IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Machine Learning Approaches For Detecting Duplicate Deepfake Videos Using CNN

[1] Mr. Vedant Wankhade, [2] Miss. Vaishnavi Dongre, [3] Miss. Pracheta Gulhane

[4] Miss. Achal Kharwade, [5] Prof. Shrikant Deshmukh

[1,2,3,4] UG Scholar [5] Professor

[1, 2,3,4,5] Computer Science & Engineering

[1, 2,3,4,5] Prof. Ram Meghe Collage of Engineering and Management, Badnera, Amravati, Maharashtra, INDIA

Abstract:-

DeepFake technology, driven by artificial intelligence, has transformed digital media manipulation, making it possible to create highly realistic synthetic images and videos. While this advancement offers benefits in entertainment and content production, it also poses significant ethical and security challenges, such as misinformation, identity fraud, and unauthorized impersonation. This research focuses on developing an advanced DeepFake detection system that leverages machine learning techniques to identify and counteract these risks.

The proposed system comprises two key components: the User Module, which allows user registration, media uploads for detection, and result tracking, and the Admin Module, responsible for content moderation and system oversight. Utilizing sophisticated deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), the system is designed to deliver high detection accuracy, real-time processing, and adaptability to evolving DeepFake techniques.

This study emphasizes the importance of automated DeepFake detection in maintaining digital authenticity and trust. It addresses the increasing threats posed by manipulated media across various sectors, including journalism, cybersecurity, and social media governance.

Keywords: -
DeepFake Detection, Machine Learning, Convolutional Neural Networks, Recurrent Neural Networks, AI in Cybersecurity, Digital Forensics, Face Forgery, Synthetic Media, GANs, Feature Extraction, Real-Time Detection, Image and Video Forensics, Adversarial Learning, Identity Theft Prevention, Misinformation Control, Ethical AI, Explainable AI, Content Moderation, Neural Networks, Data Augmentation, Pattern Recognition, Computational Photography, Flask Application, PyTorch, Deep Learning, Web-Based Detection System.

Introduction

The rapid evolution of artificial intelligence (AI) has significantly transformed digital media creation and manipulation, leading to the rise of DeepFake technology. These highly realistic synthetic images and videos, generated using advanced machine learning techniques like Generative Adversarial Networks (GANs), make it increasingly difficult to differentiate between authentic and manipulated content. While DeepFake technology has positive applications in fields such as entertainment, education, and creative media, it also introduces major ethical and security risks, including misinformation, identity fraud, and the unauthorized misuse of personal likenesses.

To combat these challenges, the development of effective DeepFake detection systems has become a crucial focus in AI-driven digital forensics. Machine learning-based detection methods, particularly those utilizing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown significant potential in identifying synthetic media by analysing facial anomalies, motion inconsistencies, and traces of image manipulation. These models are trained on extensive datasets containing both real and altered media, enabling them to recognize even the most subtle indicators of DeepFake generation.

This study examines the core technologies, methodologies, and obstacles associated with DeepFake face detection. It underscores AI's role in preserving digital authenticity and preventing the malicious exploitation of synthetic media. By evaluating the strengths and limitations of current detection techniques, this research seeks to advance the development of more precise, real-time, and scalable solutions to counter the growing threats posed by DeepFake technology.

II. Literature Survey

This paper provides a concise review of image generation techniques, categorizing existing approaches based on the type of input data used to generate new images, including images, hand sketches, layouts, and text. Additionally, it explores conditioned image generation, where a reference image is leveraged to produce the final output. The effectiveness of an image generation method is closely linked to the dataset used, necessitating the availability of large-scale datasets. To support this, the study summarizes widely used benchmark datasets for image generation and outlines evaluation metrics employed for assessing various methods. A comparative analysis is conducted based on these metrics and datasets, followed by a discussion on the key challenges associated with image generation.

Text-to-image generation has gained significant attention in fields like architecture and design, offering a practical approach to conceptualizing creative ideas through simple textual commands. A literature review reveals that AI-driven generative tools positively impact architectural stakeholders by expanding creative possibilities. However, this advancement also presents challenges for architects, as their role extends beyond design aesthetics to ensuring environmental sustainability. Additionally, users must carefully craft textual prompts to ensure AI-generated images accurately reflect their intended vision. As these technologies evolve, AI image generators hold the potential to streamline the architectural design process, reducing the time required to explore design alternatives. In the future, architects could focus more on validating and refining designs within digital environments before moving to actual construction. Early-stage digital environmental studies using AI-generated imagery could enhance inspiration, planning, and visualization for projects of varying scales, ultimately improving efficiency in the design process. However, the effectiveness of AI-driven design remains dependent on users' ability to communicate their ideas effectively through textual instructions.

