

CHAPTER 1

INTRODUCTION

1.1 Project Description

The technology of deepfake has been developing rapidly relying on the methods of deep learning to produce highly realistic fake images and videos. Although this technology has transformed such spheres of activity as entertainment, augmented reality, and digital content creation, it evokes critical ethical and security questions. Deepfake technology has become quite common in all forms of misuses, misinformation, identity theft, and digital fraud, at the same time that it becomes harder and harder to tell where genuine and where manipulated media. As the current deepfake algorithm grows advanced, the existing commanding mechanisms find it challenging to stay in pace, which is why more efficient and intelligent mechanisms should be developed.

The proposed project will concentrate upon Deepfake Image Detection founded on GAN referred to as GANception. The system uses the strength of GANs to train and detect minute artifacts of deepfake images, which are aurally undetectable by a human observer. As its algorithm trains on a set of data composed of both genuine and synthetic images, it learns to associate deep features that distinguish the property of genuine images and AI-generated images. The detection framework will facilitate security in the field of digital media by offering an effective and precise technique of detecting the deepfake content which would help in preventing digital manipulation and online misinformation.

1.2 Project Definition

The proposed project aims at the design of Deepfake Image Affirmation System on the basis of Generative Adversarial Networks (GANs) in order to precisely identify the authenticity of images and distinguish between real and artificial intelligence-generated (fake) pictures. The model is trained using a set of datasets that include both real and fake image patterns to make the model learn deep visual patterns and arts-specific to deep fakes. The system seeks to offer a significant and effective approach to deepfake image detection because the concern in this area is the existence of misinformation, identity theft, and digital manipulation in multiple issues such as media, cybersecurity, and forensic studies, using GANs.

1.2 Proposed solution

The given solution aims to design a scalable, flexible, and resource-effective Deepfake image detector relying on high-fidelity deep learning methods. Because of such challenges facing the detection of Deepfake propagation as real-time identification, batch processing, confidence scoring, and ability to evolve the Deepfake generation techniques, the system is set to tackle them.

With the power of the GAN-based model, the system is able to identify the following weaknesses that come with using synthetic imaging to generate an image such as the texture inconsistency, un-natural facial asymmetry, and light anomalies, etc. The preprocessing steps implemented using a powerful pipeline guarantee quality feature extraction, and real-time inference enables real-time identification of fake video frames, uploads in social media, and forensics pools.

1.3 Purpose

The main idea of this project is to design and develop a scalable, efficient, and reliable Deepfake image detection system to address increased misinformation issues, privacy violations, and ethical abuse of the generated AI-related contents. With the growing perfection of the Deepfake technology, the problem of differentiating between authentic and altered images has become a vital issue of the digital forensics, authentication of the media, and digital security.

This project is based on Generative Adversarial Networks (GANs) which are used to assess and identify Deepfake artifacts and report on these through patterns, inconsistencies, and manipulated modern additions identified in fabricated images. It will integrate in the system real-time detection, batch image processing and comprehensive forensics reporting, where a user will receive confidence scores and where to view suspicious regions. This solution will strive to boost trust, security, and authenticity of the digital media due to the use of state-of-the-art machine learning and continuously enhanced models that will guarantee high safety about Deepfakes-based threats.

1.4 Scope

This project involves the creation of an efficient and easily extendable Deepfake detector that will be able to examine and detect manipulated images and do so with a great level of precision. The system has been built to support real-time detection operation as well as offline processing which makes the system efficient in all forms of applications including social media, forensic, and cyber security applications.

Also, the framework designed should be able to match the developing Deepfake generation methods using GAN-based artifact detection to identify synthetic patterns in manipulated pictures. The system will also have rich forensic reporting, that helps to gain insight into confidence scores, suspected areas of manipulation, and statistically expected oddities. This will make the detection system a trusted one that can be interpreted and is future-proof to counter the new threats in AI-generated media.

CHAPTER 2

LITERATURE SURVEY

2.1 Domain Survey

The current deepfake detection systems mostly depend on conventional CNN-based schemes. Although such models have shown themselves capable at identifying some of the common kinds of artifacts produced by deepfake generation programs, like FaceSwap or DeepFaceLab, they break when combating new or previously unseen types. This brings about the fact that these systems are usually trained on a particular piece of data and thus, can not generalize to new and emerging methods of manipulation.

2.2 Related Work

This involves the study of research papers and journals. The literature survey is completed by considering the following research papers.

1. Deepfake Detection using GAN Discriminators

Publisher: IEEE (18 October 2021)

Author: Sai Ashrith Aduwala, Manish Arigala

SUMMARY:

The abstract states the issues of deepfake videos that create manipulated videos through the technique of advanced deep learning that enables the manipulation of character features by another person in a video. Such doctoried videos may be misused and therefore there is a need to have efficient detection strategies. Although deepfakes can be generated using Generative Adversarial Networks (GANs), this paper explores the possibility of using the GANs in the form of discriminators to detect it.

2. A GAN-Based Model of Deepfake Detection in Social Media

Publisher: ScienceDirect (31 January 2023)

Author: Preeti , Manoj Kumar, Hitesh Kumar Sharma

SUMMARY:

The paper addresses the approaches to the implementation of deepfakes including; deep convolution-based GAN models and techniques to manipulate as well as detect deepfakes. The abstract points to the introduction of deepfake technology, which with the adoption of Generative Adversarial Networks (GANs) allows smooth replacement of person identities in video or image files. Realistic fake content has been generated with society facing challenges, as the availability of huge chunks of public data and sophisticated tools for deep learning has enabled so.

3. Hybrid Deep Learning Model Based on GAN and RESNET for DEEPFAKE

Publisher: IEEE (19 JUNE 2024)

Author: SOHA SAFWAT, AYAT MAHMOUD, FARID ALI

SUMMARY:

This paper proposes a new hybrid deep learning approach that leverages both generative power of Generative Adversarial Networks (GANs) and discriminative skills of Residual Neural Network (RESNET) architecture to identify fake faces.

The proposal model is effective in the differentiation of real and artificially produced faces through a combination of the two methods. The hybrid model was compared to such pre-trained models as VGG16 and RESNET50 and proved to be more effective than them.

4. TITLE: Proactive Deep Fake Detection using GAN-based Visible Watermarking

Publisher: ACM Digital Library (12 September 2024)

Authors: Aakash Varma Nadimpalli, Ajita Rattani

SUMMARY:

This research covers the flaws of passive techniques of DeepFake detection which can serve as the ex-post methods of anti-forensics and which cannot stop the dissemination of the initial disinformation. The authors introduce an example of a new proactive protective mechanism of GAN-based visible watermarking to fight DeepFake production.

The proposed method inserts distinctive visible watermarks to particular regions of generated fake images by introducing a latter regularization term of reconstructive regularization into the loss function of GAN. The altered pictures can be found easily by human beings and DeepFake detectors, due to these watermarks.

5. TITLE: Fighting Deepfakes by Detecting GAN DCT Anomalies

Publisher: MDPI (30 July 2021)

Authors: Oliver Giudice, Luca Guarnera, Sebastiano Battiato

SUMMARY:

The proposed method is a new Deepfake detection pipeline that identifies the properties of Deepfake models using GAN-specific frequencies (GSF), or the fingerprint of various generative models.

This approach deals with the issue of non-generalizability and non-explainability of current Deepfake detection algorithms. Analyzing ad-hoc frequencies created by GAN engines, the suggested method assesses anomalous frequencies using Discrete Cosine Transform (DCT).

6. Advancing GAN Deepfake Detection: Mixed Datasets and Comprehensive Artifact Analysis

Publisher: MDPI(18 January 2025)

Author: Tamer Say, Mustafa Alkan, Aynur Kocak

SUMMARY:

The paper proposes a GAN artifact categorization taxonomy, a new hybrid dataset (StyleGAN3, ProGAN, InterfaceGAN), and a hybrid detection method combining frequency space analysis and RGB color correlation to extend deepfake image detection to instances of static deep fakes. Carrying out an assessment of GAN models, three transform techniques and 12 classifiers, the experiment proves to have high accuracy, precision, recall and F1 scores, establishing that artifact-based detection of synthetic face pictures is effective.

