National College of Ireland

Targeted Detection of Steganographic Content in Images Using Transfer Learning to Enhance Cybersecurity

MSc Research Project
Data Analytics

Nisarga Revannaradhya
Student ID: x23110848

School of Computing
National College of Ireland

Supervisor: Dr. Muslim Jameel Syed

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nisarga Revannaradhya
Student ID:	x23110848
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr. Muslim Jameel Syed
Submission Due Date:	12/08/2024
Project Title:	Targeted Detection of Steganographic Content in Images Using Transfer Learning to Enhance Cybersecurity
Word Count:	5876
Page Count:	20

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:	
Date:	16th September 2024

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):	

Targeted Detection of Steganographic Content in Images Using Transfer Learning to Enhance Cybersecurity

Nisarga Revannaradhya
x23110848

Abstract

Steganography is a technique used to conceal information within digital media while making it unrecognizable. It can serve for both secure communication and malicious purposes, such as embedding malware in images for cyberattacks. Through image steganography, malicious code can be hidden in natural images without arousing suspicion. Steganalysis is the counter process of detecting steganographic content. This work introduces an image steganalysis approach designed to identify images embedded with JavaScript code using spread spectrum steganography. This is done by combining traditional feature extraction methods with pretrained models, enhanced by an attention mechanism. However, despite the advancements, CNNs still tend to have some difficulty in recognizing subtle changes introduced by spread-spectrum steganography, particularly when handling JavaScript code embedding. This limitation highlights the value of the combined approach implemented in this research. Among others, EfficientNet-b0 model achieved 94.7% accuracy when trained on ImageNet, thus proving the effectiveness of the combined approach. This approach can serve as a benchmark to detect the steganographic data hidden within images.

1 Introduction

Steganography is an ancient technique of hiding data within media, has evolved significantly. S et al. (2023)Euphrasi and Rani (2016) It was initially developed for secure communication but now it can be used both positively and negatively. On one hand, it can protect sensitive information, but it can also be used by cyber criminals to embed malware, like JavaScript code, into images for cyberattacks. Hence it is important to have strong steganography detection systems. This counter action to steganography is called Steganalysis.

Among other advanced methods, spread spectrum steganography is a complex technique that makes traditional detection methods ineffective Chaudhary et al. (2023). Spread spectrum steganography slightly modifies the frequency components of an image for data embedding. The technique is proven to be highly undetectable by spreading secret information over a large range of frequencies Chaudhary et al. (2023). The produced hidden signal is so subtle that it makes the hidden data less perceptible to human eye and more robust against conventional steganalysis methods Euphrasi and Rani (2016). hence more sophisticated steganalysis techniques are needed, especially where hidden data

can be used for malicious intentions. This includes embedding malware scripts that can be executed during cyber attacks. Criminals could use various social engineering tactics to make the the user initiate a prompt as simple as clicking on a dummy button or downloading an image.

To overcome these challenges, this study focuses on detecting images with hidden JavaScript codes used commonly in cyber attacks by using both the conventional feature extraction techniques and modern deep learning approaches. This could be termed as targeted steganalysis as the embedding technique i.e., spread spectrum is already known while developing steganalysis approach.

Research Questions

- Which is the most effective pre-trained model that can detect steganographic content when enhanced with attention mechanism?
- How can feature extraction steps affect the performance of transfer learning-based steganalysis models in detecting spread spectrum-based images?

Convolutional Neural Networks(CNNs) can extract rich features that are required in most cases for classification without any external need Reinel et al. (2019) Kheddar et al. (2024) Wei et al. (2022). The hypothesis of this study is that, although CNNs work well in extracting subtle spatial domain characteristics, it will not do the same for frequency domain changes Li et al. (2024). Accordingly when frequency domain features were explicitly extracted and combined with CNNs, they proved to be effective as demonstrated by this study. Through domain knowledge and thorough understanding of the steganography method used, effective feature extraction steps were implemented. For this research, 10,000 color images from the ImageNet Large-Scale Visual Recognition Challenge 2012(ILSVRC 2012) dataset was used Russakovsky et al. (2015). A custom dataset was created by embedding javascript code into these images. The classes were named Clean and Stego(Steganographic). Various pretrained models like ResNet50, VGG19 and Efficientnet-b0 were trained among which the hyper parameter tuned EfficientNet-b0 model has achieved an accuracy of 94.7% in detecting steganographic content.

Section 1 of this report presents the field of research, it's objective, the research questions and a brief overview of the whole research . In Section 2, existing research works related to this are explored and analysed. Section 3 describes the methodology followed throughout this work and Section 4 outlines the data pipeline of the project. The implementation of the design is thoroughly discussed in Section 5. All the experiments carried out and their results are evaluated in Section 6. The report ends with a critical analysis of this work and the future directions in Section 7.

2 Related Work

The recent advances in steganalysis focus on neural network-based approaches that help make the detection capabilities far more effective. This section discusses the different steganography methods, effective steganalysis approaches, and the transition to deep learning techniques.

2.1 Steganography and Steganalysis Techniques

To develop strong steganalysis tools, thorough understanding of steganography is needed. Steganography can be broadly classified as spatial and frequency domain techniques. Spa-

tial domain techniques directly hide data in the pixel values of an image, while frequency domain techniques hide data within the frequency-transformed coefficients Euphrasi and Rani (2016) S et al. (2023).Among them, frequency domain techniques are very hard to detect. The work by Euphrasi and Rani (2016) and S et al. (2023) provide comparative analysis of steganography techniques. Euphrasi and Rani (2016) compare these techniques using metrics like Bit Error rate, standard deviation, embedding capacity and S et al. (2023) compare these techniques with metrics like embedding capacity, security against attacks, visual imperceptibitlty and computational requirement. Both the studies conclude that the frequency domain provides better imperceptibility for hidden data compared to the spatial domain.

One of the evaluation metric used by Euphrasi and Rani (2016), showed that Bit Error Rate(BER) of spatial domain was as high as 0.0194 and in frequency domain was as high as 0.0673. The authors describe that a higher BER would indicate more harder to detect the presence of steganographic content. In the proposed work, BER of spread spectrum in spatial domain was 0.0 and in frequency domain was 0.504 when tested for the same image proving that frequency domain steganography is harder to detect.

