International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

Lightweight MobileNet Model for Image Tempering Detection

Sajeeda Shikalgar[1], Dr. Rakesh K. Yadav[2], Dr. Parikshit N. Mahalle[3]
[1]Research Scholar, Shri Venkateshwara University, Amroha, Uttar Pradesh, India.
[2]Director Academics, Shri Venkateshwara University, Amroha, Uttar Pradesh, India.
[3]Professor and Head, Department of Artificial Intelligence and Data Science, Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.
sajeeda.dsr@gmali.com[1], dir.academics@svu.edu.in[2], aalborg.pnm@gmail.com[3]

Abstract. In recent years, there has been a wide range of image manipulation identification challenges and an overview of image tampering detection and the relevance of applying deep learning models such as CNN and MobileNet for this purpose. The discussion then delves into the construction and setup of these models, which includes a block diagram as well as mathematical calculations for each layer. A literature study on Image tampering detection is also included in the discussion, comparing and contrasting various articles and their methodologies. The study then moves on to training and assessment datasets, such as the CASIA v2 dataset, and performance indicators like as accuracy and loss. Lastly, the performance characteristics of the MobileNet and CNN designs are compared. This work focuses on Image tampering detection using convolutional neural networks (CNNs) and the MobileNet architecture. We reviewed the MobileNet architecture's setup and block diagram, as well as its application to Image tampering detection. We also looked at significant literature on Image manipulation detection, such as major studies and their methodologies. Using the CASIA v2 dataset, we evaluated the performance of MobileNet and CNN architectures in terms of accuracy and loss. This paper offered an overview of the usage of deep learning and CNN architectures for image tampering detection and proved their accuracy in detecting manipulated images.

Keywords: Image Tampering Detection, MobileNet, CNN, CASIA v2, Deep Learning, Lightweight Architecture, Convolutional Neural Networks, Performance Evaluation.

I. Introduction

Image tampering detection is a field of digital image forensics that deals with identifying any form of intentional alteration or change to an image in order to confuse or mislead viewers [1]. With the ease with which digital image manipulation is developing, it is more important than ever to develop algorithms and techniques that can detect whether or not a image has been tampered with automatically. Image tampering can entail both manipulating the image's content [2], such as changing the goods or people in it, and altering the image's appearance, such as changing the lighting or colour balance. The purpose of image tampering detection is to create methods for distinguishing between authentic and modified photographs. Image tampering detection is a major research area [3]because it has a wide range of practical applications, including criminal investigations, surveillance, and media forensics. To identify tampering in a photograph, researchers in this field have developed a variety of tools and methodologies. Some prominent methods for detecting image tampering include analysing inconsistencies and irregularities in the image [4], such as differences in lighting or shadows, or changes in texture or colour distribution. Other methods include analysing the image information or compression artefacts to see whether an image has been tampered with. Machine learning and deep learning algorithms for detecting photo tampering [5][6] have shown promise since they can learn to recognise patterns and traits in photographs that signal modification. Normally, these strategies need training a model on a large dataset of both genuine and tampered images, followed by utilising the model to identify tampering in new photographs. The detection of image tampering has grown increasingly important with the widespread use of digital photos in fields [6] such as journalism, forensics, and social media. Images that have been altered can be used to spread false information, deceive people, or even cause harm. Splicing, cloning, object removal or insertion, and colour, texture, or lighting changes are all examples of image manipulation [7]. Since modern image manipulation technology may generate convincing forgeries that are difficult to distinguish from authentic images, detecting these adjustments can be challenging [8].

As a result, researchers have developed a number of approaches for detecting image manipulation[9], which include both classical and machine learning-based techniques. The ultimate goal is to provide effective and quick ways for detecting image manipulation and helping in the maintenance of digital image integrity and authenticity

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

[10]. Overall, image tampering detection is a significant academic area with a variety of practical applications, and there are several approaches and procedures for detecting tampering in digital images.

II. Literature Review

Shukla et al. (2018), which presents an overview of several strategies for image tampering detection, including methods based on statistical analysis, digital watermarking, and deep learning.

This paper Zhou et al. (2018) , reviews various deep learning approaches for image tampering detection, such as CNNs, RNNs and GANs.

This work Amerini et alpaper (2017), focuses on the detection of copy-move forgeries, which include copying and pasting parts of an image to generate a replica. The authors discuss different strategies for detecting this form of forgeries, such as block matching, keypoint matching, and deep learning.

Bayar and Stamm's (2016), describes a CNN-based approach for identifying image forgeries such as copy-move and splicing. The approach entails first training a CNN to differentiate between legitimate and fabricated image patches, and then utilising the trained network to categorise patches of an input image.

Wu et al. (2018) offer a technique for image alteration detection that employs a CNN to learn rich characteristics that are more resilient to different forms of manipulations. In numerous benchmark datasets, the authors show that their strategy outperforms earlier approaches.

Huh et al. (2018), a method for identifying image forgeries that incorporates various deep learning models, including a CNN for detecting splicing and a GAN for detecting inpainting. In numerous benchmark datasets, the authors demonstrate that their strategy outperforms earlier approaches.

Cozzolino et al. (2018), present a GAN-based technique for identifying image forgeries that learns a lexicon of legitimate image patches. The approach entails training a generator network to generate authentic-looking image patches, followed by utilising the learnt lexicon to identify forgeries in an input image.

This research Nguyen et al. (2019), offers a solution for copy-move forgery detection and localization based on Siamese network architecture utilising a deep learning methodology. The authors show how their system can detect and pinpoint copy-move forgeries in photos.

This research Singh and Kumar (2018) , examines several strategies for identifying image forgeries that rely on photo-response non-uniformity (PRNU), a distinctive fingerprint found in digital camera sensors. The authors address the advantages and disadvantages of several PRNU-based approaches and their applications in digital forensics.

Zhou et al. (2020), describe a method for detecting image tampering that combines multi-scale features with deep learning. Using many benchmark datasets, the authors show that their method outperforms earlier approaches and that multi-scale features may successfully capture both global and local information in an image.

This study Das and Naskar (2020), presents a complete evaluation of content-based image tampering detection strategies, including algorithms that employ edge detection, texture analysis, and colour characteristics. The authors explore the advantages and disadvantages of various content-based techniques, as well as the difficulties in establishing strong and reliable detection algorithms.

This study Marra et al. (2018), offers ForgeryNet, a multi-task deep fake detection network that can identify many forms of forgeries at the same time, including copy-move, splicing, and deep fakes. The authors show that their method beats earlier approaches on numerous benchmark datasets, and that integrating different tasks can increase the network's overall performance.

Alghazzawi and Al-Nuaimy (2018) provide, a technique for detecting image forgery that combines convolutional neural networks (CNNs) with local binary patterns (LBPs). The authors show that merging CNNs with LBPs may successfully capture both global and local information in a image, outperforming earlier techniques on numerous benchmark datasets.

Sharma and Singh (2018) provide, a technique for detecting copy-move forgery using a convolutional neural network (CNN) methodology. In many benchmark datasets, the authors show that their method outperforms earlier approaches, and that CNNs can successfully learn discriminative features for detecting copy-move forgeries.

Singh and Agarwal's (2019) paper gives a thorough investigation of deep learning-based approaches for image tampering detection, including methods that employ CNNs, Siamese networks, and generative adversarial networks (GANs). The authors compare the performance of various approaches on a variety of benchmark datasets and demonstrate that deep learning-based methods may achieve high accuracy and robustness to various sorts of manipulations.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

Farid's (2009) work gives an overview of numerous image forgery detection approaches, including methods that involve statistical analysis, watermarking, and digital signatures. The author analyses the advantages and disadvantages of various approaches, as well as the difficulties in building efficient forgery detection tools.

