Image Forgery Detection using Cryptography and Deep Learning

Ayodeji Oke and Kehinde O. Babaagba[0000−0003−0786−2618]

School of Computing, Engineering & the Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, United Kingdom 40524773@live.napier.ac.uk, K.Babaagba@napier.ac.uk

Abstract. The advancement of technology has undoubtedly exposed everyone to a remarkable array of visual imagery. Nowadays, digital technology is eating away the trust and historical confidence people have in the integrity of imagery. Deep learning is often used for the detection of forged digital images through the classification of images as original or forged. Despite many advantages of deep learning algorithms to predict fake images such as automatic feature engineering, parameter sharing and dimensionality reduction, one of the drawbacks of deep learning emanates from parsing bad examples to deep learning models. In this work, cryptography was applied to improve the integrity of images used for deep learning (Convolutional Neural Network - CNN) based prediction using SHA-256. Our results after a hashing algorithm was used at a threshold of 0.0003 gives 73.20% image prediction accuracy. The use of CNN algorithm on the hashing image dataset gives a prediction accuracy of 72.70% at 0.09s. Furthermore, the result of CNN on the raw image dataset gives a prediction accuracy of 89.08% at 2s. The result shows that although a higher prediction accuracy is obtained when the CNN algorithm is used on the raw image without hashing, the prediction using the CNN algorithm with hashing is faster.

Keywords: Image Forgery Detection; Machine Learning; Deep Learning; Cryptography; Hashing.

1 Introduction

The availability of multiple tools for image editing, enhancement, correction, alteration, and reconstruction continue to stimulate the perpetration of crimes. The widespread distribution of forged faces and videos on the Internet has caused several cultural, moral, and security problems, including fraud and fake news [8]. Synthesising whole face, identity exchange, manipulation of attribute, and expression, are various ways picture forgery is propagated by criminal elements in the manipulation of faces of victims. Research on image forgery indicates that these kinds of crimes are typically committed to spread false information, obtain political power, and create unsavoury notoriety and benefits. It is critically necessary to either put an end to the spread of false information or find a remedy

for it [35]. This suggests that potential image forgery detection systems focus on establishing the consistency and authenticity of electronic images.

Over the years, several investigation outputs have emerged with a view to mitigating and restoring the integrity and trust of digital images. Two basic approaches being employed are blind or passive image forgery and non-blind or active image forgery detection [11]. The non-blind or active image forgery detection uses a watermark and digital signature to authenticate digital images [10]. This approach simply implies that a watermark has to be inserted at the point of storage. However, a digital camera will need to be specially equipped to make this approach a reality. A digital signature is an approach used to ensure the authenticity and integrity of information that possesses a key or signature [32]. Cryptography is a subset of digital signatures, and it is a communication technique for conveying information securely [22] through the use of a private key and hash extracted from typical digital signature.

In blind detection of image forgery, the authenticity of an image is verified and confirmed without requiring further action for the purpose of authenticity. Pixel-based techniques are one of the techniques in passive image forgery detection for deciphering manipulated images. It comprises parameters such as noise, blurring, color, and scaling from which tampered images can be distinguished from the original ones. In images, the color of an observed light, which is an interaction between the illumination and the reflection of the object surface, obeys certain structures that are consistent with the physical phenomenology of color [38]. When an image is manipulated, the consistency of the color pixels with color phenomenology is also manipulated.

In addition, deep learning is an integral part of machine learning [36] used for the detection of forged digital images [37] through the classification of images as original or forged. It commonly uses digital image splicing [23], which originated from a pixel-based technique, to develop a new classifier aimed at detecting inconsistencies stemming from image manipulations that affect the object's color appearance. Thus, this study focuses on the use of a subset of features in passive and active fake image detection. This research proposes an anomalous image detection system with the use of cryptography and deep learning techniques.

The research questions that this work will address are as follows:

- Can hashing algorithms be used in detecting image forgeries?
- When hashing is used in conjunction with deep learning does it outperform a model that does not use hashing?

We answer these two questions by carrying out experiments that first check the efficacy of an hashing algorithm in detecting image forgery. We then compare the performance of the CNN algorithm without hashing. Then, we test for performance improvement of the CNN algorithm using hashing.

The rest of the paper is structured as follows. Section 2 provides a review of related works. In Section 3, we describe the methodology of our research. Then we explain our experimental design in Section 4. We present and discuss our results in Section 5. Section 6 summarises and concludes the paper, it also provides direction for future research.

2 Related Work

Images are an essential part of the digital world and are vital for both storing and disseminating data. They can easily be edited using a variety of tools [24]. These tools were originally developed in order to enhance photographs. Some individuals, however, use the functionalities of these tools to distort photos and spread myths rather than improving the image [2]. This poses a serious risk because falsified photographs often result in severe and often irreparable harm [3]. Image splicing (copying a section of a donor image into a legacy image) and copy-move (creating forged image from only one image) are the two main types of image forgeries [3].

