South African Banknote Denomination and Authenticity Classification

Sahil Lawton[1], Rachanvinod Gowda[2], Nadia Naidoo[3], and Micara Ramnanan[4]

[1] School of Mathematics, Statistics and Computer Science
[2] University of Kwazulu-Natal, Durban KZN 4000, South Africa
{215007249,217079267,216050475,217032715}@stu.ukzn.ac.za

Abstract. Due to the rapid automation of monetary transactions through paper currency, methods for reliably performing these transactions are important. The automated systems used in these transactions need to be able to accurately identify the denomination of the paper currency, as well as establish the authenticity of the currency. In this paper, we will explore the use of various techniques used across the image processing pipeline for the classification and validation of a bank note, as well as investigate the use of the VGG-16 architecture in combination with various image preprocessing techniques, in the task of banknote classification.

Keywords: VGG-16 · Banknote classification · Image processing.

1 Introduction

In the world today, 3 percent of the worlds' money can be found in actual physical cash. As negligible as this may seem, it still amounts to trillions of dollars in valuation. Given this, it is essential for automated cash transaction systems to be as accurate as possible with the transactions. These automated systems need to be equipped with the ability to classify the denomination of a banknote, as well as to determine the authenticity of the note.

A variety of techniques have been proposed in the task of classifying a banknote. Some of these methods involve the use of the KNN algorithm[3][8], text-based feature methods which use Random Forest, Support Vector Machines and Logistic Regression methodologies, Markov models[5], Principal Component Analysis[2], Speed Up Robust Feature (SURF) models[7], and ORB descriptors[4]. However, some of these methods suffer from lacking's such as the requirement of human experts to manually design algorithms to indicate how feature extraction is done, among other things.

A transfer learning approach proposed by Mohamed A. Saleh et al. [1] overcomes many of the lacking's found in other approaches. The paper[8] investigates the use of the VGG-16 , AlexNet, and GoogleNet architectures. The architectures demonstrated impressive performances, with the VGG-16 producing the highest accuracy of 100%.

Our banknote classification system incorporates the VGG-16 architecture in combination with various preprocessing methods in an attempt to create a framework that is able to accurately classify a bank notes denomination, as well as its authenticity. Examples of doctored notes are shown in figure 1 and the comparison between a doctored note and a genuine note is shown in figure 2.

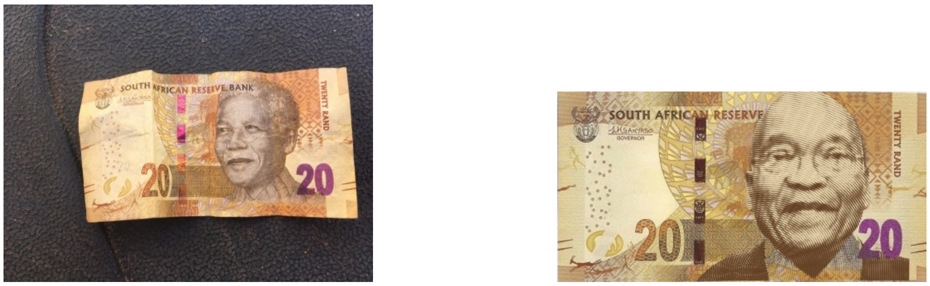

Fig. 1. Visual representation of Genuine vs Fake banknotes

2 Methodology

Our system follows a standard image processing pipeline as shown in figure 2 below. The information that follows describes each aspect of the image processing pipeline.

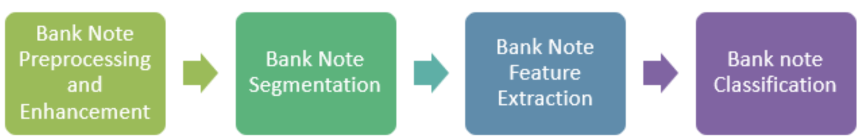

Fig. 2.

