**ABSTRACT**

The rapid evolution of artificial intelligence has ushered in an era where the creation and dissemination of deepfake videos pose unprecedented challenges to the authenticity of visual content. This report explores the burgeoning field of deepfake video detection, delving into the methodologies and technologies employed to distinguish genuine footage from convincingly manipulated fabrications. Beginning with an overview of the deepfake phenomenon and its societal implications, the report navigates through the key characteristics that define these synthetic videos.

As we embark on a journey to unravel the intricacies of deepfake detection, it is essential to recognize the ethical considerations that permeate this landscape. Striking a delicate balance between the imperative to guard against potential misuse and the preservation of individual privacy is an ongoing challenge that necessitates careful consideration.

**TABLE OF CONTENTS**

| **Chapter No.** | **Title** | **Page No.** |
| --- | --- | --- |
|  | **INTRODUCTION** | **01** |
|  | **PROBLEM STATEMENT** | **02** |
|  | **LITERATURE REVIEW**  3.1 Provably Correct Peephole Optimizations with Alive  3.1.1 Introduction  3.1.2 Characteristics and Implementation  3.1.3 Features  3.1.4 Evaluation  3.2 Automatic Generation of Code Analysis Tools: The CastQL Approach  3.2.1 Introduction  3.2.2 Components | **03** |
|  | **PROJECT REQUIREMENTS SPECIFICATION** |  |
|  | **SYSTEM DESIGN (detailed)** |  |
|  | **PROPOSED METHODOLOGY** |  |
|  | **IMPLEMENTATION AND PSEUDOCODE (if applicable)** |  |
|  | **RESULTS AND DISCUSSION** |  |
|  | **CONCLUSION AND FUTURE WORK** |  |
| **REFERENCES/BIBLIOGRAPHY** | |  |
| **APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS** | |  |

**CHAPTER-1**

**INTRODUCTION**

Deepfake videos have emerged as a major threat to cybersecurity and digital forensics in recent years, with the potential to cause significant harm by manipulating public opinion, spreading fake news, and damaging the reputation of individuals and organizations. To combat this threat, researchers and practitioners have developed various techniques and algorithms for deep fake detection.

Through a thorough literature review, this report finds that deep fake detection is a challenging problem that requires a multidisciplinary approach combining “computer vision, machine learning, and forensic analysis”. While current deepfake detection techniques have shown promising results, they are still limited in their ability to detect highly realistic and sophisticated deep fakes. Moreover, the effectiveness of these techniques may depend on the “specific characteristics of the deep fake and the context in which it is deployed”.

We aim to develop a DeepFake Detection model that can differentiate artificially created videos using deep learning from the real videos. This model uses a combination of deep learning techniques to process the input video and based on the data extracted, it will classify the video either as deep fake or real. This model can be useful in various fields such as “social media, news platforms, news channels'', etc., where they can verify if the video is real before it reaches the people and starts spreading virally.

Our model will have 2 components, one for extracting features and data from the frames and another one for processing the extracted data using deep learning techniques. The deep learning component will classify the video as either deepfake or not based on the data given by the data extraction layer. Our model will be robust against “image orientation”, “noise in the data” and “quality of the image”.

**CHAPTER-2**

**Problem Statement**

With the rapid advancements in deep learning and computer vision technologies, the creation and

proliferation of deepfake videos have become more accessible than ever before. Deepfakes are

“synthetic videos or images that are created using deep learning algorithms”, and they have the

potential to cause significant harm by “manipulating public opinion, spreading fake news, and

damaging the reputation of individuals and organizations”. As a result, the detection and prevention of deepfakes have become critical challenges in the field of cybersecurity and digital forensics.

Our aim is to develop a model that can effectively detect deepfake videos(only the Visual Part), even

when they are realistic and sophisticated, while minimizing the false positives and false negatives,

which are also robust against different deepfake evasion methods and is scalable to detect different

types of deepfakes like “face swapping”, “facial manipulations”, “identity swap”, “face reenactment”,“attribute manipulation”, and “entire face synthesis”, etc.

**CHAPTER-3**

**Literature Review**

[1] “An explainable deepfake detection framework on a novel unconstrained dataset” research paper proposes a novel deepfake detection framework that uses “deep learning techniques” to detect deep fake videos. The proposed framework is evaluated on a new unconstrained dataset. Dataset created and used here is “DeepFake Image-high-quality (DFIM-HQ)” dataset which contains “70000 real images and 70000 fake images” which have been created without any race,gender and age bias. It also includes various scenarios such as image orientation changes, low quality images and illumination degradations. The features learnt from the new high quality dataset include visual artifacts, such as unnatural skin tones and inconsistent head poses, while the features are learnt using Grad-CAM technique and the features will be trained on deep learning based model Meso-Net or MesoInception-Net. However, Mesonet and Meso Inception Net require a large amount of labeled data to be trained effectively. Acquiring and labeling data can be time-consuming and costly and they are not robust against small changes in the input data, they are prone to adversarial attacks. The accuracy achieved here was 94.52% for Meso-Net and 99.87% for MesoInception-Net.

[3] “Multi-attentional Deepfake Detection” research paper mentions a method that leverages attention mechanisms to detect deep fake videos. The proposed method consists of two stages: “feature extraction and classification”. The feature extraction stage uses a “deep convolutional neural network” to extract features from the input video frames, while the classification stage uses a “multi-attentional mechanism” to weigh the importance of different regions in the video frames for the final classification decision. The proposed method uses a multi-attentional mechanism to weigh the importance of different regions in the video frames for the final classification decision. The datasets used here are “FaceForensics++, CelebDF and DFDC”. Drawback of the proposed model is there is no detailed analysis of the complexity of the proposed model architecture and also it is unclear how the model performs when there are changes in the lighting conditions or camera angles in the input videos. The accuracy achieved here is 86.95% on low quality images and 96.37% on high quality images when Xception-net was used and it became 88.69% for low quality images, 97.60% for high quality images when Efficient-B4 was used.

