IMAGE DEEP FAKE DETECTION

A project Submitted

In partial fulfillment for the

Degree of Bachelor of Technology

in Computer Science and Engineering

Under the Guidance of

TERESSA LONGJAM

Lecturer

Submitted By:

PASUPULETI SAI PRAMOD RAM (20103013)

KAGITHALA VAMSHI VARDHAN (20103014)

KARNIPU VAMSI (20103038)

KOPPERA MADHU KISHORE(20103039)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY MANIPUR

MAY 2024

IMAGE DEEP FAKE DETECTION

A project Submitted

In partial fulfillment for the Degree of Bachelor of Technology in Computer Science and Engineering

Under the Guidance of

TERESSA LONGJAM

Lecturer

Submitted By:

PASUPULETI SAI PRAMOD RAM (20103013)

KAGITHALA VAMSHI VARDHAN (20103014)

KARNIPU VAMSI (20103038)

KOPPERA MADHU KISHORE(20103039)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY MANIPUR

Department of computer science and Engineering
राष्ट्रीय प्रोद्योगिकी संस्थान, मणिपुर
NATIONAL INSTITUTE OF TECHNOLOGY MANIPUR
Lagol,Manipur - 795004
Ph.(0385)2445812, email: admin@nitmanipur.ac.in
(An Autonomous Institute under MHRD, Govt. Of India)

Ref.No. NITM.3/(89-Acad)/CSE/B.Tech/2024/-11 ***Date: 24/05/2024.***

CERTIFICATE

*This is to certify that Dissertation Report entitled, "**IMAGE DEEP FAKE DETECTION**" Submitted by **Pasupuleti Sai Pramod Ram** bearing Enrollment No. **20103013, Kagithala Vamshi Vardhan** bearing Enrollment No. **20103014, Karnipu Vamsi** bearing Enrollment No. **20103038 and Koppera Madhu Kishore** bearing Enrollment No. **20103039** to National Institute of Technology Manipur, India, is a record of bonafide Project work carried out by them under the supervision of **Mrs. Teressa Longjam , Lecturer of NIT Manipur** under the department of **Computer Science & Engineering**, NIT Manipur and is worthy of consideration for the award of the degree of Bachelor of Technology in **Computer Science & Engineering Department** of the Institute.*

Dr. Kh Johnson Singh	**Mrs. Teressa Longjam**
(Head of Department)	(Supervisor)
Assistant Professor	Lecturer
Dept. Of Computer Science and Engineering	Dept. Of Computer Science and Engineering
National Institute of Technology, Manipur	National Institute of Technology, Manipur

NATIONAL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MANIPUR, INDIA

DECLARATION

We the undersigned, certify that:

A. The work contained in this report is original and has been done by us under the guidance of Mrs. Teressa Longjam.

B. The work in this report have not been submitted in part or full to any other Institute or University for the award of any degree or diploma.

C. We have confirmed to the norms and guidelines given in the Ethical code of conduct of the Institute.

D. Whenever we have used materials (data, theoretical analysis figure, text) from other sources, we have given the due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owner of the sources, whenever necessary.

24th May 2024, Imphal

(Pasupuleti Sai Pramod Ram)
Enrollment no- 20103013

(Kagithala Vamshi Vardhan)
Enrollment no- 20103014

(Karnipu Vamsi)
Enrollment no- 20103038

(Koppera Madhu Kishore)
Enrollment no- 20103039

ACKNOWLEDGEMENT

The satisfaction and euphoria that has accompanied successful completion of our project work would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

At the very onset, we would like to express my warm gratitude to our supervisor**, Mrs. Teressa Longjam**, Lecturer, Department of Computer Science and Engineering, for her guidance, invaluable suggestions effusive co-operation and help throughout our project work.

We would like to express our sincere gratitude to Dr. Khundrakpam Johnson Singh, Associate Professor, Department of Computer Science and Engineering, NIT Manipur, for their valuable suggestions and help during project work.

Finally, we would like to thank to all the faculty and staff of Computer Science and Engineering Department, our friends & well-wishers for their valuable help rendered during the course of our project.

24th May 2024, Imphal

(Pasupuleti Sai Pramod Ram)
Enrollment no- 20103013

(Kagithala Vamshi Vardhan)
Enrollment no- 20103014

(Karnipu Vamsi)
Enrollment no- 20103038

(Koppera Madhu Kishore)
Enrollment no- 20103039

ABSTRACT

In our rapidly globalizing world and with the rapidly increasing the technology and the artificial intelligence leads to numerous innovations, one such innovation is deep fake technology, our project focuses on the "detection of the deep fake images using VGG architecture in CNN model with deep learning".

This project harnesses the power of CNN architecture, combined with deep learning techniques, to create a robust and accurate detection mechanism. The face is the most distinctive feature of human beings. With the tremendous growth of face synthesis technology, the security risk posed by face manipulation is becoming increasingly significant.

Our system aims to detect the deep fake image which is generated by pre trained model algorithm by using the deep learning algorithms employed in this project undergo extensive training on large dataset consists of real and deep fake images, enabling the model to grasp the intricacies of both real and fake images and produce contextually accurate detections.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN: Convolutional Neural Networks

VGG: Visual Geometry Grow

DenseNet: Densly Connected Convolutional Networks

JPEG: Joint Photographic Experts Group

PNG: Portable Network Graphics

ReLu: Rectifying Linear Unit

CHAPTER – 1

INTRODUCTION

1.1 Background

Detecting deep fakes within images has become paramount in light of the escalating sophistication and prevalence of manipulated media. Deep fake technology, propelled by advancements in artificial intelligence, empowers individuals to fabricate hyper-realistic images and videos with alarming accuracy. These artificially generated visual assets pose multifaceted risks and ramifications, permeating domains ranging from privacy infringements to profound societal distrust in digital information.

The nefarious potential of deep fakes spans a broad spectrum, encompassing scenarios such as the dissemination of false narratives, identity theft, financial scams, and political manipulation, amplifying concerns regarding ethical integrity and societal stability. Nevertheless, the intricacies involved in detecting deep fakes remain formidable, as these fabricated assets often exhibit a remarkable resemblance to authentic content, lacking discernible traces of manipulation. This challenge is further compounded by the relentless innovation and refinement of deep fake generation techniques, which continually push the boundaries of realism and believability.

Despite these obstacles, the pursuit of effective detection methodologies persists, underpinned by a diverse array of approaches including forensic analysis, statistical feature extraction, and sophisticated deep learning frameworks. Recent strides in deep learning, characterized by the emergence of novel architectures and data-driven methodologies, offer promising avenues for enhancing the efficacy and scalability of deep fake detection systems.

Moreover, collaborative initiatives involving interdisciplinary expertise from academia, industry, regulatory bodies are indispensable in fostering a concerted response to the escalating threat posed by deep fakes. Through collective innovation and collaboration, the development of robust and resilient deep fake detection solutions holds the potential to fortify the digital ecosystem against the insidious spread of manipulated media and safeguard the integrity of information in an increasingly interconnected world.

