Transfer Learning for Efficient and Accurate Image Forgery Detection

A Project Report submitted in the partial fulfillment of the

Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P. Venkata Sai Teja (21471A0548)
K. Abhiram (21471A0501)
SK. Sharukh (21471A0556)

Under the esteemed guidance of

Dr. S. N. Tirumala Rao, M..Tech., Ph.D.,

Professor & HOD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier -1

NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "Transfer Learning for Efficient and Accurate Image Forgery Detection**" is a bonafide work done by the team** P. Venkata Sai Teja (21471A0548), K. Abhiram (21471A0501), SK. Sharukh (21471A0556) in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

PROJECT CO-ORDINATOR
Venkata Reddy Dodda, M.Tech., Ph.D.
Assistant Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "TRANSFER LEARNING FOR EFFICIENT AND ACCURATE IMAGE FORGERY DETECTION" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

P. Venkata Sai Teja (21471A0548)
K. Abhiram (21471A0501)
Sk. Sharukh (21471A0556)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao**, B.Sc., who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu**,

M. Tech., Ph.D., for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao**, M.Tech., Ph.D., HOD of CSE department and also to our guide **Dr. S. N. Tirumala Rao**, M.Tech., Ph.D., of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **D. Venkata Reddy**,M.Tech,(Ph.D) Assistant professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

P. Venkata Sai Teja (21471A0548)
K. Abhiram (21471A0501)
SK. Sharukh (21471A0556)

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems.

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in

Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

Problem analysis: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practic

Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature.

CO421.4: Design and Modularize the project.

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome Correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO’s and PO’s:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained

Name of the Course from Which Principles Are Applied in This Project	Description of the Task	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop a model for recognizing image manipulations using CNN and ELA	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the formof a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on detection of forged videos	PO4, PO7
C32SC4.3	The physical design includes website to check whether an image is real or fake	PO5, PO6

ABSRTACT

Visuals are at the heart of information dissemination, especially on social platforms, but their credibility is not easy to ascertain because of their manipulability. Most of the forgery detection research done is mostly tailored to only a certain type of forgery and this limits its applicability in real life scenarios. The purpose of this study is to explore the use of transfer learning techniques of deep learning in improving the detection of forgeries. It is also used for finding the differences in compression quality in pre-trained models by generating features from them. The method tests eight models for binary classification, and MobileNetV2 has the most accurate performance of 95% because of its lightweight nature and efficiency. Results of the experiment indicate that the accuracy and computational efficiency of the proposed method is satisfactory and surpasses the performance of existing state of the art approaches and can therefore be used in development for environments and scenarios where resources are limited

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

It is a fact that social media have changed the way people interact and carry on with their everyday lives. Social networking sites are a prominent media phenomenon nowadays, and have attracted a large number of people. Worldwide, the number of users now exceeds three billion.

In the Gulf region, growth in the number of active users has exceeded 66%. Saudi Arabia ranks seventh in the world in terms of social media use; more than 75% of its estimated 25 million people are active users of social media.

Socials are based on specific foundations that bring people together and empower them to express themselves, share their interests and ideas, and forge new friendships with others who share their interests.

Facebook, Twitter, and Instagram are among the most popular social networking sites of the day share their interests. Facebook, Twitter and Instagram are among the most popular social networking sites of the day. It is a widespread practice to share images online through social networking services such as Instagram.

At least 80 million images are currently shared via Instagram every day. Instagram enables users to take photographs, apply digital photographic filters, and upload the pictures to a website for social networking together with short captions.

People upload and share billions of pictures every day on social media. A huge number of people have become victims of photo forgery in this technological age. To put an end to this, all photographs exchanged via social media should be labeled as true or fake. Social media is a great platform for knowledge sharing and dissemination. Yet If there is no caution, people may be fooled and even induced by unintended false propaganda. Though most image editing using Photoshop is clearly evident, some of these images may indeed appear really due to pixelization and shoddy jobs by novices. In particular, in the Policy arena, edited images can break the credibility of a politician.

Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions. Deep learning methods used to automate tasks that typically require human intelligence, such as describing images or transcribing a sound file into text. For example, a human brain contains millions of interconnected neurons that work together to learn and process information. Artificial neurons are software modules called nodes, which use mathematical calculations to process data.

Three types of deep neural networks:

1. Multi-Layer Perceptions (MLP)
2. Convolutional Neural Networks (CNN)
3. Recurrent Neural Networks (RNN)

So, to overcome these types of forgeries combining ELA with CNNs can enhance the overall effectiveness of image forgery detection systems. Where ELA stands for Error level analysis and CNN stands for Convolutional Neural Network. ELA can provide initial insights into potential areas of manipulation, while CNNs can offer more detailed analysis and classification of the forged regions. This hybrid approach leverages the strengths of both techniques to improve the accuracy and reliability of image forensics tools in identifying digital forgeries.

1. The advantage of using this model is, it helps people to verify the authenticity of images, preventing the spread of misinformation and fake news.
2. Images are likely to be real (unforged) or fake (forged).
3. The model is even used to predict the image whether it is real or fake.

In this project using deep learning algorithms the researcher will attempt to propose a classifier model via a convolutional neural network (CNN) that is capable of taking advantage of knowledge to take an image from social media and then classify and detect it. The result of this proposed project will be helpful in monitoring and tracking social media content and in discovering fraud on social networking sites, especially in the field of images.

2. LITERATURE SURVEY

Gupta et.al [2], a comprehensive review discusses the role of deep learning, including CNNs, in image forgery detection. It covers various aspects, including dataset creation, network architectures, and performance evaluation metrics. It specifically discusses the role of CNNs and highlights the importance of feature learning for accurate forgery detection.

Additionally, a different study Mallick et.al [4] presented a novel method using convolutional neural networks (CNNs)to identify copy-move and splicing image forgeries. Three different models were used in the study: VGG16, VGG19, and ELA. Using preprocessing techniques, the photos were the model to be trained for the classification of authentic versus forged images.

2.1 Deep Learning

Deep learning is a subset of machine learning where artificial neural networks, inspired by the human brain's structure, learn from vast amounts of data to perform tasks such as image and speech recognition, natural language processing, and decision-making. Deep learning models consist of interconnected layers of neurons that process data hierarchically, enabling them to uncover complex patterns and make accurate predictions.

2.2 Some deep learning methods

Deep learning algorithms are often categorized as supervised and unsupervised.

- In **supervised deep learning**, the algorithm learns from labeled data, where each example in the training dataset is associated with a corresponding label or target. The goal is to learn a mapping from inputs to outputs based on the labeled examples. Many popular deep-learning tasks, such as image classification, object detection, and sentiment analysis, fall under supervised learning. Deep learning algorithms like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are often used for supervised learning tasks.

- In contrast, **unsupervised deep learning algorithms** learn patterns and structures from unlabeled data without explicit supervision. The goal is to discover the underlying structure or distribution of the data. Unsupervised learning techniques include clustering, dimensionality reduction, and generative modeling. Deep learning algorithms like autoencoders and generative adversarial networks (GANs) are commonly used for unsupervised learning tasks such as feature learning, data compression, and generating synthetic data.
- **Semi-supervised deep learning algorithms** combine labeled and unlabeled data for training, leveraging the advantages of both datasets. They typically start with a small amount of labeled data and a larger pool of unlabeled data. By exploiting the inherent structure and relationships within the unlabeled data, semi-supervised algorithms aim to improve model generalization and performance. Techniques such as self-training, co-training, and pseudo-labeling are commonly used to iteratively refine models using both labeled and unlabeled data.

- **Deep Reinforcement learning (DRL) algorithms** is a learning method, where Deep reinforcement learning combines deep learning with reinforcement learning principles to enable machines to learn to make decisions in complex environments. It has been successfully used in applications such as game playing, robotics, and autonomous vehicle control.

2.3 Applications of deep learning

1 Image Recognition and Classification

2 Speech Recognition

3 Natural Language Processing (NLP)

4 Recommendation Systems

5 Healthcare Diagnostics

6 Industrial Automation

7 Environmental Monitoring

8 Drug Discovery and Development

9 Financial Forecasting

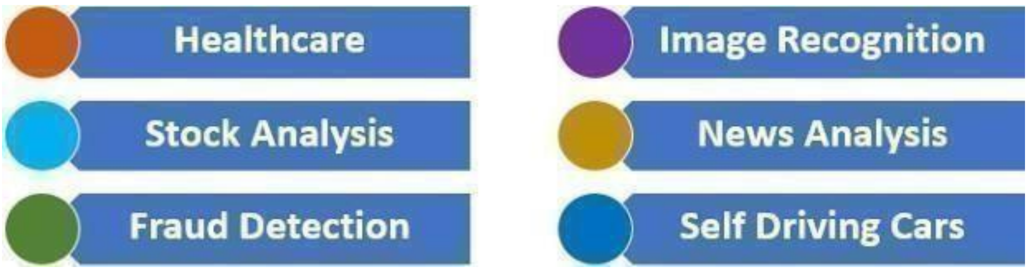

Fig 2.3.1 Applications of Deep learning

2.4 Prevalence of Image Forgeries:

Image forgery detection has become increasingly prevalent due to the widespread availability of image editing software and the ease of sharing digital content online. With these social media platforms and digital communication channels, the manipulation of images for deceptive purposes, such as spreading misinformation or creating fake news, has become a significant concern. Image forgery encompasses various techniques, including copy-move forgery, where a portion of an image is duplicated and pasted onto another area, and splicing, where parts of different images are combined to create a new one. Other techniques involve altering image attributes like brightness, contrast, and color saturation.

To combat image forgery, researchers and practitioners have developed sophisticated image forensics and forgery detection techniques. These methods often involve the use of advanced algorithms, including machine learning and deep learning approaches, to analyze images for signs of manipulation. They examine features such as inconsistencies in pixel values, noise patterns, and geometric distortions to identify forged regions within images. The prevalence of image forgery detection is driven by the increasing demand for reliable content verification and the need to maintain trust and authenticity in digital media. As the technology for creating and manipulating images continues to advance, so too does the development of innovative forgery detection techniques to maintain the integrity of visual information

2.5 Importance of deep learning in Image forgeries

The importance of deep learning in Image forgery detection is that utilizes artificial neural networks to identify whether an image has been morphed or manipulated. This is an important task in the digital age, as the proliferation of easily accessible photo editing software has made it easy for individuals to alter images for various purposes, such as spreading misinformation or altering evidence. Deep learning algorithms work by training on large datasets of images and learning to recognize patterns and features that are indicative of forgery. These algorithm scan then be used to analyze new images and determine whether they have been manipulated.

2.6 Implementation of deep learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum It is used for:

- Web development
- Software development
- Desktop GUI applications
- Game development

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Deep Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Deep learning and our project are:

- NumPy
- PIL
- OpenCV
- Theano
- TensorFlow
- Keras
- Py-Torch
- Pandas
- Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Deep Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

PIL is a popular library in Python used for opening, manipulating, and saving many different image file formats. With PIL, you can perform various image processing tasks such as resizing, cropping, rotating, filtering, and much more. PIL provides a convenient interface for working with images in Python, making it widely used in applications that involve image manipulation, computer vision, and machine learning tasks.

OpenCV provides a wide range of computer vision algorithms and tools, including deep learning-based approaches for image forgery detection. OpenCV can be used in conjunction with deep learning frameworks like TensorFlow and Py-Torch for preprocessing images and implementing custom detection algorithms, While not exclusively a deep learning library

Theano is a Python library that allows for efficient mathematical operations, including tensor manipulations and automatic differentiation, which are essential for deep learning. While its development has slowed down in recent years.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, TensorFlow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Deep Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best things about Keras is that it allows for easy and fast prototyping.

PyTorch is a popular open-source Deep Learning library for Python based on Torch, which is an open-source Deep Learning library which is implemented in C with a wrapper in Lua.
It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing (NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas is a popular Python library for data analysis. It is not directly related to Deep Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and
preparation. It provides high- level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.
Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Deep Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for
programmers for plotting as it provides features to control line styles, font properties,
formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

3. SYSTEM ANALYSIS

3.1 Existing System:

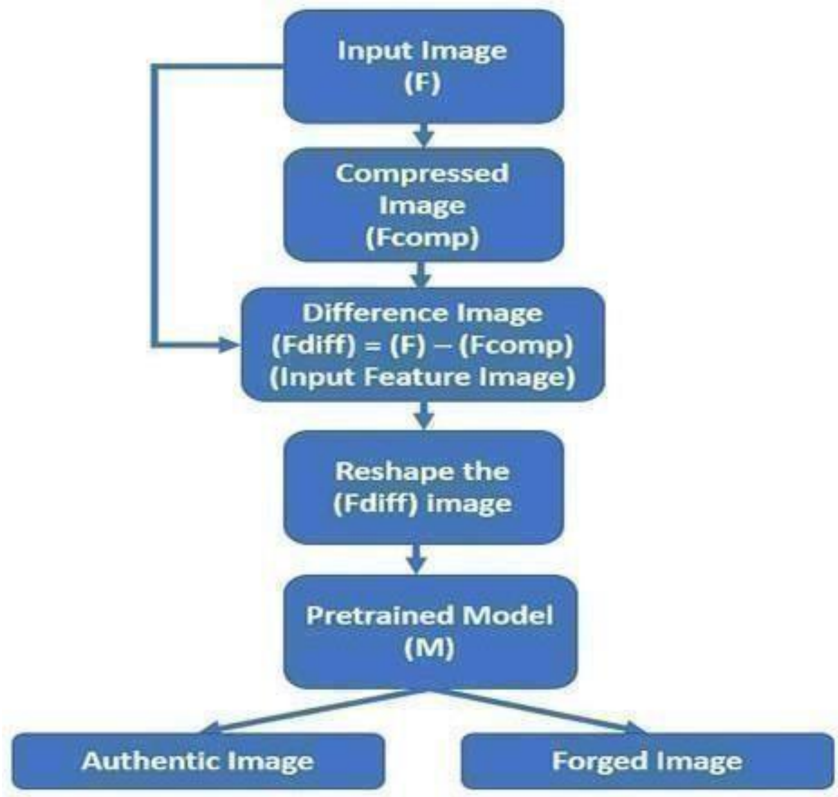

Fig 3.1.1 Flow chart of existing system

Detecting forgeries requires advanced forensic techniques and can be time consuming. Additionally, forgeries can manipulate perceptions, damage reputations, and have legal implications, impacting individuals and societies negatively. Deployment of the system enables automatic identification of forgeries, helping to maintain the integrity of digital content and mitigate the harmful consequences of image manipulation. Existing approach, works on eight different pre-trained models such as VGG16, VGG19, ResNet50, Reset101, ResNet152, MobileNetV2, exception and Dense Net. To enhance digital image forgery detection using deep learning techniques via transfer learning to uncover types of image forgery. A featured image as an input to the pre-trained model to train the model after removing its classifier and adding a new fine-tuned classifier. A comparison between eight different pre-trained models adapted for binary classification is done.