In frequency domain spread spectrum steganography, the image is first transformed into transform domain like Discrete Cosine Transform(DCT) as described by Chaudhary et al. (2023). In their work the authors describe the application of various frequency domain methods, including the use of DCT for embedding data. It provides a clear explanation of the steps starting from creating a pseudo sequence of message bits, converting the image into DCT domain and subsequent embedding. However, a quantitative comparison of techniques could have helped in a better understanding of these techniques.

2.2 Machine Learning Applications in Steganalysis

Steganalysis can be broadly categorized into targeted and universal methods Hermassi (2021) Kheddar et al. (2024) Croix et al. (2024). Targeted steganalysis requires prior knowledge of the steganography algorithm used, but universal steganalysis is a blind and generic approach to detect steganographic content. In this research, the steganography method used, Spread Spectrum is known prior and hence can be categorized as targeted steganalysis. Further, Kheddar et al. (2024) categorize steganalysis into traditional machine learning approaches and advanced deep learning techniques. Traditional methods mostly rely on handcrafted features and statistical analysis, while deep learning methods like Convolutional Neural Networks (CNNs) use automatic feature extraction.

Spread spectrum steganography causes very subtle changes which may go unnoticed by traditional machine learning models. Implementing feature extraction steps before ML models can improve the classification abilities as demonstrated by Hermassi (2021). They perform a blind steganalysis for JPEG images from BOSSbase v1.01 image database embedded with nsF5 steganographic method which operates frequency domain mainly on DCT coefficients. They extract inter and intra block relationships of DCT coefficients and co-occurance matrices. They have achieved an Area Under the Curve (AUC) of 0.846. However, they could have addressed the trade-offs between accuracy and computational efficiency due to manual feature extraction which could be critical in real-world applications.

The authors Croix et al. (2024) demonstrate statistical methods based steganalysis on seven steganography algorithms including spread spectrum. The authors describe the use of statistical properties like image histograms, color movements and higher order statist-

ics for effective steganalysis. Similar features were extracted in this research for better detection. Their research describes the shift of Machine learning(ML) based steganalysis to Deep learning(DL) based techniques. The paper identifies challenges like the curse of dimensionality in ML-based approaches which was encountered and taken care of in the proposed work through Principal Component Analysis(PCA).

Another drawback of ML-based approach is that it depends on a small set of hand-crafted features and may not generalize on new data. Qian et al. (2016) demonstrate that CNNs automatically learn features from data, and simplifies the feature extraction process and improves accuracy. They have an interesting approach to pre-train a CNN on detecting steganographic algorithms with high payloads and then transfer the learned feature representations to tasks involving lower payloads. They use a high pass filter as a preprocessing step to highlight the steganographic content similar to DCT feature extraction in the proposed work.

Another work that highlights Deep learning(DL) techniques over ML techniques is presented by Reinel et al. (2019). The researchers have used BOSSBase, BOWS2 and ImageNet datasets to show the superiority of DL based models over ML models. They have used preprocessing steps which are tailored to highlight the steganographic noise signals from the natural patterns of the image to detect frequency domain steganography. This again shows that even with DL techniques, highlighting steganographic content in some way is necessary for effective detection.

CNNs are mostly capable of statistical features, but may not effectively identify frequency differences in classes. Hence, a hybrid model of both manual feature extraction and features extracted from CNNs was implemented in the proposed research. A similar perspective is implemented by Li et al. (2024) using a transformer-inspired blocks called ResFormer - Residual transformer method. It combines the traditional feature extraction with CNN-based deep learning for effective steganalysis. The method achieves detection accuracy of 92.13% for algorithms like WOW, S-UNIWARD. Although it claims to be lightweight, the two step process might in fact be computationally expensive which needs further optimization.

The hybrid methodology implemented in this research involves explicitly extracting features which makes the results mostly dependent on a particular dataset and may result in overfitting. Boroumand et al. (2019) introduce a deep residual network SRNet specifically designed to minimize manual design elements of the data pipeline. Although their work aims to minimize design elements, they use heuristically selected channels for better performance. A more data-driven approach during channel selection could have been explored.

While implementing hybrid models including feature extraction and CNNs, computational efficiency becomes a major concern. Lin and Yang (2021) propose a method that focuses on improving the detection performance of steganalysis for color images with a lightweight network architecture. The network is designed to learn multi-frequency components separately, which enhances the detection accuracy without significantly increasing the network's complexity. While the network is lightweight, the trade-off between the model complexity and the overall network performance should be carefully considered.

The researchers Kheddar et al. (2024) have implemented a thorough comparative analysis of various Deep Learning based steganalysis models. The metrics of evaluation used to evaluate models used by them was a valuable guide for the evaluation of proposed research. Their thorough research directs towards pretrained models and their advantage of transfer learning over CNNs and RNNs built from scratch. They use their

prior knowledge to understand features in images from large datasets. This makes it suitable detect subtle patterns in steganalysis tasks without requiring much retraining. EfficientNet which is a pretrained network, is particularly useful for steganalysis due to its balances in accuracy and computational efficiency Chubachi (2020). The authors Chubachi (2020) perform targeted steganalysis for JPEG images of ALASKA2 dataset in both spatial and frequency domain and have achieved highest accuracies with pretrained EfficientNet architectures. They have implemented EfficientNet-b2 and b5 models that take DCT coefficients as input. EfficientNet for the frequency domain achieves a weighted AUC of 0.9405. However computational efficiencies while using such pretrained models could have been discussed more.

Other than a separate feature extraction step, the CNNs extract features from images too. In the architecture of CNNs for steganalysis, use of attention mechanisms can be highlight the features extracted from CNNs Kheddar et al. (2024) Fu et al. (2022). The hybrid approach of feature extraction and attention mechanism in CNNs was implemented in the proposed research to achieve effective steganalysis. The work by Fu et al. (2022) uses a feature extraction step, channel attention mechanism and convolutional pooling to improve the detection accuracy of steganalysis in spatial domain which inspired it's use in proposed work. They have implemented channel attention to capture color channel wise features which was implemented by histogram feature extracted for color channels in the proposed work. The scalability of the model could have been addressed more clearly to facilitate real time implementation.

Most existing steganalysis methods are optimized for grayscale images as they as computationally less expensive. The authors Wei et al. (2022) propose a universal deep network designed for steganalysis of color images. The authors introduce a preprocessing module that separates color channels and applies 62 high-pass filters to enhance the steganographic noise signal. This helps preserving steganographic features across different color channels using a carefully designed CNN architecture. The importance of analysis of each color channel has been an inspiration in the proposed research where histogram and other statistical features of RGB channels are separately computed and used as features.