[12] “Generalization of Forgery Detection With Meta Deepfake Detection Model” focuses on detecting deep fakes using a meta deep fake detection model.The three main data sets used were “FaceForensics++,CelebDF, DFDC”. The model can be evaluated on unseen domains" without requiring any updates, following its training on a set of source domains. To enable this, the researchers employ data preprocessing techniques and apply block shuffling transformations. Experiments on various datasets show that the proposed method outperforms existing forgery detection approaches, particularly on previously unseen forgery types. The model achieves an average precision of 0.94 on the FaceForensics++ dataset and 0.89 on the Celeb-DF dataset, outperforming the other methods by a significant margin. Further experiments on a new dataset of unseen forgery types demonstrate the model's ability to generalize to previously unseen scenarios.Overall, the proposed method provides a promising solution to the challenge of forgery detection in the age of deepfakes, demonstrating the potential of meta-learning techniques for improving generalization in this domain.

[5] “Unsupervised learning-based framework for deep fake video detection” proposes a method which is different from the previously mentioned methods. They tried “unsupervised learning” which nobody has done till now. Authors have used “FaceForensics++” dataset to train this model. The proposed model has two components: “ Photo-Response Non-Uniformity(PRNU)” and “Noise Print”. First the PRNU runs on multiple video frames for “multi classification”, later using the source ,which is obtained from the previous component there will be a “binary classification” using noiseprint mechanism. The AUC value for this model is 0.925 for original images and 0.887 for cropped images.

[9] The paper “Fighting deepfake by exposing the convolutional traces on images'' proposes an Expectation-Maximization algorithm which can be trained to detect and extract fingerprints left behind by “Generative Adversarial Networks(GANs)” called “Convolutional Traces(CT)”. These Convolutional Traces can be used to not only to classify images as deep fake or real but also help identify which GAN model might have produced it. One of the key features of the proposed model is that, since it is not a deep learning model, it can easily be run on computers with limited cpu power. Using “Expectation-Maximization” steps, the model extracts the convolutional traces from the given image which can then be passed through a simple classifier like random forest. The model displayed a good accuracy of 87.1% on average for datasets including “STYLEGAN, STYLEGAN2, FACE FORENSICS ++”, also data generated from “STARGAN, ATTGAN, GDWCT, CYCLEGAN, PROGAN, IMLE”. The model displayed the highest accuracy of 99.32% on STYLEGAN2 dataset using a 7x7 kernel. One of the major drawbacks of the model is that since EM is an iterative process, it requires a lot of time in order to extract CTs from an image making it very difficult to implement in a video deep fake identification task as the number of frames will be very high.

[15] “Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection”.The models used are “Xception Net for encoding” and then “Bi directional LSTM for deep fake detection”. This paper uses a “3-Phase architecture” where the first phase would be to use the “Dlib library” to take only the region of interest that is the face or body. The second phase would be the neutral network which will use the weights and other parameters to represent the Image input to a latent representation.This includes the Xception Net which would be responsible for filtering only the spatial information and masking the temporal Information. then the output from that would be passed onto the 2-layered bi directional LSTM Model which would then generate the output which will be passed to the next phase.The Third Phase would contain the outputLayer of the class layer. The 3rd phase will also contain the probability distributions of real and fake detections that have happened and then we have two models in the last phase so we have two loss functions which will be used one would be cross entropy between the real and fake output from neurons and other will be “KL divergence loss” for the probability distribution. The datasets used are Face Forensics ++ and Celeb DF. In the accuracy part we have 50 to 98% from the class layer output and 97 to 99% from the probability distribution of the two real and fake.

**CHAPTER-4**

**Project Requirements Specification**

**4.1 Introduction**

**4.1.1 Project Scope**

● Our project aims to develop a system that can detect deepfakes in videos with high accuracy, which is robust against image orientation changes, image quality and occlusion.

● The project scope includes features such as video analysis and classification of video.

● The target audience for the system is media companies, law enforcement agencies, and other organizations that need to detect deepfakes in videos.

**4.2 Product Perspective**

**4.2.1 Product Features**.

● The system will analyze video content to identify signs of manipulation, such as inconsistent facial expressions or lighting, and use machine learning algorithms to detect deepfakes with high accuracy.

● When a deepfake is detected, the system will trigger an alert and generate a report with details about the video and the detection results.

**4.2.2 Operating Environment**

● The deepfake detection system will require high-performance hardware, including a powerful CPU and GPU, and a large amount of memory and storage.

● The software requirements include operating systems such as Windows and Linux, and deep learning frameworks such as TensorFlow and PyTorch.

**4.2.3 General Constraints, Assumptions and Dependencies**

● The project depends on the availability of diverse and reliable data sources

● The project’s accuracy depends on the quality of the training data

● The project assumes that there will be a large number of deepfakes in the wild, and that the system will need to be updated regularly to keep up with new types of deepfakes.

● The project assumes that deepfake techniques used in the dataset are similar to those used in the real world

● The project assumes that the entire length of the video will either be real or deepfake

● The project may be constrained by limited resources, such as time and funding, and dependencies on third-party software libraries and frameworks.

**4.2.4 Risks**

● The system may produce false positives, which may lead to innocent users being falsely accused of creating or sharing deepfake

● The system may infringe on user privacy by analyzing and storing user data, leading to privacy concerns and potential legal issues

● The system may be vulnerable to adversarial attacks, where attackers may attempt to deceive the system by creating deepfakes that can evade detection

● Other risks include the potential for malicious actors to develop new types of deepfakes that the system cannot detect, and the risk of false confidence in the system's detection results.

**4.3 Functional Requirements**

● The deepfake detection system must be able to analyze videos for signs of manipulation, such as inconsistent facial expressions or lighting, and use machine learning algorithms to detect deepfakes with high accuracy.

