A Deep Learning Framework to identify real-world stego images

MSc Research Project
Data Analytics

Khushboo Lavania
Student ID: x19209835

School of Computing
National College of Ireland

Supervisor: Dr. Paul Stynes, Dr. Pramod Pathak

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Khushboo Lavania
Student ID:	x19209835
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Paul Stynes, Dr. Pramod Pathak
Submission Due Date:	16/08/2021
Project Title:	A Deep Learning Framework to identify real-world stego images
Word Count:	5062
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Khushboo Lavania
Date:	September 19, 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Learning Framework to identify real-world stego images

Khushboo Lavania
x19209835

Abstract

Image Steganalysis is the process of identifying images that have been processed using Steganography to hide information or messages. These images are called stego images. Identifying real-world stego images which vary in dimensions, scalability, lightning conditions with unknown steganography algorithms, payload, embedding capacity is a challenging task. This research proposes a deep learning framework that uses Convolutional Neural Network (CNN) to identify such real-world stego images. The framework consists of two pre-trained models namely InceptionNet V3 and EfficientNet B3. Models are trained using the ALASKA2 dataset and are evaluated using model accuracy. Additionally, evaluation metrics like Precision, Recall, and F1 scores are also calculated. Inception V3 achieves an accuracy of 73.33% while the performance of EfficientNet B3 is slightly better with an accuracy of 79.43%. Identification of real-world stego images will be of great benefit to the government security department to determine any illegal activity.

1 Introduction

The advancement of the digital era has both pros and cons. As per government officials, in the suicidal attack of 9/11, the terrorists used video steganography for information sharing[1]. There are many evident examples in which stegomalware[2] that uses steganography is used for Phishing attacks[3]. Steganalysis is the only way to be exempted from the misuse of steganography.

The technique of concealing a message behind a video, text, or image is called Steganography (Nolkha et al.; 2020). In this technique, the image used for embedding a message is called a cover image, and the message embedded is termed as Payload (Subramanian et al.; 2021). Stego image is generated after the message is hidden in the cover image. The maximum number of bits that can be embedded in an image is called its embedding capacity. Steganography modifies the statistical properties of an image, therefore the payload could not be identified by the human eye. Image Steganalysis is just the opposite of steganography. Steganalysis involves the identification of images that have been treated using steganography. Figure 1 describes the process of steganography and steganalysis.

[1] https://www.bbc.com/news/world-24784756

[2] https://en.wikipedia.org/wiki/Stegomalware

[3] https://www.pluribus-one.it/company/blog/84-cybersecurity/83-stegomalware

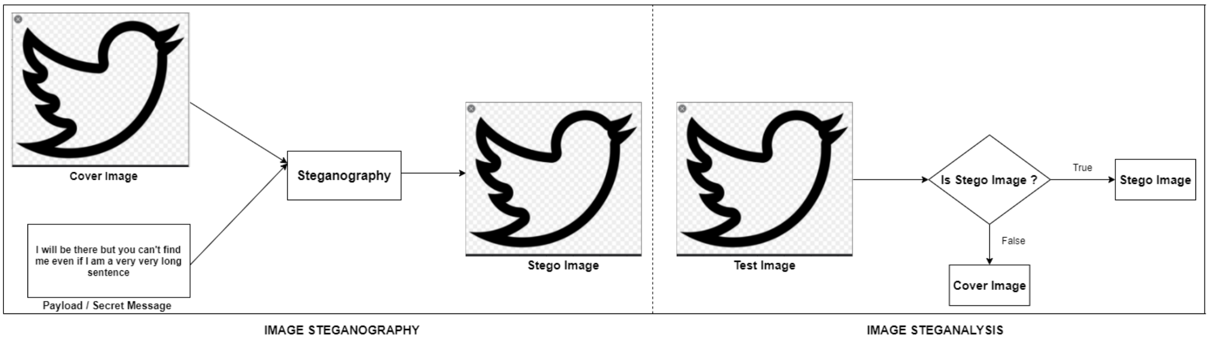


Figure 1: Steganography and Steganalysis Process

These days CNN, a state of art machine learning algorithm, is used to determine stego images with the least significant classification error. But such models use ideal images with minimal or no noise (Rezaei et al.; 2020). If the size and brightness of these images are altered, then steganalysis would be very difficult (You et al.; 2021) because brightness, size, color, etc. modifies the statistical property of an image. An efficient steganalysis model should precisely recognize such real-world stego images having varying statistical properties without any prior information on how steganography was performed and leads to the research question **"How accurately and effectively real-world stego images containing hidden messages could be classified from a given set of images using Deep Convolutional Neural Networks?."**

For this research, the ALASKA2 dataset which consists of images captured using 50 different cameras is used. Due to computational complexity restriction, a subset of the dataset is considered for the study. Deep Neural Networks i.e. InceptiontNet V3 and EfficientNet B3 are used for the classification of these images as stego or cover images. The major contribution of this research is a novel deep learning framework that uses the concept of transfer learning to effectively classify real-world colored stego images.

