Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection

Ahmed Alzahrani
Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
aaalzahrani9@kau.edu.sa (corresponding author)

Received: 7 February 2024 | Revised: 28 February 2024 and 4 March 2024 | Accepted: 5 March 2024

DOI: https://doi.org/10.48084/etasr.7029

ABSTRACT

Images sent across internet platforms are frequently subject to modifications, including simple alterations, such as compression, scaling, and filtering, which can mask possible changes. These modifications significantly limit the usefulness of digital image forensics analysis methods. As a result, precise classification of authentic and forged images becomes critical. In this study, a system for augmented image forgery detection is provided. Previous research on identifying counterfeit images revealed unexpected outcomes when using conventional feature encoding techniques and machine learning classifiers. Deep neural networks have been also utilized in these efforts, however, the gradient vanishing problem was ignored. A DenseNet model was created to tackle limitations inherent in typical Convolutional Neural Networks (CNNs), such as gradient vanishing and unnecessary layer requirements. The proposed DenseNet model architecture, which is composed of densely connected layers, is designed for precise discrimination between genuine and altered images. A dataset of forged images was implemented to compare the proposed DenseNet model to state-of-the-art deep learning methods, and the results showed that it outperformed them. The recommended enhanced DenseNet model has the ability to detect modified images with an astonishing accuracy of 92.32%.

Keywords-image forgery detection; convolutional neural networks; digital image forensics; deep learning; DenseNet

I. INTRODUCTION

With the advent of digital image browsing on individuals' devices and their widespread distribution via the World Wide Web, severe security concerns regarding the authenticity of images have emerged [1]. Due to the availability of powerful and user-friendly photo altering software, digital photographs are often susceptible to being altered [2]. Digital images frequently undergo deliberate aberrations during processing, the most typical of which are copy-move, interpolation, and merging. Each of these techniques is employed to mask or edit details in a digital image [3]. Forensic imaging methods are required to determine an image's authenticity, processing history, and originality [4].

A. Research Motivation

The motivation for this research stems from the urgent necessity to tackle the rising problem of picture fraud and manipulation in today's digital world. The advent of sophisticated image editing software has increased the possibility of coming across false and counterfeit images. Traditional counterfeit detection systems frequently fail to detect subtle alterations and produce reliable results [5]. Researchers have proposed numerous strategies for uncovering image forgeries [4, 6-8]. Traditional forgery detection approaches focus on seeing numerous artifacts inside modified images, such as changes in lighting, contrast, compression, sensor disturbances, and reflections. In the current state of research [1-4], supervised Machine Learning (ML) and Deep Learning (DL) algorithms are used in image forgery detection. Conventional methods for spotting image forgeries have accuracy and robustness limits, especially when dealing with sophisticated modifications. DL, a cutting-edge technique, offers a way to address these issues by automatically learning and extracting detailed information from images [4]. Considerable performance loss can be attributed to improper neural network parameter and layer selection. The above discussion emphasizes the DenseNet-based DL approach's potential as a feasible tool for effectively identifying forged images. While Convolutional Neural Networks (CNNs) show great promise, applying innovative architectures could improve their results. Previous investigations were limited by the lack of training data points and some other challenges, such as gradient fading and the requirement for a large layer count. This study, on the other hand, makes extensive use of training samples. As a result, the primary motivation of the former is the employment of an advanced CNN model with adequate training data for reliable forged image classification.

B. Problem Statement and Discussion

This study focuses on the issue of classifying images with the goal of creating a system that can detect and categorize types of image manipulation. The main challenge lies in spotting forging artifacts, which can undermine the trustworthiness and genuineness of visual material across various fields. To address this problem, we propose the use of DenseNet, a DL framework renowned for its ability to capture details and patterns within images. By enhancing the precision and dependability of algorithms, for detecting image tampering with the features of DenseNet, this study aims to overcome the shortcomings encountered by CNN and ML classifiers, which struggle with gradient disappearance and require complex layering to effectively identify changes in images. The goal is to boost the effectiveness of image manipulation detection, thereby safeguarding the authenticity of the content.

C. Research Contribution

The main raised Research Objectives (ROs) are:

- RO1: Apply a DenseNet-based DL model to detect image forgeries in digital forensics.
- RO2: Compare the results of the proposed DL model to those of other ML and DL strategies.
- RO3: Evaluate the effectiveness of the proposed model related to other comparable techniques.

When answering those ROs, the following important contributions were made:

- The proposed system employs a newly adopted DenseNet-based DL algorithm to carry out digital image forensics for the purpose of identifying image forgery.
- A suitable number of layers are incorporated into the suggested DL approach.
- Extensive experimental evaluations were used to evaluate the DenseNet's performance across a variety of measures, including precision, recall, f1-score, and accuracy.
- The study's originality lies in establishing the efficacy of the DenseNet121 CNN as a superior pre-trained model.
- The study demonstrates the effectiveness of using a lighter model architecture to achieve equivalent results with computationally more complex models.
- Evaluation of the suggested model's competence in the light of comparable research.

II. LITERATURE REVIEW

Several strategies based on ML and pattern identification have been explored in an effort to improve the detection efficiency of counterfeit photographs. To detect manipulative operations and the operator's sequencing for two operators in specific, authors in [2] presented a strong medium CNN. The limiting solution deepens the system and reduces processing. To maximize information exchange and prevent the layer fitting problem, a global average max-pooling was used. Authors in [4] showcased a strong DL system for detecting image counterfeiting in dual compression techniques. Their model was trained by comparing genuine and recompressed photographs. An end-to-end trainable BusterNet-based DL framework for duplicate image fraud prevention and detection was proposed in [5]. Authors in [6] employed DL methods to detect image forgeries in real-world datasets. Their compact model outperformed the baseline methods. The research conducted in [10] presents a revolutionary DL-based strategy for pinpointing copy-move forgeries. The method uses modified dense peak clustering to split images into patches, attention-based DenseNet121 to extract features, and adaptive chimp patch matching to match patches. Even in compressed or modified images, the method finds forged regions with an improved accuracy and less processing time. Prior research has demonstrated DenseNet's advantage over other CNN models in image classification, due to its low parameter count. Notably, DenseNet has displayed success in image forensics' analysis, particularly in spotting manipulated images.

