Miriam Riefel 11332549
Francis Mocsin 11426896
Delanie van der Meulen 10729038
22 December 2017

Banknote authenticity - Machine Learning Classification

Introduction

In modern day society, for cash transactions to happen smoothly it is important to be able to protect ourselves from faulty banknotes. It is almost impossible for the human eye to detect very minor differences between a forged and real banknotes. The way the banknotes that are in circulation right now are produced is done through a special printing technique called the Intaglio process. This process presses with metric tons of force onto the paper, which results in a feelable relief. This also comes with a high contrast in the paper. Due to the uniqueness of the Intaglio printing process, banknotes that are made this way can be used for authentication. With proper analytical tools, it is possible to extract relevant statistical data which can be used to train machine learning algorithms to classify a banknote as real or forged. The aim of this paper is to use a banknote dataset that has been prepared by Gillich & Lohweg (2010), and use that dataset to train three classifiers: Support Vector Machine (SVM), Logistic Regression (LR), and Neural Networks (NN). This experiment is a recreation of earlier research on banknote authenticity done by Gillich & Lohweg (2010), using the same dataset but with three different classifier, all the while tuning the parameters, to see how well those classifiers perform against the original classifier in the paper.

Commented [1]: misschien hier minor gebruiken?

Commented [2]: gotcha

Problem & Dataset

The dataset that we are working with consists of 1372 examples that contain values of the visual attributes of the banknotes, as well as the label (1 = real, 0 = forged). The attributes are separated into four features: Entropy, Skewness, Variance, and Kurtosis of the wavelet transformation. These are features that were extracted after the digitisation and frequency transformation of the banknotes. The type of transformation used for the images was Wavelet Transformation (WT). Wavelet Transformation is very useful for coding in the discontinuities of an image, which is generally found in Intaglio printings - they are high in contrast. The mechanics behind Wavelet Transformation is beyond the scope of this paper, and as such only the applications and relevance of the Wavelet Transformation tool will be mentioned. In figure 1 histograms are made for the four visual attributes.

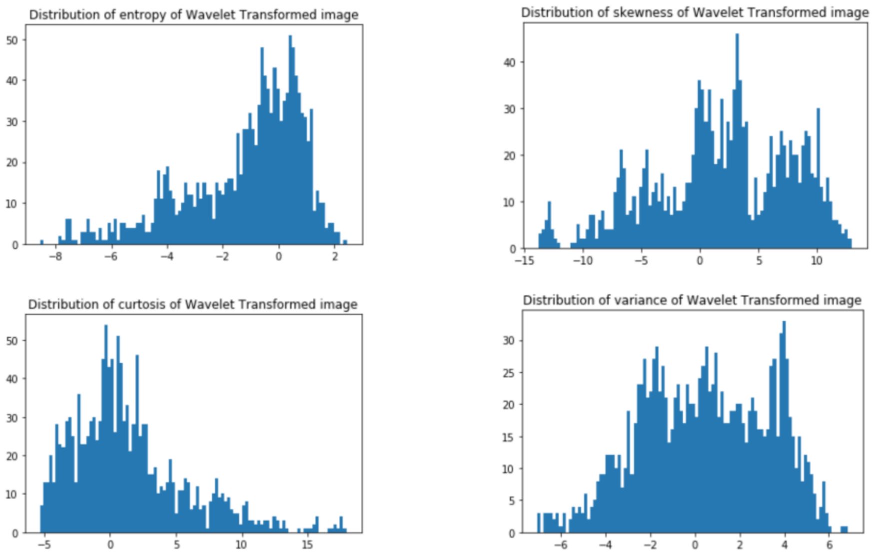

Figure 1: Histograms of the four visual attributes, displaying the distribution of each attribute

