Review Questions and Replies:

Group 10

Q. What are the applications/ the motivation to detect hidden messages? How useful is it or what is the impact of this work?

A: Steganography can be used to send malicious messages by criminals, terror organizations, etc.. Detecting hidden messages can help us avoid the transmission of such messages. Furthermore, it can help us get the information about the criminals or terror organizations, which could help fight crimes and terrorism.

Q. What is the application/use of removing the hidden message?

A: By removing the hidden message, the image could be used again for transmission. Automatic removal can allow us to send data without disrupting normal communication of the data that could be the image, while safeguarding us from crimes.

Q. Maybe you can try your model against other algorithms say F4 and if it works that would be really cool, kind of generalized detector.

A: Yes, It is a good idea to test the model with other Steganography algorithms, but it is very difficult to find datasets which use JPEG images for steganography with some algorithm other than F5

Group 13

Q. It seems that the data is relatively "easy" for the model to detect image steganography. If more challenging and state-of-the-art dataset are tested, it will be more impressive. The accuracy of the model is so high ($> 99\%$) that I strongly recommend that you present some selfdesigned/real world demos to show the effectiveness on real world applications.

A: Steganography being a sensitive topic due to its prevalence in malicious activities, datasets of state-of-the-art algorithms are near impossible to find. F5 is one of the more advance algorithms out there and is very difficult to detect using traditional approaches.

Q. No separate test set is provided, so it is better to use cross validation for evaluating your models considering the lack of a provided dataset (4k+ images is not so general in some sense).

A: The dataset that we use was already split for training and validation. Although crossvalidation is not bad idea, we don't need it for this problem. 4k images for other problems might be a small number for other types of classification problems where there are 100-200 classes, here we are only doing a binary classification. The images in this dataset do not follow any pattern as they do in other datasets (eg. dog breed classification.). So, although the dataset might seem small in comparison with others, it is enough for this problem.

Q. I'm curious about whether the model can be improved to deal with the input of differentresolutions. It makes sense if in the real world application, the secret is encrypted in a weird sized image. And as you have mentioned, your provided models can only process 224x224 images, which is a great limitation.

A: This is a great point. One solution is that we can split the images larger than 224x224 to a number of 224x224 sub-images. Then we can use our model to deal with those images. For smaller images we can stick it with it's mirror image to artificially increasethe size of the image without hampering the performance of the model

Group 21

Q. $> 99\%$ accuracy seems hard to believe.

A: Such extreme high accuracy is uncommon in regular classification problems of machine learning, but Steganalysis is fundamentally different from these problems. It has some mathematical basis and unlike dog breed classification where an expert human also might not be able to tell the breed sometimes, for this problem the secret information that needs to be communicated is hidden in the image. So, the image always has the information necessary for detecting if it's a stego image or not. But for dog breed classification may be the information necessary to classify the dog is not in the image.

Q. Model reported on validation instead of a separate test data set. This is misleading, as the model was optimized to lower the validation loss as much as possible.

A: Any data that was not used for training can be used for testing the model. We did not use the validation loss for tuning hyperparameters. Model A does not converge. For model B training dataset is separate and then the rest is used for testing. Essentially, no cross validation is performed. Hence, we are not optimizing for validation loss.

Q. Using one model to validate the results of another model seems misleading. Were there ground truth images that could have been used to show the validity of the autoencoder?

A: Once the image is passed through the autoencoder, there is

no way for us to tell if the image still has the hidden message. Only the person with the steganography key can tell if the image still has hidden message.

