Regular Article

A Deep Learning Model for Splicing Image Detection

Hoang Linh Nguyen[1], **Kha Tu Huynh**[2]

[1] International University, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Correspondence: Kha Tu Huynh, hktu@hcmiu.edu.vn
Communication: received 29 April 2023, revised 01 June 2023, accepted 15 June 2023
Online publication: 14 December 2023, Digital Object Identifier: 10.21553/rev-jec.335
The associate editor coordinating the review of this article and recommending it for publication was Prof. Vo Nguyen Quoc Bao.

Abstract– **With the advancement of digital technology, manipulating images has become relatively easy through many photo editing techniques. One of the techniques is the splicing which crops parts of images and puts them into another image creating a new composite image. The image splicing detection system is soon regarded as an exciting topic for many researchers to solve the problems of image forgery on the Internet, especially in social networks. ResNet-50 and VGG-16 are powerful architectures of convolutional neural networks, but they reveal many weaknesses when operating on low-end computers. The ultimate goal of this paper is to create a model for image splicing detection working well in limited memory machines. The study proposes the model, which is the improvement of VGG-16 applying residual network (ResNet). As a result, the proposed model achieves a test accuracy of 92.5% while the ResNet-50 gives an accuracy of 85.6% after 20 epochs of training 9,319 images from the CASIA v2.0 dataset which are used for forgery classification. The result proves the efficiency of the proposed model for image splicing detection, especially when working on low-end computers.**

Keywords– **ResNet, VGG-16, convolutional neural networks, image classification, splicing image detection.**

1 Introduction

In recent years, social networks have been incredibly developed and very popular with a huge number of users. Social networks allow people to exchange information, share knowledge, and express their feelings through posts. Posts are often shared in many different formats such as text, word, excel, pdf files, and especially images, one of the most effective forms of information because it shows the content more clearly. Images can summarize a long text, emphasize specific details, describe data more efficiently, or be represented as evidence of a case. However, many photos are edited such that they are different from the original for many purposes. One of edited images is the splicing image [1], created by merging parts of two or more different images to create a new composite image.

Using deep learning models is one of the highly effective methods for classifying splicing images. Deep learning plays a vital role in solving Computer Vision problems with the emergence of many applications that bring high efficiency in doing Computer Vision tasks, especially Convolutional neural networks (CNN) applied for Image classification. Convolutional neural networks help to extract image features or retrieve necessary information from the image input. Typical architectures for image classification are ResNet, VGG (VGG -16, VGG-19), AlexNet, LeNet, DenseNet, Inception (GoogLeNet), and ResNeXt. Among these architectures, Residual network (ResNet) [2] makes a significant contribution that allows people to extend the model layers deeper without the effect of vanishing gradient. ResNet can give high training accuracy with many variants such as ResNet-18, ResNet-34, ResNet-50, ResNet-152, etc. However, ResNet, which requires a large batch size, does not work well in low-end computers because the memory is restricted, and the batch size must be small to avoid the resource exhausted errors when training in those devices. The small batch size can affect the training results, such as increasing the error values due to Batch Normalization layers in ResNet architectures.

The purpose of the research is to implement a new system that can detect the forgery in splicing images and have the ability to work in low-end computers with high performance. To achieve that objective, the splicing and original images are classified by the proposed model using the residual network applied in the modified version of VGG-16 [3]. The images are also processed with ResNet-50 [4], and then the models' performances are compared.

The main contributions of this paper are organized in the following sections:

- Analyzing the problems of the traditional model in image splicing forgery detection for low-end computers.
- Proposing a model which is the improvement of VGG-16 applying residual network and showing the results of the implementation.

Literature review, problem statements, research method, proposed model, dataset, experimental results, discussion and conclusion will be presented in the following sections.

2 Literature Review

Splicing image detection has been a topic of interest for researchers due to the massive popularity of digital images in recent years. One of the traditional approaches for splicing detection is the Markov feature extractor in the discrete cosine transform (DCT) and discrete wavelet transform (DWT) domain [5]. The technique captures the intra-block and inter-block correlation between block DCT coefficients and three types of dependency among wavelet coefficients. Then it uses Support Vector Machine Recursive Feature Elimination (SVM-RFE) and Support Vector Machine (SVM) to classify the images as authentic or spliced using the final feature vector. The method is shown to be better than some existing methods. Bo Su et al. [6] suggest an improved Markov state selection approach that matches coefficients to Markov states based on a successful functional model. Experiments and analysis demonstrate that it can obtain a greater recognition rate and more effectively utilize the important underlying information in transformed coefficients. Another technique that improves the Run Length Run Number algorithm (RLRN) is suggested by Zahra Moghaddasi et al. [7]. The algorithm applies two-dimension reduction methods, principal component analysis (PCA) and kernel PCA, a method for image splicing forgery detection based on DCT coefficients and Markov features. In the research, authentic and spliced images can be distinguished using the support vector machine. The findings show that for R, G, B, Y channels and grayscale images, kernel PCA is a more efficient nonlinear dimension reduction technique.