To combat the problem of picture distortion in many fields, digital image forgery detection has substantially expanded [27]. Techniques for image forgery detection are used for both copied and spliced pictures. In order to protect digital photographs, [20] presented a hybrid cryptographic and electronic technique for watermarking. While protecting the image content with cryptography, they employed the watermarking approach to authenticate the image. [19] worked on identifying JPG, PDF, PNG, and GIF file types for digital forensics and employed computational intelligence techniques to do so. They also used these algorithms to expose the right file type if the JPG, PDF, PNG, or GIF was changed. They assessed chrominance, brightness, and grayness and claimed that their method outperformed other cutting-edge forgery detection methods. [21] developed an innovative hybrid discrete cosine conversion and visual encryption technique for protecting digital photos. They condensed the image that was encrypted using the discrete cosine transform after encrypting the image's RGB channel as n shares. The method used was reversed to produce the plain image, but pixel values were lost.

Deep artificial neural networks are referred to as deep learning. Deep learning has become increasingly popular over the past few years, largely due to recent developments in machine learning and signal/information processing research as well as dramatically improved chip processing capabilities (e.g., GPU units), significantly lower hardware costs, and other factors. Autoencoder (AE) [29], Convolutional Neural Network (CNN) [15], Deep Belief Network (DBN) [12], Recurrent Neural Network (RNN) [34], and Direct Deep Reinforcement Learning [16] are examples of common deep learning approaches. Researchers can utilize a variety of deep learning frameworks to implement any deep learning technique.

The literature has suggested a number of strategies to combat picture forgery. In contrast to modern strategies based on deep learning such as CNN, which are detailed below, the majority of older techniques are based on specific artefacts left by image forgeries. Before discussing deep learning-based strategies, we first discuss different conventional methodologies. For instance [40] presented Error Level Analysis (ELA) that identify fake image. The ELA technique operates on JPEG images based on a grid, changing a portion of a grid to affect the entire grid square. However, ELA is limited in its capability to detect single pixel modifications or minor color adjustments [31]. [28] used the illumination of objects to determine whether an image is fake. Based on the difference in

illumination direction between the authentic and fake portions of digital image, it attempts to identify forgery. Different conventional methods for detecting forged image have been compared in [9]. The contourlet transform is used in [17] to regain the edge pixels for detection of forgery. Though the contourlet transform is advantageous in multi-resolution, discrete domain implementation and multi-direction, redundancy reduction, the disadvantages are the shift variant, which leads to the pseudo-Gibbs phenomenon, poor frequency selectivity, and poor temporal stability [33].

In terms of deep learning models, [7] developed a technique for identifying fake images by using resampling characteristics and deep learning. A technique for detecting manipulated images by clustering the features of camera-based CNN was proposed by [6]. [24] invented Compression Artefact Tracing Network (CAT-Net) that simultaneously captures forensic features of artefact compression on the RGB and DCT domains. HR-Net is their principal network (high resolution). As sending DCT coefficients straight to a CNN would not effectively train it, they employed the method suggested in [42], which explains how the DCT coefficient can be used. In their Dual-Order Attentive Generative Adversarial Network (DOA-GAN), [18] suggested using a GAN with dual attention to locate copy-move forgeries in an image. The generator's first-order attention is made to gather copy-move location data, whilst the second-order attention for patch co-occurrence takes advantage of more discriminative characteristics. The co-occurrence and location-aware features are combined for the final localization branches and detection for the network by using the affinity matrix to retrieve both attention maps.

[5] describes the construction of a Dual-encoder U-Net using an unfixed encoder and a fixed encoder (D-Unet). The picture fingerprints that distinguish between authentic and altered regions are learned by the unfixed encoder on its own. The fixed encoder, on the other hand, provides direction information to aid network learning and detection. A deepfake detection technique was introduced by [25]. Only fixed-size images of the face may be produced using deepfake techniques, and these images have to be affinely twisted to match the facial layout of the source image. This warping creates various artefacts because of the different resolution between the surrounding environment and the warped face area. As a result, these artefacts can be used to recognize deepfake videos. Other authors such as [41], [26] among others, have used deep learning for image forgery detection.

In summary, numerous strategies have been put forth by researchers to identify image forgeries. Traditional methods of detecting image forgeries focus on the various artefacts that can be found in forged images, such as variations in illumination, compression, contrast, sensor noise, and shadow. The use of CNN for a variety of computer vision tasks, such as picture object recognition, semantic segmentation, and image classification, has grown in popularity recently. CNN's accomplishments in computer vision is mostly due to two characteristics. Firstly, CNN makes use of the strong correlation between nearby pixels. CNN favours connections that are grouped locally over the connection on one-to-one

bases between every pixel. Secondly, a convolution process using shared weights is used to create each output feature map. Additionally, CNN employs learned characteristics from training photos and can generalize itself to detect unseen forgery, in contrast to old techniques, which rely on manufactured features to detect specific fabrication.

CNN is a promising approach for identifying image forgeries because of these benefits. A CNN model can be trained to learn the numerous artefacts present in a fake picture. Additionally, using image artefacts with hashing has been shown to achieve integrity [13]. In order to learn the tampered image artefacts as a result of changes in original image features and ensure highly confidential images that are tamper-proof, we propose an authentication system for identifying anomalies in images using cryptography and deep learning techniques.

3 Methodology

The alteration of images using various technological approaches may have an impact on aspects of images that are challenging for humans to recognize. Using an image editor, an image effect, or any other tool to alter the imaging features can result in a bogus image. Our goal is to identify such tampered images by training a model on a dataset of real and tampered images. We employ CNN to boost prediction accuracy and efficiency and cryptography to preserve the integrity of the image.