● Data Preprocessing: The system should be able to preprocess the dataset to extract relevant features from the images or videos to train the model

● Model Training: The system should be capable of training deep learning models to detect deepfakes

● Model Testing: The system should be able to test the model on a separate dataset to evaluate its performance

● Real-Time Detection: The system should be able to perform real-time deepfake detection on images and videos as they are uploaded

● Accuracy: The system should have a high accuracy rate in detecting deepfakes and minimizing false positives

**4.4 External Interface Requirements**

**4.4.1 User Interfaces**

The deepfake detection system will have a web-based user interface that allows users to upload videos for analysis and view detection results.

**4.4.2 Hardware Requirements**

The deepfake detection system will require high-performance hardware, including a powerful CPU and GPU, and a large amount of memory and storage.

**4.4.3 Software Requirements**

The software requirements include operating systems such as Windows and Linux, and deep learning frameworks such as TensorFlow,PyTorch,DlibNumpy,Pickle and OpenCV.

**4.4.4 Communication Interfaces**

The deepfake detection system will require communication interfaces, such as APIs or network connections, to integrate with other systems or applications.

**4.5 Non-Functional Requirements**

**4.5.1 Performance Requirements**

● The deepfake detection system must be able to analyze videos quickly and accurately, with a “high detection rate” and “low false positive rate”.

● The system must also be scalable, able to handle large volumes of video content and multiple users simultaneously.

**4.5.2 Safety Requirements**

● The deepfake detection system must not cause harm or endanger users in any way.

● It must not leak user data or use any user data without their consent

**4.5.3 Security Requirements**

The deepfake detection system must ensure the confidentiality, integrity, and availability of user data, and must be designed to prevent cyber attacks.

**4.6 Other Requirements**

The deepfake detection system must comply with legal and regulatory requirements, such as data privacy laws and intellectual property rights.

**CHAPTER-5**

**System Design**

**5.1 Introduction**

This section provides an overview of the proposed system design for deepfake detection in videos, with a focus on facial visualization part .

**5.2 Current System**

Existing deepfake detection systems have limitations in detecting deepfakes that primarily focus on facial visualization. These techniques are not that effective in detecting sophisticated deepfakes. They very much rely on detecting artifacts, such as inconsistent lightning or blurriness that are introduced during the manipulation process.However, with the use of advanced machine learning algorithms, it is becoming increasingly difficult to detect these artifacts and deepfakes are becoming more realistic and sophisticated.

There is a need for a deepfake detection system that specifically focuses on the facial visualization part of the videos and uses advanced machine learning techniques to detect sophisticated deepfakes in real-time and at scale.The current system is also proven to fail when there is a change in orientation.The proposed system aims to address these limitations and provide a more effective and reliable solution for deepfake detection.

**5.3 Design Considerations**

**5.3.1 Design Goals**

● Higher accuracy even for the low quality images

● Real-time performance for deepfake videos

● Ability to detect sophisticated deepfakes

● Good model performance even when the the face orientation is bad

**5.3.2 Architecture Choices**

● Convolutional Neural Networks (CNNs): CNNs have been shown to be effective at detecting deepfakes in videos. One approach is to use a pre-trained CNN on image recognition tasks to extract features from individual frames of a video, and then use a recurrent neural network (RNN) to classify the video as real or fake based on the temporal sequence of these features.

● Two-stream CNNs: Two-stream CNNs use separate networks to analyze spatial and temporal information in videos. One stream processes individual frames, while the other analyzes motion between frames. These streams are then combined to classify the video as real or fake.

● Siamese Networks: Siamese networks are a type of neural network architecture that compares two inputs and determines whether they are similar or dissimilar. For deepfake detection, a Siamese network can be trained to compare pairs of frames from a video, and classify the video as real or fake based on the similarity of the frames.

● Gradient-weighted Class Activation Mapping(Grad-CAM): is a visualization technique used to understand which regions of an image a convolutional neural network (CNN) is focusing on when making a particular classification decision. It works by taking the gradient of the class score with respect to the feature maps of a CNN layer and then weighting these feature maps by their gradient values. This produces a heat map that highlights the regions of the input image that are most relevant to CNN's decision. Grad-CAM has been widely used in computer vision research and has applications in fields such as object detection, image segmentation, and visual question answering.

● Multi-Attentional Maps (MAM): It is a technique used to visualize the attention mechanisms in neural networks. MAM works by identifying the most relevant words or phrases in a given sentence or document, based on their relative importance to the task at hand. This is done by training a neural network to generate attention weights for each word in the input sequence, and then using these weights to create a heat map that highlights the words with the highest attention scores. MAM can be used to gain insights into how neural networks process.

**5.3.3 Constraints, Assumptions and Dependencies**

● The model assumes a uniform video data format.

● The model assumes that the entire length of the video will either be real or deepfake.

● The model’s accuracy depends on the quality of training data.

● The model cannot have a fixed size for the feature extraction kernel.

● The project depends on the availability of diverse and reliable data sources

● The project’s accuracy depends on the quality of the training data

● The project assumes that there will be a large number of deepfakes in the wild, and that the system will need to be updated regularly to keep up with new types of deepfakes.

● The project assumes that deepfake techniques used in the dataset are similar to those used in the real world

● The project assumes that the entire length of the video will either be real or deepfake

● The project may be constrained by limited resources, such as time and funding, and dependencies on third-party software libraries and frameworks.