Another efficient method for image splicing forgery detection based on the roughness measure algorithm, PCA algorithm, and SVM algorithm, is proposed by Zahra et al. [8]. The method divides the image into blocks and transforms them into DCT. Then singular value decomposition (SVD) is used to extract features from DCT and calculate the roughness measure. The technique also uses kernel principal component analysis to reduce the dimensionality of the features. Finally, the method classifies the images as authentic or spliced using SVM. An approach for image splicing forgery detection based on local binary pattern (LBP) and error level analysis (ELA) was suggested by Zhang et al. [9]. The hybrid technique is used to find the splicing area, and the result shows that the tampered area is accurately located. The method uses Bagged Trees to classify the images as authentic or spliced. Improving from traditional methods, one of the modern techniques in image splicing forgery detection is applying deep learning. S. Nath and R. Naskar [10] proposed a blind approach, which does not need the original image to detect splicing using a deep convolutional residual network and a fully connected classifier network. The experimental results demonstrate that the performance is excellent, with an average accuracy of 96%. However, the limitation of the method is that it just simply determines whether an image has been spliced or not but does not have the ability to localize the sliced regions in the image.

Jaiswal and Srivastava [11] also suggested an image-splicing forgery detection technique based on a deep learning approach using the pre-trained residual network ResNet-50. The input images are divided into two groups: one for training and one for testing. Each part goes through preprocessing to resize the photos to match the size of the pre-trained network. The pre-trained feature uses SVM, naive Bayes, and K-nearest Neighbors (KNN) as classifiers for classification. Hosny, K.M. et al. [12] proposed a new lightweight CNN model for detecting splicing images using a small number of parameters compared to other approaches. The model is fast, with only a few layers of convolution and pooling and achieves high accuracy when tested on CASIA 1.0, CASIA 2.0, and CUISDE datasets.

The published methods have good performance and have the ability to solve a specific problem, but they may still have some limitations. So, image splicing forgery detection can be continually researched with more improved algorithms or more developed models.

3 Problem Statements

This paper aims to build a new deep learning model for splicing images detection. Some popular deep learning models, such as ResNet-50 or VGG-16, can be considered to use. ResNet-50 is the practical model producing high accuracy and a low error rate compared to many models for image classification tasks. However, ResNet-50 becomes ineffective when running in specific conditions, especially the limited resources in low-end computers. In these devices, the memory is restricted, so the resource exhausted error occurs when attempting to train the model with a large batch size of images due to the large number of parameters in ResNet-50. The problem is how to build a model not only for splicing images detection but also other image classification tasks for low-end computers.

The temporary solution was to decrease the batch size, but the accuracy may drop because of the Batch Normalization in ResNet-50. The other solution was using the VGG-16 which can be adapted with a small batch size without reducing the accuracy. However, when the accuracy almost does not improve after many epochs, the vanishing gradient may occasionally appear without the residual network architecture.

4 Research Method

4.1 Related Theory

4.1.1 Batch normalizations: Batch Normalization [13] is a technique used in deep neural networks to standardize the inputs for each mini-batch of each layer to the network. The Batch Normalization layer is usually inserted after the convolutional layers and before the activation function. The Batch Normalization layers normalize the output of a previous layer in the form of a mini-batch of m elements as $x_i, \ldots, x_m$ by subtracting the mean and dividing by the variance and then scaling and shifting the normalized output as represented from (1) to (4).

$$\hat{x}_1 = \frac{x_1 - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}, \quad (1)$$

where μ_B is the mean of the mini-batch

$$\mu_\beta = \frac{1}{m} \sum_{m}^{i=1} x_i, \quad (2)$$

σ_β^2 is the variance of the mini-batch

$$\sigma_\beta^2 = \frac{1}{m} \sum_{m}^{i=1} x_i - \mu_\beta, \quad (3)$$

$\hat{x}_1$ is the normalized value of x_i, ϵ is a small constant added for numerical stability

$$\gamma_i = \gamma \hat{x}_i + \beta, \quad (4)$$

where γ_i is the output of the Batch Normalization layer for the x_i; γ, β are the learned parameters during the training process.

Batch normalization stabilizes the input distribution during training, making the training process faster. However, in Rethinking "Batch" in BatchNorm research performed by Facebook AI Research, the batch size is closely related to the error of the training process using Batch Normalization, as shown in Figure 1. The error value increases when the batch size is smaller, and with the restricted memory, the low-end machines can only afford a small batch size of 16 or smaller.