This paper discusses related work in Section 2 with a focus on machine learning approaches in stego image classification. Section 3 describes the research methodology used for this research. Section 4 outlines the design specification followed by Section 5 in which implementation of the models is discussed. The research is concluded by Evaluation, Conclusion, and Future Work in Sections 6 and 7.

2 Related Work

Image Steganalysis is categorized into two types i.e. Spatial and Transform domain. In the Spatial domain, images in which the message is embedded in the pixel intensity of the image is identified whereas the transform domain steganalysis involves identifying images in which the payload is embedded using Discrete Wavelet Transform (DWT), Discrete Cosine Transformation (DCT), or Discrete Fourier Transform (DFT) (Nolkha et al.; 2020) steganography algorithms.

2.1 Traditional Machine Learning Algorithms for Image Steganalysis

Conventionally, machine learning techniques such as Random Forest or Logistic Regression are used to identify stego images. But Selvaraj et al. (2021), stated that it is a complex task as it divides the process into two steps. On contrary, CNN combines both the steps and makes the process less tedious. Xu et al. (2016) designed the first CNN model and asserted that it can be used both in the Spatial and Transform domain. But the performance of the model was not adequate in comparison to other steganalysis techniques. Yet the author highlights that a properly designed CNN can beat the performance of other stego image detectors. Shankar and Azhakath (2019) experimented on INRIA and UCID datasets. Support Vector Machine (SVM) with Particle Swarm Optimisation is used for the classification task. For performance evaluation, six different kernels were used. ANOVA kernel outperforms in comparison to other kernels but, only the LSB matching algorithm in the Spatial domain and the F5 algorithm in the Transform domain are used for hiding the payload in the image. Nowadays, image steganography is used on medical images like Digital Imaging and Communications in Medicine (DICOM) to hide a patient's name, age, and other confidential information. Maroof Özcan et al. (2020) used SVM to identify such DICOM images in which a payload is embedded using a steganography algorithm. The classification accuracy of the model is 99.28% and the correlation coefficient is 0.9856. However, the experiment was conducted only on those images in which the message is embedded by altering the Least Significant Bit (LSB) of an image.

2.2 CNN for Image Steganalysis

2.2.1 CNN in Spatial Domain

Zhang et al. (2020) designed a CNN model named "Zhu-Net". The standard BOSSBase 1.01 and BOWS2 datasets are used. In the preprocessing layer, the size of the convolutional layers has been modified from 5X5 to 3X3. To further improve the classification accuracy of the model the forward-backward-gradient descent approach is used in the convolutional layer. To maximize feature extraction, Spatial pyramid pooling is used before passing them to the fully connected network. It has been observed that Zhu-Net outperforms when it is used on images that are treated with HILL, WOW, and S-UNIWARD spatial steganography algorithms. To enhance the classification accuracy in the Spatial Domain, another CNN named "GBRAS-Net" has been designed by Reinel et al. (2021). The experiments have been performed on the same datasets. In the preprocessing step, the convolutional layer contains 30 SRM filters of size (1,1), 3TanH as an activation function, and the padding as the same. Global Average Pooling and softmax activation function has been used in the classification stage. The architecture receives significant accuracy as compared to other state-of-art models. The author suggests the use of the ALASKA2 dataset as future work to enhance the performance of the feature extraction step. The work done by the author has been replicated in this research and the cue suggested has also been implemented as another experiment.

The shortcomings of other CNN models for image steganalysis have been highlighted in the study by Kato et al. (2020). Stego images which are deliberately reduced by nearest-neighbor interpolation before applying steganography are tough to recognize due to an increase in the spatial frequency. To overcome this hassle, in the preprocessing step an additional stego signal has been embedded both in the reduced cover and stego image. The standard BOSSBase 1.01 dataset is used for the experiment and an accuracy of 89.4% has been achieved by the model. But only images that are treated with S-UNIWARD and WOW steganography algorithms with the same payload are considered for the experiment. You et al. (2021) focuses on some more struggles which make the task of steganalysis even more challenging. As per the author changes in the content, size, lightning, or the shooting equipment puts a great impact on the statistical properties of an image which makes the identification task even more challenging. Therefore, for the study, arbitrary size images from ALASKA2 and the BOSSBase 1.01 dataset are considered. The association between different sub-regions of an image is identified while designing the model. The classification accuracy of the model is satisfactory in comparison to the other models on the unaltered image. But as per the author, more research needs to be done in designing an ideal model for arbitrary size images. This research is difficult to replicate because the methodology used is not properly described.