3.1 Dataset Summary

The dataset employed in this research is the Casia dataset[1] comprising of 3416 images. The images are of the format BMP, JPG and TIF which comprises of 2391 real images and 1025 tampered images. Among the pictorial content of these images includes architectures, plants, articles, animals, textures, nature, indoor and scenes images. These images are of different sizes varying from 384256 pixels to 800600 pixels.

A deep learning model was developed using CNN to carry out learning and prediction firstly on the raw image dataset and on an image dataset that had undergone hashing. The image dataset after data cleaning consists of 3,247 images including real (2,376) and fake (871) images. In terms of the type of images in the dataset, a total of 2563 images were .JPG, 559 images were .TIF and there were 125 .BMP images. The details of the information regarding how image is distributed is presented in Table 1.

3.2 Image Processing

The process of color image segmentation and the identification of reference points are explained in this section. The procedures to achieve this are as follows:

[1] Casia Dataset - https://www.kaggle.com/datasets/sophatvathana/casia-dataset

Format	Real Image	Fake Image	Total
.JPG	1989	574	2563
.TIF	333	226	559
.BMP	54	71	125
Total	2376	871	3247

Table 1. Image Distribution after Data Cleaning

Resize the RGB Image In this phase, bi-cubic interpolation was used for the resizing of the input images to $M \times M$ matrix. The resizing of image lay credence to the fact that the input images have variety of sizes, and it is desirable that hashes of images must be of the same length. This infers that images with different sizes tend to have different length of hashes [1]. The resizing process therefore ensures that digital images that have different resolution can be identified with hashing. After the resizing of image, gaussian low-pass was then applied for the filtering of the image. The filtering process is with a view to reduce minor manipulation influence such as noise contamination which further enhance certain features in an image [39].

The source of an image contributes to the features of the image. Images derived from digital cameras or scanners have a metadata standard attaching to it which is knows as Exchangeable image file format (Exif). This Exif data consists of broad range of features such as camera settings which includes rotation, shutter speed, camera model, aperture, ISO speed information, metering mode, focal length and so on. It also includes image metrics such as color space, pixel dimensions, file size and resolution. Other features are copyright, date and time and location information. These features are mainly direct data that can be extracted from digital image. Furthermore, these digital images are stored in a computer in form of a matrix or an array of square pixels in which each element are arranged in rows and column, that is, in a 2-dimensional matrix. There is possibility of having smoother or pixelated (mosaic-like) images otherwise called image resolutions which determine the size of the square pixels. The information contained in the pixels are color and intensity. These pixels are represented as RGB in coloured images, that is, two dimensional arrays with three layers. These layers in the images depicts red, green, and blue channels together with their equivalent 8-bit digit for the intensity.

Feature Selection In this phase, after the importing features of the digital image, the interested data from the image features were selected. The basic features selected from the image data includes image shape (width, height, and size), object type (image array) and image dimension which is basically the number of array dimensions of the image which is usually three for coloured images (RGB channels). All the information in the selected features are necessary for the image pre-processing to ensure the main objective of the study is achieved.

RGB Color Pixel Normalization In this phase, the numeric values in the columns of the dataset were normalized to reduce data redundancy and improve the performance of the result expected to be derived from our implementation. The purpose of the normalization is to change the dataset value into common scale without distorting differences in the ranges of values or losing information [4]. Therefore, the pixel values which range from 0 to 255 are divided by 255 to convert it to range 0 to 1 which gives numbers with small values and makes computation faster and easier. The typical objective of normalizing image is to change the range of intensity values in the pixel. This converts the range of value in the pixel of the input image into a normal scale. Hence, this work performs normalization on the input image to scale down the pixel value without distorting or losing information in the input image. Let R, G, and B be the red, green, and blue component of a pixel, the ranges of R, G, and B are then converted from $[0, 255]$ to $[0, 1]$.

Computation of Mean from RGB Color Space In each component, every representation of RGB were divided into non-overlapping block of dimension $b \times b$. For instance, if N is the block number and Bi is the ith block ($i = 1, 2, ..., N$). Thus, the mean m was calculated to represent the block.

$$m = 1/b^2 \sum_{j=1}^{b^2} B_i(j) \tag{1}$$

The $B_i(j)$ represents the value of the jth element of the ith block. The statistical selection above is based on the consideration that the mean and variance can indicate the average energy and the fluctuation of the block. The formulas in equation (1) were applied to each channel of RGB color, and thus the color feature vector v_i of the ith block was obtained.

$$v_i = [m_R, m_G, m_B]^T \tag{2}$$

where m_R is the mean of the R channel of RGB, m_G is the mean of the G channel, and m_B is the mean of the B channel. By ordering these vectors v_i, matrix V was obtained from the feature.

$$V = [v_1, v_2, v_3, ..., v_N] \tag{3}$$

During this process, instances of images that were not in image format and possess three-dimensional space were removed. The total number of images at the end of the phase was 3247, with real images comprising of 2376 while tampered images were 871.