7. MCGAN—a cutting edge approach to real time investigate of multimedia deepfake

Publisher: Scientific Reports(26 November 2024)

Author: Shahid Karim, Xin Liu, Abdullah Ayub khan, Asif Ali Laghari

SUMMARY:

The article introduces what is called MCGAN, a multi-collaborator model that uses GANs and Transfer Learning (TL) to identify deepfakes in a format that can be audio, video, and image, with up to 17.333 percent in accuracy.

The framework uses TL such that the framework achieves fast training time, generalization to novel deepfake patterns, as well as real-time detection with high accuracy across multimedia experiences.

8. Boosting Deep Feature Fusion-Based Detection Model for Fake Faces Generated by Generative Adversarial Networks for Consumer Space Environment.

Publisher: IEEE (30 September 2024)

Author: Fadwa Alrowais, Asma Abbas Hassan, Wafa Sulaiman Almukadi

SUMMARY:

The DF4D-GGAN model is a deep learning modelality that will secure the purpose of identifying deep fake images within the consumer environments. It uses Gaussian filtering as preprocessing of the images, EfficientNet-b4 and ShuffleNet to fusion the features, and hyperparameter optimization using an improved slime mould algorithm (ISMA). The images are then effectively classified as either real or fake inform of an extreme learning machine (ELM) classifier.

The DF4D-GGAN approach obtained better detection results and performance than other models (through large-scale simulations on benchmark datasets) that can be used to detect fine-grained anomalies and GAN artifacts on generated pictures disseminated through deep neural networks.

9. An overview of GAN-Deep Fakes detection: proposal, improvement, and evaluation

Publisher: Spring Nature(20 September 2023)

Author: Fatma Ben Aissa, Monia Hamdi , Mahmoud Mejdoub, Mourad Zaied

SUMMARY:

With the rapid advancement of GANs, highly realistic images can now be generated, making it difficult even for trained viewers to differentiate artificial images from real ones, highlighting the critical need for effective DeepFake detection methods.

The study provides an overview of GAN technology, its variants used to create DeepFakes, and a comprehensive review of existing techniques proposed in the literature for detecting DeepFakes, aiding in understanding and addressing this emerging challenge.

CHAPTER 3

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware requirements

Hardware Specification	
Specification	Desired Value
Processor	Intel i3 and above

Memory (RAM)	8GB Minimum
SSD	250 Minimum

3.2 Software requirements

Software Specification	
Specification	Desired value
Operating System	WINDOWS 10 and Above
Front end Tool	React Js v(19.0.0)
Backend Tool	Python (V3.12), FastAPI(V 0.110.2)
Image Processing Libraies	OpenCV (V- 4.10.0)
Development Tool	Visual studio code
Deep Learning frameworks	Tensor Flow,Keras

CHAPTER 4

SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Functional requirement

These are the following functional requirement:

1. Signup & Login

The users should be in a position of conducting a registration, or secure log-in using an email address. The site will demand that users log on so that his/her information can be shielded and personalised experiences be administered including his/her stashed articles and his/her habits.

2. Images identification in real-time:

The system ought to be in a position to guarantee that it labels accurately an input image as original or deepfake using sophisticated AI models. The detection pipeline offers real-time image processing, working attribute of the GANception abilities, bringing fast and reliable results. The basis of the mechanism is the study of various characteristics of the face, distinctions between the texture, and the use of abnormalities that are injected when a deepfake.AI models are created.

3. Batch Image Processing

The system should also process the multiple images in a way that they can smoothly operate when dealing with huge amounts of information. It is a useful attribute to potential users who require fast work to handle large data of images with requirements of various sets such as validation of data sets, forensic analysis, and moderation. The system carries out the processing within a small period of time and with a high accuracy using the parallel processing capabilities.

4. Image Preprocessing:

In a bid to have standardized and precise analysis, the system executes an extensive image preprocessing pipeline. The purpose of this stage is to prepare raw images to the detection model by dealing with variations in size, brightness, and noise level which otherwise may influence the level of detection accuracy. The preprocessing processing pipeline is fundamental to deriving worthwhile features especially in their detection of minor deepfake signs.

5. GAN based artifact detection:

The images produced using GAN usually have little artifacts that are a unique characteristic of authentic images. Such discrepancies occur because of what the training of GAN is and in such aspects of texture synthesis and illumination consistency and feature aggregation. The detection of such artifacts is significant in their identification of a deepfake since a manual visual inspection will ignore them.

6. Detailed Reporting:

The system produces detailed reports on each image that is to be analyzed, giving information on identified artifacts, confidence, and suggestions that can be applied. The reports are an important weapon of the forensic analysts, police, and end-users who require to discover the character of possible manipulations.

7. Confidence Scoring:

The Confidence Scoring system is an objective measure of how probable an image is likely to be a deepfake and is an easily interpretable metric. This score is based on artifact detection, feature analysis and in-house probabilities of GANception.

4.1 Non-Functional requirement

These are following non-functional requirement:

1. **Performance:** Low-latency detection is required in order to be able to use its real-time applications. Optimize computations, so that the software can run on commodity hardware (e.g., consumer computers).
2. **Scalability:** Ability to process large data sets in batches and support databases.
3. **Adaptability:** Be flexible towards new methods of deepfake production and new GAN designs.
4. **Reliability:** Be sound and take few false positives and negatives in the detection process.
5. **Accuracy:** Maximum accuracy rate in the detection of deepfake through use of state of art neural networks and feature extraction methods.
6. **Security:** Encrypt data integrity and privacy of images and user data

CHAPTER 5

SYSTEM DESIGN

5.1 Architecture Diagram

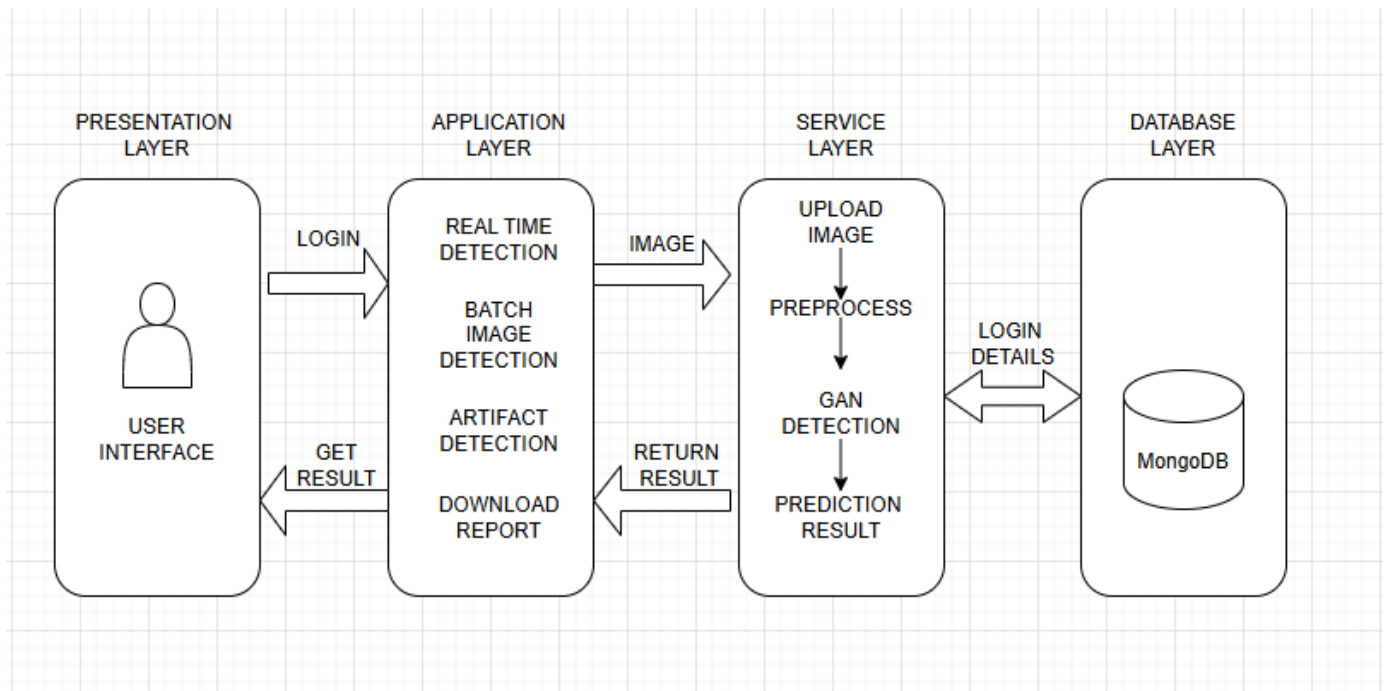


Figure 1 Architecture diagram of Deepfake Image detection

The Deepfake Image Detection using GANception System Architecture is lightweight, modular, scalable, and extensively user-friendly. It is built of three primary parts, including the UI layer, the Backend services, and the Database. These elements are integrated to give the user a convenient package to identify deepfake images and render in-depth reports of the artifacts detection and confidence values.