2.2.2 CNN in Transform Domain

Mohamed et al. (2020b) highlighted some of the key points that can play an important role in researching stego images. As per the author, more work has been done on greyscale images and research needs to be done on a realistic database of color images. J-UNIWARD and UED are the common steganography algorithms that are considered in most of the researches. UCID, BOSSbase, BOWS2, and ALASKA are some of the popular datasets for image steganalysis. The author strengthens on the fact that CNN should be analyzed for further study in comparison to conventional methods and more work needs to be done in the transform domain steganalysis.

2.2.3 CNN in Spatial and Transform Domain

Internet of Things (IoT) is widely used in many areas. But, these applications are less secure due to steganography. Therefore, a study has been done by the author Mohamed et al. (2020a) to discover the breaking point of the most secure steganography algorithm i.e. J-UNIWARD using CNN. The standard BOSSbase dataset has been used. The CNN model designed consists of a convolutional layer, batch normalization layer, average pooling layer, ReLU layer, fully connected layer, and a softmax layer. The experimental result shows that the breaking point of the J-UNIWARD algorithm is 2 bpnzAC. But the experiment has been done on a smaller dataset and more research needs to be done on images with a payload less than 0.5 bpnzAC. To further improve the classification accuracy of the CNN model, Lichy et al. (2020) proposed a model which includes 3 convolutional layers and 2 dense layers. Tanh activation function has been used in the convolutional layers whereas softmax activation function has been used in the dense layers. CIFAR-10 dataset has been used for training the CNN model in which the images are treated with WOW, UNIWARD, and HUGO steganography algorithms. In terms of the total error rate, the performance of the proposed CNN is better than state-of-art models. But the experiment has been performed on greyscale images. Just like most of the researchers Lu et al. (2019), used BOSSBase 1.01 dataset to train and test the CNN model. The author highlighted that by using truncation linear activation function (TLU) in the pre-processing step, the performance of the model could be improved and in turn increases the training speed of the model. But since these images are ideal and do not contain any kind of noise, the author recommends using data augmentation techniques for future studies.

2.2.4 CNN containing Additional techniques

A dual CNN model has been designed by the author Kim et al. (2019), to identify stego images. This model consists of two parallel CNNs and an additional input image in which information is embedded using the S-UNIWARD steganography algorithm. Just like most of the researchers BOSSBase 1.01 dataset has been used for the study. The accuracy of the model is 80.43%. Since this dual CNN architecture has been applied on the conventional CNN, so the performance of the model is limited. The author suggested studying the dual CNN architecture on the current state-of-art models. Just like Kim et al. (2019), the author Wu et al. (2018) also used BOSSBase 1.01 dataset for designing the CNN model for the identification task but with a little twist. Firstly, the CNN model is denser compared to other models which are beneficial in extracting weak stego signals. Secondly, by using the concept of residual learning the author helps to preserve even a very slight stego signal which has a positive influence on the performance of the model. The model fails to identify images in which the message is embedded using a compressed domain steganography algorithm.

CNN-based image steganalysis models require huge storage and are computationally expensive. Training and deploying these models is a difficult task. To handle these issues, Tan et al. (2021) proposed a CNN model named CALPA-NET. The model prunes the network architecture which in turn reduces its computational cost. Research has been carried out on BOSSBase, BOWS2, ALASKA, and CLS-LOC datasets. The proposed model uses a popular deep learning technique named Network Pruning. CALPA-NET successfully reduces the computational complexity of the model by 2% in comparison to other models. The road blocker in developing an image steganalysis model as faced by many other researchers such as Kato et al. (2020), You et al. (2021), Mohamed et al. (2020a), Kim et al. (2019), Wu et al. (2018) is the size of the dataset. The performance of the CNN model is greatly influenced by the number of images used to train the model. Yedroudj et al. (2020) introduced a data augmentation technique to deal with such an issue called "pixel-off". It embeds the noise in an image in such a way that its pixel distribution remains unaltered. Mohammadi et al. (2021) highlights the importance of developing a universal steganalysis model. But developing a universal image steganalysis model will lead to a problem of Curse of dimensionality (CoD) as it requires the model to extract as many features as possible. To avoid this issue, the author suggested the use of evolutionary algorithms such as Artificial Bee Colony, Particle Swarm Optimization, Firefly Algorithms, Grey Wolf Optimizer, and Pine Tree Optimization.

