Comparing Machine Learning Classification Algorithms for the Detection of Counterfeit Banknotes from Image Data

Andrea Fleming, Israel Montiel, and Dustin Oakes

***Abstract*— This project uses different machine learning algorithms to build a classifier that can identify counterfeit banknotes from image data. Governments across the world spend untold amounts incorporating anti-counterfeiting measures into their physical currency, so it is important to be able to easily distinguish genuine banknotes from fake money. With the widespread adoption of mobile phone cameras, an image-based counterfeit detector could improve transaction security and confidence in a currency's stability. Banknote data is obtained from the UCI Machine Learning Repository, to which we apply four different classification algorithms: K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF). The SVM and KNN algorithms perform best according to the results from the confusion matrix and the area under the ROC curve.**

***Index Terms*—Machine Learning, Classification, Probability**

I. Introduction

Even with the recent rise of cashless and contactless payments, combating the circulation of counterfeit money is an important priority of treasuries and central banks worldwide. With improvements in digital photocopying and editing technologies it has become easier than ever before to produce counterfeit banknotes, which makes accurate detection of genuine and fake money necessary to ensure the public's confidence in a currency. UV light and iodine detector pens are commonly used but are prone to rejecting genuine notes [1]. Central banks use special paper, security threads, and watermarks to ensure the security of the money supply, which theoretically could expose flaws in images of counterfeit banknotes [2]. Using wavelet transformations of images of counterfeit and real banknotes, we can apply machine learning classification algorithms to explore the statistical variations between genuine and fake notes.

Below is a brief introduction of the four classification algorithms used.

A. Logistic Regression

Logistic regression is a basic classification technique that calculates the probability of a target variable being positive or negative, or one or zero. Each response represents a binary class, which is predicted based on a logic specified in the model. Commonly, probabilities over 0.5 will be classified as a positive result, while probabilities lower than 0.5 will classify as a negative response. In our data set, a negative (0) result indicates genuine currency, while a positive (1) result indicates counterfeit money being detected.

B. Random Forest

Random forest is a supervised learning algorithm that creates an ensemble of decision trees, and it works well to classify large data sets with accuracy. The algorithm works by combining a series of random 'weak learners' to build a single 'strong learner'. The added randomness of random forests should work to fix overfitting issues the decision trees can be prone to.

C. Support Vector Machine

SVM is a classification technique that identifies the best possible boundary between binary labels given their features. For example, in a two-feature dataset, a simple SVM model will attempt to find the thickest line that separates the classes. The same intuition extends to the multi-feature space by using hyperplane separators. Kernels allow the model to take non-linear shapes in their classification. Though there are several flavors of SVM kernels, we focus on Radial Bias Function which allow for classes to be completely encircled. Linear and polynomial variants were also tested, but produced worse performance.

D. K-Nearest Neighbors

The KNN algorithm is one of the simplest Supervised Machine Learning algorithms mostly used for classification. It stores all cases and classifies new cases based on a similarity measure. The K in KNN is a parameter that refers to the number of nearest neighbors to include in the majority voting process. To find the nearest neighbor, it will calculate the Euclidean distance between two points in the plane with coordinates (x, y) and (a, b). KNN can ideally be used when data is labeled, noise free, and the dataset is small. How do we choose K? Choosing the right value of K is a process called parameter tuning and is

Andrea Fleming is with the Department of Economics, UCLA, CA 90025 USA (e-mail: akflemin21@ucla.edu).

Israel Montiel is with the Department of Economics, UCLA, CA 90025 USA (e-mail: imontiel@ucla.edu).

Dustin Oakes is with the Department of Economics, UCLA, CA 90025 USA (e-mail: dustinoakes@ucla.edu).

important for better accuracy. As the K changes, the answer changes depending on where the point is, which drastically changes the result. We want to eliminate too much bias when choosing the K parameter. If K is too low, bias is based on those observations closest and it might be too noisy. If K is too big, it might take too long to process.

II. Task Description

Using real and counterfeit banknote image data, we construct several models from their extracted features for legitimacy classification. In total, we fit four models to the preprocessed data. Classification score is used to choose between model variants (ex. Linear and RBF SVM) while ROC and confusion matrix metrics decide our final model evaluation.