Despite the efficiency and widespread usage of simple ML and DL approaches in image forgery detection, most previous efforts struggled to improve classification accuracy. Performance degrades due to improper neural network model parameter and layer selection. Furthermore, despite the widespread implementation of standard ML methods, only a few researchers applied DenseNet121 to categorize real and forged images. As a result, difficulties such as improving model accuracy while simplifying the model by using less parameters, layers, depth, runtime, and model size, persist. The proposed DenseNet model detects digital image counterfeiting using several layers and hyper parameters. The proposed model was also compared with benchmark research.

III. THE PROPOSED TECHNIQUE

The primary goal of this study is to use DL technique to conduct digital forensics by identifying instances of image forgery. The proposed procedure works as follows: First, a benchmark dataset is needed to evaluate performance; second, preprocessing methods are employed to clean the data of unwanted noise, and third, a DL system, namely DenseNet, is deployed to reach the ultimate prediction (Figure 1).

A. Data Set

The publically available CASIA V2.0 image tampering detection evaluation database [7] was utilized, as outlined in Table I. This accessible for research purposes database allows for the comparison and assessment of tampering detection methods. It is specifically created for identifying forgeries and is divided into two groups; altered/fake and genuine/original. The dataset consists of a total of 12,323 images with 7,200 authentic and 5,123 altered images.

TABLE I. INFORMATION ABOUT THE CASIA.2.0 DATASET

Real	Fake	Dimensions	Format
7200	5123	From 320×240 to 800×600 color images	JPEG,BMP,TIFF

B. Input Image Preprocessing

The ImageDataGenerator class and API [8] was put into service to perform some preliminary processing on the images in Keras. The imageDataGenerator class was deployed, resizing both training and testing images to 1./255, applying a shear intensity of 0.2, and randomly varying the zoom range by 0.2. It is essential to arrange the data in such a way that they can be processed by the CNN after they have been preprocessed, since the CNN can only comprehend the numerical description of the picture data. Consequently, this is why the convolutional layer's "input shape" parameter displays information as a vector. In an array, numbers represent the values of individual pixels. For instance, the network interprets an input piece of data as a 64×64×3 numeric array (where 64 represents height, 64 represents width, and 3 represents depth) [4]. Following the generation of the raw image matrix, the CNN processes the data.

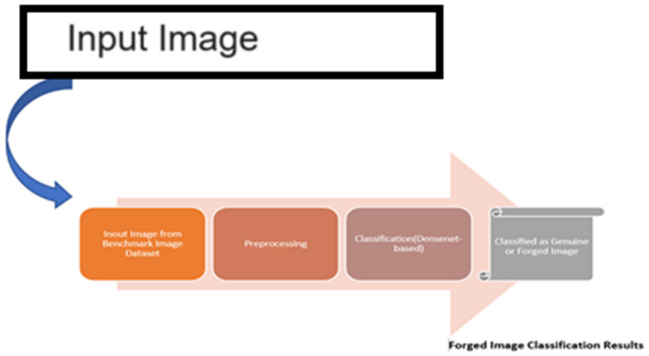
Input Image
Inout Image from Benchmark Image Dataset
Preprocessing
Classification(Densenet-based)
Classified as Genuine or Forged Image
Forged Image Classification Results

Fig. 1. Block diagram of the proposed system.

C. *Existing DenseNet Architecture*

The baseline DenseNet architecture [10] is a robust CNN architecture for visual object recognition that is noted for providing top-tier performance with fewer parameters. It is closely related to ResNet with some important modifications. DenseNet uses a concatenation attribute to integrate the output of previous and subsequent layers, whereas ResNet utilizes an additive attribute to combine previous and future levels [11, 12]. Traditional CNNs compute the output of the l^{th} layer by subjecting the previous layer's output to a nonlinear transformation designated as $H_l(.)$.

$$x_l = H_l\,(x_{l\text{-}1}) \qquad (1)$$

DenseNet differs from the standard practice of merely adding layer output feature maps to inputs by concatenating them. DenseNet's distinct methodology provides a smooth communication paradigm, improving the flow of information across layers. In particular, the inputs in the l^{th} layer are taken from the properties of all preceding layers:

$$x_l \;=\; H_l\,(\llbracket [x_0, x_1, x\rrbracket_{l-1}]) \qquad (2)$$

The tensor $[x_0, x_1, x_2, \ldots\ldots, x_{l\text{-}1}]$ is created by concatenating the output maps from the previous layers [10]. $H_l(.)$ is a non-linear transformation function. This function consists of three primary operations: Batch Normalization (BN), activation (ReLU), and a pooling and convolution combination. The growth rate, indicated as k, contributes to the generalization of the l^{th} layer as follows:

$$k_l \;=\; k_0 + \;k\; \times (l-1) \qquad (3)$$

where k_0 represents the array of the channels.

D. *Improved Densenet Architecture for Forged Image Detection*

The proposed model is a modified version of the DenseNet121 framework that has been enhanced with additional layers that have been painstakingly tailored to the unique requirements of identifying forged and genuine image data. More particular, the model makes use of the DenseNet121 neural network's intrinsic densely interconnected topology. However, as shown in Figure 2, it incorporates tailored changes to improve performance with respect to the targeted dataset.

1) *How it Works*

The proposed DenseNet architecture is made up of dense blocks with different repetitions of convolutional layers. Each dense block has two layers (1×1 and 3×3), with the former acting as a bottleneck layer to minimize input channels before convolution. Transition layers include 1×1 convolution and 2×2 average pooling (stride 2). The initial layers in the architecture include a 7×7 stride-2 convolutional layer and a 3×3 stride-2 max pooling layer. It then has three thick blocks (with 6, 12, and 24 repeats) followed by corresponding transition layers. The last dense block has 16 repetitions and leads to global average pooling for classification and the output layer. Because of its dense connections and bottleneck layers, DenseNet121 excels in computer vision applications. The proposed extension (Figure 3) improves model performance and training efficiency with fewer parameters [12]. The configuration parameters of the suggested approach are depicted in Table II.