The client application that is developed based on React provides the User Interface (UI) which could be considered an interactive platform where the user could effortlessly upload images and read detection results. It contains a user-friendly interface, allowing the user to interact with the service in form of drag-and-drop uploads and real time responses to the process of detection. After one uploaded an image, the UI communicated with the backend via RESTfulAPI calls.

This guarantees a fast exchange of information between the browser of the user and the backend services, offering a convenient access to the results and detail reports with graphical pointers to the identified artifacts.

The Backend is an important component in dealing with the processing and deepfake processing logic. It processes images with GANception model that was trained on Python-based REST services. Some processing steps that are carried out by the backend are preprocessing of images, feature extraction, and artifact detection. The system uses a number of workers to run every one of those services at the same time to guarantee scalability and efficient management of numerous requests. Such multi-worker arrangement warrants that even in case multiple users upload images at the same time, the system will continue to exhibit quick response rate and predictable processing. Their activities are run based on a queuing system where there is even allocation of work to the workers.

The architecture includes MongoDB in its data storage tier because it is used to facilitate the storage of and quick retrieval of historical out- comes. The strength of NoSQL enables MongoDB to be flexible in storing diverse data such as images uploaded by users, readings by detection and reports generated. It also allows the system to effectively perform the frequent reading and writing, a key component to saving large amounts of images and their respective analysis. Logs and historical data are also stored in the database, hence users are able to check past results and trends in detection over time.

5.2 Context Flow Diagram

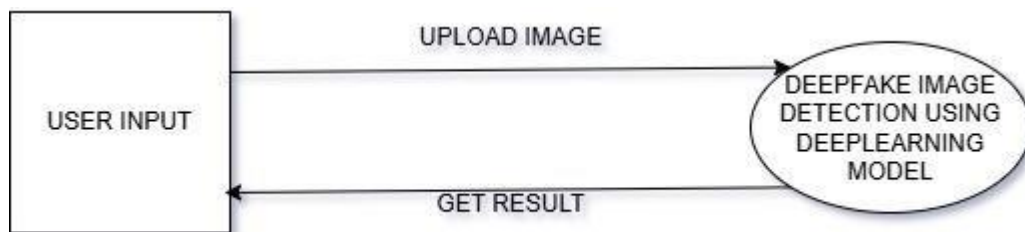


Figure 2 Context Flow Diagram of Deep Fake Image Detection

The scheme demonstrates the simple user interaction process within the deepfake image detection system. The stage to be reached first in the process is the User Input one, during which the copy of an image is uploaded with the help of the interface by the user. Then the uploaded image is forwarded to the deep learning model. The interaction between the user input and the model is established as an upload mechanism that makes sure that the image is in the right format and preprocessed before its analysis. It is an essential first step because it preconditions the proper detection of deepfakes by preconditioning the image to investigate it in detail.

When an image is submitted to the Deepfake Image Detection Using Deep Learning Model, then the image is subjected to additional processing to provide hints of possible fake image signs. The model processes multiple aspects of the image including texture inconsistency, GAN-specific artifacts, unnatural edges, and so on to understand the probability of it being a deepfake. Once the analysis is finished, the system produces a finding which contains a confidence value and a report of found artifacts.

This output is relayed back to the user interface where one can see the result of detection. This two-step interaction (upload an image, get a result) is so simple that it makes the system user-friendly, at the same time, deepfake detection will be precise and effective.

CHAPTER 6

DETAIL DESIGN

6.1 Use Case Diagram

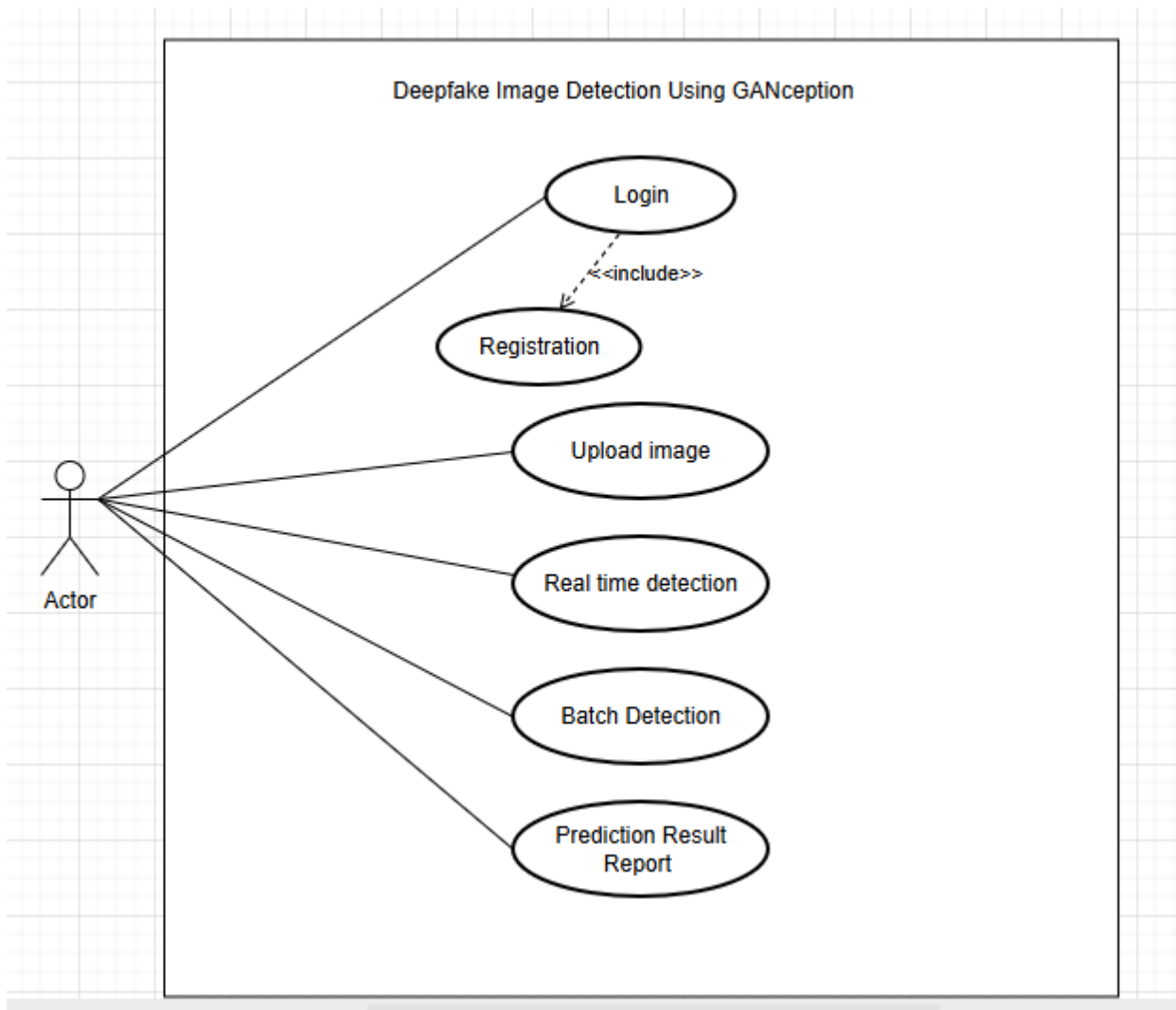


Figure 3 Use case Diagram of Deepfake Image detection

GANception-based system with the view of the user. Fundamentally, the user communicates with the system in a number of modules, the first one of which is Login and Registration module, which is required to guarantee user access and customizable using records. Such modules will make sure that the system detection functionality will be accessible only to authenticated users thereby improving data privacy and security. After logging in, the users can continue with Real-time Image Detection or Batch Image Processing. The real time option enables one to analyze a single image at a time, and the batch processing addition is offered to those users who have the need to analyze a number of images together. Both these use cases use the Image Processing module that helps in doing the primary preprocessing functions like resizing and removing noises in order to make images usable in the detection.