**5.4 High Level System Design**

**5.4.1 Steps**

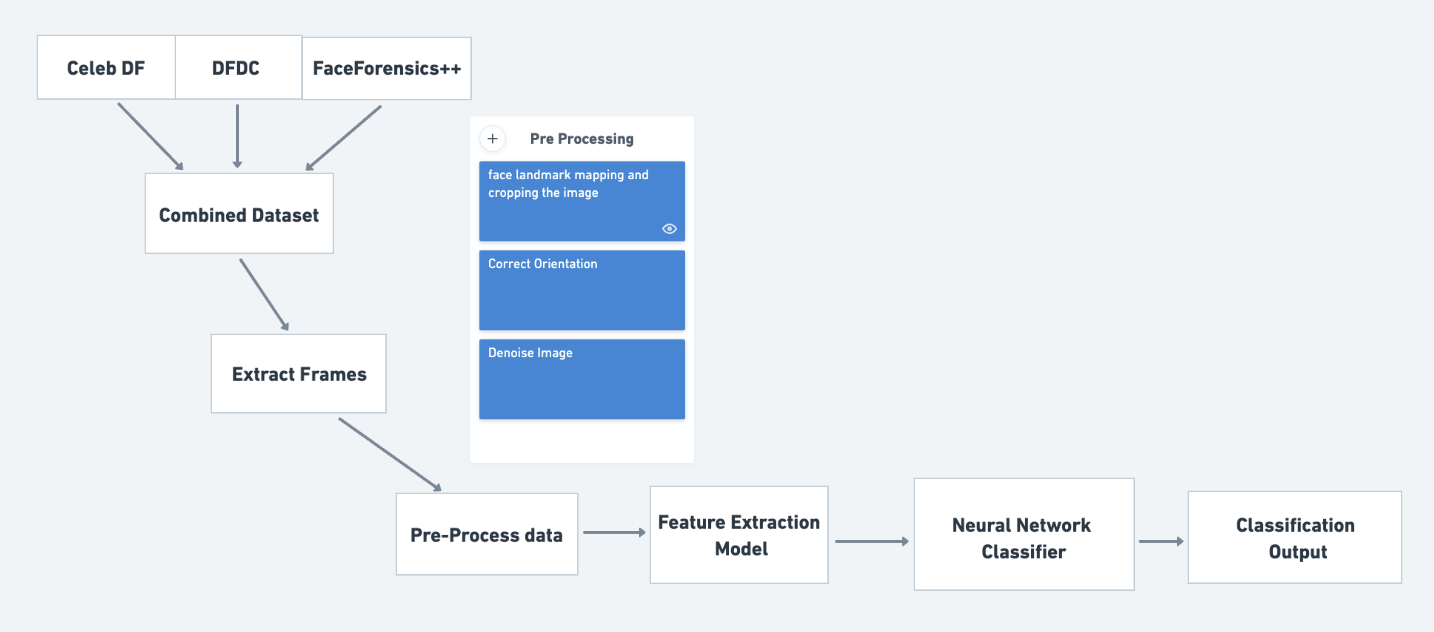
● Obtain datasets from references research

● Merge all the datasets,extract frames from videos and pre process it

● Pass the processed data through feature extraction model

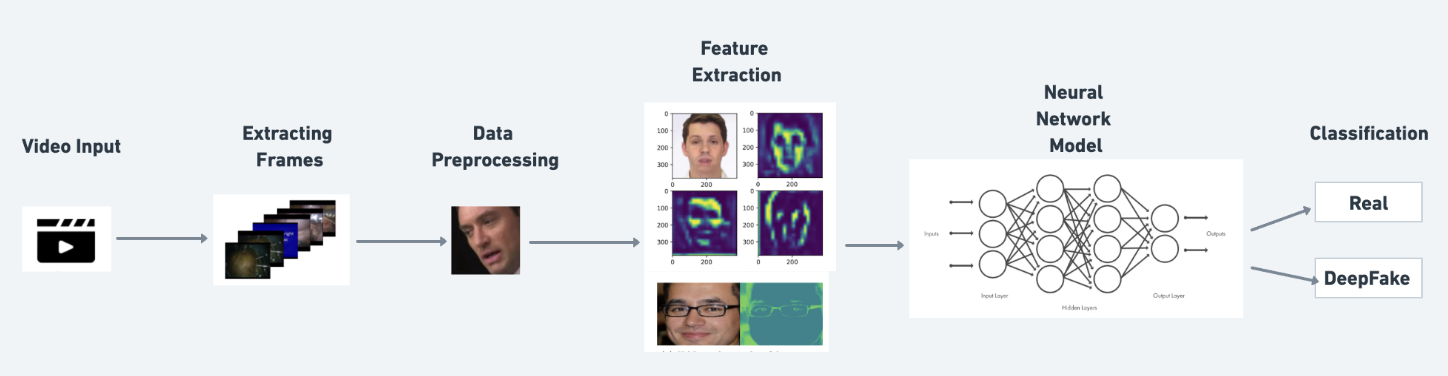
● Pass the above obtained data to deep learning model

● Predict the output

**5.4.2 Block Diagram **

| Fig1. Block Diagram |
| --- |

**5.4.3 High Level Design Diagram**

****

| Fig2. High Level Design Diagram |
| --- |

**5.5 Design Details**

**5.5.1 Novelty**

● Effective on low quality images also

● Robust against bad orientation of face

● Works even when occlusion occurs

**5.5.2 Innovativeness**

The project can be considered innovative in its approach to detecting deepfakes in real-time or in its use of novel deep learning architectures.

**5.5.3 Interoperability**

The deepfake detection project can work with different forms of video files, such as mp4,mov,mkv,etc,. It will also be compatible with different platforms, such as mobile devices or cloud-based services. Ensuring interoperability may involve designing an appropriate data schema, creating adapters for different media formats, or using standardized APIs.

**5.5.4 Performance**

The performance of our model will be the same on low quality images or bad face orientations. The model will trained and optimized properly to get higher accuracies and faster detection speeds

**5.5.5 Security**

The deepfake detection project incorporates advanced security measures, such as using secure communication protocols and encryption, to prevent data leakage and protect against attacks. It also uses adversarial training and other techniques to make the model more robust to attacks and manipulations.

**5.5.6 Reliability**

The deepfake detection project is highly reliable, achieving consistent levels of accuracy across different types of deepfakes and media file formats. It is also designed to be easily scalable and adaptable to different contexts, ensuring reliability in production environments.

**5.5.7 Maintainability**

The deepfake detection project is designed with maintainability in mind, using modularized code and automated testing to make it easy to manage code changes and updates. It also incorporates version control tools and other software development best practices to ensure long-term maintainability.