Detecting deep fakes poses several challenges due to their high level of realism and sophistication. Traditional forensic techniques may struggle to differentiate between authentic and manipulated images, especially in the absence of discernible artifacts or inconsistencies. The rapid evolution of deep fake generation methods and the proliferation of low-cost tools make it challenging to keep pace with emerging threats.

Various approaches have been proposed for detecting deep fakes in images, including forensic analysis, feature-based methods, and deep learning-based techniques. Forensic analysis involves examining metadata, compression artifacts, and other traces of manipulation to identify potential forgeries. Feature-based methods extract statistical features or anomalies from images to distinguish between real and synthetic content. Deep learning-based approaches leverage convolutional neural networks (CNNs) to learn discriminative features and patterns indicative of deep fakes.

Recent research has focused on advancing deep learning-based techniques for deep fake detection, including the development of novel network architectures, loss functions, and training strategies. Adversarial training, multi-modal fusion, and attention mechanisms are among the techniques being explored to improve the robustness and generalization capabilities of detection models. Collaborative efforts involving academia, industry, and government agencies are underway to develop standardized benchmarks, datasets, and evaluation protocols for benchmarking and comparing detection methods.

1.2 Analysis of Problem Statement

Analyzing the problem statement in image deep fake detection involves examining the specific challenges, objectives, and implications associated with the detection of manipulated media. Evaluate the complexity and nuances of detecting deep fakes within images. Identify the technical, ethical, and societal challenges inherent in differentiating between authentic and manipulated content. Consider factors such as the rapid advancement of deep fake generation techniques, the lack of standardized datasets, and the potential for adversarial attacks against detection algorithms.

Define the primary objectives of image deep fake detection, such as mitigating the spread of misinformation, protecting individual privacy, and preserving the integrity of visual content. Specify the desired outcomes of detection efforts, such as accurately identifying manipulated images, minimizing false positives and false negatives, and enabling timely intervention to counteract the dissemination of deep fakes. Identify gaps in existing research and knowledge pertaining to image deep fake detection, such as unexplored detection modalities, understudied datasets, or emerging threats.

Highlight opportunities for future research and innovation, including the development of novel detection algorithms, the creation of comprehensive benchmark datasets, and interdisciplinary collaborations to address multifaceted challenges.

1.3 Objectives

- Identification of Manipulated Images
- Minimization of False Positives and False Negatives
- Privacy Preservation
- Interpretability and Explainability

1.4 Uses of Deep fake Image Detection

- Deep fake image detection can help identify and flag manipulated images used in spreading misinformation and fake news across social media platforms and online channels.
- By detecting and removing deep fake images, detection systems contribute to the prevention of false narratives and the preservation of truthfulness in digital communication.
- Deep fake image detection aids in protecting the reputations of individuals, public figures, and organizations by identifying and removing maliciously manipulated images used for defamation or character assassination.
- Deep fake detection systems can help safeguard privacy by identifying and removing deep fake images that infringe upon individuals' privacy rights or exploit sensitive personal data.

- Deep fake detection assists forensic investigators and law enforcement agencies in identifying tampered images and videos used as evidence in criminal investigations or legal proceedings.
- Social media platforms, content-sharing networks, and online communities use deep fake image detection to moderate user-generated content and enforce community guidelines.
- Deep fake image detection contributes to media literacy initiatives and educational programs aimed at raising awareness about the risks of manipulated media.
- Deep fake image detection fuels research and innovation in the fields of computer vision, machine learning, and digital forensics.

CHAPTER – 2

LITERATURE SURVEY

2.1 Literature review

Introduction:

Deep fake technology has emerged as a significant threat to the integrity of visual content, with manipulated images and videos increasingly being used to spread misinformation and deceive viewers. Detecting deep fake images is a critical task to mitigate these risks and preserve trust in digital media. In this literature survey, we review existing research and methodologies in deep fake image detection, focusing on recent advancements, challenges, and future directions in the field.

Historical Context and Evolution:

The field of deep fake image detection has witnessed rapid evolution in recent years, driven by advancements in machine learning and computer vision. Seminal works such as "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces" by Rossler et al. (2019) and "DeepFake Detection Challenge (DFDC): Dataset and Baseline" by Darius et al. (2020) laid the groundwork for benchmark datasets and evaluation metrics in deep fake detection.

Methodologies and Techniques:

A variety of approaches have been proposed for deep fake image detection, including forensic analysis, feature-based methods, and deep learning-based techniques. Forensic analysis techniques leverage metadata, compression artifacts, and other traces of manipulation to identify deep fake images. Feature-based methods extract statistical features or anomalies from images to distinguish between authentic and manipulated content. Deep learning-based approaches, particularly convolutional neural networks (CNNs) and generative adversarial networks (GANs).

Deep Learning-based Detection Methods:

Recent advancements in deep learning-based detection methods have demonstrated significant improvements in detection accuracy and robustness. State-of-the-art architectures such as DeepFakeDetection (DFD) by Li et al. (2020) and FaceForensics++ by Rossler et al. (2019) employ convolutional neural networks (CNNs) with attention

mechanisms and adversarial training to effectively distinguish between real and manipulated images. These models leverage large-scale benchmark datasets such as Celeb-DF and DeepFakeDetection to learn discriminative features and generalize across diverse deep fake generation techniques.

Datasets and Evaluation Metrics:

Benchmark datasets play a crucial role in training and evaluating deep fake image detection models. Commonly used datasets include Celeb-DF, FaceForensics++, and DeepFakeDetection, each with its unique characteristics and challenges. Evaluation metrics such as accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curves provide quantitative measures of detection performance and generalization capabilities.

Applications and Use Cases:

Deep fake image detection has diverse applications across various domains, including social media moderation, forensic investigations, content authentication, and media forensics. Detection technologies enable platforms to identify and remove harmful or deceptive deep fake images, safeguarding user trust and integrity in digital media.

Challenges and Future Directions:

Despite recent advancements, deep fake image detection faces several challenges, including the rapid evolution of deep fake techniques, adversarial attacks, and scalability issues. Future research efforts should focus on addressing these challenges and exploring novel approaches for enhancing detection accuracy, robustness, and efficiency.

Conclusion:

In conclusion, deep fake image detection is a rapidly evolving field with significant implications. By leveraging innovative methodologies and techniques, researchers and practitioners can develop robust detection systems capable of mitigating the risks posed by manipulated media and preserving trust in online platforms and communities.

CHAPTER – 3

REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

3.1 Requirements Analysis

3.1.1 Functional Requirements

- Image Input: The system should be able to accept image files in various formats (e.g., JPEG, PNG).
- Deep Fake Detection: The core functionality is to analyse and detect whether an image is a deep fake or not.
- Accuracy Reporting: The system should provide a confidence score or probability indicating how likely an image is a deep fake.
- User Interface: Provide a user-friendly interface for uploading images and viewing results.
- Batch Processing: Capability to process multiple images simultaneously.
- Reporting: Generate reports of analysis including metadata and detection results.
- Integration: Ability to integrate with other systems or APIs for automated processing.

