An Analysis of Image Splicing Detection using Convolutional Neural Network (CNN)

Nor Bakiah Abd Warif, *IEEE Member*
Centre for Information Security Research, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia.
norbakiah@uthm.edu.my

Muhammad Faiz Hamdan
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia.
faizhamdan16@gmail.com

Mohd. Faizal Ab. Razak
Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia.
faizalrazak@umpsa.edu.my

Salwana Mohamad @ Asmara
Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia.
salwanamohamad@umpsa.edu.my

Ahmad Firdaus
Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang, Malaysia.
firdausza@umpsa.edu.my

2024 1st International Conference on Cyber Security and Computing (CyberComp) | 979-8-3503-8772-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/CYBERCOMP60759.2024.10913841

***Abstract*— Image splicing is the act of manipulating digital images by copying and pasting pixels from one image onto another, and it can be difficult to detect due to the existence of many editing software. Recently, deep learning approaches have been used to effectively detect image splicing forgery because they learned high-level features from images that are not visible to naked eye. This research aims to evaluate the use of Convolutional Neural Network (CNN) as an image splicing detection method against CASIA V1, Columbia Gray and Columbia Color datasets. The analysis shows that CNN is able to detect 83% accuracy in Columbia Gray, 75% accuracy in CASIA V1, and 69% accuracy in Columbia Color. These results have led to the highest performance, specifically for fake images, which are 96%, 85% and 88%, for each dataset, respectively. This means that the CNN able to detect image splicing correctly, while having difficulties identifying an authentic or real image.**

Keywords— Deep learning, Image splicing, Neural Network, Image Forgery Detection

I. Introduction

Image forgery is the manipulation of digital images to hide or change information. It has become more common with the growth of technology and the availability of easy-to-use editing software like Adobe Photoshop. This allowed people to manipulate images easily, effortlessly and for the shortest period. Image forgery can lead to misunderstandings and misinformation, especially when the fake images are being used as evidence in court. To prevent these negative consequences, image forgery detection has been developed to ensure the authenticity of digital images and allow them to be accepted as evidence in legal proceedings.

An image splicing is a technique that involves combining two or more images to create a fake image without removing any elements from the original image. This technique is being studied more than copy-move and image morphing. Image splicing involves removing or changing elements in the original image because it requires two or more images rather than copy-move and image morphing which uses only one image. Alahmadi et al. [1] proposed a forgery detection method based on Local Binary Pattern (LBP). However, it has some limitations such as not effectively capturing the distinctive features or patterns present in different images and sensitive to noise.

To overcome this problem, this research designs an algorithm for image splicing detection by using Convolutional Neural Network (CNN) model. CNN is being used because it learns input images without losing information and has been widely used in many applications without relying on human skills. Three image splicing datasets that are used for evaluations are CASIA V1.0 which is from Chinese Academy of Science, and Columbia Gray and Columbia Color which are from Columbia University.

The rest of the paper is organized as follows: Section II explains the related works on image forgery and splicing detection. The explanation of the CNN model is given in Section III. The experiment results and discussions are presented in Section IV and the conclusion is drawn in Section V.

II. Literature Review

A. Related Works

Image splicing is a common method of image forgery that involves combining two or more images into a single image. This is typically achieved by cropping a region from one image and pasting it onto another image. While this method is relatively simple compared to others, such as copy-move forgery, it does require the use of at least two images. Image splicing is often used in digital photomontage to create a single image from multiple sources using tools like Adobe Photoshop [2]. Despite its simplicity, image splicing can be difficult to detect and can have serious consequences if used to manipulate information or mislead others.

Deep learning approaches have been widely used for image forgery detection due to their ability to extract features directly from raw input such as images and provide a novel approach to identifying tampered regions [3]. The use of deep learning in image forgery detection is growing rapidly due to its high accuracy and convincing results. Several deep learning algorithms are commonly used for image forgery detection, including Recurrent Neural Networks (RNNs), Feedforward Neural Networks (FFNNs), and Convolutional Neural Networks (CNNs) [4]. Among these, CNNs have been particularly successful for medical image analysis.

CNNs are the most commonly used neural networks for image forgery detection due to their specialization in image processing and ability to learn the characteristics of images without human involvement [5]. CNNs consist of several layers that transform the input using convolution filters [4]. These networks can be used to detect various types of image forgery, including copy-move forgery.

