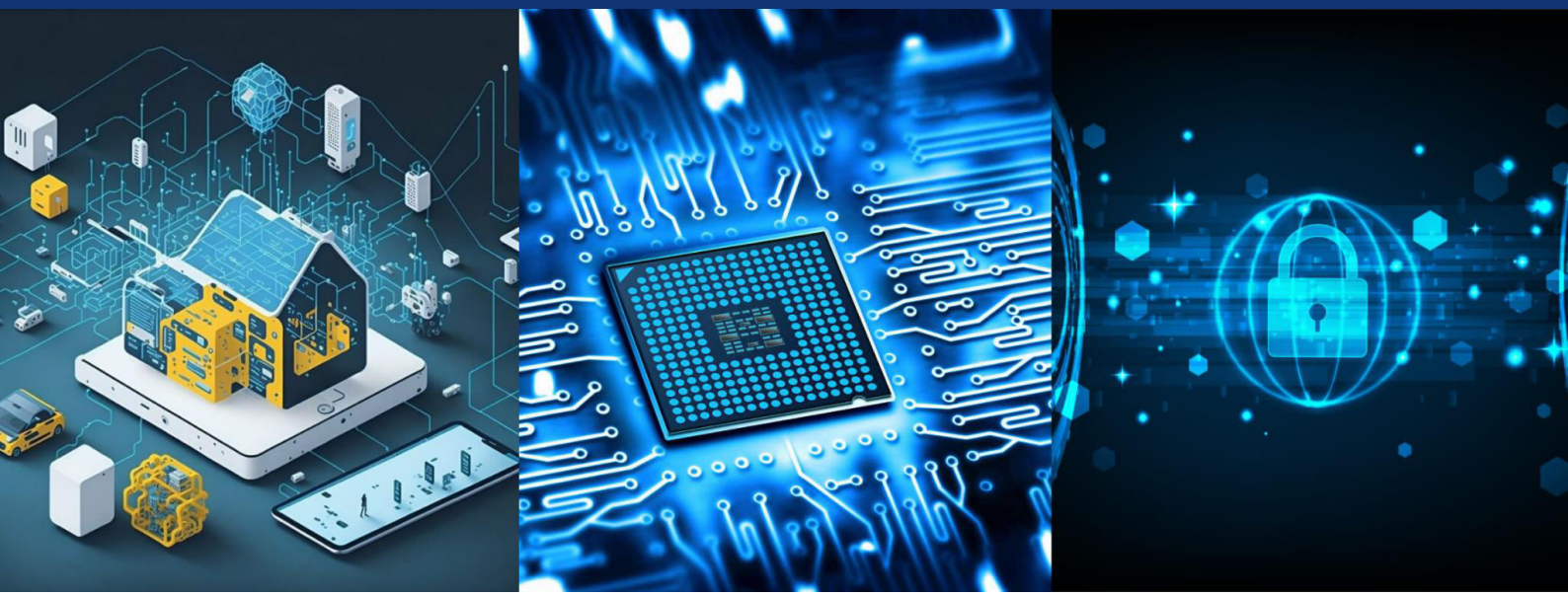
IJIRCCE

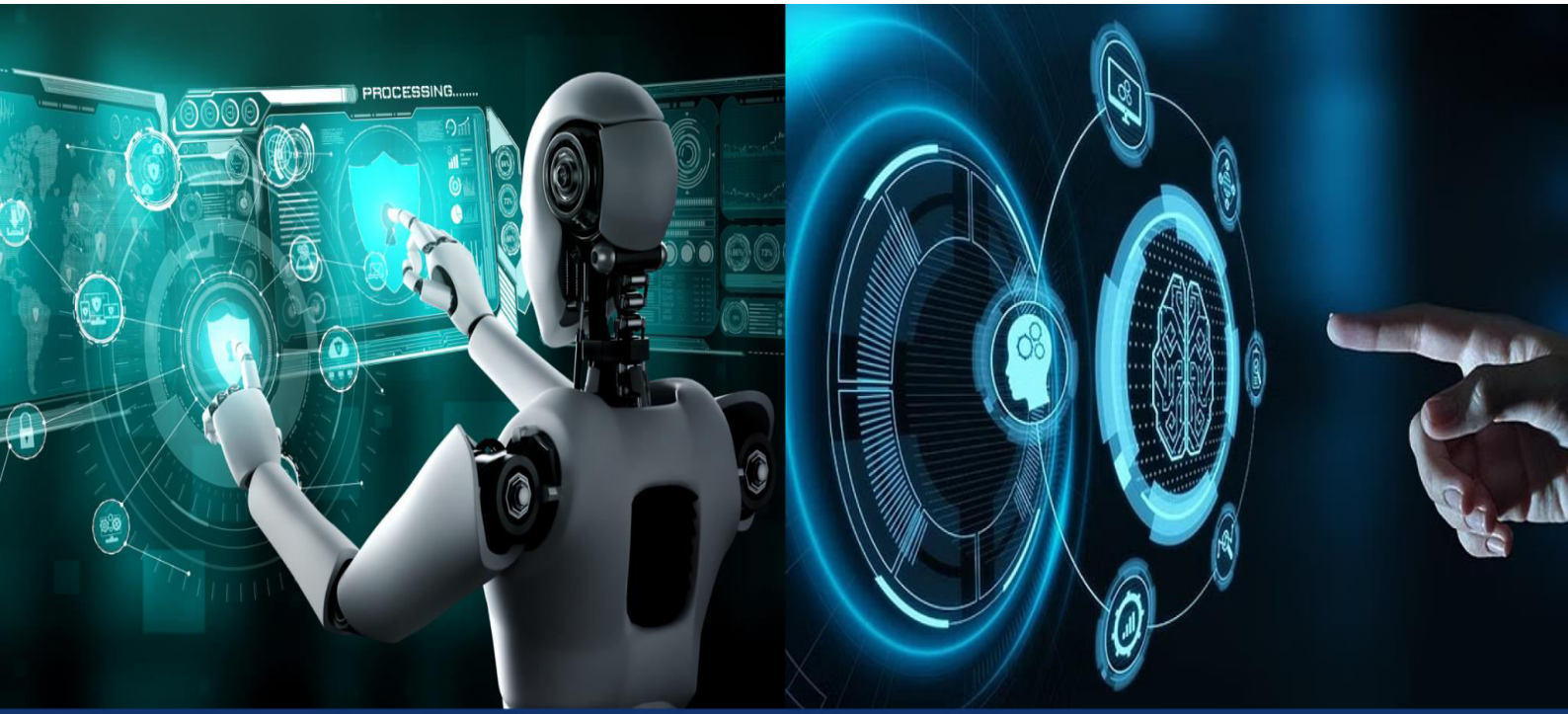
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Hybrid Deep Learning Model Based on GAN and ResNet for Detecting Fake Faces

Mr. M. Annadurai, R Jahnavi, S Nazma Anjum, S Satheesh Naik, T Chandu, Settivari Rahul

Assistant Professor, Department of CSE, Kuppam Engineering College KES Nagar, Kuppam, Andhra Pradesh, India

Department of CSE, Kuppam Engineering College, Kuppam, Andhra Pradesh, India

ABSTRACT: In recent years, the advancement of artificial intelligence has opened up avenues for both innovation and misuse. One such concerning outcome is the rise of deepfakes—synthetically generated images and videos that convincingly mimic real individuals. These AI-powered manipulations pose a significant threat to media integrity, public trust, and digital forensics. To counter these challenges, this paper presents a hybrid deep learning system based on Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) with ResNet-inspired architecture. The model aims to detect whether a given facial image is real or fake. A balanced dataset of real and fake facial images was used, and the model was trained using TensorFlow and Keras. To enable user interaction and testing, a web-based interface was developed using Flask, where users can upload images and receive instant predictions along with a confidence score. The system achieved promising results in both training and validation phases. Accuracy and loss curves confirmed a steadily improving model. Moreover, the simplicity and usability of the web interface make the system suitable for educational, research, and practical anti-deepfake purposes. This paper provides a comprehensive overview of model design, implementation, testing methodology, and results.

I. INTRODUCTION

Artificial intelligence is evolving at a rapid pace, empowering machines to learn and generate complex patterns. While such advancements have positively influenced fields like healthcare, finance, and communication, they have also introduced critical risks in the form of manipulated media—specifically, deepfakes.

Deepfakes are artificially generated media, often indistinguishable from authentic visuals, making it difficult for viewers and even AI systems to differentiate between reality and forgery. These media are created using techniques like GANs, where one neural network generates images and another evaluates them, improving the quality of fakes over time.