During the implementation, slight overfitting could be observed with quite a bit of difference in training and validation accuracies. Hyperparameter tuning was then performed and the learning rate was decreased to reduce the overfitting. This issue of manual tuning is solved to adaptive learning rate approach by Mustafa et al. (2019) They have implemented Dynamic Learning Rate-Based CNN for flexible training of CNNs for WOW and S-UNIWARD steganographic algorithms on BOSSbase 1.01 dataset. It shows significant improvement from other CNN based architectures but at the cost of computational efficiency. It requires powerful GPUs which may be less suitable for scenarios with limited computational resources. Hence it could not be incorporated in this research.

2.3 Preprocessing for Frequency Domain Steganalysis

In this research, preprocessing is considered a critical step as it enhances the subtle features indicating steganographic content that may otherwise be unnoticed in the raw images especially in frequency domain. Although Spread specrum affects frequency domain components, the steganographic data is added to the original image in such a way that it is distributed across the image's pixels. This causes slight modifications to the spatial domain too V.K et al. (2021) Xu et al. (2015). The spatial and frequency domain analysis needed to detect for steganalysis is detailed by V.K et al. (2021). In frequency

domain, DCT coefficients are analysed and in spatial domain, they examine the subtle changes in pixel values across the images. This inspired the use of frequency and spatial domain analyses to detect DCT based steganography. However, discussing the actual datasets and evaluation metrics used could have been more helpful.

The authors Qian and Manoharan (2015) implement steganalysis on JPEG images in spatial domain. Histogram features capture the distribution of pixel intensities across different color channels essentially capturing spatial domain changes. This is a technique used by Qian and Manoharan (2015) in one of their steganalysis models which works quite well for Least Significant Bit(LSB) steganography. Although they evaluate color and grayscale images with different JPEG compression rates, they could have highlighted the effect of such compressions with quantitative results.

JPEG steganalysis is one of the frequency domain techniques dealing with DCT coefficient changes Maryam Seyed Khalilollahi and Mansouri (2022) Cheng et al. (2024). The researchers Cheng et al. (2024) implement a method that integrates frequency domain analysis with deep learning to detect JPEG steganography. They describe the need to highlight DCT coefficients before feeding the images to neural network architectures. They have achieved over 95% accuracy in single compressed images. However, generalizability to other compression formats or non-JPEG images has not been addressed.

The DCT coefficient analysis is needed only for those parts of the image which may contain steganographic content. Otherwise it may cause high dimensionality issues. While in most cases, high frequency components are affected, in the proposed work, the steganography mostly affects low to mid frequency components. Such mid-frequency component analysis for steganalysis is implemented by Maryam Seyed Khalilollahi and Mansouri (2022). Their work guides towards using extracting specific DCT coefficients from the JPEG images and they have used 20 components to achieve balance between accuracy and complexity. In the proposed work, 10 coefficients are utilized to achieve this balance. With 20 DCT components, Maryam Seyed Khalilollahi and Mansouri (2022) have achieved an accuracy of 97.75% which is quite high for the low complexity of their network. It still seems like a shallow network which may not generalize well for larger or complex datasets.

Apart from DCT features, spatial features like co-occurance matrix features were extracted for texture analysis. The authors Xu et al. (2015) state that texture analysis is needed to capture the correlation among neighboring pixels for steganalysis. Their assumption is that pixels exhibit strong local correlations, which can be disrupted by steganographic embedding. This theory seemed logical and hence texture related features were extracted.

3 Methodology

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a a structured approach used during development of data mining projects Sakly et al. (2022). It has six steps which are elaborated as follows.

3.1 Business Understanding

This step defines the motivation and goal of this research. When used with wrong intent, steganography can be used for dangerous cyberattacks S et al. (2023). Hence, primary objective of this research is to develop a strong steganalysis technique to detect

steganographic images using a hybrid approach. This includes combining traditional feature extraction methods with advanced deep learning models augmented with attention mechanisms.

3.2 Data Understanding

There were no ready image datasets available with JavaScript embedding through spread spectrum steganography. Hence a custom dataset had to be created. The images were taken from the ImageNet Large-Scale Visual Recognition Challenge 2012(ILSVRC2012) Russakovsky et al. (2015). About 10,000 JPEG images of the test subset of this dataset were used in this work.The dataset was divided into 5000 cover and 5000 clean images.

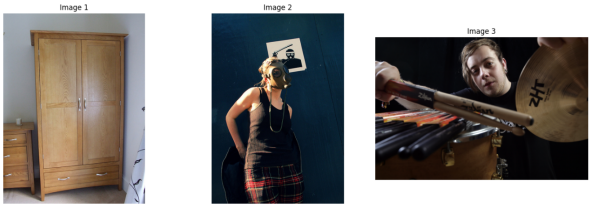

Figure 1: Cover Images

Figure 1 shows the cover images used to create the Steganographic(Stego) images through Javascript code embedding. As observed, the images were of varying dimensions.

3.3 Data Preparation

This step includes the steps involved in creating the custom dataset. The cover images were first resized to 224x224 pixels for compatibility with the input size required by most pretrained models. Resizing the images was done before embedding the data to avoid any distortion of the steganographic content that might occur if resizing were done later Reinel et al. (2019).

For the purpose of this research, a simple JavaScript code was created to act as a placeholder for the malware inspired by Petrak (2019). This code included functions that mimic typical malicious activities, such as manipulating the DOM or initiating HTTP requests.

The JavaScript code was embedded into the cover images using Spread Spectrum Steganography in both frequency domain and spatial domains. First the code was converted to it's binary equivalent. This sequence was then modulated by XORing with a key to create a pseudorandom sequence Chaudhary et al. (2023). In frequency domain, Discrete Cosine Transform (DCT) was applied to the image, and the code was embedded within the DCT coefficients of RGB channels. In Spatial domain steganography the code was directly embedded into the pixel values of the cover image but spread across wide spectrum of pixels and not concentrated at a specific location Euphrasi and Rani (2016). Hence this can be perceived as spread spectrum in spatial domain. Metrics for comparison were as follows.

- **Visual Imperceptibility:** It can be observed in Figure 2 that both the frequency domain and spatial domain stego images show no noticeable visual distortions.