TABLE II. DENSENET ARCHITECTURE

Layers	**Output size**	**DenseNet-121**	**DenseNet-169**	**DenseNet-201**	**DenseNet-264**
Convolution	110×110	5x5 Conv, stride 2			
Pooling	40×40	4×4 max pool, stride 2			
Dense block-1	40×40	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 8$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 8$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 8$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 8$
Transition layer-1	40×40	1×1 Conv			
	20×20	2x2 average pool, stride 2			
Dense block-2	20×20	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 16$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 16$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 16$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 16$
Transition layer-2	20×20	1×1 Conv			
	15×15	2x2 average pool, stride 2			
Dense block-3	15×15	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 32$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 48$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 64$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 82$
Transition layer-3	15×15	1×1 Conv			
	5x5	2x2 average pool, stride 2			
Dense block-4	5x5	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 24$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 48$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 48$	$\begin{bmatrix}1 & 1\\4 & 4\end{bmatrix} \times 64$
Classification layer	1×1	2x2 global average pool, stride 2			
		10000D fully connected soft max			

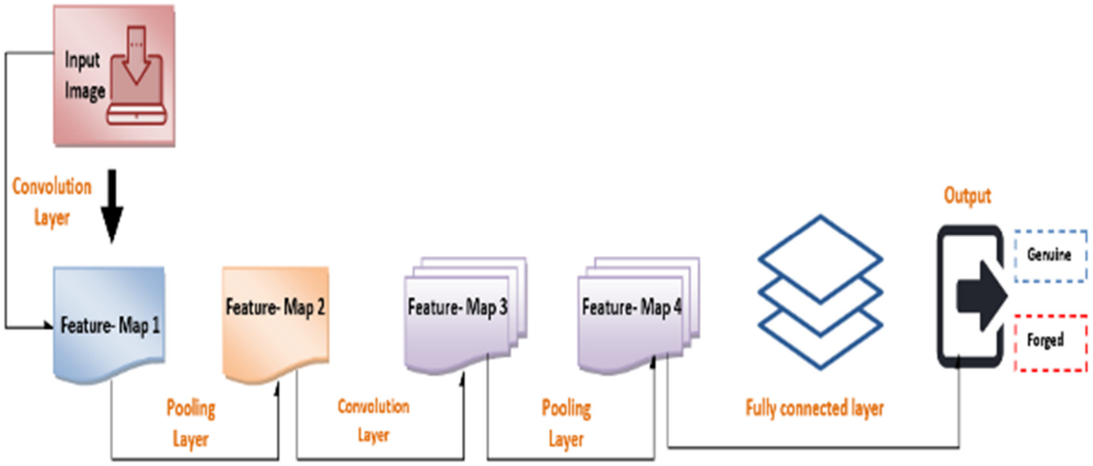

Fig. 2. Improved DenseNet's proposed architecture.

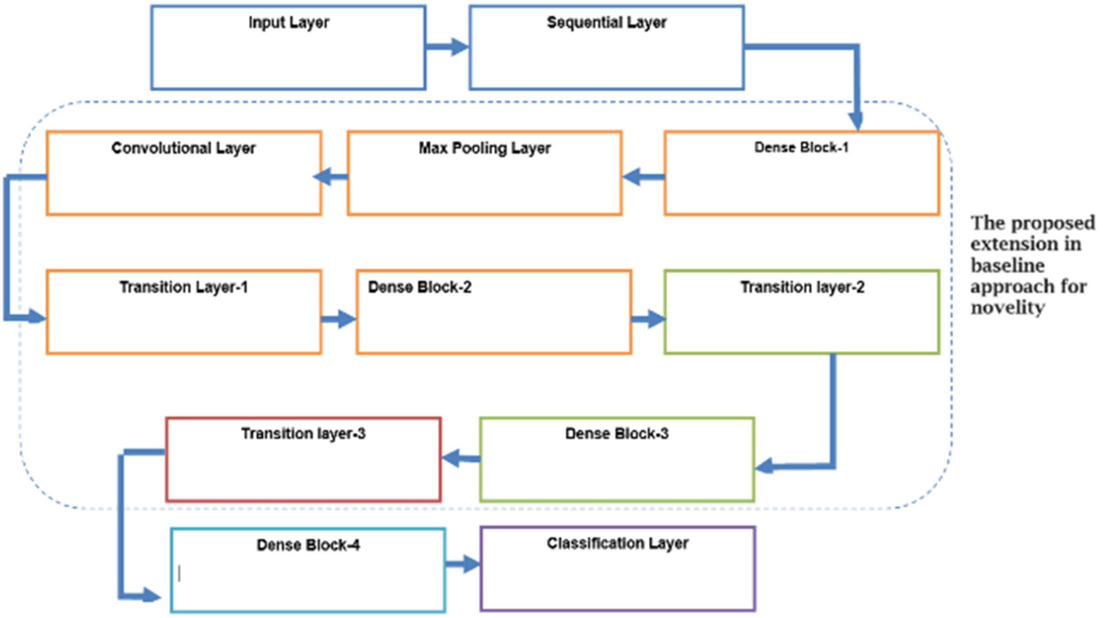

Fig. 3. The proposed improved DenseNet architecture for forged image detection.