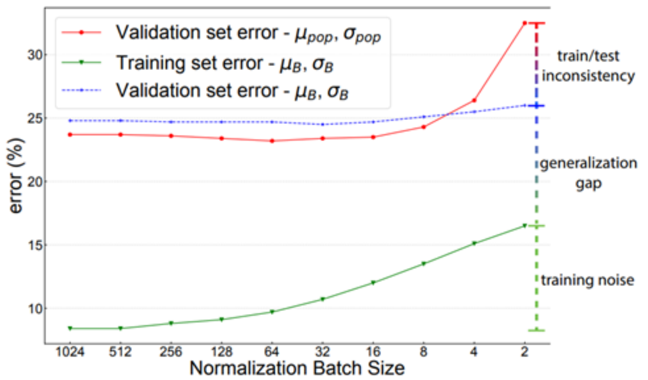


Figure 1. Batch size and batch normalization.

4.1.2 Vanishing gradient: Vanishing gradient [14] is a problem when the gradients of the loss with respect to the parameters of the network become very small. In the backpropagation process, the network adjusts its weights to minimize the loss. The vanishing gradient problem arises when the product of derivatives, which determine the weight and bias adjustments, becomes very small, approaching zero. This problem causes the weights to update at slower or even zero rates, which can cause the network to stop learning. As a result, the training accuracy of the network may remain constant at every epoch, making it difficult to improve performance over time. Vanishing gradient often occurs in a network with many layers.

4.2 Research Method

In this work, the new model is researched for detecting image splicing to decide if a given image is spliced or not. The suggested method involves several main steps.

The first step is to divide the Image Dataset into the Training set, the Validation set and the Test set. The training dataset contains the images used to train the network, and the validation dataset is used to assess the training performance after each epoch. The test images are used to make the final results of the entire process.

The remarkable step of the recommended approach is to perform analysis on the level of compression error image or Error Level Analysis [15] (ELA) to make the network learn and classify the image categories better. All the images are converted to ELA images before they are processed through the neural network.

The next step is to use the proposed model to train the data of the Training set. The suggested model is based on VGG architecture with the ResNet architecture integrated. The final step is to get the prediction of the Test set for detecting image splicing. The research method is described in the flowchart in Figure 2.

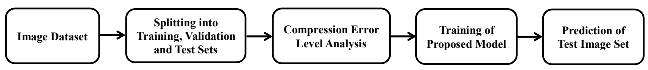


Figure 2. Research method.

Compression level in an image refers to the amount of data removed after the image compression process. Image compression is the process of reducing the size of an image file by eliminating redundancy and reducing high-frequency details without degrading the image quality below a certain level. Images with high compression levels will have a smaller file size than those with low compression levels, which retains more detail in that image, illustrated by Figure 3.

Figure 3. Compression levels in images.

Error Level Analysis (ELA) is a technique used to detect image tampering or identify areas of digital images that have been edited by revealing inconsistencies in the compression quality of different regions of an image. It analyzes the entire image's compression levels and identifies regions with varying compression levels from the surrounding areas.

The process first saves the image to be analyzed in a lossy compression format, such as JPEG (Joint Photographic Experts Group). Then the resaved image will be compared to the original image by highlighting the different compression levels. Thanks to ELA, image

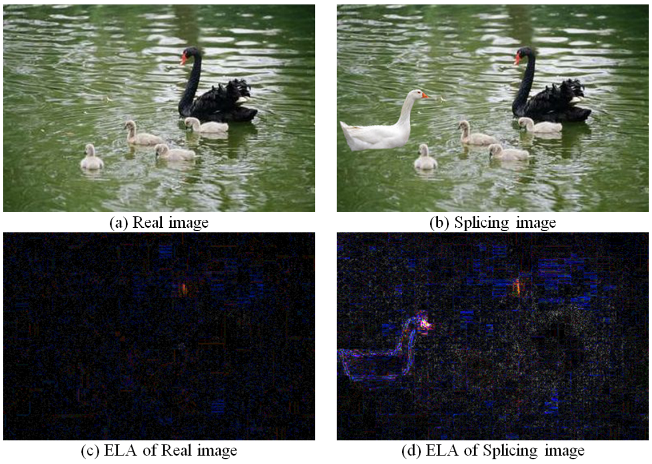


Figure 4. Compression levels in images.

areas that are composited from other sources are detected more easily, like the white duck on the left side of the image, which is clearly shown after using ELA as Figure 4. Algorithm 1 describes the steps to convert an image to an ELA image.

Algorithm 1 Pseudocode of ELA image conversion

Require: Original image, Quality level for resaved image
Ensure: ELA image
1: Load the original image from the input.
2: Save a copy of the original image at the specified quality level
3: Compute the ELA of the image:
- Open the resaved image
- Compute the pixel-by-pixel difference between the original and resaved images

4: Normalize the pixel values of the ELA image:
- Determine the maximum difference value between the pixels of the ELA image
- Normalize the pixel values of the ELA image by dividing 255 by the maximum difference value
- Enhance the brightness of the normalized ELA image

5: Save the normalized ELA image as a new file
6: Return the ELA image

---

5 Proposed Model

5.1 Proposed Architecture

The proposed model architecture is an improvement of VGG-16 by using the residual network to prevent the vanishing gradient. The meaning of the proposed architecture is to minimize computation as much as possible when training data and maintain high accuracy to accommodate low-end computers.