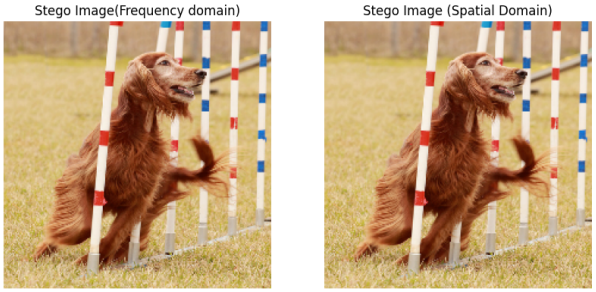

Figure 2: Visual Imperceptibitlity

- **Bit Error Rate(BER):** BER for frequency domain was 0.504 and spatial domain was 0.0.
- **Histogram of pixel intensities:** Figure 3 shows that the histogram of pixel intensities for frequency domain shows more noticeable differences between the original and the stego image than the spatial domain

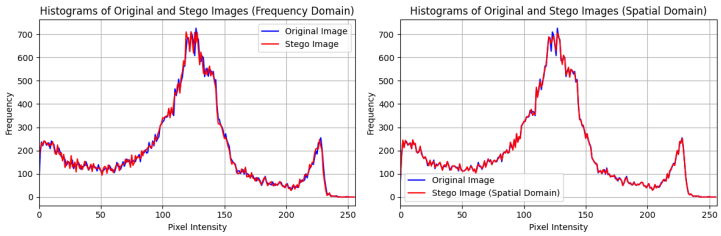

Figure 3: Histogram comparison

It can be concluded that Spread Spectrum Steganography in the Frequency Domain is the more challenging technique for decoding. Hence it was chosen over spatial domain. Once the steganography techniques was chosen, 5000 cover images were used to create 5000 stego images containing the embedded JavaScript code. Clean images were also resized. These images were then divided into 8000 train, 1000 test, and 1000 validation sets of CLean and Stego classes.

Next, during Exploratory Data Analysis(EDA) basic image characteristics like channel wise mean pixel values, brightness and contrast distributions in clean and Stego images were analysed. The mean pixel values showed very small differences which were later extracted as features during histogram analysis.

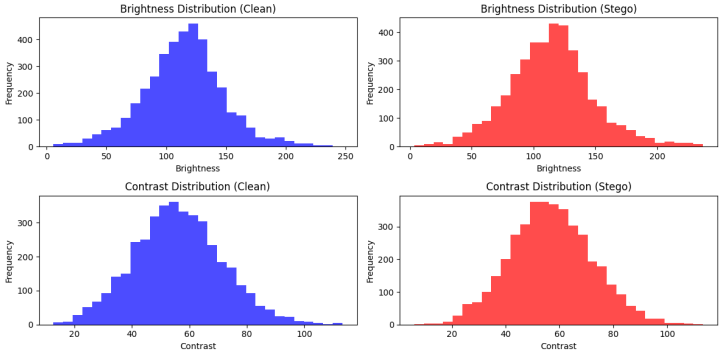

Figure 4: Brightness and Contrast Analysis

Figure 4 shows the Brightness and Contrast distributions. The subtle variations observed in Brightness and Contrast are further captured during the histogram analysis and co-occurance matrix analysis respectively. Further feature extraction steps will be discussed in implementation section.

3.4 Modeling

Data augmentation is usually used in image-based machine learning to enhance training data diversity and prevent overfitting Li et al. (2024). Techniques like zooming, flipping, and rotating images are common, but for steganalysis, these methods could distort the hidden data by either making it too obvious or completely hiding it Boroumand et al. (2019). To ensure the steganographic content remained intact and realistic, these augmentations were intentionally excluded in this study. To test hypothesis that pure CNN based techniques may fail in effective steganalysis in frequency domain, simple CNN models without any feature extraction steps were created. Then the extracted features were combined with image data through data generators. This combined data was used as the input for pretrained models. The results are discussed in further sections.

3.5 Evaluation

The models in this project are evaluated by key metrics like training & validation accuracy and loss, test accuracy, precision, recall, F1-score, Receiver Operating Characteristic (ROC) Curve and Precision-Recall Curve Kheddar et al. (2024)Lin and Yang (2021)Wei et al. (2022).

3.6 Deployment

Deployment step involves implementing real-time detection of steganographic content in images with the developed approach. This is not involved in the scope of this research. The models are however saved in .h5 formats and can be exported to potentially integrate them into existing cybersecurity frameworks.

4 Design Specification

Figure 5 shows the model architecture of this research. There are two parallel paths for data processing. The first path contains image data as input to a pre-trained model. The model, through transfer learning, extracts deep features of images and captures high-level patterns and characteristics. Next is a global average pooling layer that reduces the features further. Next is a self-attention layer is used to stress the most relevant featuresRef5 Fu et al. (2022). In the other side, domain-specific features from the image data are extracted. These features capture the frequency domain changes caused by steganography which the CNN might not capture Li et al. (2024). The features are reduced to keep only the most informative and relevant features. These features are then combined with features extracted by CNN. The combined feature set captures hence both high-level patterns and low-level domain-specific information. This combined feature set is then passed through dense layers, and finally the classification layer.

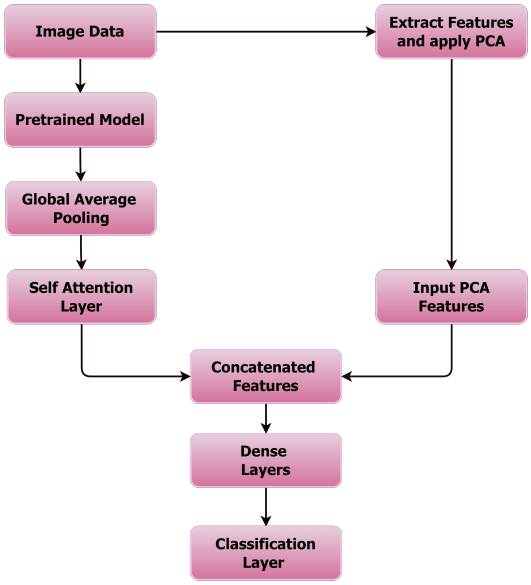

Figure 5: Model Architecture

5 Implementation

This section outlines the final implementation of the proposed steganalysis framework.

5.1 Tools and Technologies

The implementation is done using Python language. Python libraries used included TensorFlow for building and training deep learning models, Pandas for data manipulation, and NumPy for numerical computations Mustafa et al. (2019). Matplotlib and Seaborn were used for data visualization. Scikit-learn provided tools for data preprocessing. Google Colab was used as the development environment. Specifically, T4 GPU and T2 TPU with High RAM were used to process the large datasets, the training of deep learning models.