Following the image processing part, the GAN-Based Artifact Detection module is a very important component to detect slight inaccuracies and GAN artifacts specific of the deepfake images identified by GANs. This module advanced deep learning to inspect textures, lighting and anomalous blending. The detection mechanism serves a Confidence Score, which offers quantitative data to the users, showing how probable an image could be considered a deepfake. Moreover, the system can generate a Detailed Report, which describes the results of the analysis, artifacts that were detected, their confidence level, and potentially recommendable further actions. Such an all-inclusive organization guarantees smooth user experience, informative, and dependable experience, including account and image assessment in details.

6.1 Sequence Diagram

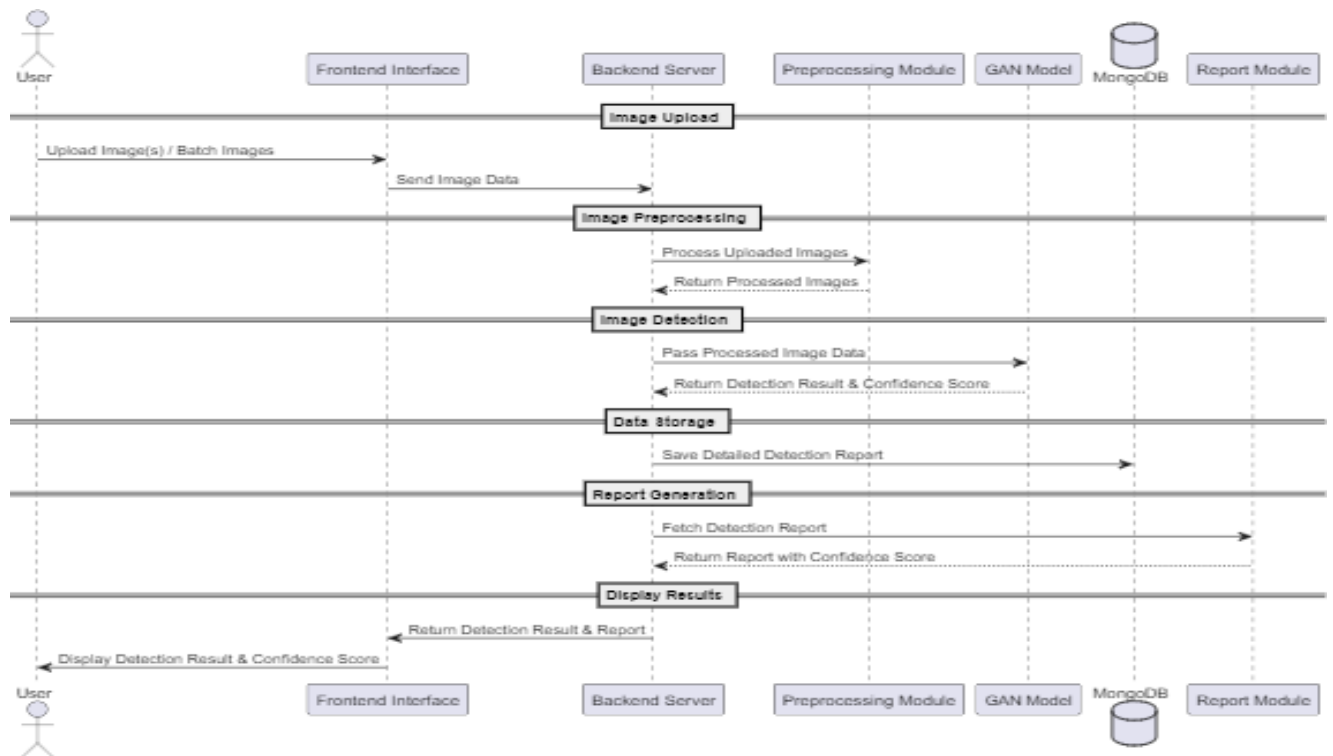


Figure 4 Sequence Diagram of Deepfake Image detection

The sequence diagram shows how operations run in the Deepfake Image Detection System with GANception with the emphasis on the relationship between the user, frontend interface, backend server, preprocessing module, GAN model, MongoDB and report generation module. This is done by the user uploading one or several images via the Frontend Interface that transfers image data to the Backend Server. As soon as the images are received, the backend starts the process of Image Upload sending the data through Preprocessing Module. To detect the pictures, this module undertakes necessary actions, such as resizing and noise cleaning, as well as normalization of the images. After the preprocessing of the image is done the module will release back the processed image to the backend.

Then the Image Detection stage commences where the generated images are passed to the GAN Model. The model performs analysis of the images, identifying any deepfake features and comes up with a Confidence Score that indicates a possibility of the image being distorted. These detection results and scores are passed back to the backend and this process further calls Data Storage process. The detailed detection report is stored in MongoDB so that it can be referred in the future. Thereafter, Report Generation module retrieves the stored data and assembles the results of the detection and provides a combined report to the backend server.

Finally, the Display Results step is performed, during which the final outcome of the detection, confidence score, and the detailed report are forwarded to the user using the frontend interface. Such procedural approach makes the communication between its parts smooth and allows to have the right results on deepfake detection in the proper time.

DATABASE DESIGN

Document Structure

Users	
<pre>{ _id: "ObjcetId", name: String, email: String, password: String, saved_articles: List<Map<String,String>>, createdAt: Date, updatedAt: Date, }</pre>	<pre>{ "_id": { "\$oid": "679112bac9ff56b57e48038d" }, "name": "Shrinivas", "email": "test@test.com", "password": "\$2a\$10\$V9PEe6bQJifcLFSNG6Zkt.RP17udFL0oMtRJIAYezgJT nZsc kklG2",</pre>

CHAPTER 7

METHODOLOGY

7.1 Real-Time and Batch Detection Systems:

Current models of detecting deepfakes are focused mostly on the accuracy of the detection, yet little regard is paid to the situation in which it might be put to use. Proprietary solutions like Deepware Scanner or Microsoft Video Authenticator are more system-specific and do not support flexible deployment options, in particular, in the case of batch processing and real-time response.

Real-Time Detection:

The system suggested can connect with the OpenCV interface in order to make the image capture and analysis in real-time resulting in the reception of the envisioned signal using the live video input. This has a low latency detection and hence can be used in applications that are time sensitive like the surveillance systems, biometric authentication and the media streaming live.

Batch Detection:

The system has features of batch processing in order to support the analysis on a large scale. This aspect enables users to upload and analyze a large amount of images at once, which is especially useful in areas, such as digital forensics, archival analysis, and dataset.

7.2 Image Normalization and Preprocessing

Preprocessing is a key element to increasing performance of deepfake detection systems. The system has a well-developed and well tested image pre-processing component that allows consistency and increases accuracy of the detection by lowering inter-class variance and highlighting GAN generated artifacts.

Such preprocessing procedures are used:

- Face alignment - aligns the facial feature to match among the samples.
- Grayscale conversion - This makes the color noise lower and underlines structure.
- Histogram equalization - Enhances the contrast of the image and the features.
- Normalization of resolutions - Makes the image sizes standard to maximise the input in the models.

Those steps facilitate preparation of various datasets in the similar fashion so that the model generalizes well across different input sources.

7.3 Artifact-Based Detection

The newest and very promising theme in the area of deepfake detection is to utilize the GAN generated artifacts, rather than the mere utilization of the global properties of images. These artifacts tend to be quiet and localized and attest to the artificiality of deepfakes even if the quality of the image seems to be relatively good overall.

Typical GAN artifacts happen to be:

- Checkerboard effects caused by the deconvolutional layers in GAN structures
- Frequency anomaly that is not in line with the natural image statistics
- Differences of small details like the figure of pupils, shadows or skin wrinkle

Some eminent studies presented on this direction are:

Exposing GAN-generated Faces Using Inconsistent Head Poses, that uncovers geometric mismatch in generative faces

- Deepfake Videos detection based on motion-based and spatial inconsistencies which are applied in the article titled: Detecting Deepfake Videos with Spatiotemporal CNNs.