Research by Chen et al. and parallel studies indicate that traditional GAN architectures are less effective compared to style-based models in terms of quality and controllability. Investigations into high-level attribute separation, stochastic effects, and the linearity of the intermediate latent space suggest that refining the latent space during training can enhance GAN synthesis. Furthermore, metrics such as average path length and linear separability could be leveraged as regularization techniques to improve model performance. Future research is expected to focus on direct modifications to the intermediate latent space to optimize control over generated outputs.

Advancements in StyleGAN have addressed several image quality issues, significantly improving state-of-the-art performance across multiple datasets. Some of these enhancements are more evident in motion, as demonstrated in supplementary visual materials. StyleGAN2 further refines attribution capabilities, allowing generated images to be more clearly linked to their source. Training efficiency has also improved—at a resolution of 1024^2, the original StyleGAN achieved a speed of 37 images per second on an NVIDIA DGX-1 with 8 Tesla

V100 GPUs, whereas an optimized configuration now trains 40% faster at 61 images per second. This improvement is attributed to optimized data flow, lazy regularization, and code refinements. Training times remain comparable, with StyleGAN2 requiring 9 days for FFHQ and 13 days for LSUN CAR datasets. The complete project, including exploratory work, consumed approximately 132 MWh of electricity, with 0.68 MWh allocated to training the final FFHQ model. The total computational effort equated to approximately 51 single-GPU years using Volta-class GPUs, as discussed in greater detail in Appendix F.

A novel generative flow model, termed MACOW, has been introduced, utilizing masked convolutional neural networks. By constraining local dependencies within a limited masked kernel, MACOW achieves stable and efficient training while enabling rapid sampling. Experimental results on both low- and high-resolution benchmark image datasets demonstrate that MACOW excels in density estimation and high-fidelity image generation, outperforming previous top-performing models in either likelihood scores or sample quality. A promising direction for future research involves extending MACOW to other data types, particularly text-based data, where flow-based generative models have not yet been extensively applied. Additionally, integrating MACOW with variational inference could facilitate the automatic learning of meaningful low-dimensional representations from raw data, opening new possibilities in generative modeling.

III. System Diagram

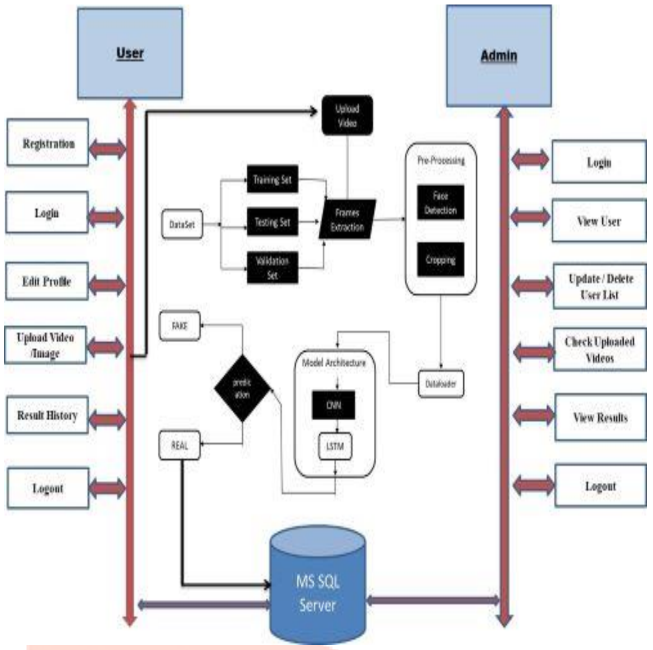


Fig: Proposed System

IV. Working Methodology

This project is a web-based application designed to detect deepfake images and videos. It integrates Flask as the web framework, PyTorch for deepfake detection using machine learning models, and MySQL for managing user data and detection results. The application enables users to register, log in, upload images/videos for analysis, and access their detection history.