B. Comparison Methods

Table I list selected three authors that might be related to this research. Zhang et al. [6] first applied deep learning based on Stacked Auto Encoders (SAE) to extract features from the images, which are then fed into conventional neighborhood classification using non-overlapping matching block. It learns to identify patterns in the features extracted by the SAE that are indicative of tampering. They found that their method was able to detect tampering with high accuracy even if the dataset is combined with copy-move forgery.

TABLE I. COMPARISON THREE RELATED PAPERS

Detection Method	Type of forgery	Evaluation
Zhang et al.[6] • Stacked Auto Encoders (SAE) • Contextual neighborhood classification	Splicing & Copy-Move	CASIA V1 Accuracy = 91.09% CASIA V2 Accuracy = 87.51% Columbia Gray Accuracy = 81.91%
Kuznetsov [7] • CNN • VGG-16	Splicing & Copy-Move	CASIA V2 Accuracy = 96.4%
El-Latif et al. [8] • Haar Wavelet Transform • CNN	Splicing & Copy-Move	CASIA V1 Accuracy = 94.55% CASIA V2 Accuracy = 96.36%

Kuznetsov [7] proposed a new method for detecting image forgeries using CNN. They made several changes to the way the CNN processes information, including changes to the first convolutional layer and the way the network handles duplicated information. They proposed VGG16 to improve the robustness of the network while identifying forged or genuine images. However, they only test their method with CASIA V2 dataset which we know is combined with copy-move forgery.

El-Latif et al. [8] combined CNN with the Haar wavelet transform. The proposed method consists of two steps. First, the Haar wavelet transform is applied to the input image to extract features. The Haar wavelet transform is a type of mathematical transformation that can be used to analyze the frequency content of an image. Next, the frequency feature obtained from Haar will be trained by CNN on a dataset of images that have been manually labeled as spliced or non-spliced. Their experimentation results proved to be a high accuracy to both CASIA datasets.

III. METHODOLOGY

In this section, the research methodology will be explained from phase to phase. Fig. 1 shows the research methodology of this research.

Fig. 1 Research Methodology

A. Analysis

In the analysis phase, existing techniques that is related to image splicing have been studied. Related papers including paper using CNN is explained in the previous subsection. Comparison is also been discussed to compare the techniques, datasets, and also evaluation metrics.

B. Data Collection

For the data collection phase, three datasets are gathered. The datasets are CASIA V1.0 from Chinese Academy of Science [9] and Columbia Gray and Color from Columbia University [10]. The total number of images in CASIA V1.0 are 1721 images that includes 800 authentic images and 921 spliced images. Meanwhile, Columbia Gray contains 933 authentic images and 912 tampered images with a total of 1845 images. Lastly, Columbia Color contains 183 real images while 180 spliced images. Each dataset contains different types of files. CASIA V1 images are in jpg and jpeg format with 384×256 sizes, Columbia Gray contains bmp image files with grayscale color while Columbia Color contains tiff image format with 128×128 sizes. Table 2 summarizes the data characteristics.

TABLE II. SUMMARY OF DATASET CHARACTERISTICS

Dataset	Total images	Format images	Size of images
CASIA V1	1721 800 authentic 921 spliced	Jpg, jpeg	384×256
Columbia Gray	1845 933 authentic 912 tampered	bmp	128×128
Columbia Color	363 183 real 180 spliced	tiff	128×128

C. Development

The development phase is where the CNN algorithm is being implemented. The framework was adopted from Kim's paper [11]. The CNN architecture includes convolutional, pooling, fully connected, and output layers are displayed in Fig. 2 while the framework is shown in Fig. 3.

Raw images will be inserted into the algorithm for testing. The images for input are from the datasets that have been listed in the previous subsection. The algorithm is developed using Python programming language in the Jupyter Notebook platform. As stated before, the total images from all datasets are 1721, 1845 and 363 images respectively.

