Capstone

by Deepali Bhagat

Submission date: 08-May-2024 03:36PM (UTC+0530)
Submission ID: 2374115253
File name: FinalReportForPlagCheck.pdf (851.14K)
Word count: 6606
Character count: 35687

Chapter 1: INTRODUCTION

1.1 General Introduction

Machine Learning (ML) and Deep Learning (DL) have become indispensable in contemporary society, a surge in diverse applications aimed at enhancing user experiences. However, a key challenge lies in designing applications that demand minimal installation and computing resources on user terminals while maximizing convenience. Digital imagery stands out as a pivotal medium for information exchange, with cameras and smartphones churning out vast volumes of images and videos.

The proliferation of ML and DL has facilitated the development of applications that harness the power of these technologies while prioritizing user comfort. These applications leverage sophisticated algorithms to process digital imagery efficiently, enabling seamless user experiences without burdening their devices with excessive computational demands.

In essence, the convergence of ML, DL, and digital imagery underscores a transformative trend in application development, wherein innovation is driven by a relentless pursuit of user-centric solutions that deliver optimal performance with minimal overhead.

Machine and deep learning are commonly used in designing Intrusion Detection Systems (IDS), which aim to protect networks from cyber-attacks. With the rise of streaming services delivered via cloud servers, such as Google Stadia, there's a growing need for robust security measures. These services allow users to access high-end applications without needing powerful local hardware, relying instead on internet connectivity to stream data back and forth. However, this increased reliance on cloud-based services also expands the potential for cyber-attacks, including DDoS, port scans, and infiltration attempts. To counter these threats, developing a reliable IDS has become crucial. Traditional IDS typically rely on attack signature databases and predefined rules to identify malicious activity by analysing audit trails of network traffic.

In simpler terms, IDS are like security guards for computer networks. They use advanced technology to detect and prevent cyber-attacks, especially as more services move to the cloud. By analysing patterns in network traffic, they can identify and block potential threats, helping to keep data safe and systems running smoothly.

1.2 Project Objective

1. To improve or boost the performance of prediction.
2. To accurately and effectually predict the original images and forgery images.
3. To reduce the forgery images.

1.3 Problem Statement

With the widespread availability of editing software, creating fake images has become a common problem. To determine the authenticity of an image, forensic experts use techniques like Error Level Analysis (ELA) to compare the compression ratios of the original and altered images. Since fake images typically undergo different compression processes than genuine ones, this method can reveal discrepancies. Additionally, analyzing the metadata of an image can help assess its authenticity. However, it is important to note that metadata can be altered, posing a challenge to this method.

Chapter 2: LITERATURE SURVEY

2.1 Existing System

The existing approach has difficulty classifying and predicting faked photos. Without relying on pre-established equations, machine learning algorithms use computational approaches to extract knowledge from data. As more samples are encountered, these algorithms perform better. Machine learning's subfield of deep learning excels at this kind of work.

2.2.1. Disadvantage

- Possibility of high error.
- Sometimes cause data inconsistency
- Pre-trained models on one task might not perform well on another. The model can need a lot of retraining if the pre-trained features aren't appropriate for the new task.
- Transfer learning may not work if the pre-training task's data distribution greatly varies from the target task's. It's possible that characteristics acquired in one area won't transfer well to another.
- Sometimes, especially when the target dataset is small, transfer learning might result in overfitting. A tiny dataset can be used to fine-tune the pre-trained model, which may have a high capacity and lead to noise in the data being remembered.

2.2 Proposed System

It is difficult for the existing setup to accurately predict and sort fake photos. Instead than relying on pre-established formulas, machine learning algorithms employ a variety of techniques to acquire knowledge directly from data. As they study more samples, these algorithms become more proficient at what they do. This is where deep learning, a subfield of machine learning, really shines.

2.2.1. Advantages

- Relying as little as possible on pre-processing chores minimizes the need for human interaction during the development of capabilities.
- A comparison was conducted between the Deep Learning classification system .
- CNNs don't need human feature engineers; instead, they automatically extract features from raw data, such photos. They are very good at catching complex patterns because of their capacity to acquire hierarchical representations.

- Convolutional layers, which learn local patterns and progressively merge them to capture global information, are how CNNs take advantage of the spatial structure of data. For applications like object detection and image classification, where spatial relationships are crucial, this hierarchical approach works effectively.
- Convolutional Neural Nets have an intrinsic characteristic called translation invariance, which allows them to recognize patterns regardless of where they are located in the input. This feature is critical for applications like object detection and picture segmentation, where objects can appear in different places within an image.
- Convolutional layers in CNNs employ shared weights, which lowers the number of parameters and improves model efficiency. The model's capacity to generalize to previously unknown data is enhanced by this parameter sharing, which also aids in the transfer of patterns discovered in one area of an image to other areas.

2.3 Review of Literature

Concerns over the integrity and authenticity of visual content transmitted across several platforms have increased in recent years due to the widespread availability of digital alteration tools. In order to address this difficulty, scientists have investigated a wide range of methods for identifying image forgeries. With an emphasis on machine learning techniques, this review of the literature offers an overview of the most recent techniques for detecting image forgeries.

- Traditional Techniques:

  Conventional methods for detecting image forgeries mostly depended on heuristic-based analysis and manual inspection. These techniques frequently involved looking for discrepancies in an image's lighting, shadows, and angles in order to spot possible changes. Although somewhat successful, these methods were labor-intensive and unable to scale, which prevented them from being useful for large-scale dataset analysis.

- Forensic Analysis:

  Error Level Analysis (ELA) is one of the forensic analysis techniques that gained popularity since it may identify discrepancies that are introduced during picture alteration and compression. ELA works on the premise that different areas of an image will have varying amounts of compression depending on the changes made to it and then recompressed. Examining these differences closely may allow analysts to find evidence

of manipulation. Nevertheless, the capacity of forensic analysis methods such as ELA to identify more complex types of picture modification is limited.

- Machine Learning Approaches:

  With the development of machine learning, scientists started investigating data-driven methods for detecting image forgeries. These methods use attributes that are taken from photos to train models that can tell the difference between fake and real information. Conventional classifiers that rely on manually created features like texture, color histograms, and edge patterns, including Support Vector Machines and Random Forests, have been used extensively for this purpose.

- Deep Learning Models:

  Deep learning algorithms, which can automatically extract pertinent features from unprocessed picture data, have become effective instruments for detecting image forgeries. Convolutional Neural Networks (CNNs) have proven to be remarkably effective in a range of computer vision applications, such as object detection and image classification. For the purpose of detecting forgeries, researchers have modified CNN architectures. To acquire discriminative characteristics, models are trained on extensive datasets containing both real and altered images.

