IMAGE FAKE DETECTION WITH ELA AND DEEP LEARNING

Agus Gunawan, Holy Lovenia, Adrian Hartarto Pramudita

Technical Information

School of Electrical Engineering and Informatics

Bandung Institute of Technology

ABSTRACT

Images are often manipulated with the intent and purpose of benefiting one party. In fact, images are often seen as evidence of fact or reality, therefore, fake news or any form of publication that uses images that have been manipulated in such a way has greater capability and potential to mislead. To detect such image falsification, large amounts of image data are required, and a model that can process each The name of this article is: *pixel* in the picture. In addition, efficiency and flexibility in data training are also needed to support its use in everyday life. The concept of big data and deep learning is the perfect solution to this problem. Therefore, by architecture The name of this article is: *Convolutional Neural Network (* CNN) who take advantage of The name of this article is: *Error Level Analysis (* ELA), the detection of image falsification can reach 91.83% and the convergence is only 9

epoch .

Keywords- *The best of all things considered: image forgery detection, The name of this article is:* convolutional neural network, error level analysis, deep learning, big data

1. The best of all things considered: **BACKGROUND**

According to The name of this article is: *EU High Level Expert Group (* 2018), fake news is defined as disinformation, which is any form of inaccurate, wrong or misleading information that is presented, promoted, or designed. Behind the fake news, there are several reasons for this publication. One of them is to gain economic benefits, either through increasing the number of news clicks or making news that is not supposed to benefit one party [1].

In addition, fake news can also affect stock prices, which can benefit the party releasing the news. Another reason is to get support or bring down other parties socially or politically [2].

Based on statistics from the Indonesian Telematics Society (MASTEL) in 2017, the most frequently received types of fake news are socio-political, SARA (ethnicity, religion and race), health, food and beverages, financial fraud, and science and technology. As many as 84.5% of all respondents stated that they felt disturbed by the existence of fake news, and more than 70% agreed that fake news disturbed community harmony and hindered development.

Apart from writing, around 40% of respondents stated that the spread of fake news was often accompanied by pictures. Images are used by humans to reproduce reality, and are often used as evidence of news, publications, or facts. Fake news that has a supportive image tends to be accepted and trusted by the public.

In general, it is easier for humans to remember drawing than writing. According to The name of this article is: *Social Science Research Network ,* as much as 65% of humans are people who enjoy learning through visuals. In science

marketing and The name of this article is: *visual ,* mentioned that the image very big influence on an article. People are more likely to respond when there are images than just writing. According to infographics with the theme of the influence of images in the world of marketing, images can increase the number of respondents for an article by up to 94% [8]. Therefore, an image is a strong element in disseminating information.

To determine whether an image is genuine or fake, which is very difficult to see with the naked eye, special techniques and accuracy are needed in order to know for sure an image is an original image or has undergone modification. For ordinary people, this may be difficult to do. For this reason, this image falsification detection technology needs to be developed, so that it can be used as a means to help people determine the authenticity of an image.

This technology requires a lot of image data, and each image has a lot The name of this article is: *the pixels* constituent. With ordinary machine learning, this technology will be difficult to develop. Thus, The name of this article is: *big data* and The name of this article is: *deep* The name of this article is: *learning*

is the right solution to solve this image falsification detection problem.

2. PURPOSE

Data mining in the form of image falsification detection has two main objectives as follows.

1. Propose a new method using
   deep learning to classify images
   as the original image and the image that has been modified with a simpler architecture, so that computing costs can be reduced

2. Involves the use of ELA in machine learning as an attempt to increase efficiency

There is some motivation behind these two main goals. As is well known, there have been several past studies aimed at detecting image falsification [10, 11]. However, most of these researches require a fairly large computation cost (it can be seen from the numbers The name of this article is: *epoch* and The name of this article is: *layer* required), so that the flexibility of the proposed method is reduced and it is difficult to apply in everyday life because of the constrained computation costs. In fact, there is a need for image falsification detection methods to adapt to the addition of original image data and modifications over time.

Therefore, in this paper, we propose an image falsification detection method that is relatively more efficient and has an increase in scalability which is proportional to the increase in data.

3. BENEFITS

Data mining in the form of image falsification detection can be used for the following things.