**5.5.8 Portability**

The deepfake detection project is designed to be portable across different hardware and software environments.

**5.5.9 Legacy to Modernization**

The deepfake detection project is designed to transition from older, rule-based methods of detecting manipulated media to newer, machine learning-based methods. It uses algorithms to automatically generate labeled data and collects a large dataset of manipulated and unmanipulated media files to improve detection accuracy.

**5.5.10 Reusability**

The deepfake detection project is designed to be reusable across different applications and contexts, using standard APIs to make it easy to integrate the system with other software tools and platforms. It also incorporates custom adapters to handle different media formats and ensure compatibility with different applications.

**5.5.11 Application Compatibility**

The deepfake detection project is designed to be compatible with different types of applications, such as social media platforms and video editing software. It incorporates custom adapters to handle different media formats.

**CHAPTER-6**

**Proposed Methodology**

**6.1 Introduction**

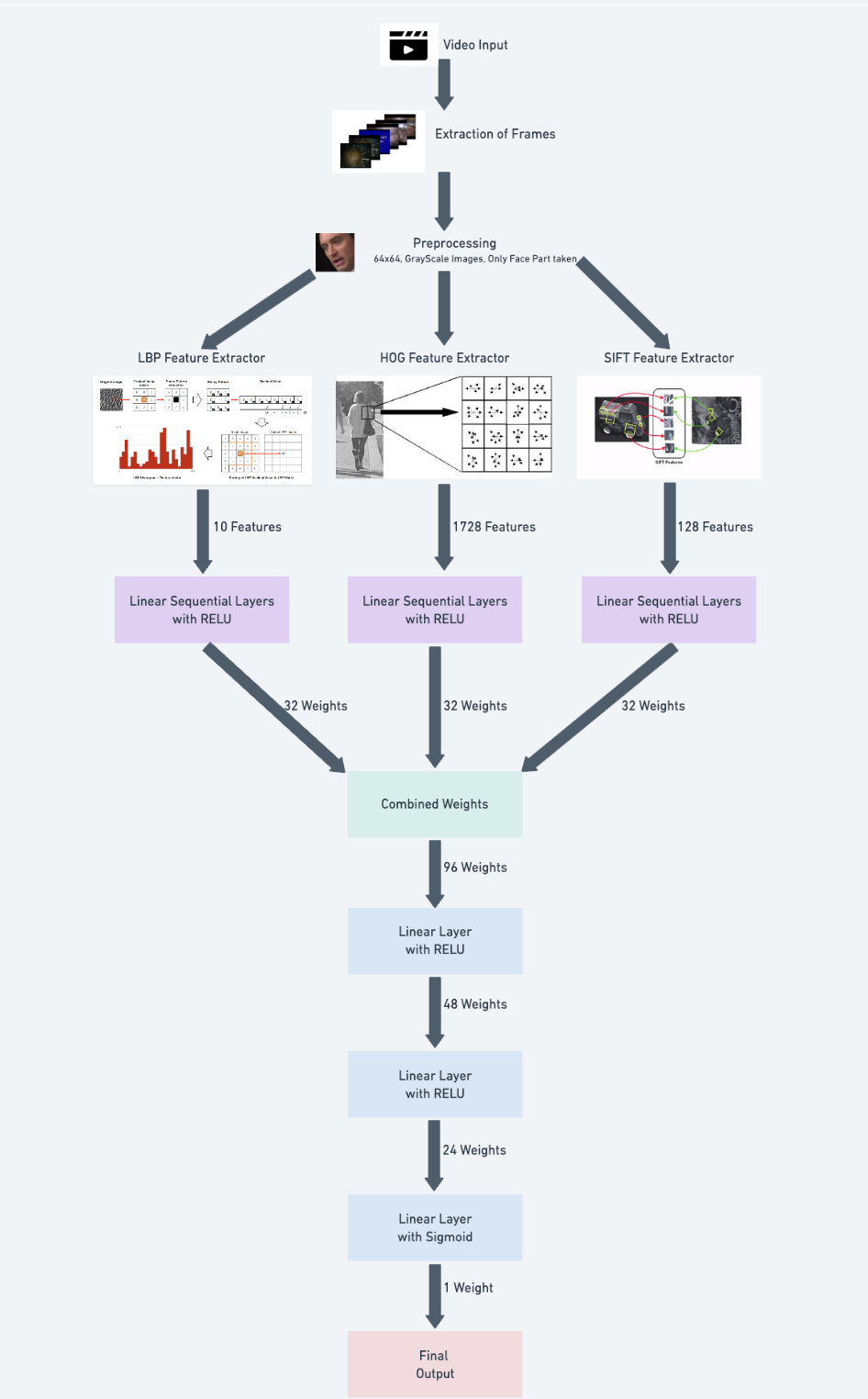
The intent of the paper is to concentrate more on classifying the video as deepfake or not, even if the video has faces with different orientations (person seeing left or right,etc). Most of the surveys have mentioned in their limitations that the model would loose the efficiency if there is orientation factor in the image ,So we have tried to reduce that dependency for better results in Deepfake classification. Main part of our method lies in preprocessing and feature extraction unlike other existing models.

**6.2 Data Collection**

To ensure the quality and prediction accuracy of deep learning models, proper dataset preparation is crucial. In our research, we leverage 3 top datasets “DFDC, Celeb-DF and FaceForensics++”, which comprise both authentic and manipulated videos, along with corresponding authenticity labels. These videos are combined and then sampled to extract individual frames. Unlike many other research work, our model has been trained on merged dataset, so that it won’t overfit for a particular kind of dataset and to ensure that it considers different deepfake evasion methods while training.