3.1.2 Non-Functional Requirements

- Performance: The system should process images quickly, ideally within a few seconds per image.
- Scalability: Should handle increasing numbers of images and users without significant performance degradation.
- Reliability: The system should be robust, with minimal downtime and error rates.
- Security: Ensure that uploaded images and results are securely stored and processed, with appropriate encryption and access controls.
- Usability: The interface should be intuitive and easy to use for individuals with varying levels of technical expertise.
- Compatibility: Should work across different operating systems and devices, particularly within Jupyter Notebook.

3.1.3 Technical Requirements

3.1.3.A. Hardware Requirements

- Processor: A multi-core processor (e.g., Intel i7 or above, AMD Ryzen 7 or above).
- RAM: At least 16 GB of RAM, 32 GB or more recommended for large batch processing.
- GPU: A dedicated GPU (e.g., NVIDIA RTX 2070 or higher) is highly recommended for faster deep learning model inference.
- Storage: At least 500 GB of SSD storage for quick read/write operations and sufficient space for storing datasets and results.

3.1.3.B. Software Requirements

- Operating System: Linux (Ubuntu preferred), Windows 10, or macOS.
- Python: Python 3.7 or higher.
- Jupyter Notebook: Installed and configured.
- Libraries: Essential libraries include TensorFlow, Keras, PyTorch, OpenCV, NumPy, Pandas, Matplotlib, Scikit-learn, etc.
- Deep Learning Frameworks: TensorFlow or PyTorch for model development.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

Ensure the team has expertise in deep learning, computer vision, and software development. Jupyter Notebook is compatible with the required libraries and frameworks and supports interactive development. Adequate computational resources (e.g., GPUs) are available for training and inference. Deep learning models and frameworks for image analysis and deep fake detection are well-developed and mature.

3.2.2 Economic Feasibility

Investment in hardware (GPUs, high-performance servers) and software (licenses, cloud services). Costs associated with hiring skilled personnel and development time. Ongoing costs for cloud services, data storage, and maintenance. Potential for high ROI due to the increasing demand for deep fake detection in various sectors (media, security, legal).

3.2.3 Operational Feasibility

Minimal training required if the system is designed to be user-friendly. The system can be integrated into existing workflows, particularly with APIs and automated processes. Designed to scale as the number of users and volume of images increases. Maintenance involves regular updates to detection models to handle new types of deep fakes. Ensure compliance with data protection regulations and ethical guidelines for handling and processing images.

CHAPTER – 4

PROPOSED WORK

4.1 Data Collection

For developing an effective deep fake image detection system, it is essential to gather a comprehensive and representative dataset. The dataset used in this research has been sourced from the Kaggle website, a reputable platform that hosts a variety of datasets suitable for machine learning and deep learning tasks. This section details the process of selecting, obtaining, and preparing the dataset for our deep fake image detection system.

The dataset selected for this study is the Deep Fake Detection Challenge Dataset available on Kaggle. This dataset has been specifically chosen due to its relevance, quality, and comprehensiveness. It includes a large number of images labeled as either authentic or deep fake, providing a robust foundation for training and evaluating our detection model.

The dataset includes a diverse set of subjects, lighting conditions, and manipulation techniques, which helps in creating a generalized model.

The dataset was split into training, validation, and test sets to ensure an unbiased evaluation of the model's performance:

- Training Set: 70% of the dataset was used for training the deep learning model. This subset was augmented to prevent overfitting and to improve the model's generalization capabilities.
- Validation Set: 15% of the dataset was used for validating the model during training, allowing for hyperparameter tuning and model selection.
- Test Set: The remaining 15% of the dataset was reserved for final testing and evaluation of the model's performance.

4.2 Deep Fake Image Detection Approach

4.2.1 Input Part

The input to our deep fake image detection system consists of digital images. These images are acquired from various sources, including the Kaggle Deep Fake Detection Challenge dataset. Each image is processed and analyzed to determine whether it is an

authentic image or a deep fake. The primary characteristics of the input images are:

- Format: JPEG, PNG, or similar common image formats.
- Dimensions: Resized to a consistent dimension suitable for the convolutional neural networks (CNNs), typically 224x224 pixels.
- Color Space: RGB (Red, Green, Blue) color space is used for most image processing tasks.

The images that are in Figure 4.2.1 are preprocessed to normalize pixel values, augment data for training purposes, and ensure they are in a format suitable for the proposed neural networks.