The following algorithm describes the pseudocode steps of the proposed system:

```
Data Preprocessing:
def preprocess_data(data_path):
#Load the CASIA 2.0 dataset
real_images,fake_images =
load_dataset(C://CASIA2.0dataset//images)
#Image resizing and normalisation
real_images =
resize_and_normalize(real_images)
fake_images =
resize_and_normalize(fake_images)
#Combine actual and forged images with
their respective labels
data =
np.concatenate((real_images,fake_images))
labels =
np.concatenate((np.zeros(len(real_images))
,
 np.ones(len(fake_images))))
#Shuffle data and labels To avoid biases
data,labels = shuffle(data,labels)
Model Definition:
defbuild_model():
#Load pre-trained DenseNet-121 model with
ImageNet weights
model =
DenseNet121(weights="imagenet",include_top
=False)
#Freeze pre-trained layers
for layer in model.layers:
layer.trainable = False
```

```
#Add custom layers for binary
classification
x = model.output
x = Flatten()(x)
x = Dense(1,activation="sigmoid")(x)
#Compile the model with appropriate loss
function,optimizer,and metrics
model =
Model(inputs=model.input,outputs=x)
model.compile(loss="binary_crossentropy",
optimizer="adam",metrics=["accuracy"])
return model
Model Training:
deftrain_model(model,data,labels,epochs):
#Define data augmentation for improved
generalizability (optional)
datagen = ImageDataGenerator(...)
#Train the model with the specified
data,labels,and epochs
model.fit(datagen.flow(data,labels,batch_s
ize=32),epochs=epochs)
Image Prediction:
defpredict_image(model,image_path):
#Load and preprocess the image
image =
load_and_preprocess_image(image_path)
#Predict the class probability
prediction =
model.predict(np.expand_dims(image,axis=0)
)[0][0]
#Determine and print the result
if prediction < 0.5:
print("Image is classified as Real")
else:
print("Image is classified as Fake")
Evaluation and Analysis:
defevaluate_model(model,data,labels):
#Evaluate model performance on test data
loss,accuracy =
model.evaluate(data,labels)
print(f"Test loss: {loss:.4f}")
print(f"Test accuracy: {accuracy:.4f}")
#Generate and analyze Class Activation
Maps (CAMs) for insights into model's
decision
```

2) Applied Example

A two-dimensional grid of pixel values was the first representation of the digital image 2620.jpg, which was stored on a hard drive. Red, Green, and Blue (RGB) values were used to create a 3-tuple that represented the color information for each pixel. It was crucial to convert this 2D image into a 3D array with 220×220×3 dimensions in order to prepare it for inclusion into a DL model. The RGB values of each pixel were divided into several channels and the image was enlarged to 224×224 pixels. For model compatibility, the RGB values were scaled down from their original range of 0 to 255 to a range of 0 to 1 by dividing each value by 255. Figure 4 provides a graphic breakdown of the procedure [13]. The common method of pooling was implemented to optimize the input image for use in DenseNet. The supplied image's dimensions were reduced from 224×224×3 to 7×7×3 with this technique. Pooling minimizes the quantity of the input data while preserving key characteristics. The input image is divided into more manageable, non-overlapping portions, and a representative statistic is computed for all sections. In this illustration, the preprocessed 3D array image was separated among two segments of 2×2 size, and the highest value inside each segment was kept. The 3D array's dimensions were effectively decreaased to 7×7×3 with this step.

After the pooling operation, a flatten layer, a crucial component in many DL models, processed the resulting 3D array. This layer combined all array values along a single dimension, resulting in a vector spanning of 147 units, and converted the 3D array into a single 1D array. This flatten layer's function is to transmute the output of the preceding layer. It may have various dimensions, inside a vector of fixed length suitable for the next layers. This procedure is essential for creating a model input structure that is consistent and matches the predicted insertion data shape. This 1D array incorporates, in a condensed and standardized structure, all the appropriate information that was taken from the input image and has been preprocessed and pooled [14]. The model's later layers can analyze data more quickly thanks to this format. Figure 5 illustrates the entire series of actions outlined above.

Following the preprocessing and flattening of the input image, the ensuing vector/ray is directed via the concealed layers of the suggested DL model. The flattened vector, spanning 147 units, is initially inputted into a concealed layer harboring 50 neurons. Within this layer, each neuron takes the input vector (x) and conducts a multiplication with a distinctive weight matrix (w) designated for that neuron. Subsequently, the outcome of this multiplication is augmented by a distinct bias term (b) allocated to the same neuron. These actions yield a scalar value known as z for every neuron residing in the layer.

Rectified Linear Unit (ReLU) activation function transforms the z-values, adding a nonlinear component to each neuron's output. The output of the layer, 50 dimensional vectors, has been sent to the next layer that is hidden with 30-neurons. Within this layer, a similar process takes place and results in a vector with a length of 30. Two neurons are housed in the output layer of the model. A softmax activation function is encountered by the z-values in this layer, normalizing the data and generating a probability distribution between the two classes. This leads to an output 2D vector in which every component depicts the probability that the given class will handle the input image. The two considered classes are genuine and forged. The input image classification is affected by the softmax result being measured against a threshold. The output of the model is shown as a [0, 1] vector, denoting the image's classification. Figure 6 provides a clear illustration of the process involving the connection of the flattened vector to the layers in the suggested DL model and the operations in the hidden and output layers.

3) Implementation of the Proposed Model

Python programming language and appropriate libraries designed for the DenseNet121 architecture were employed to implement the proposerd DenseNet model. A DL framework like TensorFlow or PyTorch allows to build and train the model by streamlining the computation of the architecture's parameters and layers. Table III offers a brief summary of the key features of the DenseNet model. This summary is extremely helpful in understanding the model's structural composition and provides a starting point for comparison with similar models in the field of natural image identification.