5.2 Feature Extraction and Dimensionality Reduction

The goal of feature extraction was to maximize the detection by focusing on features that are most sensitive to the alterations caused due to steganography V.K et al. (2021) Xu et al. (2015). The extracted features were saved in csv files.

5.2.1 Histogram Analysis

The histogram analysis captures the distribution of pixel intensities in an image Croix et al. (2024) Qian and Manoharan (2015). Features extracted in this process are for the RGB channels separately: peak frequencies, mean pixel value and standard deviation, mid range frequency slope, and histogram entropy. The peak frequencies are calculated for the darkest and brightest occurrences of a pixel. The mean and standard deviations of pixel values capture the pattern of distribution of pixel intensities Fu et al. (2022). The slope of the mid-range frequency shows the effect that embedding would have imposed on

low to mid range pixels. Histogram entropy indicates the complexity of pixel distribution, which may become greater due to data embedding Wei et al. (2022).

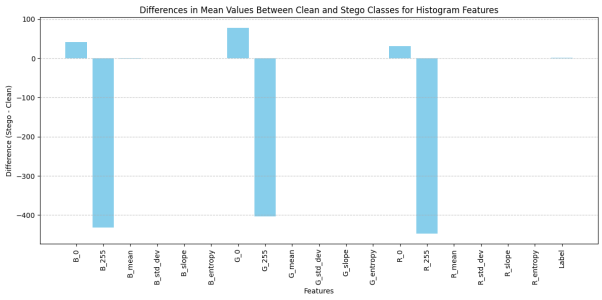

Figure 6: Mean Histogram Analysis

Figure 6 shows the differences in mean values between clean and stego classes across various histogram features. Significant differences are observed in the frequency of extreme pixel values (0 and 255) across all color channels (B, G, R), which suggests that steganographic embedding affects these histogram characteristics.

5.2.2 Co-occurrence Matrix Analysis

The co-occurrence matrix captures the spatial relationships between pixels by computing how often pairs of pixel values occur together in a specified spatial relationship within the image Xu et al. (2015). High contrast indicates regions of high activity or noise, which could be manipulated in steganography to embed data. Dissimilarity is calculated to measure variations and edges of the image. Homogeneity was extracted to check any irregular textures and energy was captured as hidden data may exhibit variations in energy levels due to alterations in pixel patterns Croix et al. (2024). Deviations in correlations may also indicate hidden data.

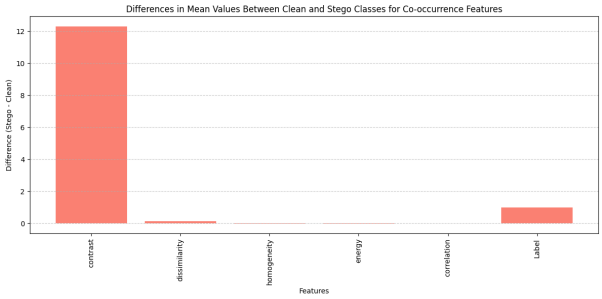

Figure 7: Co-occurance feature analysis

Figure 7 shows that when mean values are computed, most significant difference between clean and stego images is observed in the contrast feature.

5.2.3 Discrete Cosine Transform (DCT) Coefficients

Spread spectrum is a frequency domain technique. Hence DCT coefficients analysis is important Qian et al. (2016) Maryam Seyed Khalilollahi and Mansouri (2022). Low-frequency coefficients were analyzed because they influence the overall image appearance,

while high-frequency coefficients, which captured finer details, were also analysed to detect hidden data Hermassi (2021)V.K et al. (2021) Cheng et al. (2024). The images were divided into 8x8 blocks for DCT analysis for finer frequency analysis within each block. A total of 10 DCT coefficients were extracted from each block, with both low-frequency and high-frequency coefficients being analyzed to detect hidden data. This was chosen after experimenting to observe the computational requirements needed for more co efficients. 10 coefficients with 8x8 block size showed optimum performance.

5.2.4 Higher-Order Statistics

Higher-order statistics includes skewness and Kurtosis. Skewness measures the asymmetry of the pixel value distribution while Kurtosis measures the outliers in distribution Chubachi (2020). Steganographic methods may vary these features of the image by introducing more outliers and asymmetry compared to the clean images.

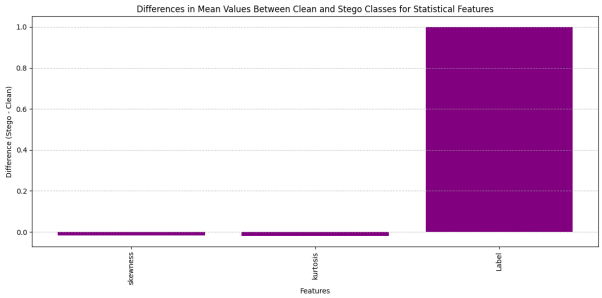

Figure 8: Skewness and Kurtosis analysis

Figure 8 shows that the mean skewness and kurtosis values have quite a bit of difference between clean and stego images.

5.2.5 Dimensionality Reduction with PCA

The extracted features had very high dimensions which can cause computational inefficiencies and overfitting during modeling Croix et al. (2024). Hence the extracted features from the training, validation, and test datasets were combined and first normalized using StandardScaler. Principal Component Analysis(PCA) was initially applied and cumulative variance was used to determine the number of principal components needed to capture at least 95% of the total variance in the data Maryam Seyed Khalilollahi and Mansouri (2022). This analysis revealed that 3,927 components were necessary to meet this threshold. These PCA-transformed features were then saved into CSV files for model training and evaluation. This approach ensured that the most significant features, causing the majority of the variance of data, were retained while reducing the dimensionality of the dataset.

5.3 Modeling

5.3.1 Baseline Models

In the initial modeling stage, baseline models were developed without data augmentation or extensive preprocessing. This was to establish a reference point to evaluate more complex approaches. A sequential model was created with six convolutional blocks, each

followed by max-pooling. Then two dense layers with 256 and 128 units were added. The final output layer used a sigmoid activation function for binary classification. The model was trained using the Adam optimizer with a learning rate of 0.001 for 30 epochs. It achieved moderate performance with a training accuracy of 49.67

Subsequently, a more advanced baseline was set using a pre-trained ResNet50 model, leveraging its 50-layer deep architecture to handle complex features. This pre-trained model was used to set a benchmark for measuring the effectiveness of additional feature extraction techniques, such as PCA and attention mechanisms. The model was loaded with ImageNet weights, excluding the top layers, and customized with a Global Average Pooling layer and two dense layers, similar to the sequential model. The ResNet50 layers were frozen, focusing training on the new layers. Despite the improved architecture, the ResNet50 baseline achieved a training accuracy of 69.45%, showing only a modest improvement over the initial baseline, indicating the need for further enhancement through feature extraction and attention mechanisms.