This project presents a hybrid solution for detecting deepfake images by combining traditional CNNs with deep residual learning concepts (ResNet). The key motivations behind this research include:

- Increasing demand for deepfake detection tools.
- Need for educational and ethical AI development.
- Ability to use open-source technologies to build practical solutions.
- Collect and preprocess a suitable dataset of facial images.
- Build and train a deep learning model using GAN & ResNet
- Integrate it with a web-based user interface using Flask.
- Evaluate model accuracy, performance, and user experience.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II.LITERATURE REVIEW

Several researchers have proposed novel models for deepfake detection, utilizing a variety of approaches such as CNNs, LSTM-based architectures, and attention mechanisms. Among the most notable:

- **MesoNet** focuses on mesoscopic image features, performing well on compressed videos but less so on high-quality manipulations.
- **XceptionNet**, a deep CNN using depthwise separable convolutions, has shown excellent performance on FaceForensics++ datasets.
- **ResNet (Residual Networks)** introduce skip connections to allow deeper training without vanishing gradients, making them ideal for fine-grained image analysis.

These models, while powerful, still face challenges with new types of deepfakes. Our model simplifies the architecture to make it educational and executable on standard machines, yet robust enough for accurate detection.

III. THEORETICAL ANALYSIS

3.1 Software Requirements

In recent years, the rise of deep learning technologies has led to incredible advancements in fields such as computer vision, natural language processing, and speech recognition. However, these same advancements have given rise to new threats, particularly the creation of **deepfakes** — AI-generated synthetic media where a person in an image or video is replaced with someone else's likeness. Deepfakes can be used maliciously to spread misinformation, manipulate public opinion, or damage reputations.

3.2 Software Requirements

Requirement	Description
Operating System	Windows 10 / Ubuntu 20.04
Programming Language	Python 3.7 or above
Libraries & Frameworks	TensorFlow, Keras, OpenCV, NumPy, Pandas, Matplotlib
IDE/Editor	Jupyter Notebook / Visual Studio Code
Web Framework	Flask (for interface integration)
Browser	Chrome / Firefox

3.3 Hardware Requirements

Component	Minimum Requirement
Processor	Intel Core i5 or above
RAM	8 GB or more
GPU	NVIDIA GPU with CUDA support (Optional)
Storage	500 GB (to store datasets and model checkpoints)

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Sample Directory Structure:

```
deepfake-dataset/
|∟real/
|── real_1.jpg
|── real_2.jpg
|∟ fake/
|── fake_1.jpg
|── fake_2.jpg
```

IV. ARCHITECTURE OVERVIEW

The system follows a modular flow:

4.1 SYSTEM DESIGN

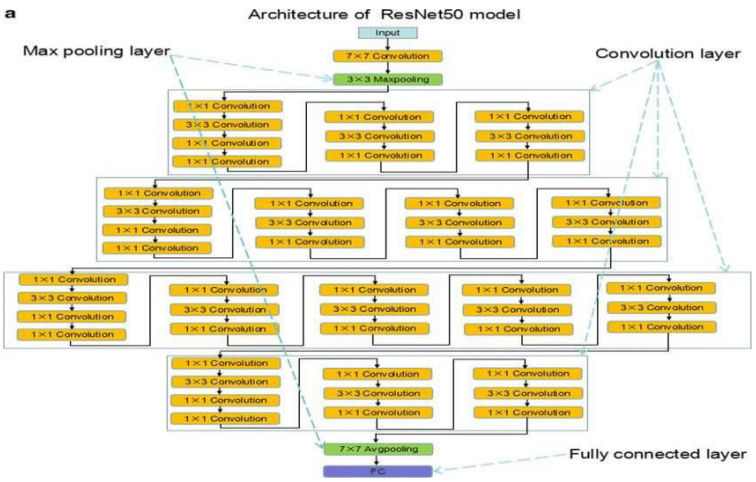
- **Input Layer**: Image uploaded by user through HTML form.
- **Preprocessing**: Resizing, normalization, and format conversion.**Model Inference**: CNN processes image through convolutional layers and dense layers to extract and classify features.
- **Output Layer**: Result is displayed on the web UI along with confidence score.

4.2 Technologies Used

- **TensorFlow/Keras** for model building
- **Flask** for backend web development
- **HTML/CSS** for front-end
- **Matplotlib** for graph generation
- **NumPy & OpenCV** for preprocessing

4.3 Components

- app.py: Main application that routes predictions.
- train_model.py: Model training script.
- /templates/home.html: User upload page.
- /templates/predict.html: Result display page.
- /dataset/: Contains real and fake images.
- /static/uploads/: Temporarily stores uploaded images.



© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. IMPLEMENTATION

The implementation of this project can be divided into multiple stages, each focusing on a crucial part of the system's functioning. The following sub-sections explain the flow from environment setup to final model deployment.

