SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I8P123

Volume 11 Issue 8, 227-243, August 2024

Original Article

Beyond Pixels: Exploring Deep Learning Methods for Image Forgery Detection

Pramod Chathampally[1], V. Mary Amala Bai[2]

[1]*Department of Computer Science and Engineering,*
[2]*Department of Information Technology, Noorul Islam Centre for Higher Education, Tamilnadu, India.*

[1]*Corresponding Author : pramod.Chathampally@outlook.com*

Received: 17 June 2024 Revised: 29 July 2024 Accepted: 14 August 2024 Published: 31 August 2024

Abstract *- Image forgery detection is a critical task in digital forensics, aiming to identify manipulated images to maintain trust and authenticity in digital content. Conventional techniques for identifying image forgeries often rely on handcrafted attributes and heuristics, which have limitations in detecting sophisticated forgeries. The capacity of deep learning algorithms to automatically extract pertinent features from data has made them a promising solution to this problem in recent years. The effectiveness of Convolutional Neural Networks (CNNs), a type of deep learning, in identifying image forgeries is investigated in this paper. The proposed research begins by collecting a dataset from the CASIA V2, comprising authentic and tampered images. Initially, a custom CNN model is constructed and trained on the dataset to establish a baseline performance. Subsequently, transfer learning using the MobileNet V2 architecture pretrained on the ImageNet dataset and is applied to leverage its feature extraction capabilities. However, the MobileNet V2 model demonstrates suboptimal accuracy before fine-tuning, prompting further enhancement. To improve the MobileNet V2 model's efficiency, fine-tuning is employed at epoch 25, resulting in a notable accuracy increase to 94.14%. Compared to the baseline CNN model (93.98% accuracy) and the initial MobileNet V2 model (77.85% accuracy), fine-tuning significantly enhances the model's efficiency in identifying image forgeries. The proposed methodology showcases the potential of deep learning in image forgery detection, offering improved accuracy and robustness in identifying manipulated digital content.*

Keywords *- Image forgery, Authentic, Tampered, Transfer learning, CASIA V2, Compression error analysis.*

1. Introduction

In the modern era, often referred to as the period of technological and informational supremacy, numerous crucial challenges revolve around managing information. Among these challenges is the critical task of safeguarding information against manipulation and identifying those responsible for such attacks.

This task becomes especially pertinent given the prevalence and significance of images and videos as the most pervasive forms of information. Globalization and advancements in technology have made electronic equipment, such as digital cameras, more broadly accessible and reasonably priced. As a result, the use of digital cameras has increased dramatically, leading to an abundance of images taken with different kinds of camera sensors.

The necessity for electronic image formats is increasing in the modern digital era due to social media sharing, online filing, and recordkeeping [1]. A notable feature of images is their universal accessibility; they may provide information to people with low literacy levels as well. Images are a fundamental element of the digital world and are vital for data storage and distribution. Although image-editing software was initially created to improve images, some people abuse it to create false images and disseminate false information [2]. This poses a serious risk because altered images can have permanent effects.

The tampering of digital images, known as digital image forgeries, presents a serious problem since the changes are frequently imperceptible to the human eye. False information is primarily disseminated through social media sites like Facebook and Twitter owing to this kind of manipulation [3].

It is necessary to use digital image forgery algorithms and approaches to detect such forgeries in order to preserve image security, particularly in situations when access to the original information is restricted.

These techniques detect anomalies that have been added to images. These anomalies can appear as uneven feature distributions and heterogeneous alterations in image attributes [4]. There are two primary methods for manipulating images [5], as depicted in Figure 1.

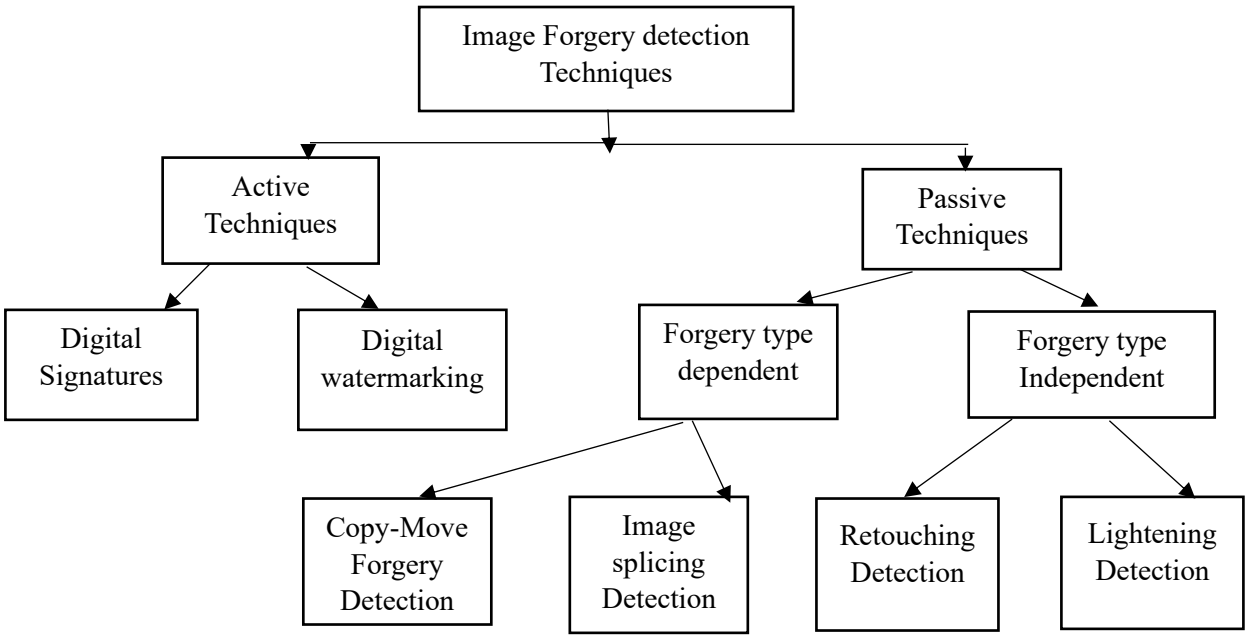


Fig. 1 Hierarchy of forgery detection

The image is enhanced with a digital signature or watermark using the active approach while it is being created, making it possible to analyze it afterwards to see whether it has been altered. Conversely, the passive technique, often known as the blind technique, does not rely on pre-embedded information such as watermarks; instead, it only uses features that are directly derived from the images. There are two further categories into which the passive method can be separated: dependent and independent. While the dependent technique concentrates on finding operations like copy/move and splicing, the independent approach finds changes like compression and resampling. This distinction aids in understanding the various approaches used to detect image forgeries, each of which presents different difficulties and insights.

Forgery detection techniques are one of the primary issues in image security, and they are employed in conjunction with passive authentication in situations where access to the original information is unavailable. This method relies on examining the characteristics of the image while searching for any unusual patterns.

Additionally, this research advances this field of investigation in the following ways:

- To propose a MobileNetV2-based architecture for authenticating genuine and forged images.
- To successfully train the suggested model on the CASIA V2 benchmark datasets by utilizing the transfer learning technique.

The paper is organized as follows in the ensuing sections: Section 2 explores the literature review, while Section 3 defines the proposed system architecture. Following that, Section 4 outlines the experimental findings and provides a discussion, and section 5 provides conclusion.

2. Related Works

The latest strategies for identifying digital image modifications have been made possible by advancements in image forensic techniques. Prior studies [6-8] have investigated methods that examine different phases of an image's history, from capture to compression, in order to detect processing traces. These traces serve as markers of digital authenticity, which are ascertained by means of digital signature verification.

Prajakta Kubal et al. (2023) [9] addressed the growing problem of image forgeries in the digital domain, highlighting the significance of image verification in preserving integrity and thwarting misuse. Their method, called EACN (Error Analysis and Convolutional Neural Network), combined CNNs with error level analysis to evaluate error rates that arise from quality degradation in order to authenticate images. Even though metadata analysis is easily manipulated, EACN, which uses Deep Learning (DL) to detect robust forgeries, obtained an accuracy of 92.10%. Shobhit Tyagi and Divakar Yadav (2023) [10] investigated how easily digital images and videos can be altered, either for good intentions like improving appearance on social media or for bad intentions like stealing someone's identity or damaging someone's reputation. They underlined the necessity for law enforcement to use automated technologies to discern between real and fake media. In an effort to promote security and privacy within the research community, the survey thoroughly examined a variety of image and video modification techniques, commonly

employed methodologies, and sophisticated forgery detection strategies.

The increasing use of image capturing as a result of the broad availability of cameras was addressed by Syed Sadaf Ali et al. (2022) [11]. They stressed the usefulness of images in daily life due to their informational value, but they also mentioned the growing concern about image modification resulting in false information. CNNs have received interest recently despite the existence of established approaches for detecting forgeries. Nevertheless, a lot of CNN-based methods concentrate on particular kinds of forgeries. Therefore, an effective method for spotting hidden forgeries is required. Their suggested lightweight CNN-based solution uses the differences between the original and recompressed images to identify image forgeries, especially double image compression. Although the technique achieves a promising validation accuracy of 92.23%, its applicability to smaller images is hindered by its minimum image resolution requirement of 128 × 128.