5.3.2 Pretrained models

A custom data generator was created to combine image data with PCA-extracted features. This generator preprocessed images by resizing them to 224x224 pixels and then merged them with the PCA features to help the models to learn from both spatial and statistical information.

The ResNet50 model used as baseline was then adapted to incorporate PCA features and a self-attention mechanism. The model was initialized with pre-trained ImageNet weights. The last three layers of pretrained model was unfrozen for fine tuning to this dataset. The combined data generator was used to feed the model with both image data and PCA features. After the initial feature extraction by ResNet50, a self-attention layer was introduced to help the model focus on important regions of the image. The ResNet features from attention layer and PCA features were further trained on dense layers. The final layer used a sigmoid activation function for binary classification. The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and for 30 epochs. It achieved a training accuracy of 99%. The testing accuracy was 92.90% which indicated overfitting issues. It still showed significant improvements from baseline ResNet model.

VGG19, with its 19-layer architecture, was also adapted to include PCA features and a self-attention mechanism, resulting in a slightly better performance than ResNet50. The VGG19 model achieved train accuracy of 98.9% and a test accuracy of 93.4%. This difference showed some reduced overfitting than ResnNet model. Despite these improvements, more advanced EfficientNetB0 was chosen for further refinement.

EfficientNet-b0 which is known for its efficiency and performance, was utilized for its balanced scaling of network dimensions. Consistent training parameters were applied to maintain experimental uniformity. It achieved a training accuracy of 98.9% and testing accuracy of 93.3%. Although its performance was almost same as VGG19, Efficientnet-b0 was chosen for further tuning due to the scalability of it's architecture.

In the tuned EfficientNetB0 model, hyperparameters were adjusted to enhance performance further. The learning rate was reduced to 0.0005 to allow more controlled and gradual learning, and the dropout rate was increased to 0.6 to better prevent overfitting. Additionally, all but the last layer of the pre-trained EfficientNet network were frozen, allowing the model to retain valuable features from the ImageNet pre-training while ad-

apting to the specific steganalysis task Kheddar et al. (2024). These adjustments resulted in a significant improvement, with train accuracy of 99.40% and a test accuracy of 94.70%. There is slightly reduced overfitting in this model.

6 Evaluation

Accuracy calculates the portion of correctly classified images. Precision calculates how accurate the model is in classifying images. Recall (sensitivity) measures the ability to identify all true images in the dataset. The F1-Score is a balanced metric of precision and recall. The ROC curve shows how well the model distinguishes between classes. The Precision-Recall Curve plots precision against recall for different threshold levels Kheddar et al. (2024) Reinel et al. (2019) Li et al. (2024).

6.1 Baseline Sequential Model

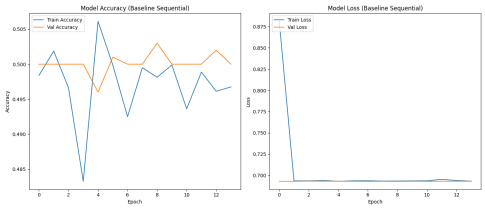

Figure 9: Model accuracy and loss during training

Figure 9 shows considerable fluctuations around 50% accuracy, indicating that the model is not learning effectively due to issues like insufficient model complexity.

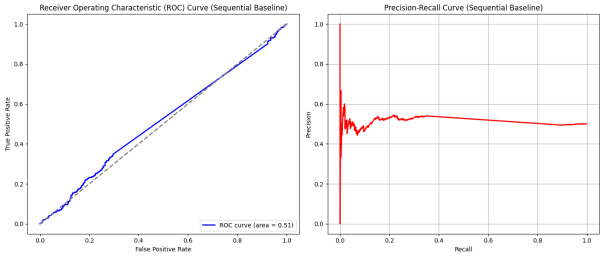

Figure 10: ROC and Precision-Recall curve

The ROC curve in Figure 10 shows that the classification is equivalent to random guessing, with an AUC of 0.51.

6.2 Baseline Resnet50 Model

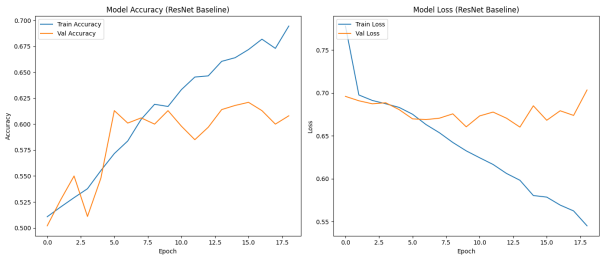

Figure 11: Model accuracy and loss during training

Figure 11 shows the loss curves that indicate a reduction in both training and validation loss, suggesting that the model is learning but may not be generalizing well to the validation set.

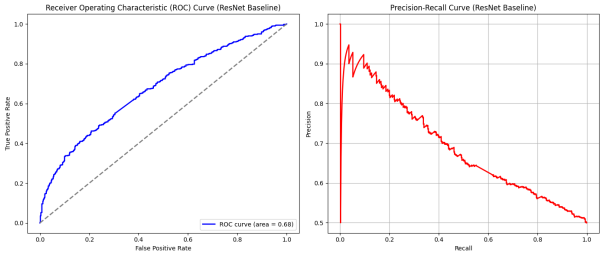

Figure 12: ROC and precision-Recall curve

The ROC curve in Figure 12 shows a moderately better ability to distinguish between classes, with an AUC of 0.68, indicating a performance better than random guessing. However, the precision-recall curve reveals a decline in precision as recall increases, meaning areas for further model optimization.

6.3 Resnet50 Model

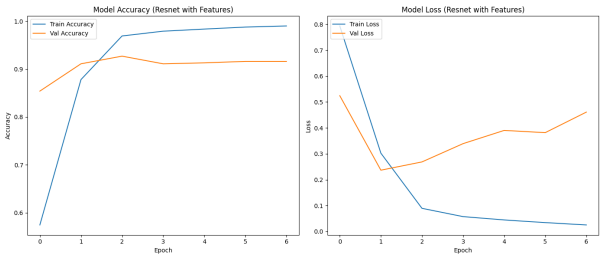

Figure 13: Model accuracy and loss during training

The ResNet model with features and data augmentation demonstrates significant improvements in both training and validation accuracy. Figure 13 shows the convergence of the model during training, with some reduced but presence of overfitting.