- Ensemble Methods:

  The robustness and generalization of forgery detection systems have been demonstrated to improve with the use of ensemble approaches, which aggregate predictions from several base models. Combining the outputs of several classifiers using strategies like bagging and boosting has improved performance when compared to using the models alone.

Latest Developments in Deep Learning for the Detection of Image Forgeries.

- Advanced CNN Architectures: Examine the efficacy of different CNN designs, such as Resent, Dense Net, and Efficient Net, in identifying subtle characteristics suggestive of image modifications. Talk about how these designs allow for deeper model training and deal with problems like disappearing gradients.

- Attention Mechanisms: Examine current research that uses attention mechanisms to improve the detection of forgeries. Emphasize the ways in which attention mechanisms enable models to concentrate on pertinent image regions and enhance the precision of detection, particularly for subtle forgeries.

- Generative Adversarial Networks (GANs): Examine how GANs can simulate various manipulation strategies to train robust detection models, in addition to producing realistic forgeries. Talk about the usage of GAN-based techniques for data augmentation in datasets used for forgery detection.

- Transfer Learning: Give a detailed explanation of how transfer learning approaches, such as train on huge data sets (like ImageNet) and adjusting on smaller, more domain specific datasets.

Exploring Novel Feature Extraction Techniques

- Learned Representations: Examine current methods for extracting features using deep learning, with an emphasis on learnt representations such hierarchical features and embeddings. Talk about how learned representations are superior to handcrafted features for identifying intricate patterns in image frauds.

- Graph-based Representations: Learn how to represent image structures and interactions using graph neural networks (GNNs) and see how graph-based methods may capture context-aware information that is essential for forgery detection.

Adversarial Attacks and Défense Strategies

- Adversarial Examples: Describe specific instances of adversarial attacks on forgery detection systems and discuss the most recent Défense strategies, such as input preprocessing, resilient optimization, and adversarial training.

- Domain Adaptation: Talk on the latest developments in domain adaptation strategies for forgery detection, with a focus on methods that improve model generalization for a variety of manipulation types and imaging scenarios.

Interdisciplinary Approaches and Collaborations

- Cross-Disciplinary Research: Showcase effective partnerships between research institutions, businesses, and law enforcement organizations working to create useful forgery detection systems. Demonstrate how multidisciplinary perspectives lead to better solutions.

- Datasets and Benchmarking: Examine current initiatives in benchmarking and dataset curation for the purpose of detecting forgeries, with a focus on the significance of standardized evaluation metrics and procedures.

Ethical Considerations and Privacy Implications

- Fairness and Bias: Talk about possible biases in forgery detection algorithms and how to counteract them so that different demographic groups perform equally.

- Privacy Concerns: Talk about the privacy consequences of forgery detection technologies, especially as they relate to user consent, data storage, and the protection of individual private rights.

Real-World Applications and Case Studies

- Journalism and Media Forensics: Provide case studies that illustrate the use of forgery detection technologies in journalism to thwart disinformation and confirm the legitimacy of visual content.

- Legal and Law Enforcement Use Cases: Emphasize the function of forgery detection in court cases and law enforcement inquiries, illustrating the difficulties and achievements encountered in the actual world.

Title: Image Forgery Detection Based on Gabor Wavelets and Local Phase Quantization [2]

Year: 2015

Author: Isaac MM, Wilscy M.

Methodology-

Image forgery detection is an important field of study because of the growing amount of fake photos that are making their way around the internet and social media, which has sparked concerns about ethical and legal issues. Researchers are developing methods to accurately distinguish between real and fake images, as well as pinpoint the precise position of the phony. There is still no foolproof way for detecting image counterfeiting, despite the several approaches that have been proposed.

With our method, we suggest a fresh approach to picture forgery detection that makes use of data pattern analysis. Our method makes use of CNN to forecast the phony images by extracting information from the data. The layered technique makes it easier for the system to anticipate and identify bogus images. This method's main goal is to distinguish real photographs from manipulated ones with accuracy by focusing on feautre-based information.

Advantage

- Effectively more accuracy.

Disadvantage

- Fail to generalize to detect new fake images.

Title: Pixel and Edge Based Illuminant Color Estimation for Image Forgery Detection [4]

Year: 2015

Author: Youseph SN, Cherian RR.

Methodology-

Although images are essential for communication, it is becoming more and more critical in digital domains to ensure their security. Image manipulation has become commonplace due to technological improvements like cameras with excellent resolution and software for editing photos. In particular, image splicing—a popular kind of photo manipulation—is covered in this work.

The study is focused on finding irregularities in the illuminant color of photos, which is an effective way to spot fake photographs. Inconsistencies can be found by comparing the illuminant colors of different photographs. This is accomplished by calculating the color of the illumination at the pixel and edge levels in various image regions.

Advantage

- Easy to detect the pixel and edges.

Disadvantage

- Sometime more loss of pixels.

Title: Pixel Based Image Forensic Technique for Copy-move Forgery Detection Using Auto Color Correlogram.

Year: 2016

Author: Malviya AV, Ladhake SA.

Methodology-

In blind image forensics, establishing a digital image's reliability is essential. We are concentrating on identifying cloned or removed fakes, a prevalent kind of counterfeit that is often seen in this industry. removing and making clone fake creates pixel-by-pixel inconsistencies in the altered image by copying portions of the original image within the same image.

Advantage

- Easy to predict the pixel.

Disadvantage

- Sometimes loss of pixels.