Core Modules and Initial Setup

1. Importing Required Libraries

The application utilizes several essential libraries to perform its functions effectively:

Flask: A lightweight framework for building web applications.

SQLAlchemy: An Object-Relational Mapper (ORM) to interact with the MySQL database.

Bcrypt: Ensures secure password hashing and verification.

PyTorch & Torchvision: Used to define and execute the deepfake detection model.

Pillow (PIL): Handles image processing.

OpenCV: Manages video file processing and frame extraction for deepfake analysis.

os & datetime: Assists in file management and timestamp recording.

random & numpy: Ensures reproducibility through fixed random seeds.

2. Flask Application Setup

The Flask app is initialized with essential configurations:

UPLOAD_FOLDER: Defines the directory for storing uploaded images and videos.

ALLOWED_EXTENSIONS: Specifies valid file formats (e.g., .jpg, .mp4).

Secret_key: A randomly generated key that secures session data.

SQLALCHEMY_DATABASE_URI: Provides the connection string for MySQL integration.

Why Are Random Seeds Set?

Setting random seeds ensures consistent and reproducible results when executing the deepfake detection model. This is crucial for debugging and model validation. Since PyTorch, NumPy, and the random module have separate random number generators, their seeds must be set individually to maintain consistency.

Database Models

1. User Table

The User table manages registered user information, including:

2.id:

A unique identifier for each user username & email: Both are unique to prevent duplicate accounts.

3.password:

Stores securely hashed passwords using Bcrypt.

2. AnalysisResult Table

The AnalysisResult table records deepfake detection outcomes for uploaded media files. It includes:

1.user_id: Links the result to a specific user.

2.file_path: Stores the location of the uploaded file.

3.result: Stores the prediction outcome as either 'FAKE' or 'REAL'.

confidence_score: Represents the model's confidence in its prediction (ranging from 0.0 to 1.0).

Deepfake Detection

1. Model Definition

The application utilizes a modified Xception model, a Convolutional Neural Network (CNN) architecture widely used for image classification. To tailor it for deepfake detection, the original fully connected layer is replaced with a customized layer that outputs a single probability score.

This output represents the likelihood of the input image or video frame being a deepfake. The sigmoid activation function is applied to normalize the raw model output, ensuring the probability falls within the 0 to 1 range. This allows the system to determine whether the uploaded media is authentic or manipulated based on the model’s confidence level.

In binary classification tasks, a probability threshold of 0.5 is commonly used to distinguish between two classes. This means:

If the model's output probability is greater than 0.5, the input is classified as 'FAKE'

If the probability is less than or equal to 0.5, it is classified as 'REAL'

This threshold is chosen because it evenly splits the probability range (0 to 1), making it a logical decision boundary. However, in practical applications, this value can be adjusted based on factors such as the model’s performance, precision-recall balance, or real-world constraints.

In binary classification tasks, a probability threshold of 0.5 is commonly used to distinguish between two classes. This means:

If the model's output probability is greater than 0.5, the input is classified as 'FAKE'

If the probability is less than or equal to 0.5, it is classified as 'REAL'

This threshold is chosen because it evenly splits the probability range (0 to 1), making it a logical decision boundary. However, in practical applications, this value can be adjusted based on factors such as the model’s performance, precision-recall balance, or real-world constraints.

2. Upload and Detection

- Upload Image (/upload-image)

When a user uploads an image, the system performs the following steps:

Validates the file type (e.g., ensuring only .jpg, .png, etc., are allowed).

Preprocesses the image (resizing to 299x299, normalizing pixel values).

Runs the model multiple times to generate stable predictions.

Calculates the average confidence score and classifies the image as 'FAKE' or 'REAL'.

- Upload Video (/upload-video)

For video uploads, the system:

Extracts individual frames using OpenCV.

Processes each frame using the same image preprocessing techniques.

Runs the deepfake detection model on multiple frames.

Aggregates frame-level results to classify the entire video as 'FAKE' or 'REAL'.

Why Are Model Predictions Run Multiple Times?

Running the model multiple times on the same input ensures stability by reducing:

Hardware inconsistencies

Rounding errors

Random fluctuations in the model's forward pass

Averaging the predictions provides a more reliable confidence score, minimizing false positives/negatives.