Conclusion: By examining all these researches it can be concluded that the approach of Transfer Learning has not been used by many researchers for developing an image steganalysis model. Also, state-of-the-art indicates that a well-designed CNN effectively identifies ideal grey-scale images but not enough research has been carried out in recognizing real-world colored stego images that vary in size, brightness, pixel distribution with unknown steganography algorithms, hidden payload, and embedding capacity. Hence this study focuses on determining such real-world colored stego images using transfer learning models.

3 Methodology

This section focuses on the data mining methodology used for achieving the desired objective i.e. identification of stego images. The steps are planned, organized, and accomplished using the CRISP-DM methodology. The figure 2 below describes all the stages of the research methodology adopted:

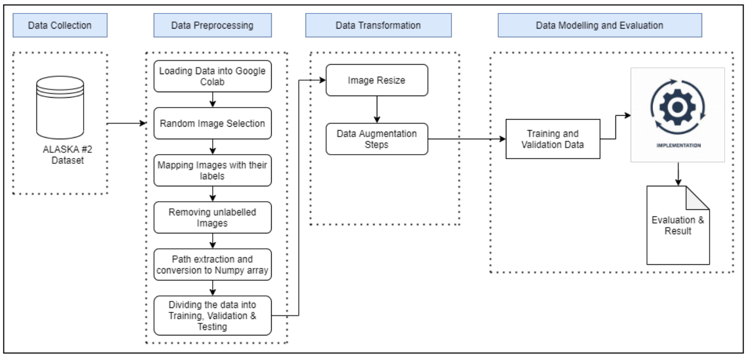

Data Collection
Data Preprocessing
Data Transformation
Data Modelling and Evaluation
ALASKA #2 Dataset
Loading Data into Google Colab
Random Image Selection
Mapping Images with their labels
Removing unlabelled Images
Path extraction and conversion to Numpy array
Dividing the data into Training, Validation & Testing
Image Resize
Data Augmentation Steps
Training and Validation Data
IMPLEMENTATION
Evaluation & Result

Figure 2: Process Flow

3.1 Data Collection

To imitate the real-world scenario the images from ALASKA 2[4] dataset is used. The public usage license is available for this dataset. The images are acquired using 50 different cameras including smartphone and high profile cameras which are processed differently. There is no information regarding the payload and its respective length. To make the identification task more challenging, images with smooth content are embedded with a short message, and more bits are hidden in high-textured images which implies that each image is different from the other.

3.2 Data Pre processing and Transformation

Data preprocessing and transformation is a crucial step before applying any deep learning algorithm. In reality, data consist of noise, and handling noisy data is a difficult task. It increases the computational time, reduces the model performance, and is more prone to errors. Hence handling such issues is an essential step before implementing the model on data. The steps followed for data pre-processing and transformation are discussed below:

- The dataset is quite large and contains more than 305,000 images. Due to computational complexity restriction, 20,000 images are randomly selected from the ALASKA #2 dataset using Numpy randint function.

- The images are linked with their respective labels i.e. either cover or stego. It is a crucial step for appropriate model training. Python Numpy library is used to map images with their corresponding label.

- Few images of the dataset which were not labeled correctly as stego or cover have been removed.

[4] https://www.kaggle.com/c/alaska2-image-steganalysis/data

- The images and their labels are converted into a Numpy array to enhance model performance.

- The images are resized into 256x256 using OpenCV resize library function.

- The prediction accuracy of the deep learning model is greatly influenced by the amount and diversity of data provided during model training [5]. The dataset used for this research is quite vast as it includes real-world images varying in resolution, pixel intensity, brightness and contains noise. But, apart from that different data augmentation techniques using ImageDataGenerator function of TensorFlow library as described in the below figure 3 are applied to the data to make sure that the model is trained with as much variety of images as possible. These include enabling horizontal_flip=True, fill_mode=nearest, zoom_range=0.2, shear_range=0.2, rescale=1./255, height_shift_range=0.2, width_shift_range=0.2,rotation_range=40.

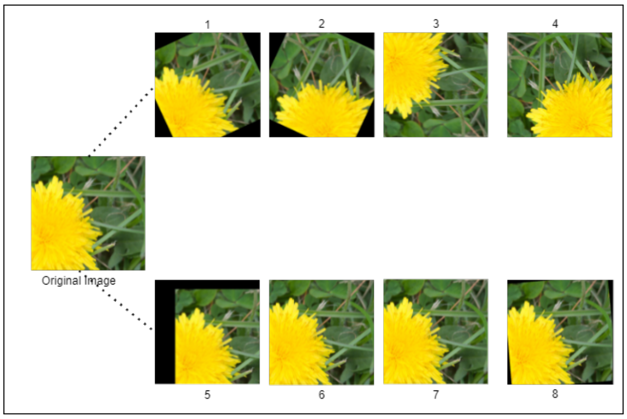


Figure 3: Data Augmentation - 1) Random Rotation 2) Shearing 3) Horizontal Flip 4) Vertical Flip 5) Horizontal Shift 6) Rescaling 7) Random Zoom 8) Vertical Shift

- The augmented data is divided into 80:20 ratio using Sklearn library in which 80% of the data is used for training and the remaining 20% is used for testing the performance of the trained model.