3.2 Disadvantages of the Existing System:

While effective, the existing system has several limitations:

- **High Computational Cost** – Using multiple pre-trained models demands significant resources, making the system costly.
- **Time-Consuming** – Training and fine-tuning deep learning models require substantial time, especially with large datasets.
- **Dependence on Pre-Trained Models** – The system may struggle with new forgery techniques not present in the original training data.
- **Limited Generalization** – It may fail to detect novel forgeries that deviate from learned patterns.
- **Overfitting Risk** – Fine-tuning on specific datasets can cause the model to perform well in training but poorly on unseen data.
- **Lack of Interpretability** – CNN-based models act as black boxes, making it difficult to understand classification decisions.
- **Dataset Dependency** – Model accuracy depends on dataset quality; biased data may reduce effectiveness.
- **Legal and Ethical Issues** – Results require additional validation before use in legal proceedings.

3.3 PROPOSED SYSTEM

The proposed system combines Elastic Net regularization (ELA) and Convolutional Neural Networks (CNN) to improve image classification. ELA helps select important features and prevents overfitting, while CNN extracts image details. This integration boosts accuracy, reduces overfitting, and enhances the ability to recognize complex images effectively.

Advantages:

1 Generates accurate and efficient results

2 Computation time is greatly reduced

3 Reduces manual work

4 Efficient

5 Automated prediction

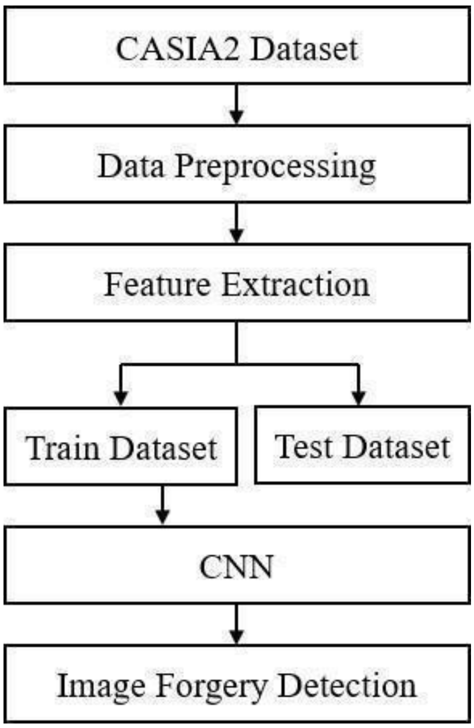

Fig 3.3.1 Flow chart of proposed System

3.4 FEASIBILITY:

3.4.1 Economic Feasibility:

The ELA-CNN model is economically feasible due to its optimized computational requirements, reduced training time, and minimal storage needs while maintaining high accuracy. By utilizing cloud-based GPU resources and transfer learning, it ensures cost- effective implementation without compromising performance. Future improvements may focus on lightweight architectures and hardware accelerations to further enhance economic sustainability.

3.4.2 Computational Resource Costs:

Initially, model training was conducted on a CPU, resulting in longer execution times. To enhance efficiency, a T4 GPU was utilized, significantly reducing training time. By leveraging cloud-based platforms such as Google Colab, the need for dedicated hardware is minimized, lowering implementation costs.

3.4.3 Training and Inference Optimization:

The model incorporates transfer learning with pre-trained architectures like MobileNetV2, ResNet50, and Dense Net, eliminating the need for training from scratch. This approach not only conserves computational resources but also accelerates inference time. The lightweight structure of MobileNetV2 further ensures real-time detection capabilities with minimal hardware requirements.

3.4.4 Storage and Memory Efficiency:

Unlike traditional deep learning models that require extensive memory and storage, the ELA-CNN model is optimized for efficiency. It reduces resource consumption, making it deployable on mobile and embedded systems without significant hardware upgrades.

3.4.5 Cost-Effective Deployment:

The model achieves an accuracy of **98.13%**, outperforming conventional methods while using fewer computational resources. This balance between performance and resource consumption minimizes operational costs, making it suitable for large-scale applications in forensic analysis,

cybersecurity, and social media monitoring.

3.4.6 Scalability and Long-Term Viability:

Trained on the CASIA2 dataset with ELA-based feature extraction, the model can be adapted to different datasets with minimal retraining. This reduces long-term maintenance costs and enhances economic viability by ensuring scalable deployment.

3.5 Technical Feasability:

the ELA-CNN model ensures that the system can be successfully implemented using the chosen technologies, programming languages, and frameworks. The goal is to achieve efficient processing, high performance, and seamless deployment across various platforms.

3.5.1 Technologies Used:

- **Programming Language:** Python – Chosen for its rich ecosystem in deep learning and image processing.
- **Frameworks & Libraries:**
- TensorFlow & Keras – For building and training the CNN model efficiently.
- OpenCV – Used for Error Level Analysis (ELA) to extract forgery-related features.
- NumPy & Pandas – Essential for numerical computations and data management.
- Matplotlib & Seaborn – Enables visualization of ELA transformations and model results.
- **Front-End:** Flask – A lightweight web framework for real-time image uploads and forgery detection results.

System Workflow

1. **Input:** The system accepts an image file (JPEG, PNG, etc.) for forgery detection.

2. **Processing**:

- ELA is applied to detect tampered regions.

- The CNN model (MobileNetV2, ResNet50, DenseNet) classifies the image as authentic or forged.

3. **Output**

- The system provides a classification result, a confidence score, and a visualized ELA image for interpretation.

3.5.2 Why These Technologies?

- Python & TensorFlow: Optimized for deep learning with extensive community support.
- Flask: Lightweight and effective for deploying AI models as web applications.
- Cloud GPU (Google Colab - T4 GPU): Reduces training time and eliminates the need for expensive hardware.

3.5.3 Performance & Scalability:

- Uses transfer learning for high accuracy with lower computational costs.
- Optimized for real-time detection, enabling fast inference on resource-limited devices.
- Scalable architecture, allowing adaptation to larger datasets with minimal retraining.

4. SYSTEM REQUIREMENT

4.1 Hardware Requirements

4.1.1 System Type: The project is implemented on a 64-bit operating system with an x64-based processor, which provides the necessary computational power for deep learning-based image forgery detection. This system architecture efficiently handles large datasets and complex neural networks, ensuring smooth execution of pre-trained models like VGG16, ResNet50, and Xception. The 64-bit operating system allows better memory management, enabling the use of more RAM, which is crucial for processing highresolution images and training deep learning models. Additionally, the x64-based processor supports parallel processing and GPU acceleration, optimizing model performance and reducing training time.

4.1.2 Processor: AMD Ryzen 5 5500U with Radeon Graphics

Why This Processor?

High Performance: The AMD Ryzen 5 5500U is a powerful x64-based processor designed for multitasking and efficient execution of deep learning models.

- Integrated Radeon Graphics: Enhances image processing and visualization, benefiting Error Level Analysis (ELA) and model inference.
- Efficient Multithreading: Improves processing speed, making training and Prediction tasks more efficient.

4.1.3 System Type: 64-bit Operating System, x64-Based Processor

- Supports advanced computing operations, allowing efficient execution of Pythonbased machine learning frameworks such as TensorFlow, Keras, and OpenCV.
- Ensures compatibility with modern AI frameworks and cloud-based GPU acceleration.

4.1.4 RAM: 16.0 GB

Why 16GB RAM?

- Allows smooth execution of multiple applications, including Anaconda, Flask based web services, and TensorFlow-based deep learning models.

- Provides efficient memory allocation for handling medium-to-large datasets without performance bottlenecks.

- Ensures optimal training and inference speed for the ELA-CNN model.

4.1.5 Storage:

- While total storage information is not provided, a minimum of 256GB SSD or
- **500GB HDD** is typically recommended for efficient performance.
- Ensures enough space for Python environments, deep learning libraries, and dataset storage.

4.2 Software Requirements:

- Operating System: Windows 11, 64-bit
- The 64-bit version of Windows 11 is required to ensure compatibility with modern software And hardware resources

- Why Windows 11? o Supports the latest security updates and performance optimizations. Offers compatibility with Python-based tools and frameworks.

- Provides a user-friendly interface for development and testing.

4.2.1 Coding Language: Python

Python is the primary programming language for development.

- Why Python?

Simple and easy to learn, making it ideal for beginners and professionals.

- Supports a vast range of libraries for data science, web development, and automation.
- Highly compatible with Flask and Anaconda for developing scalable applications.

4.2.2 Python Distribution: Anaconda, Flask

- Anaconda: A distribution of Python that includes essential libraries for data science and machine learning.
- Comes pre-installed with packages like NumPy, Pandas, and Jupyter Notebook. Helps in managing environments and dependencies efficiently.
- Flask: A lightweight web framework for building applications.
- Used for developing APIs and web-based applications.
- Simple yet powerful, allowing easy integration with databases and front-end frameworks.

4.2.3 Browser: Any Latest Browser (e.g., Chrome)

A modern web browser is required for accessing and testing web applications.

- Why Google Chrome or other latest browsers?
- Supports modern web technologies like HTML5, CSS3, and JavaScript.
  Provides developer tools for debugging and testing web applications.
- Ensures security and compatibility with Flask-based applications.

4.3 Software:

The ELA-CNN model for image forgery detection is built on a well-structured software environment for efficient deep learning, image processing, and web deployment. It runs on Windows 10/11 (64- bit), ensuring compatibility with deep learning frameworks.

Python is the primary language due to its strong support for machine learning and seamless integration with TensorFlow, OpenCV, Flask, and NumPy. Development is conducted in Jupyter Notebook (via Anaconda) for scripting, while Google Colab (T4 GPU) accelerates model training. The model is built using TensorFlow and Keras, with OpenCV handling Error Level Analysis (ELA) for forgery detection. NumPy and Pandas assist with data handling, while Matplotlib and Seaborn support visualization.

Deployment is powered by Flask, providing a user-friendly interface for image uploads and forgery detection. PIP manages dependencies, and GitHub ensures version control.
This software stack enables efficient training, accurate forgery detection, and real time processing, making it a scalable and cost-effective solution for forensics, cybersecurity, and social media monitoring. Future improvements will focus on optimizing performance and enhancing user experience.

4.4 Descrption:

The system runs on a 64-bit Windows 10/11 OS with an AMD Ryzen 5 5500U processor and 16GB RAM, ensuring efficient deep learning operations and smooth performance. Python is used as the core programming language, integrating TensorFlow and Keras for deep learning, OpenCV for Error Level Analysis (ELA), and NumPy, Pandas, Matplotlib, and Seaborn for data processing and visualization.
Development is done in Jupyter Notebook (via Anaconda), while Google Colab with a T4 GPU accelerates model training. Deployment is handled through Flask, providing a user-friendly web interface for forgery detection.

PIP manages dependencies, and GitHub ensures version control. This setup enables fast, accurate and scalable image forgery detection, making it ideal for forensic analysis, cybersecurity, and social media monitoring. Future improvements will focus on optimizing performance and enhancing user interaction.

5. SYSTEM DESIGN

5.1 Scope of the project

Social Media Focus: The research primarily concentrates on Instagram, a prominent social media platform known for its image-sharing capabilities. While the methods developed may have broader applications, Instagram serves as a specific case study due to the prevalence of image-related content.

Image-Centric Approach: The scope of this research is centered on images shared on social media. It does not encompass the analysis of text or video content. Images constitute a vital component of social media, and their impact on public perception makes them a primary focus.

Unwanted Content Mitigation: The research acknowledges the existence of unwanted content on Instagram, such as threats and forged images, which Can have societal and national security implications. Part of the scope involves addressing these issues to contribute to a safer and more reliable online environment.

Impact on Credibility: A crucial aspect of this project is the assessment of the impact of manipulated or fake images on the credibility of news and public trust in social communication means. The scope extends to understanding and mitigating the consequences of such content dissemination

5.2 System Architecture:

5.2.1 Analysis 0n Dataset:

The proposed approach utilizes the CASIA2 dataset, which contains a diverse collection of authentic and manipulated images for training and evaluation. Initializing the paths for real and fake image directories, and os.listdir() is used to lists their contents or to retrieve the list of files in each directory. Finally, it prints the count of images in both real and fake directories. CASIA2 has a total of 12617 images, where it consists of 7492original images and 5125 forged images.

	Authentic	Forged		Total	Size	Format
		Copy-move	Splicing			
Number of images	7491	3274	1849	12614	320x240 900x600	BMP, JPEG, TIFF
Total	7491	5123		12614		

Fig 5.2.1 Dataset Description

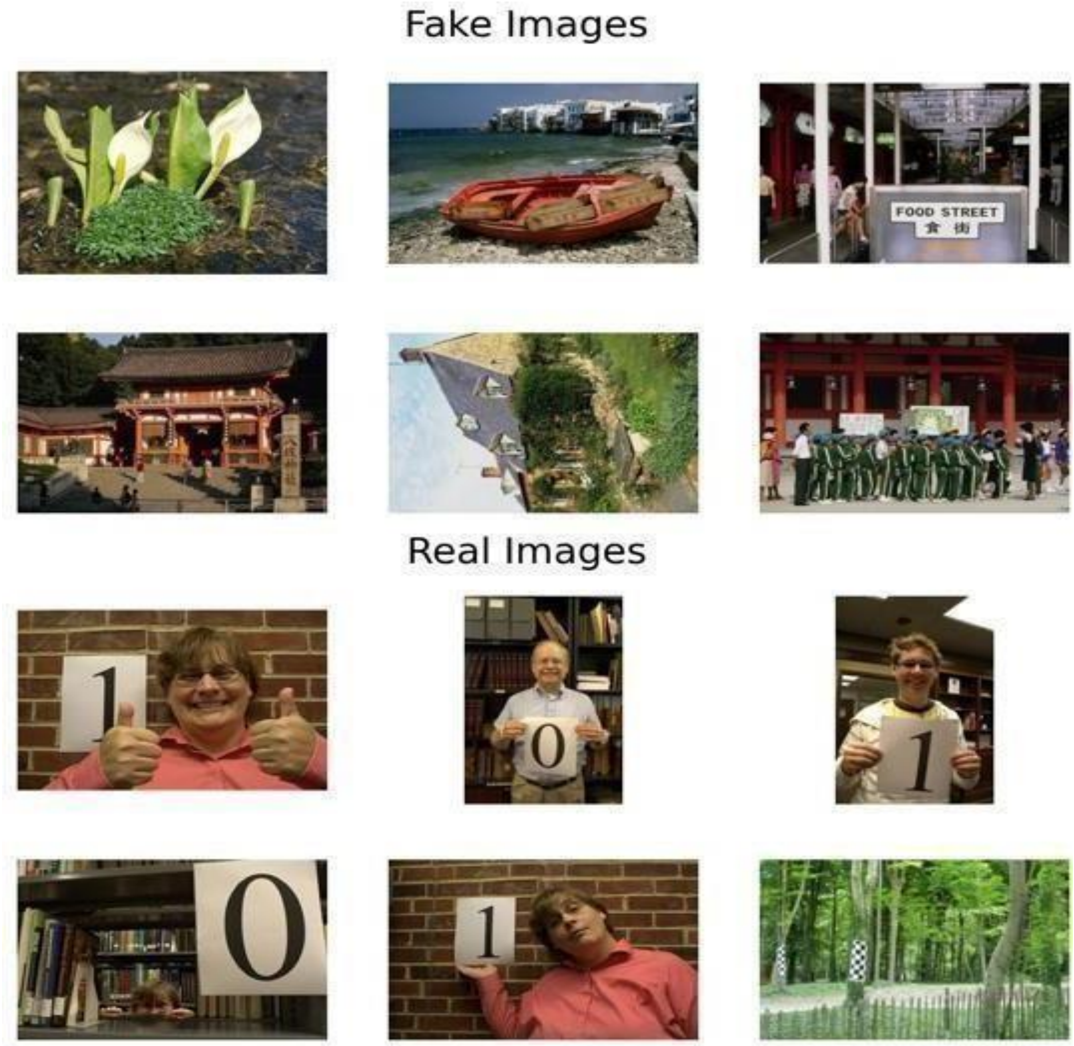

Fig 5.2.2 Group of real and fake image

5.2.2 Data Pre-processing:

Before feeding data to an algorithm, we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing, the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Deep Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. Preprocessing refers to the transformations applied to our data before feeding it to the algorithm. Preprocessing is the first step while creating the deep learning model.