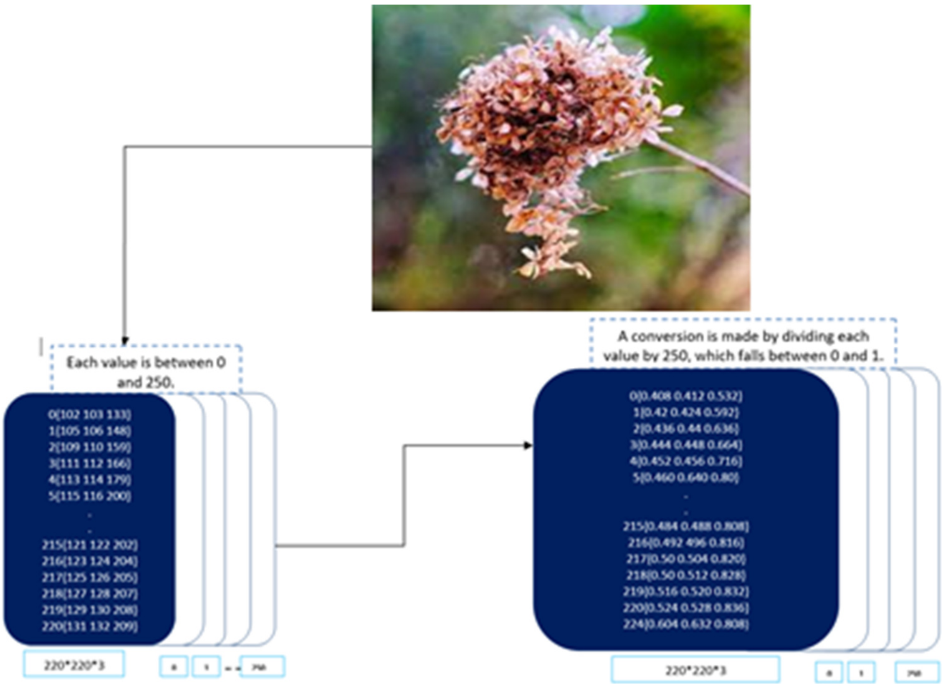

Fig. 4. Arrays of integers from an image.

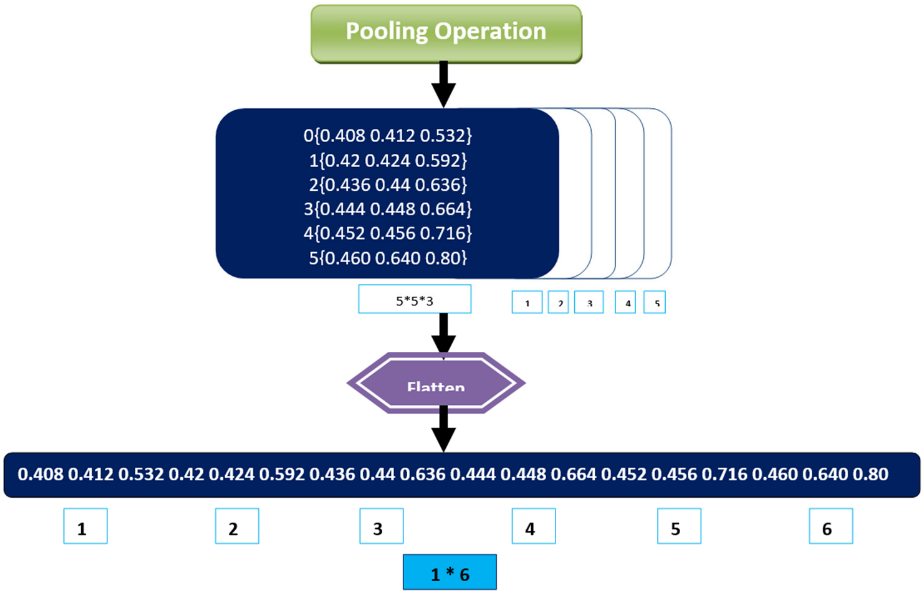

Fig. 5. Pooled and flattened array.

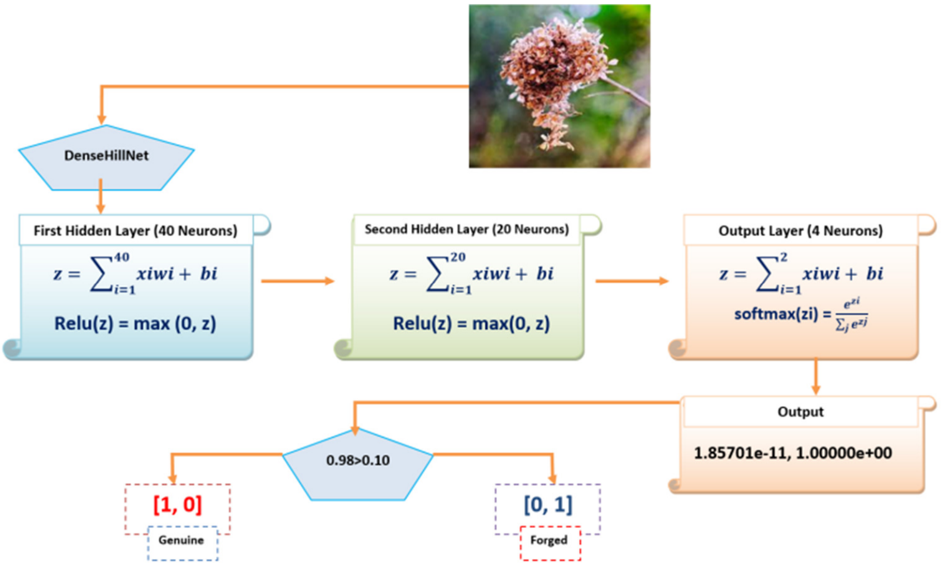
DenseHillNet
First Hidden Layer (40 Neurons)
$z = \sum_{i=1}^{40} xiwi + bi$
Relu(z) = max (0, z)
Second Hidden Layer (20 Neurons)
$z = \sum_{i=1}^{20} xiwi + bi$
Relu(z) = max(0, z)
Output Layer (4 Neurons)
$z = \sum_{i=1}^{2} xiwi + bi$
softmax(zi) = $\frac{e^{zi}}{\sum_j e^{zj}}$
Output
1.85701e-11, 1.00000e+00
0.98>0.10
[1, 0]
Genuine
[0, 1]
Forged

Fig. 6. Using DenseNet to classify an input image.

TABLE III. SUMMARY OF THE PROPOSED MODEL'S PARAMETERS

Layer (type)	Output shape	Param #	Connected to
5conv_16block_con (Concatena)	(False, 6, 6, 1024)	0	5conv_15block_con[0][0]
5conv_16block_2_conv[0][0]			
B_N (BatchNormalization)	(False, 6, 6, 1024)	4090	5conv_16block_con[0][0]
ReLu (Activation)	(False, 6, 6, 1024)	0	B_N[0][0]
flatten (Flatten)	(False, 50170)	0	ReLu [0][0]
den (Dense)	(False, 40)	2508840	flatten[0][0]
den_1 (Dense)	(False, 20)	2520	den[0][0]
den_2 (Dense)	(False, 2)	102	den_1[0][0]
Total params: 9,549,006			
Trainable params: 2,511,502			
Non-trainable params: 7,037,504			

DenseNet use DenseBlocks, a modular framework that maintains feature map size inside each block while changing filter numbers. The ransition layers, which exist between DenseBlocks, cut channels in half. This improves the information flow across network layers, which helps with tasks like detecting faked images. DenseNet is created by adding an additional layer to the DenseNet blocks, as portrayed in Figure 7.

