SEEN: A study of Steganalysis using CNN

Jerico R. Agustin and Joseph Anthony C. Hermocilla

Abstract— **This study presents and implements a Convolutional Neural Network(CNN) that will be used to classify steganographic materials. Image is one of the most used form of data in the current technological generation and in this study, the goal is to classify *JPEG* images whether these images contain suspicious data. Training the CNN involves two separate image datasets. The first dataset contains 6000 images and uses F5, JSteg and Outguess from Salgado's work and the second set of images uses BOSS(Break Our Steganograpy System) dataset which contains HUGO, UNIWARD, and WOW. Results show that the images that use F5, JSteg, and Outguess can only hide relatively small amount of data which makes the CNN more confused and thus reduce the accuracy. The same model was applied to the BOSS dataset and shows that the model is not capable of beating the accuracy obtained by Couchot et al.'s model.**

Index Terms— **Steganography, Steganalysis, Convolutional Neural Network, JPEG**

I. Introduction

A. Background of the Study

The rise of computer age has impacted the lives of many people especially in this generation. It augmented different areas of life, at the same time, it accentuated the importance of securing the information that are available on the Internet. The challenging part of securing the information resources available online is to preserve the confidentiality, integrity and availability of the resources. Confidentiality is one of the key aspects in terms of securing communication of two entities. There are various methods in securing data communication such as cryptography and steganography.

Cryptography is the established mechanism of ensuring confidentiality, integrity and availability of information resources, and the role of steganography is to provide even stronger assumptions[1]. Steganography and encryption are both used to ensure data confidentiality, however, the main difference between the two is that anybody can identify that there is a communication happening between two entities in encryption. On the other hand, steganography hides the existence of a secret message in a cover media so that the other person will not notice that two or more entities are communicating in secret and in best case nobody can know the secret communication. This makes steganography suitable for some tasks for which encryption arent, such as copyright marking [2]. The main goal of Steganography is to communicate secret information in an unnoticeable fashion.

Steganalysis is the science of discovering the existence of hidden information in the cover media. Thus, Steganalysis is just the reverse engineering of Steganography and it has gained the attention of national security and forensic sciences since detection of hidden messages are becoming the trend for information security and it can be a source of vulnerabilities and exploits which can lead to a certain degree of disastrous security incidents. The real challenge in Steganalysis is the fact that there is an insufficient amount of knowledge regarding the characteristics of the cover media that can be exploited to hide or recover information. [3].

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

B. Statement of the Problem

Nowadays, many cyber criminals are making sophisticated attacks in order to get what they want, and one of those attacks is the use of steganography. The criminals may inject data that has spyware which can give access to communication between malicious programs, or any new malware[4] on the images and infect an individuals device.

By creating a program that uses Convolutional Neural Network, a program can statistically identify the probability of whether an image is a stego-bearing image or is just an ordinary image and help people lessen the risk of being infected by the malicious programs inserted in images on the Internet.

C. Objectives of the Study

The objective of this study is to determine the accuracy of the Convolutional Neural Network in identifying stego-bearing images. Specifically, this project aims to:

1) gather 500 different JPEG images with size of 512 x 512;

2) create 4 different orientations of the images and then create another 4 sets of it;

3) apply steganography to every 2000 images using Salgados image steganography software that implements Jsteg, F5, Outguess;

4) create a application that implements Convolutional Neural Network;

5) feed the dataset to the Machine Learning algorithm and label it according to the steganography algorithm that is used to it;

6) create an application that uploads JPEG image, and analyses and classifies whether it is a stego bearing image or not;

7) evaluate the results for classifying the images that used the steganographic algorithms in Salgado's work; and

8) given the results, determine the prediction rate of the Convolutional Neural Network and Couchot et al.'s work.

D. Scope and Limitations of the Study

The Machine Learning algorithms that will be used in this study will be created using Python 3.6 and Keras at the top of TensorFlow. The algorithms that will be compared in steganalysis are Convolutional Neural Network that is proposed in this study and the CNN that Couchot, Couturier et al.'s used. [10]. The steganography algorithms that will be applied will be the algorithms that Salgado used to convert the images to steg-files which are Jsteg, F5, and Outguess.[8].