3.3 Hash Generation

The RGB colour channel was parsed to the SHA-256 hash function to generate the hash value for each image in the dataset. SHA-256 is a 256-bit encryption

hash function that may also be represented as a 64-digit number system in base 16 [30]. Firstly, 512 bits are added to the entry message Q to pad it. In order to generate a length that is multiples of 512 bits, a bit "1" and a variable number of zero bits are appended to a 64-bit binary denoting the length of Q. The length of the actual message is stored as a 64-bit value in the final 64 bits of the padded message. The padded message is broken up into 512-bit blocks $B^{(0)}$, $B^{(1)}$, $B^{(2)}$, ..., $B^{(N-1)}$ and each block of data $B^{(i)}$ is sequentially processed by the main function over the course of 64 loops. The final hash HN is computed and supplied after the last block of data $B^{(N-1)}$ is computed, and a hashing H_i having partial 256-bit is obtained when a whole execution of the present $B^{(i)}$ is completed [14].

Hash Similarity To measure the similarity between hashes of the original and the attacked images, the d norm is computed. Let h_1 and h_2 be two image hashes. Thus, the d norm is defined as:

$$d = \sqrt{\sum_{i=1}^{N} |h_1(i) - h_2(i)|^2} \tag{4}$$

Here $h_1(i)$ and $h_2(i)$ are the *ith* element of h_1 and h_2. The more similar the input images of the hashes, the smaller the d value. If d is smaller than a pre-defined threshold T, the original and the fake images are classified as a pair of images that are visually identical. Otherwise, they are different.

Evaluation of the Hashing The hashing was evaluated to determine its ability to differentiate between real images and anomalous images. The following are the evaluation terms used for the experimentation: Total images (T_{IMG}) - the total number of tested images, True positive (T_p) - properly labeled real images, True negative (T_N) - properly labeled manipulated image, False negative (F_N) - incorrectly labeled manipulated images, the manipulated images that have been proven to be real images, False positive (F_P) - incorrectly labeled real images, the real image that have been proven to be fake images. Thus, the accuracy ($\frac{T_p+T_N}{T_{IMG}} \times 100$), recall ($\frac{T_p}{T_p+F_N}$), and precision ($\frac{T_p}{T_p+F_P}$) are calculated to evaluate the comparative performance of hashing at different threshold levels to set the benchmark for the hash function. Table 2 represents the result of the evaluation of the hashing algorithm for accuracy, recall, and precision.

Threshold	T_p	F_N	F_p	T_N	Recall (%)	Precision (%)	Accuracy (%)
0.0001	2376	870	0	0	73.20	1.0	73.20
0.0002	2375	870	1	0	73.19	99.96	73.17
0.0003	2375	869	1	1	73.21	99.96	73.20

Table 2. Hashing Evaluation

4 Experimental Design

The experimental procedure taken to carry out the model formulation and prediction is explained in this section.

4.1 Deep Learning Experiment

After the dataset preparation as explained in Section 3, in order to predict a tampered image, CNN was used to train the dataset of real and tampered images. The process followed to achieve this is described in the following sections.

Pre-processing Since the real and tampered images were of varying sizes, the first step in data pre-processing was to resize them. This strikes a balance between offering sufficient resolution for real and manipulated image detection by the machine learning algorithm and for efficient training. To meet ImageNet[2] requirements, all images were normalized. For the purpose of standardizing the experiments, the same image size used in the experiment with hashing is deployed in the experiment without hashing. Hence, the image size used in our work is 150 by 150 pixels, and those images are converted to RGB color space. For the experiment that uses the hashing dataset, the derived hash value from the image dataset from Section 3 was deployed. The length of the hash value was established, and this information was used to modify the input tensor's shape. This entails altering CNN's Keras input layer using the Input() function.

Splitting of Dataset The training, validation, and testing sets were created from the data by splitting the dataset. The validation set is used to modify the hyperparameters, the testing set is used to assess how well the trained model performed, and the training set is used to train the network. The ratio of training to testing dataset used was 80:20.

Data augmentation If more data are generated, the model will often be more robust and minimize overfitting. A huge dataset is necessary for the algorithm used for deep learning to function effectively. Data augmentation was used on the training image dataset to increase the model's performance. Deep learning systems typically include a library of tools for enhancing the data. Rescale, horizontal flip, and vertical flip was used to create fresh training sets. The image is rotated right or left on a plane within 1° and 360° to perform rotation augmentations. The rotating degree parameter has a significant impact on the reliability of rotation augmentations. Images can be transformed to capture extra information about things of interest by shifting and flipping them.

[2] ImageNet - https://www.image-net.org/

CNN Architecture The CNN architecture was defined for the training image dataset for the purpose of classification. We built a model with six convolutional layers, followed by a fully connected hidden layer. In order to output a probability for each class, the output layer utilizes SoftMax activation. The Adam optimization algorithm and a rate of learning were employed to optimize the network. During the training process, the model iterates across batches of the dataset for training, for each batch size. Gradients are then calculated for each batch, and the weighting of the network is automatically updated. Typically, training is continued until the algorithm converges. A checkpoint was employed to save the model with the highest level of validation accuracy. This is helpful because, after a given number of epochs, the network can begin to overfit. The Keras library's callback functionality was used to enable this functionality. A sequence of functions known as a callback is used at specific points in the training process, such as the conclusion of an epoch. Model check-pointing and learning rate scheduling are both built-in features of Keras.