Chapter 3. PROCESS DIAGRAMS

3.1 Conceptual illustration

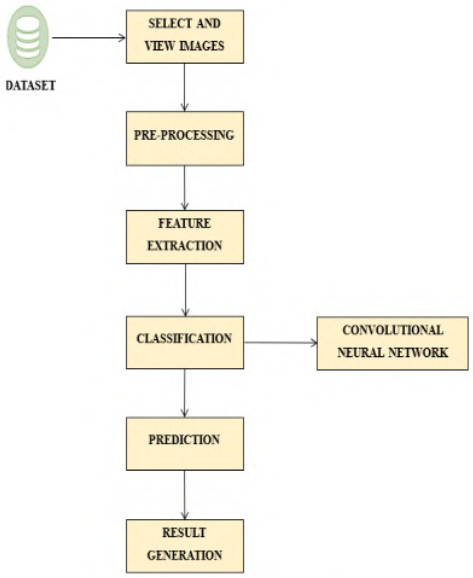

Figure 3.1 Conceptual diagram

3.2. Flow chart

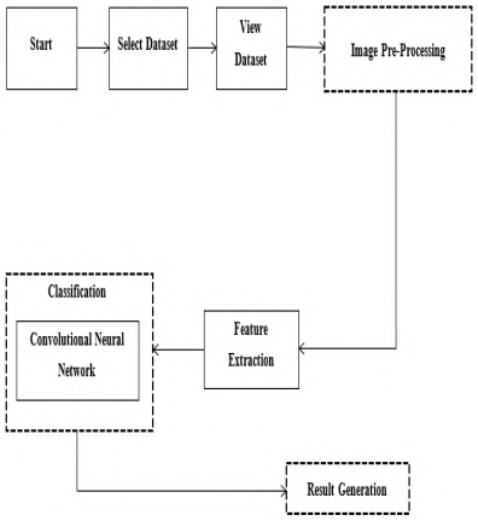

Figure 3.2 Flow sheet

3.3 Activity

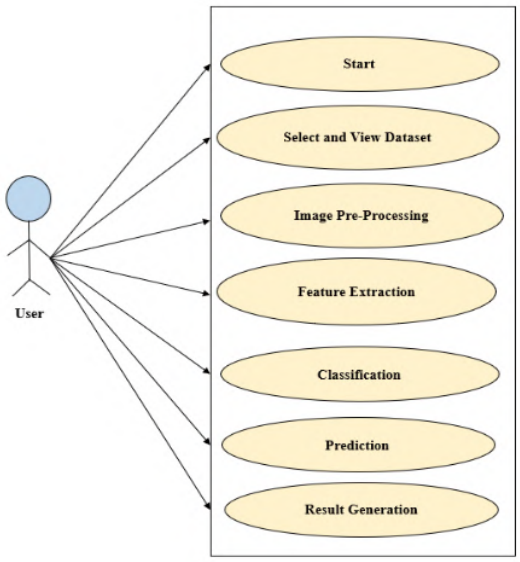

Figure 3.3 Use case

3.4 Entity relationship

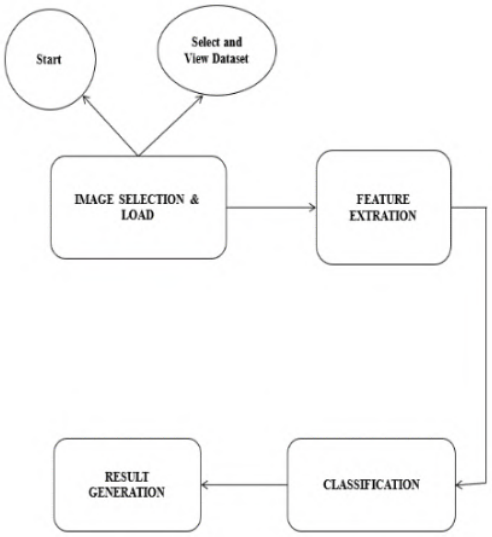

Figure 3.4 ERD

3.5 Class

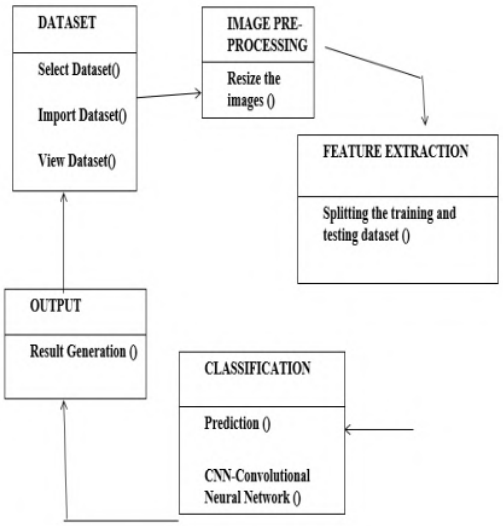

Figure 3.5 Class

3.6 Event

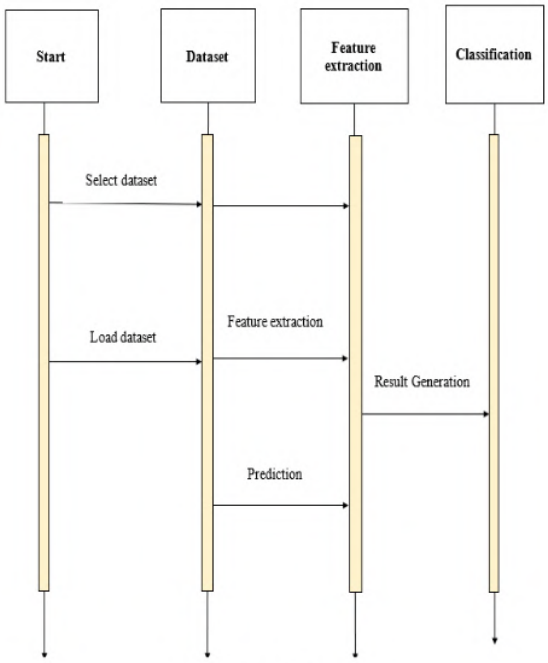

Figure 3.6 Event

3.7 Control flow

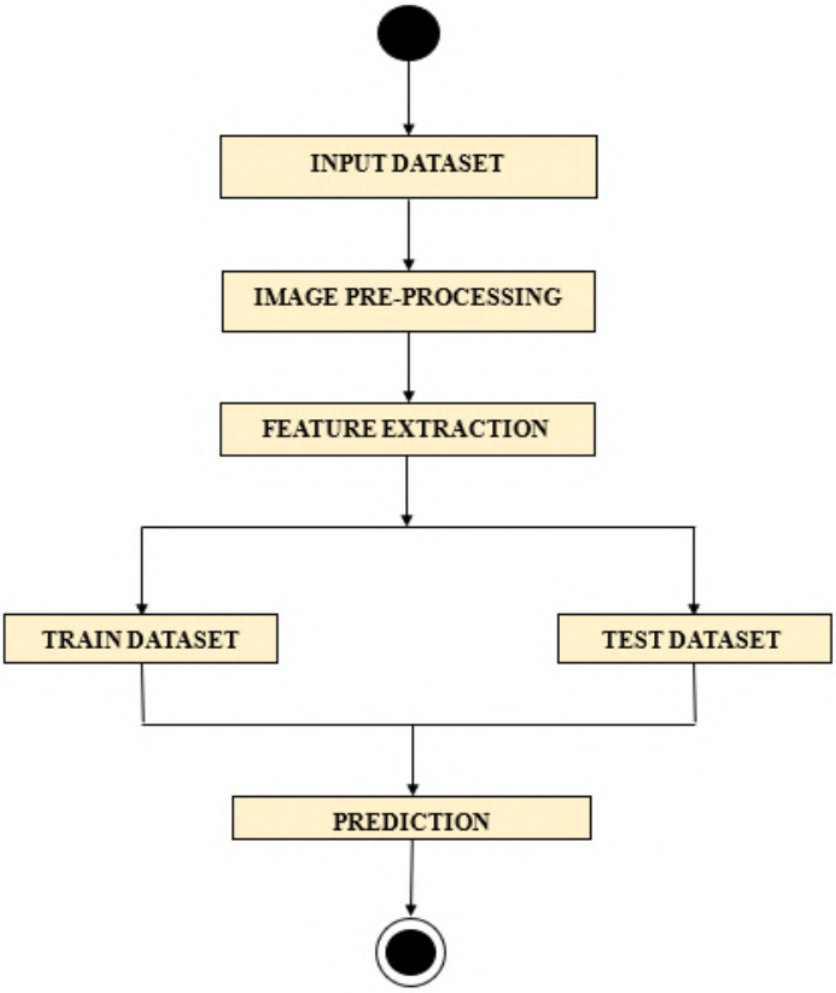

Figure 3.7 Control flow

Chapter 4. IMPLEMENTATIONS

4.1. Existing System

The current methodology struggles to precisely classify or forecast image forgery. Machine learning algorithms, devoid of predefined equations as models, employ computational methods to directly glean information from provided data. Moreover, as the volume of samples escalates and increases, these algorithms dynamically adjust, exhibiting diminished performance. A specialized branch within machine learning, known as deep learning, encompasses such techniques.

4.1.1. Disadvantages

- With data and samples increases the possibility of getting error high increases.
- Can cause data inconsistency.

4.2. Proposed System

A convolutional neural network is dependent on the depth of the feature. There are few neural networks that work on two layers; they do not have the depth, or they do not have more layers. Developing multiple layers and complex as well as deeper neural networks is known as deep learning. There are many types of neural networks; one of the more used neural networks is convolutional neural network also known as CNN. There are many tasks that CNN can perform, like speech recognition, natural language processing, classification, and more, but from all of this image processing is by far the best thing CNN can do. CNN belongs to deep learning because of the layers it has at least a dozen. However, for basic tasks, we can use CNN with less layers, which really isn't as deep as deep learning. The reason for CNN to become so popular and widely chosen neural network for image classification is because of its capacity to learn from the data and extract very complex attribute from the unprocessed image data. The real reason for using CNN in image processing is for its ability to predict pattern and extract attribute or feature from the data provided.

In CNN we will be working on the sequential model, which is a common architecture used for building the network. The sequential model helps us to create a CNN by adding layers one by one that are sequentially stacked, forming a linear stack. First, the initial layer of the sequential model is the input layer, which receives the input data. When it comes to images, each image is usually assigned as a matrix which has given its own pixel values, which is