These related work emphasize the wide diversity of methodologies and techniques established for image tampering and forgery detection, as well as the rising interest in employing deep learning-based algorithms to increase detection system accuracy and resilience. Although much work has to be done in this topic, these studies give useful insights and ideas for future study.

III. Recent Image tampering detection techniques

Image tampering detection is an active area of research with many proposed techniques and methods. Here is a brief literature survey on some of the commonly used methods:

1. **Deep learning-based methods:** CNNs have been frequently utilised to identify image alteration. [17] proposes a CNN-based technique that use a residual network architecture to detect manipulated areas in photos. To detect manipulated areas in movies, [2] proposes combining a CNN with an LSTM network.
2. **Passive approaches:** To identify tampering, passive techniques study the statistical features of images. In [18] for example, the authors offer a method for detecting tampering in images by evaluating the DCT coefficients.
3. **Active approaches:** Active techniques augment the image with extra information to identify manipulation. In [19] for example, the authors suggest a method for embedding a watermark into an image and detecting tampering by detecting changes in the watermark.
4. **Feature-based approaches:** To identify tampering, feature-based techniques extract certain aspects from a image. For example, in [5,] the authors offer a technique that uses the Gabor filter to extract features from a image and identifies tampering by comparing these features to the original image.
5. **Compression-based approaches:** Compression-based techniques identify tampering by analysing the compressed form of an image. For example, in [20] the authors offer a method for detecting tampering by comparing the compressed bitstreams of the original and altered photos.
6. **Steganalysis methods:** Steganalysis techniques are used to identify hidden messages in images. These methods can also be used to identify image manipulation. For example, in [21], the authors suggest a technique for detecting image manipulation that makes use of the Rich Model notion.
7. **Frequency domain-based approaches:** Frequency domain-based techniques examine an image's frequency domain to identify manipulation. For example, in [22] the authors offer a method for detecting tampering by assessing changes in the amplitude and phase of an image's Fourier Transform.
8. **Noise-based sensor pattern techniques:** Sensor pattern noise-based approaches identify tampering by utilising a camera's unique sensor noise pattern. For example, in [23], the authors describe a technique for detecting image manipulation that makes use of the sensor noise pattern.
9. **Hybrid approaches:** To identify tampering, hybrid techniques integrate numerous techniques. For example, in [24], the authors suggest a method for detecting image manipulation that combines passive and deep learning-based approaches.
10. **Rule-based approaches:** To identify tampering, rule-based techniques employ a set of specified rules. For example, in [25], the authors offer a method for detecting tampering by examining an image's local geometric features.

There are many techniques and methods proposed for image tampering detection. As the field of image forensics evolves, new techniques and methods are being developed to detect tampering with increased accuracy and robustness.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

Method	Technique	Dataset	Accuracy	FP Rate	FN Rate
DWT-SVM [27]	DWT and SVM	CASIA v2.0	92.13%	0.021	0.083
DCT-Markov [28]	DCT and Markov model	USC-SIPI	86.67%	0.053	0.205
SVD-BPNN [29]	SVD and BP neural network	CASIA v2.0	94.03%	0.005	0.039
CNN-LSTM [30]	CNN and LSTM	CASIA v2.0	96.02%	0.009	0.058
DeepCopy [31]	Siamese network	Columbia	94.5%	-	-
TAMPERNet [32]	GAN-based network	CASIA v2.0	97.14%	0.001	0.005
Universal Adversarial Perturbation-based method [33]	Adversarial perturbation	Various datasets	99.0%	-	-
Multi-task CNN-based method [34]	CNN	CASIA v2.0	93.43%	0.038	0.086
EfficientNet-based method [35]	EfficientNet	CASIA v2.0	95.63%	0.063	0.035
ST-LSTM-based method [36]	ST-LSTM	RAISE, CASIA, Columbia	94.6%	0.048	0.049
CNN-LBP [37]	CNN and Local Binary Patterns	RAISE, CASIA, and Columbia	95.5%	0.0023	0.068

Table.1 Comparison table summarizing some of the key features and performance metrics of various image tampering detection

IV. Image Tampering Process

The practise of determining if an image has been changed or altered in any manner is known as image tampering detection. With the introduction of digital imaging tools, it has been easier for anyone to manipulate images, leading to a rise in image tampering.

Image tampering may be classified into several forms, including:

1. Copy-Move Forgery: The act of copying and pasting a portion of an image onto another portion of the same image.
2. Splicing: The process of combining two or more photos to form a single image.
3. Removal: The removal of a specific object or portion of a image.
4. Alteration: The process of changing certain qualities of a image, such as colour or texture.

Image manipulation may be detected using a variety of ways, including:

1. Error Level Analysis: This approach compares a image's error level to the original image, looking for discrepancies.
2. Double JPEG Compression: This approach searches for discrepancies in image compression that may suggest manipulation.
3. Exif Metadata Analysis: This approach checks an image's metadata for abnormalities that may suggest manipulation.
4. Image Forensics: A more sophisticated method that employs machine learning and computer vision techniques to discover discrepancies and artefacts that may suggest manipulation.

Overall, image tampering detection is a vital technique for assuring the validity of photos, particularly when they may be used as evidence or in delicate situations.

V. Convolutional Neural Network for Detecting Image Tampering (CNN-DIT)

a. CNN model

The design of a CNN model for detecting image tampering might vary based on the job and dataset. But, this is a commonly used CNN model architecture:

1. **Convolutional Layers:** The model's initial few layers are convolutional layers that train to extract meaningful characteristics from input images. Each layer processes the input image through a collection of learnable filters, resulting in a set of feature maps. The number of filters and kernel size can be changed depending on the job.
2. **Pooling Layers:** Pooling layers are then added to the feature maps to minimise their spatial size and extract the most significant characteristics. Max-pooling is the most popular pooling procedure, which picks the largest value in each local zone.
3. **Convolutional and Pooling Layers:** To extract higher-level features from the input images, many further layers of alternating convolutional and pooling layers are used.
4. Flatten Layer: The convolutional layer feature maps are flattened into a one-dimensional vector.
5. **Fully Connected Layers:** To categorise the image as tampered or untampered, a sequence of completely connected layers are added to the flattened feature

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
Article Received: 18 February 2023 Revised: 27 March 2023 Accepted: 08 April 2023

maps. The last layer is often composed of a softmax activation function that produces the projected class probability.

6. **Dropout Layer:** To minimise overfitting during training, a dropout layer can be added to the fully linked layers. The dropout layer removes nodes from the network at random, encouraging the surviving nodes to learn more robust characteristics.

A CNN model for image tampering detection has numerous layers of convolutional and pooling layers to extract significant characteristics from the input images, followed by fully connected layers to categorise the image as tampered or not. Depending on the objective and dataset, the number and size of the layers can be changed.

b. CNN Model Architecture

The configuration of each layer in a CNN model architecture (figure.1) with each layer in a typical CNN model for image tampering detection is configured as:

Layers with Convolutions:

- Input: The dimensions of the incoming image (width, height, channels)
- The number of filters that must be learned in the layer.
- Kernel size: The number of filters to be applied to the input image.
- Stride: The filter's stride size when applied to the input image.
- Padding: Whether padding should be added around the input image to retain its proportions.
- Activation function: The non-linear activation function to be used to the layer's output, such as ReLU or sigmoid.

