name is resnext50 32x4d () [22]. This model consists of 50 layers and 32 x 4 dimensions. Figure shows the detailed implementation of model.



stage	output	ResNeXt-50 (32×4d)	
conv1	112×112	7×7, 64, stride 2	
conv2	56×56	3×3 max pool, stride 2	
		1×1, 128 3×3, 128, C=32 1×1, 256	×3
conv3	28×28	1×1, 256 3×3, 256, C=32 1×1, 512	×4
conv4	14×14	1×1, 512 3×3, 512, C=32 1×1, 1024	×6
conv5	7×7	1×1, 1024 3×3, 1024, C=32 1×1, 2048	×3
	1×1	global average pool 1000-d fc, softmax	
# params.		25.0×10^6	

Figure 5.6: RexNext Architecture

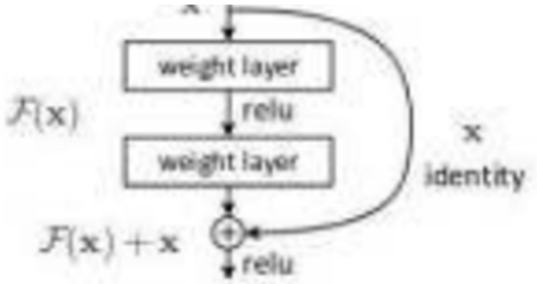


Figure 5.7: RexNext Working

5.5 TOOLS AND TECHNOLOGIES USED

5.5.1 Planning

- Open Project

5.5.2 UML Tools

- draw.io

5.5.3 Programming Language

- Python
- Java

5.5.4 Libraries

- Torch
- pyTorch
- gradcam
- Gradio
- Torch.nn.functional
- Cv2
- Face_net pytorch
- Pytorch_gradcam

CHAPTER 6
PROJECT PLAN

6.1 Cost Estimation Model (COCOMO MODEL)

Since we have small team, less-rigid requirements, long deadline we areusing the organic COCOMO model.

1. **Efforts Applied**: It defines the Amount of labor that will be required to complete a task. It is measured in person-months units.

   Effort Applied(E)=(KLOC)

   E-2.4(20.5) 105

   E57.2206PM

2. **Development Time:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months**.**

   Development Time(D) = c(E) d

   D2.5(57.2206) 0.38

   D-11.6M

6.2 RISK ANALYSIS

6.2.1 Risk identification

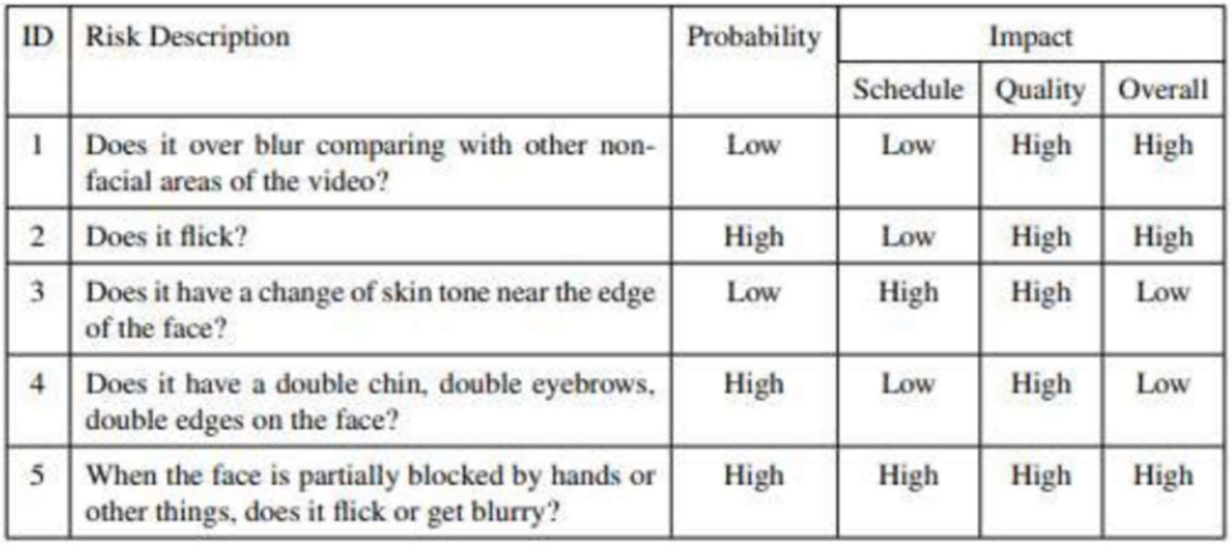
Before the training, we need to prepare thousands of images for both persons. We can take a shortcut and use a face detection library to scrape facial pictures from their images. Spend significant time to improve the quality of your facial pictures. It impacts your result significantly.

