MAY 19, 2018

SOUTH AFRICAN BANK NOTES RECOGNITION

COMP 702 PROJECT

VEROSHA PILLAY-214539347

Introduction

Functionality for recognizing banknotes are required in various machines, such as banknote-counting machines and automatic teller machines (ATMs). Banknote recognition is defined as the recognition of the type of banknote (e.g., R10, R20, and R200), the direction of the banknote, and the date of issue of the banknote (e.g., least recent, recent, and most recent). Furthermore, banknote recognition allows the condition of banknotes to be monitored.

Supervised learning was used for the recognition of South African bank notes. The Rand Notes dataset was used in the training of bank notes. A test image was then selected and classified. This system was made with Eclipse and Scene Builder using Java.

Methodology

With the ideas and research that have been done previously in this field. Firstly, an image is inputted from the Rand Notes dataset. Image is then pre-processed using the techniques median filtering ,histogram equalization and adaptive thresholding. Next the image is segmented using canny edge detection. Thereafter, feature extraction was done using Hu 7 invariant moments. Lastly In the classification and detection stage the technique k-nearest neighbour is then implemented. The necessary steps taken for bank note recognition is summarized in Figure 1.

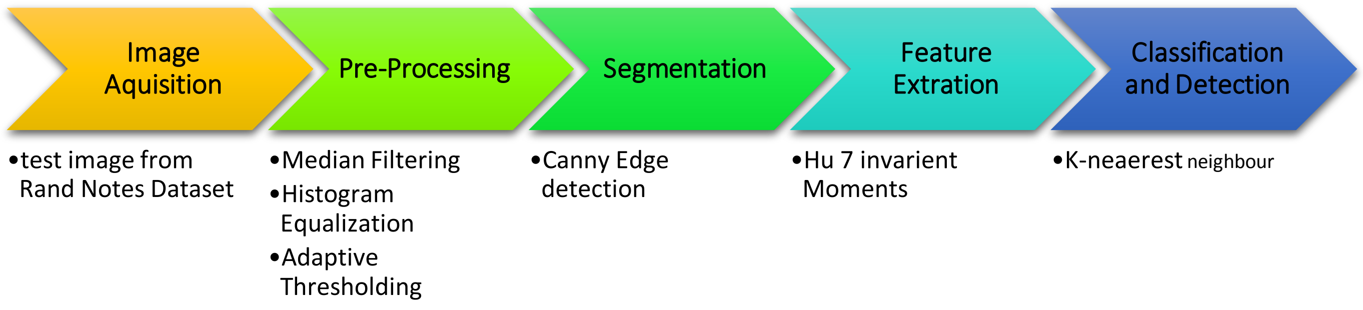
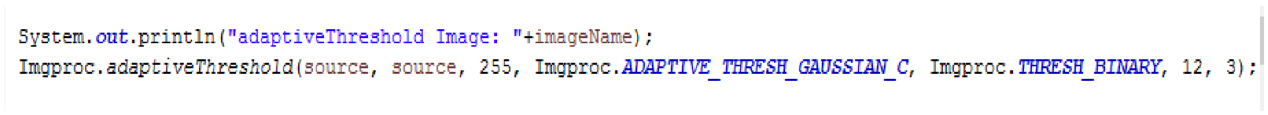


Figure 1: A diagram showing the steps of South African Bank Notes Recognition.

Image Acquisition:
Acquire the test image.

Pre-Processing:
Pre-processing begins with the use of the median filter(shown in **figure 9**) then histogram equalization is done on the image which is a technique for adjusting image intensities to enhance contrast (shown in **figure 6**). Adaptive thresholding is lastly applied to the image. In adaptive threshold unlike fixed threshold, the threshold value at each pixel location depends on the neighbouring pixel intensities (shown in figure 7 and a code snippet shown in figure 2). added to the bank note The median filter is a nonlinear digital filtering technique, often used to remove noise from an image. Last step in pre-processing is to enhance the image by brightening the bank note. Different alpha and beta values were tested till a set which was most proved to be the most useful was found and applied (shown in figure 11).

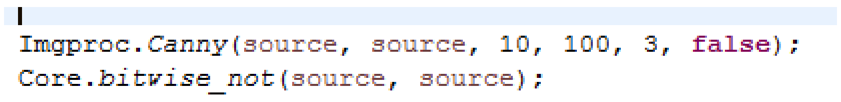


```
System.out.println("adaptiveThreshold Image: "+imageName);
Imgproc.adaptiveThreshold(source, source, 255, Imgproc.ADAPTIVE_THRESH_GAUSSIAN_C, Imgproc.THRESH_BINARY, 12, 3);
```

Figure 2: Code Snippet of adaptive threshold and values used

Segmentation

In this stage, the bank note is segmented using the technique Canny edge detection.
The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images (shown in **figure 3** code snippet).