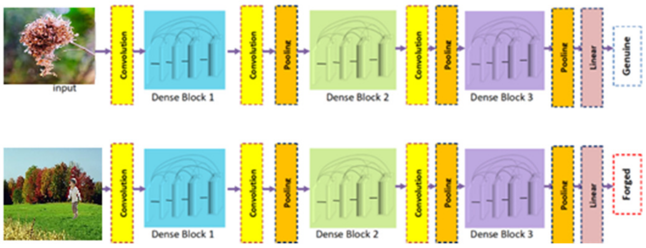

Fig. 7. DenseNet model implementation plan.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Addressing RO1

Numerous settings were deployed, which led to the evaluation of many DenseNet-based DL classifiers on the task of separating images into two distinct categories: genuine and forged. This study experimented with a variety of parameters and laboratory settings, described as follows:

To ensure an unbiased evaluation of the proposed DenseNet model's performance, the dataset was randomly divided into distinct training and testing subsets, each containing half of the original data. The model was trained with the training subset. Training was done in batches of 19 images over the course of 10 epochs. Table IV provides a thorough overview of the successes the suggested model experienced on the training dataset. The results shown in Table V can be closely examined to provide important insights into the strengths and weaknesses of the proposed DenseNet model. These discoveries might serve as a roadmap for upcoming developments and improvements in the field of digital forensics.

A comparison of the expected and projected results for true forged and true genuine image was carried out. These results were represented by corresponding vectors, [0,1] for the latter and [1,0] for the former. The projected values exhibited in Table VI have two elements [15]. The vector materializes as [1,0] if the first constituent exceeds the other, otherwise, it has the form [0,1]. Table VI displays the determined class that was derived from the values presented in Table V.

TABLE IV. PROPOSED MODEL'S ACCURACY, TEST LOSS, AND TRAINING DURATION

Epochs (E)	Time (s)	Training loss	Training accuracy	Testing loss	Testing accuracy
E-1	87	0.95	0.65	0.50	0.81
E-2	75	0.38	0.84	0.55	0.78
E-3	75	0.36	0.86	0.71	0.72
E-4	80	0.25	0.91	0.41	0.85
E-5	77	0.24	0.89	0.451	0.82
E-6	79	0.20	0.94	0.452	0.83
E-7	100	0.18	0.94	0.450	0.84
E-8	104	0.12	0.970	0.50	0.85
E-9	108	0.10	0.971	0.481	0.86
E-10	103	0.11	0.96	0.483	0.87

B. Results on Testing Data

Table V showcases the results of using the DenseNet model on a selection of 5 initial forged images and 5 initial genuine images of the testing dataset.

TABLE V. VALUES PREDICTED FROM TESTING DATA

Image	Predicted value
Forged-345	(7.1218686e-03, 9.8983681e-01)
Forged-346	(0.02381727, 0.9763827)
Forged-348	(0.02158027, 0.9784198)
Forged-355	(0.01666011, 0.9833399)
Forged-356	(0.05612681, 0.9438732)
Genuine-930	(9.9943285e-01, 4.725437e-04)
Genuine-938	(0.94824463, 0.05175542)
Genuine-941	(0.997224, 0.00277604)
Genuine-946	(9.9983799e-01, 1.6193767e-04)
Genuine-947	(0.41195405, 0.5880459)

Image Genuine-547, despite being genuine in reality, obtained an incorrect forged prediction in the context of this experiment. As a result, it is referred to as a False forged. This situation highlights the value of accurate categorization and identification of natural features while also highlighting the limitations of predictive algorithms [16-18].

The proposed model's prediction results were: 240 images were False genuine (i.e. identified as genuine whereas they were forged), 226 images were True genuine (genuine images predicted as genuine), 230 images were False forged (genuine images predicted as forged), and 127 images were True forged (forged imaged predicted as forged). Table VII shows the total results, along with the calculated accuracy, recall, and precision values. Accuracy, measures the total prediction score. Precision assesses the accuracy of identifying forged images while recall gauges the models ability to detect all forgeries. The F1 score combines precision and recall to give an assessment of the model performance. High precision reduces alarms while high recall ensures most forgeries are caught. Understanding the F1 score is crucial for grasping the models effectiveness in datasets with distributions of classes. These metrics collectively provide an assessment of how well the model performs, highlighting its strengths and areas that need improvement.

TABLE VI. IDENTIFYING THE PREDICTED CLASSES

Image	Actual (vector)	Actual (class)	Predicted (vector)	Predicted (class)	Result
Forged-221.jpg	[0,1]	Forged	[0,1]	Forged	True forged
Forged-229.jpg	[0,1]	Forged	[0,1]	Forged	True forged
Forged-291.jpg	[0,1]	Forged	[0,1]	Forged	True forged
Forged-340.jpg	[0,1]	Forged	[0,1]	Forged	True forged
Forged-799.jpg	[0,1]	Forged	[0,1]	Forged	True forged
Genuine - 191.jpg	[1,0]	Genuine	[1,0]	Genuine	True genuine
Genuine - 533.jpg	[1,0]	Genuine	[1,0]	Genuine	True genuine
Genuine - 534.jpg	[1,0]	Genuine	[1,0]	Genuine	True genuine
Genuine - 537.jpg	[1,0]	Genuine	[1,0]	Genuine	True genuine
Genuine - 547.jpg	[1,0]	Genuine	[0,1]	Forged	False forged

TABLE VII. CONFUSION MATRIX FOR THE PROPOSED MODEL'S BINARY CATEGORIZATION

Class	Precision	Recall	F1-score
Genuine	0.87	0.86	0.88
Forged	0.91	0.92	0.92
Average	0.89	0.90	0.91
Accuracy	92.32		