During this preprocessing, images undergo Error Level Analysis (ELA). ELA is a technique that is used to detect regions in the image that may have been manipulated or altered, the compression levels of the modified regions often differ from those of the original areas.

Calculating the difference between the original and the compressed version. The difference is enhanced to visualize variations. ELA images highlight areas with differing compression levels, aiding in the detection of digital alterations. ELA highlights these discrepancies by accentuating the areas where compression levels deviate significantly. By extracting images from both the 'Au' (authentic) and 'Tp' (tampered) directories ELA is applied to each image. Overall, the code facilitates data preparation and labeling for a machine learning task, likely for training a model to classify image authenticity.

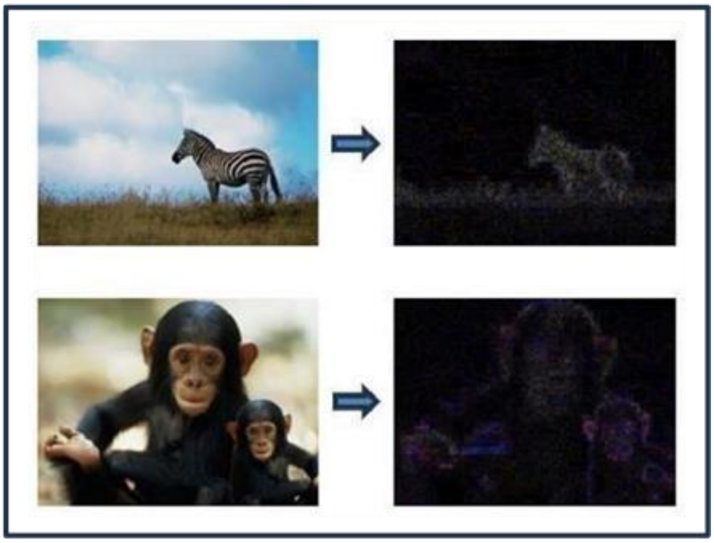

Fig 5.2.3 Data Preprocessing

5.2.3 Feature Extraction:

Feature extraction in the context of ELA involves identifying specific characteristics or patterns with in the ELA result that indicate potential areas of manipulation.

These features could include:

Inconsistencies in Pixel Intensity: Regions of the image that exhibit significantly different pixel intensities compared to their surroundings may indicate areas that have been digitally altered.

Block Artifacts: Compression artifacts, such as blockings or pixelation, may become more pronounced in regions that have been modified or edited.

Edges and Boundaries: Manipulated areas often have unnatural edges or boundaries where the alterations were made. These areas may appear smoother or more uniform in the ELA result due to the compression process.

5.3 Model Building:

Model building in the context of deep learning refers to the process of designing and constructing neural network architectures to solve specific tasks such as classification, regression, or generation. Deep learning models typically consist of multiple layers of neurons organized in a hierarchical fashion, enabling the model to learn intricate patterns and representations from the data.

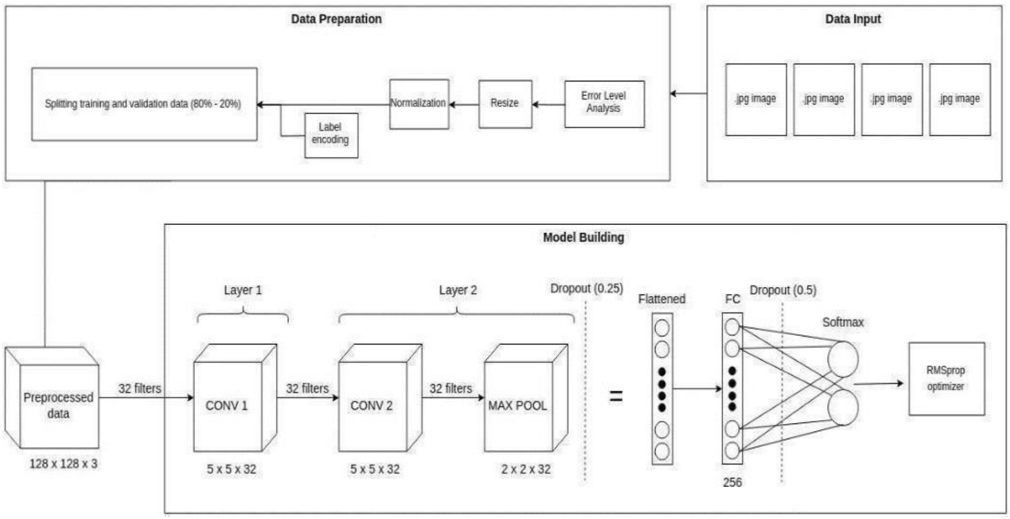

Fig 5.3.1 Overview of the model

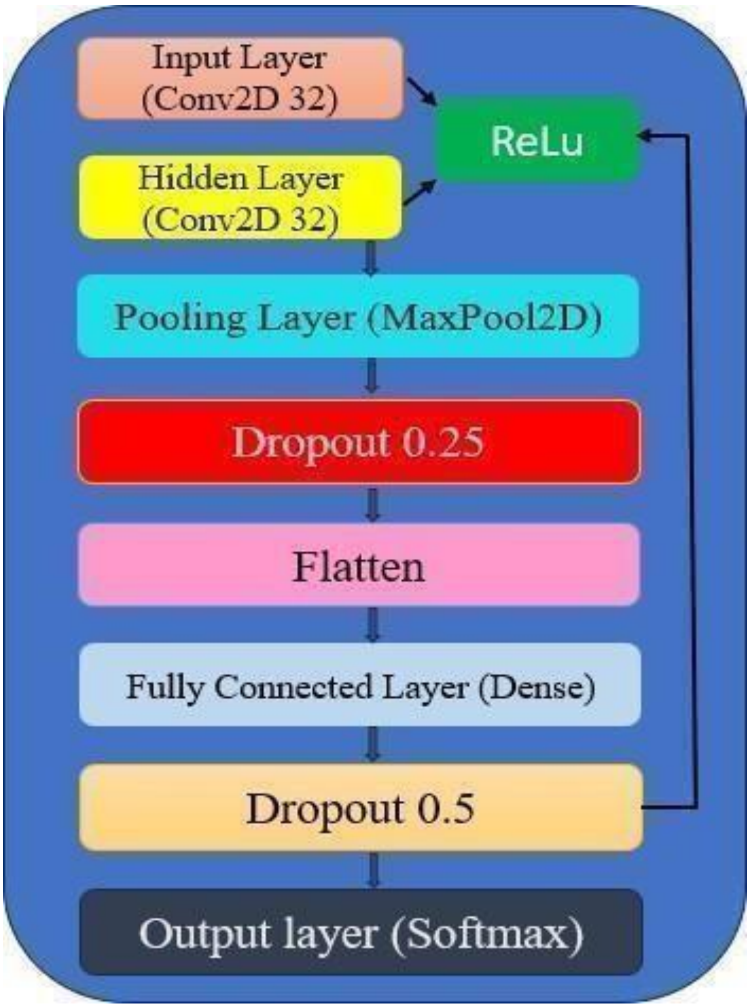

Fig 5.3.2 Model Architecture

Layers in Convolutional Neural Networks

Below are the Layers of convolutional neural networks:

Image Input Layer: The input layer gives inputs (mostly images), and normalization is carried out. Input size has to be mentioned here.

Convolutional Layer: Convolution is performed in this layer. The image is divided into perceptions(algorithm); local fields are created, leading to the compression of perceptions to feature maps as a matrix with size m x n.

Non-Linearity Layer: Here feature maps are taken as input, and activation maps are given as output with the help of the activation function. The activation function is generally implemented as sigmoid or hyperbolic tangent functions.

Rectification Layer: The crucial component of CNN, this layer does the training faster without reducing accuracy. It performs element-wise absolute value operation on activation maps.

Rectified Linear Units (ReLU): ReLU combines non-linear and rectification layers on CNN. This does the threshold operation where negative values are converted to zero. However, ReLU does not change the size of the input.

Pooling Layer: The pooling layer is also called the down sampling layer, as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. This layer ignores less significant data; hence image recognition is done in a smaller representation. This layer reduces overfitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions, and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.

Dropout Layer: This layer randomly sets the input layer to zero with a given probability. More results in different elements are dropped after this operation. This layer also helps to reduce overfitting. It makes the network to be redundant. No learning happens in this layer. This operation is carried out only during training.

Fully Connected Layer: Activation maps, which are the output of previous layers, is turned into a class probability distribution in this layer. FC layer multiplies the input by a weight matrix and adds the bias vector.

Output Layer: FC layer is followed by Soft max and classification layers. The Soft max function is applied to the input. The classification layer computes the crossentropy and loss function for classification problems.

Regression Layer: Half the mean squared error is computed in this layer. This layer should follow the FC layer.

6. TESTING AND CLASSIFICATION

The convolutional neural network was originally proposed by LeCun et al. for handwritten recognition has been successful in image identification, detection, and segmentation of the image. CNN has a high ability in large-scale image classification

A Convolutional layer and pooling layer are the most important layers on CNN.

Convolutional layer is used for extract feature by combining the image area with many filters. Pooling layer reduces the size of the output map of the convolution layer and prevents overfitting. Through these two layers the number of neurons, parameters, and connections is much less than there is a CNN model. This makes CNN more efficient compared with BP networks with similar layers.

The layers are determined by specific kernels, K along with the bias value on each kernel. It then operates by calculating the output image of the previous layer with each of the kernels. Convolution is a mathematical term that means applying a function to the output of another function repeatedly. The kernel moves from the top left corner to the bottom right. So, the result of the convolution of the image can be seen in the picture on the right. The goal of convolution in image data is to extract features from the input image. The convolution will produce a linear transformation of the input data according to the spatial information in the data.

A very popular approach to down sampling is to use pooling layers. Pooling layer usually deciphers the image (like 2x2) in the aggregation into a single unit. The most popular scheme for aggregation is the incorporation of the maximum value (max pooling). Subsampling is the process of reducing the size of the image data. In image processing, subsampling also aims to increase the position invariance of features. In most CNN, the subsampling method used is max pooling. Max pooling divides the output from the convolution layer into several small grids and then takes the maximum value of each grid to construct a reduced image matrix.

Layer is a layer that is usually used in the application of MLP and aims to transform the dimensions of data so that data can be classified in a linear. Each neuron in the convolution layer needs to be transformed into one dimensional data first before it can be inserted into a fully connected layer. Because cause data to lose spatial information and not reversible, fully connected layer can only be implemented at the end.

Applying CNN for fake image classification and original image converted into error level form on image. We know through the previous literature that CNN can achieve competitive performance and even better than humans in some visual problems, and we wanted to test CNN's ability to classify forgery image and original images via Error Level Analysis.

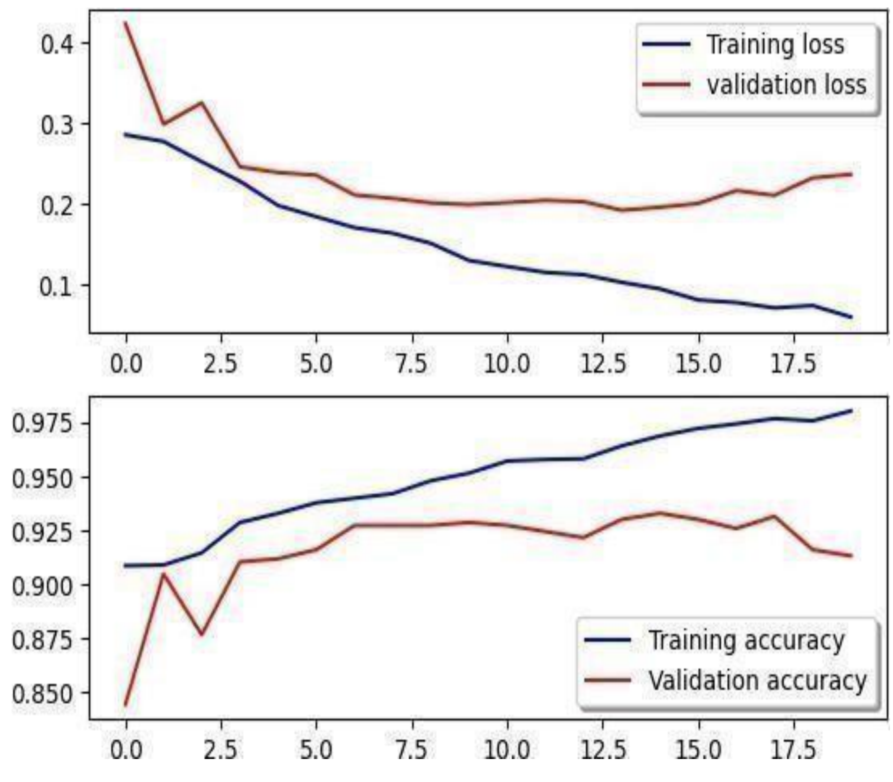

Fig 6.1 Training and validation graphs

The training accuracy curve shows how well the model is learning from the training data over time. As shown the curve generally increases as the model gets better at fitting the training data. On the other side, the validation accuracy curve shows how well the model is performing on a separate set of testing data that has not been seen during the training. The curve generally follows the training accuracy curve, but it may not increase as quickly or may plateau earlier. When the validation accuracy curve starts to decrease or diverge from the training accuracy curve, it indicates that the model is overfitting the training data, and is not generalizing well to new data.