2.1 Banknote Image Preprocessing and Enhancement

If a banknote has been in circulation for a long time, it may be difficult to easily extract the relevant features that allow for accurate classification of its denomination and authenticity. This difficulty is due to the quality of the banknote due to degradation. Degradation can occur due to constant contact with dirt and sebum from multiple users' hands and other surfaces, as well as other factors.

Degradation of the banknote, combined with the fact that an image captured for image processing may not have been obtained under ideal conditions, emphasizes the need for the preprocessing and enhancement stages of the image processing pipeline.

The methods which constitute the preprocessing phase of our image processing pipeline are described below:

a) The use of filtering: Our framework uses a Wiener filter[1][10][9] to allow for the reduction of noise and blur found in the banknote images. The Wiener filter is a low-pass filter that performs equally well on both clean and dirty notes. It uses adaptive filtering based on the mean and variance estimation of the local neighborhood of each pixel. Wiener filter also preserves the edges and other useful details. A Wiener filter is shown to outperform a Median filter in both the reduction of Speckle and Gaussian noise, however, the Median filter is shown to outperform the Wiener filter in the reduction of salt and pepper noise.[**?**]

Figure 3 below demonstrates the effectiveness of the Median and Wiener filter in restoring an image that was subject to degradation. The addition of Gaussian noise is used to demonstrate the performances of both filters, and a kernel of 9x9 is used for both filtering. The figure suggests that the Wiener filter is more suited to our purposes than the Median filter, due to the fact that while both filters seem to reduce noise well, the Wiener filter outputs an image that is clearer than the Median Filter.

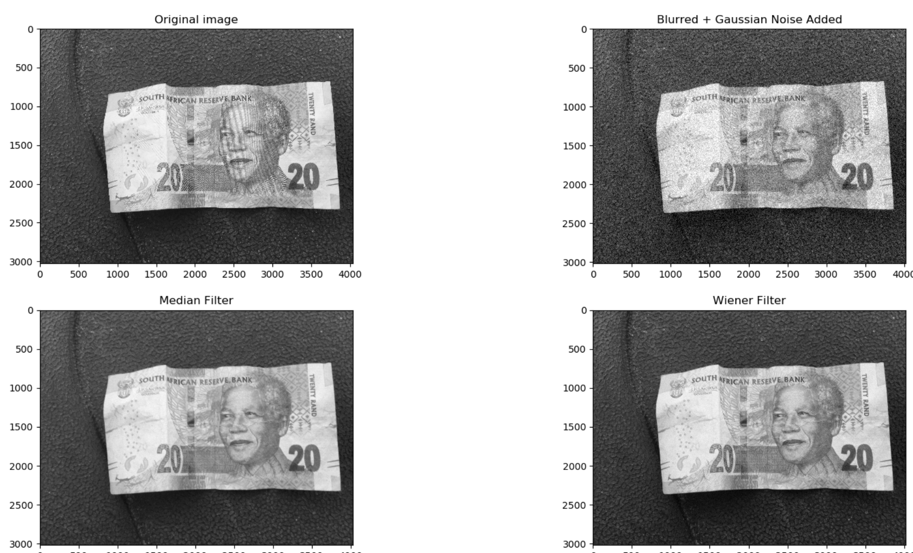

Fig. 3.

b) The Use of Contrast Limited Adaptive Histogram Equalization: The use of histogram equalization on the filtered images allows for a contrast adjustment of the image using the image's histogram. However, the limitation of regular histogram equalization is the fact that it only works well when the distribution of the pixel values is similar throughout the image. When some regions are significantly different (e.g., lighter or darker) from other parts of the image, the contrast in some regions might not be enhanced correctly. To compensate for this shortcoming, one can compute multiple histograms at the same time using adaptive histogram equalization. However, adaptive histogram equalization can over-amplify the contrast in near-constant regions of the image. This over-amplification may cause noise in particular regions of the image.