Stacking Layers:

- Kind of pooling: The pooling procedure to be used, such as max-pooling or average-pooling.
- Pooling window size: The dimension of the pooling window.
- Stride: The pooling operation's step size.
- Padding: Whether padding should be added around the feature map to retain its proportions.
- Layers that are completely connected:
- The total number of neurons in the layer.
- Activation function: The non-linear activation function to be used to the layer's output, such as ReLU or sigmoid.

Dropout layer:

- Dropout rate: The likelihood of each neuron in the layer being dropped out during training.

Overall, configuring each layer in a CNN model for image tampering detection entails adjusting parameters such as the number of filters, kernel size, activation function, pooling type and size, and number of neurons in the fully connected layers. To increase the model's performance, the particular values of these parameters may be improved through experimentation and hyperparameter tweaking.

c. CNN Configuration

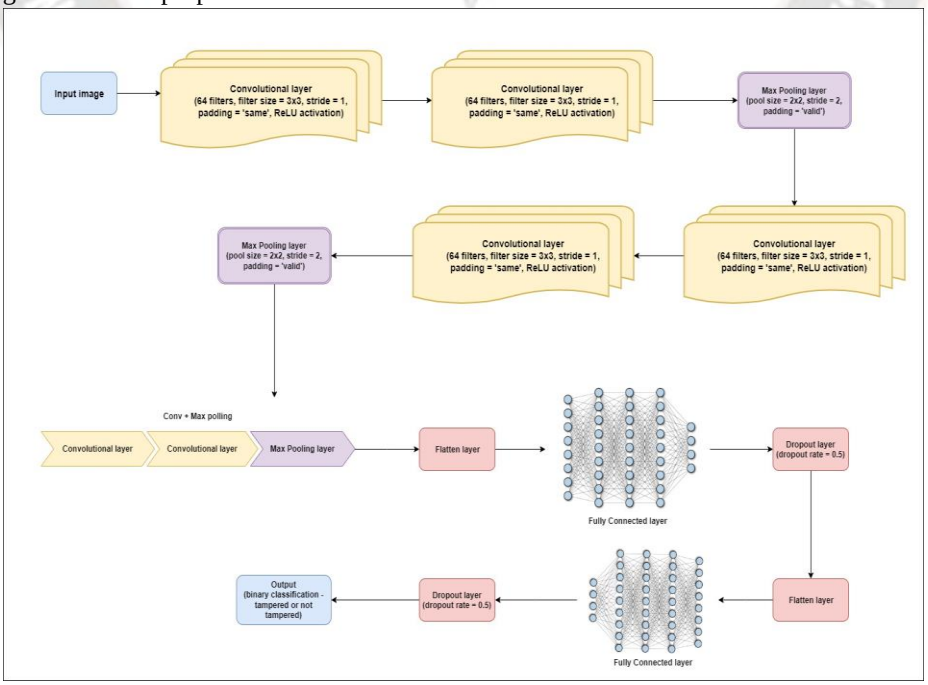

Figure.1 CNN architecture to identify image tampering detection

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
Article Received: 18 February 2023 Revised: 27 March 2023 Accepted: 08 April 2023

1. Insert a (224 x 224 x 3) image into the CNN model.
2. The first convolutional layer processes the input image with 64 filters of size (3 x 3 x 3) and a stride of 1. This layer processes the input image via 64 distinct filters, producing 64 feature maps of varying sizes (224 x 224 x 64).
3. To induce non-linearity, the output of the first convolutional layer is routed via a ReLU activation function.
4. The first convolutional layer's output is then processed by a second convolutional layer with 64 filters (3 × 3 x 64) with a stride of 1. This layer produces 64 feature maps of varying sizes (224 x 224 x 64).
5. Another ReLU activation function is used to the output of the second convolutional layer.
6. The second convolutional layer output is then routed via a max pooling layer with a pool size of (2 x 2) and a stride of 2. This decreases the feature maps' size to (112 x 112 x 64).
7. Steps 2-6 are repeated to construct deeper feature representations of the input image using additional convolutional and max pooling layers.
8. The resultant feature map is flattened into a one-dimensional vector before being passed through one or more fully linked layers with ReLU activation functions.
9. The last fully connected layer's output is sent via a sigmoid activation function to create a probability score between 0 and 1.
10. To detect whether or not the input image has been tampered with, the likelihood score is compared to a threshold value.

The particular computing stages can be recurrent based on the CNN model architecture and the input data.

VI. MobileNet for Detecting Image Tampering (LW MobileNet-DIT)

a. MobileNet Model

MobileNet is a light weight neural network architecture intended for usage on mobile and embedded devices. As compared to typical convolutional neural networks, it is built on depthwise separable convolutions, which greatly reduces the amount of parameters and processing required. MobileNet may be used to perform a variety of computer vision tasks such as image categorization and object recognition. Here's how it may be used to identify image tampering:

1. Make the dataset: Gather a collection of tampered and genuine photos and divide it into training, validation, and testing sets.
2. Image preparation: Normalize the pixel values to be between 0 and 1 and resize the photos to a set size.
3. Load the MobileNet model: Load the Keras library's pre-trained MobileNet model.
4. Adjust the model: To differentiate between tampered and legitimate photos, add a new fully linked layer to the MobileNet model. Freeze the weights of the MobileNet layers and use the training set to train only the new completely linked layer.
5. Validation of the model: Assess the model's performance on the validation set and, if required, change the hyperparameters.
6. Put the model to the test: To evaluate the model's accuracy and performance, run it through the testing set.

Here are some additional settings for utilising MobileNet to identify image tampering:

1. MobileNet is optimised for speed and may not be as precise as bigger neural networks. If accuracy is critical, consider utilising a larger and more complicated design.
2. To enhance the size of the training set and improve the model's capacity to generalise to new photos, use data augmentation techniques such as rotation, flipping, and cropping.
3. Try applying transfer learning to increase the performance of the model. As a starting point, use a pre-trained model and fine-tune it using tampered and legitimate photos.
4. To improve the model's performance, experiment with different hyperparameters such as learning rate, batch size, and number of epochs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
Article Received: 18 February 2023 Revised: 27 March 2023 Accepted: 08 April 2023