Emad Ul Haq Qazi et al. (2022) [12] discussed how advances in technology have led to a spike in data misuse, which calls for reliable techniques to spot manipulation. Their research concentrated on employing a ResNet50v2-based method to detect image splicing, a prevalent type of digital manipulation.

The CASIA v1 as well as CASIA v2 datasets were used in their studies. Through the use of transfer learning and the architecture's residual layers and pre-trained weights from the YOLO CNN model, their model demonstrated higher detection rates for manipulated images.

Davide Alessandro Coccomini et al. (2022) [13] discussed how deepfake generation techniques are developing quickly and how this could seriously threaten social peace by enabling the creation of incredibly lifelike manipulated images and videos. They emphasized how difficult it is for deepfake detection algorithms to update themselves rapidly in order to recognize manipulations carried out using the latest techniques. Using the ForgeryNet dataset, the study evaluated the effectiveness of Vision Transformers and EfficientNetV2 in a cross-forgery scenario.

EfficientNetV2 tended to specialize and perform better on well-known techniques, whereas Vision Transformers showed stronger generalization and were, hence, more capable of recognizing images created using novel techniques.

In response to the increasing misuse of image altering software and the distribution of these images via Online Social Networks (OSNs), Haiwei Wu et al. (2022) [14] addressed the increasing concern regarding the authenticity of digital images. They discovered issues caused by noisy activities like resizing and compression brought on by OSN, which made it harder to detect image forgeries. They suggested a training plan to address this problem by dissecting OSN-induced noise into visible and invisible parts. They greatly increased the robustness of their fraud detection system by including this noise modeling in their training framework. Amit Doegar et al. (2021) [15] addressed the challenge of image forgery detection in real-time applications and online platforms. They introduced a fusion-based decision approach utilizing lightweight deep learning models like SqueezeNet, MobileNetV2, and ShuffleNet. This approach involved two stages: using pretrained weights to assess image forgery and fine-tuning the weights for comparison. Their experiments on the MICC-F220 dataset demonstrated superior accuracy compared to existing methods: SqueezeNet achieved 89.39%, MobileNetV2 reached 92.42%, and ShuffleNet attained 90.90%. Despite these promising results, the method's performance might be limited by the dataset's size and diversity, potentially affecting its generalizability across various image manipulation scenarios and datasets of different characteristics.

The ubiquity of image forgery in the digital age was discussed by Sumaira Bibi et al. (2021) [16], who concentrated on copy-move and splicing as common forms. Although current approaches focused mostly on JPEG images, they argued for solutions that are independent of image formats. They suggested using Stacked Autoencoders (SAE) to detect forgeries using a variety of compression methods. For feature extraction, CNNs with prior training, such as VGG16 and AlexNet, were used. Their method, which used an Ensemble Subspace Discriminant classifier, produced very impressive accuracies, 93.3% for TIFF images, in particular. Time complexity remained a challenge even after the success.

Yohanna Rodriguez-Ortega et al. (2021) [17] proposed recognition algorithms in response to the proliferation of high-quality false images in the media. They took on copy-move forgery detection, pointing out the shortcomings of conventional techniques for classifying large amounts of data. Although they showed promise, deep learning-based methods had trouble with hyperparameter selection and generalization. They examined the influence of their depth and generalization across various datasets to develop two deep learning models—one based on Transfer Learning (TL) and the other custom—in an effort to lessen this. The TL-based VGG-16 quadrupled the inference time but performed 10% better than the modified model. They demonstrated how single-dataset trained models performed poorly when evaluated on a variety of data sets, underscoring the difficulty of generalizing models. The disparity between recall and precision measures, as well as the longer inference time of the VGG-16 model, are among its drawbacks.

A technique for identifying and locating image forgeries using Deep Convolutional Neural Networks (DCNN) and semantic segmentation was presented by Abhishek and Neeru Jindal (2021) [18]. Utilizing color illumination and a transfer

learning methodology, they trained a VGG-16 model with two classes to identify fake or authentic image pixels. Several datasets, including GRIP, DVMM, CMFD, and BSDS300, were used to test the technique. First, a 54-layer DCNN transfer learning network was trained using copy-move, spliced, and fake video images, after which a 27-layer DCNN network was retrained using the original and fabricated images. Ultimately, all fake images were used to train a 91-layer VGG network to distinguish between forged and real pixels.

According to Shilpa Dua et al. (2020) [19], the majority of detection techniques focus on copy-move or splicing forgeries. They developed a unique algorithm that can identify both kinds of forgeries at the same time in order to overcome this restriction. Their method takes advantage of the distortions created by JPEG-encoded image transformations, concentrating on modifications to the statistical characteristics of the AC components of block Discrete Cosine Transform (DCT) coefficients. They developed a feature vector by independently calculating the standard deviation and non-zero DCT coefficients for each AC frequency component. Accurate identification of real and fake photos was made possible by this vector in conjunction with Support Vector Machine (SVM) classification. Tests using the CASIA v1.0 and v2.0 datasets showed higher detection rates than previous approaches.

The issue of image manipulations on online sharing platforms and the difficulties they present for forgery detection algorithms was tackled by Boubacar Diallo et al. (2020) [20]. By taking into account application-specific image quality, their methodology aimed to improve robustness. They highlighted lossy compression—specifically, JPEG—as a popular modification and used a CNN for camera identification. They experimented with different grades of compressed and uncompressed images, and their CNN performed better than earlier methods. They also suggested a thorough examination of CNN's characteristics to improve its accuracy and interpretability.

A technique to identify facial image forgeries was presented by Lingzhi Li et al. (2020) [21]. By using a grayscale representation, this approach was capable of determining whether an input face image could be divided into two separate images by blending them.

Face X-rays have shown efficacy in identifying modifications ubiquitous in face alteration techniques by emphasizing the blending border for falsified images and its absence for authentic ones. When face X-ray was trained on fictitious images beforehand, it remained resilient against different manipulation techniques, in contrast to many detection algorithms that depend on particular artifacts.

State-of-the-art techniques for identifying image forgeries often suffer from slow processing speeds and low accuracy. They typically excel at detecting either image splicing or copy-move forgeries, but not both. To address these drawbacks, an entirely novel image forgery-detecting method has been unveiled in this research. This framework is suitable for real-world applications, as it dramatically improves detection accuracy and response time. Its efficiency ensures usability even on slower devices, expanding its reach to a larger user base. Further details about the suggested framework are provided in the following sections.

3. Materials and Methods

The proposed approach begins with sourcing data from the CASIA V2 dataset, available on the Kaggle repository. This dataset comprises three main folders: authentic, tampered, and ground truth. Subsequently, Compression Error Analysis (CEA) is applied to quantify information loss caused by compression and its impact on image quality. The images in the test set are then converted into CEA format by computing the difference between original and resaved images, followed by adjusting pixel values for optimal visualization. The converted images are categorized based on authenticity (authentic or tampered), with counts displayed for each category. Data augmentation techniques are then employed to enhance the dataset. Three CNN approaches are explored: developing a custom CNN model, implementing transfer learning with MobileNet V2, and fine-tuning hyperparameters for enhanced performance. These strategies aim to identify the most effective method for image tampering detection. Finally, model evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to assess model performance. Figure 2 shows the block schematic of the proposed model.

3.1. Dataset Description

The research employs the CASIA 2.0 dataset [22], which was chosen because of its appropriateness in evaluating many forms of image alteration. Three separate categories that are essential to the goals of the research are included in this dataset. In order to provide a baseline for comparison, it first contains Pristine (Authentic) images, which are original, unedited images. Second, to simulate a frequent type of tampering, the dataset includes Copy-move (Tampered) images, in which some sections have been duplicated inside the same image.

Lastly, there are Spliced (Ground Truth) images, which replicate a common method of image forgery by including altered areas that have been duplicated from several images and pasted into the main image. The CASIA 2.0 dataset's varied assortment of image categories provides researchers with an extensive array of scenarios to assess and improve forgery detection methods. Figure 3 shows the sample images in the dataset.

