

## Image Authenticity Verifier - An Approach to Identify DeepFakes

Md. Ayathulla<sup>1</sup>, M. Naga Suresh<sup>1</sup>, M. Kiran Akhilesh<sup>1</sup>, M. Mani Prachand<sup>1</sup>,  
Ramya.P<sup>1</sup>

<sup>1</sup> Department of ECE, Seshadri Rao Gudlavalleru Engineering College,  
Gudlavalleru, Andhra Pradesh, India

**Abstract.** In today's digital era, social media is a crucial part of people's lives worldwide. With the advancement of multimedia technology, creating and altering content has become incredibly realistic. Deep Fake technology, powered by advanced deep learning algorithms, can create or modify facial features to look indistinguishable from real images. Therefore, it's essential to identify these fake images and stop the spread of misinformation. Researchers are developing methods to detect DeepFake images and using Convolutional Neural Networks (CNNs). These networks are trained on datasets of DeepFake images to learn how to recognize them. One technique for identifying DeepFake images involves comparing them to real images using Error Level Analysis (ELA). This paper will provide all the required information how these images are identified and discuss the challenges of creating and identifying DeepFake content in today's digital world.

**Keywords:** ELA analysis, CNN model, Feature maps, Pooling, ReLU layers.

### 1 Introduction

Methods for generating and altering multimedia content have advanced significantly, reaching a level where they can achieve remarkable realism. DeepFake, a type of generative deep learning algorithm, specializes in creating or altering facial characteristics with such precision that distinguishes between real and fake features becomes challenging. It allows for the creation of highly realistic manipulated videos and audio recordings. This leads to the spread of false narratives, identity theft, fabrication of speeches and interviews and misleading the public. Thereby, it completely erodes trust in media and people may become increasingly skeptical of video evidence.

### 2 Literature Survey

The main goal of this article is to introduce DeepFake tools that are used to manipulate the different aspects of images and videos [1]. Deepfake represents a recent breakthrough in deep learning applications. It entails the use of artificial intelligence (AI)

techniques to generate modified images or videos that accurately imitate real ones [2]. Several methods are used to identify phoney photographs, such as image tampering detection, digital watermarking, and error level analysis [3]. Following each Convolution (CNV) layer has a pooling layer that generates the outputs that are transferred to the Fully-Connected (FC) layer, which generates the output [4]. Convolution and pooling layer serial pairs are employed to carry out the feature extraction procedure [5]. With CNN's application in computer vision, it finds utility in the field of image forensics [6]. Data augmentation involves creating a new dataset from an existing one, thereby providing more diverse and referenced data for testing and training purposes and thereby increasing the accuracy [7]. In neural networks, feature extraction involves extracting learned image elements from a pre-trained CNN [8]. On the contrary, in image forensics, resizing tends to destroy precious high-frequency details, impacting heavily on performance [9]. The suggested architecture looks a lot like traditional methods that rely on patch-wise feature extraction, pooling, and classification when looking simply at the post-training procedures [10].

### 3 Proposed Method

Our aim is to develop an effective approach to identify Deep Fake by utilizing the deep learning algorithms, particularly focusing on CNN. This method integrates ELA and feature maps extraction within the CNN architecture to enhance detection accuracy. ELA serves as an initial step to identify potential areas of manipulation by analyzing discrepancies in compression levels, while the CNN model is trained to discern patterns and features indicative of fake images.

In this paper, a CNN model is used help is taken from ELA analysis to analyze the image and confirm if any manipulation or alterations have been performed on the image. Fig. 1 depicts the flow diagram of proposed network.

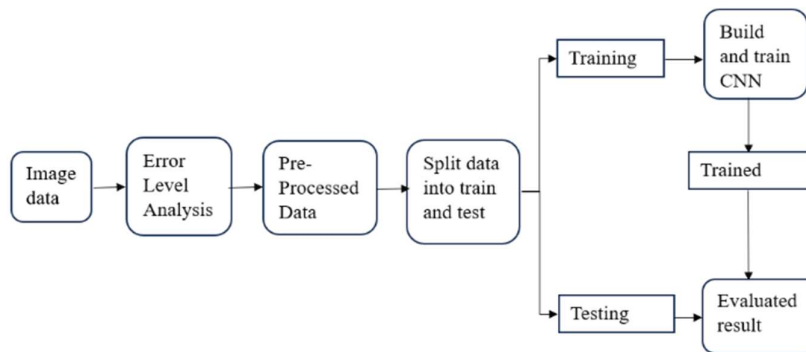


Fig. 1. Block diagram of Proposed Network

### Step 1: Image Dataset

The process begins with importing the dataset from the Kaggle website. Using the "Kaggle.json" file, which contains the credential details of all the datasets on Kaggle, we will import the dataset into the directory we created [11].