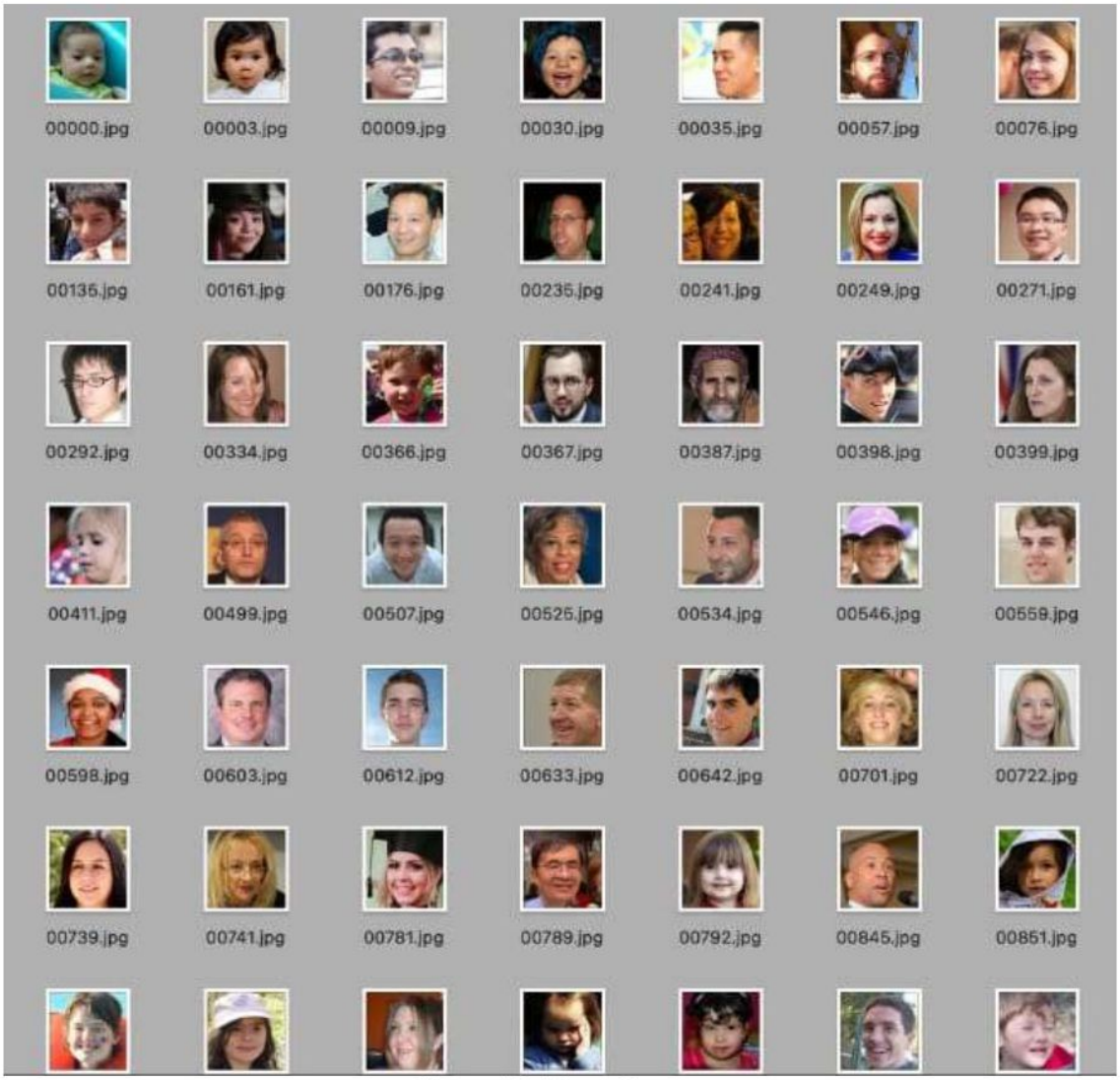

Figure 4.2.1 Input Images

4.2.2 Overflow Diagram

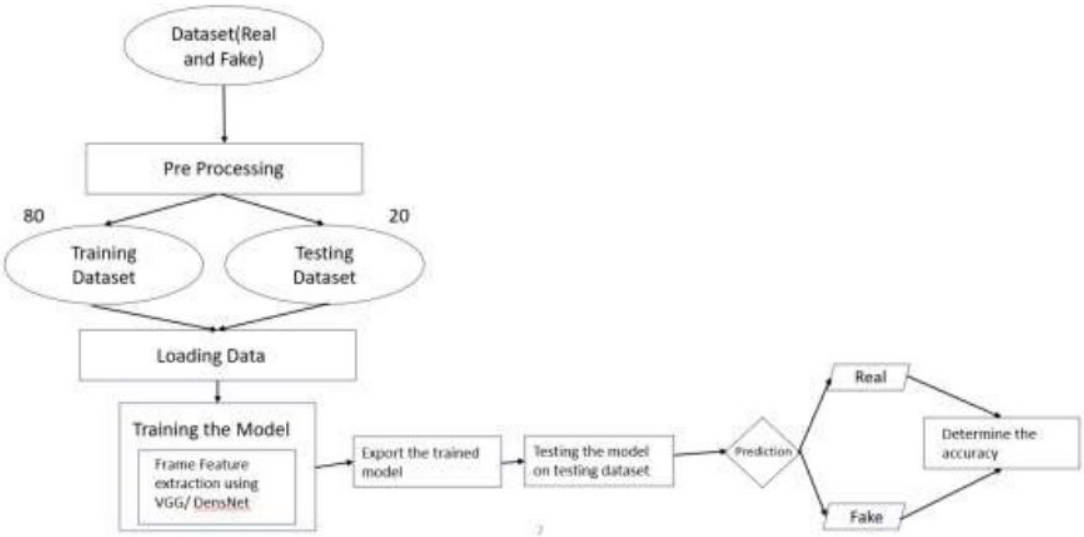

Figure 4.2.2 Project flow

4.2.3 Proposed Convolutional Neural Networks

In this project, we propose using two well-established convolutional neural network architectures: DenseNet and VGG.

Why Convolutional Neural Networks: -

1) CNNs automatically learn hierarchical features, crucial for discerning between real and manipulated content in deep fake detection.
2) Spatial Hierarchies: CNNs' convolutional layers operate on local spatial regions, capturing spatial hierarchies of features, which is essential for analyzing deep fake alterations.
3) Parameter Sharing: CNNs use shared weights, allowing them to recognize patterns regardless of their position in the image, enhancing the detection of translations or distortions.
4) Feature Reuse: Pooling layers in CNNs facilitate downsampling and feature reuse, helping the model focus on essential characteristics while disregarding irrelevant details.

5) Availability of Pre-trained Models: Pre-trained CNN models offer knowledge from diverse datasets, aiding transfer learning for improved detection of anomalies in deep fake images.
6) Robustness to Variations: CNNs learn robust representations, making them resilient to variations in facial expressions, lighting conditions, and backgrounds, characteristics often present in deep fake images.
7) Effective for Large-Scale Data: CNNs are well-suited for handling large-scale datasets, a common requirement in deep fake detection for effective learning of complex patterns associated with manipulated images.

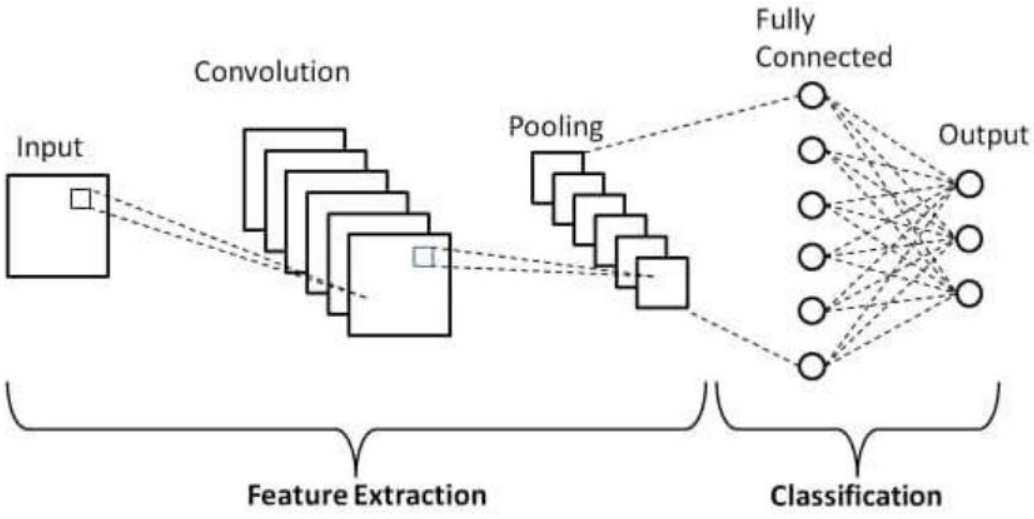

Figure 4.2.3 Convolutional Neural Network

DenseNet (Dense Convolutional Network):

Architecture: DenseNet connects each layer to every other layer in a feed-forward fashion. This densely connected architecture helps in mitigating the vanishing-gradient problem, enhancing feature propagation, and encouraging the reuse of features.