III. Major Challenges and Solutions

We chose this dataset in part to understand how machine learning models recognize image data. The transformed nature of the features made fitting our models straightforward. However, our major challenge lies in the interpretability of our results and its application to other datasets. If high kurtosis, for example, was found to be a significant indicator of counterfeits, it would be difficult to give recommendations to a human on what this means and how to identify it. Considering the traffic of retail stores, it might be unrealistic to expect consistency when these indicators are small. Indeed, electronic fraud detection machines are now commonplace and offer a reliable method for achieving the high accuracy we found in this project.

IV. Experiments

A. *Dataset description*

Three of the four features (Variance, Skewness & Kurtosis) used in this analysis are derived from the wavelet transformation of each 660 dpi image (*n=1372)*. In short, a wavelet can be thought of as a brief, wave-like disturbance in an otherwise flat line. If we imagine a bird in the sky, for example, the bird would represent the "wave-like disturbance" against a flat background. Since these disturbances usually represent the most interesting parts of an image, wavelet transformations are commonly used for object detection and noise reduction. The entropy variable is a measure of randomness of an image, constructed through analysis of histograms where each discrete pixel value acts as a bin on the histogram's axis. Specifically, the variables are:

1) Variance of Wavelet Transformed image (continuous); min = -7.0421, max = 6.8248
2) Skewness of Wavelet Transformed image (continuous); min = -13.7731, max = 12.9516
3) Kurtosis of Wavelet Transformed image (continuous); min = -5.2861, max = 17.9274
4) Entropy of image (continuous); min = -8.5482, max = 2.4495

These features were used as inputs to our four classification algorithms to decide from an image whether a banknote was genuine (0) or a counterfeit (1).

Figure 1
Scatterplot Matrix

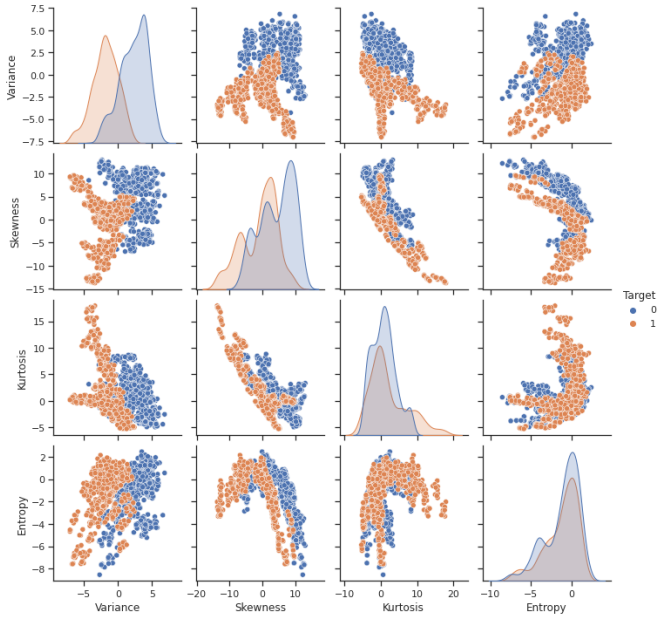
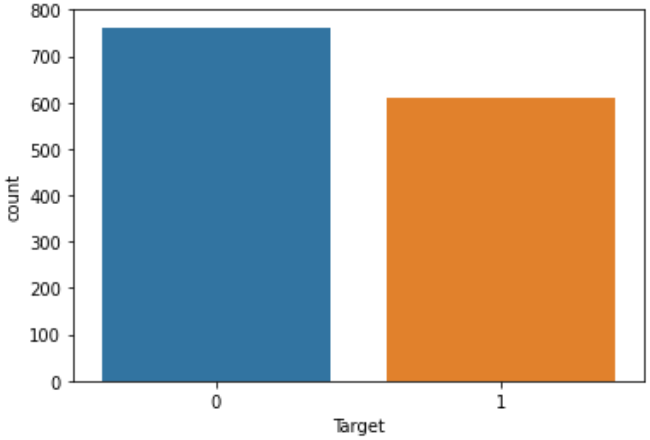


Figure 2
Count of Target Variable



We randomly split the data into training dataset and testing dataset, respectively. In this research, we train on 70% of the data (960 data points) and test with the other 30%.