### Step 2: Error Level Analysis

The next step involves converting the input image to an ELA image, aiding in the detection of manipulations like copy-paste or morphing through lossy compression. ELA analysis is a commonly used technique in digital image forensics, allowing us to pinpoint areas of manipulation within an image. Understanding ELA analysis is facilitated by knowing how the JPEG format works. Images are typically compressed from higher to lower data, with subtle changes occurring between the original and compressed versions that are imperceptible to the naked eye. However, ELA analysis can detect these changes. In a JPEG image, the 8x8 pixel grid is treated and compressed separately, ensuring equal compression levels if no manipulation occurs. Each time the image is resaved, it undergoes compression to a certain level. By comparing the compressed and original images, the ELA representation highlights the differences, revealing varying compression potentials [3]. Fig. 2 depicts the image with a few modifications made on it and Fig. 3 shows the ELA image of Fig. 2.



Fig. 2. Input image



Fig. 3. ELA of input image

### Step 3: Data Preprocessing

Data preprocessing involves steps like image loading, enhancement, normalization, data augmentation, and labeling to organize raw data for analysis and training machine learning models. These steps increase dataset size and improve accuracy by making data suitable for analysis and model training. They are:

#### i. Image Loading

Loads dataset into memory, decoding image files into numerical arrays for further processing.

#### ii. Image Enhancement

Adjusts image quality and appearance by modifying brightness, contrast, sharpness and applying filters to improve visibility of objects or patterns for better classification.

**iii. Normalization**

Converts images to ELA images and performs Image Chops. Normalization transforms features to a common scale, ensuring no single feature dominates the learning process due to its larger magnitude, thus improving machine learning algorithm accuracy.

**iv. Data Augmentation**

Applies random transformations like flipping, rotating, scaling, cropping or adding noise to generate new data samples from existing data, expanding data volume and variety to represent real-world situations better.

**v. Labelling**

As data augmentation process increased the memory in the data set by creating new samples, assigning labels or meaningful tags to them is necessary. This labelling makes it easier for machine learning algorithms to understand and learn from the data.

**Step 4: Training and Testing Data**

Training data is used to train machine learning algorithms to identify patterns and build decision-making models, while testing data is used to assess the performance of the model. The testing data is kept separate from the model until evaluation to prevent over-fitting. Differences in distribution between testing and real-world data are noted. The model's performance is evaluated by generating predictions on the testing data and comparing them to real labels. In this model, data was divided into 80% training and 20% testing.

**Step 5: Convolutional Neural Network (CNN)**

CNN is a kind of deep learning model that is used to process grid-patterned data, such photographs, and is intended to recognize and adjust to the hierarchical features that are produced in the image. The convolution layer, pooling layer, and fully linked layer are the three basic layers that make up this model.

**i. Convolution Layer**

A variety of mathematical processes are included in this layer, one of which is convolution, a particular kind of linear operation. A tiny grid of parameters called a kernel is applied at each location in the image, serving as an optimizable feature extractor. Another important process in the Convolution layer is the activation mode, which introduces non-linearity to the previous output layer, allowing the network to capture complex patterns. Two common activation modes are "ReLU" and "SoftMax". The size of output feature map is determined as follows

$$\text{Output Size} = \frac{(\text{Input Size} - \text{Filter Size} + 2 * \text{Padding})}{\text{Stride}} + 1 \quad (1)$$

The input image size is taken as (128, 128) and the filter we are using has a kernel size of (5, 5). Padding of valid type is used, which doesn't add any padding to the input feature map, resulting in a reduction of spatial dimensions. Since we are using the keras 'Conv2D' layer, by default this layer uses a stride of (1,1). For further feature details refer to Fig. 4.