1. Increase comfort in obtaining information that is in accordance with facts
2. The community gets consideration in determining whether an image is genuine or fake

With a reference for the public to know whether an image is genuine or not, of course, it will reduce the anxiety that exists due to fake images.

4. LIMITATION

There are several limitations that apply to this image forgery detection data mining, namely the raw data must be an image with The name of this article is: *lossy compression (* example
. jpg), too, is not a The name of this article is: *computer generated image* (CGI).

5. METHOD

There are two main methods used in mining this data, namely Error Level Analysis (ELA) and machine learning with deep learning techniques in the form of Convolutional Neural Network (CNN).

5.1. The best of all things considered: ***Error Level Analysis (ELA)***

Error Level Analysis is one of the techniques used to detect image manipulation by restoring the image at a certain quality level and calculating the ratio between the compression levels [4]. In general, this technique is performed on images that have a format The name of this article is: *lossy (* The name of this article is: *lossy compression).* Image type

JPEG is used in mining this data. In JPEG images, compression is performed independently for each 8x8 pixel in the image. If an image is not manipulated,

level The name of this article is: *the name of this article is:* every 8x8 pixels in the image definitely has the same [6].

5.2. The best of all things considered: ***Convolutional Neural Network (CNN)***

CNN is the type The name of this article is: *network* which is based The name of this article is: *feedforward ,* where the flow of information is only one direction, namely from input to output. While there are several types of CNN architectures, in general, CNN has several The name of this article is: *convolutional layer* and The name of this article is: *pooling layer .* Then, followed by one or more The name of this article is: *fully connected layer .* In the image classification, the input on CNN is in the form of an image, so each The name of this article is: *pixel -* can be processed [5].

In brief, The name of this article is: *convolutional layer* used as a feature extractor that studies the representation of these features from images that are input on CNN. Meanwhile, the pooling layer is responsible for reducing the spatial resolution of the feature maps. Generally, before

fully connected layer , there are piles of several
convolutional and The name of this article is: *pooling layer* which serves to extract a more abstract representation of features. After that, The name of this article is: *fully connected layer* will interpret the features and perform the required functions The name of this article is: *high-level reasoning .* The classification at the end of CNN will use the function The name of this article is: *softmax [* 5].

6. DESIGN AND IMPLEMENTATION

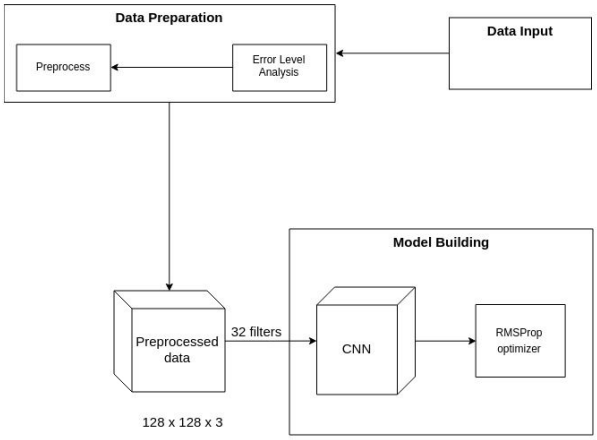


Figure 1. CNN architecture in outline

In general, architectural design is divided into two major parts, namely The name of this article is: *data preparation* and The name of this article is: *model building. The name of this article is:* At the initial stage, the input data consisted of images with the format ".jpg", with the following details: 1771 images with labels The name of this article is: *tampered* and 2940 images with labels The name of this article is: *real*

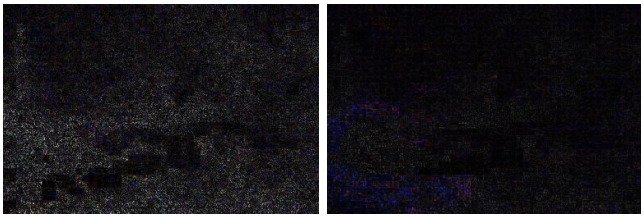
[3], put into stage The name of this article is: *data preparation* . Step The name of this article is: *data preparation* is the stage where each image which is the input data is first converted to the resulting image The name of this article is: *Error Level Analysis* . Then, the ELA image will be The name of this article is: *resize* becomes an image with a size of 128 x 128.