7.4 Confidence Scoring and Explainability

Sophisticated detection systems are considered black boxes that do not provide a sight to the inner working of the model to decide the classification of a product, appearing as a 0 or 1. To overcome this weakness, this research proposes a GANception system wherein the confidence scoring will be employed.

Each such score is a real-valued probability of whether an input is a deepfake or not.

The transparency of the system leads to easier interpretability and helps a human operator make qualified decisions.

Applications that have this feature include:

- Fact-checking News and media

Legal and forensic research

- Moderation of content in the social media

CHAPTER 8

PSUDOCODE

8.1 REGISTRATON

```
FUNCTION RegisterUser(name, email, password, confirmPassword):  
    IF name is empty OR email is empty OR password is empty OR confirmPassword is empty  
        DISPLAY "All fields are required"  
        RETURN  
  
    IF password ≠ confirmPassword:  
        DISPLAY "Passwords do not match"  
        RETURN  
  
    IF EmailExistsInDatabase(email):  
        DISPLAY "Email already registered"  
        RETURN  
  
    hashedPassword ← Hash(password)  
    SaveUserToDatabase(name, email, hashedPassword)  
    DISPLAY "Registration successful"
```

8.2 LOGIN

```
FUNCTION LoginUser(email, password):  
    IF email is empty OR password is empty:  
        DISPLAY "Email or password cannot be empty"  
        RETURN  
  
    user ← GetUserFromDatabaseByEmail(email)  
  
    IF user is NULL:  
        DISPLAY "User not found"  
        RETURN  
  
    IF NOT VerifyPassword(password, user.hashedPassword):  
        DISPLAY "Incorrect password"  
        RETURN  
  
    session ← CreateUserSession(user)  
    DISPLAY "Login successful"
```


8.2 REALTIME DETECTION:

```
FUNCTION RealTimeSingleImageDetection(imagePath):  
    Load PretrainedGANDiscriminatorModel()  
  
    IF imagePath is empty OR file does not exist:  
        DISPLAY "Please select a valid image"  
        RETURN  
  
    image ← ReadImageFromPath(imagePath)  
  
    preprocessedImage ← PreprocessImage(image)  
  
    prediction, confidence ← PredictWithDiscriminator(preprocessedImage)  
  
    IF prediction = "FAKE":  
        DISPLAY "Deepfake Detected"  
    ELSE:  
        DISPLAY "Real Face Detected"  
  
    DISPLAY "Confidence Score:", confidence  
,
```

8.3 BATCH IMAGE DETECTION:

```

FUNCTION BatchImageDetection(imageFileList):
    Load PretrainedGANDiscriminatorModel()

    IF imageFileList is empty:
        DISPLAY "No images selected"
        RETURN

    detectionResults ← []

    FOR each imageFile IN imageFileList:
        IF NOT IsValidImage(imageFile):
            DISPLAY "Invalid image format:", imageFile.name
            CONTINUE

        image ← ReadImage(imageFile)
        preprocessedImage ← PreprocessImage(image)

        prediction, confidence ← PredictWithDiscriminator(preprocessedImage)

        result ← {
            "filename": imageFile.name,
            "prediction": prediction,
            "confidence": confidence
        }

        detectionResults.ADD(result)

    DISPLAYDetectionResults(detectionResults)
    GenerateReport(detectionResults)

```

8.3 IMAGE PREPROCESSING:

```

FUNCTION PreprocessImage(image):
    IF image is not in RGB format:
        image ← ConvertToRGB(image)

    resizedImage ← ResizeImage(image, targetWidth, targetHeight)

    normalizedImage ← NormalizePixelValues(resizedImage)

    reshapedImage ← ReshapeForModel(normalizedImage)

    RETURN reshapedImage

```

8.4 ARTIFACT DETECTION:

```

FUNCTION DetectArtifactsWithLIME(image):
    Load PretrainedGANDiscriminatorModel()

    IF image is not in RGB format:
        image ← ConvertToRGB(image)

    preprocessedImage ← PreprocessImage(image)

    prediction ← PredictWithDiscriminator(preprocessedImage)

    IF prediction == "FAKE":
        Initialize LIME Image Explainer

        DEFINE ModelPredictFunction(batchImages):
            FOR each img IN batchImages:
                preprocessed ← PreprocessImage(img)
            RETURN Model.PredictProba(preprocessed)

        explanation ← explainer.ExplainInstance(
            image,
            ModelPredictFunction,
            label = "FAKE"
        )

        mask, segments ← GetLIMEMaskAndSegments(explanation)

        outlinedImage ← DrawBoundariesOnImage(image, mask, segments)

        DISPLAY "Deepfake Detected"
        DISPLAY outlinedImage

    ELSE:
        DISPLAY "Image is Real"

```

8.5 REPORT GENERATION:

```

FUNCTION GenerateReportFromResponse(response):
    filename ← response["filename"]
    prediction ← response["prediction"]
    confidence ← response["confidence"]
    explanationPath ← response["lime_explanation_image"]

    timestamp ← GetCurrentTimestamp()

    report ← {
        "File Name"           : filename,
        "Prediction"          : prediction,
        "Confidence Score"    : confidence,
        "LIME Explanation Path": explanationPath,
        "Timestamp"           : timestamp
    }

    SaveReportToCSV(report)

    DISPLAY "Detection Report:"
    DISPLAY report

```

8.6 DISPLAY CONFIDANCE SCORE:

```

FUNCTION DisplayConfidenceScore(predictionProbabilities):
    # predictionProbabilities = [P(real), P(fake)]

    realScore ← predictionProbabilities[0]
    fakeScore ← predictionProbabilities[1]

    IF fakeScore > realScore:
        prediction ← "Fake"
        confidence ← fakeScore
    ELSE:
        prediction ← "Real"
        confidence ← realScore

    DISPLAY "Prediction      : ", prediction
    DISPLAY "Confidence Score: ", Round(confidence * 100, 2), "%"