3.3 Model Implementation

For the fourth step of model implementation, two pre-trained models namely InceptionNet V3 and EfficientNet B3 are loaded using the Keras TensorFlow library. Both the models are trained using 20,000 images while their performance is evaluated by 2000 images. The detailed specification of the implemented models are discussed in the section 4.

3.4 Model Evaluation and Result

The last step, Evaluation, and Result involve evaluating the performance of the trained model using accuracy, precision, and Recall. The performance of the evaluated models is discussed in the section 6.

[5] https://www.mygreatlearning.com/blog/understanding-data-augmentation/

4 Design Specification

This section focusses models implemented to identify real-world coloured stego images:

4.1 GBRAS-Net based CNN

In an attempt to replicate the state-of-art model for image steganalysis performed by the author Reinel et al. (2021), GBRAS-Net is implemented. The popular BOSSBase 1.01[6] dataset which contains around 10,000 greyscale images is used. The size of the image is processed and changed to 256 X 256 from 512 X 512 as done in the original work. The preprocessing layer of the CNN contains 30 SRM filters, padding as same, strides of (1,1), and 3TanH as an activation function. In the feature extraction stage, 6 convolutional layers, padding as same, and Exponential Linear Unit as an activation function is used. Further in the classification stage softmax activation function is used for predicting the category of an image. The model uses a batch size of 32, a learning rate of 0.001, and an Adam optimizer. The model is executed for 100 epochs.

4.2 Transfer Learning based InceptionNet V3

Inception V3 is computationally less expensive in comparison to many other pre-trained networks with respect to the number of parameters generated and computational expense such as memory and training time (Szegedy et al.; 2016). Computational expense is the biggest difficulty in developing a model for image steganalysis. Due to the adaption of several techniques like factorized smaller and asymmetric convolutions, auxiliary classifier, and grid-size reduction the network architecture gets optimized. It helps in smoother model adaption and reducing the computational cost. The ALASKA2 real-world images are used for training and testing the performance of the model. An additional global average pooling layer, dropout, and dense layer with sigmoid as an activation function are further added to the network. The hyperparameters used are the batch size of 32, binary cross-entropy as an activation function, learning rate as 0.0001, and Adam as optimizer. The training and validation dataset is passed through the Image Data generator which generates real-time augmented images. The model implemented can be referred from the figure 4.

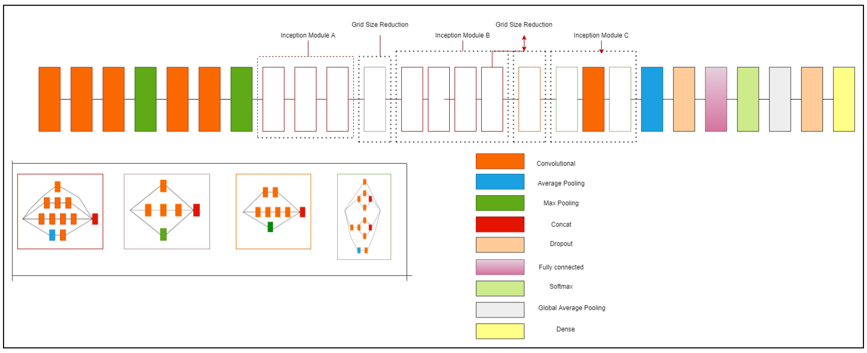


Figure 4: InceptionNet V3 Architecture

[6] http://agents.fel.cvut.cz/boss/index.php?mode=VIEWtmpl=materials

4.3 Transfer Learning based EfficientNet B3

EfficientNet is based upon the principle of scaling up the CNN model. The EfficientNet family consists of 7 models, but for this research EfficientNet, B3 is used. This model maintains a perfect balance between accuracy and computational resource (Alhichri et al.; 2021). The ALASKA2 real-world images are used for training and testing purposes. An additional global average pooling layer and dense layer with sigmoid as an activation function have been used. The hyper-parameters used are the batch size of 32, binary cross-entropy as an activation function, learning rate as 0.0001, and Adam as optimizer. To train the model with a diverse set of images, an image generator is used to provide real-time augmented images. The model implemented can be referred from the figure 5.