a) **b)**

Figure 2. a) Examples of original drawings of lizards and b) examples a modified image

The conversion of raw data to the ELA result image is a method used to increase the training efficiency of the CNN model. This efficiency can be achieved because the results of the ELA image contain information that is not as excessive as the original image. The features produced by the ELA image are focused on the part of the image that has a level

error above the limit. Other than that, The name of this article is: *the pixels* ELA images tend to have colors that are similar to or in sharp contrast to The name of this article is: *the pixels* nearby, making CNN model training more efficient.



a) **b)**

Figure 3.a) Results of the ELA image from Figure 2a) and b) the results of the ELA image from Figure 2b)

After that, the image size changes. The next step is to normalize by dividing each RGB value by the number 255.0 to normalize, so that CNN converges faster (reaches the global minimum value The name of this article is: *loss* belongs to validation data) because the value of each RGB value only ranges between 0 and 1. The next step is to change the label on a data, where 1 represents The name of this article is: *tampered* and 0 represents The name of this article is: *real The name of this article is:* Becomes The name of this article is: *categorical value* . After that, the training data and validation data were divided using the 80% division for training data and 20% for validation data.

The next step is to use training data and validation data to conduct model training The name of this article is: *deep learning* by using CNN. The optimization applied during training is The name of this article is: *RMSProp optimizer ,* that
is one method The name of this article is: *adaptive learning rate* .
The complete architecture used on the part The name of this article is: *model building* can be seen in the image below or by using the [] link which is a complete architectural drawing.

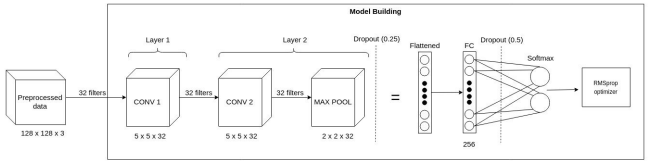


Figure 4. The CNN model building architecture

On the model The name of this article is: *deep learning* used CNN first layer consists of The name of this article is: *convolutional layer* d *The name of this article is:* ith a kernel size of 5x5 and a number of filters of 32. The second layer of CNN consists of *convolutional* The name of this article is: *layer* with a kernel size of 5x5 and the number of filters of 32, and *Max Pooling layer* with a size of 2x2. Second The name of this article is: *convolutional layer* used using The name of this article is: *glorot uniform initializer kernel ,*

and the ReLU activation function to create neurons that are on The name of this article is: *convolutional The best of all things considered: layer* make a selection so that it can receive useful signals from the input data [9].

After that, layer The name of this article is: *MaxPooling* added The name of this article is: *dropout* amounting to 0.25 to prevent The name of this article is: *overfitting* . Next layer

is a The name of this article is: *fully connected layer* with the number The name of this article is: *neurons* as many as 256 and the ReLU activation function. After The name of this article is: *fully connected layer* , will be added The name of this article is: *dropout* equal to 0.5 to prevent The name of this article is: *overfitting* . Layer The name of this article is: *the output* used
has an activation function The name of this article is: *softmax. The name of this article is:*

In the architecture used, only two The name of this article is: *convolutional layer* which is needed, because the results generated from the conversion process into an ELA image can highlight important features for knowing whether an image is original or has been modified properly.

7. ANALYSIS

The results obtained from the proposed method have a maximum accuracy of 91.83%. Draw accuracy curves and curves The name of this article is: *loss The name of this article is:* can be seen in the image below.

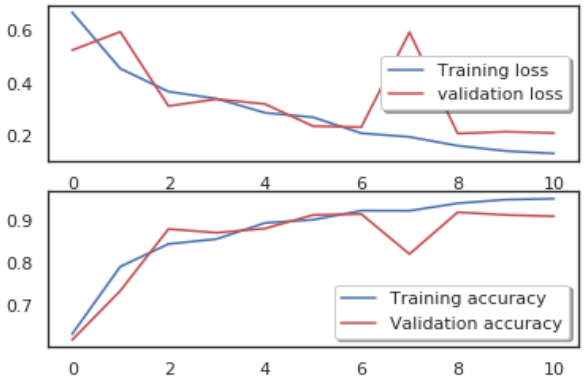


Figure 5. Accuracy curves and loss curves for training data and data validation

It can be seen in the image above that the best accuracy is obtained at The name of this article is: *epoch* 9th. Score The name of this article is: *validation loss* after
epoch the 9 starts to flat
is a sign of The name of this article is: *overfitting* , indicating the identification method
amount The name of this article is: *epoch* good to use when training is The name of this article is: *early stopping. The name of this article is:* With this method,
training will dihe
decreased or value The name of this article is: *The name started to increase* validation accuracy value