B. *Data exploration*

The data direct from the UCI Machine Learning Repository is complete with no missing values. Descriptive statistics were taken of the features, but as stated in the previous section, their interpretation is unclear due to the wavelet transformation. Even so, a graphical representation of the features can still be useful since visual patterns can emerge that are not apparent in the raw data. For instance, the following 3D-plot of three features might suggest that the two classes could be linearly separable on the (Skewness, Kurtosis) plane. The 2D-scatterplot matrix projections, however, are not so clear.

Though the intersection between features does not look severe, we hypothesized that it would be enough to render the proximity models such as SVM and KNN useful, given that there does seem to be separation between the classes.

For data exploration, we take a look at a correlation plot to see if any of our variables would have a strong correlation with one another. Variables are said to have a significant correlation to one another if the correlation coefficient is above .70 or below -.70. Variables are said to have a moderate correlation if above .50. In our variables, we see moderate correlation between Skewness and Kurtosis, and between Entropy and Skewness. We notice a strong correlation between the Variance and the Target Variable.

Variance seems like it will be a strong predictor for whether a banknote is real or counterfeit. Our classifiers are now trained with the training data according to each of their separate algorithms.

FIGURE 3
CORRELATION MATRIX

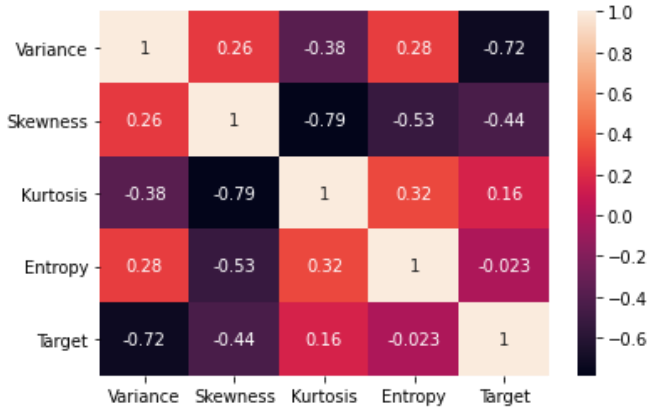


FIGURE 4
3D VISUALIZATION

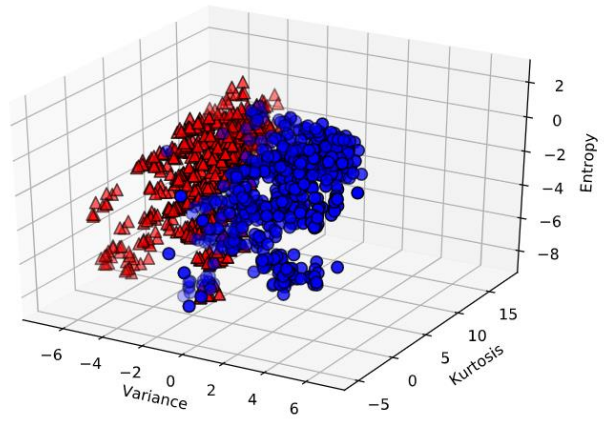
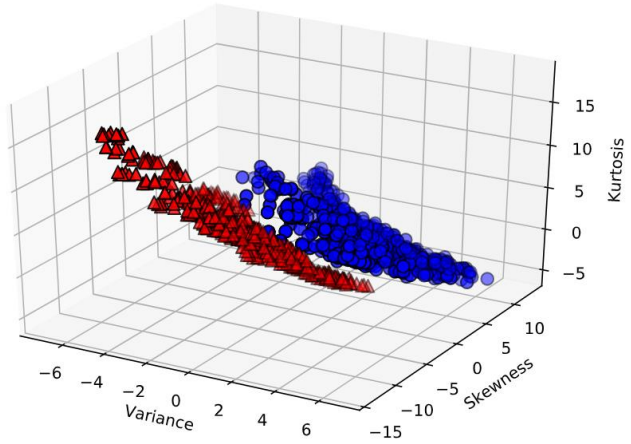