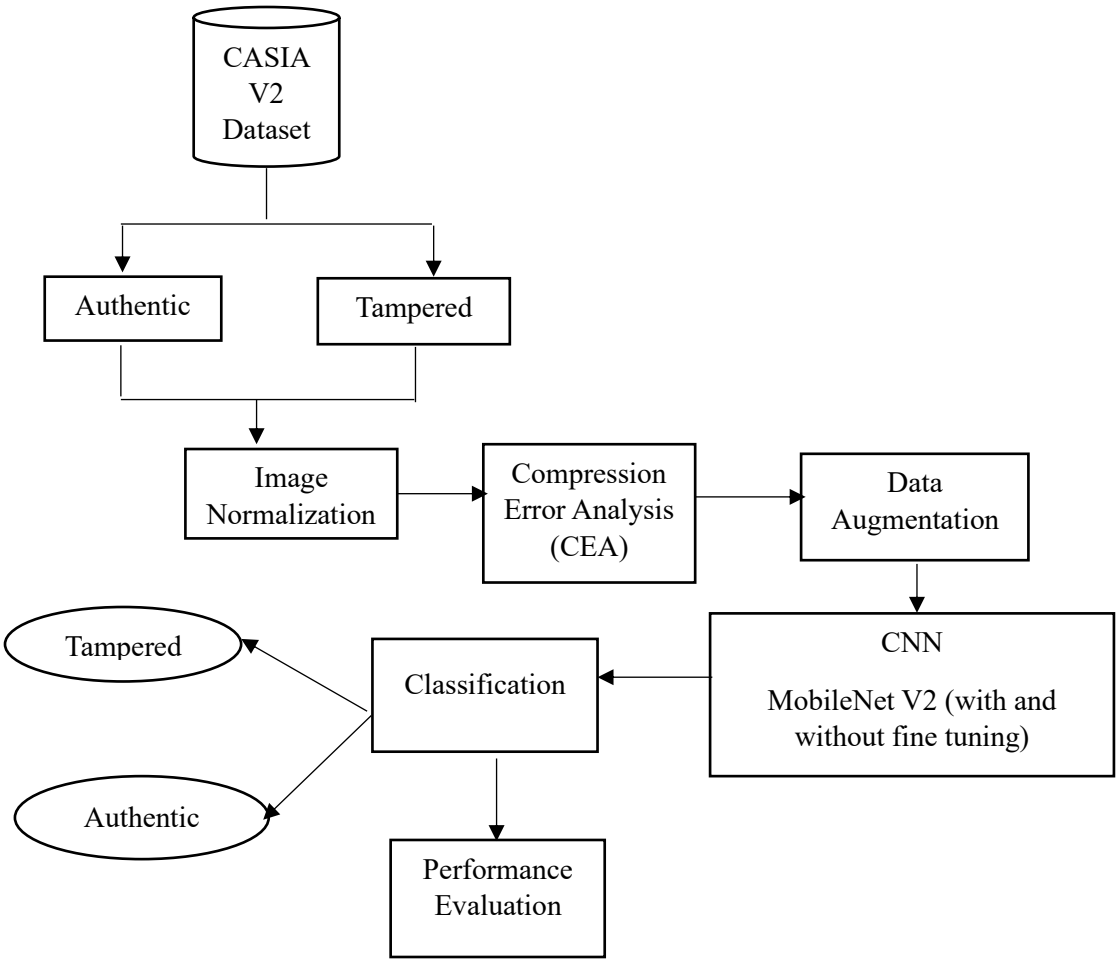


Fig. 2 Block schematic of the proposed model

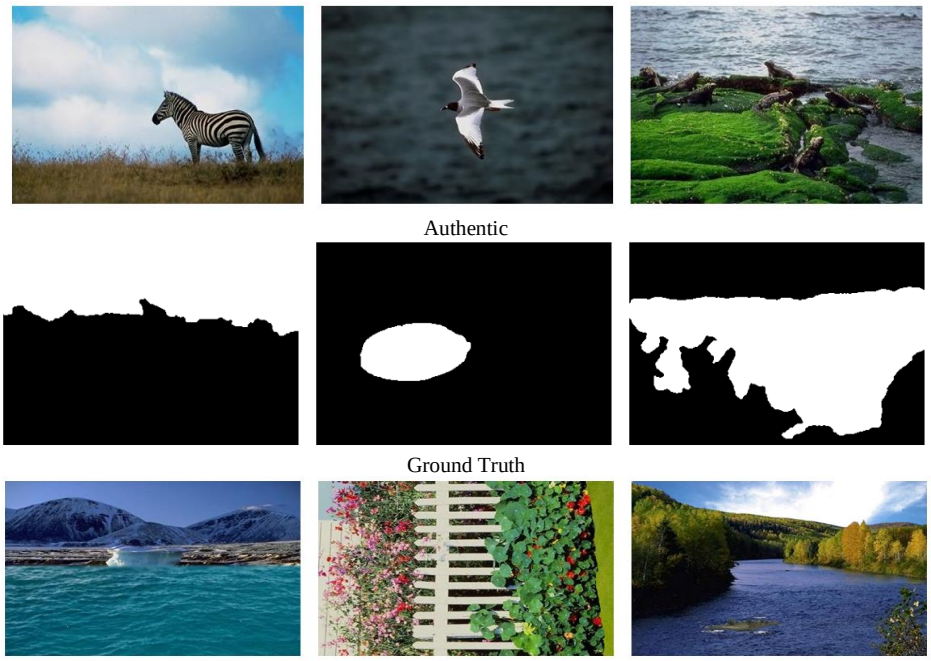


Fig. 3 Sample images in the dataset

Figure 4 illustrates the distribution of images within the dataset across different classes. Specifically, the dataset comprises a total of 1171 images categorized as authentic and 1158 images classified as tampered.

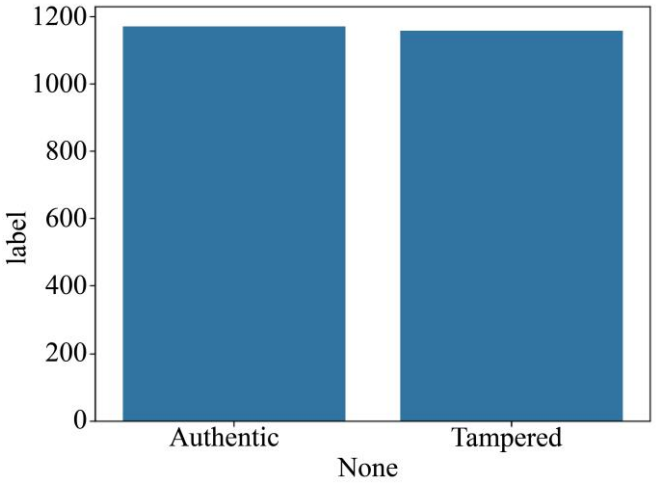


Fig. 4 Image count in the dataset

3.2. Compression Error Analysis (CEA)

An approach for assessing how compression methods affect image quality is CEA. Whether images are compressed using lossy or lossless techniques, a reduced file size results in some information loss. The goal of CEA is to measure and comprehend this information loss and how it impacts the application and visual quality of the compressed images.

CEA entails several key steps in evaluating the effects of compression algorithms on image quality. Initially, the process involves compressing an uncompressed image using a chosen compression algorithm to produce a compressed version. Subsequently, error measurement techniques are employed to compare the compressed image with the original, identifying any discrepancies or introduced errors. Following this, error visualization techniques are applied to visualize the errors, facilitating an understanding of their manifestation in the compressed image relative to the original. Through analysis and interpretation of these error measurements and visualizations, insights are gained into the impact of compression on image quality, including the identification of common compression artifacts and trade-offs between compression ratio and image fidelity. Finally, leveraging these insights, optimization and improvement strategies are devised to refine compression algorithms, adjust parameters, or explore alternative methods to minimize information loss while achieving desirable compression ratios, thereby enhancing overall image quality.

CEA is becoming a widely used technique in image forensics research, especially for identifying image manipulation and tampering [23]. Using this procedure, the error pattern is interpreted by comparing the original image with a modified version of the same image. CEA compares pixels in matching locations from the original and altered images to determine how they differ. Eight by eight blocks are used for the analysis, and two conditions are usually noted for JPEG images:

- If all 8 × 8 blocks display a comparable error pattern, indicating that every block has attained a local minimum, it is possible to identify the original JPEG image.
- If any 8 x 8 block shows a higher error pattern, suggesting that a block has not attained its local minimum, that block is classified as a modified JPEG image.

Equation 1 illustrates the steps involved in the ELA process, which entails resaving an image with a predetermined level of compression quality and comparing the variations among different compression levels.