For the experiment with the hashing dataset, to create a CNN architecture that can take a training image dataset with hash codes as an input for classification, CNN's input shape was defined to match the shape of the hash codes the hashing function produces. As a result, the input layer was shaped in a way that matches the length of the hash codes. The CNN's convolutional layers were made to work with hash codes. This was accomplished by modifying the convolutional operation's stride, number of filters, and filter size to take the lower input size into account. In general, the CNN's architecture was selected to capture the pertinent characteristics of the hashed data. The size of the pooling windows was carefully chosen to prevent losing too much information from the hashed data because the pooling layers of the CNN can be used to down sample the feature maps produced by the convolutional layers. The CNN's output layer was created to deliver the results required for the task at hand. For instance, as the assignment involved classifying images, it was desirable to set the output layer to be a Sigmoid layer with as many nodes as there were classes in the dataset.

The model was created with four convolutional layers, followed by a fully connected hidden layer. The SHA-256 hash codes, which have a length of 64, were defined to have a shape that matches the input shape of the CNN. This indicates that the input layer's shape should be (64, 1). The CNN's convolutional layers were made to work with SHA-256 hash codes. One strategy is to employ a 1D convolutional layer with a modest kernel size, like 3 or 5 since the input contains binary data. A limited number of filters, such as 16 or 32, may be present in this layer. To decrease the spatial dimensions of the feature maps while retaining the most crucial features, we employed a max pooling layer with a pool size of 2. The CNN's output layer was created to deliver the results required for the task at hand. The output layer was made up of a single node with a sigmoid activation function as our objective was binary classification. As explained earlier, a checkpoint was used to preserve the model with the best validation accuracy.

Training and Validating the Models Using backpropagation and the Adam optimizer, the dataset was trained using the defined CNN architecture. The training of model involves setting the model parameters, such as the learning rate and the number of iterations, and then using an optimization algorithm, such as Adam, to adjust the parameters and minimize the error between the predicted and actual values. In the same vein, model validation was carried out after the model had been trained to evaluate its performance on the validation data. This involved measuring metrics such as accuracy, precision, recall, and F1 score. The model was trained and retrained with different parameters to optimize the model's performance on the validation data until the model's performance was satisfactory.

5 Results and Discussion

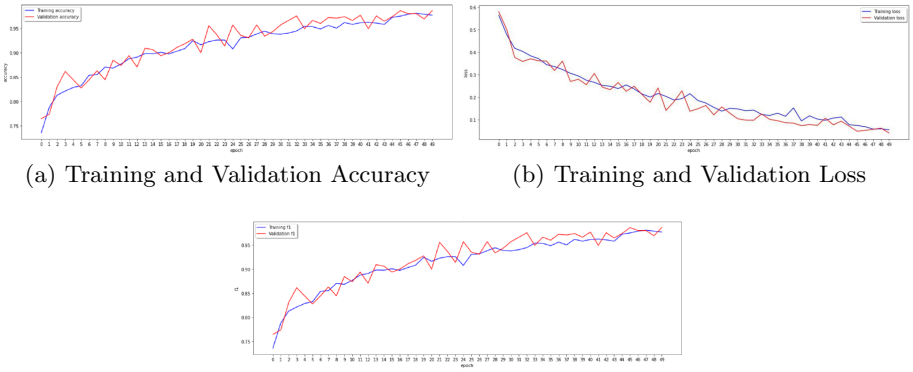
This section focuses on presenting the results obtained from the implementation and comparing them in a progressive manner as per the research methodology.

5.1 Deep Learning without Hashing

The model was trained and verified until the highest possible training and validation accuracy was achieved. The network's learning curve throughout training and validation is depicted in Fig. 1(a). In the training and validation phases, accuracy both increased, reaching 97.68% training accuracy and 98.62% validation accuracy, as can be observed. Also, the training and validation losses were 0.0548 and 0.0404, respectively. Fig. 1(b) represents the network's loss curve during training and validation, while Fig. 1(c) represents the network's F1 curve during training and validation. The result shows that both validation loss and training loss decreased with an increasing number of iterations, which demonstrated the model's effectiveness.

Evaluation of the Model For the testing of the model, the testing dataset comprising the same image features used in the training phase but different from the images that were used in original dataset for training, are used to test the model. By loading the test images dataset and identifying the classes utilizing the model, the models were tested. Through the use of the predict_classes() function, it was possible to determine whether an image was real or artificially altered based on whether it represented 1 or 0. The model shows the prediction accuracy, precision, recall, F-score, and loss to be 89.08%, 0.88.89%, 89.07%, 88.94%, and 34.7%, respectively.

Fig. 2 represents the result of the confusion matrix. From the confusion matrix we can gain better understanding of the classifiers. It can be seen from Fig. 2 that the classifier classifies 133 (76%) of the real instances correctly and 42 (24%) of the real instances incorrectly. However for the tampered instances, it classifies 446 (94%) of those instances correctly and 29 (6%) incorrectly.