3. History (/history)

This route displays the detection history of the logged-in user:

Fetches past analysis results from the MySQL database

Orders records by upload date for easy tracking

Allows users to review previous results and confidence scores.

Templates

The web application uses HTML templates (with Flask's Jinja2 templating engine) to render the following pages:

Signup/Login Pages: Contains forms for user registration and authentication.

Uses Bootstrap for styling and form validation.

Dashboard:

Displays user-specific content such as detection statistics, recent uploads, and account details.

Upload Pages: Provides file upload functionality for images and videos.

Displays detection results after processing.

History Page: Lists previous analysis results with timestamps.

Allows users to review and track deepfake detection history.

Additional Features

- **Error Handling:** Ensures invalid files (unsupported formats, corrupt data) are rejected.

Logs errors for debugging and troubleshooting.

- **Reproducibility:** Uses fixed random seeds to ensure that model predictions remain consistent across multiple runs.

- **Security:** Bcrypt is used to hash passwords, preventing direct storage of sensitive user data.

Implements secure session management to prevent unauthorized access.

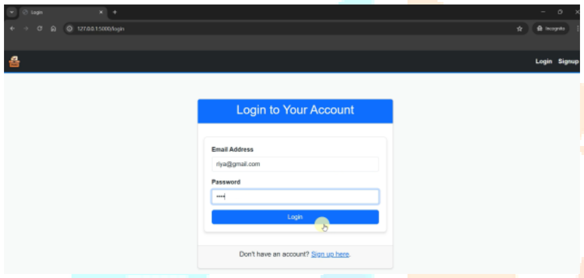
Main Application

The application is run with debugging enabled during development.

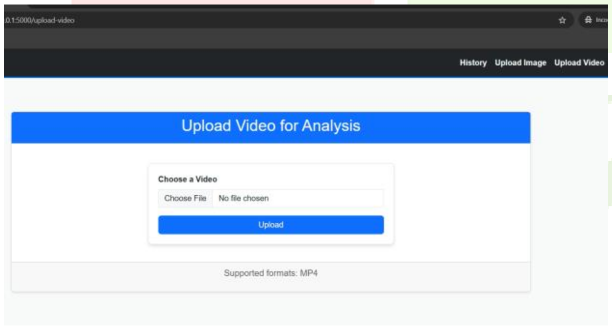
This allows for real-time feedback on code changes, making debugging easier.

Screen Shots:

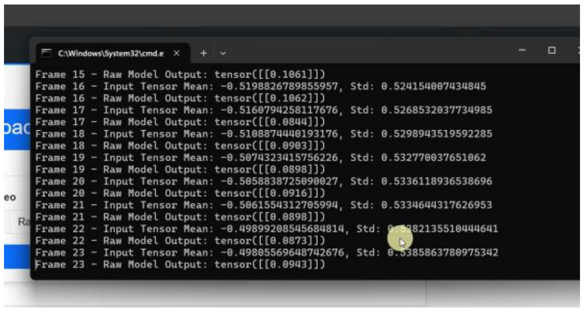
Login Page



Upload Video

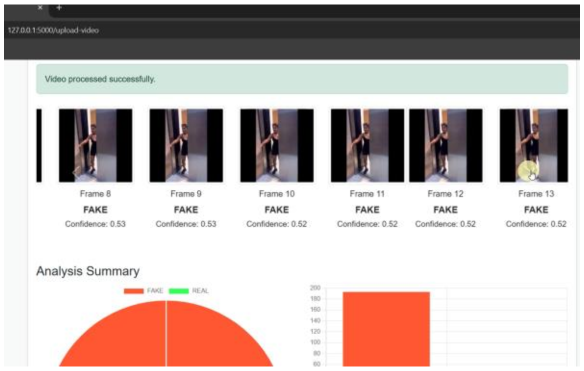

0.1:5000/upload-video
History Upload Image Upload Video
Upload Video for Analysis
Choose a Video
Choose File No file chosen
Upload
Supported formats: MP4