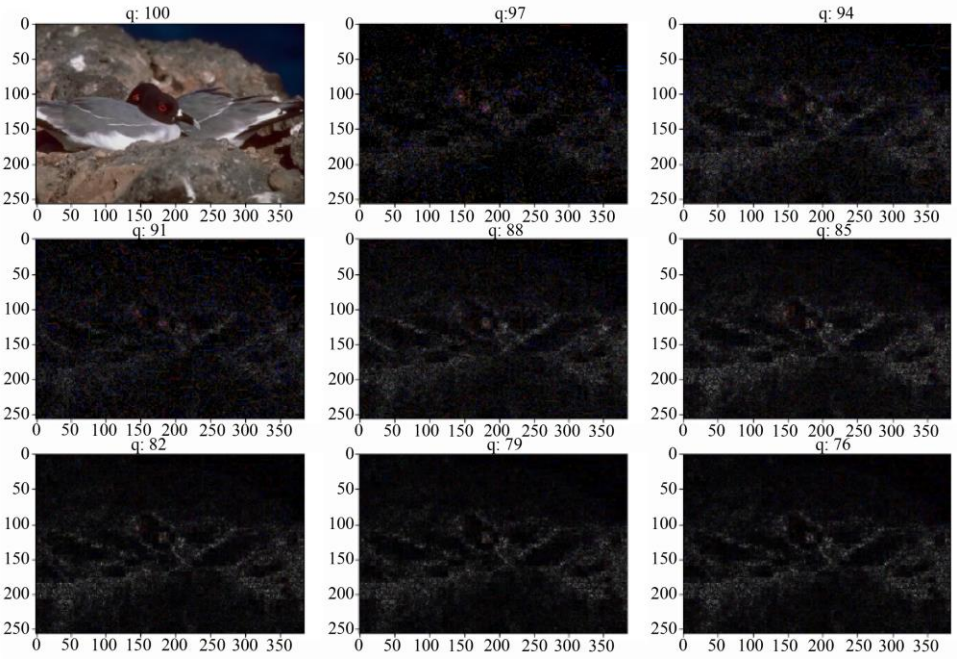
$$\begin{aligned} \overbrace{I_{A0}(i,j) - I_{B1}(i,j)}^{\text{Resavings}}\overbrace{}^{\text{Recompress}} &= CEA_1 \\ I_{A1}(i,j) - I_{B2}(i,j) &= CEA_2 \\ I_{A2}(i,j) - I_{B3}(i,j) &= CEA_3 \\ &\vdots \\ &\vdots \\ I_{An}(i,j) - I_{Bn}(i,j) &= CEA_n \end{aligned} \tag{1}$$

The provided Equation outlines the functioning of CEA, particularly concerning JPEG images. To delve into the calculation of CEA, let us delve into an example. Here, I denotes a JPEG image. A JPEG image that has been resaved n times with a quality setting of 75% is represented by A_n, whereas a JPEG image that has been recompressed n times with a quality setting of 95% is represented by B_n,. To evaluate the error value that would arise from recompressing the JPEG image at a 95% compression quality, the CEA calculation known as "using CEA of 95%" is utilized. The CEA continuously reduces with each additional image resave, represented by the variable n. Therefore, every 8 x 8 block in the JPEG image gradually gets closer to its local minimum as a result of numerous resave, producing a darkening effect.

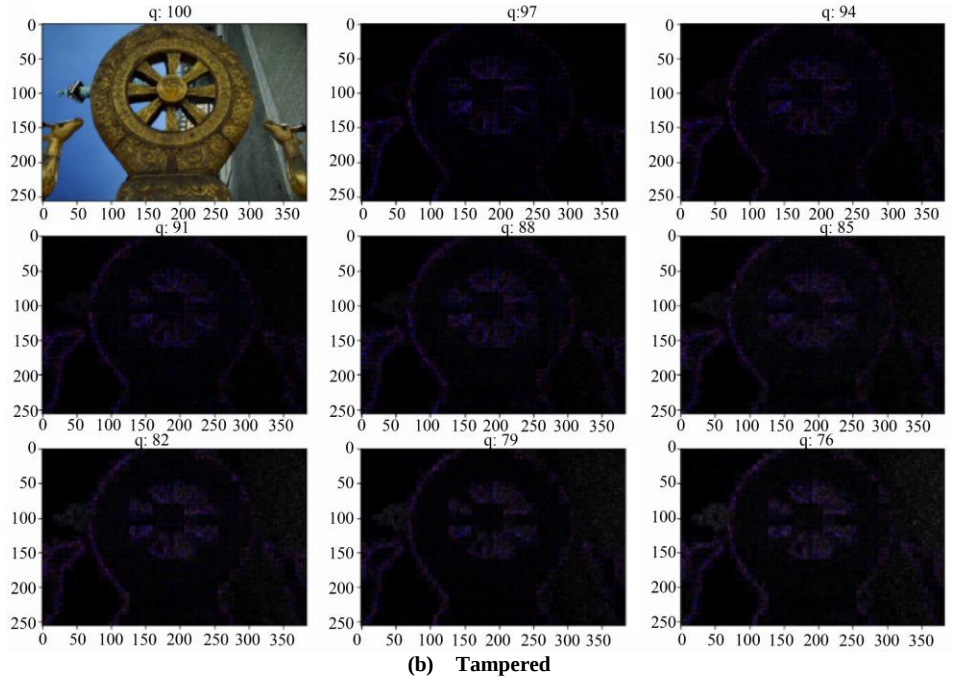
In the proposed research, the input image file undergoes processing along with a specified quality parameter, resulting in the transformation of the image into a CEA representation. Initially, a temporary JPEG file is created with the designated quality level, followed by the computation of differences between the original image and its compressed counterpart, resulting in the generation of the CEA image. Subsequently, this CEA image is subjected to enhancement techniques to effectively highlight discrepancies between the original and compressed images.

Figure 5 demonstrates the CEA representation of the image across different quality levels. It iterates through a spectrum of quality values, creating and presenting the CEA images corresponding to each quality level alongside the original image. This process enables the examination of how the CEA representation changes with varying compression qualities. The images from the test set are then converted into their CEA format, making them suitable for input into the proposed model. This involves generating the CEA representation by calculating the difference between the original and resaved images. Additionally, the pixel values of the CEA image are adjusted to fall within the range [0, 255] for improved visualization.

The converted images are then organized into separate directories based on their authenticity status (authentic or tampered), and the total count of converted images in each category is displayed. This pre-processing step effectively prepares the images for subsequent utilization in either training or testing the model aimed at detecting image tampering. Figure 6 shows the final CEA visualization of the sample images.



(a) Authentic



(b) Tampered

Fig. 5 CEA results for various levels of compression

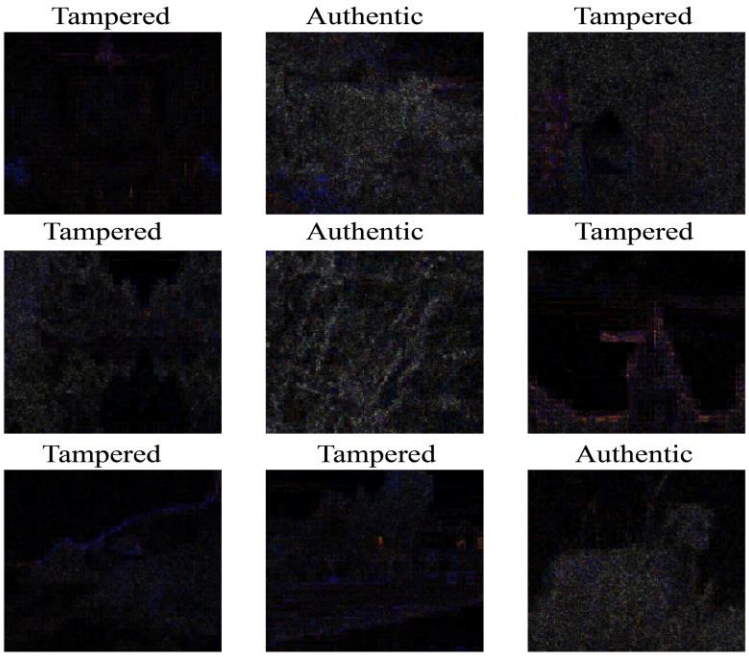


Fig. 6 CEA visualization of the sample images

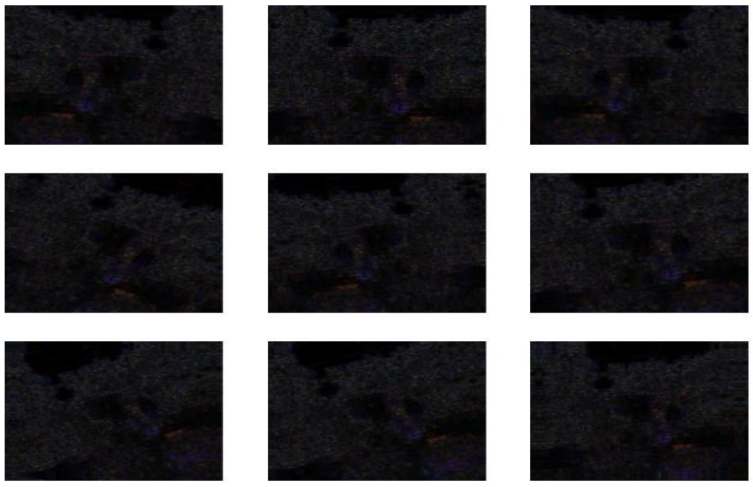


Fig. 7 Data augmented images

3.3. Data Augmentation

Data augmentation, a crucial component of the proposed methodology, involves enhancing the dataset by introducing variations in the images.