(a) Training and Validation Accuracy

(b) Training and Validation Loss

(c) Training and Validation F1 Score

Fig. 1. Training and Validation Performance

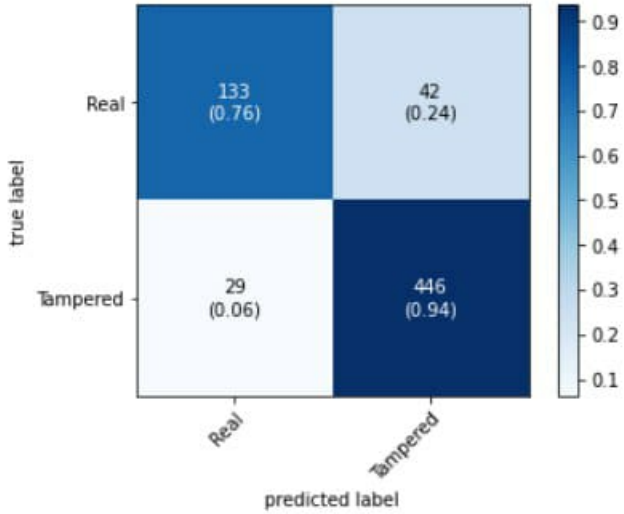
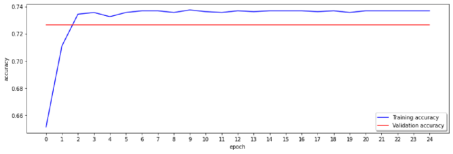


Fig. 2. Performance Evaluation of Image Prediction

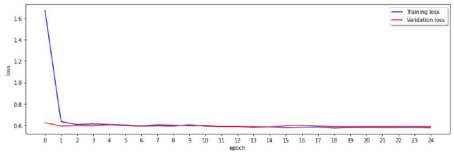
5.2 Deep Learning with Cryptography

To obtain the desired accuracy during training and validation, the model was trained and tested. Figs. 3(a), 3(b) and 3(c) represent the network's learning curve, loss curve and F1 curve for the hash image dataset respectively.

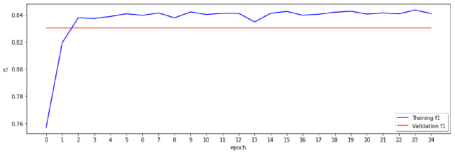
It can be seen that pattern of line on validation training, loss and F1 for the CNN learning model shows a straight-line pattern. A straight line for the validation loss or accuracy indicates that the model is not improving its performance on the validation set even though the training loss is still decreasing. This can mean that the model has already learned everything it can from the training



(a) Training and Validation Accuracy for Hashing



(b) Training and Validation Loss for Hashing



(c) Training and Validation F1 Score for Hashing

Fig. 3. Training and Validation Performance for Hashing

data and is overfitting, but it also means that it is unable to generalize to new data. Overfitting occurs when a model is too complicated and starts to fit the noise in the training data instead of the underlying patterns that generalize to new data. The model may thus begin to perform well on the training data but poorly on the validation data.

To find a solution to the problem of overfitting, an attempt was made to simplify the model using different approaches, such as lowering the number of layers or nodes, including regularization strategies, or expanding the validation set. However, all the possible attempts to resolve the overfitting problem did not result in a significant improvement in the model. At epoch 20, the highest training and validation accuracy that could be achieved were 73.69% and 72.66%, while the result of the testing and evaluation were 72.70% and 83.90% for accuracy and F1 score, respectively.

5.3 Results Evaluation

Going by the result of the hashing evaluation, which showed 73.20% accuracy, 99.96% precision, and 73.21% recall at 0.0003 threshold, which is surprisingly within the same range as the result of the deep learning algorithm, indicates that a more robust hashing algorithm that ensures high euclidean distance is required to achieve high prediction performance. This implies that to use cryptography to improve the integrity of image prediction in deep learning, more effort is needed at the hashing phase to ensure the accuracy of prediction.

In the overall assessment of our result, the hashing algorithm at the threshold 0.0003 gives 73.20% image prediction accuracy. The use of the CNN algorithm on the hashing image dataset of 3247 at epoch 20 gives a prediction accuracy of 72.70% at 0.09s. In the same vein, the result of CNN on the raw image dataset

at epoch 50 gives a prediction accuracy of 89.08% at 2s. The result shows the high capacity of deep learning prediction accuracy on the raw image without hashing. However, the prediction on hashing image dataset is faster on CNN, though with less accuracy.

6 Conclusion

The advancement in technological and image generation techniques have provided the possibility to create fake images. Many approaches have been developed to predict fake images using different approaches such as through the use of cryptography and deep learning algorithms. Despite many advantages in deep learning algorithms to predict fake images such as automatic feature engineering, parameter sharing and dimensionality reduction, one of the deep learning drawbacks emanates from parsing of bad examples to deep learning models.

This work investigated applying cryptography - hashing to an image based on the RGB channel and how this could be used to improve the integrity of the image prediction as fake or real by deep learning. The hashing was performed on the RGB channel of the image dataset of both real and manipulated images using SHA-256 hash function and the euclidean distance was used to differentiate between the real and manipulated images. The lower the hash value the higher the possibility of the image to be classified as fake while the higher the hash value the higher the possibility of the image to real. Based on this, a threshold was set to determine the accuracy of the hashing to predict fake image. The generated hash value of the image dataset was parsed to a deep learning model - CNN algorithm for image prediction. The raw image dataset without hashing was also parsed into CNN algorithm for prediction for the purpose of comparison.

The result of our experimental analysis shows the viability of hashing to improve the integrity of images that a deep learning model will predict provided a robust hashing algorithm with a very high accuracy is used. This will go a long way in reducing the adversarial attack on CNN through the feeding of bad examples into deep learning networks and make prediction at lesser time. Hence, using cryptography on an image before deep learning can guarantee the integrity of the image being predicted, which can be done at a lesser time. However, the cryptography algorithm to be applied must be robust enough to give high euclidean distance between real and manipulated images to guarantee high accuracy.