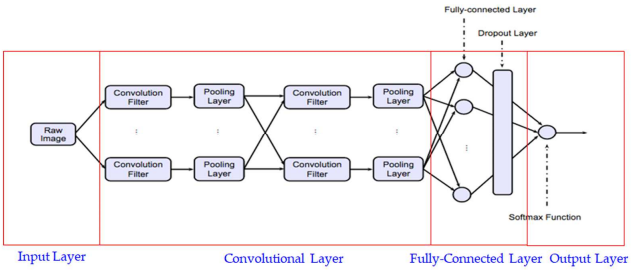

Fig. 2 Common CNN architecture [11]

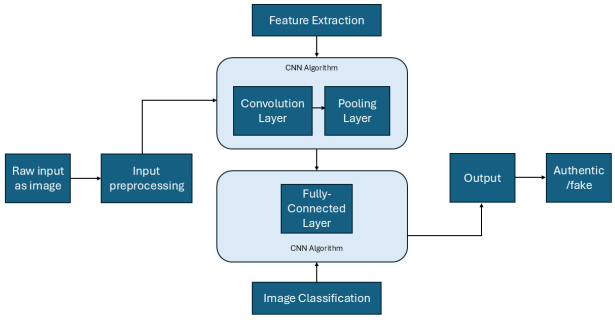

Fig. 3 Framework of the CNN

1) Image Pre-processing and Feature Extraction

Image preprocessing is a crucial step in deep learning-based image forgery detection. It involves enhancing image quality, extracting relevant features, and ensuring consistent image characteristics before analysis. For this research, the preprocessing phase includes importing the dataset, converting images to Error Level Analysis (ELA), and resizing them to 128×128 pixels. ELA detects image manipulation by compressing the image and comparing differences between the original and compressed versions. In grayscale ELA, variations in intensity indicate potential tampering. In RGB ELA, each color channel is analyzed independently, detecting inconsistencies in color distribution. Grayscale ELA provides a general analysis of brightness and texture changes, while RGB ELA offers more detailed information about color-specific modifications or tampering.

Then, features from the images will be extracted using CNN. The layers involved in these processes are Convolutional Layer and Pooling Layer. The feature extraction process in a CNN typically begins with the convolutional layers. These layers use a set of filters or kernels that are convolved with the input image to extract local patterns or features from the image. These filters can be thought of as a kind of feature extractor that is able to detect specific patterns in the image, such as edges or textures. After the convolutional layers, the feature maps are often passed through a pooling layer to reduce the spatial dimension of the feature maps, while maintaining the most important information. This is achieved by applying a pooling operation, such as max pooling, which takes the maximum value from each pooling window and keeps only the max value in the pooling layer output. This helps reduce the computational cost and helps the model to focus on more important features.

2) Image Classification

Image classification is the process that happens at the final layer of CNN which is the Fully Connected Layer. The classification could only happen after the feature extraction which happened in the earlier layer, Convolutional Layer. After that, the pooling layer will reduce the spatial dimensions of the features extracted at Convolutional Layer. CNN could be pre-trained before it could be able to classify the images as either fake or authentic image.

In the convolutional layer, filters are applied to small sections of the input data to produce feature maps. These filters slide across the input, performing calculations by multiplying their weight with the corresponding values in the receptive field. The resulting values form a feature map, which represents the presence of specific features or patterns in the input data. Each filter in the convolutional layer generates its own feature map, enabling the network to detect and capture different meaningful features from the input.

Following the convolutional layers, the feature maps obtained undergo a pooling layer. This layer serves to reduce the spatial dimensions of the feature maps from the previous layer. Downsampling is achieved by summarizing or selecting essential information within small regions. The pooling layer helps to condense the representation of the features, retaining important information while reducing the overall spatial resolution.

After the pooling layer, the next stage is the dense layer, also known as the fully connected layer. This layer carries out high-level feature extraction and mapping by establishing connections between every neuron in the previous layer and each neuron in the current layer. This enables the network to capture complex relationships and interactions among the features. Dense layers are essential in learning global representations and making informed predictions as they integrate information from across the network and contribute to the overall decision-making process.

The last layer of the model is the output layer, which generates the model's prediction. The choice of activation function for the output layer depends on the specific task being performed. In this study, the 'SoftMax' function is utilized to adjust the output of CNN because the model is employed for multiclass classification. The function generates a probability distribution across the different classes, indicating the likelihood of each class being the correct prediction. Fig. 4 shows the model summary of parameters that are implemented.