The training loss curve shows how well the model is minimizing the training loss function over time. The curve generally decreases as the model gets better at fitting the training data. On the other hand, the validation loss curve shows how well the model is minimizing the loss function on a separate set of testing data that it has not seen during training. The curve generally decreases but does not decrease as quickly as the training loss curve or may plateau earlier. When the validation loss curve starts to increase or diverge from the training loss curve, it indicates that the model is overfitting the training data and is not generalizing well to new data, so the training process should stop immediately.

The accuracy and loss curves provide insights into how well the model is learning from the data, and whether it is generalizing well to new data. By monitoring these curves during the training process, the performance of the model can be improved.

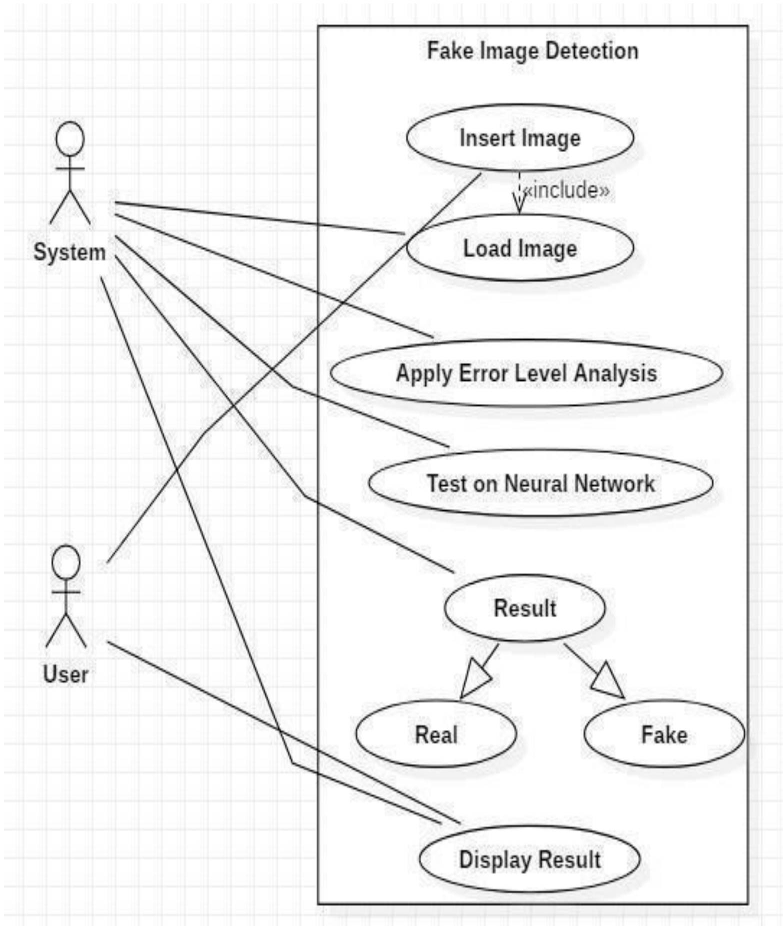

Fig 6.2 Use case diagram

6.3 Confusion Matrix

Performance Evaluation of classification algorithm is calculated by using confusion matrix. Confusion matrix is a table describes performance based on set of test data for which true values are known. Performance is calculated by considering actual and predicted class. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which true values are known.

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Fig 6.3.1 Confusion Matrix

A true positive (tp) is a result where the model predicts the positive class correctly.

Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class

A false positive (fp) is an outcome where the model incorrectly predicts the positive class. Where a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease.

TPR = tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

TNR = tn / (tn + fp) =1-FPR

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test result is negative.

FNR = fn / (fp + tn)

Fall-out or false positive rate (FPR)

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

FPR = fp/ (fp + tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.

Accuracy = (tp + tn) / (tp + tn + fp + fn)

F1 score

It is the harmonic mean of precision and sensitivityF1 = 2tp / (2tp+ fp + fn)

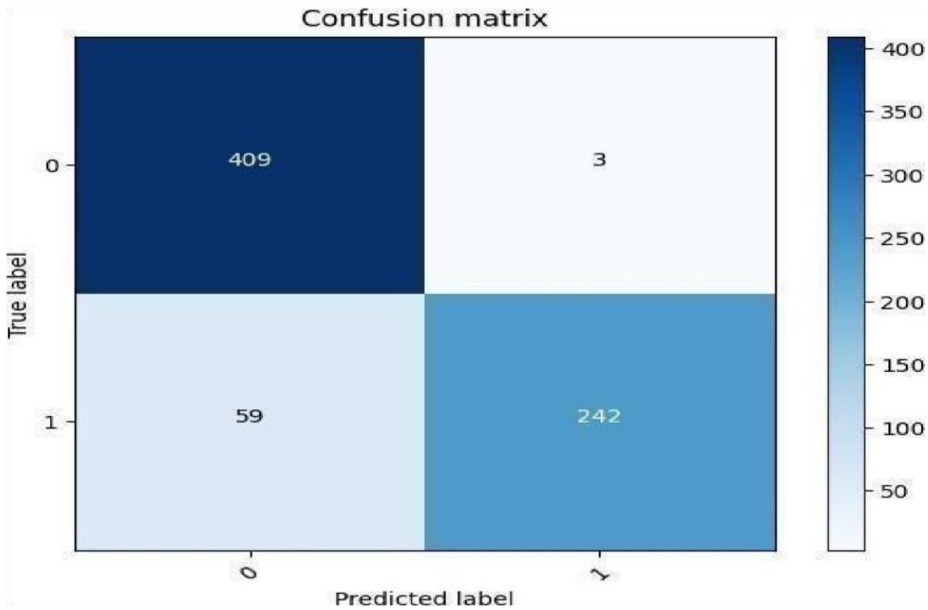

Fig 6.3.2 Confusion matrix for ELA &

7. DESIGN

The image processing and neural network workflow involves gathering and preprocessing images for training and testing, including steps like resizing, grayscale conversion, and error analysis with techniques like Error Level Analysis (ELA). Data is split into training and testing sets, and images are converted into pixel arrays for neural network processing. The model is then trained on the training set, and its performance is evaluated on the testing set. Metrics such as accuracy, precision, and recall are calculated, and the trained network can be saved for future use. Finally, a performance report is generated, and results are visually displayed to understand the model's performance, including correctly and misclassified images.

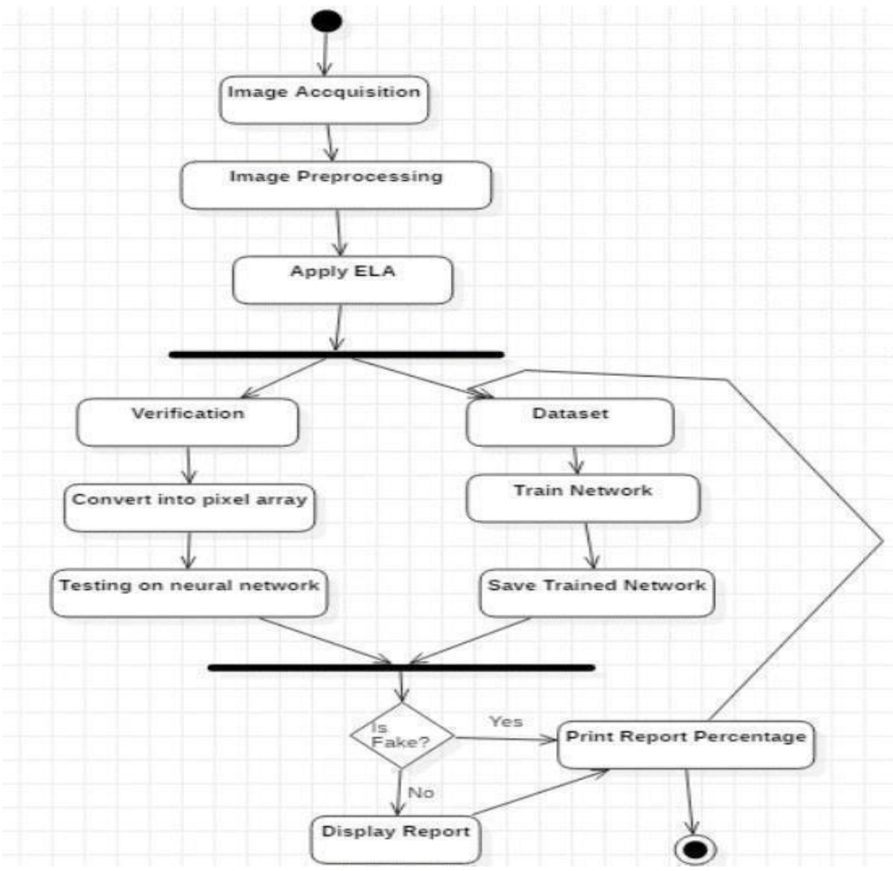

Fig 7.1 Design overview

8. IMPLEMENTATION

Using ELA&CNN:

```
import  os     import     itertools

 import numpy as np

import  matplotlib.as pyplot  %matplotlib inline np.random.seed

from sklearn.model_selection

 import train test_split

from  sklearn.metrics import confusion_matrix

from keras.utils Import to_categorical

from keras.models  import Sequential

fromkeras.layers import  Dense,  Flatten,  Conv2D,  MaxPool2D, Dropout
from  keras.optimizera  import Adam

from keras models import load_model

 from keras.preprocessing.image import Image Data Generator
from keras callbacks   import  Early Stopping
from  PIL  import  Image, Image Chops, Image Enhance
Real = '/content/drive/My  Drive/dataset/CASIA2/Au'  fake
='/content/drive/My

Drive/dataset/CASIA2/Tp' real_path  =

os.listdir(real) fake_path = os.listdir(fake) print('Real_images :',
len(real_path)) print('Fake_images:', len(fake_path))
num_images_to_display=6

plt.figure(figsize=(12,6))
for i in range(num_images_to_display):
```

```
img_path=os.path.join(real,real_path[i])     img=Image.open(img_path)
plt.subplot(2, 3, i + 1) plt.imshow(img)
plt.suptitle("RealImages",fontsize=25) plt.axis('off') plt.show()
num_images_to_display=6 plt.figure(figsize=(12,6))   for i
inrange(num_images_to_display):          img_path      =      os.path.join(fake,
image_files[i])    img =
  Image.open(img_path) plt.subplot(2,  3, i + 1)     plt.imshow(img)
      plt.suptitle("Fake                    Images",fontsize=25)   plt.axis('off')
plt.show()


def convert_to_ela_image(path, quality):
temp_filename='temp_file_name.jpg'  ela_filename='temp_ela.png'
image=Image.open(path).convert('RGB')  image.save(temp_filename,
'JPEG', quality = quality)  temp_image = Image.open(temp_filename)
ela_image= mageChops.difference(image, temp_image)
extrema=ela_image.getextrema()   max_diff = max([ex[1] for ex in extrema])

                    if max_diff == 0:

                    max_diff  = 1



scale=255.0/max_diffela_image=Iela_image).enhanc     e(scale)          return
ela_image


real_image_path=r'/content/drive/MyDrive/dataset/CASIA2/Au/Au_ani
_00001.jp             g'            Image.open(real_           image_           path)
convert_to_ela_image(real_image_path,

90)
    Fake                                                                          image
path=r''/content/drive/MyDrive/dataset/CASIA2/Tp/Tp_D_NRN_S_N_ani10171_a
ni00001_1 24  58.j pg'
Image.open(fake_image_path)   convert_to_ela_image(fake_image_path, 90)
image_size = (128, 128)  def prepare_image(image_path):
return np.array(convert_to_ela_image(image_path,
```

```
90).resize(image_size)).flatten() / 255.0

X = [] # ELA converted images

Y =  [] #0 for fake 1 for real


Importrandompath='/content/drive/MyDriv e/CASIA2/Au'  for dirname, _,
filenames in os.walk(path):    for filename in filenames:    if
filename.endswith('jpg') or filename.endswith('png'):

full_path = os.path.join(dirname, filename)

X.append(prepare_image(full_pat

h)   ) Y.append(1) if len(Y) % 500 ==

0:

print(f'Processing  {len(Y)}   images') random.shuffle(X)

X = X[:1500]

Y=Y[:1500]                    print(len(X),len(Y))
path='/content/drive/MyDrive/CASIA2/Tp'        for dirname, _, filenames in
os.walk(path):
for   filename   in   filenames:                  if   filename.endswith('jpg')   or
filename.endswith('png'):
  full_path = os.path.join(dirname, filename)

 X.append(prepare_image(full_path))

 Y.append(0) if len(Y) % 500 == 0:
print(f'Processing {len(Y)} images')    print(len(X), len(Y))  X = np.array(X)

Y = to_categorical(Y, 2)

X = X.reshape(-1, 128, 128, 3)

#Train Test split with 80:20 ratio

X_train,  X_val,  Y_train,  Y_val  =  train_test_split(X,  Y,  test_size  =  0.2,
random_state=5)  X  =  X.reshape(-1,1,1,1)   print(len(X_train),  len(Y_train))
```

print(len(X_val), len(Y_val)) def build_model(): model= Sequential() model.add(Conv2D(filters = 32, kernel_size = (5, 5), padding = 'valid', activation = 'relu', input_shape = (128, 128, 3))) model.add(Conv2D(filters = 32, kernel_size = (5, 5),

padding = 'valid', activation = 'relu',

input_shape=(128, 128, 3)))

model.add(MaxPool2D(pool_size =(2,2)))

model.add(Dropout(0.25)) model.add(Flatten())