FIGURE 5
3D VISUALIZATION



C. *Evaluation metrics*

We use the confusion matrix concept to evaluate our classification algorithms for effectiveness. Accuracy score is the primary measure to gauge whether a particular algorithm works well to identify fake banknotes, but we also aim to minimize the presence of false rejections, where genuine currency is identified as fake. Rejecting customers' real notes creates hassle for making transactions and indicates higher than actual rates of counterfeiting, which can unsettle peoples' feelings about the security of the money supply. F-1 score provides a balance between the precision and recall statistics generated by the confusion matrix, so we will also consider that in our evaluations. Finally, we examine the ROC curve (Receiver Operating Characteristic curve) to provide a visual analysis of how our classifiers work, and to guide our choice of a classifier that is adequate to avoid the detection of real currency as fake so we can avoid false rejection occurring too often.

TABLE 1
EVALUATION METRICS FOR CLASSIFICATION ALGORITHMS

Metrics	Equation	Evaluation Emphasis
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	Accuracy metric measures the ratio of correct predictions over the total number of instances evaluated.
Precision	$\frac{TP}{TP+FP}$	Precision metric measures the proportion of counterfeit identifications that were correct.
Recall	$\frac{TP}{TP+FN}$	Recall metric is used to measure the proportion of actual counterfeits measured correctly.
F1-score	$\frac{2*Recall*Precision}{Recall+Precision}$	F1-score metric represents the harmonic mean between recall and precision values.

Note: TP - true positive; FP - false positive; FN - false negative; TN - true negative

D. *Results and analysis*

The results of our classification techniques varied in terms of the accuracy results, but certain methods yielded impressive results. The application of advanced methods like the Support Vector Machine (SVM) algorithm returned good results, with the SVM (1.000) classifier correctly identifying 100% of the banknotes correctly. In addition, even simpler methods like the KNN (1.000) classifier worked very well. Similarly, the classifiers created using the Logistic Regression and Random Forest algorithms worked well, on the order of identifying about 98-99% of the banknotes correctly depending on the adjustment of random states and hyper-parameters. However,

since the LR (0.987) and RF (0.996) methods were prone to returning false positives and false negatives, the SVM and KNN classifiers seem to be our top-performing models.

TABLE 2
CLASSIFICATION ACCURACY

Classification Method	Accuracy Score
Logistic Regression (LR)	0.987
Random Forest (RF)	0.996
Support Vector Machine (SVM)	1.000
K-Nearest Neighbors (KNN)	1.000

Aside from the simple accuracy scores reported in Table 2, other evaluation metrics were constructed from the confusion matrices and evaluated in the model comparison (See Table 3 for the other measures). Every model scored a perfect 1.0 in the recall metric, but the precision metric serves as a way to distinguish between model performance. The SVM (1.000) and KNN (1.000) models both identified every banknote correctly, so they also have perfect precision scores. Meanwhile, the LR (0.973) and RF (0.989) classifiers both mistakenly picked up genuine currency as counterfeit, which impacted the precision scores of those models.

As the harmonic mean of the two, F1 Score serves as a balance between the Precision and Recall statistics. According to this F1 classification, SVM (1.000) and KNN (1.000) remained the best models with their perfect scores, while the RF (0.994) and LR (0.986) algorithms were again penalized for their mistakes.

TABLE 3
EVALUATION METRICS

Method	Precision	Recall	F1-Score	ROC Area
Logistic Regression	0.973	1.000	0.986	0.989
Random Forest	0.989	1.000	0.994	0.995
Support Vector Machine	1.000	1.000	1.000	1.000
K-Nearest Neighbors	1.000	1.000	1.000	1.000