1. Remove any picture frames that contain more than one person.
2. Make sure you have an abundance of image. Extract facial pictures contain different pose, face angle, and facial expressions.
3. Some resembling of both persons may help, like similar face shape

6.2.2 Risk Analysis

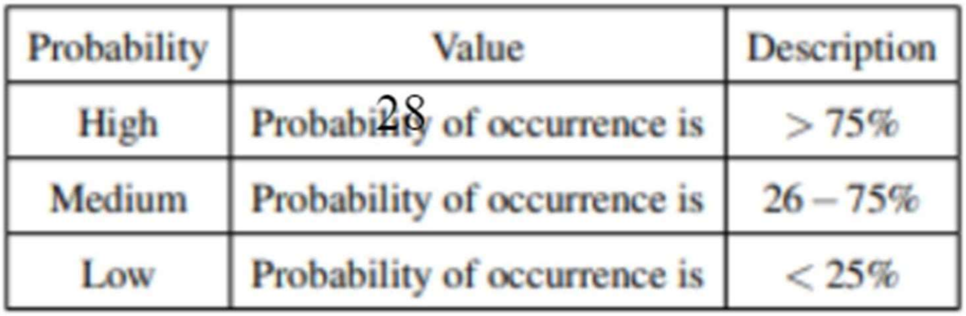
In Deepfakes, it creates a mask on the created face so it can blend inwith the target image. To further eliminate the artifacts

1. Apply a Gaussian filter to further diffuse the mask boundary area.
2. Configure the application to expand or contract the mask further.
3. Control the shape of the mask



ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Does it over blur comparing with other non-facial areas of the video?	Low	Low	High	High
2	Does it flick?	High	Low	High	High
3	Does it have a change of skin tone near the edge of the face?	Low	High	High	Low
4	Does it have a double chin, double eyebrows, double edges on the face?	High	Low	High	Low
5	When the face is partially blocked by hands or other things, does it flick or get blurry?	High	High	High	High

Table 6.1: Risk Description



Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 6.2: Risk Probability Definition

CONCLUSION

In Conclusion, Our deepfake detection project, utilizing the MTCNN architecture, Inception ResNet, and Gradio, has proven highly effective in combating the threats posed by deepfake media. The MTCNN architecture played a crucial role in precise face detection, laying a strong foundation for subsequent analysis. The Inception ResNet model excelled in capturing intricate patterns and features, enhancing the system's ability to classify deepfakes accurately. Furthermore, the incorporation of Gradio introduced an interactive and user-friendly interface, simplifying the deployment and real-time testing of our detection system.

This project highlights the power of combining sophisticated neural network architectures with practical tools to tackle modern challenges in ensuring the authenticity of digital media. Future improvements could focus on fine-tuning the models for enhanced detection precision, broadening the dataset to encompass a wider range of deepfake variations, and integrating additional user feedback mechanisms through Gradio to continuously enhance the system's performance.

In summary, our project contributes to the expanding field of research on deepfake detection and offers a practical solution for combating digital media manipulation. By leveraging cutting-edge technologies and user-friendly interfaces, we have developed a robust system that not only detects deepfakes effectively but also empowers users to engage with the detection process seamlessly.

REFERENCES

[1] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Nießner, "Face Forensics++: Learning to Detect Manipulated Facial Images" in arXiv:1901.08971.

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

[3] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi and Siwei Lyu "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics" in arXiv: 1909.12962

[4] Deepfake Video of Mark Zuckerberg Goes Viral on Eve of House A.1.

[5] 10 deepfake examples that terrified and amused the internet: https://www.creativebloq.com/features/deepfake-examples Accessed on 26 March 2020

[6] PyTorch: https://pytorch.org/

[7] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv: 1702.01983, Feb. 2017

[8] Korshunov, P., & Marcel, S. (2019). Deepfakes: A New Threat to Face Recognition? Assessment and Detection. arXiv preprint arXiv:1812.08685.

[9] Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. (2019). Deep Learning for Deepfakes Creation and Detection: A Survey. arXiv preprint arXiv:1909.11573

[10] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M.

(2019). FaceForensics++: Learning to Detect Manipulated Facial Images. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 1-11

[11] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. Information Fusion, 64, 131-148.

[12] Verdoliva, L. (2020). Media Forensics and DeepFakes: An Overview. IEEE Journal of Selected Topics in Signal Processing, 14(5), 910-932

[13] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019). Protecting World Leaders Against Deep Fakes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 38-45.

[14] Güera, D., & Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS),

[15] Jiang, H., Wang, Y., Valstar, M., & Pantic, M. (2020). STDN: A Spatial-Temporal Domain Network to Detect Manipulated Facial Videos. Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG)

[16] Li, Y., Chang, M. C., & Lyu, S. (2018). In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), 1-7

[17] Matern, F., Riess, C., & Stamminger, M. (2019). Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 83-92