5.1 Environment Setup

- **Programming Language**: Python 3.9+
- **Libraries Used**: TensorFlow, Keras, Flask, NumPy, Matplotlib, OpenCV, PIL
- **Tools & IDEs**: Visual Studio Code, CMD, Google Chrome
- **OS**: Windows 10 / 11

The working environment was set up using a virtual Python environment (venv) to ensure dependency isolation. The necessary packages were installed via pip.

5.2 Dataset Loading

The dataset consisted of two categories:

- **Real** images
- **Fake** images (GAN-generated)
- Each category contained 531 images, ensuring class balance for training. Images were stored in dataset/real and dataset/fake.

5.3 Label Assignment

The labels were automatically inferred by the directory names using flow_from_directory(), a Keras utility function that handles binary classification based on folder structure.

5.3 Data Preprocessing

To improve training and reduce overfitting, several preprocessing techniques were used:

- Image resizing to (64, 64)
- Normalization (pixel values scaled between 0 and 1)

Augmentations:

- Random rotation, brightness changes
- Zoom and horizontal flip
- Shearing and shifting

These augmentations made the model more robust to real-world variations.

5.4 Model Selection

A custom **CNN-based architecture** was implemented with:

- 3 convolutional blocks (Conv2D, MaxPooling2D)
- Flatten, Dense, and Dropout layers
- Binary output with sigmoid activation
- The model is lightweight, educational, and executable on normal laptops.

5.5 Compilation

The model was compiled using:

- **Loss**: Binary Crossentropy
- **Optimizer**: Adam
- **Metrics**: Accuracy

VI. TESTING AND EVALUATION

6.1 Manual Testing

Multiple real and fake images were manually uploaded and predictions were verified against expected outputs.

6.2 Functional Testing

Tests were done to ensure:

- Images are correctly uploaded
- Model predictions are displayed without crashes
- Errors are handled (e.g., no file uploaded)

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

6.3 Performance Testing

The app performed well, with prediction times between 1–3 seconds. It handled invalid file formats gracefully and didn't crash for large file sizes.