D. Evaluation

Lastly, an evaluation of performance will be observed to discuss and analyze the ability of CNN against image splicing detection. For this research, accuracy will be calculated as the metric measure stated in Eq. 1. TP, TN, FP and FN in the formula represents True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positive (TP) is the sample labelled as real predicted as real while True Negative (TN) is the fake or spliced samples predicted as fake. Meanwhile, False Positive (FP) is the sample that is fake but labelled as real

Layer (type)	Output Shape	Param #
conv2d_13 (Conv2D)	(None, 124, 124, 64)	4864
max_pooling2d_13 (MaxPooling2D)	(None, 62, 62, 64)	0
dropout_17 (Dropout)	(None, 62, 62, 64)	0
conv2d_14 (Conv2D)	(None, 58, 58, 64)	102464
max_pooling2d_14 (MaxPooling2D)	(None, 29, 29, 64)	0
dropout_18 (Dropout)	(None, 29, 29, 64)	0
conv2d_15 (Conv2D)	(None, 25, 25, 64)	102464
max_pooling2d_15 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_19 (Dropout)	(None, 12, 12, 64)	0
flatten_4 (Flatten)	(None, 9216)	0
dense_8 (Dense)	(None, 256)	2359552
dropout_20 (Dropout)	(None, 256)	0
dense_9 (Dense)	(None, 2)	514

Total params: 2,569,858
Trainable params: 2,569,858
Non-trainable params: 0

Fig. 4 CNN model summary of parameters

and False Negative (FN) is the samples that are real but identified as fake.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (1)$$

IV. Result and Discussion

In this section, the results of CNN against image splicing is discussed. To conduct the experiment using the model, we selected and utilized three abovementioned datasets. Prior to feeding the datasets into the model, all images are underwent a pre-processing stage. It includes datasets separation into training and testing sets to ensure a fair comparison between them. We assigned 80% of the data for training and 20% for testing. Specifically, for CASIA V1, we trained the model with 1376 images and tested it with 345 images. Similarly, for Columbia Gray, we used 1476 images for training and 369 images for testing while only 290 images for training and 73 for validation are used in Columbia Color.

The experimental results are presented in Table 3, showcasing the accuracy values for each dataset. The experiments were conducted using a consistent number of epochs, specifically 30 epochs. According to the results , the validation accuracy achieved for CASIA V1 is 76%, Columbia Gray attained 83% while Columbia Color achieved 68%. The validation loss is also provided in the table. Since Columbia Color has the least amount of image, the loss percentage is highest compared to the other two datasets maybe due to not enough training features in provided images. Further analysis are discussed in the next subsections.

TABLE III. EXPERIMENTAL RESULTS FOR EACH DATASET

Metrics	CASIA V1	Columbia Gray	Columbia Color
Accuracy (%)	75.36	82.65	68.49
Loss (%)	49.81	41.08	59.55

A. CASIA V1

A total of 345 randomly selected and shuffled images were used for validation testing. The model achieved successful predictions by correctly identifying 171 fake images as fake. However, it incorrectly classified 25 fake images as real. Additionally, the model accurately identified 89 real images as real but misclassified 60 real images as fake. Fig. 5 presents the accuracy of the model when fed with a folder of fake or spliced images. The achieved accuracy in this scenario is 85%. This indicates that the model performed exceptionally well in correctly classifying the fake or spliced images. Fig. 6 shows random testing done on the model using random image picked from the real image dataset. The output shows the classification (real image), confidence (100%) and the image picked for the validation testing.

```
1/1 [==============================] - 0s 16ms/step
Class: fake
1/1 [==============================] - 0s 21ms/step
Class: fake
1/1 [==============================] - 0s 25ms/step
Class: fake
1/1 [==============================] - 0s 21ms/step
Class: fake
1/1 [==============================] - 0s 22ms/step
Class: fake
1/1 [==============================] - 0s 34ms/step
Class: fake
1/1 [==============================] - 0s 26ms/step
Class: fake
1/1 [==============================] - 0s 22ms/step
Class: real
1/1 [==============================] - 0s 25ms/step
Class: real
1/1 [==============================] - 0s 24ms/step

In [28]: print(f'Total: {total}, Correct: {correct}, Acc: {correct / total * 100.0}')
Total: 921, Correct: 782, Acc: 84.90770901194354
```

Fig. 5 Random testing on model using CASIA V1

```
1/1 [==============================] - 0s 68ms/step
Class: real Confidence: 100.00
```

Fig. 6 Random selected image results for CASIA V1

B. Columbia Gray

With a total of 369 randomly shuffled testing images, the model's predictions are as follows: it accurately identified 161 fake images as fake, while incorrectly classifying 25 fake images as real. Moreover, it correctly recognized 144 real images as real but misclassified 39 real images as fake. Fig. 7 displays the accuracy of the model when being fed a folder of spliced or fake images from the Columbia Gray dataset. This step evaluates the accuracy of the model in classifying image classes by itself. As can be seen, the model correctly classified 872 images as fake with an accuracy of 96%.