This is achieved through techniques such as rotation or skewing, which slightly modify the orientation or perspective of the images. By applying these transformations, the dataset is enriched with diverse instances, thereby enhancing the ability of the model to generalize and recognize patterns efficiently.

Figure 7 showcases a selection of augmented images, demonstrating the effectiveness of these techniques in diversifying the dataset and improving the model's robustness to variations in image characteristics.

3.4. Deep Learning Classifier

Following the augmentation process, the augmented images are subsequently inputted into the DL classifier. For the CNN model, three distinct approaches are explored. Initially, a custom CNN model is constructed and trained on the dataset to assess its performance. Secondly, transfer learning is employed using MobileNet V2, leveraging its pre-trained weights from the ImageNet dataset, and the performance is compared. Lastly, hyperparameter tuning is conducted using the second approach to optimize its performance further. These approaches aim to evaluate different strategies for leveraging deep learning techniques in image classification tasks, with the overarching goal of identifying the most effective approach for detecting image tampering in the dataset.

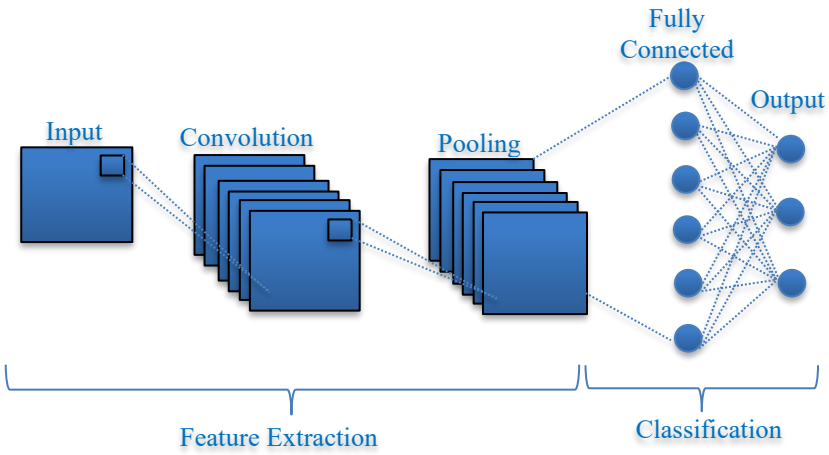


Fig. 8 Basic architecture of CNN

3.4.1. Convolutional Neural Network

A subclass of deep neural networks known as CNNs is designed especially for handling structured, grid-like data, like images [24]. Owing to its capacity to automatically learn hierarchical representations of features straight from raw pixel input, CNNs have become the backbone of various computer vision applications. A CNN's architecture usually comprises multiple layers organized in a particular order, as depicted in Figure 8.

The raw data is given to the input layer; usually, this is an image shown as a grid of pixel values. The convolution layer mostly carries out the process of extracting features from the input data. Convolution maintains a relationship between pixels as they move across the input data by using small squares of input data to learn image features. A mathematical technique known as convolution needs two inputs, such as an image matrix and a learnable filter or kernel. The dot product of the filter's values and the matching area of the input data are calculated at every position. A convolution operation is an elementwise matrix multiplication operation. After this procedure, a feature map is produced that shows the spatial patterns found in the input. The convolutional layer is able to efficiently capture pertinent information like edges, textures, and forms by picking up the values of these filters during training.

After every convolutional operation, activation functions such as ReLU are employed to introduce non-linearity and aid in the learning of intricate patterns. Then, using techniques like max pooling and average pooling, pooling layers minimize the feature maps' spatial dimensions while preserving crucial information. Advanced features gained by convolutional layers are mapped to output classes by fully connected layers, which are positioned towards the end of the network and connect every neuron in one layer to those in the next. Ultimately, the output layer generates the network's output, which, depending on the task at hand, is frequently presented as class probabilities or continuous values.

The feature map produced by a given layer is subjected to an activation function in order to assess the output of that layer. Deep learning models typically employ the ReLU as their activation function. The feature maps' spatial dimensions are decreased, yet the pooling layer retains crucial information. Typical pooling procedures, such as max pooling and average pooling, select the maximum or average value within each pooling window to downsample the feature maps.

A fully connected layer's layout consists of an output layer, a hidden layer, and flattening, which make up the traditional neural network model. The convolutional and pooling layers produce three-dimensional outputs, while a fully connected layer requires a one-dimensional vector as input. Consequently, the output of the pooling layer is flattened into a vector format, which serves as the input to the fully connected layer. Dense layers, or fully connected layers, link each neuron in one layer to every other layer's neuron. Fully connected layers, which are usually found at the end of the network, translate the high-level information that the convolutional layers have learned to the output classes. These layers use the retrieved characteristics to learn how to categorize or predict, which allows them to do tasks like regression or classification. The CNN's last layer generates the network's output, which is typically presented as continuous values for regression tasks (like linear activation) or class probabilities for classification tasks (like softmax activation). The particular function that the CNN is intended to carry out determines the structure of the output layer.

In the proposed model, a Rescaling layer normalizes the pixel values of input images by dividing them by 255, thereby scaling them to the range [0, 1]. The initial convolutional (CONV) layer applies a 2D convolution operation with 16 filters of size 3x3 to the model, followed by a Rectified Linear

Unit (ReLU) activation function, enhancing the output element-wise. Subsequent Max Pooling layers perform 2D max pooling to reduce spatial dimensions while preserving crucial information. Layers 4 to 7 utilize Conv2D and MaxPooling2D blocks with increasing filter sizes (32 and 64), enabling the capture of more intricate features from input images.

A flattened layer then converts the 2D feature maps into a 1D vector, facilitating the transition to fully connected (dense) layers. The final layers consist of two Dense layers, with the first containing 128 neurons employing ReLU activation to capture high-level abstract features from the flattened feature maps. The second and last layer, with 2 neurons, corresponds to image categories and generates raw logits for each class, facilitating probability calculation and predictions. Figure 9 shows the model architecture of the proposed CNN.

3.4.2. MobileNet V2

MobileNetV2 is a convolutional neural network architecture designed for efficient and lightweight deep learning tasks, particularly on mobile and embedded devices. It builds upon the success of its predecessor, MobileNetV1, by introducing several key improvements to enhance performance and efficiency. The core components of MobileNetV2's architecture include inverted residual blocks, depth wise separable convolutions, and linear bottlenecks. Because of the careful design of these blocks, which balance model complexity and computational performance, MobileNetV2 is an excellent choice for situations with limited resources.

Depth wise separable convolutions, which split the conventional convolution operation into two distinct layers—depth wise convolution and pointwise convolution—are one of MobileNetV2's distinguishing characteristics. By factorizing spatial and channel-wise processes, depth wise convolution lowers computational costs by applying a single filter to each input channel independently.

Cross-channel interactions are then made possible by pointwise convolution, which combines the output channels of the depth wise convolution with a 1x1 convolution.

MobileNetV2 adds inverted residual blocks in addition to depth wise separable convolutions to capture richer feature representations with less computational overhead, as shown in Figure 10. A lightweight expansion layer, a depth wise convolution, and a linear projection layer constitute these blocks.

Fig. 9 Model architecture of CNN

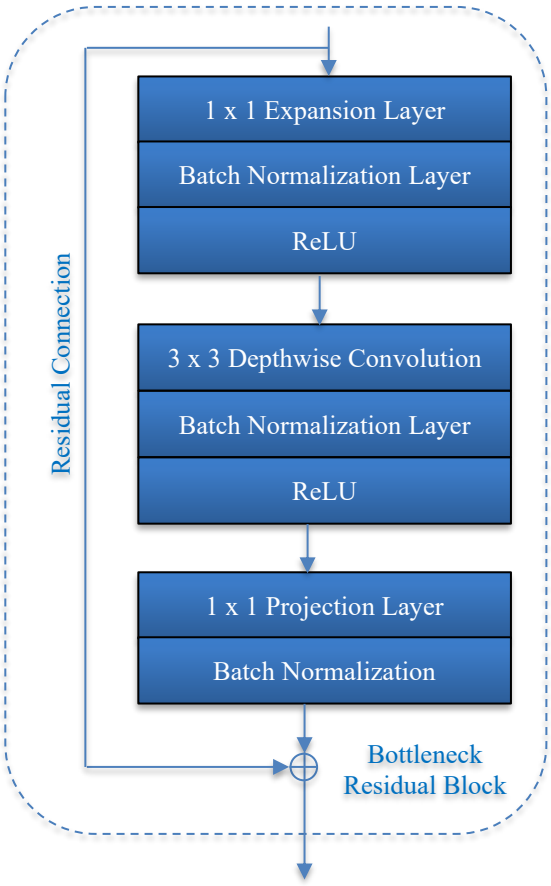


Fig. 10 Inverted residual block on MobileNet V2

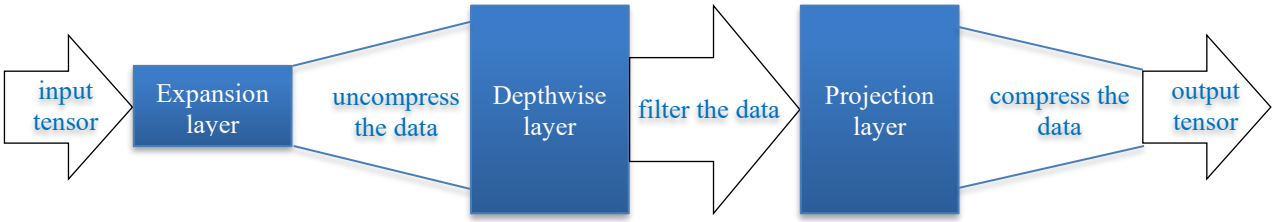


Fig. 11 MobileNetV2 expansion-filtering-compression system

By extending the input channels into a higher-dimensional space, the expansion layer enables the depth wise convolution that follows to extract more intricate features, as shown in Figure 11. After that, the linear projection layer compresses the enlarged features to their original dimensionality while keeping all relevant data.