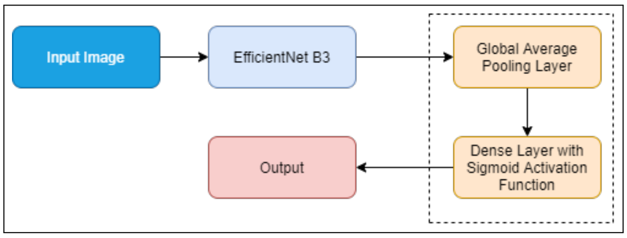

Input Image
EfficientNet B3
Global Average Pooling Layer
Dense Layer with Sigmoid Activation Function
Output

Figure 5: EfficientNet B3 Architecture

5 Implementation

A comprehensive summary of the steps followed while implementing GBRAS-Net and pre-trained InceptionNet and ResNet models are discussed here. The data preparation, pre-processing, and model implementation are done on Google Colab Pro. It is a Jupyter Notebook environment that runs entirely on a cloud server. Python is used as a scripting language as it renders support to many deep learning frameworks[7]. Google Drive is another cloud-based platform which is used for data storage. Deep learning framework Keras which serves as an interface for TensorFlow is installed for model processing. Some additional libraries such as Numpy, Pandas, matplotlib, OpenCV2 are also installed for some additional operations. Model performance parameters such as accuracy, and loss is stored in a CSV file after each experiment and the best performing model is saved on the Google drive for future reference using the Keras ModelCheckpoint function. To evaluate the performance of the trained model the best performing model is loaded from the Google drive using the Keras load_weights function.

6 Evaluation

After model implementation, the next stage is performance evaluation. The performance of the model could be evaluated using different evaluation metrics. But for this research, Accuracy, Precision, and Recall are used as evaluation metrics. As recommended by

[7] https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7

Mohamed et al. (2020b), accuracy should be considered as an evaluation metric to determine the performance of the image steganalysis model. Detailed evaluation of all the experiments are discussed below:

6.1 Baseline Experiment: GBRAS-Net based CNN

This experiment is the replication of the state-of-art by the author Reinel et al. (2021) in which BOSSBase 1.01[8] dataset that contains 10,000 greyscale images is used. The implemented model achieves an accuracy of 71.23% which is 16% less than the accuracy achieved by the model in the original work. The reason behind this difference is the processing steps implemented on the dataset before applying the GBRAS-Net model are not mentioned in the research paper. Due to which the image processing could not be done accordingly for this experiment. Moreover, the author used a processor with 128GB RAM, while the processor used for this research is 16GB RAM. But the experiment has been performed on the ideal grey-scale images and the real-world colored images containing noise are not considered. Therefore further experiments in this research work have been performed considering this aspect.

6.2 Baseline Experiment: GBRAS-Net based CNN on ALASKA 2 Dataset

This experiment is motivated by the architecture of GBRAS-Net. As recommended by the author Reinel et al. (2021), more research needs to be done in the identification of stego images considering the ALASKA2 dataset. Hence, in this experiment, the architecture of GBRAS-Net is implemented on the ALASKA2 dataset. The accuracy of the model is 64.48% which is around 23% less than the accuracy achieved by the original model. It is because the original experiment was performed on the BOSSBase 1.01 dataset which contains grey-scale images while the ALASKA2 dataset contains real-world colored images bearing noise and altered statistical properties. Therefore, further work has been done using the ALASKA2 dataset to improve the classification accuracy.

6.3 Transfer Learning based Inception V3

The accuracy of the InceptionNet V3 model on real-world colored images is 73.33%. The accuracy and the loss learning curve for the training and the validation dataset could be referred from the below figure 6. There seems to be some inconsistency in both, accuracy and loss curve. It signifies overfitting which is due to the false impression of images by the validation dataset. The model is trained using only 20000 images due to computational complexity restriction. By increasing the size of the dataset, this problem could be avoided.

6.4 Transfer Learning based EfficientNet B3

The accuracy of the EfficientNet B3 model on the real-world colored images is 79.43%. The accuracy and the loss learning curve for the training and the validation dataset could be referred from the below figure 7. The accuracy and loss curve is more consistent in comparison to the accuracy and loss curve of the InceptionNet model. It signifies that

[8] http://agents.fel.cvut.cz/boss/index.php?mode=VIEWtmpl=materials

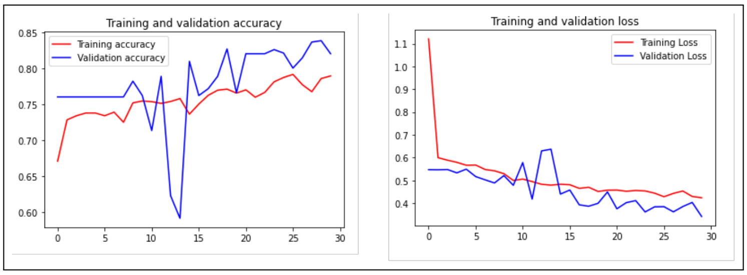


Figure 6: Accuracy and Loss Learning Curve for InceptionNet V3

the performance of the EfficientNet B3 model is better compared to the InceptionNet V3 model.