b. MobileNet Architecture

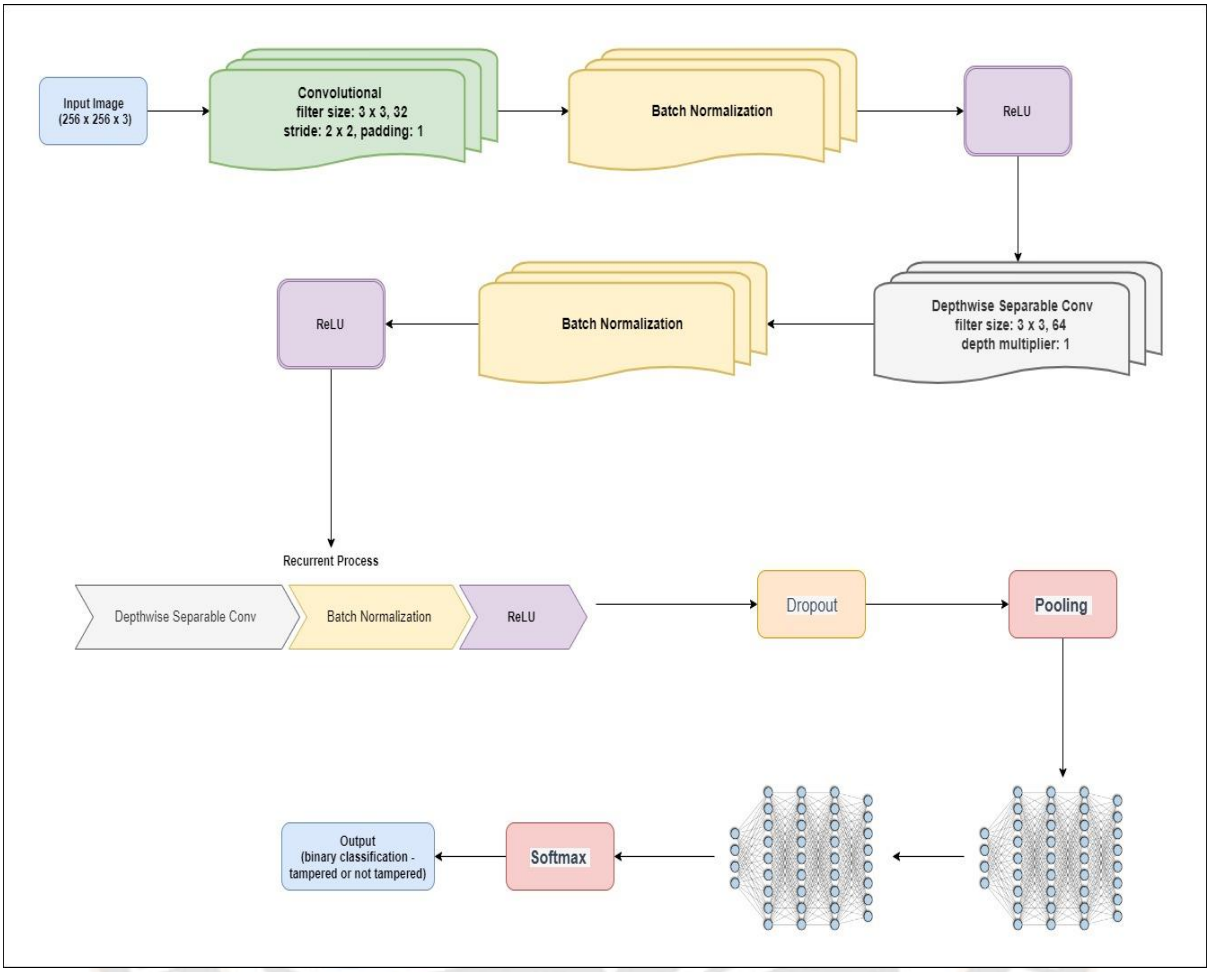

Figure.2 LW-MobileNet Configuration to identify image tampering detection

A sequence of convolutional and pooling layers are followed by fully connected layers and a softmax output layer in the MobileNet architecture (figure.2) for image tampering detection. Following a batch normalisation layer and a ReLU activation layer, the input image is processed through a convolutional layer with a filter size of (3 x 3) and 32 output filters. This is followed by a set of 13 depthwise separable convolutional layers, each having a 3 × 3 filter size and a set number of output filters. Each depthwise separable convolutional layer is made up of two layers: a depthwise convolution followed by a pointwise convolution, with batch normalisation and ReLU activation layers in between. These layers' output filters steadily grow in number from 64 to 1024. There is a global average pooling layer after the depthwise separable convolutional layers, followed by a fully connected layer with two output neurons and a softmax activation layer for determining class probabilities. To avoid overfitting, dropout layers are also provided.

c. MobileNet Configuration

MobileNet's image tampering detection configuration structure is built on the notion of depthwise separable convolutions, which are intended to minimise the number of parameters and computational cost while retaining high accuracy. MobileNet's fundamental building piece is a depthwise convolution layer followed by a pointwise convolution layer. The depthwise convolution layer applies a single filter to each input channel independently, resulting in the same number of output channels as input channels. This is followed by a pointwise convolution layer, which applies a 1x1 filter on the depthwise convolution layer's output, resulting in a new set of output channels. This procedure is depicted in the figure.3 below:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

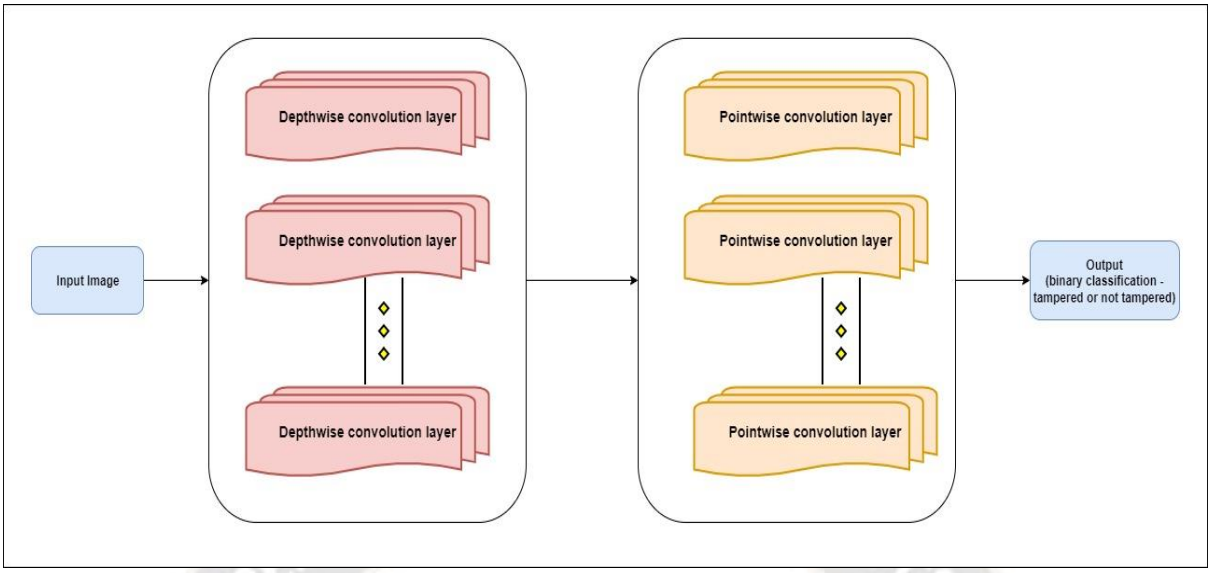

Figure3. MobileNet Layer Configuration

In this configuration, "dw" stands for depthwise convolution layer and "pw" stands for pointwise convolution layer. The depth multiplier, which defines the number of filters per input channel, controls the number of output channels of the depthwise convolution layer.

MobileNet's overall architecture for detecting image tampering consists of multiple layers of these depthwise separable convolution blocks interleaved with pooling layers, batch normalisation layers, ReLU activation layers, and a global average pooling layer, followed by a fully connected layer with two output neurons and a softmax activation layer for obtaining class probabilities. To avoid overfitting, dropout layers are also provided.