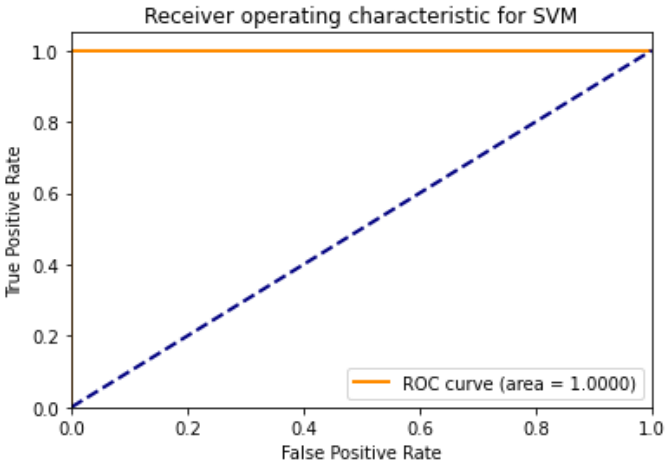
Finally, we take the Receiver Operating Characteristic (ROC) curves and compare the area underneath the curves (AUC) to decide what our best model is. The Logistic Regression (0.989) classifier returned the ROC curve with the smallest area underneath, and the Random Forest (0.995) algorithm generated an ROC curve with an area slightly larger. Even though both these classification methods proved to work quite well, we strive to avoid mistakenly identifying counterfeit money as real. Given that, the ROC curve returned by the SVM (1.000) and KNN (1.000) classifiers are much more attractive for their seemingly perfect ability to identify real currency as authentic and counterfeit as fake.

V. CONCLUSION AND FUTURE WORK

Though our best-performing models certainly indicate success, we must consider the possible anomalies that limit the applicability of our method. For future work, a larger, more diverse set of data should be surveyed. Since we are secondary users of this dataset, we do not know how simple the sample banknotes were designed or how egregious the counterfeit errors looked. As mentioned previously, many central banks have introduced visual and physical security features to their notes that are difficult to reproduce through simple printing. More advanced features, though, might not be present in our 2013 sample. The variability of the wave transformations, therefore, might be measuring features that very clearly indicate fraud, such as miscentering or printing quality.

Even so, we believe this project is indicative of the strong potential for machine learning in fraud detection. Our results show that SVM and KNN produce the highest quality results across several metrics. This is reasonable when the amount of viable features is limited and there is little data clustering, but we predict worse results as new technology improves the ability of counterfeit printers. Therefore, it is important that future researchers remain open to multiple models and incorporate additional non-visual parameters.

FIGURE 6
ROC CURVE – SUPPORT VECTOR MACHINE



REFERENCES

[1] Parliamentary Office of Science and Technology (POST). 1996. POSTnote 77 - Counterfeit Banknotes. [online] Available at: https://post.parliament.uk/research-briefings/post-pn-77/ [Accessed 21 March 2021].

[2] European Central Bank. 2021. *Security features*. [online] Available at: https://www.ecb.europa.eu/euro/banknotes/security/html/index.en.html [Accessed 21 March 2021].

[3] V. Lohweg, "Banknote Authentication Data Set." August 2012. Distributed by University of California, Irvine Machine Learning Repository. [online] Available at: https://archive.ics.uci.edu/ml/datasets/banknote+authentication [Accessed 21 March 2021].