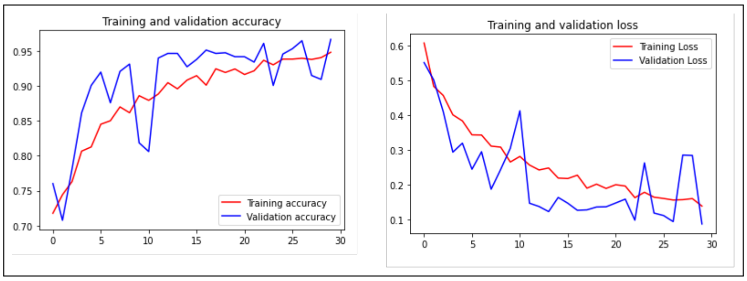


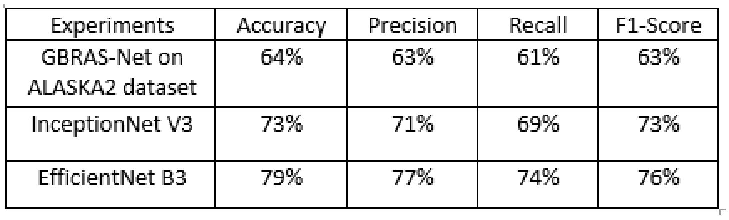
Figure 7: Accuracy and Loss Learning Curve for EfficientNet B3

6.5 Model Comparison, Discussion & Final Result

The models are trained and tested multiple times to determine the best-performing model to identify real-world stego images. Pre-trained models namely InceptionNet V3 and EfficientNet B3 are implemented on the ALASKA2 dataset. The research is initiated by implementing the state-of-art model for image steganalysis. GBRAS-Net-based CNN architecture was implemented on BOSSBase 1.01 dataset which contains grey-scale images. This experiment act as a basic foundation for other experiments. Another experiment is carried out using the ALASKA2 dataset but with the same methodology as used in the state-of-art model.

Moreover, this research proposes pre-trained CNN i.e. InceptionNet V3 and EfficientNet B3 to identify the real-world colored images containing noise with an unknown steganography algorithm. As observed from the learning curves of InceptionNet V3 from figure 6, the training and the loss curve shows an inconsistent spike. It is because of the misinterpretation of images by the validation dataset. The issue could be avoided by including more images for the experiment. The learning curve of EfficientNet B3 as seen from figure 8 is more consistent in comparison to the InceptionNet V3 model. It signifies that the performance of the EfficientNet B3 model is better than InceptionNet V3 but could be further improved by considering a larger dataset. Apart from accuracy, other evaluation parameters such as Precision, Recall, and F1-Score are also considered to determine the best performing model.

It can be observed from the table 8, that the performance of the EfficientNet B3 model on real-world noisy images is better in comparison to the other two models. The



Experiments	Accuracy	Precision	Recall	F1-Score
GBRAS-Net on ALASKA2 dataset	64%	63%	61%	63%
InceptionNet V3	73%	71%	69%	73%
EfficientNet B3	79%	77%	74%	76%

Figure 8: Model Comparison

main limitation of this project is computational resources. If rather than considering the subset of images from the ALASKA2 dataset, the whole dataset would have been considered, then the performance of the models could be improved.

7 Conclusion and Future Work

The main purpose of this project is to determine real-world stego images whose payload, steganography algorithm, and embedding capacity are unknown. Such real-world images also contain noise. The ALASKA2 dataset is used for training and testing the performance of the model. Two pre-trained models namely InceptionNet V3 and EfficientNet B3 are implemented to identify the real-world stego images. The accuracy of the InceptionNet V3 model is 73.33% while the accuracy of the EfficientNet B3 model is 79.43%. Other parameters such as Precision, Recall, and F1-Score are also evaluated to determine the best-performing model. The performance of the EfficientNet B3 model is better in comparison to other model. But due to computational complexity, the subset of the ALASKA2 dataset is used.

In the future, this research could be extended by using the complete ALASKA2 dataset to improve the accuracy of the image steganalysis model. Moreover, the payload could be retrieved from the image and Natural Language Processing could be applied to analyze the sentiment of the hidden message to imply the context of the hidden message.