To solve the shortcomings of regular adaptive histogram equalization, our framework implemented contrast limited adaptive histogram equalization. Contrast limited adaptive histogram equalization(CLAHE)[6] differs from adaptive histogram equalization in its contrast limiting. In the case of CLAHE, the contrast limiting procedure is applied to each neighborhood from which a transformation function is derived, which allows for a reduction in noise. Figure 4 below shows the difference between the input image, an image that regular histogram equalization is used on, and an image that CLAHE is used on.

Fig. 4.

c) The use of data augmentation: The final step in our preprocessing methodology is the use of data augmentation. This data augmentation involves processes such as shearing, rotating, and scaling of our training data to create dummy data from our original data, which will allow for a higher validation accuracy when training the model. It also assists in training the model to be invariant to images of different sizes and rotations. The data augmentation is an essential step in preventing overfitting and will result in a model with higher overall accuracy.

Figure 5 below shows a performance graph of an earlier model we tested for our framework. The graphs show a large difference between validation and training accuracy before the data augmentation(left graph), and a lesser difference after data augmentation(right graph). This indicates that there is a lower degree of overfitting after the use of data augmentation, and hence it results in an increase in the performance of our model.

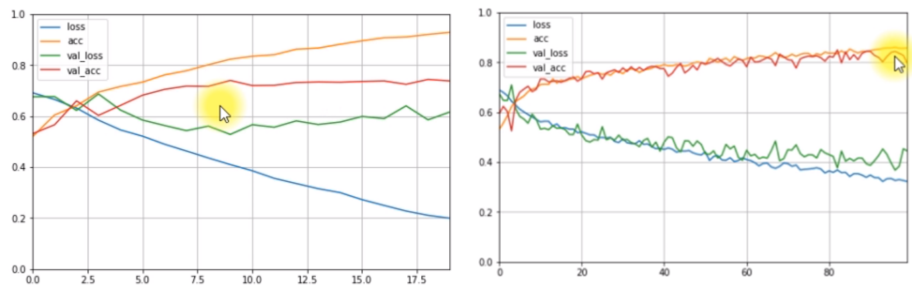

Fig. 5.

2.2 Bank Note Segmentation

The image segmentation portion of our framework involves dividing an input into segments to simplify image analysis. Image segmentation sorts pixels into larger components, eliminating the need to consider individual pixels as units of observation. A few image segmentation techniques include:

a) Thresholding: a process of dividing an image into a foreground and background by separating pixels into one of two levels, using a threshold to determine which of the two levels an original pixel gets separated into. Thresholding converts grayscale images into binary images or distinguishes the lighter and darker pixels of a color image.

b) K-means clustering: K-means clustering is an unsupervised learning algorithm which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest centroid. The algorithm aims to minimize the squared Euclidean distances between the observation and the centroid of the cluster to which it belongs.

c) Histogram-based image segmentation: Edge detection uses the process of identifying sharp changes or discontinuities in brightness. Edge detection usually involves arranging points of discontinuity into line segments, or edges. Edge detection makes use of such things like filters and using these filters to convolve over an image, identify regions of discontinuity based on outlying values, and define boundaries based on this information.

d) Edge detection segmentation: Edge detection uses the process of identifying sharp changes or discontinuities in brightness. Edge detection usually involves arranging points of discontinuity into line segments, or edges. Edge detection makes use of such things like filters and using these filters to convolve over an image, identify regions of discontinuity based on outlying values, and define boundaries based on this information.

These segmentation techniques, however are less efficient than the deep learning approach that we took due to their reliance on rigid algorithms, as well as their requirements of human intervention and expertise.

We use a VGG-16 architecture that we used in our implementation. The VGG-16 architecture performs automatic semantic segmentation of an image. Semantic segmentation results in each pixel being labeled with the class of its enclosed object or region. For example, in figure 6[6], we can see each pixel of a person, a bicycle, and the background being semantically differentiated.