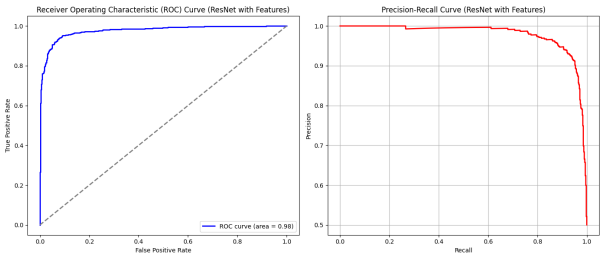

Figure 14: ROC and Precision-Recall curve

The ROC curve in Figure 14 shows a improved ability to distinguish between classes with an AUC of 0.98.

6.4 VGG19 Model

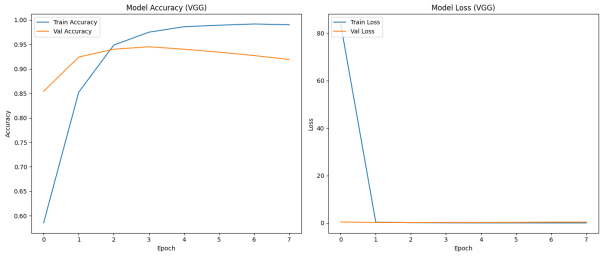

Figure 15: Model accuracy and loss during training

The accuracy and loss curves depicted in figure 15 demonstrate a strong learning process with training accuracy reaching near 100%. However, there is a noticeable gap between the training and validation accuracy, suggesting potential overfitting.

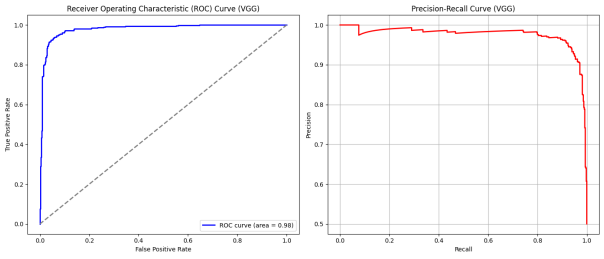

Figure 16: ROC and Precision-Recall curve

The ROC curve in figure 16 shows a high area under the curve (AUC) of 0.98, indicating excellent classification performance. Similarly, the precision-recall curve suggests high precision across most recall values, reinforcing the model's strong performance in distinguishing between classes.

6.5 EfficientNet-b0 Model

In the EfficientNet model, the accuracy and loss curves as shown in figure 17 indicate a well-fitting model with high training and validation accuracy, approaching nearly 97%.

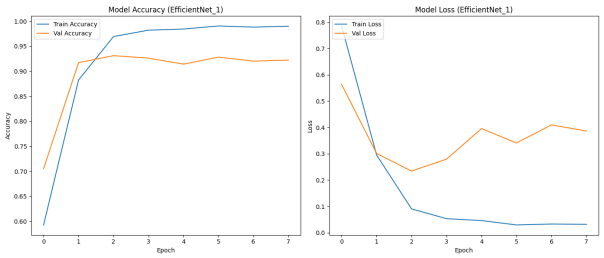

Figure 17: Model accuracy and loss during training

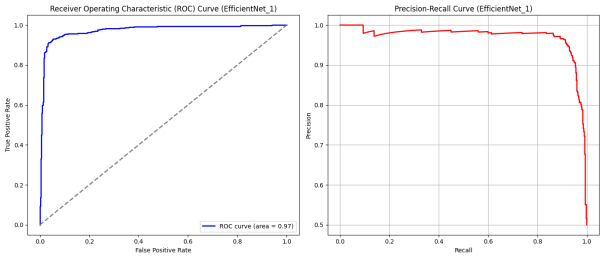

Figure 18: ROC and Precision-Recall curve

The ROC curve in figure 18 demonstrates a strong ability to distinguish between classes, with an AUC of 0.97.

6.6 EfficientNet-b0 - Hyperparameter tuned Model

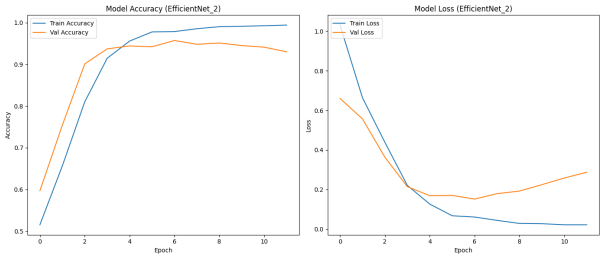

Figure 19: Model accuracy and loss during training

From figure 19 the training and validation accuracy curves indicate high performance, with training accuracy approaching 100% and validation accuracy slightly lower but still high enough. The model's loss curves exhibit a consistent downward trend, suggesting effective learning with minimal overfitting.

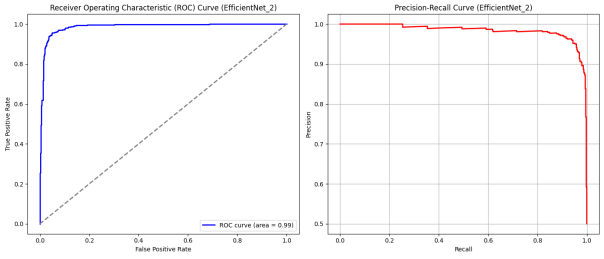

Figure 20: ROC and Precision-Recall curve

The ROC curve as shown in figure 20 shows an AUC of 0.99, highlighting the model's strong ability to differentiate between classes. The precision-recall curve also shows the robust performance of the model. This model so far showed the highest accuracy and hence is chosen for steganalysis task of spread spectrum steganography.