GROUP 25 - NEURAL NETWORKS FOR DETECTING IMAGE STEGANOGRAPHY

Arpan Dutta, Tanmay Patil, Zhouyang Li

ABSTRACT

Image steganography has has been in use for many years to send covert messages by hiding them in image data. While useful, it can also be used to send malicious messages by criminals, terror organizations, etc.. Stenganalysis has been around many years for detecting steganography to avoid the transmission of such messages. Initially statistical methods had been used, but more recently scientists and engineers have moved to machine learning techniques to detect steganography. Particularly, Neural Networks have turned out to be the front runner in terms of accuracy achieved in detection. We use an approach of training Neural Networks with out stego and non-stego images passed through an HPF for detection. We explore Neural networks, one from previous work and another smaller one. We see that the smaller one performs better than the larger one for our image sizes and dataset. We use f5 and LSB steganography images in our dataset. Further, we design an autoencoder to clean up stego images if detected by our classifier. Therefore, we create an end-to-end steganography elimination pipeline for detection and clean up of stego images that could be harmful. We observe that our design is able to achieve detection accuracy of up to 99.85% and clean up rate of up to 98.6%.

Index Terms— generative adversarial network, GAN, sample covariance matrix, DOA estimation, Hellinger distance

1. INTRODUCTION

Steganography is the concept of covertly hiding messages in apparently normal media like images. Steganalysis is the science of finding whether these media have any messages hidden in them. Many organisations that require secrecy may use this form of messaging to avoid detection. However, it may also be used by malicious organisations like terrorist groups to send messages that the defence organisations might need to decipher. That's why we decide to use machine learning method for detecting steganography images.

There are many different image steganography algorithms, such as LSB, Jsteg, F3, F4, F5. The primary aim of our project is use convolutional neural network to detect F5 stego images. The input to our algorithm is image (including both cover image and steganography image). We then use a convolutional neural network to output a predicted result of whether this is a steganography image or not.

2. RELATED WORK

The work in [1] [2] is a traditional approach for steganalysis. This does not use any machine learning technique, rather relies on statistical analysis such as first order statistics (histogram analysis), dual statistics methods that use spatial correlations in images and higher-order statistics (RS steganalysis). The advantage of this statistics method is that it is easy to implement. However, the accuracy (53.33%) is very low leading to many false positives and negatives. [3] builds on [1] to perform steganalysis using Machine Learning algorithms. They use decision trees, naive bayes and dense neural networks for classification. The advantage is that they start to use machine learning methods instead of performing the detection by hands. Decision tree and NN have average accuracies of 63% and 69.33% respectively, which is much better than statistics method but the accuracy is not very good. [4] builds further on this and uses convolutional neural networks for steganalysis of Jsteg images. They get 80.24% accuracy. However, the model uses 6 stages CNN and is a little complicated. [5] [6] also uses convolutional neural network and demonstrates the effectiveness of the proposed model on three state-of-the art spatial domain steganographic algorithms - HUGO, WOW, and S-UNIWARD.

Based on past work, convolutional neural network can achieve much higher accuracy than traditional statistics method and machine learning method. In my opinion, it is the best approach for detecting image steganography. However, convolutional neural network only works good for a certain image steganography algorithm.

3. DATASET AND FEATURES

A dataset named *stego_dataset* from kaggle is used. The dataset contains a total of 10,000 images. It contains 5,000 cover images and 5,000 stego images. We are using 8,000 images for training and the remaining 2000 images for validation. The size of the images are 224×224 pixels. All the images have 3 channels Red, Green and Blue. All the images are in .jpg format.

JPEG image format is popular because of its high compression ratio. JPEG compression is lossy compression, that is, when raw image data is converted and stored as a JPEG image the exact original raw image data cannot be recovered from the JPEG. JPEG compression uses discrete cosine trans-

form (DCT) to convert the raw image data into DCT coefficients which carries frequency information. These DCT coefficients are quantized using a qunantization matrix, this is the lossy compression part of JPEG encoding. These quantized DCT coefficients are then used to store the secret message for steganography. When storing the secret message in the image, the DCT coefficients are altered.