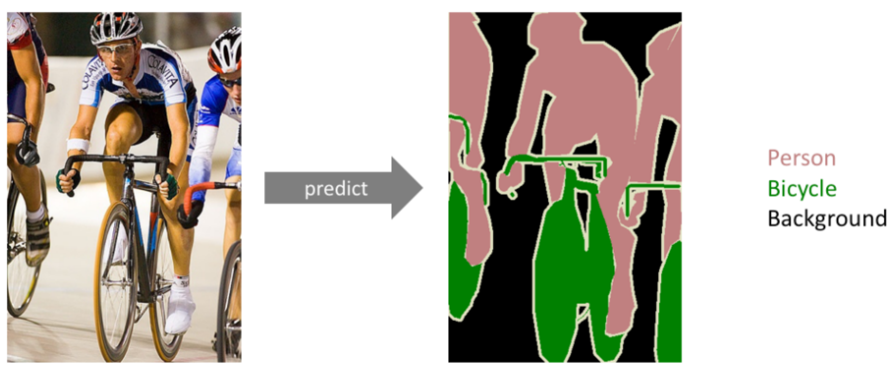

Fig. 6.

Effectively, our architecture's goal is to take an image and produce a segmentation map, where each pixel contains a class label, represented as an integer, as shown below[biker]. The semantic segmentation component differs from the classification component, in that during semantic segmentation, individual pixels are assigned to specific classes, while in classification, the entire image is assigned to a certain class. Figure 7 shows the desired output of semantic segmentation, which is each individual pixel being given a label, assigning it to individual classes. The banknote recognition system semantically segments sections of the image, such as the number that indicates the banknote value and the animals or flowers present on certain notes, which indicate it belonging to a certain class.

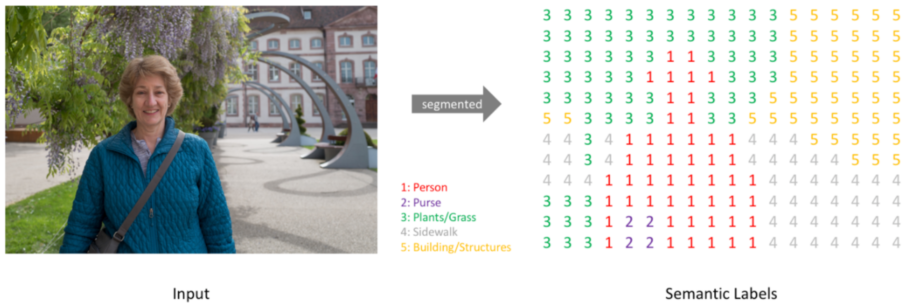

Fig. 7.

Our semantic segmentation is automatically performed by the VGG-16 architecture that we used in our framework. Figure 8[14] below shows a structural diagram of the VGG-16 architecture.