FIGURE 7
ROC CURVE – LOGISTIC REGRESSION

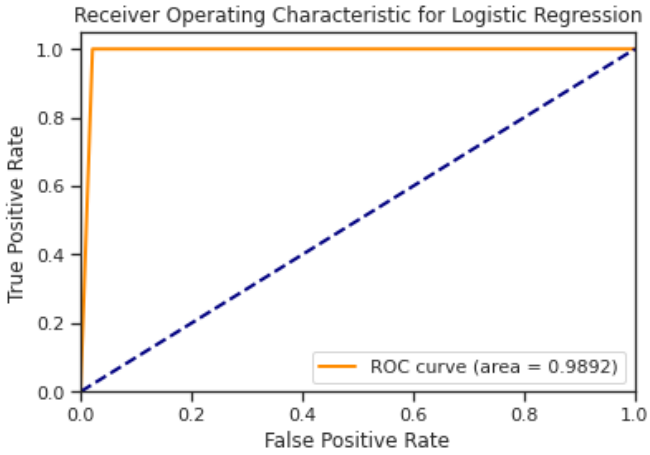


FIGURE 8
ROC CURVE – RANDOM FOREST

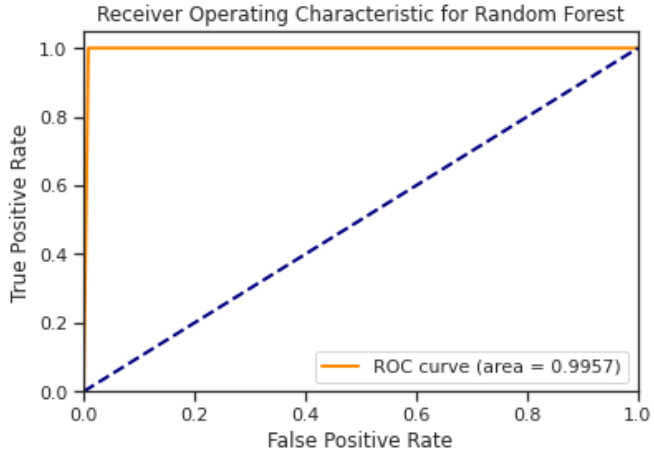


FIGURE 10
ROC CURVE – K-NEAREST NEIGHBORS

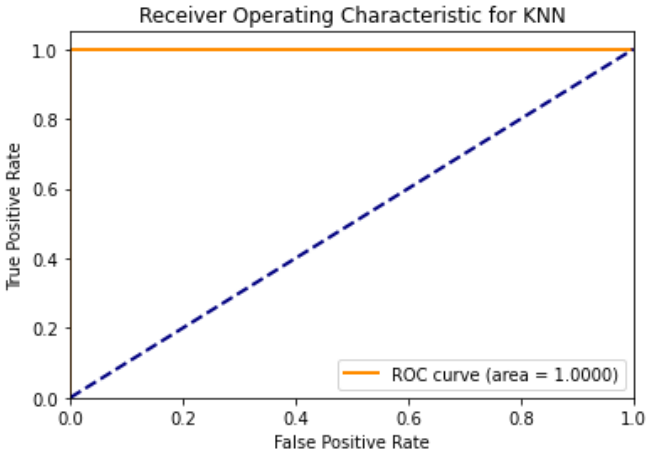


FIGURE 11
RANDOM FOREST – IMPORTANT FEATURES

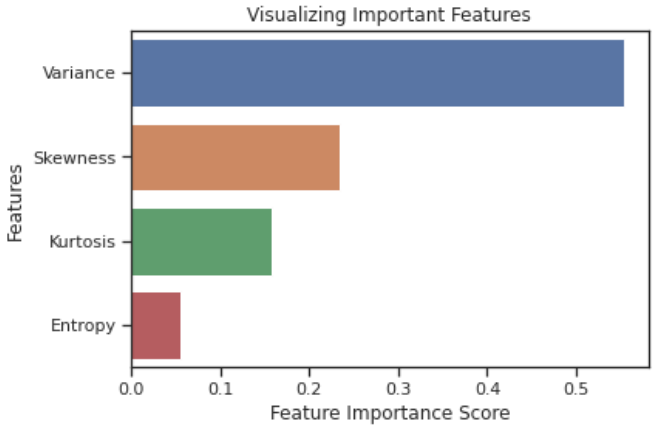


FIGURE 12
KNN – SELECTING OPTIMAL K-VALUE

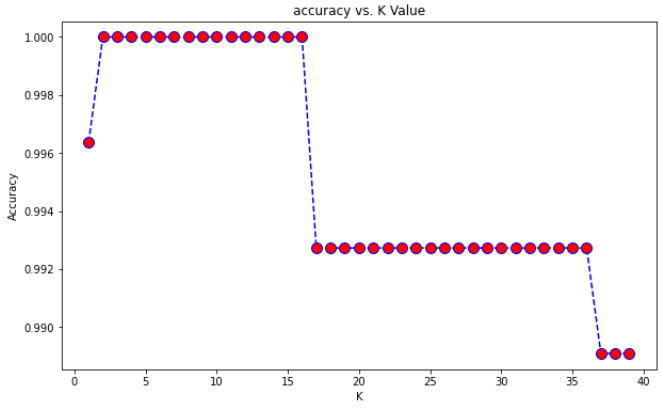


FIGURE 13
3D VISUALIZATION