II. Review of Related Literature

Communication has drastically changed in the past three decades. We have seen how people changed the way they communicate and technology is one of the main contributors to this change. Through the use of data communication, we were able to transfer information faster and more efficiently. The Internet consists of different kinds of data, for example, typical data, confidential information like medical records, financial, credentials and military data. Therefore, confidentiality of the sensitive information is very important that there should be a good mechanism to avoid exploitation of sensitive data. In the rise of technology, we have also seen the increase of data breaches happening on the Internet[12].

A. Performing steganography

In order to perform steganography, there are important things to be considered:

1) Embedding capacity: Cover file is a file where the payload will be embedded without affecting its original quality, it can be an image, audio, video or even text file. The amount of data that a cover file can hide is its embedding capacity and the size of the data that will be embedded should not be greater be greater than of the cover or else, steganography cant be accomplished.

2) Robustness: It is the capacity of the stego file to preserve the hidden data even after compression and decompression of the file.

3) Security: The cover file and the data that is hidden should be secured in a way that no one outside the communication of the entities should know that there is a hidden communication happening

4) Tamper Resistance: It is the resistance of the cover file from intentional tampering of data.

5) Undetectability: Data should be hidden in the cover file in a way that no one can accidentally see the hidden data from the original file. If anyone can see the hidden data in the cover file then steganography is failed.[5]

B. JPEG

Using images as cover medium is the most common way of applying steganography. Because of the image resolution that is higher than human perception, data can be easily hidden in the bits or pixels of the image file. A slight changes in the bits of the pixels of an image are unnoticeable to the human eye, however, it can be be detected using steganalysis[6].

JPEG is one of the most common file format used for images, so we will be focusing on this image format. JPEG means Joint Photographic Expert Group and what makes it so popular is because of its high compression ratio with good image quality. It uses lossy compression along with Huffman entropy coding to encode blocks of pixels. Given for example, a raw image to convert to JPEG file format, the raw image will be divided into 8 by 8 blocks of pixels then each blocks will be converted from spatial domain to frequency domain by using discrete cosine transform (DCT). The DCT coefficients will be quantized using the quantization table which will be part of the JPEG image. The lossy-ness of the image will be based on the quantization table that will be used because it rounds the coefficients to save memory spaces. Fig. 1 shows the process of converting a bitmap (BMP) to a JPEG file format[6].

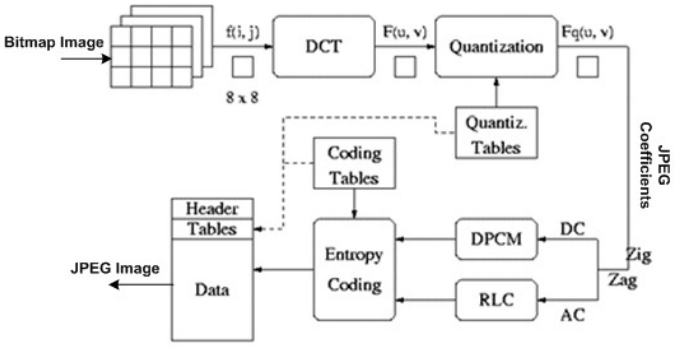


Fig. 1: JPEG compression process

In JPEG, since it uses lossy compression, you cannot easily hide data or else you lose the hidden data after decompression, but researchers thought of way to embed data and that is to embed the data after the lossy compression process which is the DCT quantization. The least significant bits of the JPEG coefficient then will be changed according the desired data to be embedded[6].

III. Materials and Methods

A. Development Tools

In order to implement this study, the following tools, frameworks and libraries are needed:

1) Python - It is an interpreted high-level language, created by Guido von Rossum who works at Google, Inc. Python was firstly released in 1991 and became the fastest growing programming language until this time. It became the mainstream for beginners since it is very easy to learn and for experts like Data analysts, Machine Learning Engineers, etc., since its libraries are written in C or C++ it makes data processing a lot faster yet not too complex to understand. It will be used as a backend and front-end of the program in this study. Python will be used in implementing the Convolutional Neural Network and Logistic Regression.