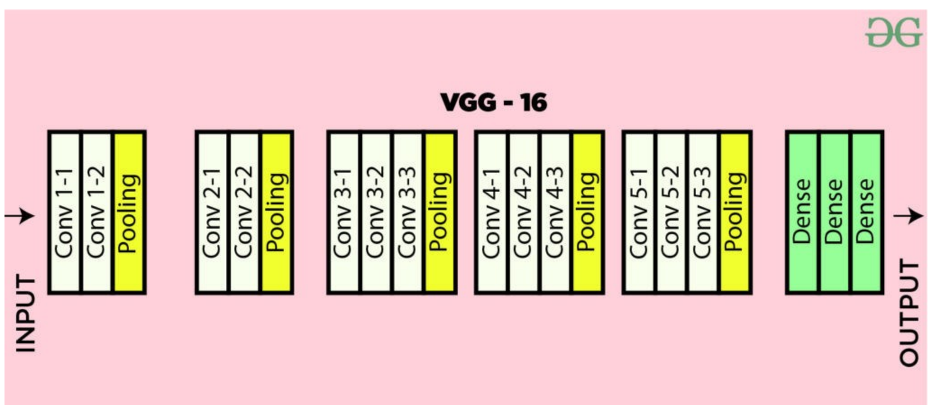

Fig. 8. Structural diagram of the VGG-16 architecture

The architecture makes use of convolutional and max-pooling layers for various pixel-level convolutions of the input image, which converts 2D matrices of features into different 2D matrices of features. The convolution layers are made up of sets of filters which are moved over each pixel of an image, and the operations performed on the initial images create different feature maps of images. The feature maps are used for both the semantic segmentation and feature extraction aspects of the architecture. The pooling layers reduce the size of the input representation, as well as the required number of parameters in the model, hence reducing the computation time required. The pooling layers applied to feature maps also contribute to the creation of pooled feature maps. The pooling layers also help with spatial variance, as well as control overfitting.

It is also worth noting, that due to our architectures automatic creation of feature maps based on the input data, the use of data augmentation in the preprocessing step results in features that are more invariant to such things as scale, rotation, and flipping, which may not occur if no data augmentation is performed.

2.3 Bank Note Feature Extraction

As mentioned above, an automatic process of feature extraction if performed through the use of convolution layers in the network. The system learns to do feature extraction with no pre-trained features required, but instead extracts features directly from images. As shown above, the features across layers are convoluted with different filters to generate more invariant features. No manual feature extraction is required.

The final pooled feature map is flattened at the last section of the convolutional base of our network, and a 4096 dimension feature vector is obtained (4096 is the standard feature vector size of a VGG-16 architecture). This feature vector is processed in the classification stage, which will be discussed next.

2.4 Bank Note classification

The classification stage of our architecture makes use of a fully connected artificial neural network. The ANN's main purpose is to combine our features into more attributes, which will allow for more accuracy when predicting the classes. The initial layers of the neural network are responsible for the processing of lower-level features, such as lines, and the layers, further along, are responsible for the processing of higher-level, complex features such as numbers and shapes. The neurons are made up of various activation functions, such as ReLu functions, to help process the data. The final output layer is made up of a Softmax layer with ten neurons(five for real notes and five for fake notes), to allow for the classification of the banknote. The Softmax function, as shown below, results in each output at the output node adding up to one.

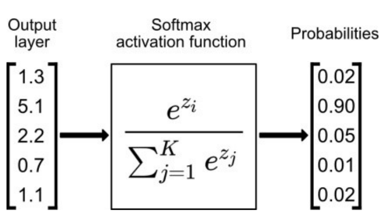

Fig. 9.

The outputs are decimal numbers, which correlate to percentage chances of a certain banknote belonging to a certain class.

The architecture trains itself to be as accurate as possible by using a process of error calculation and back-propagation to adjust the network. The adjustments are made on the weights and feature detectors to help optimize the model's performance. It is an iterative process that allows for the training of the data. The Adams optimizer is used, along with a categorical cross-entropy loss function. The longer the model is trained, the greater the chances of it increasing its classification accuracy.

3 Results and Discussion

This section provides details about the training and testing of our data. Our final models showed a testing accuracy of 100%. The figure below shows a training performance evaluation, indicating that the network reached an optimal training and validation accuracy approximately 200 epochs in.