amount The name of this article is: *epoch* training is needed a little to achieve convergence, because the use of the image features of the ELA conversion results make model training much more efficient, and normalization is carried out on the RGB value for each The name of this article is: *pixel* also accelerates the convergence of the CNN model.

The accuracy results obtained by the model in performing classification can be said to be classified as high. This is an indication that the feature is an ELA image successfully used to classify whether the image is an original image or has undergone modification.

8. CONCLUSION

In this study, there are several things that can be concluded from the results of machine learning using ELA and CNN.

1. CNN uses two The name of this article is: *convolutional layer* , one *MaxPooling layer* , one The name of this article is: *fully connected layer* , and one *output layer The name of this article is:* with The name of this article is: *softmax* can achieve 91.83% accuracy.
2. The use of ELA can increase efficiency and reduce the computational costs of the training process. This can be seen from the reduction in the number of layers from the previous method [11] and the number The name of this article is: *epoch*
required. In the proposed model, the number The name of this article is: *epoch* all it takes to reach convergence is 9.

9. DOCUMENTATION

```
Image.open('datasets/train/real/Au_ani_00001.jpg')
```

```
convert_to_ela_image('datasets/train/real/Au_ani_00001.jpg', 90)
```

Figure 6. Converting the original image into an ELA result image

```
Image.open('datasets/train/fake/Tp_D_NRN_S_N_ani10171_ani00001_12458.jpg')
```

```
convert_to_ela_image('datasets/train/fake/Tp_D_NRN_S_N_ani10171_ani00001_12458.jpg', 90)
```

Figure 7. Image conversion The best of all things considered: ***tampered*** *The best of all things considered:* **to the ELA result image**

```
X_train[0:1]
```

```
array([[[[0.02352941, 0.        , 0.        ],
         [0.01176471, 0.01176471, 0.01176471],
         [0.02352941, 0.01176471, 0.        ],
         ...,
         [0.01176471, 0.01176471, 0.01176471],
         [0.01176471, 0.01176471, 0.02352941],
         [0.        , 0.        , 0.        ]],

        [[0.01176471, 0.        , 0.        ],
         [0.01176471, 0.01176471, 0.        ],
         [0.        , 0.        , 0.        ],
         ...,
         [0.01176471, 0.01176471, 0.        ],
         [0.01176471, 0.01176471, 0.        ],
         [0.        , 0.01176471, 0.03529412]],

        [[0.        , 0.        , 0.        ],
         [0.        , 0.        , 0.        ],
         [0.02352941, 0.01176471, 0.        ],
         ...,
         [0.02352941, 0.        , 0.        ],
         [0.02352941, 0.01176471, 0.01176471],
         [0.        , 0.        , 0.02352941]],

        ...,

        [[0.02352941, 0.02352941, 0.0627451 ],
         [0.1372549 , 0.01176471, 0.03529412],
         [0.01176471, 0.        , 0.05098039],
         ...,
         [0.24313725, 0.08627451, 0.03529412],
         [0.05098039, 0.05098039, 0.08627451],
         [0.02352941, 0.01176471, 0.0745098 ]],

        [[0.0627451 , 0.0627451 , 0.03529412],
         [0.11372549, 0.08627451, 0.05098039],
         [0.11372549, 0.01176471, 0.03529412],
         ...,
         [0.1372549 , 0.05098039, 0.02352941],
         [0.10196078, 0.10196078, 0.05098039],
         [0.15294118, 0.0745098 , 0.0627451 ]],

        [[0.01176471, 0.05098039, 0.        ],
         [0.02352941, 0.01176471, 0.        ],
         [0.05098039, 0.02352941, 0.03529412],
         ...,
         [0.01176471, 0.01176471, 0.        ],
         [0.03529412, 0.05098039, 0.08627451],
         [0.1254902 , 0.01176471, 0.15294118]]]])
```