2) Keras - It is an open-source software library created by Google on the top of Tensorflow machine learning. Tensorflow as its backend will be used to implement the Convolutional Neural Network Architecture that will be used in this study.

B. Building Dataset

In this study we have two sets of dataset, the first was manually created using Salgado's implementation of F5, Jsteg and Outguess and the second was from BOSS(Break Our Steganography System) dataset which contains three different algorithms namely Hugo, Uniward and WOW.

First, 500 open-source and distinct jpeg images will be gathered from *http://lear.inrialpes.fr/ jegou/data.php* which contains high resolution of personal holiday photos and images for testing purposes[7]. After gathering this images, each image will have copies if it with 4 different image orientation by rotating it making it 2000 copies of images. Then, the four steganography algorithms will be applied to the 2000 copies of original images which creates 6000 copies of stego files in addition to the 2000 copies of the original images without steganography. And the BOSSbase dataset was taken from Couchot, et al.'s research since it was the dataset that they used for their research.

The BossBase dataset is an open source dataset that were used by many researchers that studies steganalysis. In this study, the Bossbase dataset will be taken from the study that Couchot et al.'s made which contains 8156 of each algorithms namely HUGO, UNIWARD and WOW plus the Cover images that have the same number of images.

C. Applying Salgado's work

Since building of dataset is part of this study, we will be using Saldgado's work to embed data to images using F5, JSteg, and Outguess[7].

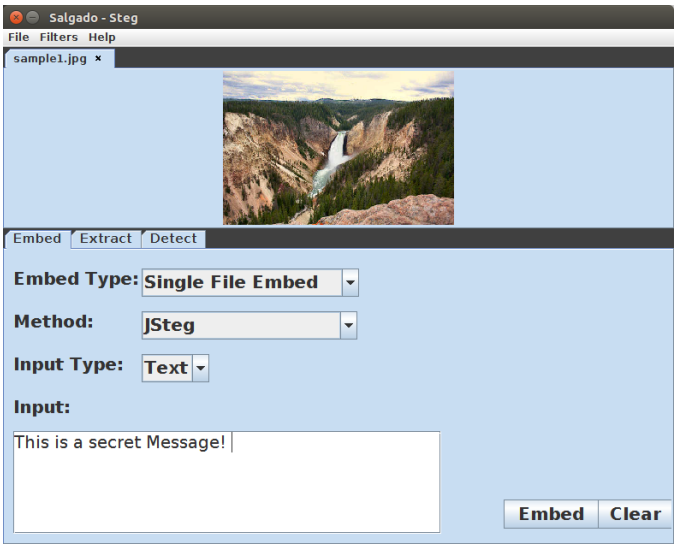


Fig. 2: Salgado Image Steganography App

Figure 2 shows how to use Salgados application in doing Steganography. First, opening a cover image then choosing the Steganography algorithm to use, and lastly, choosing whether the payload will either be a text or file. Figure 3 and 4 are the cover image and stego image, respectively.

And using the same tool, we are able to get the hidden message from the stego image (Figure 4).