Processing



```
Frame 15 - Raw Model Output: tensor([[0.1061]])
Frame 16 - Input Tensor Mean: -0.5198826789855957, Std: 0.524154007434845
Frame 16 - Raw Model Output: tensor([[0.1062]])
Frame 17 - Input Tensor Mean: -0.5160794258117676, Std: 0.5268532037734985
Frame 17 - Raw Model Output: tensor([[0.0844]])
Frame 18 - Input Tensor Mean: -0.5108874440193176, Std: 0.5298943519592285
Frame 18 - Raw Model Output: tensor([[0.0903]])
Frame 19 - Input Tensor Mean: -0.5074323415756226, Std: 0.532770037651062
Frame 19 - Raw Model Output: tensor([[0.0898]])
Frame 20 - Input Tensor Mean: -0.5058838725090027, Std: 0.5336118936538696
Frame 20 - Raw Model Output: tensor([[0.0916]])
Frame 21 - Input Tensor Mean: -0.5061554312705994, Std: 0.5334644317626953
Frame 21 - Raw Model Output: tensor([[0.0898]])
Frame 22 - Input Tensor Mean: -0.49899208545684814, Std: 0.5382135510444641
Frame 22 - Raw Model Output: tensor([[0.0873]])
Frame 23 - Input Tensor Mean: -0.49805569648742676, Std: 0.5385863780975342
Frame 23 - Raw Model Output: tensor([[0.0943]])
```

Result



V. Conclusion

The advancement of DeepFake detection through machine learning is a crucial step in preserving digital authenticity and countering the growing risks posed by synthetic media manipulation. This study demonstrates the effectiveness of deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), in accurately distinguishing between real and manipulated content.

The proposed system, featuring user and admin modules, enables efficient media analysis, real-time detection, and systematic content moderation. Given the continuous evolution of DeepFake generation techniques, it is imperative to integrate robust detection frameworks within cybersecurity systems, media verification platforms, and social networks to ensure widespread protection.

Future research should focus on enhancing model resilience against adversarial attacks, improving detection accuracy for low-resolution and compressed media, and addressing ethical considerations surrounding DeepFake technology. As AI-generated content becomes more sophisticated, collaborative efforts between researchers, policymakers, and technology providers will be vital in mitigating the potential misuse of DeepFake technology while harnessing its creative potential responsibly.

Reference

[1]. Citation: Bhatt, C.M.; Patel, P.; Ghetia, T.; Mazzeo, P.L. Effective Heart Disease Prediction Using Machine Learning Techniques. Algorithms 2023, 16, 88. https:// doi.org/10.3390/a16020088

[2]. Vardhan Shorewala, Early detection of coronary heart disease using ensemble techniques
https://doi.org/10.1016/j.imu.2021.100655
Received 2 April 2021; Received in revised form 28 June 2021; Accepted 28 June 2021

[3]. Dr. Rakhi Waigi1 , Dr Sonali Choudhary2 , Dr Punit Fulzele3 , Dr. Gaurav Mishra Predicting The Risk Of Heart Disease Using Advanced Machine Learning Approach European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 7, Issue 07, 2020

[4]. Rupali Atul Mahajan1 , Dr. Balasaheb Balkhande2 , Dr. Kirti Wanjale3 , Dr. Abhijit Chitre4 , Tushar Ankush Jadhav5 , Dr. Sheela Naren Hundekari Enhancing Heart Disease Risk Prediction Accuracy through Ensemble Classification Techniques Submitted: 25/05/2023 Revised: 07/07/2023 Accepted: 26/07/2023

[5]. Mohan, S.; Thirumalai, C.; Srivastava, G. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. IEEE Access 2019, 7, 81542–81554. [CrossRef].

[6]. Waigi, R.; Choudhary, S.; Fulzele, P.; Mishra, G. Predicting the risk of heart disease using advanced machine learning approach. Eur. J. Mol. Clin. Med. 2020, 7, 1638–1645.

[7]. Gonsalves AH, Thabtah F, Mohammad RM, Singh G. Prediction of coronary heart disease using machine learning. Proceedings of the 2019 3rd international conference on deep learning. Technologies - ICDLT 2019. https://doi.org/ 10.1145/3342999.3343015..

[8]. Svetlana ulianova. Cardiovascular disease dataset. Retrieved from, https://www.kaggle.com/sulianova/cardiovascular-disease-dataset; 2019, january 01.

[9]. Latha CB, Jeeva SC. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. July 02, https://www.sciencedirect. com/science/article/pii/S235291481830217X; 2019