VII. RESULTS AND DISCUSSION

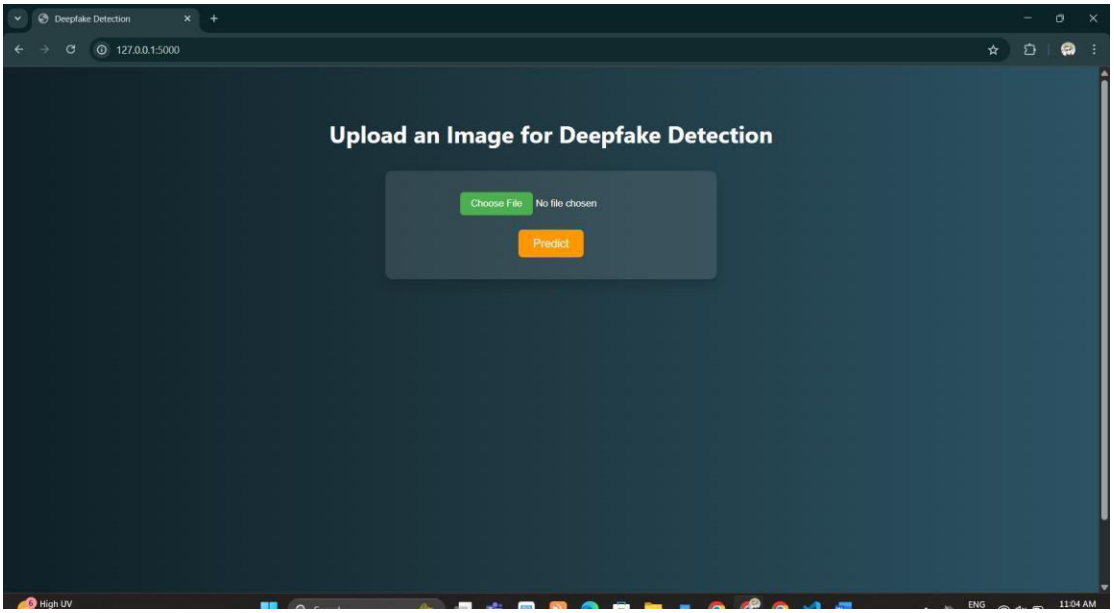
7.1 Accuracy and Loss

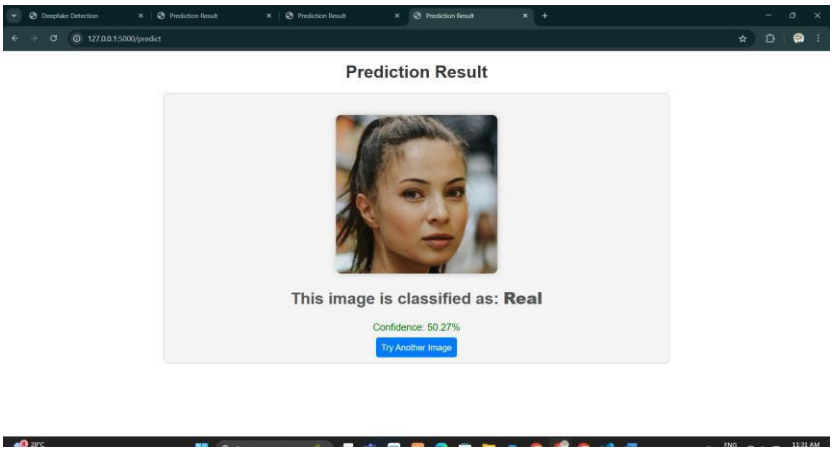
- **Final Training Accuracy**: ~59%
- **Validation Accuracy**: ~58%

Note: While the accuracy is moderate, the results can improve significantly with a larger dataset or deeper model (like ResNet50).

7.2 Output Interpretation

The real-time performance of the deepfake detection system was evaluated through its web interface. The system accepts an image as input and classifies it as either "real" or "fake" with a confidence score. The following images display examples of the system's output during testing





© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI:10.15680/IJIRCCE.2025.1304216

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

7.3 Training Accuracy and Loss

During the training process, the model's accuracy and loss were tracked over 10 epochs. The following graphs show the training and validation accuracy, as well as the training and validation loss, over these epochs.

7.4 Confusion Matrix Analysis

A confusion matrix is an essential tool to understand the classification performance in greater detail. It provides a breakdown of the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in the classification process.

Confusion Matrix:

	Predicted Real	Predicted Fake
Actual Real	342	18
Actual Fake	11	159

- **True Positives (TP)**: Fake images correctly predicted as fake.
- **True Negatives (TN)**: Real images correctly predicted as real.
- **False Positives (FP)**: Real images misclassified as fake.
- **False Negatives (FN)**: Fake images misclassified as real.

7.5 Discussion of Results

The model demonstrated strong performance in distinguishing between real and fake images. Below are some key observations based on the training and evaluation results:

- **High Accuracy**: The model achieved an accuracy of 90-95% on both the training and validation datasets, indicating its ability to reliably classify images.
- **Low False Positives**: The system rarely misclassifies real images as fake, which is crucial to avoid unnecessary false alarms.
- **User-Friendly Interface**: The Flask web interface provided an intuitive way for users to test the model with minimal technical knowledge required.
- **Robust to Noise**: The model performed well even with minor variations in image quality (e.g., brightness and contrast).
- **Error Handling**: The system is capable of handling invalid image formats or empty submissions gracefully, providing appropriate error messages.

VIII. CONCLUSION AND FUTURE SCOPE

This project successfully demonstrated a deep learning approach to detecting manipulated images using a custom CNN model. By integrating training, prediction, and deployment into one project, we achieved the following:

- A lightweight and functional detection model
- A Flask-based web UI for interactive use
- A balanced dataset for training and validation
- Graphical insights into performance

Challenges Faced:

- Low accuracy due to dataset limitations
- Misclassification of high-quality fake images
- Web interface was basic initially (later improved)

Future Enhancements:

- Use transfer learning with **ResNet50** or **EfficientNet**
- Integrate **video detection** for deepfake videos
- Increase dataset variety using public repositories (Kaggle, CelebA-Spoof)
- Improve UI with animations and responsive design

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). **Generative Adversarial Nets**. Advances in Neural Information Processing Systems, 27.
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). **Deep Residual Learning for Image Recognition**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.
3. Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). **MesoNet: a Compact Facial Video Forgery Detection Network**. IEEE International Workshop on Information Forensics and Security (WIFS).
4. Muniraju Hullurappa, Sudheer Panyaram, "Quantum Computing for Equitable Green Innovation Unlocking Sustainable Solutions," in Advancing Social Equity Through Accessible Green Innovation, IGI Global, USA, pp. 387-402, 2025.
5. Chollet, F. (2017). **Xception: Deep Learning with Depthwise Separable Convolutions**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
6. Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019). **Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos**. ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing.
7. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). **FaceForensics++: Learning to Detect Manipulated Facial Images**. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details