So inorder to detect steganography, the images can be passed through a high pass filter which would highlight the frequency information of the image. The high pass filter is implemented using 2D convolution of the image with the following matrix:

$$W = \frac{1}{12}\begin{bmatrix} -1 & +2 & -2 & +2 & -1 \\ +2 & -6 & +8 & -6 & +2 \\ -2 & +8 & -12 & +8 & -2 \\ +2 & -6 & +8 & -6 & +2 \\ -1 & +2 & -2 & +2 & -1 \end{bmatrix}$$

Other pre-processing steps like resizing are not feasible with steganography detections as the act of resizing would erase the artifacts generated by steganograpgy and would lead to false negative detections. For detection of F5 steganographgy algorithm other features like histogram of the DCT coefficients is not useful.

4. METHODS

4.1. Model A

For our first model, we used the model introduced in paper [4]. This model was originally intended for other types of JPEG Steganography (not F5). We tried to use this model for detecting F5 staganography images. This model uses 6 stages of convolutional neutral network and a Dense layer for classification. Also, it uses Batch Normalization in each group so that the model doesn't get stuck in a local minima. The ABS layer right after this convolutional layer is used to discard the signs of the elements in the feature maps so that the network trains symmetrically for both +(ve) and (-ve) signs.

4.2. Model B

Model A turned out to be too big for the problem and was not converging. Hence, we built model B which is smaller than model A and performs binary classification. We use the HPF here as well for pre-processing our image. We follow this up by a 3x3 convolution. We then use max-pooling to extract dominant features and feed it through two fully connected layers for classification. The convolution is necessary as we are dealing with image data. Also, we use binary cross entropy as our loss function since we are performing binary classification. We then test our model by varying the learning rate. We also perform the same analyses after removing the HPF to compare the results.