### ii. Pooling Layer

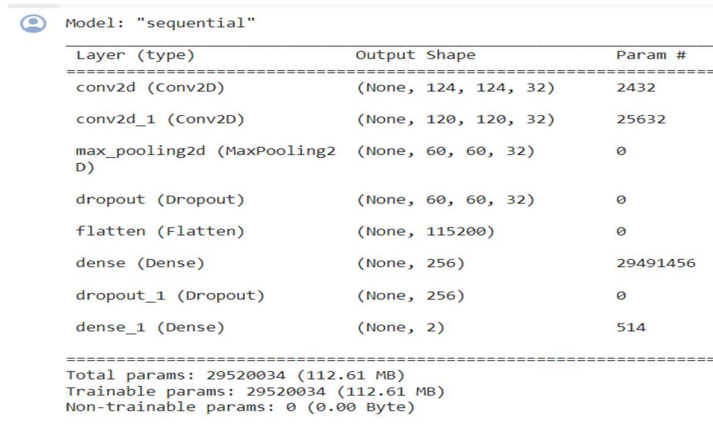
This layer preserves key information while reducing the input feature maps' spatial dimensions (width and height). This reduction facilitates the extraction of dominating features from the input, minimises computing cost, and controls overfitting. Pooling is of two types, and we currently use max pooling. Max pooling selects maximum value from each non-overlapping region of the input feature map [5]. Mathematically,

$$f_{\max}(x) = \max\{x_i\}_{i=1}^{N_i} \quad (2)$$

### iii. Fully connected layer

Every neuron in one layer is connected to every other layer's neuron by fully connected layers. Usually, the output feature map of the pooling or final convolution layers is connected to one or more completely connected dense layers after being flattened, or converted into a one-dimensional (1D) array of numbers.

## 4 Result And Discussion



```

Model: "sequential"
-----
Layer (type)                 Output Shape              Param #
-----
conv2d (Conv2D)              (None, 124, 124, 32)     2432
conv2d_1 (Conv2D)            (None, 120, 120, 32)     25632
max_pooling2d (MaxPooling2D) (None, 60, 60, 32)       0
dropout (Dropout)            (None, 60, 60, 32)       0
flatten (Flatten)            (None, 115200)           0
dense (Dense)                (None, 256)              29491456
dropout_1 (Dropout)          (None, 256)              0
dense_1 (Dense)              (None, 2)                514
-----
Total params: 29520034 (112.61 MB)
Trainable params: 29520034 (112.61 MB)
Non-trainable params: 0 (0.00 Byte)

```

**Fig. 4.** CNN model configuration

This figure showcases the results after completing the convolution layer execution. It is observed that this model alone is taking a significant amount of time to complete the feature extraction process. Therefore, we need a platform where the model is deployed, and the output is achieved instantly.

### Web Development

A web page has been designed to improve the user experience of the CNN model. This web interface offers a user-friendly platform for obtaining outputs quickly and efficiently. Users can access the CNN model from any internet-connected device without the need for complex setup or installation. This accessibility ensures that users, regardless of their technical expertise, can effectively analyze and process images using the CNN model.



Below Fig .5 is given to the web and corresponding outputs are observed.



**Fig. 5.** Sample Input image

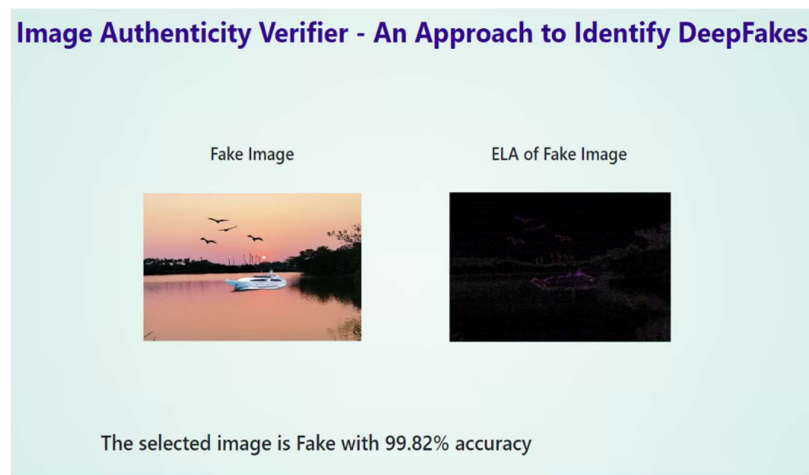
For the above input image, a few modifications have been made and those are considered as test cases. After feeding these modified images to web the outputs are as follows:

**Image Authenticity Verifier - An Approach to Identify DeepFakes**

Real Image	ELA of Real Image
	

The selected image is Real with 96.12% accuracy

**Fig. 6.** The output of sample input image with zero modifications



**Fig. 7.** Output of sample input image with few modifications

In total, 5 test cases were created. Let's discuss the output of each test case. Firstly, if an image is fed to the web, it provides an ELA image of the input, indicating whether the image is fake or not and with what level of accuracy it is making this determination

**Test case 1:** When a high brightness image is provided, the web identified it as a real image with 78.83% accuracy.

**Test case 2:** In the case of low brightness, it recognized the image as fake with an accuracy of 99.47%.

**Test case 3:** When a high contrast image is fed to the web, identified as a real image with 63.63% accuracy.

**Test case 4:** A low contrast image was also identified as a real image with an accuracy of 98.18%.

**Test case 5:** A black and white image was identified as a real image with 93.4 % accuracy.

Additionally, the sample input image was modified by adding a few fake elements. When this modified image is fed to the web, the ELA image clearly shows the areas where changes were made.

## 5 Conclusion

This study used a CNN algorithm to distinguish between manipulated and normal photos. We found that by using different filters throughout the grid layer, CNN is able to extract characteristics from images. The CNN automatically looks for feature correspondences and dependencies using the average of the feature maps that are generated. The CNN has been trained, and now the system is prepared to test and categorise the photos in order to identify copy-move forgeries. The suggested approach has been tested on many datasets with varied copy-move scenarios, such as one or more clones with distinct cloning regions. One crucial factor that is essential to the suggested

method is the quantity of training sessions. Different numbers of epochs have been used in a number of investigations. The results reveal that the best performance has been achieved during 30 epochs.

## 6 References

- [1] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," in *IEEE Access*, vol. 10, pp. 18757-18775, 2022, doi: 10.1109/ACCESS.2022.3151186.
- [2] S. R. G., A. D., Muskan, R. K. Panchalingalu, and S. Modi, "Deepfake: Creation and Detection using Deep Learning," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 11, no. 5, May 2023, doi: 10.22214/ijraset.2023.52674.
- [3] Akash, Miss. Ahalya, Mr. Dhinesh, and Miss. Diya Shreef, "Detecting Fake Images Using Machine Learning," *International Journal of Research Publication and Reviews*, vol. 4, no. 4, pp. 2063-2069, April 2023, doi: 10.55248/gengpi.2023.4.4.35702.
- [4] M. A. Elaskily et al., "A novel deep learning framework for copy-move forgery detection in images," *Multimedia Tools and Applications*, vol. 79, pp. 19167-19192, 2020.
- [5] R. Nirthika et al., "Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study," *Neural Comput & Applic*, vol. 34, pp. 5321-5347, 2022, doi: 10.1007/s00521-022-06953-8.
- [6] J. L. Zhong and C. M. Pun, "An end-to-end dense-inceptionnet for image copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2134-2146, 2019.
- [7] Abhishek and N. Jindal, "Copy-move and splicing forgery detection using deep convolution neural network, and semantic segmentation," *Multimedia Tools and Applications*, vol. 80, pp. 3571-3599, 2021.
- [8] Y. Abdalla, M. T. Iqbal, and M. Shehata, "Convolutional neural network for copy-move forgery detection," *Symmetry*, vol. 11, no. 10, p. 1280, 2019.
- [9] F. Marra et al., "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *IEEE Access*, vol. 8, pp. 133488-133502, 2020.
- [10] Y. Rao, J. Ni, and H. Zhao, "Deep learning local descriptor for image splicing detection and localization," *IEEE Access*, vol. 8, pp. 25611-25612, 2020.
- [11] <https://www.kaggle.com/datasets/sophatvathana/casia-dataset>.