accepted as input by the CNN's input layer. After this, you add a convolutional layer which is the first layer of our model. This layer works on the data to provide it with filters. This filter helps the CNN detect patterns like edges, textures, shapes, or important features. As the filter window slides across the grid, it highlights and makes a new "feature map" where certain features are present in the image. After the convolutional layer, Max pooling is frequently used

to reduce and simplify the information in feature maps while preserving the most crucial information. This makes the grid into smaller squares, and for each square, and also the existing method faces challenges in accurately categorizing or predicting image forgery. Machine learning algorithms, free from predefined equations as models, utilize computational approaches to directly extract insights from available data. Furthermore, as the dataset size increases, these algorithms adapt dynamically, experiencing a decline in performance. Deep learning, a specialized subset within machine learning, encompasses such methodologies.

Table 4.1. Details of dataset CASIA 2.0

	FAKE	REAL	TOTAL
Total	980	1081	2061
Training(70%)	678	779	1457
Testing(30%)	302	302	604

The dataset we have consists of fake and real images (Table 4.1). We have to preprocess the data to improve the quality of training and accuracy (Figure 4.1). After the images are preprocessed, we feed the data into the sequential model, which will undergo multiple convolutional layers and max pooling as stated before, then into the flattening layer, finally feeding into the dense layer. We make the sequential model consist of three convolutional layers which will is not consecutive, then after the filters are applied we have to use max pooling which are three in total for this model, then after this we have to connect the layers to the fully connected layer for that we will use one flattening layer, and then this will connect to the dense layer, which is defined below-

-The initial layer is convolution which comprises 32 filters, the size of the grids are 3 and 3, and utilizes the "relu"

- Then Max pooling sized 2-by-2

-Next layer is another convolutional which comprises 64 filters, the size of the grids are 3 and 3, and utilizes the “relu”

- Then second max pooling layer sized 2 and 2

- Then there is third layer which is again convolutional comprises 128 filters, each size of the grids are 3 and 3 , and utilizes the “relu”
- Then the third max pooling layer sized 2-by-2
- Then we use flatten layer
- Then dense layer which utilizes “relu” activation function with 256 neurons
- Then it is connected to 1 neuron with “sigmoid” activation function

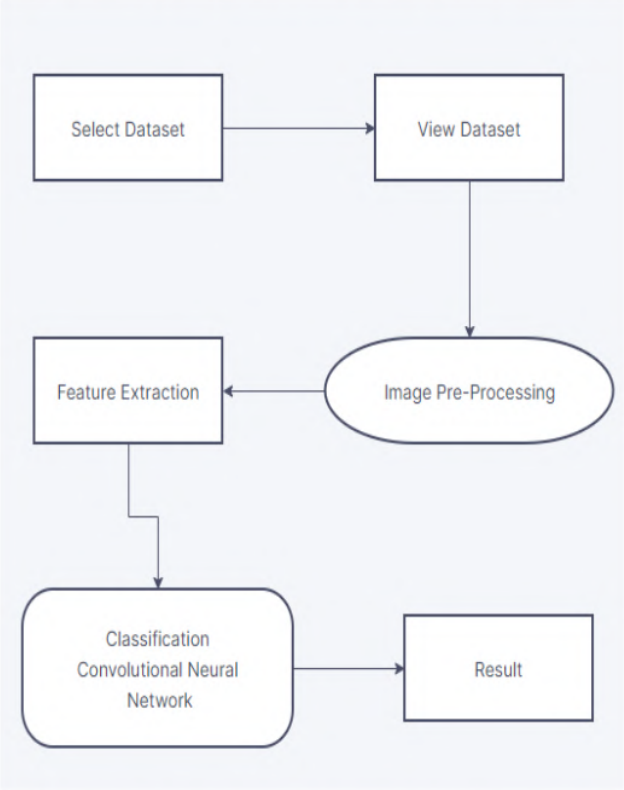

Figure 4.1. Flow chart of image classification system

Figure 4.2 Preprocessed images from dataset in grayscale format (CASIA 2.0)