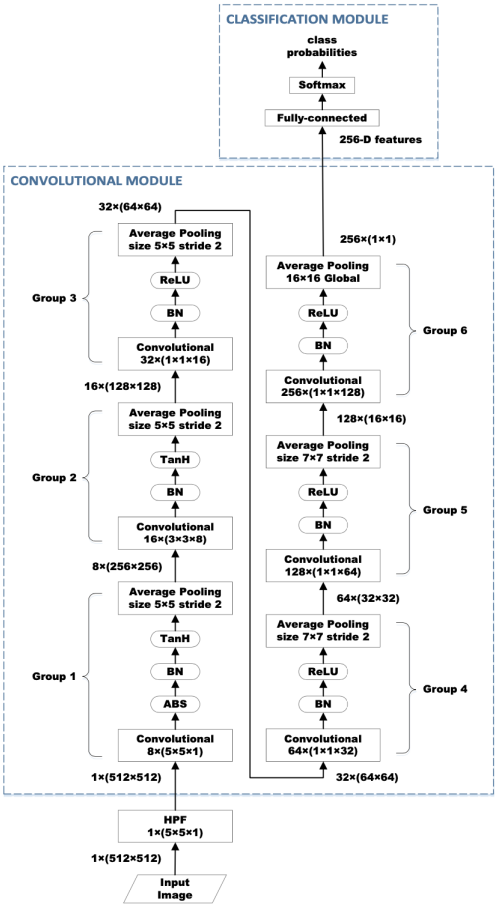

Fig. 1. Model - A

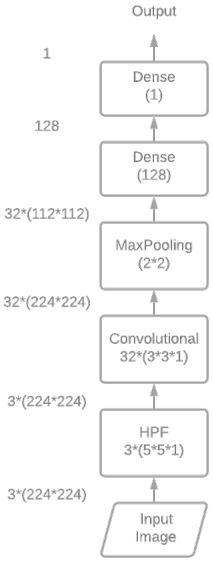

Fig. 2. Model - B

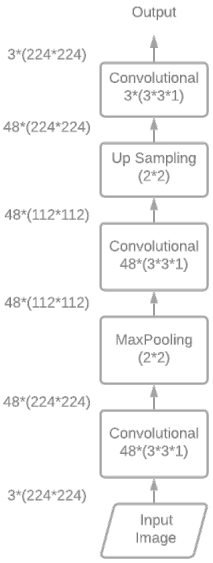

Fig. 3. Model - C

4.3. Model C

Once we have detected an image to be a Stego image, we would like to remove the steganography and clean the image. This is important as if we stop stego images from being uploaded online, then people might find missuge this policy and convert any image they do not want to be posted online into a stego image. So to achieve this, we use an autoencoder to clean the images. The autoencoder was trained with F5 stego images as the input and cover images as the target output. The aim of this autoencoder is to get rid of steganography while still preserving the image quality. So the autoencoder needs large feature map so that no information loss occurs during the encoding process. The autoencoder cannot be too deep as it must train completely with the available dataset. The architecture of this model is shown in Fig. 3.

5. EXPERIMENTS/RESULTS/DISCUSSION

5.1. Model A

From Fig. 4. we can see that the validation curves for loss and accuracy are not improving gradually and have lots of spikes. This is a clear indication that the model is not converging. To alleviate this problem we tried removing the pre-processing HPF (Fig. 5.) and adding dropout layers (Fig. 6.) but the results didn't improve. By comparing Fig. 4. and Fig. 5 we can clearly see that the pre-processing is improving the results.