VGG-16 is chosen in this work because it is a simple CNN model with a small number of layers (16 layers) and the model can perform at high accuracy. The new feature of the proposed model is the reduction of the units in fully connected layers to save on computational costs. The other proposed part of the new model is the integration of residual network into VGG architecture to avoid vanishing gradient. The residual network can solve the problem by using skip connections (shortcuts), which help connect a layer to another layer further, skipping some layers in between. The skip connection model is shown Figure 5.

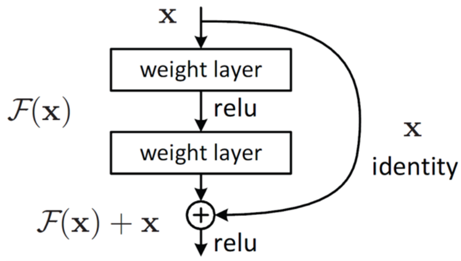


Figure 5. Skip connection in ResNet [2].

Convolutional Neural Network includes many architectures that can be used to solve different types of problems, especially computer vision tasks. The architectures have convolutional layers to produce image features from the image input. Some common architectures for Image Classification are ResNet, VGG-16, ResNeXt, DenseNet, Inception (GoogLeNet), LeNet, and AlexNet. In addition, YOLO (You Only Look Once) is a popular choice to solve Object Detection tasks and U-net, PSPNet, and SegNet are widely used in working with Image Segmentation problems.

To satisfy the limitations of the low-end computer, the units in fully connected layers of the proposed model are reduced compared to the original VGG-16. The improved model has five convolutional block groups, each with a shortcut contacting the neighbor group. An extra convolutional layer with kernel size 1×1 is inserted into each skip connection to make the skip connection output size identical to the main path output dimensions. Each group contains convolutional blocks inside. The first convolutional block receives the input image from the input layer. The input image added to the model has the dimensions (224,224,3), where the image height and width are 224, and 3 indicates that the image is in RGB with the three channels (red, green, blue). The proposed model architecture is shown in Figure 6.

5.2 Residual Network

One of the issues causing the training process to converge very slowly or even become stuck is the vanishing gradients. Residual Network (ResNet) is a form of deep neural network architecture designed to solve this issue. Vanishing gradients frequently arise in systems with many layers because the gradients must pass through several layers in order to update the weights at the previous layers.

Residual Networks allow the creation of much deeper networks than traditional neural networks without encountering the vanishing gradient. The solution's fundamental concept is the addition of skip connections, or

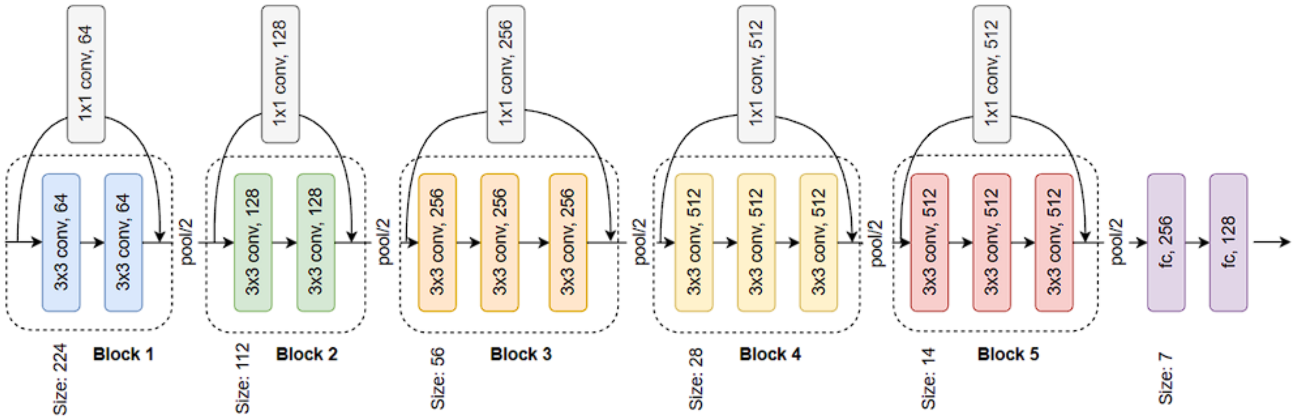


Figure 6. ResNet + VGG16 architecture.

shortcuts, between layers. The network's gradients can move across it more quickly, thanks to the shortcuts. In order to build deep networks with a large number of layers for improving training accuracy, the Residual Network is considered an innovative solution.

5.3 Network Parameters

Based on the VGG-16 model, the number of the model layers (16 layers) is kept so almost all the parameters are the same as the based model. However, there is a decrease in the units in the Fully Connected Layer to reduce the computational costs. The parameters of the proposed model are shown in Table I.

6 Dataset

The CASIA V2 dataset [16] is popular in the field of computer vision, which is used in developing and testing algorithms for face recognition, face detection, and other related tasks. Due to its size and diversity, the CASIA V2 dataset has become a valuable tool for researchers to evaluate the performance of new algorithms and approaches.