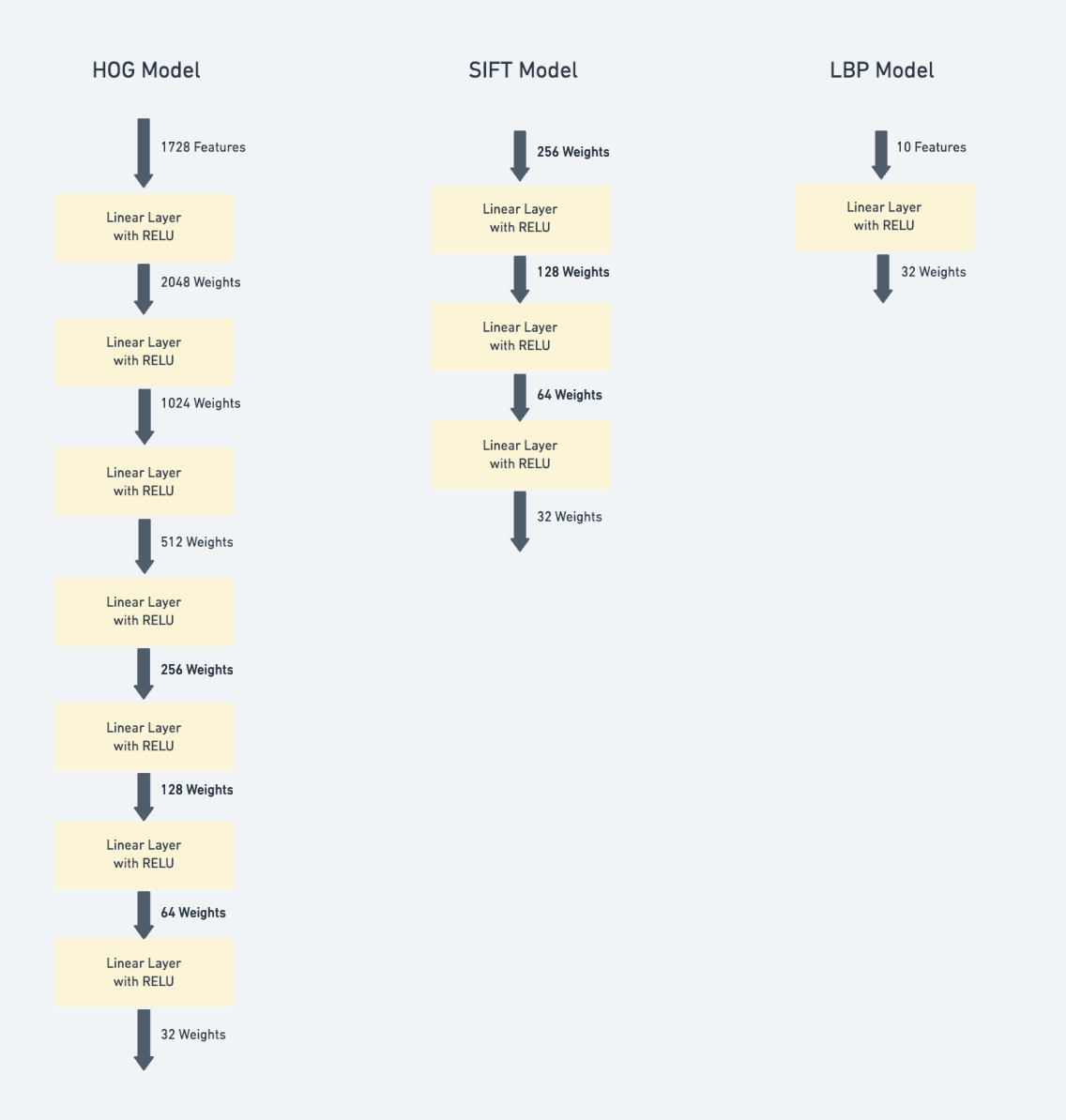
**6.3 Pipeline**

Pipeline of our model is as shown below:

****

| Fig3. Pipeline Diagram |
| --- |

.

****

| Fig4. Neural Network Architecture |
| --- |

**6.3.1 Frames Extraction**

The Architecture takes in input in image or frame format, thus this stage is used to convert the input Video into frames. This Stage uses the “Dlib” Library of python, for frames extraction.for few videos the number of frames is very high and for few its small number so we have also implemented a Helper function to select almost equal number of frames per video and to reduce the total number of frames per video.

**6.3.2 Data Preprocessing**

The Surveys tell that the image quality which is going in as input must be improved, and some papers have also put this in their limitation section, therefore the architecture uses a preprocessing stage which will enhance the image quality and also do down-sampling for regularization.The steps involved here are converting the color image to gray scale, down-sampling the pixels to (64,64).

**6.3.3 Feature Extraction**

We have introduced this particular stage to answer our needs on the orientation of face in frames.A deep neural Network has a characteristic of doing feature Extraction implicitly, but from the survey this type of implementation would have the above mentioned research gap, so We are getting arrays which will have Features of the frame and then pass this instead of the image. Experimentally we understood that combining more than one feature extraction would give a better result, hence we used “HOG(Histogram of Oriented Gradients), LBP(Local Binary Patterns), SIFT(Scale-Invariant Feature Transform)”.

**6.3.3.1 Histogram of Oriented Gradients(HOG)**

HOG is a feature extraction technique used in computer vision for object detection and image classification.This algorithm computes gradient information from the image to create histograms of oriented gradients in local regions.The key parameters for this method include the size of the cell, the size of the block, and the number of bins in the histograms. It computes a feature vector for each frame based on these histograms.By using this as an explicit feature extraction technique, we can capture the image's gradient information and create a feature vector that represents the distribution of gradients across the frame. This feature vector combines with the other 2 and is used as input to our model for deepfake detection.

**6.3.3.2 Local Binary Patterns(LBP)**

* Local Binary Pattern classifier is used in deepfake detection as a texture analysis technique that captures texture patterns within an image, encodes them into a feature vector, and then uses machine learning to distinguish real images from manipulated ones.
* Its ability to handle variations in texture and illumination makes it a valuable component in the detection of deepfake content.
* After calculating LBP codes for all pixels, a histogram is constructed. Each unique LBP pattern represents a bin in the histogram, and its count (frequency) in the image serves as the value in that bin. The LBP histogram is typically used as a feature vector.
* This feature vector is invariant to changes in illumination, which makes it particularly useful for detecting inconsistencies or artifacts introduced by deepfake techniques.