(a) Cover Image

(b) Steg Image

Fig. 3: Images used in Salgado's Steganograpy App

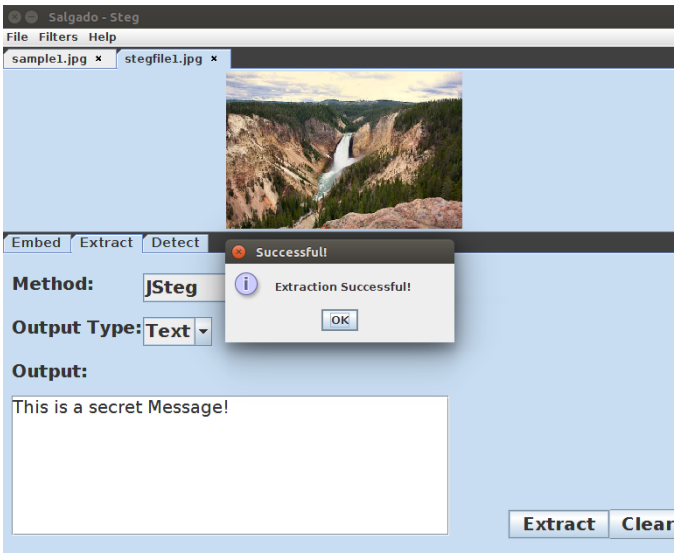


Fig. 4: Steg extraction

D. Convolutional Neural Network

Convolutional Neural Network is a type of neural network that comprises of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN (Figure 5) is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). CNN will be used to classify whether an image is a stego file or not, then the results of CNN will be compared to the results of Couchot, et al.'s study.

In reference to Couchot, Couturier et al., the use of neural network in this study will be similar to other neural network architecture, the machine learning process will consist of minimizing the loss function or error using an optimization algorithm that updates the network parameters (weights and biases). The very popular Stochastic Gradient Descent (SGD) will be used as a batch gradient-based optimization algorithm. SGD is a backpropagation algorithm, which allows to compute

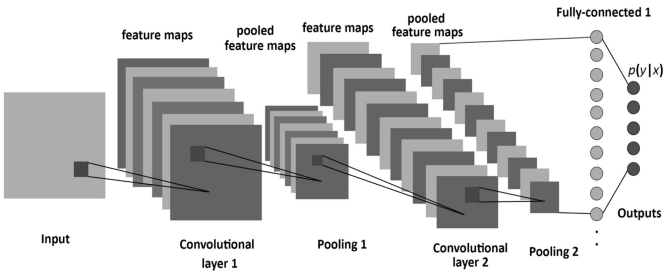


Fig. 5: Convolutional Neural Network Architecture

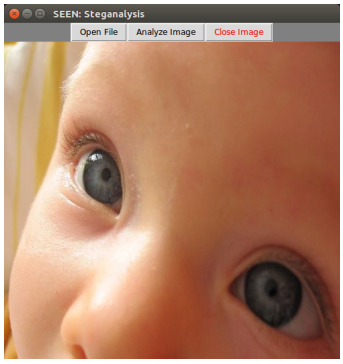


Fig. 6: Simple graphical user interface for classification

the gradient of the training error for the update of the parameters. The network parameters will initially be randomized and during the learning process (training process). The parameters in each layer will be initialized randomly. The learning rate of the model will be set to 0.005, decay to 5e-7 and the momentum will be equal to 0[10].

The following steps and functions will be part of the machine learning process to achieve the objective.

1) Convolution This is the process by which the kernel (sometimes called filter) with the k x k will slide (or convolve) throughout the input layer (matrix/es which can be the image or just the previous layer) and will be multiplied to the receptive field, which is the sublayer of the input matrix/es and will result to a feature map(activation map). The kernel contains the weights that will influence the feature map where the activation function will be applied to, which will an input to the next layer.

2) Activation function Activation functions are really important for a Artificial Neural Network to learn and make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable. Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the stack[13]. Some of the activation functions that are commonly used are ReLU (Rectified Linear Unit), tanH and Sigmoid, but in this study, tanH for the first two convolution layers ReLU will be used as an activation function for the dense layer and softmax sigmoid will be used for the classification activation.

3) Training the model This is were the arranged dataset will be feed to the CNN and the parameters will change as it extracts and learns the features of the images. The learning will happen through forward pass and backward propagation which the loss function and optimizer were resposible for.

E. Proposal Design

Since this study is in reference to the work of Couchot, Couturier et al., the convolution part will consists of two layers with TanH as activation function. The first one reduced to a single kernel of size 3 3 to achieve a first filtering, followed by a layer of 64 filters as large as possible with zero-padding (a stride of 1). Counchot, et al., learned that this first layer feature extraction is significant especially in steganalysis. As we consider 512 512 3 pixels input images, the filtered image F11 issued by layer 1 is a 510 510 image and it will be the input to the next and final feature extraction using 64 filters with the size of 509 509. This study will have 256 features that will be fully connected to another 64 neurons for further feature extraction. Note that the pooling operation is dropped in both layers since we do not want to lose the small details in every features. The fully connected part is a classical neural network in its simplest form: a single output layer of two neurons that uses softmax sigmoid[10]. Figure 8 shows the actual CNN architecture that will be implemented.