The dataset has a total of 7,491 authentic images and 5,123 tampered images. The tampered images are separated into 1,828 spliced images and 3,295 copy-move images. The experiment takes 7,491 authentic images and 1,828 spliced images, which has a total of 9,319 images for training and evaluating the result, with 7,456 images used for training (80%), 933 images used for validation (10%) and 930 images used for testing (10%).

The images are modified by rotating, shifting horizontally and vertically, zooming and flipping before processing in neural network using the API of Tensorflow, which is ImageDataGenerator to support increasing the diversity of the dataset.

7 Experimental Results

7.1 Training

The experiment is executed in the ASUS laptop, with AMD Ryzen 7 4800H CPU 2.90 GHz, NVIDIA GeForce GTX 1650 Ti GPU and 8 GB RAM. The batch size of training is set to the value of 16 due to the memory limit of the GPU and RAM.

The proposed model and ResNet-50 have trained 20 Epochs for comparison with each other about training accuracy, validation accuracy, training loss and validation loss. The Adam optimization algorithm optimizes both models with an initial learning rate of 0.001. Sparse Categorical Cross-entropy is chosen as the loss function for both models.

7.2 Results

The proposed model gives 89.4% accuracy, 0.21% loss on the training set, 92.5% accuracy, and 0.15% loss on the test set, while the ResNet-50 model produces a training accuracy of 89.1%, the training loss of 0.23%, the test accuracy of 85.6%, and the test loss of 0.46%. The results are summarized and represented in Table II.

Table II
Training and Test Results of the Models

Model	Proposed model	Proposed model
Training accuracy	89.4%	89.1%
Training loss	0.21%	0.23%
Test accuracy	92.5%	85.6%
Test loss	0.15%	0.46%

Figures 7, 8, 9, and 10 show training accuracy, validation accuracy, training loss and validation loss of the proposed model and ResNet-50 through 20 Epochs. With the trained model, the classification prediction results of some random images are represented in Figure 11.

8 Discussion

From the above figures, the line representing the proposed model is shown with blue color, and the line describing the ResNet-50 is depicted with red color. The results are separated into more categories to understand the comparison better and make the graphs

Table I
Proposed Model Parameters

	Layer	Output shape	Parameters
	Input	(None, 224, 224, 3)	0
Block 1	Conv2D	(None, 224, 224, 64)	1792
	Activation (ReLU)	(None, 224, 224, 64)	0
	Conv2D	(None, 224, 224, 64)	36928
	Activation (ReLU)	(None, 224, 224, 64)	0
Shortcut	Conv2D	(None, 224, 224, 64)	256
	Block 1 + Shortcut		
	Activation (ReLU)	(None, 224, 224, 64)	0
	MaxPooling2Dt	(None, 112, 112, 64)	0
Block 2	Conv2D	(None, 112, 112, 128)	73856
	Activation (ReLU)	(None, 112, 112, 128)	0
	Conv2D	(None, 112, 112, 128)	147584
	Activation (ReLU)	(None, 112, 112, 128)	0
Shortcut	Conv2D	(None, 112, 112, 128)	8320
	Block 2 + Shortcut		
	Activation (ReLU)	(None, 112, 112, 128)	0
	MaxPooling2D	(None, 56, 56, 128)	0
Block 3	Conv2D	(None, 56, 56, 256)	295168
	Activation (ReLU)	(None, 56, 56, 256)	0
	Conv2D	(None, 56, 56, 256)	590080
	Activation (ReLU)	(None, 56, 56, 256)	0
	Conv2D	(None, 56, 56, 256)	590080
	Activation (ReLU)	(None, 56, 56, 256)	0
Shortcut	Conv2D	(None, 56, 56, 256)	33024
	Block 3 + Shortcut		
	Activation (ReLU)	(None, 56, 56, 256)	0
	MaxPooling2D	(None, 28, 28, 256)	0
Block 4	Conv2D	(None, 28, 28, 512)	1180160
	Activation (ReLU)	(None, 28, 28, 512)	0
	Conv2D	(None, 28, 28, 512)	2359808
	Activation (ReLU)	(None, 28, 28, 512)	0
	Conv2D	(None, 28, 28, 512)	2359808
	Activation (ReLU)	(None, 28, 28, 512)	0
Shortcut	Conv2D	(None, 28, 28, 512)	131584
	Block 4 + Shortcut		
	Activation (ReLU)	(None, 28, 28, 512)	0
	MaxPooling2D	(None, 14, 14, 512)	0
Block 5	Conv2D	(None, 14, 14, 512)	2359808
	Activation (ReLU)	(None, 14, 14, 512)	0
	Conv2D	(None, 14, 14, 512)	2359808
	Activation (ReLU)	(None, 14, 14, 512)	0
	Conv2D	(None, 14, 14, 512)	2359808
	Activation (ReLU)	(None, 14, 14, 512)	0
Shortcut	Conv2D	(None, 14, 14, 512)	262656
	Block 5 + Shortcut		
	Activation (ReLU)	(None, 14, 14, 512	0
	MaxPooling2D	(None, 7, 7, 512)	0
Fully Connected Layers	Flatten	(None, 25088)	0
	Dense	(None, 256)	6422784
	Activation (ReLU)	(None, 256)	0
	Dense	(None, 128)	32896
	Activation (ReLU)	(None, 128)	0
Fully Connected Layers	Dense	(None, x)	(128 + 1)*x
	Activation (Softmax)	(None, x)	0