Advantages: DenseNets are known for their efficiency in terms of parameter usage and their ability to achieve high accuracy with fewer parameters compared to traditional CNNs.

Implementation: We utilize pre-trained DenseNet models (e.g., DenseNet121) fine-tuned on our deep fake dataset to leverage their powerful feature extraction capabilities.

VGG (Visual Geometry Group Network):

Architecture: VGG networks are characterized by their use of very small (3x3) convolution filters, which allows them to have a uniform architecture with a consistent structure. The depth of VGG networks varies (e.g., VGG16, VGG19) depending on the number of convolutional layers.

Advantages: VGG networks are simple and easy to implement, with a straightforward architecture that has been proven to be effective for various image classification tasks.
Implementation: We use pre-trained VGG models (e.g., VGG16) and fine-tune them on our dataset, which enables efficient feature extraction and deep fake detection.

4.2.4 Training Phase

During the training phase, the preprocessed images are used to train the proposed DenseNet and VGG models. The training process involves the following steps:

- Data Augmentation: To prevent overfitting and improve model generalization, data augmentation techniques such as rotation, flipping, and scaling are applied to the training images.
- Model Initialization: Pre-trained DenseNet and VGG models are initialized with weights trained on the ImageNet dataset. This transfer learning approach accelerates the training process and improves accuracy.
- Training Configuration:
- Loss Function: Cross-entropy loss is used to measure the difference between the predicted and actual labels.
- Optimizer: Adam optimizer is used for efficient gradient descent and weight updating.
- Learning Rate: A suitable learning rate is chosen and adjusted using techniques such as learning rate decay.
- Training Process: The models are trained on the training dataset with the augmented images, iteratively updating the weights to minimize the loss function. Validation data is used to tune hyperparameters and monitor the model's performance.

- Activation function: he Rectified Linear Unit (ReLU) activation function is widely used in deep learning due to its simplicity and effectiveness. Defined as f(x)=max(0,x), ReLU outputs the input directly if it is positive; otherwise, it outputs zero. This function introduces non-linearity, enabling neural networks to learn complex patterns. One of the key advantages of ReLU is its ability to mitigate the vanishing gradient problem, which helps in training deep networks by maintaining stronger gradients during backpropagation.

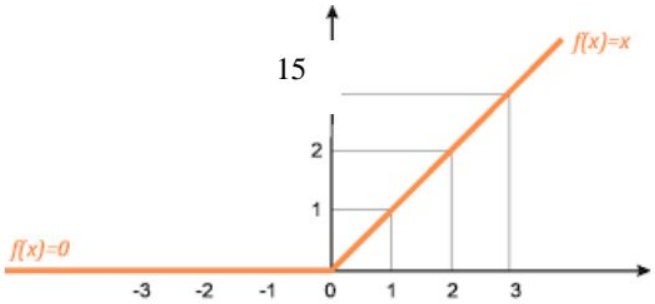

Figure 4.2.4 Relu Activation function

4.2.5 Testing Phase

The testing phase involves evaluating the trained models on a separate test dataset to assess their performance. This phase includes:

- Data Preparation: The test dataset, which was not used during training, is prepared in the same manner as the training data (resized, normalized, etc.).
- Model Evaluation: The trained DenseNet and VGG models are used to predict the labels of the test images.
- Performance Metrics:
- Accuracy: The ratio of correctly predicted images to the total number of images.
- Precision and Recall: Metrics to evaluate the model's performance in identifying deep fakes.
- F1 Score: The harmonic mean of precision and recall, providing a single metric to evaluate the model's performance.
- Confusion Matrix: A matrix to visualize the performance of the classification model by comparing predicted and actual label.

CHAPTER – 5

IMPLEMENTATION

5.1 Data Preprocessing

Normalization and Resizing: -

Employing techniques like resizing and color normalization to ensure uniformity in input data. Standardizing the size and color space reduces variations caused by lighting, background, and facial expressions. Crucial for creating consistent patterns in the data, enhancing the detection model's ability to learn effectively.

Noise Reduction and Artifact Removal:

Applying advanced noise reduction techniques such as Gaussian filtering to address artifacts and enhance image quality. Important for improving the clarity of facial features and reducing the impact of unwanted elements.

Data Augmentation: -

Employing augmentation techniques like flipping, rotating, cropping, and introducing artificial noise to increase the diversity of the dataset. Enhances the model's robustness by exposing it to a variety of scenarios, making it more adept at handling real-world variations.

Data Balancing: -

Addressing class imbalances in the dataset, which may arise from discrepancies in the number of real and fake images. Mitigates the risk of model overfitting to the majority class, ensuring balanced representation and improved generalization.

Data Formatting and Storage: -

Organizing preprocessed images into a format compatible with the chosen deep learning framework, such as NumPy arrays or TensorFlow tensors. Essential for seamless integration into the model training pipeline, including proper labeling to distinguish between real and fake images.

5.2 Implementation of CNN Model

5.2.1 DenseNet 121:

DenseNet121 is a highly efficient convolutional neural network model that introduces the concept of dense connectivity, where each layer receives input from all preceding layers. This architecture mitigates the vanishing gradient problem, strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters compared to traditional CNNs. DenseNet121 is composed of dense blocks and transition layers. In each dense block, every layer is connected to every other layer in a feed-forward fashion, meaning that the feature maps of all preceding layers are used as inputs for each subsequent layer. Transition layers, which include convolution and pooling operations, are placed between dense blocks to control the complexity of the model. The model consists of 121 layers, hence the name DenseNet121. This dense connectivity pattern results in more compact and robust representations, leading to improved accuracy and efficiency, particularly in handling intricate image classification tasks. The dense connectivity pattern of DenseNet, where each layer receives inputs from all previous layers within the same dense block. This innovative design facilitates feature reuse, improves gradient flow, and allows for the construction of very deep networks without the risk of vanishing gradients.