The low-dimensional data flowing between the blocks serves as a compressed representation of the original data. Before filters are applied to this data, it needs to be uncompressed. The expansion layer functions as a decompressor, restoring the data to its full form. Subsequently, the depth wise layer conducts the necessary filtering operations at this stage of the network. Finally, the projection layer compresses the data once more to reduce its dimensionality.

Moreover, MobileNetV2 uses linear bottlenecks to lower the intermediate feature map computing cost. MobileNetV2 guarantees that the number of channels stays constant across the network by implementing a 1x1 convolution with a linear activation function after the depth wise convolution, avoiding excessive computational overhead.

Transfer learning is a machine learning technique where a model trained on one task is adapted to a related task by leveraging the knowledge gained during the initial training. Instead of training a model from scratch, transfer learning involves using pre-trained models as a starting point and fine-

tuning them on the new task. This approach is particularly useful when the new task has limited labeled data or computational resources. In the proposed study, the pre-trained MobileNetV2 model is loaded with weights trained on ImageNet, excluding the final fully connected layers, to prepare it for training on the dataset. Features are then extracted from this model, serving as input for our final layers.

The pre-trained MobileNet V2 model has already learned to extract useful features from images, which can be beneficial for detecting forged or tampered images. By fine-tuning the MobileNet V2 model on a dataset specific to image forgery detection, the model can adapt its learned features to better discriminate between authentic and manipulated images. To preserve the knowledge captured by the original model, the layers of the base model are frozen. Additionally, a layer is added to preprocess the input, performing necessary transformations such as normalization and reshaping to align with the expected input shape of MobileNetV2. A Global Average Pooling layer is introduced to reduce spatial dimensions and compress the extracted features. Subsequently, a dense layer with a single neuron is appended to produce predictions based on the processed features. Finally, all layers are combined to form the proposed detection model.

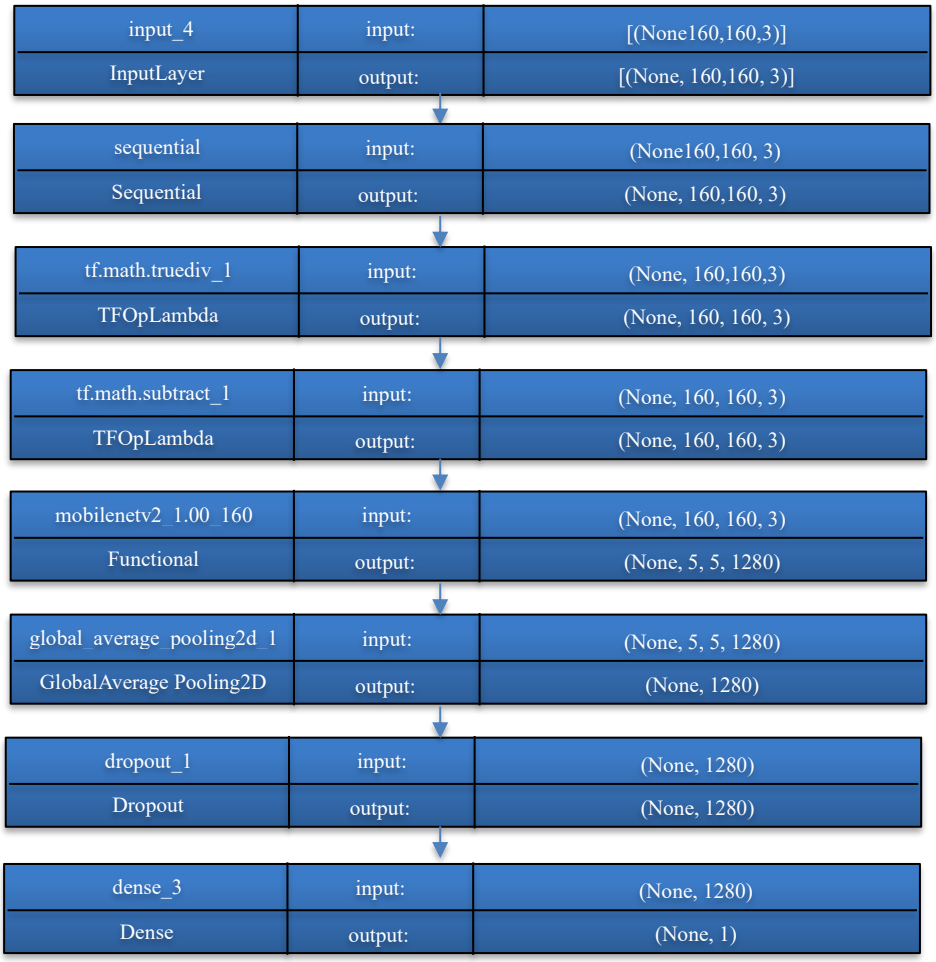


Fig. 12 Model architecture of the proposed MobileNet V2 model

To enhance the model's performance, a fine-tuning approach is employed by training specific layers of the base model. Among the 154 layers, the last 100 layers are selected for training, while the remaining layers are kept frozen. Following this selection, the model is compiled using the same hyperparameters and subjected to an additional 40 epochs of training to further refine its performance. The model architecture of the proposed MobileNet V2 model is shown in Figure 12.

4. Results and Discussion

4.1. Hardware and Software Setup

The research utilized a high-performance computational setup comprising an Intel Core i7 processor, 32GB of RAM, and the powerful NVIDIA GeForce GTX 1080Ti GPU. Model implementation was conducted through the Keras library, which operates as a prototype built upon the TensorFlow framework and executed using Python. Renowned for its user-

friendly interface and robust capabilities, Keras played a pivotal role in designing complex neural network architectures. This framework ensures efficient resource utilization across CPU, GPU, and TPU environments. To harness extensive computational power and streamline model training, the deployment was orchestrated on Google Colab, a cloud-based Python notebook environment.

Hyperparameters play a crucial role in determining how a machine learning framework behaves during the training process. Unlike model parameters, which are learned from the data, hyperparameters are predefined by the user prior to training. These hyperparameter selections, including the optimizer, learning rate, loss function, and number of epochs, collectively define the training configuration aimed at optimizing the model's performance for the proposed image forgery detection task. The specific model configuration is detailed in Table 1.

Table 1. Model configurations

Hyperparameters	CNN	MobileNet V2
Learning rate	-	0.0001
Optimizer	ADAM	ADAM
Loss Function	Sparse Categorical Cross entropy	Binary cross entropy
Epoch	15	Initial -25 Fine tune-30
Activation function	ReLU	ReLU

4.2. Experimental Results

The accuracy and loss plot for the proposed study illustrates the performance of the model during the training and validation phases. The accuracy plot displays the percentage of correctly classified samples over each epoch, indicating how well the model is learning to classify authentic and manipulated images.

A rising accuracy curve suggests that the model is improving its ability to make accurate predictions as training progresses. Conversely, the loss plot illustrates the error between the predicted and actual labels for each batch of data. A decreasing loss curve indicates that the model is minimizing its error, meaning it is becoming more proficient at classifying images correctly.

Figure 13 shows the accuracy and loss plot for the CNN model. The model is trained for 15 epochs. Initially, during the first epoch, the model achieves a training accuracy of approximately 86.75% and a validation accuracy of around 92.47%, with corresponding training and validation losses of approximately 0.3265 and 0.2711, respectively.

As training progresses, both the training and validation accuracies consistently improve, reaching approximately 93.98% accuracy by the final epoch.

Fig. 13 Accuracy and loss plot of CNN

However, the validation loss fluctuates slightly throughout training, peaking at around 0.3556 during the fourth epoch before decreasing and stabilizing around 0.3039 by the fifteenth epoch.

This pattern indicates that while the model steadily improves its ability to classify images correctly, there may still be some overfitting occurring, as evidenced by the slightly higher validation loss compared to the training loss.

As the low accuracy of the CNN model indicates poor performance, the approach of transfer learning using MobileNet V2 is adopted, with the model trained over an initial period of 25 epochs.