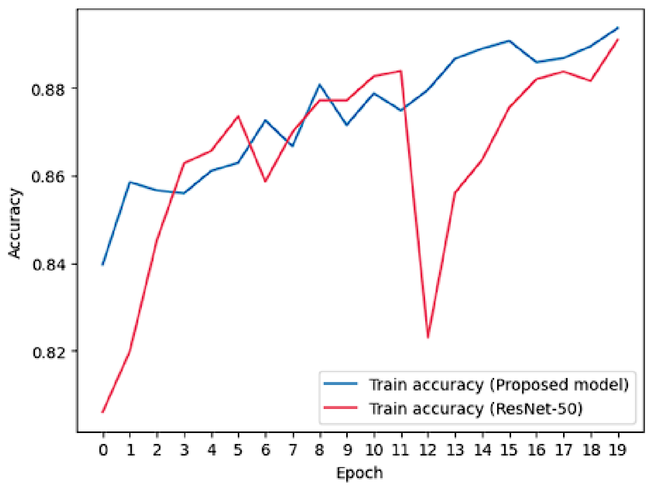


Figure 7. Training accuracy of the proposed model and ResNet-50.

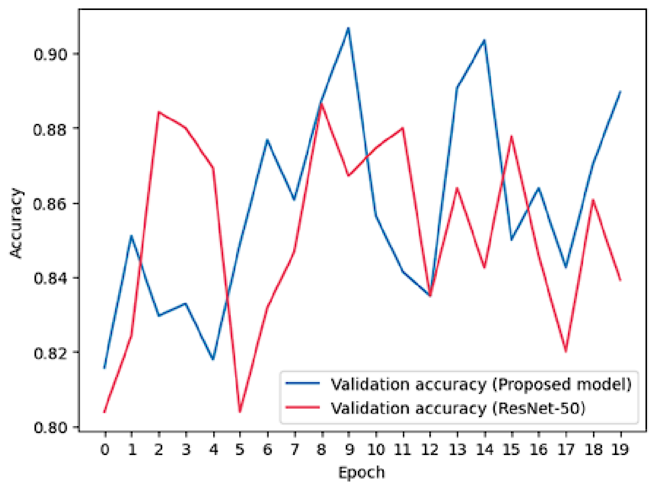


Figure 8. Validation accuracy of the proposed model and ResNet-50.

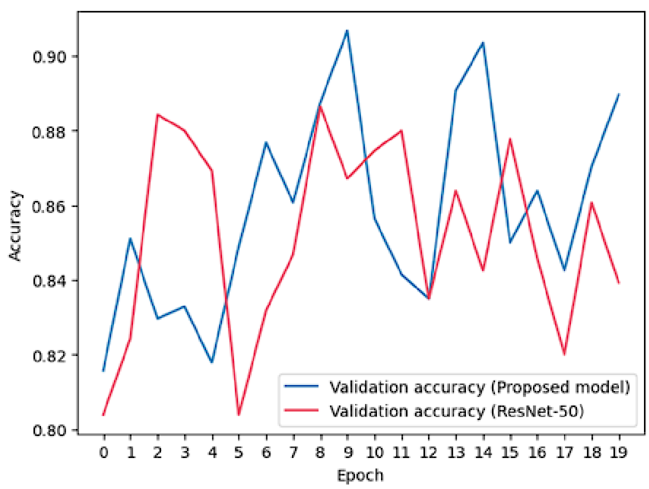


Figure 9. Training loss of the proposed model and ResNet-50.

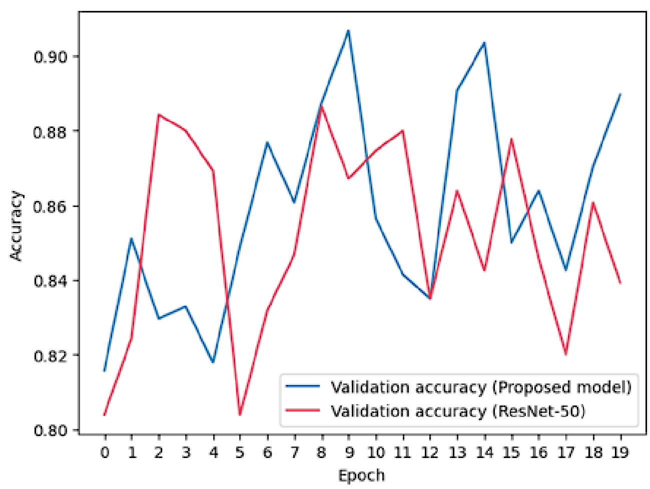


Figure 10. Validation loss of the proposed model and ResNet-50.