Acknowledgement

I would like to express my sincere thanks and gratitude to my supervisors Dr. Paul Stynes and Dr. Pramod Pathak for their constant guidance, and feedback. They both have been a constant support and motivation throughout the semester. I am also thankful to my friends and family for their constant motivation.

References

Alhichri, H., Alswayed, A. S., Bazi, Y., Ammour, N. and Alajlan, N. A. (2021). Classification of remote sensing images using efficientnet-b3 cnn model with attention, *IEEE Access* **9**: 14078–14094.

Kato, H., Osuge, K., Haruta, S. and Sasase, I. (2020). A preprocessing methodology by using additional steganography on cnn-based steganalysis, *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6.

Kim, J., Kang, S., Park, H. and Park, J.-I. (2019). Dual convolutional neural network for image steganalysis, *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–4.

Lichy, K., Lipinski, P. and Grzelak, M. (2020). Deep convolutional network for steganalysis of hugo, wow, and uniward algorithms, *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1–4.

Lu, Y. Y., Yang, Z. L. O., Zheng, L. and Zhang, Y. (2019). Importance of truncation activation in pre-processing for spatial and jpeg image steganalysis, *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 689–693.

Maroof Özcan, F. B., Karakiş, R. and Güler, (2020). Steganalysis on medical images with support vector machine, *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4.

Mohamed, N., Rabie, T. and Kamel, I. (2020a). Iot confidentiality: Steganalysis breaking point for j-uniward using cnn, *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–4.

Mohamed, N., Rabie, T. and Kamel, I. (2020b). A review of color image steganalysis in the transform domain, *2020 14th International Conference on Innovations in Information Technology (IIT)*, pp. 45–50.

Mohammadi, F. G., Shenavarmasouleh, F., Amini, M. H. and Arabnia, H. R. (2021). Evolutionary algorithms and efficient data analytics for image processing, *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–8.

Nolkha, A., Kumar, S. and Dhaka, V. S. (2020). Image steganography using lsb substitution: A comparative analysis on different color models, *in* A. K. Somani, R. S. Shekhawat, A. Mundra, S. Srivastava and V. K. Verma (eds), *Smart Systems and IoT: Innovations in Computing*, Springer Singapore, Singapore, pp. 711–718.

Reinel, T.-S., Brayan, A.-A. H., Alejandro, B.-O. M., Alejandro, M.-R., Daniel, A.-G., Alejandro, A.-G. J., Buenaventura, B.-J. A., Simon, O.-A., Gustavo, I. and Raúl, R.-P. (2021). Gbras-net: A convolutional neural network architecture for spatial image steganalysis, *IEEE Access* **9**: 14340–14350.

Rezaei, M., Riahi, M. and Hayati, H. (2020). Stegrt1: A dataset for evaluating steganalysis systems in real-world scenarios, *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–5.

Selvaraj, A., Ezhilarasan, A., Wellington, S. L. J. and Sam, A. R. (2021). Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning based techniques, *IET Image Processing* **15**(2): 504–522.
URL: *https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12043*

Shankar, D. D. and Azhakath, A. S. (2019). Steganalysis of minor embedded jpeg image in transform and spatial domain system using svm-pso, *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 46–49.

Subramanian, N., Elharrouss, O., Al-Maadeed, S. and Bouridane, A. (2021). Image steganography: A review of the recent advances, *IEEE Access* **9**: 23409–23423.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016). Rethinking the inception architecture for computer vision, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Tan, S., Wu, W., Shao, Z., Li, Q., Li, B. and Huang, J. (2021). Calpa-net: Channel-pruning-assisted deep residual network for steganalysis of digital images, *IEEE Transactions on Information Forensics and Security* **16**: 131–146.

Wu, S., Zhong, S. and Liu, Y. (2018). Deep residual learning for image steganalysis, *Multimedia tools and applications* **77**(9): 10437–10453.

Xu, G., Wu, H. and Shi, Y. (2016). Structural design of convolutional neural networks for steganalysis, *IEEE Signal Processing Letters* **23**(5): 708–712.

Yedroudj, M., Chaumont, M., Comby, F., Oulad Amara, A. and Bas, P. (2020). Pixels-off: Data-augmentation complementary solution for deep-learning steganalysis, *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, IHMMSec '20, Association for Computing Machinery, New York, NY, USA, p. 39–48.
URL: *https://doi.org/10.1145/3369412.3395061*

You, W., Zhang, H. and Zhao, X. (2021). A siamese cnn for image steganalysis, *IEEE Transactions on Information Forensics and Security* **16**: 291–306.

Zhang, R., Zhu, F., Liu, J. and Liu, G. (2020). Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis, *IEEE Transactions on Information Forensics and Security* **15**: 1138–1150.