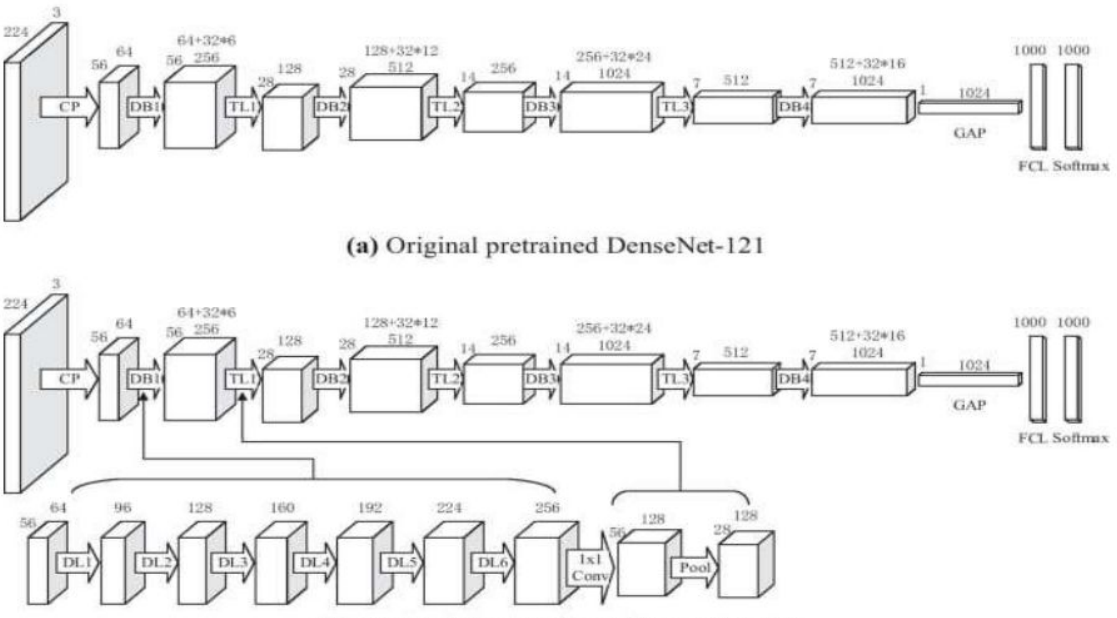

Figure 5.2.1 Overview of Dense Net 121.[6]

5.2.2 VGG16

The VGG (Visual Geometry Group) network is a convolutional neural network (CNN) is known for its simplicity and effectiveness in image classification tasks. Architecture Depth: VGGNet achieved state-of-the-art performance in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2014. It's characterized by its deep architecture, featuring 16 or 19 weight layers (16-layer or 19-layer version), mainly consisting of stacked convolutional layers. The network consists of a sequence of convolutional layers followed by max-pooling layers.

Max-pooling layers are utilized to downsample the spatial dimensions and reduce the number of parameters. Typically, 2x2 max-pooling with a stride of 2 is applied after convolutional layers.

Preprocessing: Preprocessing involves subtracting the mean RGB value from each pixel of the input image to normalize the data. VGG has variants such as VGG16 (16-layer version) and VGG19 (19-layer version), differing in the number of convolutional and fully connected layers.

The VGG architecture's simplicity and homogeneous structure with small convolutional filters enabled it to achieve remarkable performance in image classification tasks.

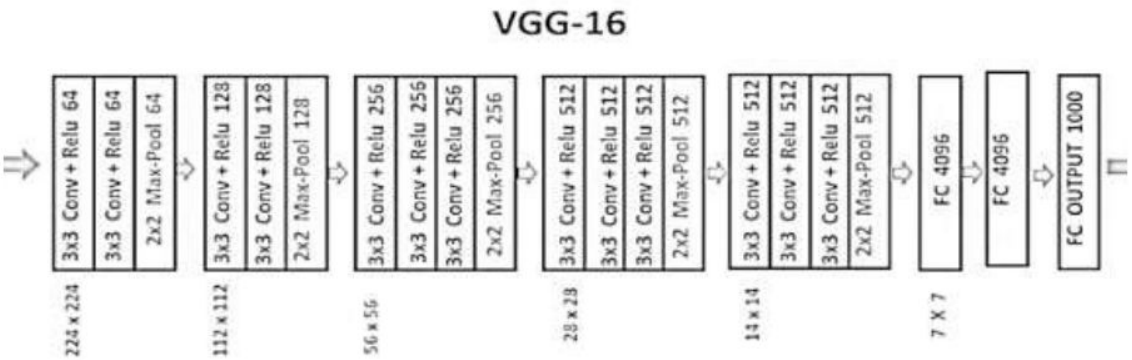

Figure 5.2.3 VGG 16 architecture

5.3 Training the model.

DenseNet 121

The training process for DenseNet-121 in the context of image deep fake detection involves several key steps. Initially, a comprehensive dataset comprising both real and deep fake images is collected and preprocessed, which includes resizing, normalization, and data augmentation to enhance model robustness and generalization. The DenseNet-121 model, pre-trained on ImageNet, is then fine-tuned by modifying its final layers to fit the binary classification task.

Figure 5.4.1 Training epochs for DenseNet121

VGG16

The training process for the VGG16 model involves several key steps, spread across a series of epochs to ensure optimal performance. Each epoch represents one complete pass through the entire training dataset, allowing the model to learn and adjust its weights iteratively. For this study, the VGG16 model was trained over 140 epochs, balancing the need for sufficient training time with the risk of overfitting.

Figure 5.4.2 Training epochs for VGG16

5.4 Performance analysis

5.4.1 Visualizing graphs

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a fundamental tool for data scientists and researchers, widely used for its ability to generate high-quality graphs and plots that can be easily integrated into academic reports and presentations. Matplotlib's versatility allows users to produce a variety of plots, including line charts, bar charts, histograms, scatter plots, and 3D plots, with fine-grained control over their appearance and behavior.

Matplotlib's integration with NumPy and pandas makes it particularly powerful for data analysis workflows. Users can seamlessly visualize data stored in these formats, enabling efficient exploration and communication of data insights. The library also supports interactive features through its integration with IPython and Jupyter notebooks, allowing for dynamic and interactive data exploration.

Matplotlib is an essential tool for generating high-quality visualizations in Python. Its rich functionality, coupled with extensive customization options, makes it invaluable for researchers and data scientists looking to present their findings clearly and effectively.