The proposed DenseNet model's excellent accuracy in recognizing real and forged images for digital image forensics analysis with the CASIA 2.0 benchmark collection can be attributed to several factors:

- Dense Connectivity: By directly connecting each layer to all subsequent levels, DenseNet's architecture encourages efficient information flow and feature reuse. When compared to standard CNNs, which depend purely on feedforward connections, this allows the model to acquire richer and more discriminative characteristics.
- Bottleneck Layers: These layers reduce computational complexity and overfitting by limiting the amount of input channels before each 3×3 convolution. This is especially useful for smaller datasets, such as CASIA 2.0, where overfitting is a typical concern.
- CASIA 2.0 Dataset Properties: CASIA 2.0 dataset contains a wide range of image forgery types, like copy-paste, splicing, and compression. Because of the variety and difficulty of these modifications, it serves as a solid standard for testing image forgery detection models.
- Preprocessing approaches: Using appropriate image normalization and de-noising techniques has considerably improved model performance.

In conclusion, the proposed DenseNet's excellent accuracy in distinguishing authentic and forged images for digital image forensics analysis can be attributed to its efficient design, data augmentation approach, and the nature of the CASIA 2.0

dataset. More advancements can be made by optimizing training techniques, experimenting with different feature engineering methods, and integrating many models using ensemble learning.

C. Complexity of the Proposed Algorithm

The following factors affect the algorithmic complexity:

- Image size input: Since there are more pixels and calculations are required, difficulty rises with greater image sizes.
- Network depth: Compared to shallower networks, DenseNet's 21 Dense Blocks add more layers and connections to the network, increasing its complexity.
- Growth rate: Within the Dense Blocks, DenseNet uses a growth rate parameter to regulate how many feature mappings are added per layer. Complexity rises with increasing growth rate.
- Computational operations: The network's layers carry out a variety of computations, including activation functions, pooling, and convolutions. The particular operations and their parameters determine the level of complexity.

A breakdown of the complexity in big O notation follows:

- Time Complexity: $O(N^2 \times K^2 \times D)$ is the formula for convolution operations, where N is the size of the input image, K is the size of the kernel, and D is the total number of input and output channels.
- Pooling operation: $O(N^2)$
- Activation function: $O(N^2)$
- Dense connections have a size of $O(L \times D \times N^2)$, where L is the network's layer count.
- Space Complexity: features Map: $O(L \times D \times N^2)$ and weights and biases: $O(L \times D \times K^2)$.

As a concequence, the entire complexity of the proposed DenseNet algorithm for digital image forensics analysis can be expressed as:

Time complexity: $O(N^2 \times D \times K^2 \times L) + O(N^4)$

Space complexity: $O(L \times D \times N^2) + O(L \times D \times K^2)$

D. Addressing RO2

To answer RO2, the performance of the proposed DenseNet model was compared with those of different ML and DL classifiers . The results are listed in Table VIII. Established metrics were used to evaluate how well it can distinguish genuine from from manipulated images.

E. Addressing RO3

The findings of the baseline methods and the suggested method are compared in Table IX, with the proposed model outperforming the baseline methods. The suggested model performed best in forge picture recognition (92.32% accuracy).

TABLE VIII. COMPARATIVE RESULTS

ML/DL model	Precision	Recall	F1 score	Accuracy
Xception	0.77	0.74	0.73	76.51
Mobilenetv3-small	0.78	0.75	0.77	72.21
Resnet-50	0.59	64	0.63	69.24
LSTM	0.61	0.63	0.62	63.31
RNN	0.71	0.71	0.71	70.64
Improved DenseNet (proposed)	0.89	0.9	0.91	92.32

TABLE IX. PROPOSED MODEL VS. RELATED RESEARCH

Work	Method	Results (accuracy)
[2]	Forged image recognition with DL	0.8542
[10]	Forged image detection with DL	0.882
Proposed	A novel model based on the DenseNet121 architecture, a CNN version with densely connected layers	0.9232

The proposed CNN is contrasted with the CNN model from [2], with four fully connected layers, which recognized falsified images. The quad convolution CNN architecture had lesser effectiveness (accuracy: 85.42%) than the proposed CNN model (92.32%). The CNN different parameter settings and insufficient layer count are held responsible for the baseline model's inadequacy, thus, the effectiveness of the CNN degrades whenever the number of concealed layers is increased by even more than 2. The performance of the proposed CNN model driven by DL was also contrasted with the research conducted in [10]. The recommended method surpassed the benchmark work across a variety of parameter combinations of layers in a deep neural network, including filters, step size, block size, number of iterations, and various measures like precision, recall, and F1-score. A significant difference is revealed when contrasting the current study with the findings in [10]. The latter used a CNN model and was limited to categorizing photos of glaciers as "yes" or "no," with an accuracy rate of 72%. The proposed study, on the other hand, adopted a more thorough methodology by including both forged and real photos in the training data. This method produced a higher accuracy of 84% by adopting a less complicated model architecture.

This comparison emphasizes how essential it is to take a greater variety of natural factors into account when developing ML models that correctly categorize images. The proposed work also shows how a simpler model design can produce outcomes that are equivalent to those of more complex, computationally expensive models. The DenseNet model suggested provides significant advantages in various areas of digital image forensics. In media the former can counteract the proliferation of fake photos and videos, playing a crucial role in combating misinformation and protecting users from online scams. The journalism industry can benefit from the model's capacity to authenticate images used in news reports ensuring correctness and preventing the spread of fake news. In cybersecurity and law enforcement, the proposed model may assist in investigating crimes involving manipulated images, such as identity theft and cyber fraud, by identifying photos and deepfake identities. Moreover, the medical field can be favored from the model's ability to guarantee the accuracy of diagnoses and treatments. These instances highlight the model's adaptability and its potential to improve image integrity and security across various sectors. With the advancements in DL technology, it is anticipated that DenseNets applications will broaden further as it strengthens efforts, against image manipulation.