The accuracy and loss plot of the MobileNet V2 before fine-tuning is shown in Figure 14. At the beginning of training, during the first epoch, the model achieves a training accuracy of approximately 53.33% and a validation accuracy of around 52.69%, with corresponding training and validation losses of approximately 0.7301 and 0.6895, respectively.

As training progresses, both the training and validation accuracies steadily improve, reaching approximately 83.91% for training and 77.85% for validation accuracy by the final epoch. Similarly, the training and validation losses decrease over the epochs, stabilizing around 0.3633 for training loss and 0.3762 for validation loss by the end of training.

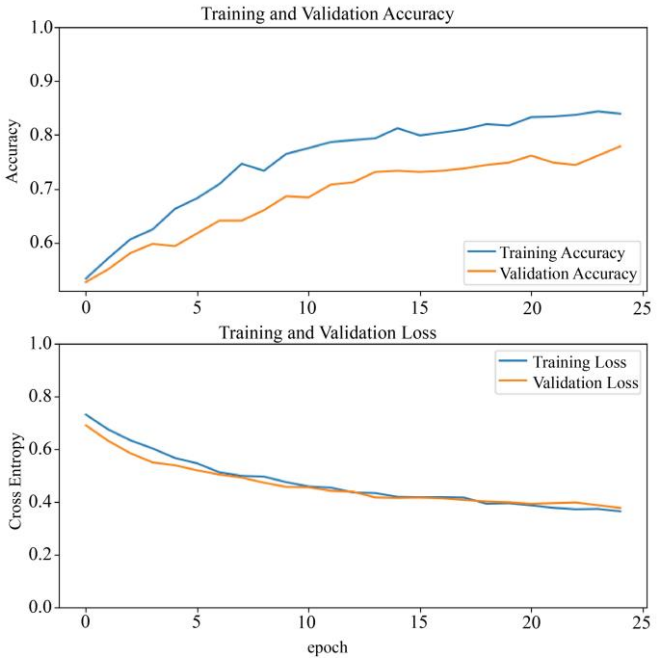


Fig. 14 Accuracy and loss plot of MobileNet V2 before fine tuning

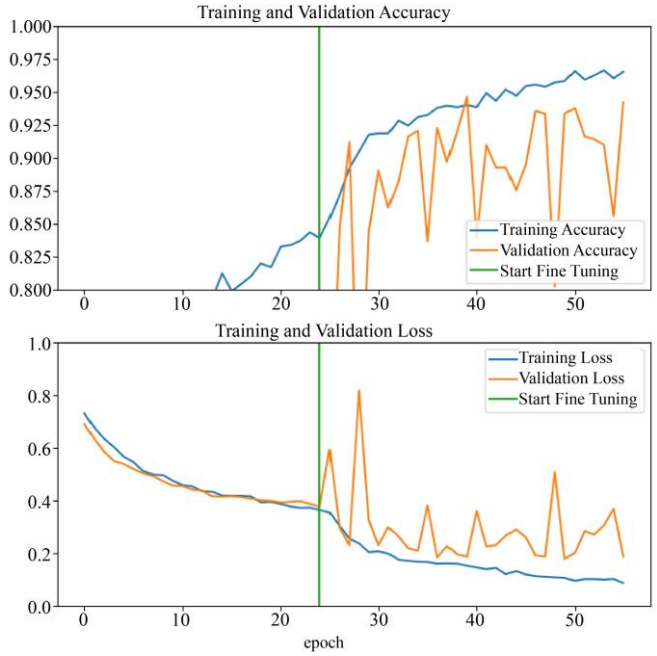


Fig. 15 Accuracy and loss plot of MobileNet V2 after fine tuning

The visualization indicates a consistent decrease in both training and validation loss, suggesting that further fine-tuning of the model and training for additional epochs is warranted. To refine the model, specific layers of the base model are trained: the last 100 out of 154 layers are adjusted while the remaining layers remain frozen. The model is then compiled with the same hyperparameters and trained for an additional 40 epochs. Fine-tuning of MobileNet V2 is initiated from epoch 30 onwards, with a total of 55 epochs. Initially, the accuracy and loss metrics indicate a substantial improvement in model performance, as shown in Figure 15. For instance, at epoch 30, the training accuracy stands at approximately 91.85%, while the validation accuracy reaches around 89.03%. However, as the training progresses, both the training and validation accuracies steadily increase, reaching a peak accuracy of about 96.51% and 94.19%, respectively, by epoch 55. Similarly, the loss metrics show a decreasing trend over epochs, indicating a consistent improvement in the model's ability to minimize prediction errors. The final validation loss is notably reduced to 0.1865, reflecting the enhanced performance of the fine-tuned MobileNet V2 model in detecting image forgery. Overall, the iterative fine-tuning process effectively refines the model's capability, resulting in significant enhancements in accuracy and reductions in loss, thereby demonstrating its effectiveness in image forgery detection tasks.

In order to thoroughly evaluate the efficacy and operational efficiency of the proposed model for image forgery detection, the F1-score, accuracy, precision, and recall are the four primary metrics utilized. These measures, which are based on the concepts of False Positive (FP), False Negative (FN), True Negative (TN), and True Positive (TP), are essential for assessing the model's performance. These performance parameters have mathematical formulations that are shown in Equations (2), (3), (4), and (5).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1-score = 2 \times \frac{precision \times Recall}{Precision+Recall} \tag{5}$$

Table 2 shows the classification report of the proposed methodology. The accuracy achieved is 94.19%, indicating the model's ability to correctly classify authentic and tampered images. With a precision of 92%, the model shows a high level of correctness in identifying tampered images out of all detected positive cases. Moreover, the recall score of 95% highlights the model's capability to effectively identify the majority of tampered images present in the dataset. These metrics collectively suggest that the proposed model, after fine-tuning MobileNet V2, exhibits strong capabilities in detecting image forgery with high accuracy and reliability.

Table 2. Classification of the proposed model

Evaluation Metrics	Result Obtained (%)
Accuracy	94.19
Precision	92
Recall	95
F1-score	93

A performance measuring tool used in classification tasks to assess how well a model predicts the future state of the data is the confusion matrix. Through the comparison of the actual and projected classes for a particular dataset, it offers an overview of the model's performance. As shown in Figure 16, the confusion matrix indicates that out of 30 images, 11 tampered images were correctly classified as tampered, while 19 authentic images were correctly classified as authentic. However, there were 2 instances where authentic images were misclassified as tampered, and no tampered images were mistakenly classified as authentic. Overall, the confusion matrix suggests that the model exhibits strong performance in correctly identifying both tampered and authentic images, with only a few misclassifications observed.

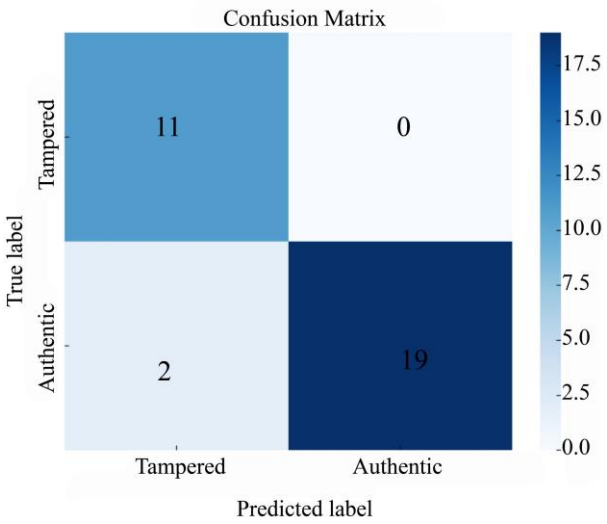


Fig. 16 Confusion matrix

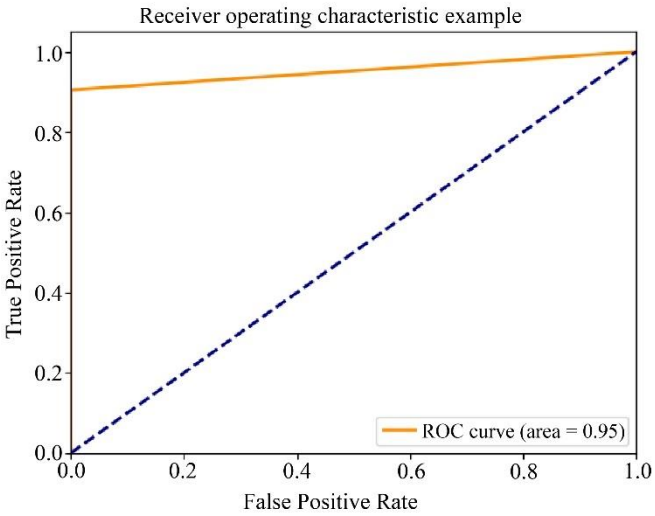


Fig. 17 ROC curve of the proposed model

Receiver Operating Characteristic (ROC) curves are utilized to assess the model's capability to differentiate between tampered images and authentic ones in the proposed image forgery detection task. The area under the ROC curve, a measure of the model's discriminative ability, is determined to be 95%, as shown in Figure 17.