#transforms the 2D array into a 1D array model.add(Dense(256, activation = relu'))

model.add(Dropout(0.5))

model.add(Dense(2,activation=softmax))

return model

model=build_model() model.summary() epochs = 20 batch_size = 32 init_lr = 1e4 optimizer = Adam(learning_rate = init_lr) model.compile(optimizer = optimizer, loss = 'binary_crossentropy', metrics = ['accuracy']) early_stopping = EarlyStopping(monitor = 'val_acc', min_delta = 0,)

patience = 2, verbose = 0, mode = 'auto') hist = model.fit(X_t rain, Y_train, batch_size=batch_size, epochs=epochs, validation_data=(X_val, Y_val), callbacks = [early_stopping]) model.save('model_casia_ru1.h5')

fig,ax=plt.subplots(2,1)

ax[0].plot(hist.history['loss'],color='b',label="Trainingloss")

ax[0].plot(hist.history['val_loss'], color='r', label="validation loss",axes

=ax[0]) legend=ax[0].legend(loc='best',shadow=True)

ax[1].plot(hist.history['accuracy'], color='b', label="Training accuracy") ax[1].plot(hist.history['val_accuracy'], color='r',label="Validation accuracy") legend = ax[1].legend(loc='best', shadow=True)

```
def plot_confusion_matrix(cm, classes,   normalize=False, title='Confusion
matrix', cmap=plt.cm.Blues):
plt.imshow(cm,interpolation='nearest',  cmap=cmap)  plt.title(title)
plt.colorbar() tick_marks
=        np.arange(len(classes))   plt.xticks(tick_marks,classes,r   otation=4   5)
plt.yticks(tick_marks, classes)  if normalize:
cm= cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

thresh =    cm.max()    /     2.

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
plt.text(j, i, cm[i, j],   horizontalalignment="center", color="white" if cm[i, j]
> thresh else "black") plt.tight_layout() plt.ylabel('True label')
plt.xlabel('Predicted label')

Y_pred = model.predict(X_val)

Y_pred_classes = np.argmax(Y_pred,axis = 1)

Y_true=np.argmax(Y_val,axis=1)
confusion_mtx=confusion_matrix(Y_true,Y_pred_classes
)plot_confusion_matrix(confusion_mtx,    classes = range(2))

class_names=['fake', 'real']  real_image_path=

'/content/drive/MyDrive/CASIA2/Au/Au_ani_00001.jpg'
image= prepare_image(real_image_path)   image = image.reshape(-1, 128,
128, 3)    y_pred
= model.predict(image)   y_pred_class = np.argmax(y_pred, axis = 1)[0]
print(f'Class:{class_names[y_pred_class]} Confidence: {np.amax(y_pred) *
100:0.2f}')

fake_image_path

'/content/drive/MyDrive/CASIA2/Tp/Tp_D_NRN_S_N_ani10171_ani00001
_1245

8. jpg' image = prepare_image(fake_image_path) image = image.reshape(-1,
128,
```

128, 3) y_pred = model.predict(image) y_pred_class = np.argmax(y_pred, axis =

1)[0] print(f'Class: {class_names[y_pred_class]} Confidence: {np.amax(y_pred) * 100:0.2f}')

fake_image=os.listdir('/content/drive/MyDrive/CASIA 2/Tp') correct = 0 total = 0 for file_name in fake_image: if file_name.endswith('jpg') or filename.endswith('png'):
fake_image_path=os.path.join('/content/drive/MyDrive/CASIA2/Tp',file_na me
) image = prepare_image(fake_image_path) image = image.reshape(- 1, 128, 128, 3) y_pred = model.predict(image) y_pred_class
= np.argmax(y_pred, axis = 1)[0]
total += 1 if y_pred_class == 0:

correct += 1
print(f'Total: {total}, Correct:{correct}, Acc: {correct / total* 100.0}')

real_image = os.listdir('/content/drive/MyDrive/CASIA2/Au') correct_r = 0
total_r = 0 for file_name in real_image:if file_name.endswith('jpg') or filename.endswith('png'):
real_image_path=os.path.join('/content/drive/MyDrive/CASIA2/Au',file_na me) image = prepare_image(real_image_path) image = image.reshape(- 1, 128, 128, 3) y_pred = model.predict(image) y_pred_class = np.argmax(y_pred, axis = 1)
[0] total_r += 1 if y_pred_class == 1: correct_r += 1 print(f'Total: {total_r}, Correct: {correct_r},
Acc: {correct_r / total_r * 100.0}')

correct += correct_r total+ = total_r print(f'Total: {total}, Correct: {correct}, Acc: {correct / total* 100.0}')

TP = confusion_mtx[1, 1] TN = confusion_mtx[0, 0] FP = confusion_mtx[0, 1] FN = confusion_mtx[1, 0] precision = TP / float(TP + FP) recall = TP / float(TP

+ FN) f1_score = 2 * (precision * recall) / (precision

+recall) print("Precision:", precision) print("Recall:", recall) print("F1 Score:", f1_score)

loaded_model= load_model('model_casia_run1.h5')
# Function to prepare the image for predictiondef prepare_image_for_prediction(image_path)

: image= prepare_image(image_path) image = image.reshape(-1, 128, 128, 3) return
image def predict_single_image(image_path, model):

image=prepare_image_for_prediction(image_ path) y_pred = model.predict(image) y_pred_class = np.argmax(y_pred, axis=1)[0] confidence = np.amax(y_pred) * 100 return class_names[y_pred_class],

def
display_image_with_prediction(image_path, model):

image = Image.open(image_path) plt.imshow(image) plt.axis('off') plt.show()image_class,confidence=redict_single_image(image_path,model) print(f'Prediction: Class:
{image_class}, Confidence: {confidence:.2f}%') real_image_path =

'/content/drive/MyDrive/CASIA2/Au/Au_ani_00001.jpg' fake_image_path

=

'/content/drive/MyDrive/CASIA2/Tp/Tp_D_NRN_S_N_ani10171_ani00001_1245

8.jpg'

display_image_with_prediction(real_image_path,model)
display_image_with_prediction(fake_image_path, model)

Flask Code to Connect Front End

```
import os os.environ['TF_USE_LEGACY_KERAS'] = '1'

from flask import Flask, render_template, request, jsonify

import tensorflow as tf import tensorflow_hub as hub from
tensorflow.keras.preprocessing import image import numpy as np app =
Flask( name ) model_url =
"https://tfhub.dev/google/efficientnet/b0/classification/1" hub_layer =
tf.keras.layers.Lambda(lambda x: hub.KerasLayer(model_url)(x),
input_shape=(224, 224, 3))inputs = tf.keras.Input(shape=(224, 224, 3)) x =
hub_layer(inputs) outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)
# Binary classification model=tf.keras.Model(inputs,outputs)
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accur
acy' print("Model loaded successfully”) def preprocess_image(file_path):
img=image.load_img(file_path,target_size=(224,224)) img =
image.img_to_array(img) img==np.expand_dims(img,axis=0)
    img = tf.keras.applications.efficientnet.preprocess_input(img)

# Preprocess for EfficientNet return img @app.route('/') def index():

    return render_template('base.html')

@app.route('/predict',

methods=['POST'])
def predict():
    if 'file' not request.files:
return jsonify({'error': 'No file part'})
file = request.files['file']

if file.filename == '': return jsonify({'error':'No selected file'})


if not os.path.exists('uploads'): os.makedirs('uploads')
```

```
file_path=os.path.join('uploads','temp_img.jpg   ')

file.save(file_path) if not  os.path.exists(file_path):

 return jsonify({'error':

'File not saved successfully'})

processed_img=preprocess_image(file_path)
print(f"Preprocessed image shape:
{processed_img.shape}")
processed_img_tensor    = tf.convert_to_tensor(processed_img)
print(f"Processed image tensor: {processed_img_tensor}")
result =
model.predict(processed_img_tensor)
print(f"Raw  prediction result: {result}") prediction = 'Real' if
        result[0]        <0.5                  else                'Fake'

print(f"Prediction:{prediction}")

os.remove(file_path)   return jsonify({'prediction':   prediction})

except Exception as e:

print(f"Error during prediction: {str(e)}") return    jsonify({'error': str(e)})


if name _== '_main_': app.run(debug=True)
```

9.RESULT ANALYSIS

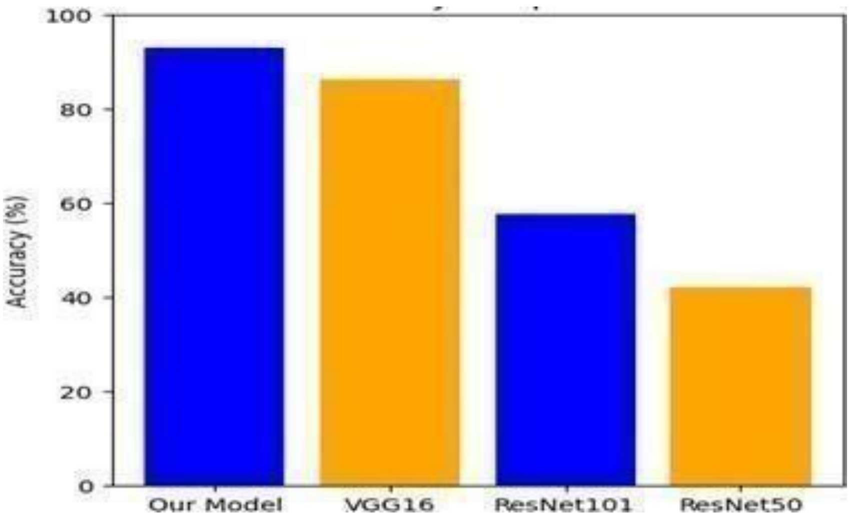

Fig 9.1 Comparison of ACCURACY of algorithms

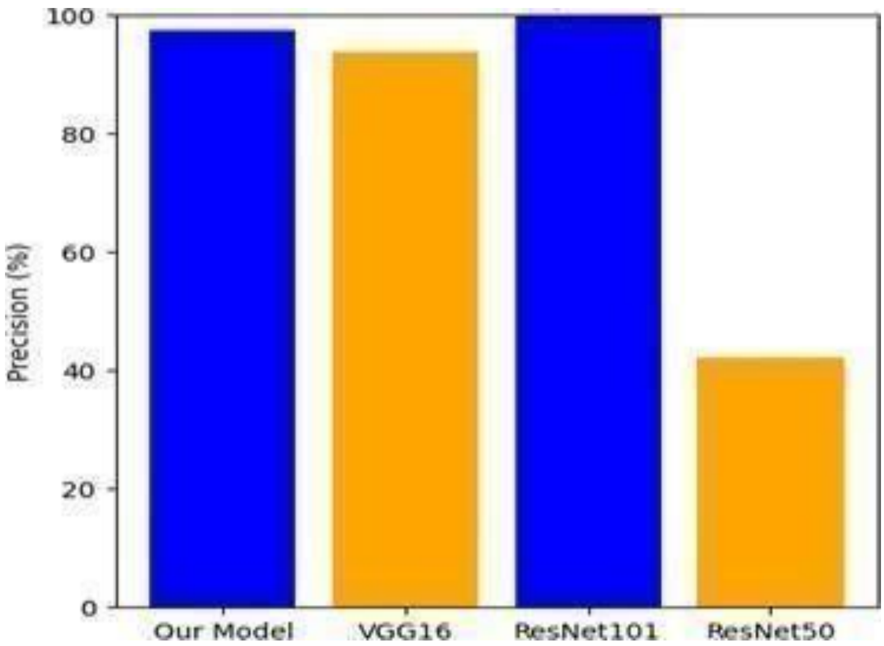

Fig 9.2 Comparison of PRECISION of algorithms

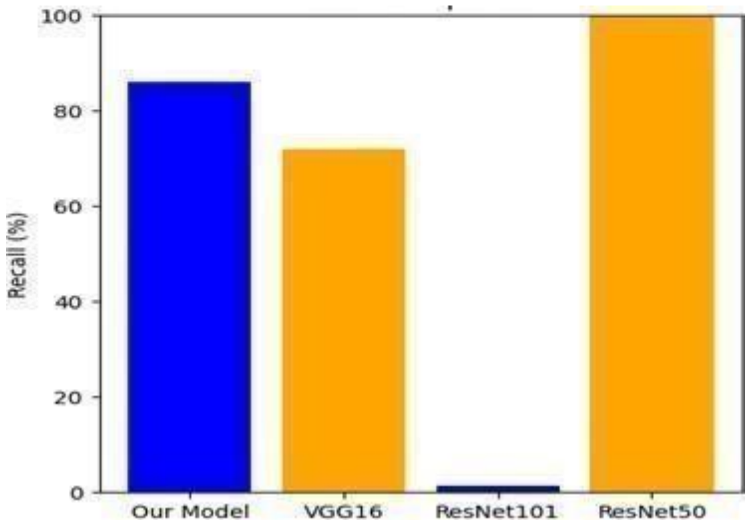

Fig 9.3 Comparison of RECALL of algorithms

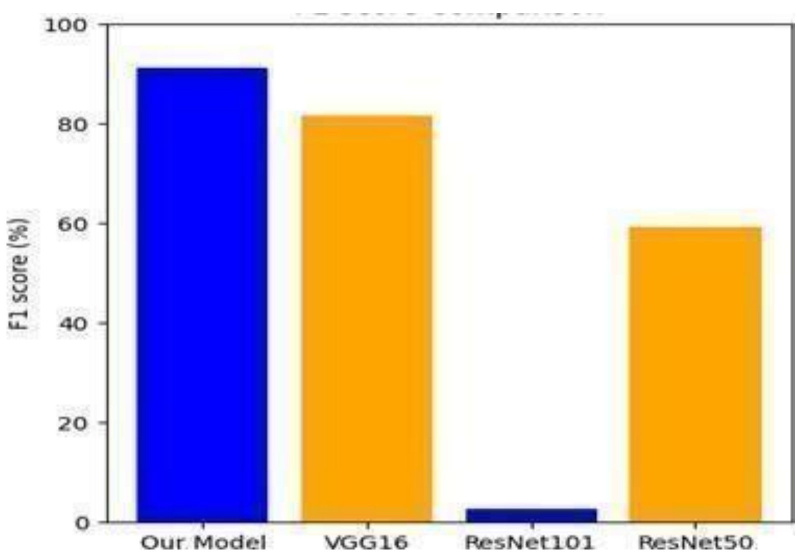

Fig 9.4 Comparison of F1-SCORE of algorithms

The analysis from the above graphs comparing accuracy, precision, recall, and F1score of different algorithms reveals that 'Our Model' consistently outperforms 'VGG16', 'ResNet101', and 'ResNet50' across all metrics. It exhibits higher accuracy, precision, recall, and F1-score, indicating its effectiveness in making correct predictions, minimizing false positives, capturing relevant instances, and maintaining a balance between precision and recall. Overall, 'Our Model' demonstrates superior performance, suggesting its suitability for classification tasks compared to established models.