Data augmentation was applied to enhance the model to generalize to new data. This expands the dataset by using random changes in images such as shifting it, rotating it in some angles and even zooming in. Rescaling was done between 0 and 1. Images were randomly rotated by up to 10 degrees and were randomly shifted 10% horizontally as well as vertically. Images were randomly zoomed in by 20%.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 61, 61, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 30, 30, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 28, 28, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 256)               6422784

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 6,516,289
Trainable params: 6,516,289
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.3 Proposed model for image forgery detection

Total trainable parameters in the model were 6,516,289. During the training phase, the neural network's weights and biases are represented by these parameters. How the model turns the input image into a classification output depends on the precise values of these parameters.

4.2.1. Advantages

- There is minimal requirement of preprocessing using cnn which reduces the efforts of human and also the inconsistency of data due to that.
- Comparison was made between the classification algorithm on Deep learning

4.3. Implementation

4.3.1. Modules

- Choosing the correct images for the dataset
- Making the data clean
- Splitting for further prediction
- Implementing algorithm
- Predicting the Image
- Results

4.3.1.1 Choosing the correct images for the dataset

- We have to gather the images which are fakes and original for dataset. which will be ground truth.
- Or choosing the existing dataset which provides both information of the originality and fake images.

4.3.1.2 Making the data clean

- Since the data we get might have inconsistency to it or maybe the scaling might differ we have to rescale it
    - Normalizing the scales of the data
    - Inserting the data
- Normalizing the scales of the data: Rescaling the given data of both original and fake ones.
- Inserting the data: The data which has been rescaled will be used in categorical data. For which we have to make it in array form so that the model can take it which is very common in deep learning.

4.3.1.3 Splitting for further prediction

- For this we have to split the given data we have. This will be used for training purpose and then checking the model if it is predicting correctly or not.

- When we split the data, the large part goes to the model which will be used to train, and the smaller part of the data goes to the Testing or the evaluating the prediction oof the data.
- Dividing the data into two parts in very important for the model for performance check.
- Usually, we use a large part of data for training to get the model more training while we use smaller part for evaluating or performance check.

4.3.1.4 Implementing Algorithm

Convolutional neural networks are basically a type of deep neural network which is extensively used in deep learning for various purpose and understanding the data. This method has been used in many different fields for predicting like financial time series, NLP, for recommending and even for medical purpose like diseases prediction and other things.

We have other techniques as well which can be used like Multilayer perceptron. In this the layers are connected to each other like the neuron from previous layer to the next layer. This makes the MLP's connectivity dense which tends to overfit. To tackle this situation most of the times we try to adjust loss function with some of the weights.

CNN uses different type of regularization. They Basically try to use the pattern from the data which is hierarchical to convert it into more complex pattern using the easy one. This makes the CNN with lower connectivity compared to MLPs and even lowers the complication.

This neural network which are directly inspired from the animal visual cortex. Which is a biological process. This biological process directly influences the design of the CNNs, which is beneficial for recognizing images using patterns in visual data.

4.3.1.5 Predicting the images

- This task is where the images are feed, and it tells the difference between the fake and original images.
- This project aims to improve the prediction accuracy by enhancing overall performance.

4.3.1.6 Results

Since we did the classification and predicted the results, we will need some technique to see the productiveness of the model win which we can use multiple metrics. which can be Precision which simply means from the positive prediction we made how many were actually correct, whereas Accuracy instead of just positive we will use all the prediction we made, how many were actually correct. Other metrics are also there like F1 score and Recall.

Chapter 5. System Requirement

5.1 Hardware Requirement

- Hard-Disk requirement : 200 GB
- System-requirement : Intel-Pentium 4 3.0GHz
- Minimum Random access memory requirement : 4GB

5.2 Software Requirement

- O/S : Min Windows 7.
- Language : Python
- Front End : Anaconda Navigator – Spyder

5.3 Software Description

Python

A versatile advanced language is called Python. The user finds this language to be pleasant and is astonished by how simple it is to use; instead of worrying about the syntaxs and framework, users can concentrate on solving problems. When it comes to its formal introduction, Python is described as a powerful yet easily learned computer language. This language is one of the easiest and most effective approaches to utilize when it comes to object-oriented programming, providing its users with advanced data structures that are efficient and straightforward. Python is among the greatest languages because of its dynamic typing, simple and elegant syntax, and ease of use. This language is perfect for filtering and scripting developing apps across different systems and industries. The following section will detail the remaining highlights of this.

Some of the Characteristics

Although it has strict restrictions, Python is a language that is primarily distinguished by its elegant and simple syntax, as well as its simplicity and minimalism, which make it simpler to comprehend English. Python's ability to mimic pseudo-coding is one of its strongest points when it comes to problem solving. It frees the user from worrying about syntax mistakes and allows them to concentrate entirely on solving the problem at hand rather than on other things. which is why programmers consider this language to be perfect.

Because Python has such a basic syntax, learning it is really easy. Since it is open-source and free, anyone is welcome to use, alter, and distribute it in fresh applications, promoting a knowledge-sharing community.

Because Python is an advanced language, it abstracts away some extremely minor issues, such as handling memory, allowing programmers to focus solely on creating code. It enhances developers' performance.

For Everybody Python's open-source nature makes it incredibly easy to use on a variety of platforms, including Mac OS, Windows, GNU/Linux, and numerous others, all without the need to even launch platform-specific configurations.

Python is an interpreted language, unlike compiler-dependent languages like C/C++, which require translation to binary code. Alternatively, Python may run code straight from the source, translating it into byte codes before translating it into easily navigable machine language. This makes development more transparent and improves portability because programs can run without worrying about library dependencies or compilation.

This language gives developers versatility in their coding approaches by supporting both the sequential and OOPSstyles.

Programs written in sequential-oriented programming languages are organized around their functions and procedures, with code that can be reused serving as the foundation.

On the other hand, the oops language creates programs based on objects that include functions and data. In general, Python's approach to oops is strong and nevertheless simple when compared to other languages for programming like C/C++, C#, or Java.