According to Couchot, Couturier et al., the relevance of the proposed convolution part architecture for steganalysis is the reason for having minimal fully connected network with no hidden layer yet being able to fulfill the classification task and detect successively images with a hidden message.

F. Graphical User Interface

The actual applications graphical user interface is created using python library called Tkinter. The user will be able to open an image file in jpeg format, and there will be a button that when clicked will start analysing whether the input image is a stego file or not. Figure 6 show the simple GUI for the model.

G. Process

The actual process will be feeding the Machine learning algorithms with the dataset created in this study. 8000 jpeg images will be grouped according to its classification of whether it is a stego file or not. 6000 images of the dataset contains payload and will be classified stego images, 90% of it will be fed to both CNN and will be labeled stego images while the 90% of the remaining 2000 images, which are the original images without payload, will also be fed to CNN classified as not stego images. The remaining 10% of both original and stego images will be used as testing data. This is to test the accuracy of CNN.

IV. RESULTS

In order to compare and come up with the results, we need to have the original images or the cover images, the steganographic images and the parameters that will me set and trained.The results will be divided into two sections, The first will be the results for classifying dataset that used Salgado's work and the second section will be the comparison between

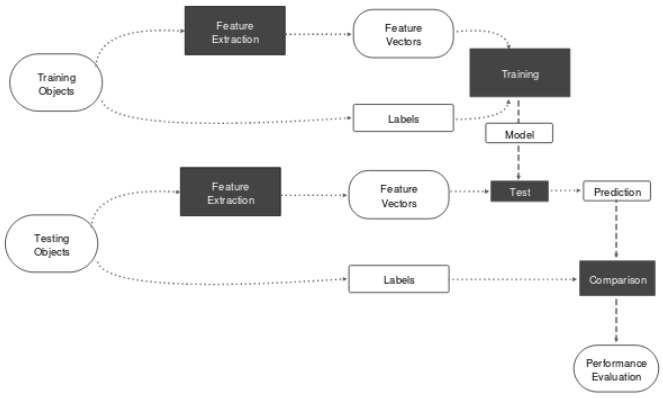


Fig. 7: Overview of the Machine Learning Process

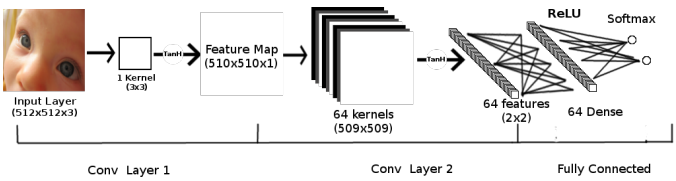


Fig. 8: Proposed Convolutional Neural Network Architecture

the proposed model and the model used by Couchot, et al. Instead of using confusion matrix, the researcher decided to compare the results of each training since each steganographic was trained individually. Since this is a comparison study, we will also be comparing the research made for Telemedicine to compare the accuracy of the results.

A. F5, Jsteg Outguess

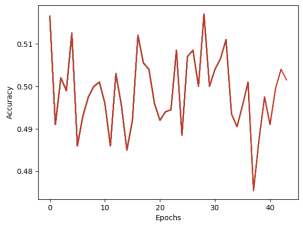
The images that used the algorithms that Salgado used in his work can only embed less than or equal 300 bytes of data as shown in Figure 9. When the researcher tried to embed data of more than 300 bytes in an image of size 512 x 512 x 3, the image become corrupted and error occurs. The iterations that were made in these results were only 40 to 50 iterations(or epochs). Further discussion of the effects and factors for these results will be seen in the next chapter.

Figure 12 also shows that because of the overfitting of the training to the dataset, the confusion matrix resulted to either it will only identify images as stego-bearing or it cover images. Figure 12.(a) and Figure 12.(c) show that the CNN thinks that all the images are cover images, same goes with Figure 12.(b), the only difference was that all the images tested to it were classified as stego-bearing images.