V. CONCLUSION AND FUTURE WORK

In conclusion, identifying fake photographs is crucial for a number of industries, including transportation and outdoor recreation. Convolutional Neural Networks (CNNs) have emerged as a popular method for classifying and recognizing images. While earlier studies focused on a variety of image categories, the topic of forged image detection is still largely untapped. The presented DenseNet-based picture forgery recognition model performs well at differentiating genuine and altered photos. Three stages are involved in applying the DenseNet model: data gathering, data preprocessing, and model deployment. The current study presents a novel model based on the DenseNet121 architecture, a version of CNNs with densely connected layers. This model effectively distinguishes fabricated and genuine photos. The model was evaluated with a dataset of 12,614 photos, both forged and genuine, and remarkable results with an accuracy rate of 92.32% were obtained. This accomplishment places the recommended model among the best in forged picture detection tasks, demonstrating its high performance.

The proposed model is a modified version of the DenseNet121 architecture, with extra layers added to meet the special requirements of forged picture detection. This architectural arrangement consists of several layers, beginning with an initial convolutional layer with 64 7×7 filters and a stride of 2. Following that, dense blocks and transition layers are incorporated into the architecture. The final dense block consists of 16 repetitions, ending in the output layer with a global average pooling layer that consolidates all network feature maps for classification. Despite its achievements, the suggested model has some limitations. These include using one dataset and a specific DenseNet design, without testing trained models. Suggestions for research involve broadening the types of studied images, integrating various datasets, and trying out pre-trained neural network models like AlexNet, VGG, and ResNet. Moreover, there are plans to investigate neural network setups. such as CNN+BiLSTM and CNN+BiGRU, to conduct a thorough image forgery analysis. This effort aims to improve the model's accuracy in various scenarios.

REFERENCES

[1] A. Khattak, M. Z. Asghar, M. Ali, and U. Batool, "An efficient deep learning technique for facial emotion recognition," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 1649–1683, Jan. 2022, https://doi.org/10.1007/s11042-021-11298-w.

[2] S.-H. Cho, S. Agarwal, S.-J. Koh, and K.-H. Jung, "Image Forensics Using Non-Reducing Convolutional Neural Network for Consecutive Dual Operators," *Applied Sciences*, vol. 12, no. 14, Jan. 2022, Art. no. 7152, https://doi.org/10.3390/app12147152.

[3] A. Kuznetsov, "Digital image forgery detection using deep learning approach," *Journal of Physics: Conference Series*, vol. 1368, no. 3, Aug. 2019, Art. no. 032028, https://doi.org/10.1088/1742-6596/1368/3/032028.

[4] S. S. Ali, I. I. Ganapathi, N.-S. Vu, S. D. Ali, N. Saxena, and N. Werghi, "Image Forgery Detection Using Deep Learning by Recompressing Images," *Electronics*, vol. 11, no. 3, Jan. 2022, Art. no. 403, https://doi.org/10.3390/electronics11030403.

[5] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization," in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 170–186, https://doi.org/10.1007/978-3-030-01231-1_11.

[6] S. I. S. M. Shazuli and A. Saravanan, "Improved Whale Optimization Algorithm with Deep Learning-Driven Retinal Fundus Image Grading and Retrieval," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11555–11560, Oct. 2023, https://doi.org/10.48084/etasr.6111.

[7] J. Dong, W. Wang, and T. Tan, "CASIA Image Tampering Detection Evaluation Database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, Beijing, China, Jul. 2013, pp. 422–426, https://doi.org/10.1109/ChinaSIP.2013.6625374.

[8] K. Team, "Keras documentation: Image data loading," *Keras*. https://keras.io/api/data_loading/image/.

[9] "Architecture of DenseNet-121," *OpenGenus IQ: Computing Expertise & Legacy*, Aug. 26, 2021. https://iq.opengenus.org/architecture-of-densenet121/.

[10] R. Rajkumar, "Deep Learning Feature Extraction Using Attention-Based DenseNet 121 for Copy Move Forgery Detection," *International Journal of Image and Graphics*, vol. 23, no. 05, Sep. 2023, Art. no. 2350042, https://doi.org/10.1142/S0219467823500420.

[11] S. Alotaibi, "A Fairness-based Cell Selection Mechanism for Ultra-Dense Networks (UDNs)," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11524–11532, Oct. 2023, https://doi.org/10.48084/etasr.6106.

[12] K. H. Hingrajiya and C. Patel, "An Approach for Copy-Move and Image Splicing Forgery Detection using Automated Deep Learning," in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, Mar. 2023, pp. 1–5, https://doi.org/10.1109/ESCI56872.2023.10100202.

[13] R. Zhang and J. Ni, "A Dense U-Net with Cross-Layer Intersection for Detection and Localization of Image Forgery," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, Feb. 2020, pp. 2982–2986, https://doi.org/10.1109/ICASSP40776.2020.9054068.

[14] V. Verma, D. Singh, and N. Khanna, "Block-level double JPEG compression detection for image forgery localization," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 9949–9971, Jan. 2024, https://doi.org/10.1007/s11042-023-15942-5.

[15] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, "Deep Fake Image Detection Based on Pairwise Learning," *Applied Sciences*, vol. 10, no. 1, Jan. Art. no. 370, 2020, https://doi.org/10.3390/app10010370.

[16] D. Alghazzawi, O. Bamasag, A. Albeshri, I. Sana, H. Ullah, and M. Z. Asghar, "Efficient Prediction of Court Judgments Using an LSTM+CNN Neural Network Model with an Optimal Feature Set," *Mathematics*, vol. 10, no. 5, Jan. 2022, Art. no. 683, https://doi.org/10.3390/math10050683.

[17] M. M. H. Milu, M. A. Rahman, M. A. Rashid, A. Kuwana, and H. Kobayashi, "Improvement of Classification Accuracy of Four-Class Voluntary-Imagery fNIRS Signals using Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10425–10431, Apr. 2023, https://doi.org/10.48084/etasr.5703.

[18] I. Sahib and T. A. A. AlAsady, "Deep learning for image forgery classification based on modified Xception net and dense net," *AIP Conference Proceedings*, vol. 2547, no. 1, Dec. 2022, Art. no. 060003, https://doi.org/10.1063/5.0112143.