Because of Python's flexibility, important code parts can be written in faster coding languages like C++ and C and then seamlessly incorporated into Python projects by the compiler.

Implementing Python in the context of C/C++ programs is a simple task. Because Python is integrated into C/C++ uses, developers are forced to support scripts in their software. With its extensive standard library, Python offers you a plethora of functionality from the very beginning. Regular-expressions, , data bases , documentation, unit testing, threads, internet viewers, and numerous additional features are included in this library. Because Python is designed with the "Batteries Included" philosophy, it is user-friendly no matter where it is installed. Additionally, this language has a wealth of excellent libraries available on the Python Package Index, which makes development incredibly simple for users.

We'll be studying feasibility analysis: studies here, so let's get started. Three key areas need to be discussed in

Economic feasibility: This means that in order to determine whether or not to implement the suggested system, its benefits must be weighed against its highly anticipated costs. Additionally, it ensures that the software and hardware needed for development won't significantly increase costs.
Technical feasibility: This discusses whether the suggested system can be supported without taking into account the need for major upgrades using the hardware and software that are currently in use. It confirms that the facilities currently in place can be used to develop the system.
Behavioural feasibility: The developers may require additional training and are frequently reluctant to accept changes, which could result in additional expenses for the company. The ability of the suggested system to produce comprehensive reports on demand could potentially lessen resistance by providing users with quick access to the data they require.

5.4 Testing of models

When we create a system the most important part to ensure the product is good and effective there are many things we need to ensure before sending it to the consumers. Like checking if the system is working the way it is intended with the best effective manner. When we use the model and runs the systems we will likely get errors and mistakes. When we test the models we basically finding how successful it is in its works. IT comprises the idea that if it is true than it is possible which means if this model works the way it intended than it will produce the results correctly as well. For making the product ready for consumers there should be multiple testings are needed to perform to ensure the products quality and that way you can understand the system work is correct and accurate.

Integration Testing:
This is the part where the system is asked some question which are answered by it. So basically we feed the system some few inputs which will run and then we can find the performance of the model. There are few types of integration testing as well.

Unit Testing:
This is the part where the models are divided into smaller parts. Like to process the whole system we need to check if the all small individual parts are working properly or not. So the model is broken down from function to function and then we test all the small parts to check if those are working perfectly or not.

Inspection Methods:

Black Box Testing:

Checks for errors or missing functions; it also addresses performance and problems, as well as external database access, initialization, and revocation. Black box testing essentially focuses on evaluating externally, system's behaviour without even trying to comprehend its inner workings.

White Box Testing:

Test cases are created by this using the control mechanism of the procedural design. Testing each of a module's separate paths at least once is the aim of white box testing.

Application Inspection Methods:

Dynamic Testing:

This is how software testing is done; once testing is finished and interface errors have been fixed, software assembly is completed and validation testing starts. Validation testing makes sure that the program performs as the user would expect.

End User Testing:

One crucial technique is this testing, which confirms that users approve and are satisfied with the system as it is designed. It also implies that in order to make the required adjustments, you will stay in close contact with potential users during the whole development cycle.

Output Testing:

This testing strategy makes sure achieved the best system you've suggested format. In order to ensure that the system satisfies user needs without any problems, it also verifies the print and screen output formats.

Chapter 6. Conclusion and Future Enhancement

6.1 Result and discussion

Well to check how good our method is we have to use this model in dataset which is CASIA 2.0. We have the dataset in multiple folders with original and fake images. The total number of pristine images are 1050 and fake images are 900. These images have different perspective and categories such as outdoor and others. The resolution of all images is a bit different which are from 800 by 600 pixels to 384 by 256 pixels. This task was performed in a device whose processor is Intel Core i5-1235U twelfth generation. Intel iris graphics. the ram is eight GB. This task was performed in visual code.

Few terms which are defined for the work -

• T_Images – each and every image which are in our smaller part of the dataset (testing data).

• True_P- Accurately distinguished altered pictures.

• True_N- authentic images that have been correctly identified.

• False_N- or false negative, refers to actually false images that have been mistakenly identified as accurate images.

• False_P- actually correct images that have been incorrectly identified as tampered with images. For the purpose of evaluating and contrasting the proposed method with others, we will use other things as accuracy etc.

These are estimated ate as follows:
Accuracy = (True_P + True_N)/T_Images

Recall = True_P/(True_P + False_P)

Precision = True_P/(True_P + Fale_N)

Fmeasure = (2xPrecisionxRecall)/(Precision+Recall)

6.1.1 Training of the model and testing

To check the performance of the method or model we used. Let us understand the data first. The database is divided into two primary folder and two subfolders each. The two primary folders are for train and test respectively. Inside the train folder the 1 folder is of original image which is 678 and the other is fake images which is 779 and for the test folder we have 302 original image and 302 fake images. Which totals around 30 percent of the data so for the 1st primary folder which is train we have 70 percent of data, and the remaining is for testing. We utilized the adam optimizer for the method. the size of the batch was 128. and an underlying learning proportion of 1:1. Figure 6.1 visualize the accuracy while the method was in training and as well as testing.

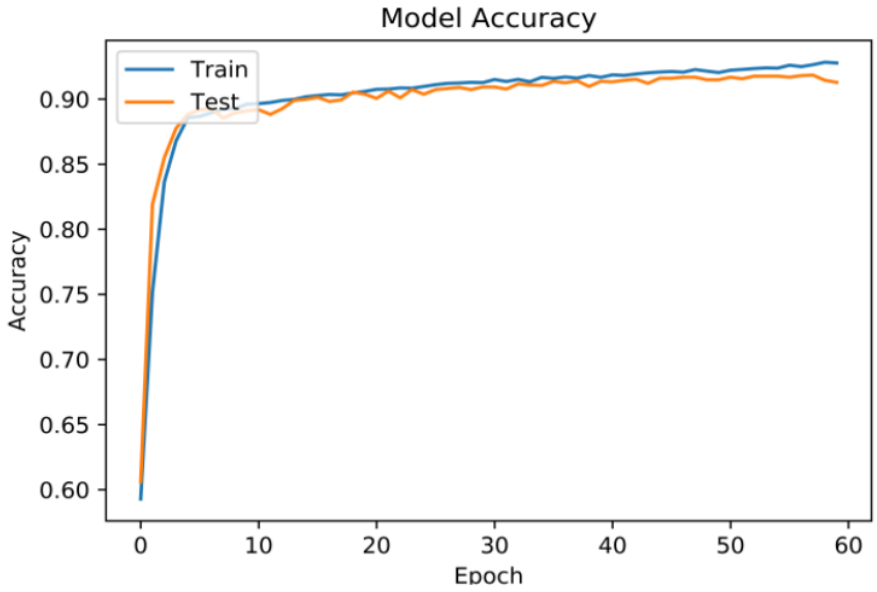

Figure 6.1. The accuracy of model while performing task

6.1.2 Image Prediction

Here in this part, we are taking a look in the code where we give the model the image. Now we took the images from the data we have and provided the location of the image to the model and its prediction of the image whether its real or fake.

```
img = input("Enter Location of Image to predict: ")
test_image_1 = ImagePrediction(img)
```

```
1/1 [==============================] - 0s 182ms/step
Prediction:  Fake
```

Figure 6.2. Prediction of whether it is fake or real

6.1.3 Comparison with Other Techniques

Since this method take some amount of time to process and predict the images authenticity. We need to compare our method to other methods.