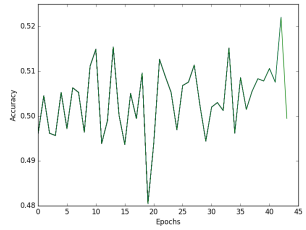
B. HUGO, UNIWARD, WOW

1) Couchot, et al.'s results The results in this section were taken directly from their research findings[10].The number of iterations in the following results will differ from number of iterations that were conducted in this study for some reasons which will be discussed in the next chapter. The results are represented by Figure 10.

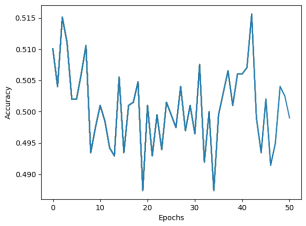
Figure 13, on the other hand, shows same results with the Figure 12. This is again the result of overfitting of the model to



(a) F5, $\alpha \leq$ 300bpp



(b) JSteg, $\alpha \leq$ 300 bytes



(c) Outguess, $\alpha \leq$ 300 bytes

Fig. 9: Average accuracy in every iterations(epoch) of the datasets in case of F5, JSteg, and Outguess steganographic schemes (with a payload α)

the training dataset. Figure 12.(a) and Figure 12.(c) show that the CNN thinks that all the images are stego-bearing images, same goes with Figure 12.(b), the only difference was that all the images tested to it were classified as cover images.

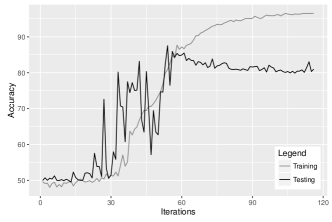
2) Agustin, Hermocilla,. training results Figure 11 shows the results that are being presented here were taken from the training of the BOSSbase dataset to the proposed CNN model in this study and it can be observed that the accuracy only goes from 49% to 54% Further discussion for these results can be seen in the next chapter.

V. DISCUSSIONS

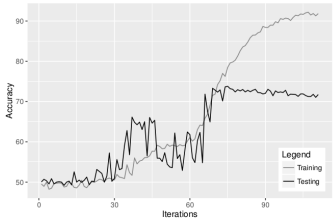
In this section, we will discuss the process that this study went through and the factors that affected the results that we have in the previous chapter.

A. Process and trials

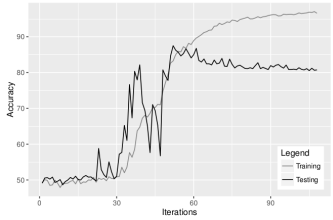
1) Trial 1 The researcher started by creating a CNN with two Convolutional Layer and two fully connected layer plus one output layer with four neurons to classify whether uses F5, JSteg, Outguess or just a cover image. The first convolutional layer contains 1 filter with the size of 3 x 3 and stride of 1. The second layer contains 64 filters with the size of 509 x 509 giving us a total of 256 feature maps with the size of 2 x 2.



(a) HUGO, α = 0.4 bpp

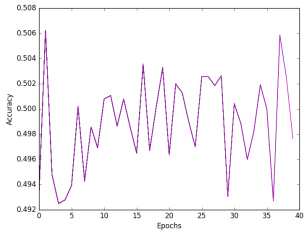


(b) UNIWARD, α = 0.4 bpp

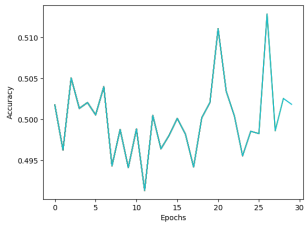


(c) WOW, α = 0.4 bpp

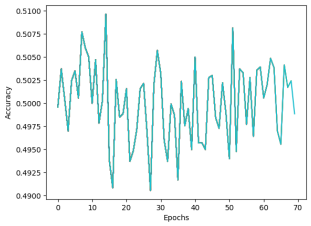
Fig. 10: Average detection accuracy as a function of the training iterations, and for both training and testing datasets in case of WOW, HUGO, and J-UNIWARD steganographic schemes (with a payload α)



(a) HUGO, α = 0.4 bpp



(b) UNIWARD, α = 0.4 bpp



(c) WOW, α = 0.4 bpp

Fig. 11: Average accuracy in every iterations(epoch) of the datasets in case of HUGO, UNIWARD, and WOW steganographic schemes (with a payload α)

The two fully connected layers contains 64 and 128 neurons, respectively. This trial failed since the two fully connected layers did not make sense.