Figure 18 displays an image from the batch along with its predicted class label and the corresponding actual label. This visualization helps in assessing how well the model is performing on the test data.

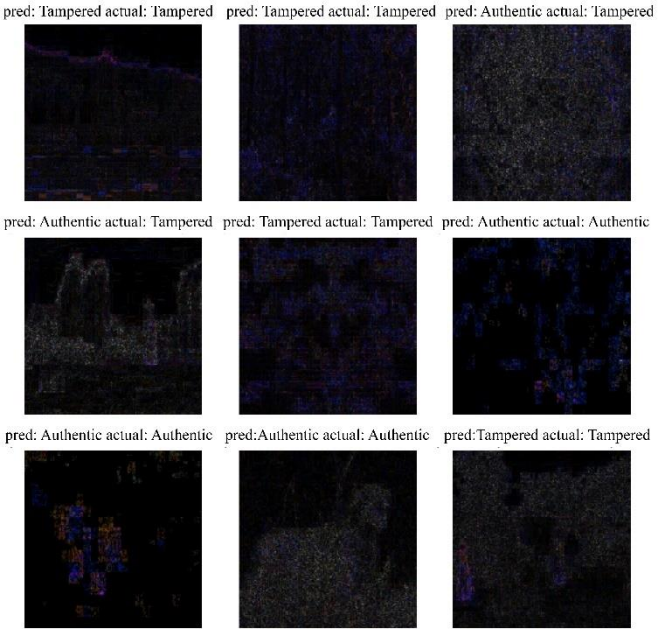


Fig. 18 Prediction output

The proposed model combines image preprocessing and model prediction to analyze images and generate binary classification results, distinguishing between authentic and tampered images. The effectiveness of the framework in discerning between authentic and tampered images is visually depicted in Figure 19.

```
CPU times: user 60.3 ms, sys: 9.08 ms, total: 69.4 ms
Wall time: 7.96 s
'This picture is Tampered'
```

```
show_img_file('/content/gdrive/My Drive/CASIA_2/Tampered/Tp_D_CND_S_N_txt00028_txt00006_10848.jpg')
```

Fig. 19 Prediction of the model from random images

Table 3. Comparison of the proposed approach with existing models

Author (Year)	Methodology	Accuracy (%)
Yohanna Rodriguez-Ortega et al. (2021) [17]	VGG-16	78
Wina Permana Sari and Hisyam Fahmi (2021) [26]	CNN	85.89
Ying Zhang et al. (2016) [25]	Stacked Autoencoder	87.51
Amit Doegar et al. (2020) [27]	Google Net and Random Forest	89.55
Proposed Methodology: CNN		**93.97**
MobileNet V2		**94.19**

5. Conclusion

The proposed research highlights the effectiveness of deep learning, particularly CNNs, in detecting image forgery. Through the exploration of various methodologies, including custom CNN models and transfer learning with MobileNet V2, significant insights have been gained into the application of deep learning techniques in this domain. The results demonstrate that fine-tuning the MobileNet V2 model at epoch 25 substantially enhances its accuracy, underscoring the importance of model optimization for improved performance. By achieving a high level of accuracy in distinguishing between authentic and tampered images, the proposed approach showcases the potential of deep learning to address the challenges posed by image manipulation. The evaluation metrics, including accuracy, precision, recall, and F1-score, provide comprehensive assessments of the models' performance, affirming their effectiveness in image forgery detection tasks. Furthermore, the study contributes to advancing the field of digital forensics by providing practical solutions for identifying manipulated images. The findings not only offer insights into the capabilities of deep learning techniques but also highlight the need for continued research and development in this area.

Acknowledgments

The author expresses profound appreciation to the supervisor for providing guidance and unwavering support throughout this study.

References

[1] Bin Xiao et al., "Image Splicing Forgery Detection Combining Coarse to Refined Convolutional Neural Network and Adaptive Clustering," *Information Sciences*, vol. 511, pp. 172-191, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan, "ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries with Anomalous Features," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 9535-9544, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Kalyani Dhananjay Kadam, Swati Ahirrao, and Ketan Kotecha, "Multiple Image Splicing Dataset (MISD): A Dataset for Multiple Splicing," *Data*, vol. 6, no. 10, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mohamed A. Elaskily et al., "Deep Learning Based Algorithm (ConvLSTM) for Copy Move Forgery Detection," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 3, pp. 4385-4405, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Beste Ustubioglu et al., "A New Copy Move Forgery Detection Technique with Automatic Threshold Determination," *AEU - International Journal of Electronics and Communications*, vol. 70, no. 8, pp. 1076-1087, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Babak Mahdian, and Stanislav Saic, "A Bibliography on Blind Methods for Identifying Image Forgery," *Signal Processing: Image Communication*, vol. 25, no. 6, pp. 389-399, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[7] Hany Farid, "Image Forgery Detection," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16-25, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[8] Tran Van Lanh et al., "A Survey on Digital Camera Image Forensic Methods," *2007 IEEE International Conference on Multimedia and Expo*, Beijing, China, pp. 16-19, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[9] Prajakta Kubal, Vanita Mane, and Namita Pulgam, "Image Manipulation Detection Using Error Level Analysis and Deep Learning," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4, pp. 91-99, 2023. [Google Scholar] [Publisher Link]

[10] Shobhit Tyagi, and Divakar Yadav, "A Detailed Analysis of Image and Video Forgery Detection Techniques," *The Visual Computer*, vol. 39, pp. 813-833, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Syed Sadaf Ali et al., "Image Forgery Detection Using Deep Learning by Recompressing Images," *Electronics*, vol. 11, no. 3, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Emad Ul Haq Qazi, Tanveer Zia, and Abdulrazaq Almorjan, "Deep Learning-Based Digital Image Forgery Detection System," *Applied Sciences*, vol. 12, no. 6, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Davide Alessandro Coccomini et al., "Cross-Forgery Analysis of Vision Transformers and CNNs for Deepfake Image Detection," *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*, Newark NJ USA, pp. 52-58, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Haiwei Wu et al., "Robust Image Forgery Detection over Online Social Network Shared Images," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 13430-13439, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Amit Doegar et al., "Image Forgery Detection Based on Fusion of Lightweight Deep Learning Models," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 4, pp. 1978-1993, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sumaira Bibi et al., "Digital Image Forgery Detection Using Deep Autoencoder and CNN Features," *Human-Centric Computing and Information Sciences*, vol. 11, no. 32, pp. 1-17, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yohanna Rodriguez-Ortega, Dora M. Ballesteros, and Diego Renza, "Copy-Move Forgery Detection (CMFD) Using Deep Learning for Image and Video Forensics," *Journal of Imaging*, vol. 7, no. 3, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Abhishek, and Neeru Jindal, "Copy Move and Splicing Forgery Detection Using Deep Convolution Neural Network, and Semantic Segmentation," *Multimedia Tools and Applications*, vol. 80, pp. 3571-3599, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Shilpa Dua, Jyotsna Singh, and Harish Parthasarathy, "Image Forgery Detection Based on Statistical Features of Block DCT Coefficients," *Procedia Computer Science*, vol. 171, pp. 369-378, 2020. [CrossRef] [Google Scholar] [Publisher Links]

[20] Boubacar Diallo et al., "Robust Forgery Detection for Compressed Images Using CNN Supervision," *Forensic Science International: Reports*, vol. 2, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Lingzhi Li et al., "Face X-Ray for More General Face Forgery Detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 5000-5009, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Divyansh Goel, CASIA 2.0 Image Tampering Detection Dataset, Kaggle, [Online]. Available: https://www.kaggle.com/datasets/divg07/casia-20-image-tampering-detection-dataset

[23] Neal Krawetz, "A Picture's Worth... Digital Image Analysis and Forensics Version 2," *Hacker Factor Solutions*, pp. 1-43, 2007. [Google Scholar] [Publisher Link]

[24] Rikiya Yamashita et al., "Convolutional Neural Networks: An Overview and Application in Radiology," *Insights into Imaging*, vol. 9, pp. 611-629, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ying Zhang et al., "Image Region Forgery Detection: A Deep Learning Approach," *Proceedings of the Singapore Cyber-Security Conference, Cryptology and Information Security Series*, vol. 14, pp. 1-11, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[26] Wina Permana Sari, and Hisyam Fahmi, "The Effect of Error Level Analysis on The Image Forgery Detection Using Deep Learning," *KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 6, no. 3, pp. 187-194, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Amit Doegar, Maitreyee Dutta, and Gaurav Kumar, "Image Forgery Detection Using Google Net and Random Forest Machine Learning Algorithm," *Journal of University of Shanghai for Science and Technology*, vol. 22, no. 12, pp. 1271-1278, 2020. [CrossRef] [Google Scholar] [Publisher Link]