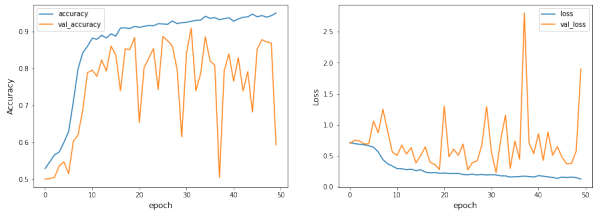

Fig. 4. Model - A - without HPF

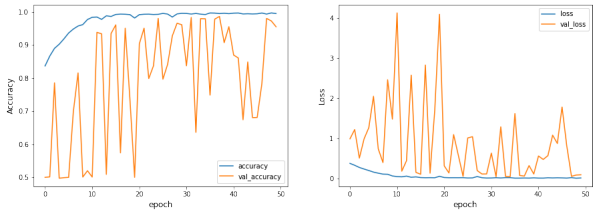

Fig. 5. Model - A - with HPF

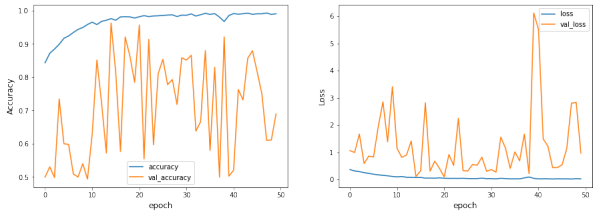

Fig. 6. Model - A - with HPF and 30% dropout layers

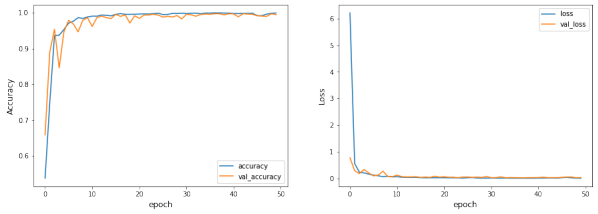

Fig. 7. Model - B - without HPF

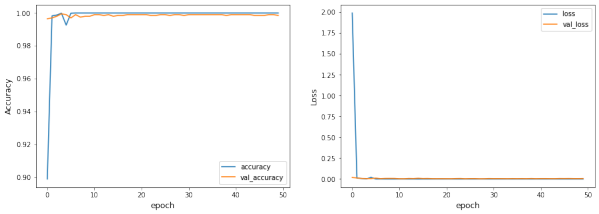

Fig. 8. Model - B - with HPF

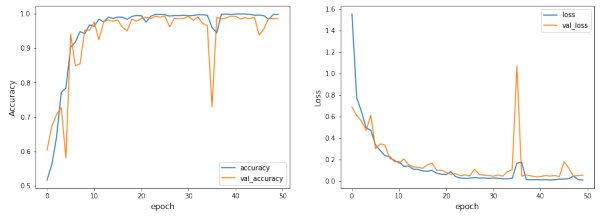

Fig. 9. Model - B - without HPF and with slow learning rate

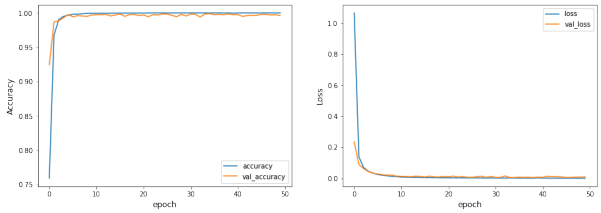

Fig. 10. Model - B - with HPF and with slow learning rate

5.2. Model B

The results of model B can be seen in Fig. 7 and Fig. 8. The accuracy and loss show a almost smooth/steady improvement. The model achieves high accuracy and low loss very quickly and it is difficult to compare the performance of the model with and without pre-processing.

Fig. 9 and Fig. 10 display the loss and accuracy curves for the same model trained at a lower learning rate. The differences are clearly visible. The accuracy and loss curves are much more smooth when HPF pre-processing is used. This demonstrates the effectiveness of HPF pre-processing for JPEG steganalysis.

5.3. Model C

For the auto-encoder we are plotting the mean-squared error loss function in Fig. 11. The model doesn't take a lot of epochs to achieve near steady-state loss. The final loss achieved was of the order 10^{-3}. 1000 Stego images and 1000 cover images were cleaned using the Autoencoder (Model - C) and detected using Model - B.

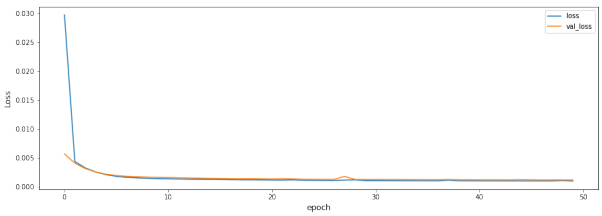

Fig. 11. Model - C - Loss

Input	Not Cleaned	Cleaned
Stego Images	992	4
Cover Images	2	1

Table 1. Number of Detected Stego Images

6. CONCLUSION

Detecting the F5 Steganography algorithm is difficult task for conventional statistical approaches, but easy for Neural Networks. An accuracy of 99.85% clearly demonstrates the effectiveness of using Neural Networks for Steganalysis. Such extreme high accuracy is uncommon in regular classification problems of machine learning, but Steganalysis is fundamentally different from these problems. Further, the use of HPF is beneficial for our models. Our image cleaner using autoencoding also gives high cleaning rate of upto 98.6%.

7. CONTRIBUTIONS

Everyone has contributed similar amounts in the project. Arpan was mainly responsible for model B, Tanmay for model C and parts of model A and Zhouyang for most of model A. Further, all 3 made changes to dataset generation as they had different needs and the final code satisfies all the needs. For literature review also we divided the task in 3 parts, each assigned equal number of papers. In the report, sections were written respectively by people who were responsible for those sections during experiments.

8. REFERENCES

[1] Andreas Westfeld. F5—a steganographic algorithm. In *International workshop on information hiding*, pages 289–302. Springer, 2001.

[2] Jessica Fridrich and Miroslav Goljan. Practical steganalysis of digital images: state of the art. In *Security and Watermarking of Multimedia Contents IV*, volume 4675, pages 1–13. International Society for Optics and Photonics, 2002.

[3] George Berg, Ian Davidson, Ming-Yuan Duan, and Goutam Paul. Searching for hidden messages: Automatic detection of steganography. In *IAAI*, pages 51–56, 2003.

[4] Guanshuo Xu. Machine learning based digital image forensics and steganalysis. 2017.

[5] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)*, pages 2752–2756. IEEE, 2016.

[6] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.