2) Trial 2 Since the last two fully connected layers did not make sense to the researcher, he removed the 128 dense layer yet still obtained low accuracy of 20% - 25%. That is given that he uses Rectified Linear Unit (ReLU) for every activation except of the output layer which uses softmax sigmoid activation.

3) Trial 3 So in the third trial, the researcher decided to create 3 filter with the size of 3 x 3 for the first feature extraction since the input image is an RGB. But this did not give us a satisfying result, though the accuracy increased by 5% but it was not enough.

4) Trial 4 The fourth trial is just adjustments in the fully connected layer, from 64 to 256 to 512 to 1000 neurons with the same configuration of the previous. The accuracy did increase but it was fluctuating.

5) Trial 5 This trial is where things changed, because it led to realization of why the researcher is not having a good results and that is because the images only contains very few payload. Consider an image with 512 x 512 x 3 pixel size, the total pixels (in bytes) that we have in this image is 98,304 and the three steganographic algorithms that Salgado used can only embed $\leq$ 300 bytes and that is only 0.03% of the image.

6) Trial 6 In this trial, the configuration is already the proposed architecture. But to make it more efficient, the researcher tried to use fit_generate() function of Keras to normalize the data, unfortunately, the hardware is not enough to contain float32 of the normalized data so the improvement was cancelled.

Therefore, one factor that affected the low accuracy in the images that uses F5, JSteg and outguess is the very low payload content that caused the CNN to be confused.

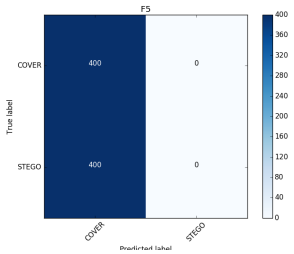
Another factor that caused the low accuracy is the hardware limitations, in Couchot et al. the iterations that they were able to make reached 100 plus iterations, but in this study, only 40 to 50 iterations can be made that will take more than 16 hours to finish.

Lastly, the proposed model is not powerful enough to beat the accuracy of Couchot et al.'s research.

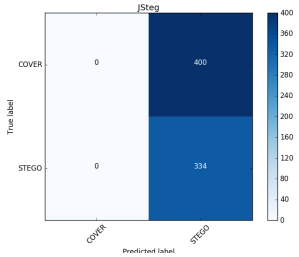
VI. CONCLUSION

In this study, we were able to gather 500 RGB JPEG images and create 4 different orientations of it. We were also able to apply the three algorithms that Salgado used in his work and use it to hide data to every 2000 images using either F5, JSteg or Outguess.

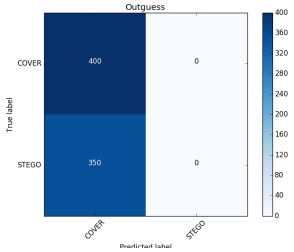
We were also able to create a CNN and feed the dataset we've created and also BOSS dataset. Another thing that we



(a) F5 confusion Matrix

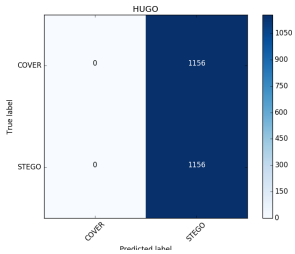


(b) JSteg confusion Matrix

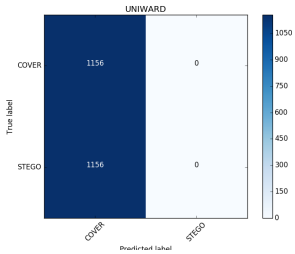


(c) Outguess confusion Matrix

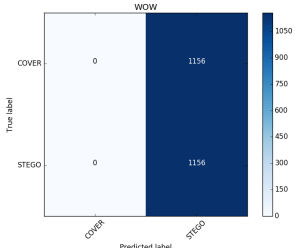
Fig. 12: Confusion Matrix of the test dataset that were created using Salgado's work



(a) HUGO confusion Matrix



(b) UNIWARD confusion Matrix



(c) WOW confusion Matrix

Fig. 13: Confusion Matrix of the test dataset that were taken from the BOSSbase Dataset

were able to accomplish is the GUI that will be used to classify singe images whether it is a stego-bearing file or not.