DenseNet121

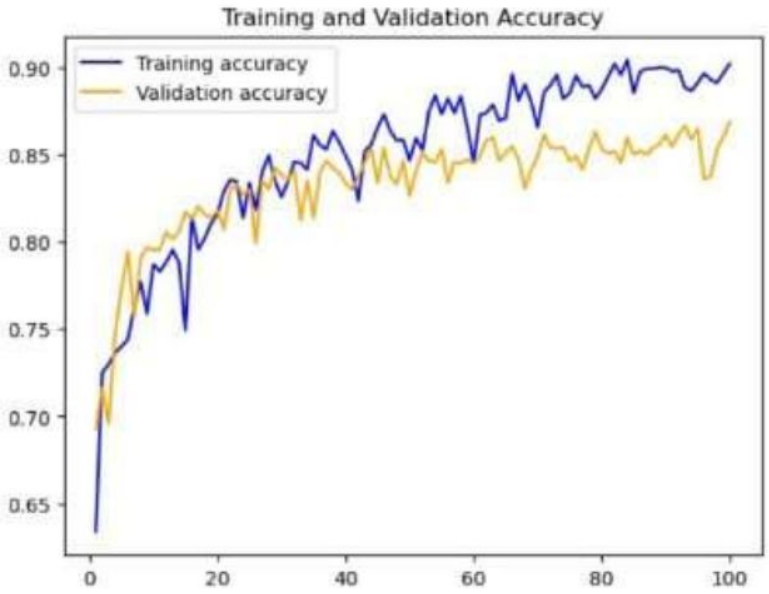

Figure 5.4.1 DenseNet121 Training and Validation Accuracy

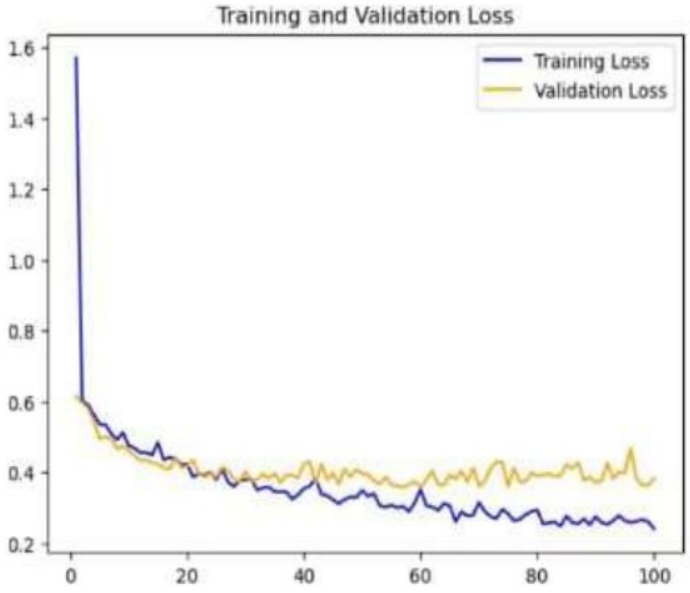

Figure 5.4.2 DenseNet121 Training and Validation loss

VGG16

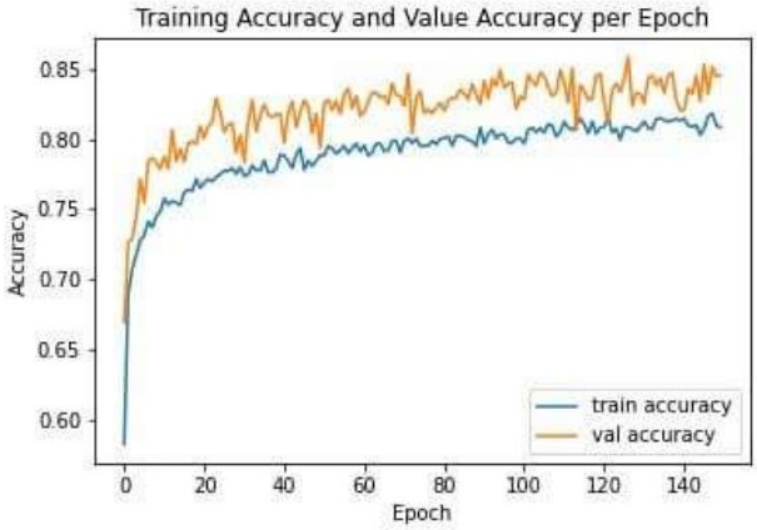

Figure 5.4.3 VGG16 Training and Validation accuracy

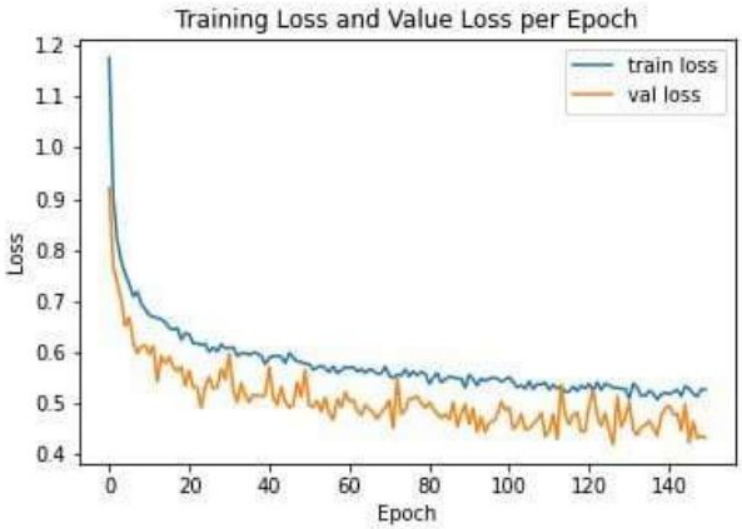

Figure 5.4.4 VGG16 Training and Validation loss

5.4.2 Performance metrics

A confusion matrix is a table used to evaluate the performance of a classification algorithm. It is particularly useful for binary classification but can be extended to multi-class classification as well.

True Positive (TP):

The number of instances that are correctly predicted as the positive class.
The count of instances where the actual class is positive, and the predicted class is also positive.

True Negative (TN):

The number of instances that are correctly predicted as the negative class.
The count of instances where the actual class is negative, and the predicted class is also negative.

False Positive (FP):

The number of instances that are incorrectly predicted as the positive class when they belong to the negative class.
The count of instances where the actual class is negative, but the predicted class is positive.

False Negative (FN):

The number of instances that are incorrectly predicted as the negative class when they belong to the positive class.
The count of instances where the actual class is positive, but the predicted class is negative.

Derived Metrics

From the values in the confusion matrix, several performance metrics can be derived:

Accuracy:

Accuracy represents the proportion of correctly classified instances out of all instances in the dataset. It gives an overall measure of how well the model is performing.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (Positive Predictive Value):

Precision, also known as Positive Predictive Value, measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the correctness of positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity, True Positive Rate):

Recall, also known as Sensitivity or True Positive Rate, measures the proportion of correctly predicted positive instances out of all actual positive instances. It focuses on capturing all positive instances.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score:

The F1 Score is the harmonic mean of precision and recall. It provides a balance between precision and recall, considering both false positives and false negatives. It's particularly useful when class imbalance is present.

$$F1\ Score = 2\ X\ \frac{Precison\ X\ Recall}{Precison + Recall}$$

DenseNet121

```
Accuracy: 0.8866666666666667
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89      1500
           1       0.90      0.87      0.88      1500

    accuracy                           0.89      3000
   macro avg       0.89      0.89      0.89      3000
weighted avg       0.89      0.89      0.89      3000
```

Figure 5.4.2.1 DenseNet 121 Classification Report and Accuracy

VGG16

```
Accuracy: 0.83
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.88      0.84      1500
           1       0.86      0.78      0.82      1500

    accuracy                           0.83      3000
   macro avg       0.83      0.83      0.83      3000
weighted avg       0.83      0.83      0.83      3000
```