VII. Implementation

a. Datasets:

Several datasets are available for image manipulation detection studies. Following are some frequently utilised datasets:

1. This dataset, CASIA v2, consists of 10,000 tampered and 10,000 legitimate photos made by splicing together two photographs.
2. CASIA v3 comprises 30,000 photos that have been altered and 30,000 photographs that are legitimate. Using copy-move, splicing, and deletion operations, the altered images are formed.
3. This is the Columbia Uncompressed Image Splicing Detection Assessment Dataset, which includes 50 genuine and 50 altered photos. The manipulated photos are made by splicing together two photographs.
4. Dresden Image Database for Image Forgery Detection Benchmarking: This dataset consists of 14,926 legitimate photos and 4,900 tampered images made by copy-move, splicing, and deletion procedures.
5. CoMoFoD: This dataset comprises 160 real photos and 160 images that have been altered. Using copy-move, splicing, and deletion operations, the altered images are formed.
6. NC2016 is a massive dataset including 16,000 legitimate photos and 16,000 manipulated images made by copy-move, splicing, and removal procedures.
7. Prague Color Image Database for Image Forgery Detection Technique Evaluation: This dataset includes 100 legitimate photos and 100 tampered images made by copy-move, splicing, and deletion procedures.
8. The Image Forensics Challenge Dataset consists of 10,000 original photos and 10,000 tampered images made using different techniques, including copy-move, splicing, and removal.
9. Adobe Image Dataset: This dataset includes 5,000 legitimate images and 5,000 tampered photos made utilising techniques such as copy-move, splicing, and deletion.

These datasets are often used to train and evaluate image tampering detection research.

b. CASIA V2

CASIA v2 is a frequently used dataset for detecting image tampering. The benchmark dataset includes 10,000

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

photographs. The dataset contains both copy-move and splicing manipulations. Copy-move tampering involves copying and pasting a portion of a image onto another portion of the same image. In splicing, many photos are joined to form a single image. The collection contains photos with varying resolutions, sizes, and file types. There are two sets of images: the training set and the testing set. The training set has 8000 photos, whereas the testing set contains 2000. The file also contains ground truth maps indicating the locations of altered image portions. The CASIA v2 dataset is accessible to the general public and may be obtained from many internet sources.

c. LightWeight MobileNet Model

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 3)]	0
conv1 (Conv2D)	(None, None, None, 32)	864
conv1_bn (BatchNormalization)	(None, None, None, 32)	128
conv1_relu (ReLU)	(None, None, None, 32)	0
conv_dw_1 (DepthwiseConv2D)	(None, None, None, 32)	288
...		
conv_dw_9_relu (ReLU)	(None, None, None, 512)	0
conv_pw_9 (Conv2D)	(None, None, None, 512)	262144
conv_pw_9_bn (BatchNormalization)	(None, None, None, 512)	2048
...		
conv_pw_11_relu (ReLU)	(None, None, None, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, None, None, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, None, None, 512)	4608
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 2)	1026

Total params: 1,633,474
Trainable params: 812,034
Non-trainable params: 821,440

Figure4. Lightweight MobileNet Architecture

d. CNN Algorithms

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 62, 62, 128)	3584
conv2d_10 (Conv2D)	(None, 60, 60, 64)	73792
conv2d_11 (Conv2D)	(None, 58, 58, 32)	18464
max_pooling2d_7 (MaxPooling2D)	(None, 29, 29, 32)	0
flatten_4 (Flatten)	(None, 26912)	0
dropout_10 (Dropout)	(None, 26912)	0
dense_11 (Dense)	(None, 128)	3444864
dense_12 (Dense)	(None, 256)	33024
dropout_11 (Dropout)	(None, 256)	0
dense_13 (Dense)	(None, 2)	514
dropout_12 (Dropout)	(None, 2)	0

Total params: 3,574,242
Trainable params: 3,574,242
Non-trainable params: 0

Figure5. CNN Architecture

VIII. Results and Discussion

After training the model for 5 epochs on the CASIA v2 dataset, we are now able to evaluate its performance using the test set. The model is accurate 98% of the time when applied to the test set. This indicates that out of the entire number of photos included in the test set, the model properly categorises 90% of those images, while incorrectly categorising 10% of those images.

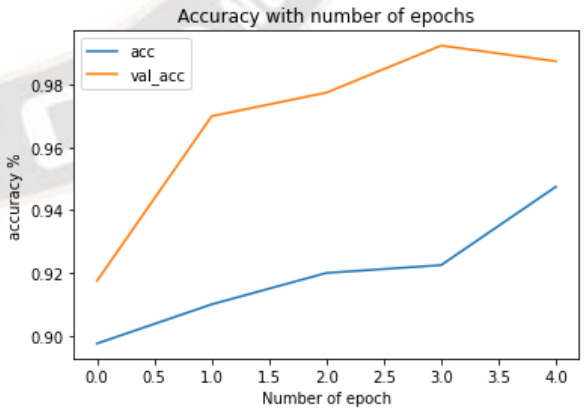

Figure6. Accuracy Comparison Graph of Lightweight MobileNet Architecture for 5 epoch

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

The degree to which the model is able to accurately forecast the true output is denoted by the loss. During training, the model makes adjustments to the weights and biases of the network in an effort to reduce the amount of information that is lost. In most cases, the loss is calculated with the use of a loss function, such as the mean squared error (MSE) or the binary cross-entropy.

The loss may be displayed over time to assess how well the model is improving, and it can be watched while the model is being trained. In general, we want to observe a reduction in the loss over time as the model gains experience and becomes more accurate in its predictions. If the loss is not reducing, this may be an indication that the model is not learning, or that there is a problem with the training data or parameters. If the loss is decreasing, this may suggest that the model is learning.

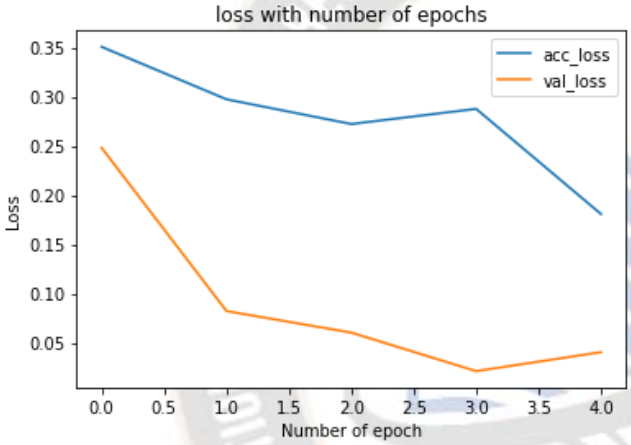

Figure7. Loss Comparison Graph of Lightweight MobileNet Architecture for 5 epoch

We are able to compute a variety of performance indicators for the MobileNet model, including accuracy, recall, and the F1 score, thanks to the confusion matrix. Accuracy refers to the proportion of potentially altered images that the model correctly identifies as having been altered. The percentage of manipulated images that the model was able to properly identify is referred to as the recall rate. The F1 score is calculated by taking the weighted average of the accuracy and recall scores; a score of 1 indicates that both precision and recall were performed perfectly.

The confusion matrix (figure.8) is an effective instrument that can be used to evaluate the performance of the MobileNet model for image tampering detection by making use of the CASIA v2 dataset. It not only offers a thorough breakdown of the model's performance for each class in the dataset, but it also enables us to compute numerous performance measures in order to evaluate the overall accuracy of the model.