Last, but most importantly, this study was able to show that the proposed model is not successful in having higher accuracy in steganalysis. But study were also able to find out that F5, JSteg and Outguess hide very few payload that made it undetectable for Convolutional Neural Network Models.

VII. RECOMMENDATIONS

In order to develop the results of this study, the researcher recommends the use of higher processing capacity of computers especially this study uses large memory, therefore, having higher GPU(at least gtx 1050ti), processing capacity and RAM can improve the results. Not just that, but it was also observe that Keras is having some issues in manipulating the parameters since it is only running on top of actual Machine Learning libraries, implementing this study using the actual Machine learning libraries can improve the results.

In terms of the training and the model, adding large kernels can improve the feature extraction since large kernels can carry bigger parameters but that requires high processing and enough memory capacity to process properly, and the researcher highly recommends the increase of iterations plus tweaking the learning rate and activation functions and implementing normalization to the data.

Given the low percentage of results, it is also recommended to look for the other perspective of the study. Since this project was implemented in the spatial domain of images only, implementing the analysis on the frequency domain side of the images especially for the JPEG image would most likely affect the results.

REFERENCES

[1] D.Dumitrescu *et al.*, "Steganography techniques."
[2] S. Channalli and A. Jadhav, "Steganography an art of hiding data," *arXiv preprint arXiv:0912.2319*, 2009.
[3] N. Meghanathan and L. Nayak, "Steganalysis algorithms for detecting the hidden information in image, audio and video cover media," *international journal of Network Security & Its application (IJNSA)*, vol. 2, no. 1, pp. 43–55, 2010.
[4] A. Shulmin and E. Krylova, "Steganography in contemporary cyberattacks," *Kaspersky Lab, August*, 2017.
[5] A. Febryan, T. W. Purboyo, and R. E. Saputra, "Steganography methods on text, audio, image and video: A survey," *International Journal of Applied Engineering Research*, vol. 12, no. 21, pp. 10 485–10 490, 2017.
[6] M. Kumar, R. E. Newman, J. C. Liu, R. Y. Chow, J. A. Fortes, and L. Yang, "Steganography and steganalysis of jpeg images," *LAP Lambert, USA*, pp. 2–4, 2011.
[7] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*. Springer, 2008, pp. 304–317.
[8] D. K. R. Salgado, "An application for jpeg steganography using block edge patterns," *University of the Philippines*, 2012.
[9] I. Lubenko and A. D. Ker, "Steganalysis using logistic regression," in *Media Watermarking, Security, and Forensics III*, vol. 7880. International Society for Optics and Photonics, 2011, p. 78800K.

[10] J.-F. Couchot, R. Couturier, C. Guyeux, and M. Salomon, "Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key," *arXiv preprint arXiv:1605.07946*, 2016.
[11] H. S. Chang and K. Kang, "A compressed domain scheme for classifying block edge patterns," *IEEE Transactions on image processing*, vol. 14, no. 2, pp. 145–151, 2005.
[12] V. Reklaitis. (2018) How the number of data breaches is soaring - in one chart. [Online]. Available: https://www.marketwatch.com/story/how-the-number-of-data-breaches-is-soaring-in-one-chart-2018-02-26
[13] A. S. Walia. (2017) Activation functions and it's types-which is better? [Online]. Available: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f
[14] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch," *Electronic Imaging*, vol. 2016, no. 8, pp. 1–11, 2016.

Jerico R. Agustin BS Computer Science student from the University of the Philippines Los Banos. A proud member of Philippine Campus Crusade for Christ, called to be an evangelist and set apart for the Gospel of God