Fig. 8 displays a random testing done on the model. This time, an image from the spliced folder of Columbia Gray is picked. This step is taken to test the accuracy of the model on classifying spliced or fake images. The confidence of prediction (100%) is also printed followed by the image.

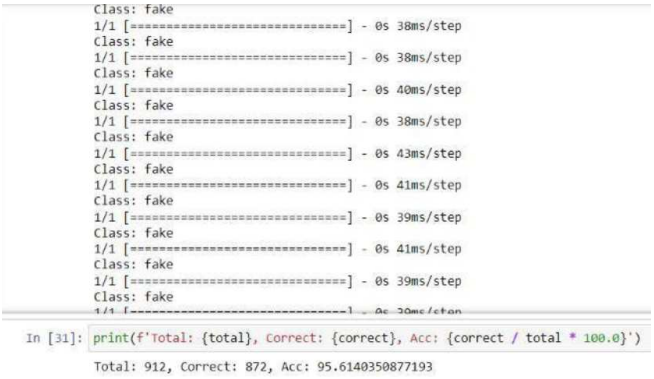

Fig. 7 Accuracy of classifying spliced images in Columbia Gray

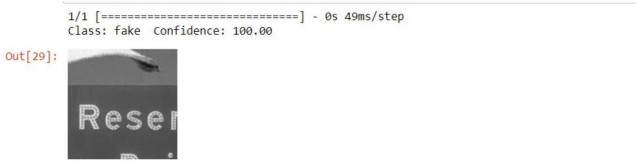

Fig. 8 Random selected image results for Columbia Gray

C. *Columbia Color*

With only 180 spliced images, the accuracy of the model in classifying fake or spliced images is still significantly high. During the validation testing, only 73 images are used. Fig. 9 shows the accuracy of the model in classifying image classes. From 180 images, the model classified 158 correctly as faked while predicted 22 as real.

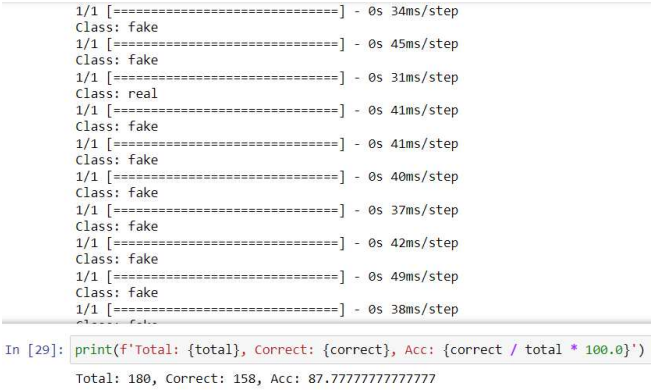

Fig. 9 Accuracy of classifying spliced images in Columbia Color

Fig. 10 shows the random testing of Columbia Color dataset. Even though the training and validation testing done on the dataset is the least, the model still managed to predict and classify the random image's class correctly with real confidence 85%.

D. *Discussion*

The results of the experiments and testing indicate that the model has effectively detected forgery within the fake datasets. Columbia Gray achieved the highest percentage of 96%, followed by Columbia Color with 88% and CASIA V1 with 85%. The detection on any random image also proved that the model has high confidence in recognizing not only fake image, but also authentic or real image.