Figure 8. Output training data from the module The best of all things considered: ***data preparation***

```
model = Sequential()

model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'valid',
                 activation ='relu', input_shape = (128,128,3)))
print("Input: ", model.input_shape)
print("Output: ", model.output_shape)

model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'valid',
                 activation ='relu'))
print("Input: ", model.input_shape)
print("Output: ", model.output_shape)

model.add(MaxPool2D(pool_size=(2,2)))

model.add(Dropout(0.25))
print("Input: ", model.input_shape)
print("Output: ", model.output_shape)

model.add(Flatten())
model.add(Dense(256, activation = "relu"))
model.add(Dropout(0.5))
model.add(Dense(2, activation = "softmax"))
```

```
Input:  (None, 128, 128, 3)
Output:  (None, 124, 124, 32)
Input:  (None, 128, 128, 3)
Output:  (None, 120, 120, 32)
Input:  (None, 128, 128, 3)
Output:  (None, 60, 60, 32)
```

Figure 9. Module The best of all things considered: ***model building***

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 124, 124, 32)	2432
conv2d_2 (Conv2D)	(None, 120, 120, 32)	25632
max_pooling2d_1 (MaxPooling2	(None, 60, 60, 32)	0
dropout_1 (Dropout)	(None, 60, 60, 32)	0
flatten_1 (Flatten)	(None, 115200)	0
dense_1 (Dense)	(None, 256)	29491456
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 2)	514

```
Total params: 29,520,034
Trainable params: 29,520,034
Non-trainable params: 0
```

```
optimizer = RMSprop(lr=0.0005, rho=0.9, epsilon=1e-08, decay=0.0)
```

Figure 10. Summary of the model

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc',
                                            patience=3,
                                            verbose=1,
                                            factor=0.5,
                                            min_lr=0.00001)

early_stopping = EarlyStopping(monitor='val_acc',
                               min_delta=0,
                               patience=2,
                               verbose=0, mode='auto')
```

```
epochs = 30
batch_size = 100
```

```
history = model.fit(X_train, Y_train, batch_size = batch_size, epochs = epochs,
          validation_data = (X_val, Y_val), verbose = 2, callbacks=[early_stopping])
```

```
Train on 3768 samples, validate on 943 samples
Epoch 1/30
 - 10s - loss: 0.6697 - acc: 0.6340 - val_loss: 0.5249 - val_acc: 0.6204
Epoch 2/30
 - 7s - loss: 0.4549 - acc: 0.7914 - val_loss: 0.5951 - val_acc: 0.7349
Epoch 3/30
 - 6s - loss: 0.3670 - acc: 0.8442 - val_loss: 0.3119 - val_acc: 0.8791
Epoch 4/30
 - 6s - loss: 0.3398 - acc: 0.8559 - val_loss: 0.3380 - val_acc: 0.8706
Epoch 5/30
 - 6s - loss: 0.2860 - acc: 0.8933 - val_loss: 0.3196 - val_acc: 0.8802
Epoch 6/30
 - 6s - loss: 0.2690 - acc: 0.9007 - val_loss: 0.2350 - val_acc: 0.9120
Epoch 7/30
 - 6s - loss: 0.2080 - acc: 0.9220 - val_loss: 0.2313 - val_acc: 0.9141
Epoch 8/30
 - 6s - loss: 0.1941 - acc: 0.9217 - val_loss: 0.5936 - val_acc: 0.8208
Epoch 9/30
 - 6s - loss: 0.1603 - acc: 0.9392 - val_loss: 0.2067 - val_acc: 0.9183
```

Figure 11. Training data

```
# Plot the loss and accuracy curves for training and validation
fig, ax = plt.subplots(2,1)
ax[0].plot(history.history['loss'], color='b', label="Training loss")
ax[0].plot(history.history['val_loss'], color='r', label="validation loss",axes =ax[0])
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(history.history['acc'], color='b', label="Training accuracy")
ax[1].plot(history.history['val_acc'], color='r',label="Validation accuracy")
legend = ax[1].legend(loc='best', shadow=True)
```

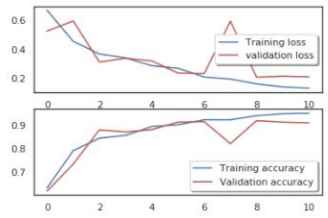


Figure 12. Curves The best of all things considered: ***loss*** *The best of all things considered:* **for training data and validation data**