6.7 Discussion

Table 1: Summary of Model Performance

Model	Accuracy	Precision	Recall	F1-Score
Baseline	0.49	0.49	0.50	0.35
ResNet Baseline	0.61	0.61	0.61	0.61
ResNet50	0.93	0.93	0.93	0.93
VGG19	0.934	0.94	0.93	0.93
EfficientNetB0	0.933	0.93	0.93	0.93
EfficientNetB0 (Tuned)	0.947	0.95	0.95	0.95

Table 1 shows the weighted average metrics of model performances. It can be seen that there is a slight improvement from each model to the next in almost all metrics. The baseline models, such as simple CNN and ResNet50 without feature extraction or attention mechanisms, showed limited effectiveness, with accuracy scores of 0.49 and 0.61. This means that the basic architectures are not sufficient to extract the subtle changes introduced by spread spectrum steganography. However, pretrained models like ResNet50, VGG19, and EfficientNetB0, combined with feature extraction and self-attention mechanisms, showed significant improvement, with the tuned EfficientNetB0 achieving an accuracy of 0.95. The previous work by Cheng et al. (2024) achieves a maximum accuracy of 0.96 for different compression rates and another work by Maryam Seyed Khalilollahi and Mansouri (2022) achieves the highest accuracy of 0.97 for JPEG staganalysis. Given the complexity of steganography method, the accuracy achieved by the proposed method seems acceptable. The final tuned EfficientNet model achieves an AUC of 0.99 which is a significant improvement over other frequency domain steganalysis methods proposed by Hermassi (2021) with AUC of 0.846 and EfficientNet based model proposed by Chubachi (2020) with AUC of 0.95. Despite the improvements, there were subtle signs of overfitting, suggesting a need for further tuning. This means more dropout layers or regularization techniques are needed.

7 Conclusion and Future Work

This study explored how combining traditional feature extraction methods with advanced deep learning models can effectively detect steganographic content, especially when using spread spectrum techniques. By employing transfer learning and enhancing it with a self-attention mechanism, the research achieved a notable detection accuracy of 94.7%. The integration of PCA-extracted features with the deep learning model proved to be stable and effective, capturing both spatial and frequency domain features essential for detecting subtle steganographic changes.

However, some limitations were noted. The models showed signs of overfitting, particularly in the early experiments, suggesting that more aggressive dropout or data aug-

mentation might be needed. While EfficientNetB0 was chosen for its balance between performance and computational efficiency, exploring other architectures or more advanced hyperparameter tuning could yield even better results.

Future research could expand this targeted steganalysis beyond spread spectrum methods. Testing the developed algorithm on different steganographic datasets and experimenting with other pretrained models could further improve performance. The scalability of the algorithm is limited, so testing on larger datasets would help in drawing more reliable conclusions.

References

Boroumand, M., Chen, M. and Fridrich, J. (2019). Deep residual network for steganalysis of digital images, *IEEE Transactions on Information Forensics and Security* **14**(5): 1181–1193.

Chaudhary, M., Sharma, R., Tiwari, S., Tomar, A. S., Singh, S., Kumar, K. and Kaur, M. (2023). Concealing information in images: A review of steganography method, *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Vol. 6, pp. 49–55.

Cheng, X., Wang, J., Wang, H., Luo, X. and Ma, B. (2024). Quantization step estimation of color images based on res2net-c with frequency clustering prior knowledge, *IEEE Transactions on Circuits and Systems for Video Technology* **34**(1): 632–646.

Chubachi, K. (2020). An ensemble model using cnns on different domains for alaska2 image steganalysis, *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6.

Croix, N. J. D. L., Ahmad, T. and Han, F. (2024). Comprehensive survey on image steganalysis using deep learning, *Array* **22**: 100353.

Euphrasi, K. R. and Rani, M. M. S. (2016). A comparative study on video steganography in spatial and iwt domain, *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pp. 104–109.

Fu, T., Chen, L., Fu, Z., Yu, K. and Wang, Y. (2022). Ccnet: Cnn model with channel attention and convolutional pooling mechanism for spatial image steganalysis, *Journal of Visual Communication and Image Representation* **88**: 103633.

Hermassi, H. (2021). Blind image steganalysis using features extraction and machine learning, *2021 18th International Multi-Conference on Systems, Signals Devices (SSD)*, pp. 673–680.

Kheddar, H., Hemis, M., Himeur, Y., Megías, D. and Amira, A. (2024). Deep learning for steganalysis of diverse data types: A review of methods, taxonomy, challenges and future directions, *Neurocomputing* **581**: 127528.

Li, H., Zhang, Y., Wang, J., Zhang, W. and Luo, X. (2024). Lightweight steganography detection method based on multiple residual structures and transformer, *Chinese Journal of Electronics* **33**(4): 965–978.

Lin, J. and Yang, Y. (2021). Multi-frequency residual convolutional neural network for steganalysis of color images, *IEEE Access* **9**: 141938–141950.

Maryam Seyed Khalilollahi, S. and Mansouri, A. (2022). Jpeg steganalysis using the relations between dct coefficients, *2022 International Conference on Machine Vision and Image Processing (MVIP)*, pp. 1–4.

Mustafa, E. M., Elshafey, M. A. and Fouad, M. M. (2019). Accuracy enhancement of a blind image steganalysis approach using dynamic learning rate-based cnn on gpus, *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 1, pp. 28–33.

Petrak, H. (2019). Javascript malware collection, `Computer programme`. Available at: `https://github.com/HynekPetrak/javascript-malware-collection/tree/master/2019/20190808`.

Qian, T. and Manoharan, S. (2015). A comparative review of steganalysis techniques, *2015 2nd International Conference on Information Science and Security (ICISS)*, pp. 1–4.

Qian, Y., Dong, J., Wang, W. and Tan, T. (2016). Learning and transferring representations for image steganalysis using convolutional neural network, *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2752–2756.

Reinel, T.-S., Raúl, R.-P. and Gustavo, I. (2019). Deep learning applied to steganalysis of digital images: A systematic review, *IEEE Access* **7**: 68970–68990.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision (IJCV)* **115**(3): 211–252.

S, S. K., Hegde, S., P, S. and P, V. R. (2023). Exploring the effectiveness of steganography techniques: A comparative analysis, *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pp. 181–186.

Sakly, H., Bjaoui, M., Said, M., Kraiem, N. and Bouhlel, M. S. (2022). Medical decision-making based on combined crisp-dm approach and cnn classification for cardiac mri, *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 25–30.

V.K, S., Nedumaran, A., Saraswathi, P. and Karthika, P. (2021). Performance analysis of stego-image security in machine learning, *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 676–678.

Wei, K., Luo, W., Tan, S. and Huang, J. (2022). Universal deep network for steganalysis of color image based on channel representation, *IEEE Transactions on Information Forensics and Security* **17**: 3022–3036.

Xu, X., Dong, J., Wang, W. and Tan, T. (2015). Local correlation pattern for image steganalysis, *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pp. 468–472.