**6.3.3.3 Scale-Invariant Feature Transform(SIFT)**

This is a feature extraction method which will convert the frame to a latent representation which will have information about key points of the frame.The Key points are the points which tell the characteristics of the face in the frame that is those are unambiguous set of points, the extractor will discard all the other ambiguous points.

**6.3.4 Equalizing the number of features from each feature extractor**

From the above feature extractor components we have now got 3 feature arrays. It So happens that the sizes of the feature array from the three feature extractors are different, which makes different feature extractors require different neural networks. To make it uniform after few layers we stop at (32\*1) weights, so the backtracking becomes easier too. We pass these on Linear Sequential Layers with ReLU activation function. The reason we are taking the same feature array size is that we need no biasing between features from different feature extractors.

**6.3.5 Combining weights from different layers of different feature extractors**

Now we have passed the features arrays on linear layers to three arrays of smaller size and equal size, then we will be merging all the three to form a 96\*1 matrix and then passing it to the next component in our architecture.

**6.3.5 Neural Network for the combined feature matrix**

After getting the merged feature matrix we will pass that to a neural network which will have input as the combined array and then will have a number of hidden layers and then will have the the output layer with only one output.which will give the desiered output whether the input image is a real image or a deepfake image. We have used Linear Sequential Layers with ReLU activation function except the last layer which uses Sigmoid Activation function. Optimizer that we have used here is Adam with 0.01 learning rate and the Loss function used here is Binary Cross Entropy.

**CHAPTER-7**

**Implementation and Pseudocode**

**7.1 Preprocessing**

****

**7.2 Preprocessing output**

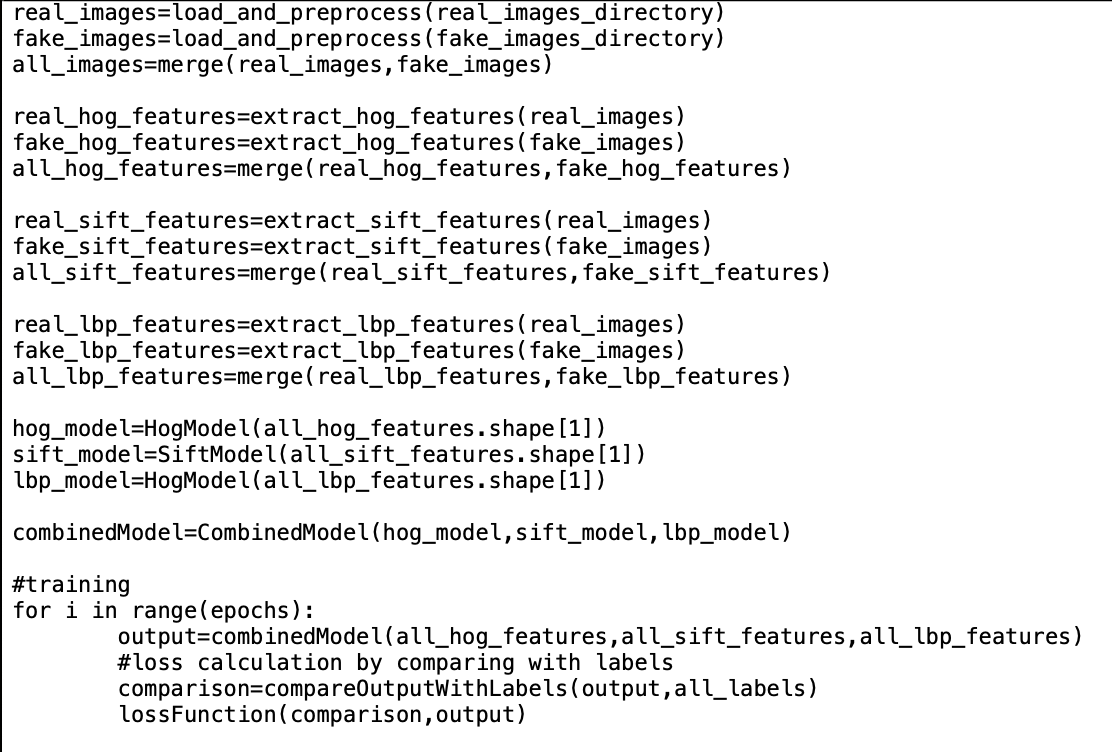
Original image:

|  |
| --- |
| Fig5. Output original image |

Cropped image:

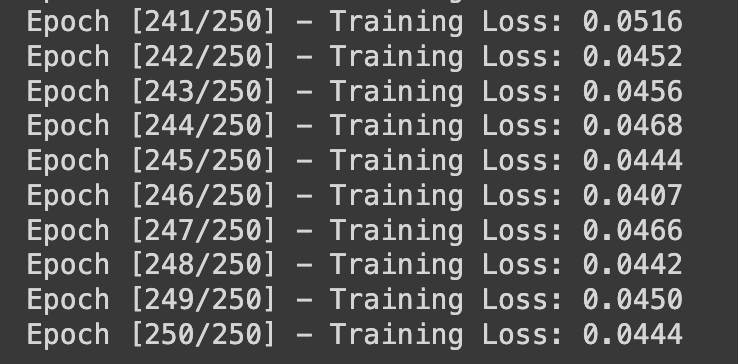
|  |
| --- |
| Fig6. Output cropped image |