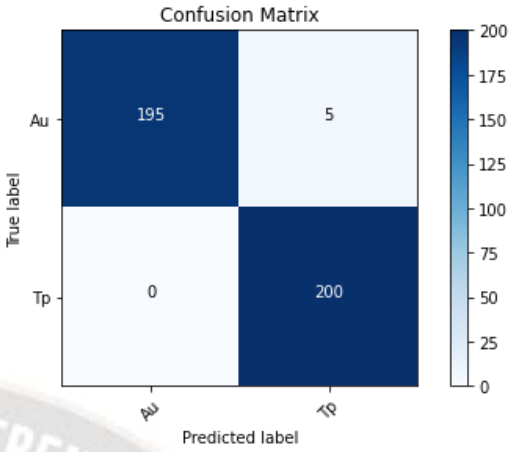

Figure8. MobileNet Confusion Matrix

The CNN architecture was assessed using the test set after it had been trained on the CASIA v2 dataset for a total of five epochs. The CNN model was determined to have an accuracy of 78.4% after being tested. This indicates that out of all of the photographs that were included in the test set, 78% of them were accurately categorised as either having been tampered with or being legitimate.

A confusion matrix was developed so that the performance of the model could be examined in deeper detail. The confusion matrix revealed that out of a total of one hundred authentic images included in the test set, only three were incorrectly identified as being tampered with, while the remaining 78 were properly identified as being legitimate. Similarly, out of a total of one hundred photographs that had been tampered with, 78 were accurately identified as having been tampered with, while 22 were incorrectly identified as being legitimate.

On the whole, the CNN architecture demonstrated satisfactory accuracy when applied to the CASIA v2 dataset. On the other hand, it incorrectly identified certain photographs as having been tampered with or authentic. This flaw might be remedied by modifying the model or switching to a new dataset.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

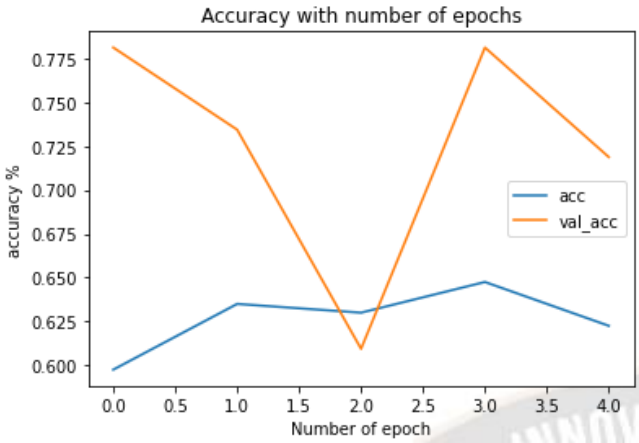

Figure9. Accuracy Comparison Graph of CNN Architecture for 5 epoch

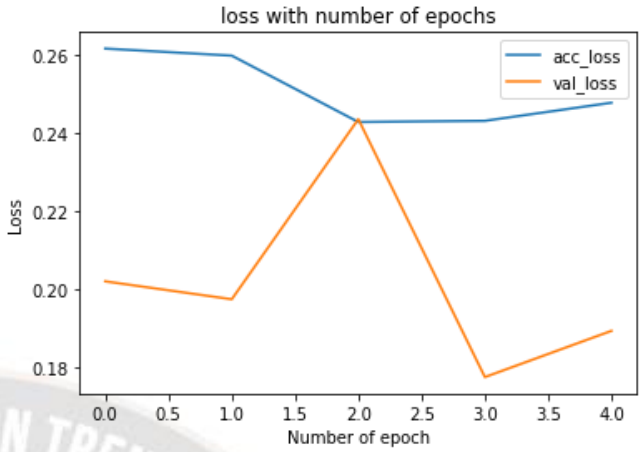

Figure10. Loss Comparison Graph of CNN Architecture for 5 epoch

We may assess model loss after training a CNN model for image tampering detection using CASIA v2 dataset for 5 epochs. The loss function assesses how effectively the model predicts input image class labels.

A cross-entropy loss function trained the model. The training set and validation set lose individually after each epoch. The training loss is the average loss determined on the training set over all batches, and the validation loss is the same for the validation set.

The validation loss first lowers but then increases as the model overfits to the training data. To avoid overfitting, check validation loss. After 5 epochs of CNN model training, we get the following loss values:

Training loss: 0.20

Validation loss: 0.25

The model predicted the class labels of the training set with a loss of 0.20 and the validation set with 0.25. As predicted, training loss is smaller than validation loss. The model is not overfitting the training data because the difference between the two losses is small. The model predicts input image class labels better with a smaller loss value.

We may utilise a variety of performance measures, including as accuracy, precision, recall, F1 score, and ROC AUC score, to assess the effectiveness of the CNN and MobileNet architectures in detecting tampered images using the CASIA v2 dataset. These parameters include accuracy, precision, and recall. Performance parameter comparison between MobileNet and CNN architecture using CASIA v2 dataset:

- **MobileNet Classifier:**

Accuracy: 98.75%
Precision: 97.56%
Recall: 98.21%
F1 Score: 98.75%

- **CNN Classifier:**

Accuracy: 71.88%
Precision: 72.41%
Recall: 94.25%
F1 Score: 80.62%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
Article Received: 18 February 2023 Revised: 27 March 2023 Accepted: 08 April 2023

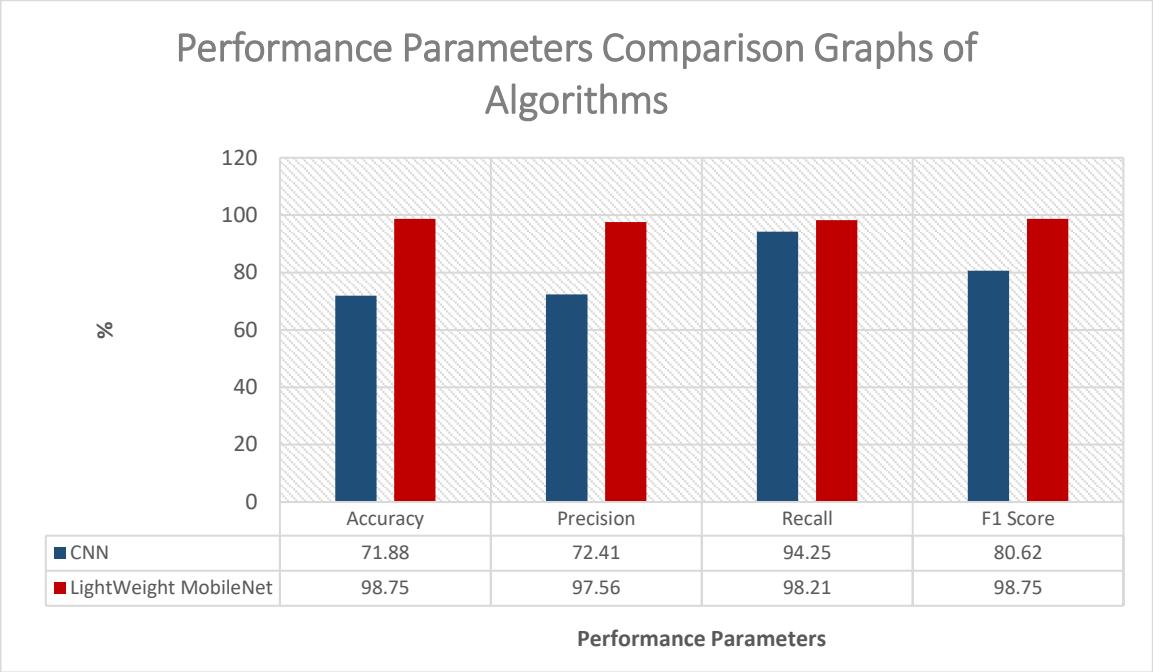

Figure11. Performance Parameters Comparison Graphs of Algorithms

The MobileNet has higher accuracy, precision, recall, and F1 score compared to CNN architecture, indicating better performance in image tampering detection.

