Haneen Zamzami

Abstract

Steganalysis is described as "the art and science of detecting secret messages hidden using steganography". Hidden files can be images or text. The goal of this project was to use classification models to predict steganographic images by extracting number of image features such as correlation, contrast, homogeneity, and energy. Extracting different image features is on of the best techniques in blind steganalysis. It makes the system more able to deal with many types of colored images regardless of their sizes and the steganography methods were used to hide the data. Steganalysis is commonly used in computer forensics, electronic security, and internet monitoring of illegal activity. Steganalysis also aids in the improvement of steganographic method protection by identifying and finding weaknesses. Steganography is used for many legall purposes such as: ownership of digital images, authentication, copyright and data integrity. Recently it has become an important issue in the computer security world. Steganalysis is an interesting topic that has became a challenge for steganalysers and forensic examiners. However, steganography was also used for illegal issues. For example, internet communication with regard to its use by terrorist organisations when communicating information to one another is still under debate, with one party thinking that terrorists use steganography to speak to one another and the other party thinking this is not the case. Wherever the truth may lie in this regard, there is no dispute about the fact that at the present time, steganography is utilised for the allocation of illicit content, including the information gleaned from stolen credit cards, allocation of user's names and passwords, etc.

Is this image clean or stego?

What is the percentage of stego images versus clean images?

Is there a correlation between predicted category with GLCM features?

Design

In this project, I will predict the catogray of images by extract GLCM features., which are represented by a 8 of features are: image name, Image size, Image format (JPG- BMP- TIFF), Contrast of image Homogeneity of image, Energy of image, Correlation of image and Catogry of Images (clean\stego), The analysis will be based on 8683 GLCM features information. Classifying statuses accurately via machine learning models would enable the for steganalysers and forensic examiners from stop illegal use of hidden data.

Data

Build a validation image dataset. Detection of some stego-images or use of some image steganography techniques to create stego-images for testing with the modified method. I download the images from different resource Kaggle , pixabay and gettyimages. then I create my own database by extract the Gray Level Co-Occurrence Matrix (GLCM) properties of correlation, contrast, homogeneity, and energy . The dataset contains 8683 records with 8 features for each, 4623 are clean images and 4060 are stego images.

Algorithms

Feature Engineering

1. study of the GLCM features
2. Create the category variable
3. Converting categorical features to binary dummy variables
4. Dividing the features into numerical and categorical features
5. Considering category as the goal to be reached after prediction
6. Use of numerical and categorical features for forecasting
7. Split the data into train and test

Models

Support vector machine , k-nearest neighbors, Linear Discriminant Analysis and Decision Tree classifiers were used before settling on KNN as the model with strongest cross-validation performance. KNN feature importance ranking was used directly to guide the choice and order of variables to be included as the model underwent refinement.

# 10-fold cross-validation with K=5 for KNN (the n_neighbors parameter)

Model Evaluation and Selection

The entire training dataset of 8683 records was split into train and test. The official metric for Driven Data was classification rate (accuracy).

Classifier	Tranning set	Testing set
k-nearest neighbors (KNN)	**97.08%**	**95.62%**
Linear Discriminant Analysi s (LDA)	**95.94%**	**95.62%**
Decision Tree	**100%**	**93.98%**
Support vector machine (SVM)	**95.94%**	**95.62%**

Tools

- Jupyter notebook for coding
- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting
- Confusion Matrix for visualizations