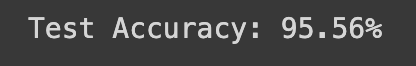
**7.2 Model pseudocode**

****

| Fig7. pseudo code |
| --- |

**7.2 Model output**

****

****

| Fig8. model output |
| --- |

**CHAPTER-8**

**Results and Discussion**

The results of the different models that we tried on the combined dataset are as shown below:

| **Architecture** | **Results** |
| --- | --- |
| CNN Model with no feature extraction | ~64% accuracy and ~44% test loss |
| images on Resnet | ~54% test loss |
| LBP features on neural network | ~75% accuracy and ~30% test loss |
| SIFT features on neural network | ~73% accuracy and ~33% test loss |
| HOG features on neural network | ~76% accuracy and ~29% test loss |
| HOG+SIFT+LBP on neural network | ~95% accuracy and 4.44% test loss |

From the above results it is evident that the standard CNN models might perform well on a particular dataset due to the parameters fine tuning, but when it comes to training on combined dataset they fail since different datasets have different evasion techniques, extraction of features and processing that data on neural network has better impacts and can give better results

**CHAPTER-9**

**CONCLUSION AND FUTURE WORK**

Phase-2 of Capstone Project named “DeepFake Detection For Videos” ensures the completion of below mentioned milestones:

● Preprocessing on the datasets and combining the datasets.

● Test existing architectures by implementing them on the new combined dataset.

● Implementation of our new model.

● Testing, Improving the accuracy and Parameters fine tuning.

● Research paper for publishing it in journals or conferences.

This project can be improved in the future by combining more datasets. It can also be improved by

finding and integrating suitable feature extractors along with the existing ones.

**REFERENCE/ BIBLIOGRAPHY**

[1] Mathews, Sherin, et al. "An explainable deepfake detection framework on a novel unconstrained dataset." Complex & Intelligent Systems (2023): 1-13.

[2] Patil, Kundan, et al. "Deepfake Detection using Biological Features: A Survey." arXiv preprint arXiv:2301.05819 (2023).

[3] Zhao, Hanqing, et al. "Multi-attentional deepfake detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[4] Wodajo, Deressa, and Solomon Atnafu. "Deepfake video detection using convolutional vision transformer." arXiv preprint arXiv:2102.11126 (2021).

[5] Zhang, Li, et al. "Unsupervised learning-based framework for deepfake video detection." IEEE Transactions on Multimedia (2022).

[6] Ikram, Sumaiya Thaseen, Shourya Chambial, and Dhruv Sood. "A Performance Enhancement of Deepfake Video Detection through the use of a Hybrid CNN Deep Learning Model." International journal of electrical and computer engineering systems 14.2 (2023): 169-178.

[7] Li, Xiaodan, et al. "Sharp multiple instance learning for deepfake video detection." Proceedings of the 28th ACM international conference on multimedia. 2020.

[8] Das, S., Seferbekov, S., Datta, A., Islam, M. and Amin, M., 2021. Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3776-3785).

[9] Guarnera, L., Giudice, O. and Battiato, S., 2020. Fighting deepfake by exposing the convolutional traces on images. IEEE Access, 8, pp.165085-165098.

[10] Tan, Z., Yang, Z., Miao, C. and Guo, G., 2022. Transformer-based feature compensation and aggregation for deepfake detection. IEEE Signal Processing Letters, 29, pp.2183-2187.

[11] Coccomini, D.A., Messina, N., Gennaro, C. and Falchi, F., 2022, May. Combining efficientnet and vision transformers for video deepfake detection. In Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III (pp. 219-229). Cham: Springer International Publishing.

[12] V. -N. Tran, S. -G. Kwon, S. -H. Lee, H. -S. Le and K. -R. Kwon, "Generalization of Forgery Detection With Meta Deepfake Detection Model," in IEEE Access, vol. 11, pp. 535-546, 2023, doi: 10.1109/ACCESS.2022.3232290.

[13] W. Lu et al., "Detection of Deepfake Videos Using Long-Distance Attention," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2022.3233063.

[14] Y. Nirkin, L. Wolf, Y. Keller and T. Hassner, "DeepFake Detection Based on Discrepancies Between Faces and Their Context," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 6111-6121, 1 Oct. 2022, doi: 10.1109/TPAMI.2021.3093446.

[15] H. A. Khalil and S. A. Maged, "Deepfakes Creation and Detection Using Deep Learning," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2021, pp. 1-4, doi: 10.1109/MIUCC52538.2021.9447642.

[16] H. Chen, Y. Lin, B. Li and S. Tan, "Learning Features of Intra-consistency and Inter-diversity: Keys towards Generalizable Deepfake Detection," in IEEE Transactions on Circuits and Systems for Video Technology, 2022, doi: 10.1109/TCSVT.2022.3209336.

[17] C. R. Gerstner and H. Farid, "Detecting Real-Time Deep-Fake Videos Using Active Illumination," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 2022, pp. 53-60, doi: 10.1109/CVPRW56347.2022.00015.

[18] J. John and B. V. Sherif, "Comparative Analysis on Different DeepFake Detection Methods and Semi Supervised GAN Architecture for DeepFake Detection," 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Dharan, Nepal, 2022, pp. 516-521, doi: 10.1109/I-SMAC55078.2022.9987265.

[19] A. Chintha et al., "Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection," in IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 5, pp. 1024-1037, Aug. 2020, doi: 10.1109/JSTSP.2020.2999185.

[20] A. Rahman et al., "Short And Low Resolution Deepfake Video Detection Using CNN," 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC), Hyderabad, India, 2022, pp. 259-264, doi: 10.1109/R10-HTC54060.2022.9929719.

**APPENDIX A: DEFINITIONS, ACRONYMS AND ABBREVIATIONS**

| CNN | Convolutional Neural Networks |
| --- | --- |
| RNN | Recurrent Neural Networks |
| DFDC | DeepFake Detection Challenge |
| FF++ | FaceForensics++ |
| GAN | Generative Adversarial Networks |
| EM | Expectation Maximisation |
| CT | Convolutional Traces |
| SOTA | State Of The Art |
| ViT | Vision Transformers |
| MTCNN | Multi Task Cascaded Convolutional Neural Networks |
| MDD | Meta Deepfake Detection |
| AUC | Area Under Curve |
| ACC | Accuracy |
| LSTM | Long Short Term Memory |