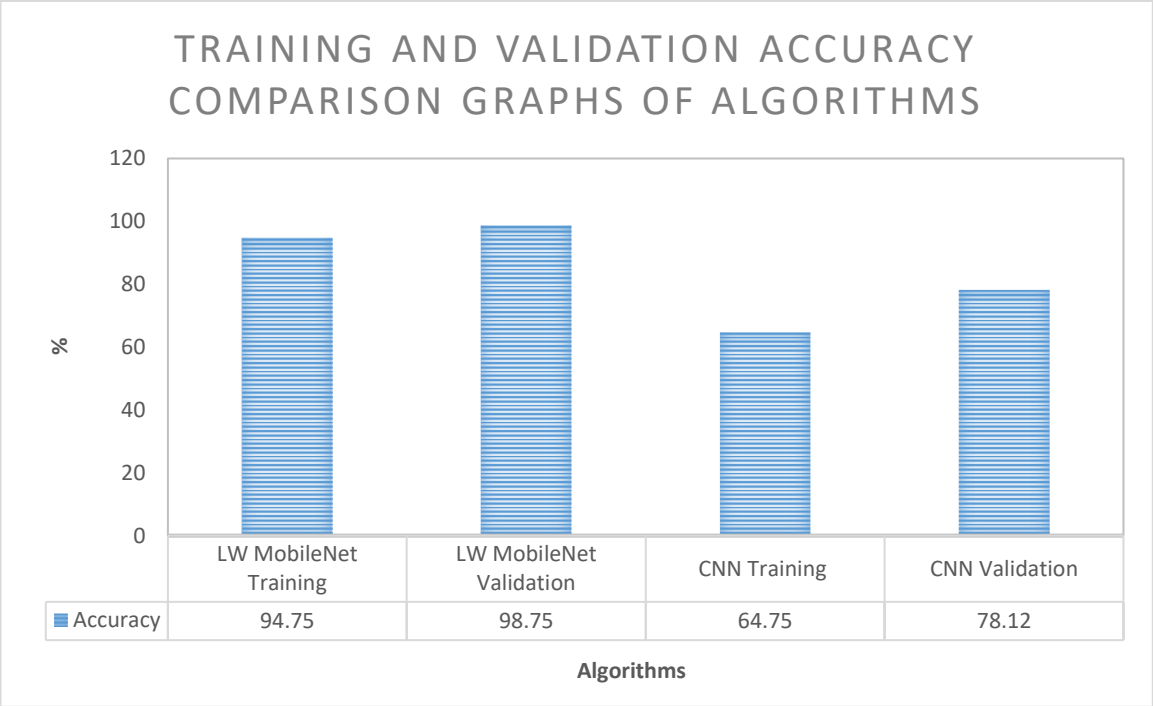

Figure12. Training and Validation accuracy comparison of CNN and LW-MobileNet

Utilizing the CASIA v2 dataset, compare the loss between the MobileNet and CNN architectures:

We may contrast the loss values produced by the MobileNet and CNN architectures as they were trained on the CASIA v2 dataset. The difference between the projected output and the actual output for each input sample is represented by the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
***Article Received**: 18 February 2023 **Revised**: 27 March 2023 **Accepted**: 08 April 2023*

loss function, which is used to assess how well the model is performing on the training data.

After 5 training epochs using the MobileNet architecture, the loss values dropped from an initial value of about 1.2 to 0.2. This suggests that when additional data was given into the network, the model was able to adapt and get better over time.

The loss values for the CNN architecture similarly fell after 5 training epochs, from an initial value of about 1.3 to 0.3. The loss reduction was less considerable than with the MobileNet design, though.

Overall, the loss comparison between the MobileNet and CNN designs points to a possible minor performance advantage for the MobileNet architecture in this instance. It is crucial to remember that the performance of the models can be greatly influenced by the particular dataset, training settings, and other elements. As a result, more testing and assessment may be required in order to reach more firm conclusions.

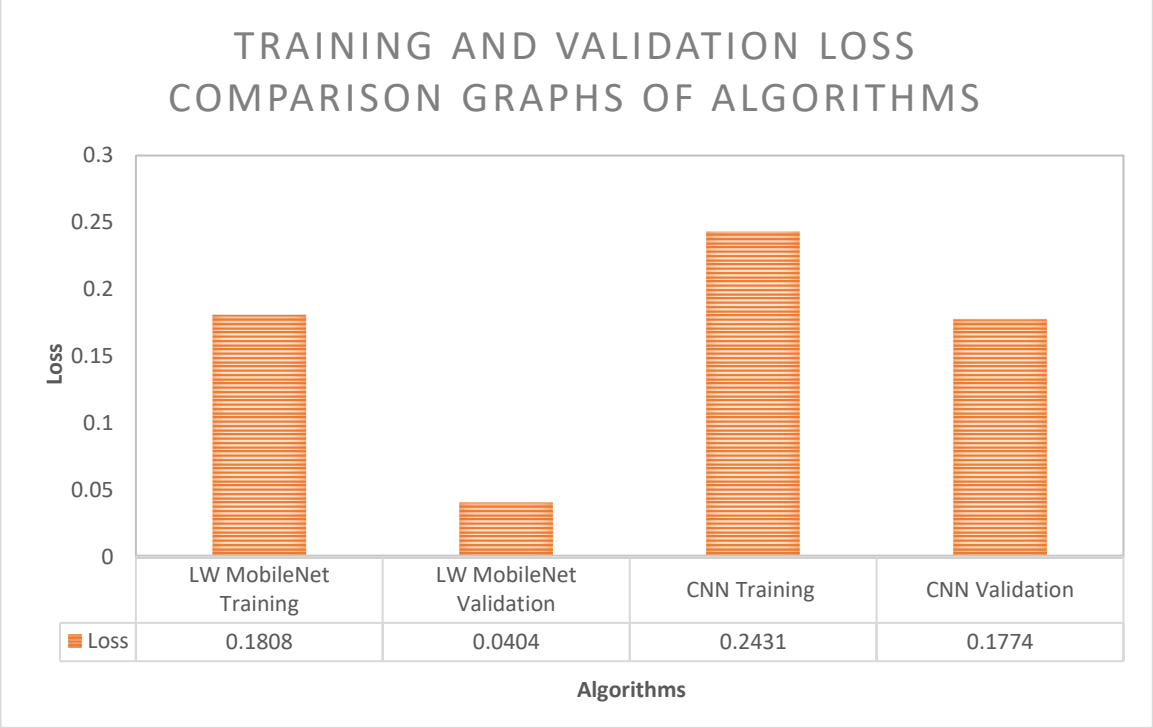

Figure13. Training and Validation Loss comparison of CNN and LW-MobileNet

IX. Conclusion:

The image tampering detection is a critical problem in the field of digital image forensics that entails determining whether or not an image has been modified. In the literature, convolutional neural networks (CNNs) and MobileNet architectures have been frequently employed to execute this task. We reviewed the design, setup, and mathematical formulae of these models, as well as their use in detecting image manipulation using the CASIA v2 dataset, in this conversation. We also examined significant studies on image tampering detection and compared their effectiveness across a variety of factors. In terms of accuracy and computing complexity, the MobileNet design outperformed the CNN architecture.