For future work, it is desirable to investigate more hashing algorithms and their prediction accuracy on deep learning models with a view to establish which hashing algorithm has better performance for the purpose of integrity of image prediction. Furthermore, more data-sets could also be used to test the efficacy of the developed model.

References

1. Ali, N.H.M., Mahdi, M.E.: Detecting similarity in color images based on perceptual image hash algorithm. IOP Conference Series: Materials Science and Engineering

737(1), 012244 (feb 2020). https://doi.org/10.1088/1757-899X/737/1/012244
2. Ali, S.S., Baghel, V.S., Ganapathi, I.I., Prakash, S.: Robust biometric authentication system with a secure user template. Image and Vision Computing **104**, 104004 (2020). https://doi.org/https://doi.org/10.1016/j.imavis.2020.104004
3. Ali, S.S., Ganapathi, I.I., Vu, N.S., Ali, S.D., Saxena, N., Werghi, N.: Image forgery detection using deep learning by recompressing images. Electronics **11**(3) (2022). https://doi.org/https://doi.org/10.3390/electronics11030403
4. Azure: Algorithm component referencefor azure machine learning designer. Python SDK Azure-Ai-Ml V2 (2023)
5. Bi, X., Liu, Y., Xiao, B., Li, W., Pun, C., Wang, G., Gao, X.: D-unet: A dual-encoder u-net for image splicing forgery detection and localization. CoRR **abs/2012.01821** (2020). https://doi.org/https://doi.org/10.48550/arXiv.2012.01821
6. Bondi, L., Lameri, S., Güera, D., Bestagini, P., Delp, E.J., Tubaro, S.: Tampering detection and localization through clustering of camera-based cnn features. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1855–1864 (2017). https://doi.org/10.1109/CVPRW.2017.232
7. Bunk, J., Bappy, J.H., Mohammed, T., Nataraj, L., Flenner, A., Manjunath, B., Chandrasekaran, S., Roy-Chowdhury, A.K., Peterson, L.: Detection and localization of image forgeries using resampling features and deep learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1881–1889. IEEE Computer Society, Los Alamitos, CA, USA (jul 2017). https://doi.org/10.1109/CVPRW.2017.235
8. Chaitra, B., Reddy, P.B.: A study on digital image forgery techniques and its detection. In: 2019 International Conference on contemporary Computing and Informatics (IC3I). pp. 127–130 (2019). https://doi.org/10.1109/IC3I46837.2019.9055573
9. Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E.: An evaluation of popular copy-move forgery detection approaches. IEEE Transactions on information forensics and security **7**(6), 1841–1854 (2012). https://doi.org/https://doi.org/10.1109/TIFS.2012.2218597
10. Devi Mahalakshmi, S., Vijayalakshmi, K., Priyadharsini, S.: Digital image forgery detection and estimation by exploring basic image manipulations. Digital Investigation **8**(3), 215–225 (2012). https://doi.org/https://doi.org/10.1016/j.diin.2011.06.004
11. Easow, S., Manikandan, D.L.C.: A study on image forgery detection techniques. International Journal of Computer (IJC) **33**(1), 84–81 (May 2019), https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/1411
12. Feng, W., Wu, S., Li, X., Kunkle, K.: A deep belief network based machine learning system for risky host detection. CoRR **abs/1801.00025** (2018), https://doi.org/10.48550/arXiv.1801.00025
13. Gadamsetty, S., Ch, R., Ch, A., Iwendi, C., Gadekallu, T.R.: Hash-based deep learning approach for remote sensing satellite imagery detection. Water **14**(5) (2022). https://doi.org/https://doi.org/10.3390/w14050707
14. García, R., Algredo-Badillo, I., Morales-Sandoval, M., Feregrino-Uribe, C., Cumplido, R.: A compact fpga-based processor for the secure hash algorithm sha-256. Computers Electrical Engineering **40**(1), 194–202 (2014). https://doi.org/https://doi.org/10.1016/j.compeleceng.2013.11.014, 40th-year commemorative issue
15. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., De, D.: Fundamental Concepts of Convolutional Neural Network, pp. 519–567. Springer International Publishing, Cham (2020). https://doi.org/https://doi.org/10.1007/978-3-030-32644-9_36