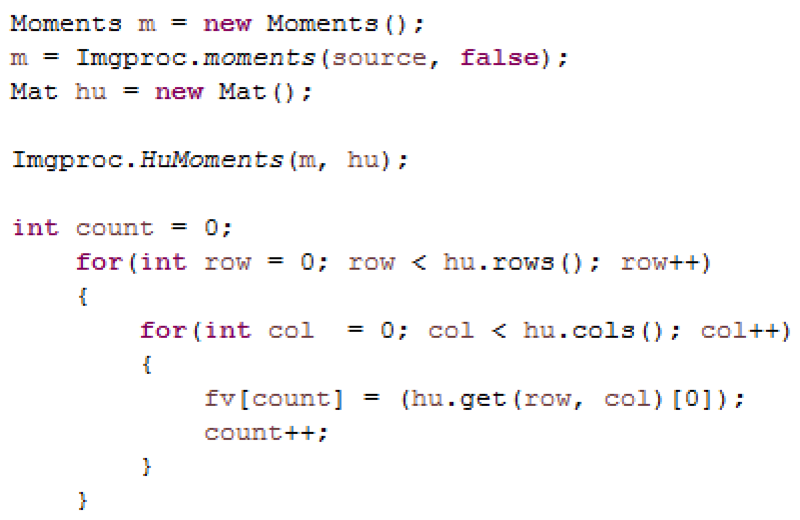


```
Imgproc.Canny(source, source, 10, 100, 3, false);
Core.bitwise_not(source, source);
```

Figure 3: Code Snippet of Canny edge detection

Feature Extraction

Hu 7 in variant moments is used for feature extraction (code snippet shown in **figure 4**). The moment invariants have been proved to be the adequate measures for tracing image patterns regarding the images translation, scaling and rotation under the assumption of images with continuous functions and noise-free.

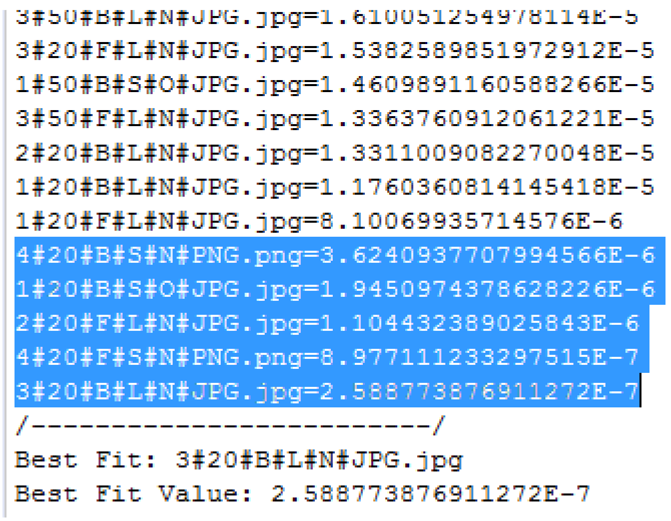


```
Moments m = new Moments();
m = Imgproc.moments(source, false);
Mat hu = new Mat();

Imgproc.HuMoments(m, hu);

int count = 0;
    for(int row = 0; row < hu.rows(); row++)
    {
        for(int col  = 0; col < hu.cols(); col++)
        {
            fv[count] = (hu.get(row, col)[0]);
            count++;
        }
    }
```

Figure 4: Code Snippet of the calculation of Hu Moments

Classification and Detection

- In this stage the technique k-nearest neighbour is used (example of results shown in Figure 5). In pattern recognition, the k-nearest neighbour's algorithm (k-NN) is a non-parametric method used for classification. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbour. In this stage the 5 nearest neighbours were used with the use of Euclidean distance.



```
3#50#B#L#N#JPG.jpg=1.6100512549781114E-5
3#20#F#L#N#JPG.jpg=1.5382589851972912E-5
1#50#B#S#O#JPG.jpg=1.4609891160588266E-5
3#50#F#L#N#JPG.jpg=1.3363760912061221E-5
2#20#B#L#N#JPG.jpg=1.3311009082270048E-5
1#20#B#L#N#JPG.jpg=1.1760360814145418E-5
1#20#F#L#N#JPG.jpg=8.10069935714576E-6
4#20#B#S#N#PNG.png=3.6240937707994566E-6
1#20#B#S#O#JPG.jpg=1.9450974378628226E-6
2#20#F#L#N#JPG.jpg=1.104432389025843E-6
4#20#F#S#N#PNG.png=8.977111233297515E-7
3#20#B#L#N#JPG.jpg=2.588773876911272E-7
/------------------------/
Best Fit: 3#20#B#L#N#JPG.jpg
Best Fit Value: 2.588773876911272E-7
```