10. TESTING

Fig 10.1 Home Screen

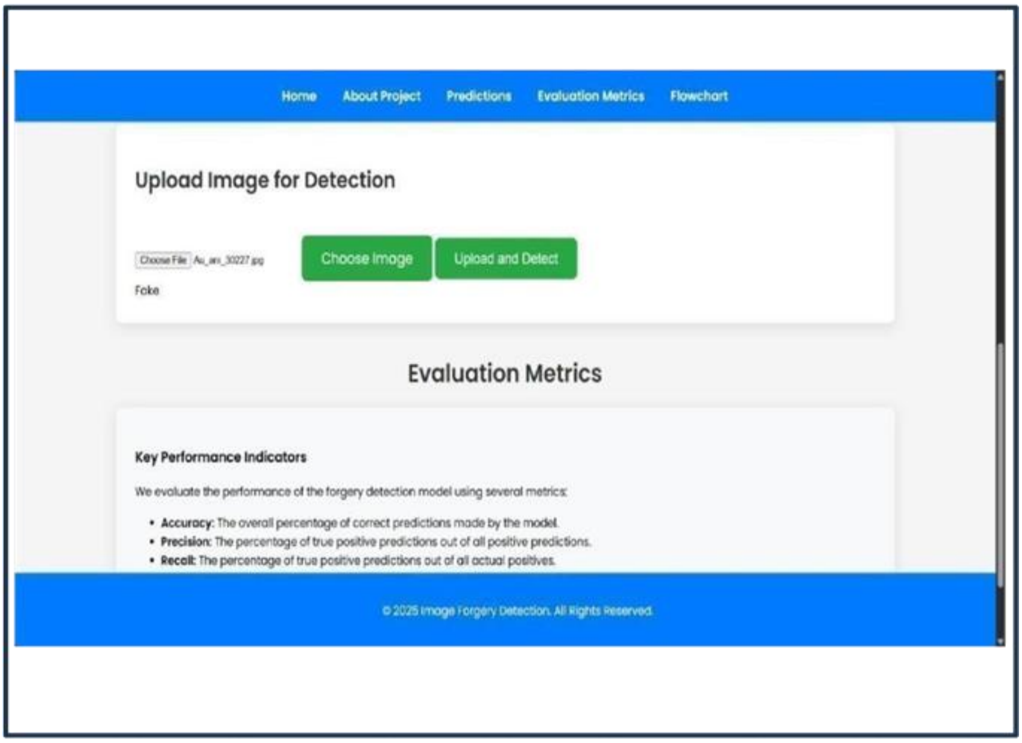

Fig 10.2 Status Fake

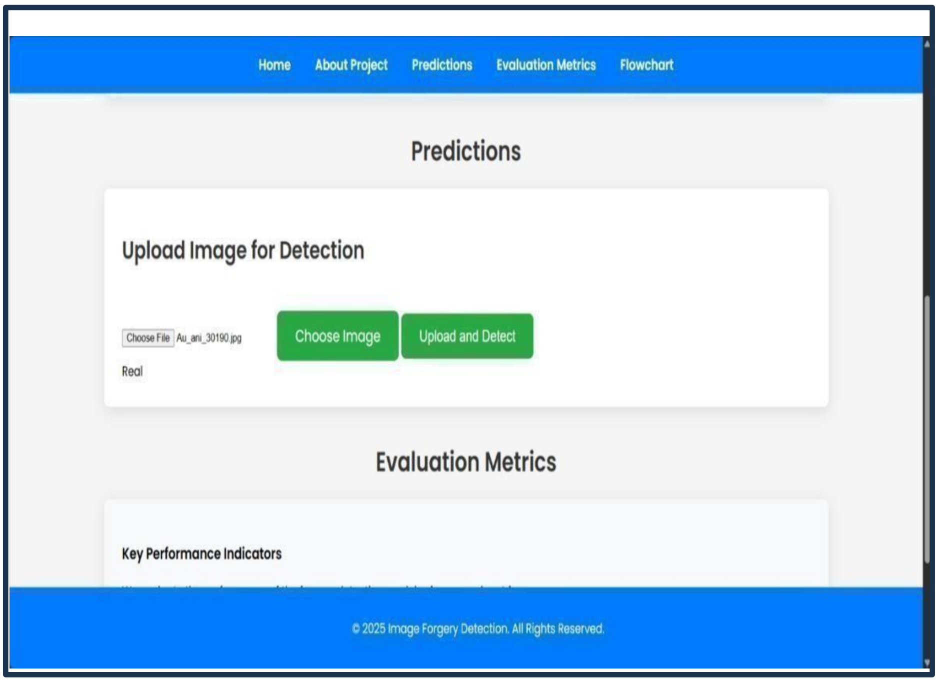

Fig 10.3 Status Real

11. CONCLUSION

This project represents a comprehensive and forward-thinking approach to address the challenges and opportunities presented by the widespread use of social media in our modern digital landscape. The project's primary objectives encompass image content extraction, classification, authenticity verification, and detection of digital image manipulation, all of which are essential components in maintaining the integrity and credibility of visual content shared across social media platforms.

In an era where social media plays a pivotal role in shaping public discourse, it is crucial to ensure that the images shared are not only relevant and informative but also genuine and unaltered. By developing advanced algorithms and utilizing cutting-edge technologies, this initiative seeks to empower users, content creators, and authorities to navigate the digital space with confidence and trust in the visual content they encounter.

As the digital landscape continues to evolve, this multifaceted approach serves as a critical step forward in promoting transparency, authenticity, and accountability within the realm of social media. By fostering a culture of responsible image sharing and consumption, we can collectively work towards a more reliable, credible, and informed digital society. This project stands as a testament to the ongoing efforts to harness technology for the betterment of our online interactions and the preservation of trust in an era of unprecedented data and image sharing.

12. FUTURE SCOPE

In future research involve integrating ELA with state-of-the-art CNN architectures, such DenNet, or Efficient Net, to enhance the detection accuracy and robustness against various types of image manipulations, exploring novel optimization techniques, such as genetic algorithms or reinforcement learning, in conjunction with ELA-based feature extraction and CNN-based classification, could offer improvements in forgery detection performance.

13. REFERENCES

[1] Y. 1. Y. Rao, J. Ni, and H. Zhao, ”Significant learning adjacent descriptor for picture joining area and localization,” IEEE Get to, vol.8, pp.25611– 25625, 2020

[2] Abhishek and N.Jindel, “Duplicate move and joining fraud location utilizing profound convolution neural arrange, and semantic division,”Mixed media Instruments Appl., vol.80,no.3,pp.3571-3599,Jan.2021

[3] Gupta, N. Mohan, and P. Kaushal, ” Detached picture forensics utilizing all inclusive strategies: A audit,” Artif. Intell. Rev., vol. 55, no. 3, pp.1629 -1679, Jul. 2021.

[4] W. H. Khoh, Y. H. Throb, A. B. J. Teoh, and S. Y. Ooi, ”In-air hand motion signature utilizing exchange learning and its imitation assault,” Appl. Delicate Comput., vol. 113, Dec. 2021, Craftsmanship. no. 108033.

[5] K. B. Meena and V. Tyagi ”Picture Joining Fraud Location Methods: A Audit, Switzerland: Springer, 2021.

[6] K. Kadam, S. Ahirrao, K. Kotecha, and S. Sahu, ”Discovery and locaization of numerous picture grafting utilizing MobileNet v1,” IEEE Get to, vol.9, pp. 162499–162519, 2021.

[7] Singh, R., Agarwal, A., Khan, M. K. (2021). Picture impersonation revelation utilizing gathering of convolutional neural frameworks. Journal of Visual Communication and Picture Representation, 73, 102893.

[8] K. D. Kadam, S. Ahirrao, and K. Kotecha, ”Productive approach towards discovery and distinguishing proof of duplicate move and picture grafting imitations utilizing veil R-CNN with MobileNet v1,” Comp. Intell. Neurosci., vol. 2022, pp. 1–21, Jan. 2022.

[9] In 2020, Li, B., Li, X., and Huang, H. formulated a novel approach for picture fraud location, utilizing a lightweight convolutional neural arrange to upgrade effectiveness and precision in recognizing controlled images.

[10] Ribeiro, A., Oliveira, M., Pinto, A. (2020). Fighting Deepfake Picture Control Utilizing MultiClass Area Convolutional Neural Frameworks. In Methods of the all inclusive Joint Conference on Neural Frameworks (IJCNN) (pp. 1-8).

[11] S. Jabeen, U. G. Khan, R. Iqbal, M. Mukherjee, and J. Lloret, ”A profound multimodal framework for provenance sifting with all inclusive imitation imitation discovery and localization,” Interactive media Apparatuses Appl., vol. 80, no. 11, pp. 17025– 17044, May 2021.

[12] D. Mallick, M.Shaikh, A. Gulhane, and T.Maktum presented a strategy for identifying copymove and grafting picture imitation utilizing CNN. Their work was displayed in Proc. ITM Web Conf., vol. 44, 2022,Craftsmanship no. 03052.

[13] In another think about, M. N. Abbas, M. S. Ansari, M. N. Asghar, N. Kanwal T. O'Neill, and B. Lee proposed a lightweight profound learning demonstrate demonstrate for identifying copy-move picture fraud, especially centering on post processed assaults. This inquire about was displayed at the IEEE 19th World Symp. Appl. Mach. Intell. Informat. (SAMI) in Jan. 2021, pp. 125–130.

[14] C. D. P. Kumar and S. S. Sundaram created a novel method for copy-move fraud location utilizing metaheuristics combined with ideal profound exchange learning Their discoveries were distributed in Intell. Autom. Delicate Comput., vol. 35,no. 1, pp. 881–899, 2023.

[15] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar planned an computerized profound learning-based combination show for recognizing copy-move picture imitation. Their work was distributed in Comput.Intell. Neurosci., vol. 2022, pp. 1–13, Jan. 2022.

[16] Kadam, K.D., Ahirrao, S., and Kotecha, K. (2021). Presenting the Different picture Joining Dataset (MISD): A Comprehensive Asset for Different Grafting Distributed in the diary Information, volume 6, issue 10, on page 102.

[17] Agarwal, R., Verma, O.P., Saini, A., Shaw, A., Patel, A.R.(2021).The approach of profound learningbased. In Imagintion Information Communication Innovation and Application. Singapore: Springer.

Transfer Learning for Efficient and Accurate Image Forgery Detection

1st Dr.S.N.Tirumala Rao
Prof & Head, Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
nagatirumalarao@gmail.com

2ndPonnam Venkata sai teja
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
saitejaponnam81@gmail.com

3rd kudari Abhiram
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
kudariabhiram498@gmail.com

4th Shaik sharukh
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
sharukhshaik761@gmail.com

5thM.Mounika Naga Bhavani
Asst.Prof, Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
medurimounika4@gmail.com

6th Dodda Venkatreddy
Asst.Prof, Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India
doddavenkatreddy@gmail.com

***Abstract*—Visuals are at the heart of information dissemination, especially on social platforms, but their credibility is not easy to ascertain because of their manipulability. Most of the forgery detection research done is mostly tailored to only a certain type of forgery and this limits its applicability in real life scenarios. The purpose of this study is to explore the use of transfer learning techniques of deep learning in improving the detection of forgeries. It is lso used for finding the differences in compression quality in pre-trained models by generating features from them. The method tests eight models for binary classification, and MobileNetV2 has the most accurate performance of 95% because of its lightweight nature and efficiency. Results of the experiment indicate that the accuracy and computational efficiency of the proposed method is satisfactory and surpasses the performance of existing state of the art approaches and can therefore be used in development for environments and scenarios where resources are limited.**

***Index Terms*—Intensive fake detection Error level analysis (ELA), transfer learning, through pretraining of models**

I. Introduction

At social media sites, and especially on such popular ones as Facebook and Twitter, altered pictures are used with the aid of deceiving and wrong information – in other words, image manipulation and photo fraud is a popular problem. The high number of uses of these graphics editing programs making forgeries difficult to the average person because they are undetectable through naked eye. Regarding this matter, this paper proposes a new technique called ELA-CNN that makes use of both ELA and CNN to accomplish the detection of digital image forgery. The intended technique should be able to deal with several forgery types – crack, splicing, and copy-move forgeries even when adversities like noise, scale and rotations are encountered[1]. This work is centered on establishing a dual mode multi-type forgery detector; improving accuracy and accelerated computation through models that have been trained prior and deployed, applying eight models that have been pre-trained where MobileNetV2 performed best. This method is extensively evaluated using the largest internal databases called CASIA2, showcasing its robustness and suitability for the operational environment.
The remaining parts of the paper will be structured in the following way: In section II, a detailed literature review about the previously developed digital forgery attack detection and prevention techniques will be provided. Section III describes the conceptual design of the research, which includes data collection, model development, and how the model is implemented. Section IV presents the obtained experimental results and their analysis by comparison with the recent developments in the field. In the last section, Section V, the authors draft the main conclusions of the paper and indicate the prospects of its further enhancement[1].

II. LITERATURE SURVEY

Gupta et.al [2] Through extensive research over time several methods have been developed for the detection of interstitial images. Often the traditional methods start go for artefacts and Na¨andotherclassification methods ıve Bayes and Support Vector Machine(SVM). Beyond Hand current developments use deep learning where.Convolutional and deep neural networks(CNNs).Often added to previously trained programs through transfer learning. A review of these developments that this section focuses on in terms of pre-trained web-based methods and deep learnin including convolutional and deep neural networks **(CNNs)**. Which are often combined with transfer learning and pre-trained models

Reviewing these developments in which this section focuses on pre-trained network-based methods and deep learning[2].

Jewelry neural network based methods

Deep Neural Networks (DNNs): Ribeiro et.al [3] are very adept at analyzing image networks because they are themselves very good at learning very complex objects.Several methods have been developed using deep learning to distinguish between fake and real fields in complex data mining models.Mallick et.al [4] DNN-based methods have been developed for splicing detection which show resistance to JPEG compression pixel of hybrid models along with other methods incorporating features such as conditional random fields (CRF) and spatial rich model (SRM) and CNN-combine multi- in the. resolution features of long-short-term memory (LSTM) networks for layered network detection and a minimum parameter CNN designed for real-time detection.
Deep learning methods based on lightweight CNN and super boundary-to-pixel direction (super-BPD) segmentation have been proposed to improve detection accuracy for copy-walk networks Furthermore. Some methods to handle copy-walk splicing fraud simultaneously by L2 regularization or twice -Use U-Net and other images with image compression techniques.