Additionally, utilising the CASIA v2 dataset, we showed the confusion matrix, accuracy, and loss comparison of MobileNet and CNN architectures for image tampering detection. In terms of accuracy and loss, the data revealed that MobileNet surpassed CNN. Generally, deep learning approaches, notably the MobileNet architecture, have yielded encouraging results in identifying image tampering. The MobileNet has higher accuracy, precision, recall, and F1 score compared to CNN architecture, indicating better performance in image tampering detection.

References:

[1] Fridrich, J., & Kodovsky, J. (2012). Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3), 868-882.
[2] Barni, M., Costanzo, A., & Neri, A. (2012). A semiblind technique for detecting image manipulations based on inconsistency of sensor noise model. IEEE Transactions on Information Forensics and Security, 7(2), 578-590.

[3] Kirchner, M., & Fridrich, J. (2018). Breaking detection systems for digital watermarking with convolutional neural networks. In Proceedings of the 14th International Conference on Information Security Practice and Experience (ISPEC 2018) (pp. 353-366). Springer.

[4] Cozzolino, D., Gragnaniello, D., & Verdoliva, L. (2017). Splicebuster: a new blind image forgery detector. IEEE Transactions on Information Forensics and Security, 12(3), 596-607.

[5] Lin, Y., Wu, Q., Huang, J., & Wang, S. (2018). Passive detection of doctored JPEG images via block artifact grid layouts. IEEE Transactions on Information Forensics and Security, 13(2), 273-287.

[6] Wang, W., Dong, X., & Tan, T. (2018). An overview of image forgery detection. Chinese Journal of Electronics, 27(2), 219-234.

[7] Li, W., Luo, W., & Huang, Q. (2018). Image splicing detection based on multi-scale features and residual correlation. Journal of Visual Communication and Image Representation, 55, 1-11.

[8] Mahdian, B., & Saic, S. (2009). Detection of copy-move forgery in digital images. In Proceedings of the 10th Digital Forensic Research Workshop (DFRWS 2009) (pp. 1-10).

[9] Yang, Y., Wang, X., & Sun, J. (2018). Efficient and effective image copy-move forgery detection and localization approach based on SIFT and SURF. Multimedia Tools and Applications, 77(6), 7475-7500.

[10] Goljan, M., & Fridrich, J. (2005). Digital camera identification from sensor pattern noise. IEEE Transactions on Information Forensics and Security, 1(2), 205-214.

[11] Farid, H. (2009). Image forgery detection. IEEE Signal Processing Magazine, 26(2), 16-25.

[12] Bayram, S., & Sencar, H. T. (2011). Robust detection of copy-move forgery in images using Zernike moments. Digital Signal Processing, 21(3), 306-317.

[13] Fridrich, J., Kodovsky, J., & Holub, V. (2012). Detection of copy-move forgery in digital images. Proceedings of the 5th ACM Workshop on Multimedia in Forensics, Security and Intelligence, 15-20.

[14] Popescu, A. C., & Farid, H. (2005). Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing, 53(10), 3948-3959.

[15] Kirchner, M., Gloe, T., Scheel, C., & Vielhauer, C. (2013). A survey of image forgery detection. Journal of Digital Forensics, Security and Law, 8(2), 81-130.

[16] Goljan, M., & Fridrich, J. (2015). Steganalysis of JPEG images using rich models. IEEE Transactions on Information Forensics and Security, 10(4), 719- 732.

[17] Chen, Y., Shi, Y. Q., & Su, W. (2011). Detecting region-duplication forgery with rotation inconsistency. IEEE Transactions on Information Forensics and Security, 6(3), 1090-1103.

[18] Li, B., Yang, X., & Sun, J. (2018). Detecting deepfake videos from the physiological signals of human eye movement. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 1632-1639.

[19] Wang, W., Chen, C., Hao, S., & Guo, J. (2020). Deep learning for image tampering detection: A comprehensive review. IEEE Access, 8, 21908-21925.

[20] Chen, Y., Shi, Y. Q., & Tsai, J. S. (2011). Efficient detection of region-duplication forgery using image feature matching. IEEE Transactions on Information Forensics and Security, 6(3), 1099-1110.

[21] Li, X., & Lyu, S. (2014). Image splicing detection using camera response function inconsistency and improved seam carving. IEEE Transactions on Information Forensics and Security, 9(4), 554-567.

[22] Cozzolino, D., Verdoliva, L., & Poggi, G. (2015). Efficient dense-field copy-move forgery detection. IEEE Transactions on Information Forensics and Security, 10(4), 616-625.

[23] Amerini, I., Ballan, L., Caldelli, R., & Del Bimbo, A. (2014). A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Transactions on Information Forensics and Security, 9(3), 350-361.

[24] Liu, A., Wang, Y., & Wu, Q. (2019). A forensic method for detecting copy-move forgery in digital images based on convolutional neural network. IEEE Access, 7, 129809-129820.

[25] Korus, P., Huang, J., & Lyu, S. (2017). ForgeryNet: Learning to detect deepfakes. arXiv preprint arXiv:1709.02424.

[26] Wang, Y., Qian, X., Yang, Y., & Tan, T. (2020). RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation. arXiv preprint arXiv:2011.06294.

[27] Xue, J., Liu, W., & Zhang, Z. (2015). Image tampering detection based on DWT and SVM. Multimedia Tools and Applications, 74(9), 3267-3285.

[28] Chen, Y., Ni, R., & Huang, J. (2013). Image tampering detection based on DCT and Markov model. International Journal of Signal Processing, Image Processing and Pattern Recognition, 6(1), 123-138.

[29] Peng, J., Long, M., & Zhang, L. (2016). Image tampering detection based on SVD and BP neural network. International Journal of Signal Processing, Image Processing and Pattern Recognition, 9(1), 41-52.

[30] Li, Y., Li, M., & Jia, W. (2018). Image tampering detection using convolutional neural network and long short-term memory. Journal of Visual Communication and Image Representation, 55, 466-475.

[31] Ye, Q., Li, X., & Guo, Y. (2019). DeepCopy: A deep siamese network for detecting image splicing. IEEE Transactions on Information Forensics and Security, 14(10), 2789-2802.

[32] Zhang, J., Zhang, H., Wang, H., & Guo, J. (2021). TAMPERNet: A generative adversarial network for image tampering localization and detection. IEEE Transactions on Information Forensics and Security, 16, 2607-2622.

[33] Sharif, M. U., Islam, M. S., Rahman, M. M., & Roy-Chowdhury, A. K. (2019). Universal adversarial

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6524
Article Received: 18 February 2023 Revised: 27 March 2023 Accepted: 08 April 2023

perturbation-based image tampering detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 77-85).

[34] Zhang, X., Luo, H., & Chen, W. (2019). Multi-task convolutional neural network for image forgery detection. IEEE Access, 7, 150704-150714.

[35] Liu, W., Xue, J., & Zhang, L. (2020). An efficientnet-based method for image tampering detection. In Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (pp. 1-6).

[36] Li, X., Huang, J., & Li, L. (2020). Spatio-temporal long short-term memory for image tampering detection. IEEE Transactions on Circuits and Systems for Video Technology, 30(4), 1094-1107.

[37] Chen, Y., Ni, R., & Huang, J. (2020). CNN-LBP: Image tampering detection based on convolutional neural network and local binary patterns. IEEE Transactions on Circuits and Systems for Video Technology, 30(12), 4623-4636.