Approach

After downloading the dataset as a text file from its source, we loaded it into jupyter. First of all, the data has to be preprocessed, before the training of the classifiers could take place. As the dataset was organized in such a way that the authentic examples and the forged examples were clustered together, the data had to be randomized before it was able to be split it in into a training and test set. After randomization took place, the features and labels were assigned to separate arrays. We also decides to standardize the data, as the values of the different features were on different scales. After the preprocessing, our data was ready to start the training phase of the classifiers. We chose three different classifiers, which were logistic regression, support vector machines and neural networks. All three classifiers were trained using the different functions form the Scikit learn package in python. In the training phase, we first used the default parameters of each classifier, and later on tuned several parameters for each classifier. For the logistic regression classifier, the parameter C, which is the inverse of the regularization parameter λ, was trained. For the support vector machine classifier, we trained the penalty parameter C, the kernel type and the kernel coefficient. Lastly, for neural networks, the number of hidden layers, the type of activation function and the penalty parameter α were tuned. After finding the optimal parameters for each classifier, they were trained again using these parameters.To evaluate the performance of each classifier, a confusion matrix was made to visualize the number of misclassifications, and the values of the error, accuracy, tp-rate, fp-rate and precision were computed.

Choices made and justifications

The first choice to be made was the types of classifier to be used. The research of Gillich and Lohweg (2010) used different types of the support vector machines classifier. Thus, in order to replicate their research, this study also used the support vector machine classifier. Neural networks, is a useful classification method to use, because it can compute very complicated functions in order to make complex predictions, while also yielding a generalisable classification function. Since this classification process only takes four features, using this classifier is not too computationally expensive. Finally, logistic regression is a rather simple and quick classifier, that is not as complex as Neural Networks or Support Vector Machine. Therefore, if logistic regression classifies well enough, it is not necessary to use more complicated programs. We chose to optimise as many parameters as was computationally doable, and chose the parameters that impact the algorithm most effectively.

Results

	Logistic regression	SVM	Neural Networks
Error	0.0128	0.0055	0.000
Accuracy	0.9872	0.9945	1.0
tp-rate	0.9916	1.0	1.0
fp-rate	0.0161	0.0096	0.000
Precision	0.9916	1.0	1.0

Figure 2: Evaluation values of the classifiers

For all three classifiers, error, accuracy, tp-rate (true-positive rate), fp-rate (false positive rate) and precision were measured using the testset. As can be seen in figure 2, all three classifiers are very accurate. The logistic regression classifier had the lowest accuracy score of 0.99872, and neural networks had the highest score with a perfect accuracy of 1.0.

	Meaning of confusion matrices		Logistic Regression		Support Vector Machine		Neural networks	
	TN	FP	306	5	308	3	323	0
	FN	TP	2	236	0	238	0	226
Number of actual forged banknotes	FN+TP		238		238		226	

Figure 3: Confusion matrices of the classifiers

The confusion matrices (fig. 3) display the number of true positives (lower right corner), true negatives (upper left corner), false positives (upper right corner) and false negatives (lower left corner) Both SVM and neural networks also had a perfect tp-rate, which can also be seen in their confusion matrices as there are no false negatives. The logistic regression classifier did have two false negatives, along with five false positives. SVM also had three false positives, while neural networks had no misclassification at all which is of course already demonstrated in the accuracy score of 1.0

Evaluation of approach and results

All classifiers worked very well. The original research by Gillich & Lohweg (2010), which used the support vector machines algorithm had a 100 percent accuracy, thus this study's classifier did not work as well as the one used in the original research. The neural network classifier had the best scores for all evaluating methods. At the same time, neural networks also is the most computationally expensive classifier. Therefore, all three classifiers have pros and cons. However, when the aim is picking out forged notes, it is vital that both the hits are as high as possible and the false alarms are as low as possible, because forged banknotes disrupt the cash flow and throwing out real banknotes, because of false alarms is expensive and literally a waste of money. Therefore, this study proposed that the neural networks algorithm arguably is most suitable to check the authenticity of banknotes.

Commented [3]: lol

References

1. Gillich E, Lohweg V. 2010. Banknote Authentication. Jahreskolloquium Bild. der Autom. 1:1–8.