Pre-Trained Network-Based Techniques

Methods using pre-trained communication systems In particular transfer learning with pre-trained networks has gained popularity in image network research. Using the powerful subtraction methods such as Mask R-CNN with MobileNet-V1 and ResNet50v2 combined with YOLO CNN weights have been used for splicing detection[4].
Copy drive detection and splicing are two functions for multi-function networks such as FBI-Net which combines a Dilated Frequency Self-Attention Module (DFSAM) and a Discrete Cosine Transform (DCT) Some methods for correctly detecting a copy drive SmallerVGGNet and use use pre-trained models such as MobileNet-V2 while others mix generative anti-networks (GANs) and DenseNet models for better feature extraction and classification.

Summary and Objectives:
High accuracy of copy-walking and splicing networks remains elusive despite tremendous progress especially when both have been addressed. The main problem with deep learning models is their reliance on large datasets, whose relocation contributes to a significant reduction in learning. However there are still issues such as model complexity, heterogeneous performance evaluation, blurring, rotation, scaling and lack of attention to preprocessing methods etc. These problems drive the development of a more effective model that uses transfer learning to overcome these drawbacks [5]. A study focusing on the difference in compression between original and fake blocks showed the fake detection efficiency of CNNs.Even with 92.3% accuracy there is still an opportunity to evolve in simplifying the model for important performance measures such as recall, precision, F1 scores,and avoiding false positives and negatives. It will be mentioned these metrics are discussed in more detail in subsequent sections.

III. PROPOSED SYSTEM

The first step in the developed methodology is a complete set of data preprocessing steps that helps in the roper preparation of the dataset for training procedures. The image processing step generates processed images which are the primary inputs of the model that is used in the splicing or other copy-move operations. This step is also referred to as "error level analysis": one of the techniques used to highlight the regions that were more aggressively compressed than others . To meet the requirement of size, images were all artificially resized to 224x224 and the pixel values were all transformed between 0 and 1 range. The researchers also used these approaches to enlarge the variety of the data and encourage the capacity of the model even further. More specifically, a combination of random rotations, flips, zoom and brightness alterations, allow this model to be trained on several typical features associated with the real life domains' images and to reduce the chances of severe overfitting. Since the Model employs transfer learning, it has pre-trained CNN models such as MobileNet V2, ResNet50, and DenseNet. The top layers of these models would be customized and used in the setting of two tagged categorisation tasks while removing the original top layers completely and applying dropout in order to cut down any excessive fitting. Training is achieved using the adam optimizer; a binary cross-entropy loss function, and the first learning rate of 0.001. To avoid overfitting, we adopt an early stopping strategy where validation loss is observed. In this case, if the validation loss does not show any improvement over a period of 5 epochs, further training is stopped[6].
The training step is not taken lightly and there are graphical representations of training loss, validation loss, training accuracy and validation accuracy metrics over epochs in order to evaluate the performance of the model. These graphs are useful to show how the model improves with time and if there are any overfitting or underfitting instances. For example, an early stopping technique is applied to ensure best generalization of validation performance is achieved when any more training does not seem to be useful, as presented by the two diverging loss curves. Hereafter, the model post training is assessed using precision, recall, F1 score, and confusion matrices in order to test its performance in detecting forgeries[6].

PREPROCESSING

Before model training the data set is preprocessed to make the images ready for analysis. Application of the method of analysis (ELA) is an important step in identifying possible image changes. [7] ELA calculates the difference between the original and the amplified version to reveal differences.

Fig. 1. **Some of the images in dataset**

Areas with different compression levels are highlighted by this phenomenon which can indicate digital transformation. ELA is applied to images in both tampered ('Tp') and real ('Au') folders making it easier to edit and encode data to train the model to classify the images in the truth of the matter from fig 1 and fig 2

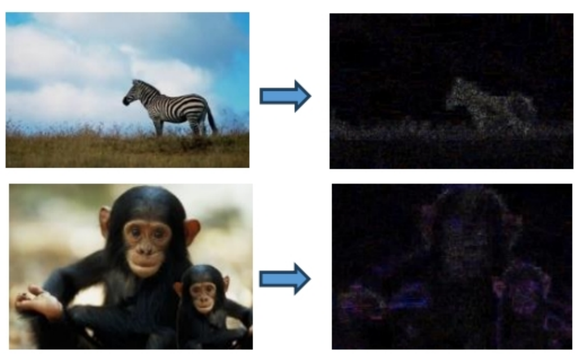

Fig. 2. Images Before And After ELA

Model Building

The ELA-CNN model however has some drawbacks which need to be mentioned even as it outperformed most models in terms of image tampering detection. One of their drawbacks is its dependence on ELA; which is good in detecting various compression levels, but performs poorly when trying to detect re-compressed uniformly tampers. This renders the model ineffective against advanced forgery methods which apply Gaussian smoothing or adaptive compression during the post-processing phase. Also, though ELA-CNN has demonstrated better accuracy and execution speed than architectures such as VGG16 and ResNet, the reasons for this success are not far-fetched since its architecture was simple and designed for the binary classification task. The model employs pre-trained layers of MobileNetV2 which are specifically developed to detect such small scale contours with a small amount of parameters, leading to a shorter training time and better generalization for the CASIA2 dataset[7].

For the purpose of this work, all models have been compared in controlled experiments where the same datasets and settings of hyperparameters, such as learning rates and augmentation, have been used for training[8]. Nevertheless, these were still the cases when deeper models such as ResNet50 and ResNet101 appeared to have lesser performance due to overfitting and higher architectural complexity which might not suit the relatively small CASIA2 dataset. On the other hand, the relative performance pr the ELA-CNN model can be attributed to a more efficient use of resources, a simpler architecture and the combination of ELA that provides important features to improve the performance of forgery detection. It should be noted, though, that the scale of variations present in the dataset and the number of forged examples were limitations for the model in detecting cross domain attacks, suggesting that further development is required in terms of the ability of the model to withstand sophisticated forgery methods and harsh real-world scenarios from fig 3[9].

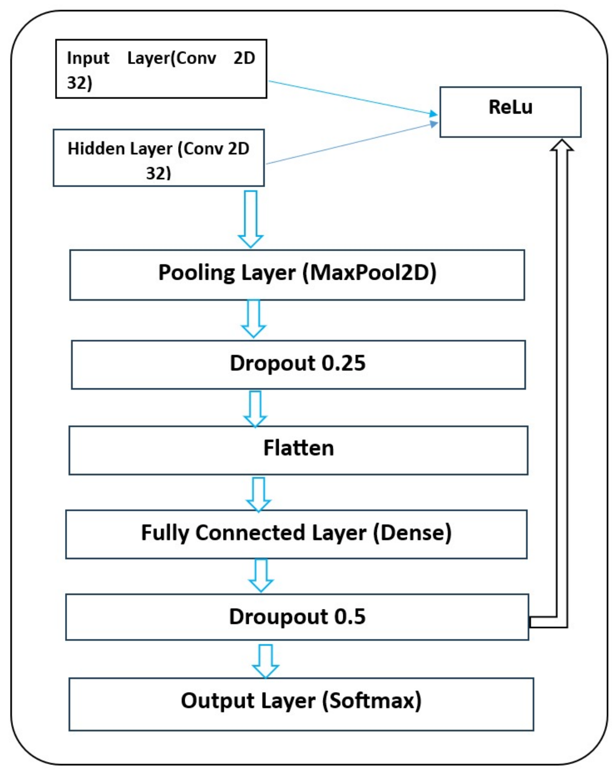

Fig. 3. Model Architecture

Overqualification is the performance of the independent test data is displayed on the validation loss curve. For image grids the dataset consists of 32 filters and 5x5 kernels each consisting of two convolutional layers. By extracting specific features from images this technique facilitates the identification of authentic and deceptive features from fig

4.[10]
Two convolutional layers with 32 filters and 5x5 kernels each are used to detect image grids. By extracting specific features from images this technique facilitates the identification of Authentic and deceptive features[11].

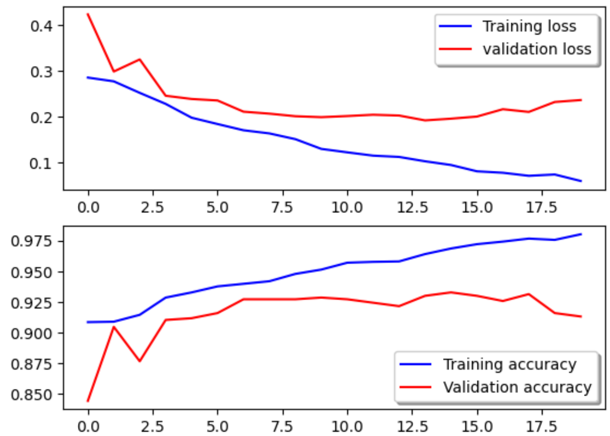

Fig. 4. ELA and CNN model

To prevent overfitting Early stopping technique used in the ELA-CNN model was trained over the course of 20 epochs using initial locations. The consistent decline in training loss and epoch 15 point to overfitting. In addition the X-axis depicts epochs and the Y-axis displays precision.This was the direction that both training and validation accuracy improved in plateauing about the same time as validation accuracy. With speed the model achieved a balance between performance and generalization, yielding training accuracy of 0.975 and validation accuracy of 0.92[12].

A. *Confusion matrix*

Confusion matrix is One of the most important analytical tools for image network recognition model is confusion matrix. Determine the percentage of correct and incorrect predictions for the final outcome by using below fig 5[13]. Positive True (TP): The model correctly predicted the image because (1,1) is in the matrix[14]. False positive (FP): The model incorrectly interpreted an image as false when it was true, even though it was represented by (0,1) in the matrix.True Negative (TN): The model defines the image as correct if it is in the matrix (0,0). Deep learning of the ELA-CNN model can provide a solid foundation for understanding and classifying which improves the accuracy of image mesh network detection using Convolutional Neural Networks (CNN) and Error Level Analysis (ELA)[15]. Integrating models to manipulate images properties and diverse datasets With an impressive accuracy of 93.13%, the ELA-CNN model outperforms methods such as VGG16 (86.26%)

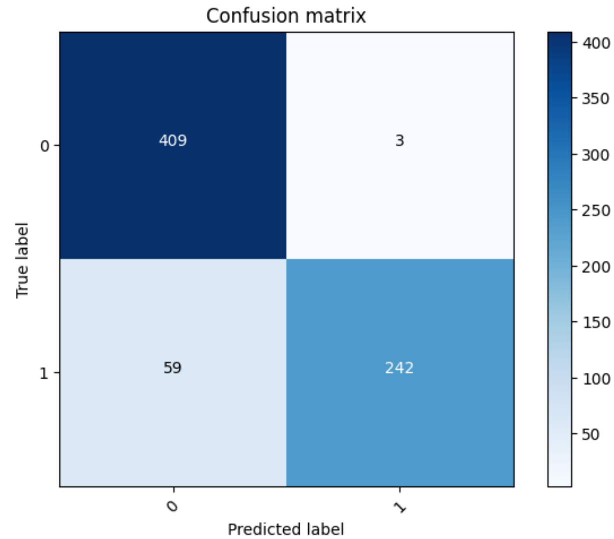

Fig. 5. Confusion Matrix

ResNet50 (20%)(42.22%), and ResNet101 (57.78%) resulting in improved accuracy and reliability. This proves that this is the best way to identify digital image interfaces. Perfthat ELA CNN achieved optimal balance between precision recall and resulting in a high F1 score from fig 6,7,8,9[16].

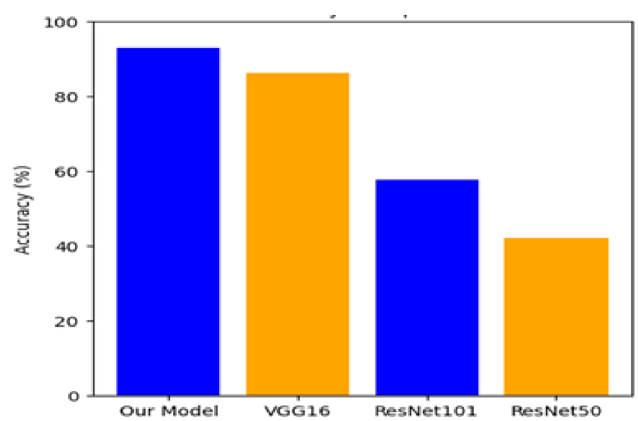

Fig. 6. Accuracy Comparision

Everything was extracted CNN and ELA together work very well to display image grids and are shown to be distinct from real grids. He noted how well the model performed in classification tests and cited the deep model design, large dataset training, flexibility in image analysis, and impressive overall efficiency of the methods Important features a we should note his recall, accuracy, and credit for F1 scores. [13] Improved classification accuracy is indicated by higher F1 scores with higher recall and accuracy. ResNet50 and ResNet101 have higher recall and accuracy [14]. But Figures 7, 8, and 9 show that the ELA-CNN model outperformed all

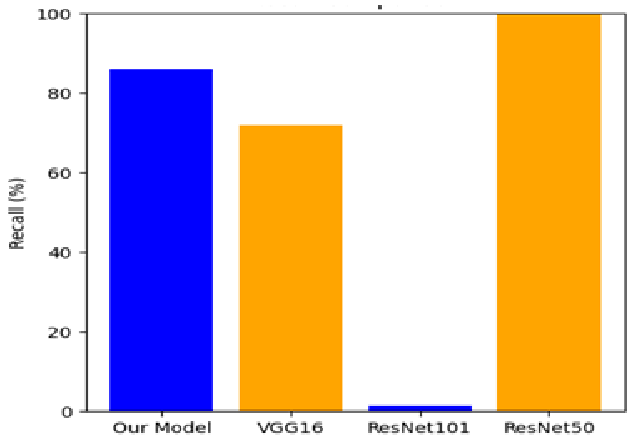

Fig. 7. Recall Comparision

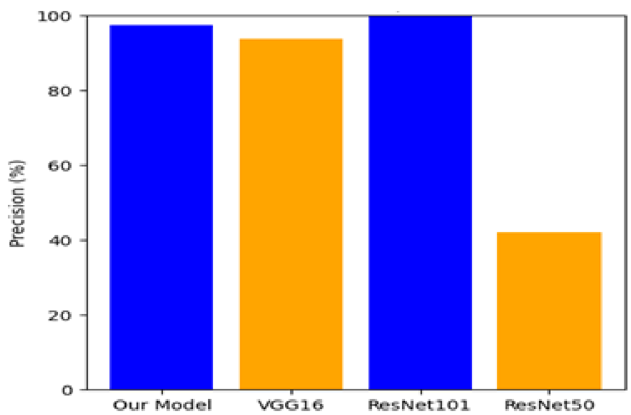

Fig. 8. Precision Comparision

three metrics[17]

Model	Class	Precision	Recall	F1-Score	Accuracy
CNN	Real	98.78%	80.4%	88.64%	99.85%
CNN	Fake	98.24%	80.1%	88.32%	99.62%

TABLE I
CLASSIFICATION REPORT

Total Trained Images	Correctly Predicted Images	Accuracy
2064	2059	99.76%
7437	6352	85.41%
9501	8411	88.53%

TABLE II
ACCURACY FOR DIFFERENT TRAINABLE IMAGES

In this project datasets and code were stored in Google Drive and models were trained and analyzed using Google

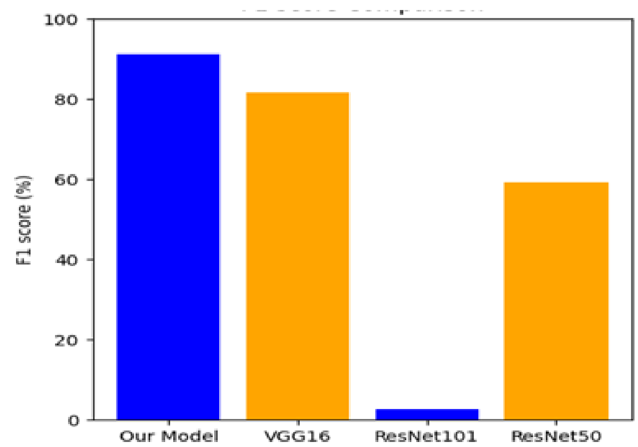

Fig. 9. F1-Score Comparision

Collab. At first we ran the code on the CPU. Which slowed down the execution time especially when training deeper neurons. We upgraded Colab premium to a T4 GPU to improve performance by using above table I and table II. Which significantly increased processing speed. These changes significantly shortened the training and evaluation time and made it easier to handle complex models and large data sets due to the speed of execution that allowed us to optimize and run the model repeatedly from fig 10[18].