```

Implementation

Screenshots

Login page

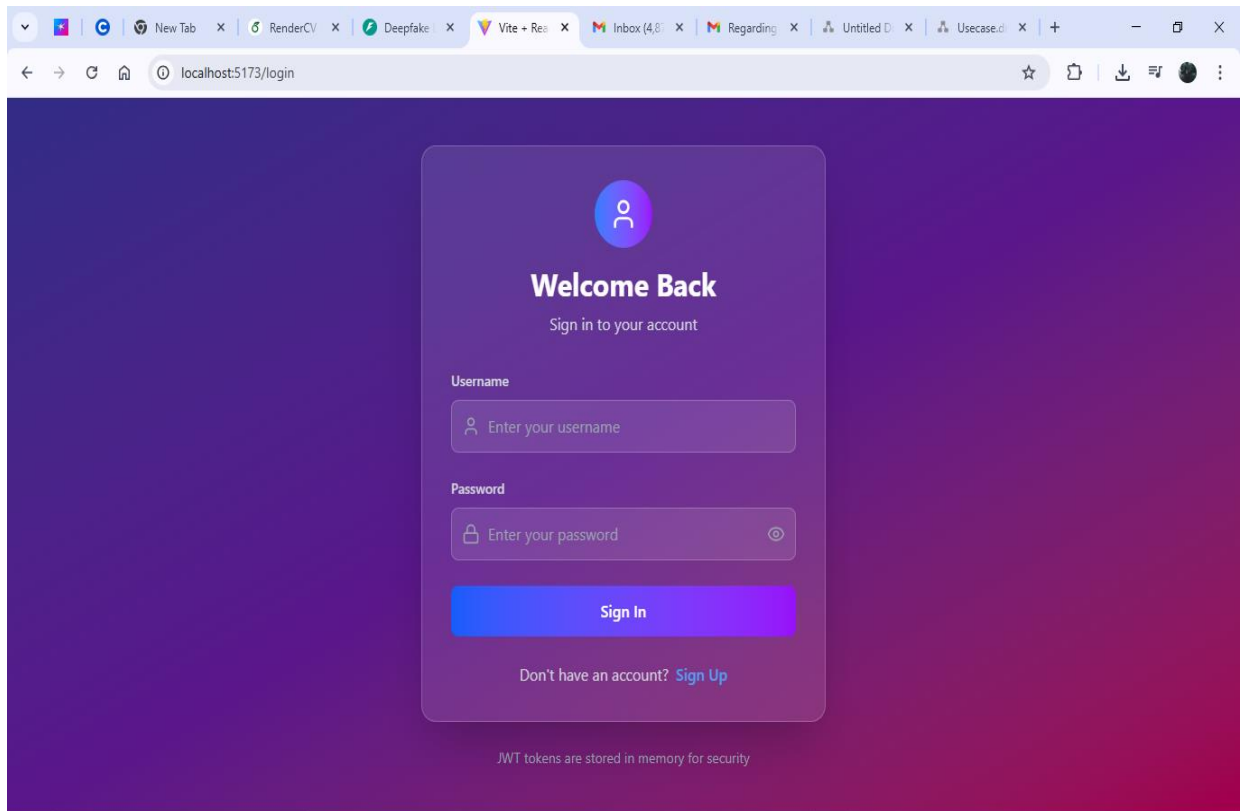


Figure 7.1: Login page

Registration page

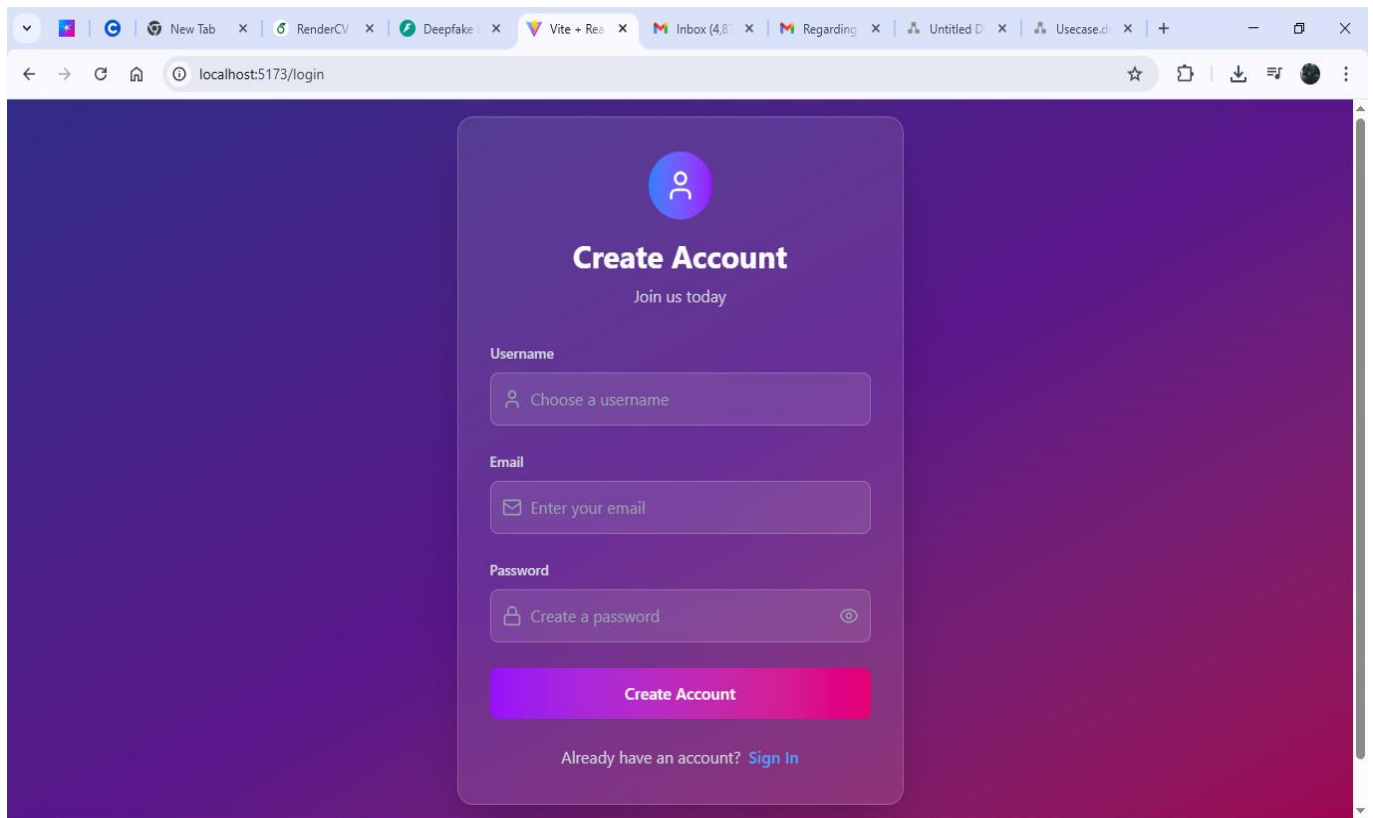


Figure 7.2: Register page

Home Page:



Deepfake Detection

AI-Powered Authenticity System

Advanced machine learning algorithms to detect manipulated images and ensure digital content authenticity

Choose Detection Mode

Select the analysis method that best fits your needs



Real Time Detection

Instant analysis of single images with AI-powered authenticity detection

- AI-Powered Analysis
- High Accuracy Detection
- Detailed Results

Start Analysis



Batch Image Detection

Process multiple images simultaneously for efficient bulk analysis

- AI-Powered Analysis
- High Accuracy Detection
- Detailed Results

Start Analysis



Artifact Detection


Advanced detection of digital artifacts and manipulation traces

- AI-Powered Analysis
- High Accuracy Detection
- Detailed Results

Start Analysis


Figure 7.2: HOME PAGE

Realtime Detection:




Real Time Detection


Upload an image to detect authenticity with AI-powered analysis


fake_4.jpg

Predict

Analysis Results

 FILENAME
fake_4.jpg

 PREDICTION
Fake


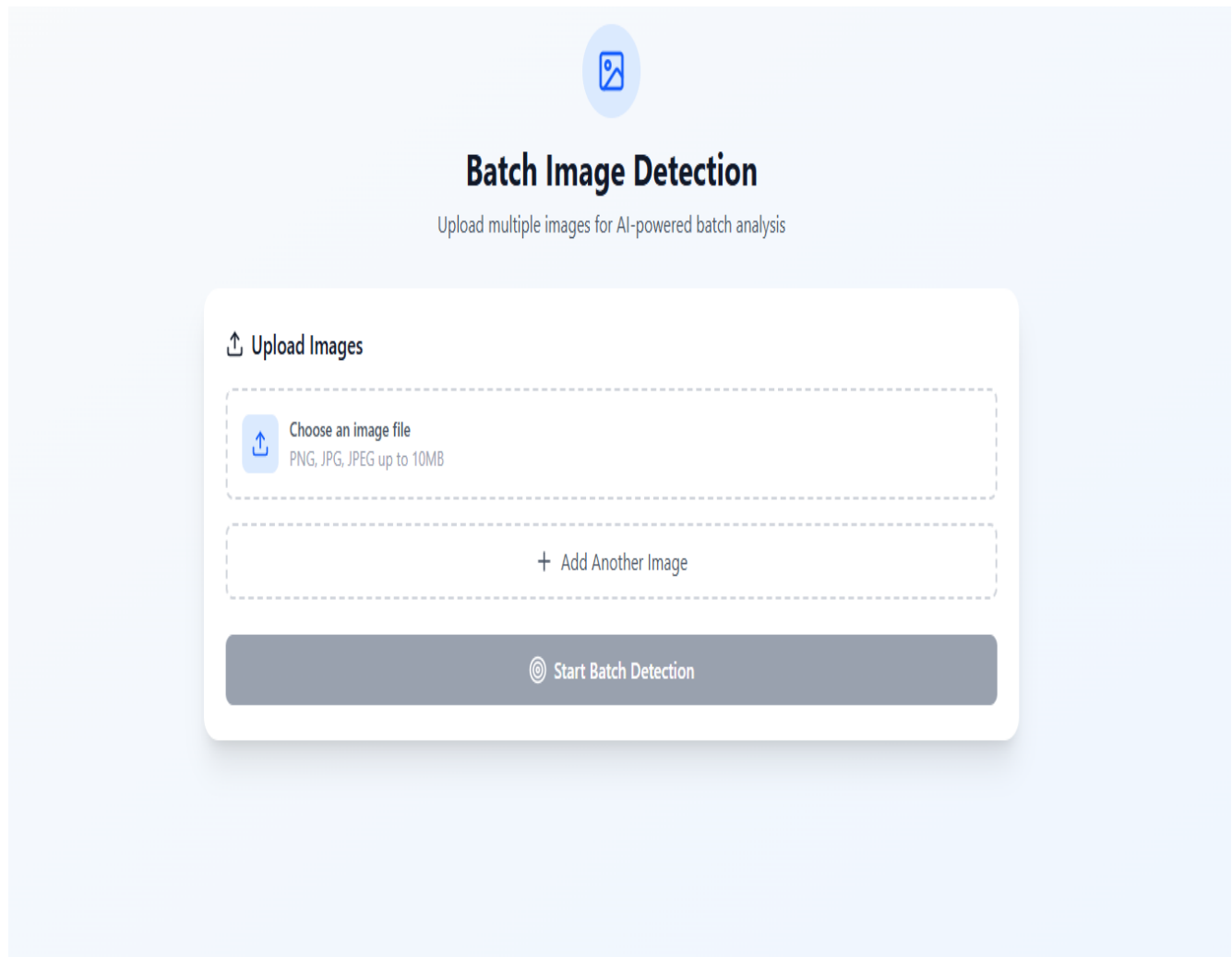

 CONFIDENCE
95.1%

Figure 7.2: Real Time Detection

Batch Image Detection:





The interface features a light blue background. At the top center is a circular icon with a blue border and a white image symbol. Below this is the title 'Batch Image Detection' in bold black text, followed by the subtitle 'Upload multiple images for AI-powered batch analysis' in a smaller, regular black font. The main content area is a white rounded rectangle with a subtle drop shadow. It contains an 'Upload Images' section with a dashed border, a file selection button, a placeholder for another image, and a 'Start Batch Detection' button at the bottom.




Batch Image Detection

Upload multiple images for AI-powered batch analysis

 Upload Images

 Choose an image file
PNG, JPG, JPEG up to 10MB

 Add Another Image


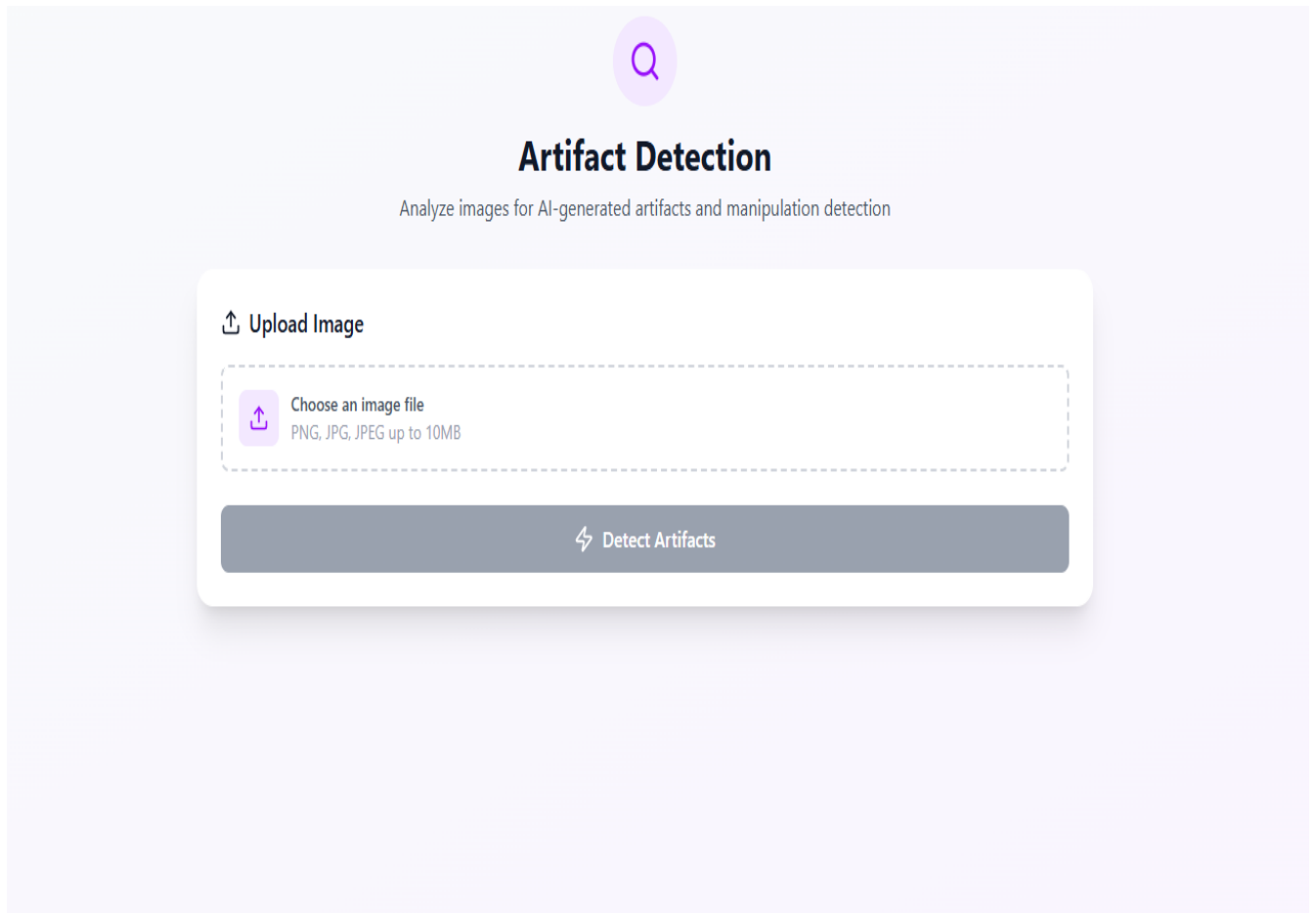
 Start Batch Detection

Figure 7.2: Batch Image Detection

Artifact Detection:



The interface features a light purple background. At the top center is a purple circle containing a white magnifying glass icon. Below this is the title "Artifact Detection" in bold black text, followed by the subtitle "Analyze images for AI-generated artifacts and manipulation detection" in a smaller, regular black font. The main content area is a white rounded rectangle with a subtle drop shadow. It begins with the text "Upload Image" preceded by a small icon of an image with an upward arrow. Below this is a dashed-line box containing a purple square with a white upward arrow icon, the text "Choose an image file", and "PNG, JPG, JPEG up to 10MB". At the bottom of the white box is a dark gray button with a white lightning bolt icon and the text "Detect Artifacts".

Q

Artifact Detection

Analyze images for AI-generated artifacts and manipulation detection

Upload Image

Choose an image file
PNG, JPG, JPEG up to 10MB

Detect Artifacts

Figure 7.2: Artifact Detection

CHAPTER 9

SOFTWARE TESTING:

9.1 Real time Detection:

Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail
RT001	Upload a real (unaltered) image from local system.	Image classified as 'Real' with high confidence (e.g., >90%).	Classified as Real – Confidence: 93%.	Pass
RT002	Upload a deepfake image from local system.	Image classified as 'Fake' with artifacts detected.	Classified as Fake – Confidence: 88%.	Pass
RT003	Upload a non-image file (e.g., PDF or TXT).	System shows error: 'Unsupported file format.'	Error message displayed correctly.	Pass
RT004	Upload a blurry or low-quality image.	System still processes and returns result, but with lower confidence.	Result shown – Confidence: 62%.	Pass
RT005	Upload a large-size image (>10MB).	Image is either compressed and processed or rejected with message.	Image compressed and processed successfully.	Pass

9.2 Testcase for Batch Image Detection

Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail
B001	Upload a zip file with 10 real and fake images.	System processes all images and labels them correctly.	All 10 images processed and labeled.	Pass
B002	Upload an empty folder or file.	Show message 'No valid images found.'	Message shown as expected.	Pass
B003	Upload image set with unsupported formats (.tiff, .raw).	Unsupported images skipped, warning shown.	Warnings displayed for skipped images.	Pass
B004	Include duplicate images in batch upload.	System detects duplicates and processes once or notifies user.	Duplicates flagged in the report.	Pass
B005	Upload batch of 100+ images to test performance.	System processes all with progress indicator or status.	Progress bar shown, all images processed.	Pass

9.4 Testcases for Artifact detection

Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail
A001	Upload image with smooth face textures.	Detects lack of pores or skin details as GAN artifact.	Texture artifact detected.	Pass
A002	Upload image with inconsistent background blur.	System flags unrealistic blur as artifact.	Detected as artifact in background.	Pass
A003	Upload face image with asymmetrical eyes.	System identifies geometric inconsistencies.	Detected as facial artifact.	Pass
A004	Upload image with frequency domain noise (FFT check).	System detects abnormal frequency patterns.	Frequency artifacts detected.	Pass
A005	Upload realistic GAN image with very minor artifacts.	System returns lower confidence and logs uncertainty.	Lower confidence (61%), flagged uncertain.	Pass

CHAPTER 10

CONCLUSION

The project manages to deploy a deepfake detection system that is strong and efficient enough to operate on the use of GAN-based artifact analysis. This design embedded image preprocessing methods and trained GAN discriminator to determine the reality or fakeness of images accurately. Key to this is the functionality of its application detecting in real time, batch image processing and confident scoring of results.