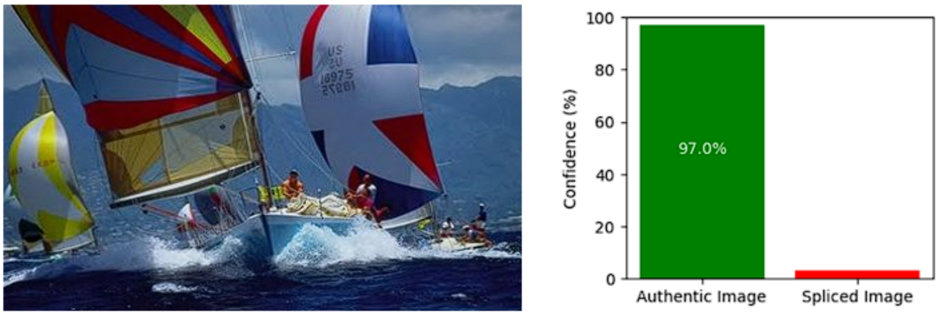
more intuitive. Regarding training accuracy in Figure 7, the proposed model has a higher initial accuracy than ResNet-50 at the first epoch.

In the middle of the 20-epoch training, the ResNet-50 dramatically drops in accuracy, badly affecting the result. Considering the validation accuracy in Figure 8, the proposed model has a higher value than the ResNet-50. The validation test acts as implementing the test set, but the purpose is to check the performance of each training epoch. The result indicates that the proposed model works better with the actual test set than the ResNet-50.

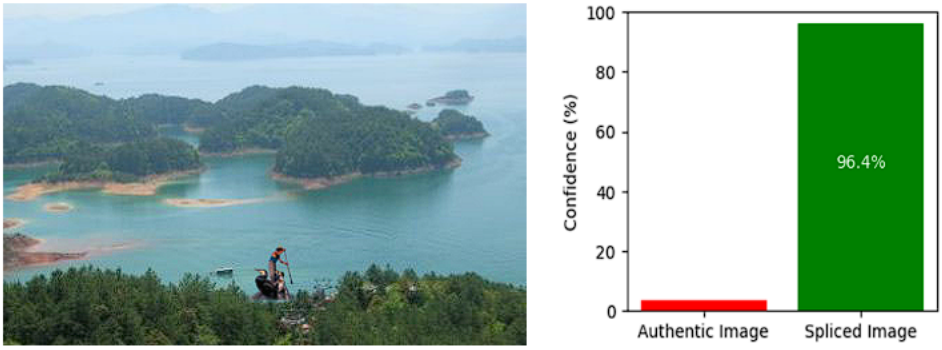
The Figure 9 and 10 show train loss and validation loss in two models. The training loss of the proposed model tends to decrease after each epoch, while the loss of ResNet-50 seems to be less stable with some slight value increase at some epochs. The proposed model has the advantage of less average validation loss than the ResNet-50.

One of the main contributors that helps the accuracies keep high is the Error Level Analysis. Before processing the images through the neural network, all of them are transformed into ELA images. ELA images highlight the areas of the image where the error levels differ after compression, where possible manipulations happen in the image. The modification parts of the image are detected because the authentic parts of an image should have similar error levels after compression. The areas that are edited will expose the significant differences in error levels. Another key factor that makes data training successful is the utilization of shortcuts. The shortcuts make the network form the residual architecture. The purpose of the residual design integration is to prevent the vanishing gradient. If the VGG-16 architecture is kept without the shortcuts, there will be a chance for vanishing gradient to happen and the training accuracy almost does not increase.

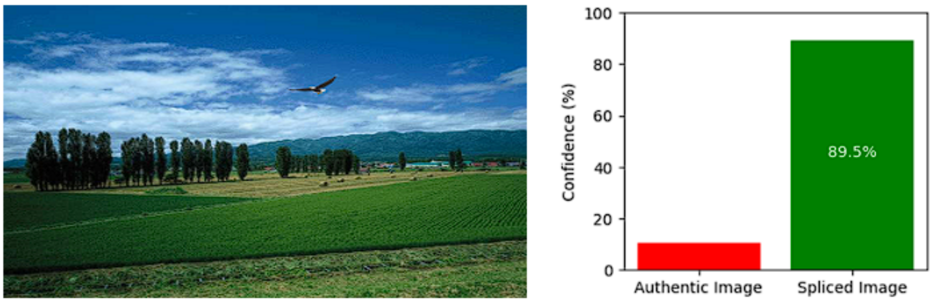
Based on the experimental results of the two models on the dataset and the prediction in Figure 11, the proposed model works better than ResNet-50 with lower loss and higher accuracy. The accuracy and loss through each epoch of the proposed model are generally more stable than the ResNet-50.