Figure 5.4.2.2 VGG16 Classification Report and Accuracy

5.5 Output

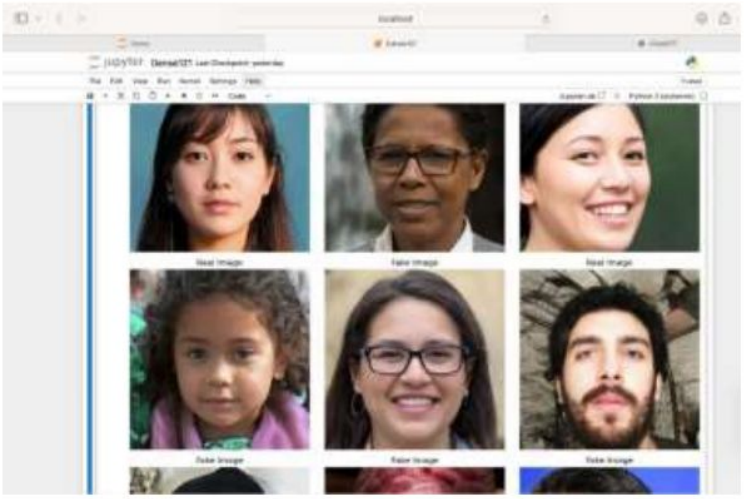

Figure 5.5.1 Output1

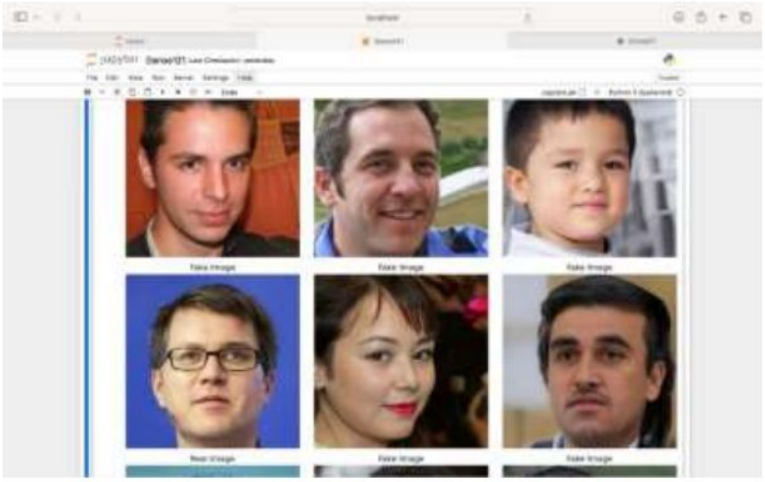

Figure 5.5.2 Output2

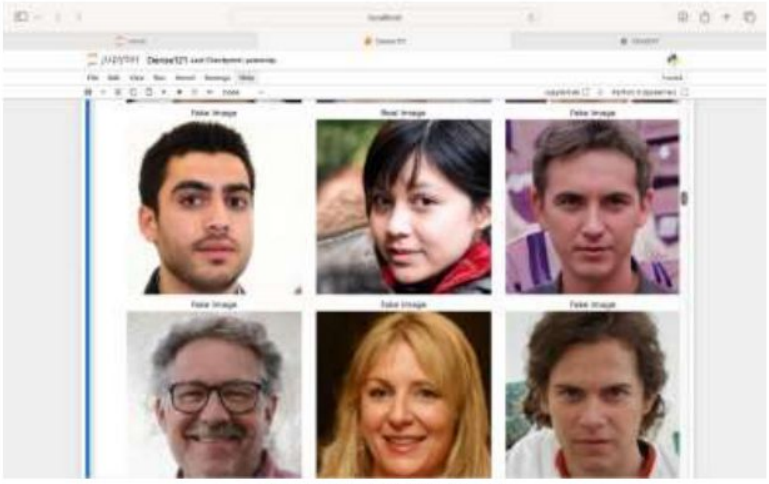

Figure 5.5.3 Output3

CHAPTER – 6

CONCLUSION

AND

FUTURE WORK

6.1 Conclusion

Our research evaluated various detection techniques, including traditional methods and deep learning approaches. We found that deep learning-based methods, particularly those utilizing convolutional neural networks (CNNs), demonstrate superior performance in detecting image deep fakes compared to traditional methods. Despite the advancements in detection technology, challenges remain, including the emergence of more sophisticated deep fake generation models that aim to evade detection. Additionally, the existence of high-quality datasets for training and testing purposes remains a bottleneck in the development of robust detection systems. As deep fake detection systems are deployed in real-world applications, the need for explainability and interpretability becomes paramount. Beyond technical considerations While it presents opportunities for creative expression and entertainment, it also poses significant risks to privacy, security, and democracy. As such, the development of reliable detection systems is not only a technological imperative but also an ethical responsibility.

6.2 Future work

In the future, I plan to focus on the detection of deep fake videos, a rapidly evolving area that presents unique challenges and opportunities. Building on the advancements in image-based deep fake detection, my goal is to develop robust models that can analyze both spatial and temporal features of video content. This involves leveraging advanced deep learning architectures, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), to capture subtle inconsistencies across frames that are indicative of deep fakes. Additionally, I aim to incorporate multi-modal data, integrating audio and contextual information to enhance detection accuracy and resilience against sophisticated fake generation techniques. To support this, I will work on optimizing computational efficiency, making the models suitable for deployment on resource-constrained devices. Furthermore, addressing the ethical implications and ensuring the transparency of detection algorithms will be crucial to gaining public trust and facilitating widespread adoption. Through interdisciplinary collaboration, I hope to contribute to creating a safer digital environment by mitigating the risks associated with deep fake videos.

References

[1] (2020). "Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection." Information Fusion, 64, 131-148. doi:10.1016/j.inffus.2020.07.002.

[2] Verdoliva, L. (2020). "Media Forensics and DeepFakes: An Overview." IEEE Journal of Selected Topics in Signal Processing, 14(5), 910-932. doi:10.1109/JSTSP.2020.3002101.

[3] Matern, F., Riess, C., & Stamminger, M. (2019). "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations." 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2521-2529. doi:10.1109/WACV.2019.00311.

[4] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019). "Protecting World Leaders Against Deep Fakes." 2019 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 38-45. doi:10.1109/CVPRW.2019.00009.

[5] Masi, I., Killekar, A., Rawls, S., & Medioni, G. (2020). "Two-Branch Recurrent Network for Isolating DeepFakes in Videos." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9039-9048. doi:10.1109/CVPR42600.2020.00906.

[6] Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. (2022). "Deep Learning for Deepfakes Creation and Detection: A Survey." arXiv preprint arXiv:1909.11573.

[7] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). "Deepfakes Evolution: Analysis of Facial Regions and Fake Detection Performance." 2020 IEEE International Joint Conference on Biometrics (IJCB), 1-8. doi:10.1109/IJCB48548.2020.9304882.

[8] Mirsky, Y., & Lee, W. (2021). "The Creation and Detection of Deepfakes: A Survey." ACM Computing Surveys (CSUR), 54(1), 1-41. doi:10.1145/3425780.

[9] https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

[10] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186. keywords: {Convolution;Neurons;Convolutional neural networks;Feature extraction;Image edge detection;machine learning;artificial neural networks;deep learning;convolutional neural networks;computer vision;Image recognition}

[11] https://www.kaggle.com/datasets/sachchitkunichetty/rvf10k