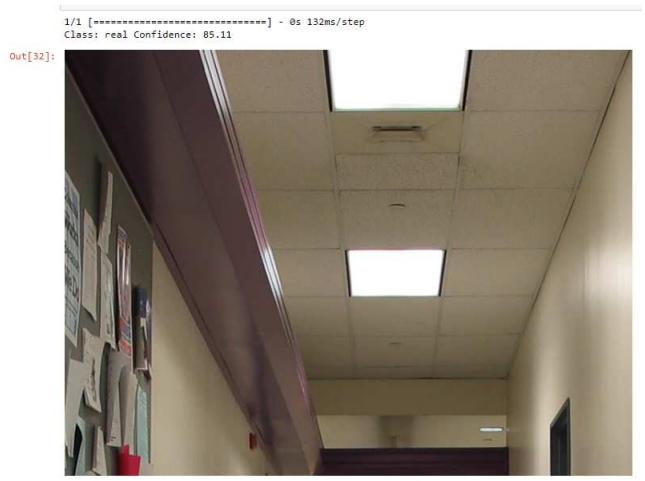

Fig. 10 Random testing on Columbia Color

However, upon feeding the model with authentic or real image folder, only Columbia Gray achieved significant accuracy which is 94% (Fig. 11) while CASIA V1 got 64% (Fig. 12) and 49% for Columbia Color (Fig.13). This shows that the model could detect spliced images better than real or authentic images. The reason of the highest performance in Columbia Gray might be because the size and total image in the dataset is significant for CNN to learn the features. Besides that, the grayscale image is also able to highlight an important feature well while learning the features in the model. We believe these findings have demonstrated a strong capability in CNN model by accurately identifying and distinguishing genuine and manipulated images.

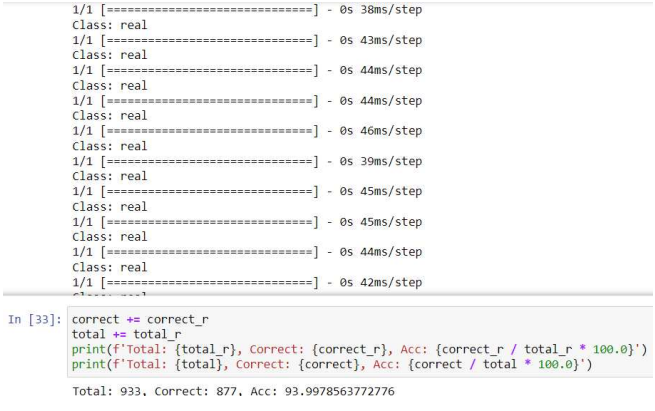

Fig. 11 Accuracy of authentic class in Columbia Gray dataset

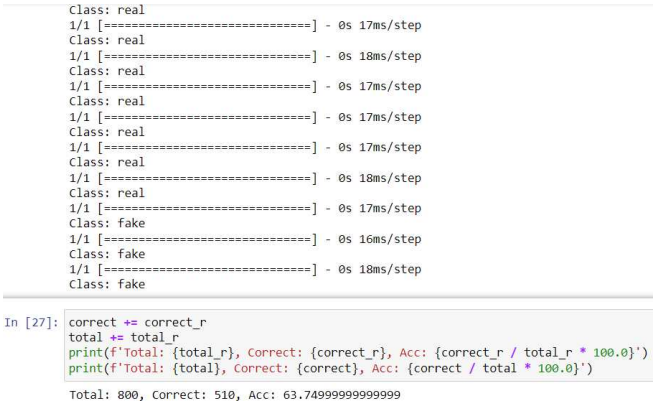

Fig. 12 Accuracy of authentic class in CASIA V1 dataset

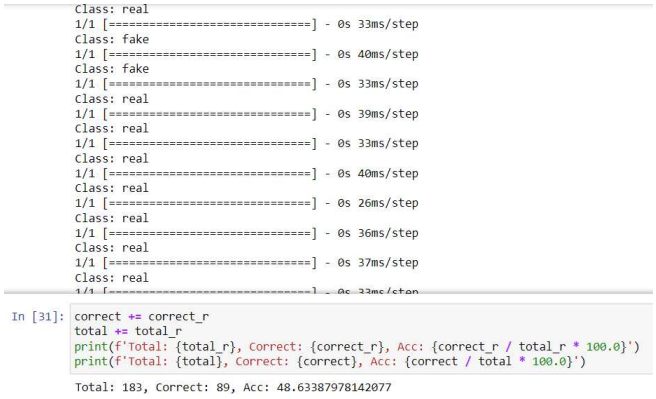

Fig. 13 Accuracy of real class in Columbia Color

V. CONCLUSION

The performance of the CNN model was assessed on three distinct datasets, comparing accuracy and loss. The findings of the study revealed that Columbia Gray exhibited higher accuracy compared to CASIA V1 and Columbia Color. The CASIA V1, Columbia Gray and Columbia Color datasets contained a relatively small number of images, totalling 1721, 1845 and 363 images respectively. Consequently, the training provided to the model was limited due to this insufficient amount of data. With larger datasets, better training outcomes could be achieved, potentially leading to higher accuracy in the predictions. This is proved in the evaluation done by Thiiban et al. [12] where high volume image achieved the highest accuracy.