16. Guan, Y., Li, S.E., Duan, J., Li, J., Ren, Y., Sun, Q., Cheng, B.: Direct and indirect reinforcement learning. International Journal of Intelligent Systems **36**(8), 4439–4467 (2021). https://doi.org/https://doi.org/10.1002/int.22466
17. Habibi, M., Hassanpour, H.: Splicing image forgery detection and localization based on color edge inconsistency using statistical dispersion measures. International Journal of Engineering **34**(2), 443–451 (2021). https://doi.org/10.5829/IJE.2021.34.02B.16
18. Islam, A., Long, C., Basharat, A., Hoogs, A.: Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4675–4684 (2020). https://doi.org/10.1109/CVPR42600.2020.00473
19. Karampidis, K., Papadourakis, G.: File type identification for digital forensics. In: Krogstie, J., Mouratidis, H., Su, J. (eds.) Advanced Information Systems Engineering Workshops. pp. 266–274. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-39564-7$_2$5
20. Kester, Q.A., Nana, L., Pascu, A.C., Gire, S., Eghan, J.M., Quaynor, N.N.: A hybrid image cryptographic and spatial digital watermarking encryption technique for security and authentication of digital images. In: 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim). pp. 322–326 (2015). https://doi.org/10.1109/UKSim.2015.85
21. Kester, Q.A., Nana, L., Pascu, A.C., Gire, S., Eghan, J.M., Quaynor, N.N.: A novel hybrid discrete cosine transformation and visual cryptographic technique for securing digital images. In: 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim). pp. 327–332 (2015). https://doi.org/10.1109/UKSim.2015.101
22. Kumar, M., Soni, A., Shekhawat, A.R.S., Rawat, A.: Enhanced digital image and text data security using hybrid model of lsb steganography and aes cryptography technique. In: 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS). pp. 1453–1457 (2022). https://doi.org/10.1109/ICAIS53314.2022.9742942
23. Kuznetsov, A.: Digital image forgery detection using deep learning approach. Journal of Physics: Conference Series **1368**(3), 032028 (nov 2019). https://doi.org/10.1088/1742-6596/1368/3/032028
24. Kwon, M.J., Yu, I.J., Nam, S.H., Lee, H.K.: Cat-net: Compression artifact tracing network for detection and localization of image splicing. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 375–384 (2021). https://doi.org/10.1109/WACV48630.2021.00042
25. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. CoRR **abs/1811.00656** (2018), https://doi.org/10.48550/arXiv.1811.00656
26. Liu, X., Liu, Y., Chen, J., Liu, X.: Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. IEEE Trans. Cir. and Sys. for Video Technol. **32**(11), 7505–7517 (nov 2022). https://doi.org/10.1109/TCSVT.2022.3189545
27. Mahdian, B., Saic, S.: Blind methods for detecting image fakery. In: 2008 42nd Annual IEEE International Carnahan Conference on Security Technology. pp. 280–286 (2008). https://doi.org/10.1109/CCST.2008.4751315
28. Matern, F., Riess, C., Stamminger, M.: Gradient-based illumination description for image forgery detection. IEEE Transactions on Information Forensics and Security **15**, 1303–1317 (2020). https://doi.org/10.1109/TIFS.2019.2935913
29. Michelucci, U.: An introduction to autoencoders. CoRR **abs/2201.03898** (2022), https://doi.org/10.48550/arXiv.2201.03898

30. Okeyinka, A., Alao, O., Gbadamosi, B., Ogundokun, R., Oluwaseun, R.: Application of sha-256 in formulation of digital signatures of rsa and elgamal cryptosystems pp. 61–66 (01 2018), https://api.semanticscholar.org/CorpusID:195800765
31. Pierluigi, P.: Photo forensics: Detect photoshop manipulation with error level analysis. (2023), https://resources.infosecinstitute.com/topic/error-level-analysis-detect-image-manipulation/
32. Raja, A.: Active and passive detection of image forgery: A review analysis. IJERT-Proc **9**(5), 418–424 (2021), https://www.ijert.org/research/active-and-passive-detection-of-image-forgery-a-review-analysis
33. Ravi, J., Durga, M.G.S., Kartheek, Y.D.R.C., Begum, M.S., Raju, T., Raju, T.V.S.: Image Fusion using Non Subsampled Contourlet Transform in Medical Field. International Journal of Engineering and Advanced Technology (IJEAT) **9**(3), 3829–3832 (Feb 2020). https://doi.org/10.35940/ijeat.C6268.029320
34. Salehinejad, H., Baarbe, J., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. CoRR **abs/1801.01078** (2018), https://doi.org/10.48550/arXiv.1801.01078
35. Sharma, P., Kumar, M., Sharma, H.: Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation. Multimedia Tools and Applications **82**(12), 18117–18150 (2023). https://doi.org/10.1007/s11042-022-13808-w
36. Shinde, P.P., Shah, S.: A review of machine learning and deep learning applications. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). pp. 1–6 (2018). https://doi.org/10.1109/ICCUBEA.2018.8697857
37. Singh, A., Singh, J.: Image forgery detection using deep neural network. In: 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN). pp. 504–509 (2021). https://doi.org/10.1109/SPIN52536.2021.9565953
38. Stanton, J., Hirakawa, K., McCloskey, S.: Detecting image forgery based on color phenomenology. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019), https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=dayton15574119887572&disposition=inline
39. Tang, Z., Li, X., Zhang, X., Zhang, S., Dai, Y.: Image hashing with color vector angle. Neurocomputing **308**, 147–158 (2018). https://doi.org/10.1016/j.neucom.2018.04.057
40. Verdoliva, L.: Media forensics and deepfakes: An overview. IEEE Journal of Selected Topics in Signal Processing **14**(5), 910–932 (2020). https://doi.org/10.1109/JSTSP.2020.3002101
41. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9535–9544 (2019). https://doi.org/10.1109/CVPR.2019.00977
42. Yousfi, Y., Fridrich, J.: An intriguing struggle of cnns in jpeg steganalysis and the onehot solution. IEEE Signal Processing Letters **27**, 830–834 (2020). https://doi.org/10.1109/LSP.2020.2993959