Here we compared our method to the Transfer learning:

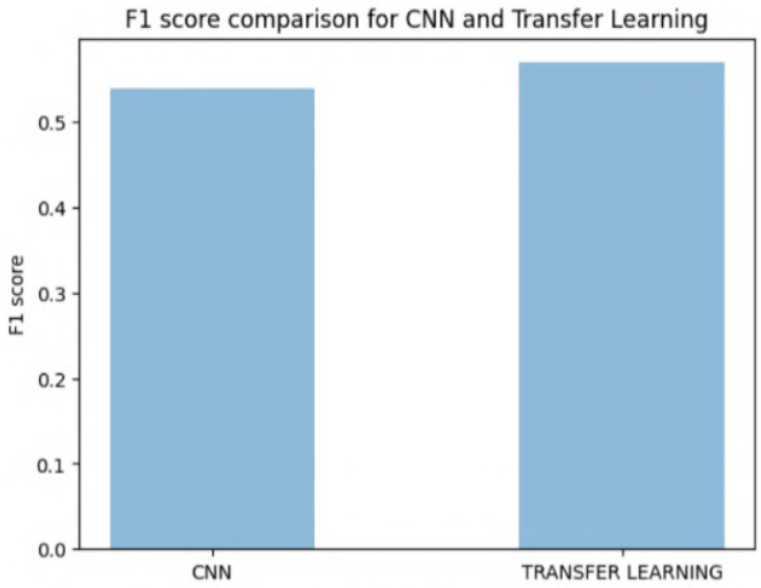

Figure 6.3. Score Comparison for CNN and Transfer Learning .

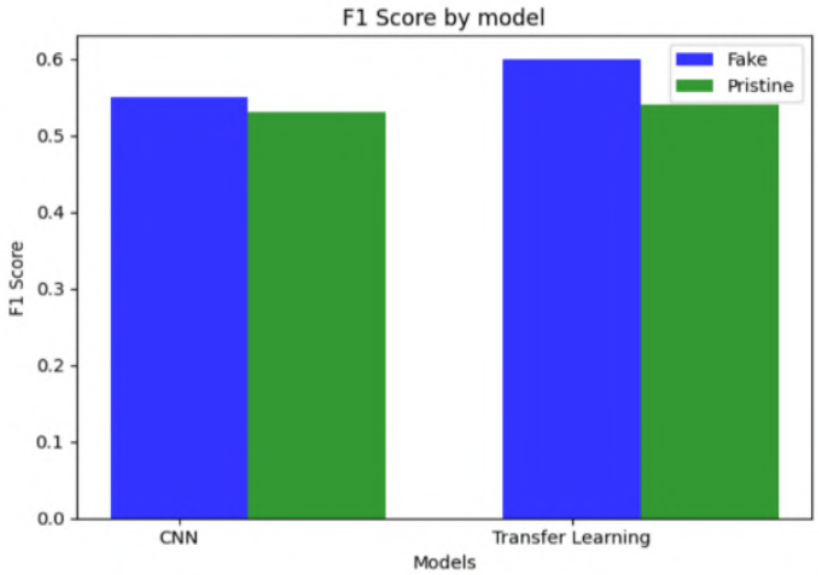

Figure 6.4. Represents the model trained, it is the pictorial visual of the F1 score by the model.

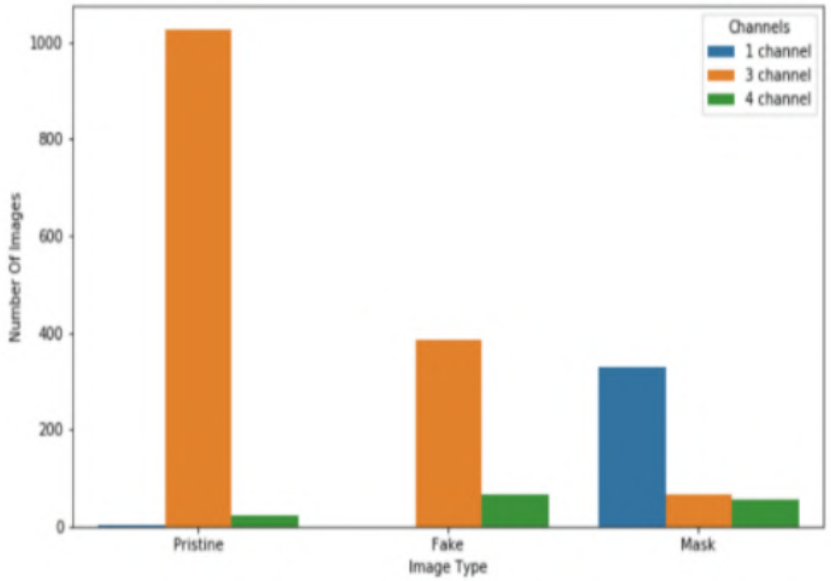

Figure 6.5. Visual representation of the three types of images in database, Pristine Fake image type and the masks.