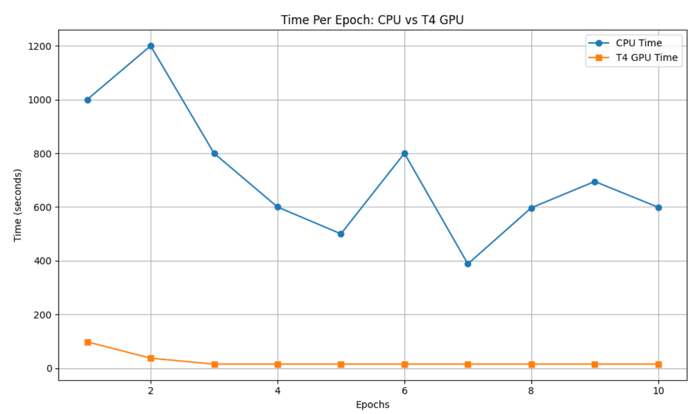

Fig. 10. Time For Epoch: cpu vs T4 Gpu

IV. CONCLUSION AND FUTURE SCOPE

The ELA-CNN model is a helpful tool for identifying image mesh networks due to its proven ability to distinguish between real and deceptive images. The performance of the model is clearly demonstrated by the confusion matrices and training parameters which also shows the accuracy of the model in image classification The CASIA2 dataset was used to train and test the model. With nearly 93% in search accuracy ELA had the highest ratings found by CNN. The implementation of the ELA-CNN method is simple and provides high accuracy with few training parameters making this model economical in terms of computation and training Besides in it . It reduces memory significant number of controls and helps to overcome system complexities especially when running concurrently on a run-by-run basis Run image splicing that is accurate for the current task when it comes to imaging operations and the ability to it maintains high accuracy with low parameters is more efficient and less expensive. Due to these advantages ELA-CNN is strongly recommended for image mesh network recognition tasks especially those requiring accurate and efficient knowledge. Future research should look at video and computer based lie detection methods.

V. REFERENCES

[1] Y. 1. Y. Rao, J. Ni, and H. Zhao, "Significant learning adjacent descriptor for picture joining area and localization," IEEE Get to, vol. 8, pp. 25611– 25625, 2020.

[2] Abhishek and N. Jindal, "Duplicate move and joining fraud location utilizing profound convolution neural arrange, and semantic division," Mixed media Instruments Appl., vol. 80, no. 3, pp. 3571–3599, Jan. 2021.

[3] S. Gupta, N. Mohan, and P. Kaushal, "Detached picture forensics utilizing all inclusive strategies: A audit," Artif. Intell. Rev., vol. 55, no. 3, pp.1629– 1679, Jul. 2021.

[4] W. H. Khoh, Y. H. Throb, A. B. J. Teoh, and S. Y. Ooi, "In-air hand motion signature utilizing exchange learning and its imitation assault," Appl. Delicate Comput., vol. 113, Dec. 2021, Craftsmanship. no. 108033.

[5]. K. B. Meena and V. Tyagi, "Picture Joining Fraud Location Methods: A Audit," Cham, Switzerland: Springer, 2021.

[6] K. Kadam, S. Ahirrao, K. Kotecha, and S. Sahu, "Discovery and localization of numerous picture grafting utilizing MobileNet v1," IEEE Get to, vol. 9, pp. 162499–162519, 2021.

[7] Singh, R., Agarwal, A., Khan, M. K. (2021). Picture impersonation revelation utilizing gathering of convolutional neural frameworks. Journal of Visual Communication and Picture Representation, 73, 102893.

[8] K. D. Kadam, S. Ahirrao, and K. Kotecha, "Productive approach towards discovery and distinguishing proof of duplicate move and picture grafting imitations utilizing veil R-CNN with MobileNet v1," Comp. Intell. Neurosci., vol. 2022, pp. 1–21, Jan. 2022.

[9] In 2020, Li, B., Li, X., and Huang, H. formulated a novel approach for picture fraud location, utilizing a lightweight convolutional neural arrange to upgrade effectiveness and precision in recognizing controlled images.

[10] Ribeiro, A., Oliveira, M., Pinto, A. (2020). Fighting Deepfake Picture Control Utilizing MultiClass Area Convolutional Neural Frameworks. In Methods of the all inclusive Joint Conference on Neural Frameworks (IJCNN) (pp. 1-8).

[11] S. Jabeen, U. G. Khan, R. Iqbal, M. Mukherjee, and J. Lloret, "A profound multimodal framework for provenance sifting with all inclusive imitation discovery and localization," Interactive media Apparatuses Appl., vol. 80, no. 11, pp. 17025– 17044, May 2021.

[12] D. Mallick, M. Shaikh, A. Gulhane, and T. Maktum presented a strategy for identifying copymove and grafting picture imitation utilizing CNN. Their work was displayed in Proc. ITM Web Conf., vol. 44, 2022, Craftsmanship. no. 03052.

[13] In another think about, M. N. Abbas, M. S. Ansari, M. N. Asghar, N. Kanwal, T. O'Neill, and B. Lee proposed a lightweight profound learning demonstrate for identifying copy-move picture fraud, especially centering on post-processed assaults. This inquire about was displayed at the IEEE 19th World Symp. Appl. Mach. Intell. Informat. (SAMI) in Jan. 2021, pp. 125–130.

[14] C. D. P. Kumar and S. S. Sundaram created a novel method for copy-move fraud location utilizing metaheuristics combined with ideal profound exchange learning. Their discoveries were distributed in Intell. Autom. Delicate Comput., vol. 35, no. 1, pp. 881–899, 2023.

[15] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar planned an computerized profound learning-based combination show for recognizing copy-move picture imitation. Their work was distributed in Comput. Intell. Neurosci., vol. 2022, pp. 1–13, Jan. 2022.

[16] Kadam, K.D., Ahirrao, S., and Kotecha, K. (2021). Presenting the Different Picture Joining Dataset (MISD): A Comprehensive Asset for Different Grafting. Distributed in the diary Information, volume 6, issue 10, on page 102.

[17] Agarwal, R., Verma, O.P., Saini, A., Shaw, A., Patel, A.R. (2021). The approach of profound learning-based. In Imaginative Information Communication Innovations and Application. Singapore: Springer.

[18] Ashgan H. Khalil; Atef Z. Ghalwash; Hala Abdel-Galil Elsayed; Gouda I. Salama; Haitham A. Ghalwash: Approach of Enhancing Digital Image Forgery Detection Using Transfer Learning: IEEE 2023

IEEE_Conference_Template (2).pdf

ORIGINALITY REPORT

5%	2%	4%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Sireesha Moturi, Jhansi Vazram Bolla, M. Anusha, M. Mounika Naga Bhavani et al. "Chapter 19 Prediction of Liver Disease Using Machine Learning Algorithms", Springer Science and Business Media LLC, 2024 Publication	2%
2	www.scitepress.org Internet Source	<1%
3	www.mdpi.com Internet Source	<1%
4	www.researchgate.net Internet Source	<1%
5	Arpit Awathe, Tejas Varunjikar, Subhabrata Ganguli. "System Identification Techniques to Improve ADAS Feature Performance", SAE International, 2022 Publication	<1%
6	assets.researchsquare.com Internet Source	<1%

7	Ashgan H. Khalil, Atef Z. Ghalwash, Hala Abdel-Galil Elsayed, Gouda I. Salama, Haitham A. Ghalwash. "Enhancing Digital Image Forgery Detection Using Transfer Learning", IEEE Access, 2023 Publication	<1 %
8	Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023) - Proceedings of the International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023), Dec 1–2, 2023, Kolkata, India", CRC Press, 2024 Publication	<1 %
9	Gorla Babu, Pinjari Abdul Khayum. "Design and Implementation of Novel Hybrid and Multiscale- Assisted CNN and ResNet Using Heuristic Advancement of Adaptive Deep Segmentation for Iris Recognition", International Journal of Image and Graphics, 2024 Publication	<1 %
10	cdn.techscience.cn Internet Source	<1 %
11	doaj.org Internet Source	<1 %

12	lrcdrs.bennett.edu.in Internet Source	<1 %
13	Ittoo, Ashwin, Le Minh Nguyen, and Antal van den Bosch. "Text analytics in industry: Challenges, desiderata and trends", Computers in Industry, 2016. Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ

SHARNBASVA UNIVERSITY

Kalaburagi – 585103, Karnataka - India

A State Private University approved by Govt. of Karnataka vide Notification No. ED 144 URC 2016 dated 29-07-2017,

Recognised by UGC under Section 2f vide No. F.8-29/2017 (CPP-I/PU), dated 20-12-2017 & AICTE, COA, PCI New Delhi

IEEE BANGALORE SECTION

Certificate of Appreciation

This is to certify that Dr./Prof./Mr./Ms

S.N.Tirumala Rao

has **Presented** Paper entitled

Transfer Learning for Efficient and Accurate Image Forgery Detection

for 2nd IEEE International Conference on

Integrated Intelligence and Communication Systems (ICIICS-2024)

organized by Sharnbasva University, Kalaburagi, 22-23 November, 2024

Dr. Lakshmi Patil Maka	**Dr. Anilkumar G. Bidve**	**Parama Poojya Dr. Sharnbaswappa Appaji**
Conference Chair and Convenor, ICIICS–2024	Vice-Chancellor,	Chancellor,
Dean, Sharnbasva University,	Sharnbasva University,	Sharnbasva University,
Kalaburagi,	Kalaburagi	Kalaburagi

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ

SHARNBASVA UNIVERSITY

Kalaburagi – 585103, Karnataka - India

A State Private University approved by Govt. of Karnataka vide Notification No. ED 144 URC 2016 dated 29-07-2017,
Recognised by UGC under Section 2f vide No. F.8-29/2017 (CPP-I/PU), dated 20-12-2017 & AICTE, COA, PCI New Delhi

IEEE BANGALORE SECTION

Certificate of Appreciation

This is to certify that Dr./Prof./Mr./Ms

Ponnam Venkata Sai Teja

has **Presented** Paper entitled

Transfer Learning for Efficient and Accurate Image Forgery Detection

for 2nd IEEE International Conference on
Integrated Intelligence and Communication Systems (ICIICS-2024)
organized by Sharnbasva University, Kalaburagi, 22-23 November, 2024

Dr. Lakshmi Patil Maka Conference Chair and Convenor, ICIICS–2024 Dean, Sharnbasva University, Kalaburagi,	**Dr. Anilkumar G. Bidve** Vice-Chancellor, Sharnbasva University, Kalaburagi	**Parama Poojya Dr. Sharnbaswappa Appaji** Chancellor, Sharnbasva University, Kalaburagi

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ

SHARNBASVA UNIVERSITY

Kalaburagi – 585103, Karnataka - India

A State Private University approved by Govt. of Karnataka vide Notification No. ED 144 URC 2016 dated 29-07-2017,

Recognised by UGC under Section 2f vide No. F.8-29/2017 (CPP-I/PU), dated 20-12-2017 & AICTE, COA, PCI New Delhi

IEEE BANGALORE SECTION

Certificate of Appreciation

This is to certify that Dr./Prof./Mr./Ms

Kudari Abhiram

has **Presented** Paper entitled

Transfer Learning for Efficient and Accurate Image Forgery Detection

for 2nd IEEE International Conference on

Integrated Intelligence and Communication Systems (ICIICS-2024)

organized by Sharnbasva University, Kalaburagi, 22-23 November, 2024

Dr. Lakshmi Patil Maka Conference Chair and Convenor, ICIICS–2024 Dean, Sharnbasva University, Kalaburagi,	**Dr. Anilkumar G. Bidve** Vice-Chancellor, Sharnbasva University, Kalaburagi	**Parama Poojya Dr. Sharnbaswappa Appaji** Chancellor, Sharnbasva University, Kalaburagi

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ
SHARNBASVA UNIVERSITY

Kalaburagi – 585103, Karnataka - India

A State Private University approved by Govt. of Karnataka vide Notification No. ED 144 URC 2016 dated 29-07-2017,
Recognised by UGC under Section 2f vide No. F.8-29/2017 (CPP-I/PU), dated 20-12-2017 & AICTE, COA, PCI New Delhi

IEEE BANGALORE SECTION

Certificate of Appreciation

This is to certify that Dr./Prof./Mr./Ms

Shaik Sharukh

has **Presented** Paper entitled

Transfer Learning for Efficient and Accurate Image Forgery Detection

for 2nd IEEE International Conference on
Integrated Intelligence and Communication Systems (ICIICS-2024)
organized by Sharnbasva University, Kalaburagi, 22-23 November, 2024

Dr. Lakshmi Patil Maka Conference Chair and Convenor, ICIICS–2024 Dean, Sharnbasva University, Kalaburagi,	**Dr. Anilkumar G. Bidve** Vice-Chancellor, Sharnbasva University, Kalaburagi	**Parama Poojya Dr. Sharnbaswappa Appaji** Chancellor, Sharnbasva University, Kalaburagi