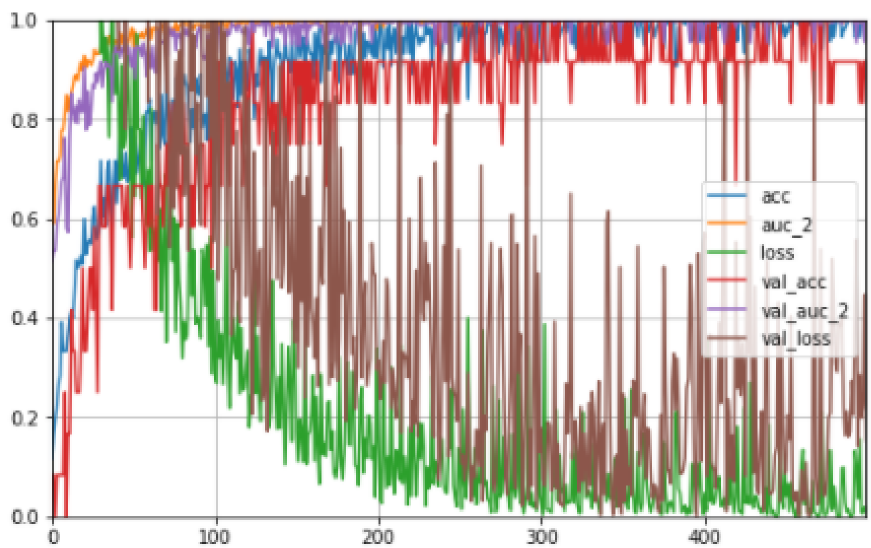

Fig. 10.

Furthermore, our model is capable of accurately classifying the denomination of a banknote in an image such as the one below as a real twenty rand note, which demonstrates its ability to isolate the relevant features necessary for processing from an image with a lot of unnecessary features, and giving an accurate classification. This also shows the models adaptability to things such as scaling and rotation. The average processing time of our model across all banknotes is approximately 0.6 seconds on an i5 4 generation laptop with 8GB of RAM.


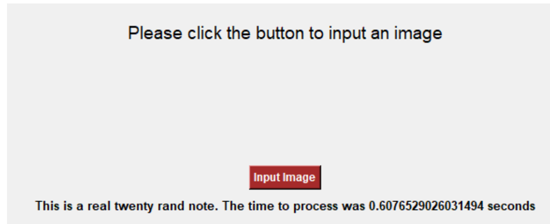

Fig. 11. Demonstration of how our model can identify a genuine R20 note on a good looking cat

4 Conclusion

Our paper explains the utilization of various preprocessing techniques and the use of Transfer Learning for the task of South African banknote recognition. The utilization of Wiener filtering in combination with contrast adaptive histogram equalization resulted in a great reduction of noise and an increase in feature contrast of our image. That in combination with the VGG-16 architecture resulted in a high-performance architecture that was able to classify our testing data with 100 percent accuracy.

References

1. Abd Almisreb, A., Saleh, M.A.: Transfer learning utilization for banknote recognition: a comparative study based on bosnian currency. Southeast Europe Journal of Soft Computing **8**(1) (2019)
2. Ahmadi, A., Omatu, S., Kosaka, T.: A pca based method for improving the reliability of bank note classifier machines. In: 3rd International Symposium on Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the. vol. 1, pp. 494–499. IEEE (2003)
3. Bae, R.E.T., Arboleda, E.R., Andilab, A., Dellosa, R.M.: Implementation of template matching, fuzzy logic and k nearest neighbor classifier on philippine banknote recognition system. Int. J. Sci. Technol. Res **8**(08), 1451–1453 (2019)
4. Hadisukmana, N., NP, A.Y.: Paper money recognizer using feature descriptor. Indonesian Journal of Electrical Engineering and Computer Science **12**(1), 117–126 (2018)
5. Hassanpour, H., Farahabadi, P.M.: Using hidden markov models for paper currency recognition. Expert Systems with Applications **36**(6), 10105–10111 (2009)
6. Kim, T.K., Paik, J.K., Kang, B.S.: Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering. IEEE Transactions on Consumer Electronics **44**(1), 82–87 (1998)
7. More, M.J.: The indian banknote recognition using speeded up robust features (surf). International Journal of Innovative Research in Computerand Communication Engineering **4**(12), 2152–2132 (2016)
8. Raho, G.I., Al-Khiat, A., Al-Hamami, A.H.: Cash currencies recognition using k-nearest neighbor classifier. Int. J. Web Semant. Technol **6**(4), 11–21 (2015)

9. Sargano, A.B., Sarfraz, M., Haq, N.: Robust features and paper currency recognition system. In: Proceedings of the 6th International Conference on Information Technology (2013)
10. Tran, L.A.: Image processing course project: Image filtering with wiener filter and median filter (04 2019). https://doi.org/10.13140/RG.2.2.15700.65921