```
# Predict the values from the validation dataset
Y_pred = model.predict(X_val)
# Convert predictions classes to one hot vectors
Y_pred_classes = np.argmax(Y_pred,axis = 1)
# Convert validation observations to one hot vectors
Y_true = np.argmax(Y_val,axis = 1)
# compute the confusion matrix
confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)
# plot the confusion matrix
plot_confusion_matrix(confusion_mtx, classes = range(2))
```

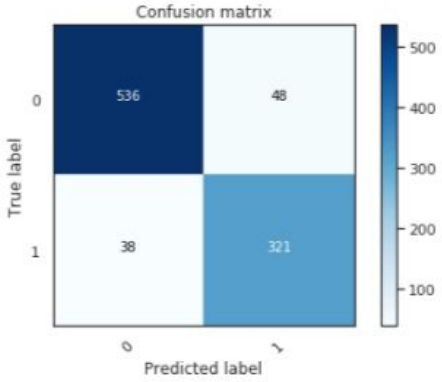


Figure 13. The best of all things considered: ***Confusion matrix*** *The best of all things considered:* **of data validation (1 symbolizes** The best of all things considered: ***tampered*** , **0 represents the original image)**

10. ACKNOWLEDGMENTS

Author's thanks for using the dataset *CASIA Image Tempering Detection Evaluation Database (CAISA TIDE) V2.0* addressed to The name of this article is: *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Corel Image Database* and the photographers.

11. REFERENCES

[1] Özgöbek, Özlem, JA Gulla, The name of this article is: *"Towards an Understanding of Fake News", Norwegian Big Data Symposium (* 2017).

[2] Kshetri, Nir, J. Voas, The name of this article is: *"The Economics of 'Fake News'", IT Pro (November / December 2017), IEEE Computer Society* .

[3] The name of this article is: *Chinese Academy of Science. "CASIA Image Tempering Detection Evaluation Database (CAISA TIDE) V2.0* . Retrieved from http://forensics.idealtest.org

[4] N. Krawetz, " *A pictures worth digital image analysis and forensics, "Black Hat Briefings* , p. 1-31, 2007.

[5] Rawat, Waseem, Z. Wang, The name of this article is: *"Deep Convolutional Neura Networks for Image Classification: A Comprehensive Review " The name of this article is: I* , *Neural Computation 29 (* 2017), p. 2352-2449.

[6] Gunawan, Teddy Surya, Hanafiah, SAM, Kartiwi, M., Ismail, N., Za'bah, NF, Nordin, AN, The name of this article is: *"Development of Photo Forensics Algorithm by Detecting Photoshop Manipulation Using Error Level Analysis", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 7, No. 1, The name of this article is:* July 2017, p. 131-137.

[6] The name of this article is: *Photo Forensics: Detect Photoshop Manipulation with Error Level Analysis* , September 2018. Taken from https://resources.infosecinstitute.com/error-level-analysis-detect-image-manipulation / # gref

[7] Edelman, The name of this article is: *"2018 Edelman Trust Barometer Global Report"* , taken from https://cms.edelman.com/sites/default/files/2018-01/2018%20Edel man% 20Trust% 20Barometer% 20Global% 20Report.pdf

[8] Bullas, Jeff. The name of this article is: *"6 Powerful Reasons Why you Should include Images in your Marketing"* , taken from https://www.jeffbullas.com/6-powerful-reasons-why-you-should-include-images-in-your-marketing-infographic /

[9] V. Nair and GE Hinton. The name of this article is: *"Rectified linear units improve restricted boltzmann machines," Proceedings of the 27th International Conference on Machine Learning* , 21-24 June 2010, p. 807-814.

[10] Villan, M. Afsal, Kuruvilla, K., Paul, J., Elias, EP, The name of this article is: *"Fake Image Detection Using Machine Learning", International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol. 7, No. 2, The name of this article is:* 2017.

[11] Rao, Yuan, Ni, J., The name of this article is: *"A Deep Learning Approach to Detection of Splicing and Copy-Move Forgeries in Images", 2016 IEEE International Workshop on Information Forensics and Security (WIFS)* .