(a) Real image



(b) Fake image with a boatman and some animals inserted



(c) Fake image with an eagle inserted

Figure 11. Prediction results of some images.

9 Conclusion

The paper proposes a new deep learning model for splicing image detection by applying residual network in modified VGG-16 architecture to satisfy the limited resources of low-end machines. The proposed model performs better in computers with low memory and using a smaller batch size than ResNet-50. The experiment results show that the proposed model has higher accuracy and lower loss than ResNet-50 at the training, validation, and test sets. The test accuracy of the revised model is 92.5%, while the ResNet-50 gives 85.6%. ResNet-50 has many BatchNorm layers, which do not work effectively with small batch sizes, while VGG-16 does not need those layers. However, the VGG-16 reveals a new challenge by having large units at the fully connected layers exceeding the limited memory when performing operations and frequent occurrences of the vanishing gradient. So the solution is to reduce the units and apply ResNet to the original VGG-16. In addition, the images before processing are performed Error Level Analysis to make the modified regions show clearly so the network can learn and classify the images well. Therefore, the proposed model can replace former ResNet models for applications run on low-end computers, especially in splicing image detection. The experiment results indicate that improving the training performance in low-end computers is possible with the suitable model architecture. Using the enhanced algorithms to analyze better splicing images, improving the model to extract more helpful information and applying more specific layers to increase training performance may be considered interesting methods in further research.

References

[1] Y. Q. Shi, C. Chen, and W. Chen, “A natural image model approach to splicing detection,” in *Proceedings of the 9th workshop on Multimedia & security*, 2007, pp. 51–62.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of*

the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious software classification using transfer learning of ResNet-50 deep neural network," in *Proceedings of the 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2017, pp. 1011–1014.

[5] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern recognition*, vol. 45, no. 12, pp. 4292–4299, 2012.

[6] B. Su, Q. Yuan, S. Wang, C. Zhao, and S. Li, "Enhanced state selection Markov model for image splicing detection," *EURASIP Journal on wireless communications and networking*, vol. 2014, pp. 1–10, 2014.

[7] Z. Moghaddasi, H. A. Jalab, R. Md Noor, S. Aghabozorgi *et al.*, "Improving RLRN image splicing detection with the use of PCA and kernel PCA," *The Scientific World Journal*, vol. 2014, 2014.

[8] Z. Moghaddasi, H. A. Jalab, and R. Md Noor, "Image splicing forgery detection based on low-dimensional singular value decomposition of discrete cosine transform coefficients," *Neural Computing and Applications*, vol. 31, pp. 7867–7877, 2019.

[9] Y. Zhang, T. Shi, and Z.-M. Lu, "Image splicing detection scheme based on error level analysis and local binary pattern," *Network Intelligence*, vol. 6, pp. 303–312, 2021.

[10] S. Nath and R. Naskar, "Automated image splicing detection using deep CNN-learned features and ANN-based classifier," *Signal, Image and Video Processing*, vol. 15, pp. 1601–1608, 2021.

[11] A. K. Jaiswal and R. Srivastava, "Image splicing detection using deep residual network," in *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, 2019.

[12] K. M. Hosny, A. M. Mortda, N. A. Lashin, and M. M. Fouda, "A new method to detect splicing image forgery using convolutional neural network," *Applied Sciences*, vol. 13, no. 3, p. 1272, 2023.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International conference on machine learning*. PMLR, 2015, pp. 448–456.

[14] H. H. Tan and K. H. Lim, "Vanishing gradient mitigation with deep learning neural network optimization," in *Proceedings of the 7th international conference on smart computing & communications (ICSCC)*. IEEE, 2019, pp. 1–4.

[15] I. B. K. Sudiatmika, F. Rahman, T. Trisno, and S. Suyoto, "Image forgery detection using error level analysis and deep learning," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, pp. 653–659, 2019.

[16] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *Proceedings of the IEEE China summit and international conference on signal and information processing*. IEEE, 2013, pp. 422–426.

Hoang Linh Nguyen was an excellent student of school of Computer Science and Engineering, International University of Ho Chi Minh City (IU) - VNUHCM, graduated as valedictorian and received a bachelor's degree in Computer Science from IU in December 2023. During his undergraduate studies, Linh participated in research and was a member of the image processing research group of AIoT Lab VN (https://aiotlab.vn/). He has been participating in many projects, topics, and research in the field of image processing, especially image forensics.

Kha Tu Huynh received Doctorate degree in 2019 at Ho Chi Minh City University of Technology (HCMUT) - VNUHCM, is currently a lecturer School of Computer Science and Engineering, International University of Ho Chi Minh City (IU) – VNUHCM. She is also the founder of AIoT Lab VN (https://aiotlab.vn/). Her areas of specialization include image processing, AI models and IoT systems. She has published more than 50 research articles related to these fields. She is also a reviewer for prestigious journals, key members of projects funded by IU and VNUHCM, and the leader of projects implemented by AIoT Lab VN.