In future studies, it is highly recommended to utilize larger datasets to conduct similar research. This would provide a more comprehensive and diverse range of images for training and testing the model, potentially leading to improved results. Additionally, researchers can explore applying the algorithm to detect other forgery techniques such as Copy-Move Forgery (CMF), Image Retouching, and various others. By expanding the scope to include different types of forgery, a more comprehensive understanding of the model's performance and its adaptability can be achieved. Furthermore, combining multiple algorithms or employing advanced techniques could be explored to enhance the overall detection accuracy and robustness of the model.

ACKNOWLEDGMENT

The authors would like to thank the Universiti Malaysia Pahang Al-Sultan Abdullah for providing financial support under Internal Research grant PPU230106.

REFERENCES

[1] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on DCT and Local Binary Pattern," in *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, 2013, pp. 253–256. doi: 10.1109/GlobalSIP.2013.6736863.

[2] S. Easow and L. C. Manikandan, "A Study on Image Forgery Detection Techniques," *International Journal of Computer*, vol. 33, no. 1, pp. 84-91, 2019.

[3] F. Hakimi, M. Hariri, and F. Gharehbaghi, "Image splicing forgery detection using local binary pattern and discrete wavelet transform," in *Conference Proceedings of 2015 2nd International Conference on Knowledge-Based Engineering and Innovation, KBEI 2015*, Institute of Electrical and Electronics Engineers Inc., Mar. 2016, pp. 1074–1077. doi: 10.1109/KBEI.2015.7436195.

[4] A. Kaur, Y. Singh, N. Neeru, L. Kaur, and A. Singh, "A Survey on Deep Learning Approaches to Medical Images and a Systematic Look up into Real-Time Object Detection," *Archives of Computational Methods in Engineering*, vol. 29, no. 4. Springer Science and Business Media B.V., pp. 2071–2111, Jun. 01, 2022. doi: 10.1007/s11831-021-09649-9.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems,* 2012, pp1097-1105.

[6] Y. Zhang, J. Goh, L. L. Win, and V. Thing, "Image region forgery detection: A deep learning approach," in *Cryptology and Information Security Series*, IOS Press, 2016, pp. 1–11. doi: 10.3233/978-1-61499-617-0-1.

[7] A. Kuznetsov, "Digital image forgery detection using deep learning approach," *J Phys Conf Ser*, vol. 1368, no. 3, p. 032028, Nov. 2019, doi: 10.1088/1742-6596/1368/3/032028.

[8] E. I. Abd El-Latif, A. Taha, and H. H. Zayed, "A Passive Approach for Detecting Image Splicing using Deep Learning and Haar Wavelet Transform," *International Journal of Computer Network and Information Security*, vol. 11, no. 5, pp. 28–35, May 2019, doi: 10.5815/ijcnis.2019.05.04.

[9] Dong, J., Wang, W., & Tan, T. (2013). CASIA Image Tampering Detection Evaluation Database. 2013 IEEE China Summit and International Conference on Signal and Information Processing, 422–426. https://doi.org/10.1109/ChinaSIP.2013.6625374

[10] Ng, T.-T., & Chang, S.-F. (2004). *Columbia Image Splicing Detection Evaluation Dataset.* [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm

[11] D.-H. Kim & H.-Y. Lee, "Image Manipulation Detection using Convolutional Neural Network," *International Journal of Applied Engineering Research*, vol. 12, no. 21, pp. 11640-11646, 2017.

[12] Thiiban , M., Nor, B. A. W, Ahsiah, I., and Noor, A. M. A., "An Evaluation of Convolutional Neural Network (CNN) Model for Copy-Move and Splicing Forgery Detection," *International Journal of Intelligent Systems and Application in Engineering*, vol. 11, no. 2, pp. 730-740, 2023.