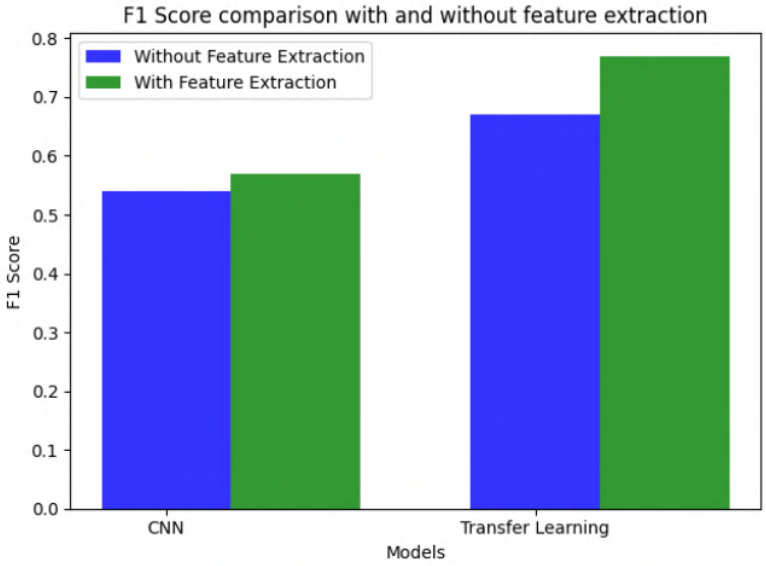

Figure 6.6 F score of the cnn and transfer learning

Figure 6.7 A fake sample

In the figure 6.8 we have given the visual examples of our methods performance base on some cases on

the model we have made.

(A) (B) (C)

Figure 6.8. Some visual images representing our methods presentation: (a) TN case, (b) FN case, (c) TP case.

6.1.4. Processing Time Comparison

Here we have some time comparison of our model vs some other models. We have a graphical representation of the comparison between the mantra, cat and buster with our model as well (Figure 6.9). The comparison is based on the time taken by the methods to process the data or image and perform the prediction on the data to determine which ones are real and which ones are tampered. The proposed method is quicker and faster to show us the result of identity of the image is tampered or real.

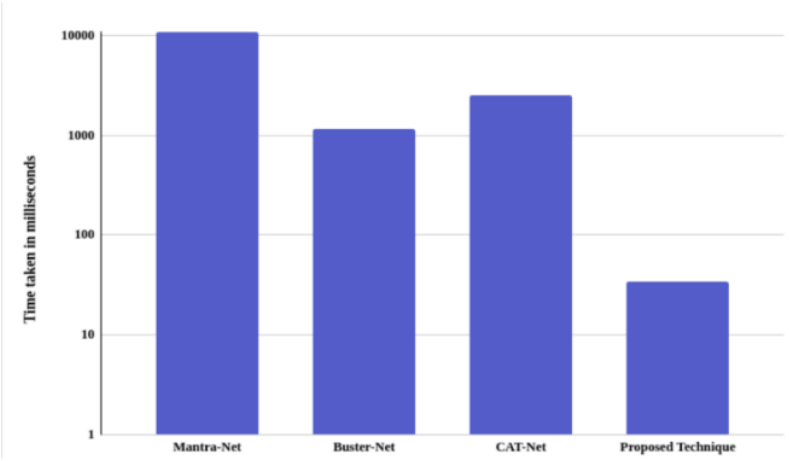

Figure 6.9. Comparison of our model to other models.

Here in this table 6.1, we are going to see and compare between our model and the Existing System.

Table 6.1.Comparing our method to some existing methods

Sr. No.	Our Method	Existing System
1	CNN is usually used with multiple layers . But for simpler task we can use less layer. But if we use model with 2 layer it is not considered as deep.	The system which are already there are not capable to predict as well as classify the actual and fake images.
2	When we use multiple layers with complex neural networks it falls in deep learning methods. CNN are most commonly and worldwide used to build a neural network.	Machine learning algorithms use computational methods to learn information directly from data without relying on a predetermined equation as a model.
3	Cnn are widely known for its speech recognition, nlps tasks image processing and more.	The models tend to decrease their performance as their number of data increases. Whereas deep learning is doesn't.
4	This has the higher accuracy of predicting if the	Because the other technique works on

	images are tampered or not	pixel level to determine the tampered images. This makes their model prone to false positive. With is really decrease the classifying ability of the model.
5	This model is suitable for both fast and slow machines	These system works for fast machine since the model is slow working on slow machine makes it slower.

Thusly, considering the exploratory outcomes, the accompanying ends can be drawn: the recommended procedure predicts accurately three out of four pictures, regardless of the model not being thoroughly prepared. Expanding epoch close to around 40 will probably make the model impeccably prepared, however preparing will require five to six hours.

6.1 Conclusion

In conclusion, deep learning has established itself as a potent instrument for detecting tampered or fake images, assisting in the preservation of the integrity of visual content in a variety of fields, including security, forensics, and journalism. By utilizing new and advanced neural network, specialists have taken critical steps in recognizing tampered pictures with high exactness, even in situations where natural eyes could battle to distinguish changes.

Even with this far there are still so many challenges which are yet to be overcome in this field. For this type of image detection, we need bigger and more labelled data toward the model's training. And to acquire this type of dataset is very difficult due to the complexity of tampered images as well as diversity. Moreover, to make sure the robustness and generalizability of the model across the different types of tampered detection methods and the meaning or content of the image is ongoing concern.

Looking forward, future exploration in image fraud detection utilizing profound learning ought to zero in on tending to these difficulties. This incorporates growing more productive and successful calculations

for preparing profound learning models with restricted named information, investigating novel strategies for recognizing complex forgery techniques, for example, deepfakes, and improving the interpretability of these models to all the more likely comprehend their dynamic cycles.

Moreover, coordinated effort between specialists, industry specialists, and policymakers is significant to creating norms and rules for the arrangement of fake image detection advances dependably and morally. By proceeding to enhance and team up, we can additionally propel the field counterfeit image recognition and add to establishing a more reliable digitalized environment.

6.2 Future Enhancement

For future works, we are looking for more depth knowledge on CNN architecture to get better accuracy. While also not stop looking for other fake image detection techniques and approaches for determining which is fake and real images.

This includes understanding improving and enhancing the existing CNN architecture by utilizing more advanced methods and techniques to optimize and increase accuracy. We will experiment and work on different types of network architectures, layer designs, activation functions and other thing to identify what is the best and good way to optimize our task.

Since there are more and excellent image forgery techniques which can be used. We will learn more about some techniques like Generative AI and some other deep learning methods. RL is also a type of machine learning methods.

Our goal is to greatly improve the accuracy and resilience of our fake image detection system by fusing CNN architectures with innovative and image processing techniques. By taking a comprehensive approach, we will be able to distinguish between authentic and fake photographs more effectively, which will enhance the system's overall functionality and dependability.

Capstone

ORIGINALITY REPORT

3%	2%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

1%

★ Dipak Agrawal, Hitesh Makwana, Shrinal S Dave, Sheshang Degadwala, Vidhi Desai. "Error Level Analysis and Deep Learning For Detecting Image Forgeries", 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), 2023

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		