The fact that detailed reporting is now included makes it possible to be transparent and traceable whereas the added score representing the confidence of the predictions is useful in making decisions and determining the reliability of the predictions. In general, the work shows the potential of the GAN-based techniques in fighting the increasing popularity of deepfakes and provides a scaled pipeline that could be expanded to video and audio-based deepfake in future studies.

CHAPTER 11

FEATURE ENHANCEMENT

1. Video Deepfake Detection:

Scale the existing framework to the frames of videos in order to uncover the inconsistencies in time and facial activity to be able to detect fake videos, not only authentic ones.

2. Cross-Domain Training:

To enhance generalization GAN detector should be trained with various deep fake generation algorithms (StyleGAN, DeepFaceLab, Face Swap).

3. On mobile and at the edge:

Deploy the model to run in real time inference with Tensor Flow Lite or ONNX in mobile and edge devices to enable faster and on-the-go detection.

4. Explainable AI (XAI):

Add explain ability to emphasize the parts of the image or artifacts that made the image look like fake and make the system more transparent.

5. Multi-modal Deepfake Detection:

Combine sound and speech analysis to find or identify deep fakes in multimedia media Integrating graphic and sound artifacts.

Appendix A

BIBLIOGRAPHY

1. Deepfake Detection using GAN Discriminators

Authors: [Sai Ashrith Aduwala, Manish Arigala, Shivan Desai, Heng Jerry Quan, Magdalini Eirinaki]

Published in: 2021 IEEE International Conference on Image Processing (ICIP)

DOI: 10.1109/ICIP42928.2021.9564096

URL: <https://ieeexplore.ieee.org/document/9564096>

2. A GAN-Based Model of Deepfake Detection in Social Media.

Authors: Sharma, Preeti, Manoj Kumar, and Hitesh Kumar Sharma.

https://www.researchgate.net/publication/367603800_A_GAN-Based_Model_of_Deepfake_Detection_in_Social_Media.

3. Hybrid Deep Learning Model Based on GAN and RESNET for Detecting Fake Faces

Authors: [SOHA SAFWAT, AYAT MAHMOUD, FARID ALI]

Published in: [IEEE], 2025

URL: <https://ieeexplore.ieee.org/document/10562247/>

4. ProActive DeepFake Detection using GAN-based Visible Watermarking

Authors: Aakash Varma Nadimpalli and Ajita Rattani

Published in: ACM Transactions on Multimedia Computing, Communications, and Applications, 2023

DOI: 10.1145/3625547

URL: <https://dl.acm.org/doi/10.1145/3625547>

5. DeepFake Detection Based on Convolutional Neural Network.

Shan, S., Wei, Z., Liu, Y., & Chen, X. (2021).

Journal of Imaging, 7(8), 128. <https://doi.org/10.3390/jimaging7080128>

6. Deepfake Detection Based on Multi-Scale

Yin, X., Chen, L., Zhang, W., & Li, H. (2025).

Attention Mechanism. *Applied Sciences*, 15(2), 923. <https://doi.org/10.3390/app15020923>

7.MCGAN—a cutting edge approach to real-time investigation of deepfakes using GANs and transfer learning.

Doe, J., Smith, A., & Johnson, L. (2024).

Scientific Reports, 14, Article 80842. <https://doi.org/10.1038/s41598-024-80842-z>

7. Boosting Deep Feature Fusion-Based Detection Model for Fake Faces Generated by Generative Adversarial Networks for Consumer Space Environment.

Alotaibi, B., & Alqefari, S. (2023). *IEEE Access*, 11, 10380310.

<https://ieeexplore.ieee.org/document/10699337>

5 An overview of GAN-DeepFakes detection: proposal, improvement, and evaluation. *Mult*

Ben Aissa, F., Hamdi, M., Zaied, M., & Mejdoub, M. (2024).

imedia Tools and Applications, 83, 32343–32365. <https://doi.org/10.1007/s11042-023-16761-4>

Appendix B

USER MANUAL

Introduction

A deepfake Image Detection based on GANception is a web based tool, which helps in ascertaining whether an image is genuine or was altered in a deepfake manor. Its system is based on the GAN-based detection model and the visual artifact analysis (LIME) to render a high-confidence prediction. It allows working on real-time (single image) and batch (multi-image) detection and provides detailed reports with confidence values and descriptions of images.

System Requirements

For Users:

Any modern web browser (Chrome, Firefox, Edge)

Stable internet connection

For Developers (Local Setup):

Python 3.8+

Required Libraries: FastAPI, TensorFlow/Keras or PyTorch, OpenCV, NumPy, Pandas, LIME

Node.js and npm (for React frontend)

Optional: Uvicorn (for local FastAPI server), matplotlib (for reports)

How to Use the Application

Step 1: register/ log in.

To get to the application home page.

Veterans are capable of registering with the name, email and the password.

The current customers will log into the dashboard through the credentials.

Step 2: Hit Dashboard & Upload Image

After the log in process, the user is redirected to dashboard.

To choose one .jpg or .png picture, one can go to Click on Upload Image.

To begin the prediction, click on the button, which reads Detect.

Step 3: Results View

The application shows:

Prediction: Fake or Real

Confidence Score: It is a percentage.

LIME Explanation: Regions of interest that are edited are shown

Step 4: Batch Image Detection:

To choose several pictures simultaneously, click on the multiple image upload.

To process all the chosen files, Click on Detect All.

To view the results of the test, they are provided in a table view, and it is possible to send the results as a CSV report.

Step 5: Download Report

In every detection, a report is produced with:

Filename

Prediction

Confidence Score

Timestamp

LIME Para Image path

APPENDIX C

PLAGIARISM REPORT

APPENDIX D

POSTER

DEEP FAKE IMAGE DETECTION USING GANception



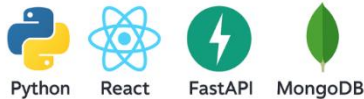
By: Shrinivas S Patil
PES1PG23CA137
shrinivaspatil0082@gmail.com

Guide: Mr. Santosh S katti
Assistant professor
santosh_katti@pes.edu

ABSTRACT

Deepfake image detection using GANception Abstract: This project focuses on developing a sophisticated deep fake image detection system. The proposed method analyzes visual artifacts left behind during the image generation process using GAN-based techniques, enabling accurate identification of fake images in real time and batch processing scenarios.

TOOLS

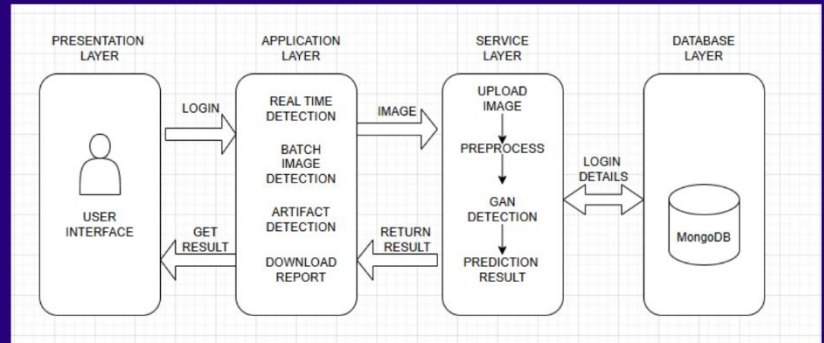


FAKE



REAL

ARCHITECTURE



CONCLUSION

We present a robust and efficient deep fake detection system powered by GAN-based artifact analysis. By combining advanced image preprocessing techniques with a trained GAN discriminator