Figure 5: Code Snippet of 5 nearest neighbours (testing with the back of a large R20.00)

Bank Note Pre-processing and Enhancement

Image is first converted to grayscale.

Techniques that were tested in pre-processing and enhancement are as follows:

- Histogram Equalization (shown in **figure 7**)
- Adaptive Thresholding (shown in **figure 8**)
- Median Filtering (shown in **figure 9**)
- Blur Filtering (shown in **figure 10**)
- Enhancing Brightness of an image with the use of different alpha and beta values (shown in **figure 11**)

Various orders for pre-processing were tested. With order 3 resulting in the best results for pre-processing the back and front of a bank note.

Tested Orders:

1. Blur filter → Histogram Equalization → Threshold → Enhance image

2. Histogram Equalization → Adaptive Threshold → Blur filtering → Enhance Image

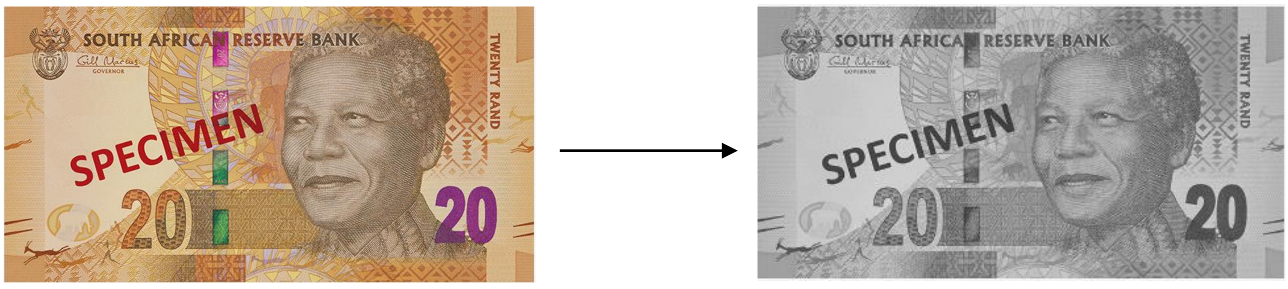
3. Median filter → Histogram Equalization→ Adaptive Threshold

4. Histogram Equalization → Adaptive Threshold → Median filtering → Enhance Image

5. Blur Filter → Histogram Equalization → Threshold → Median filtering → Enhance Image

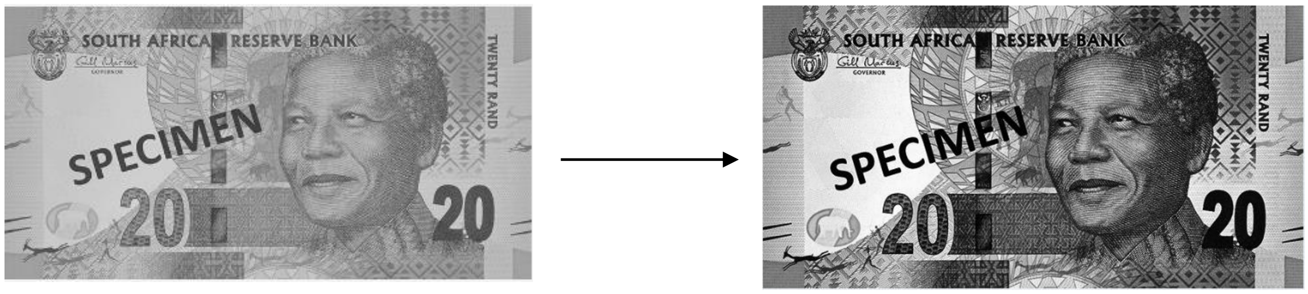
6. Histogram Equalization → Threshold → Median Filtering

Below are the different results from the above mentioned techniques on the front of a R20.00.



(a) Original Image (b) Image Converted to Grayscale

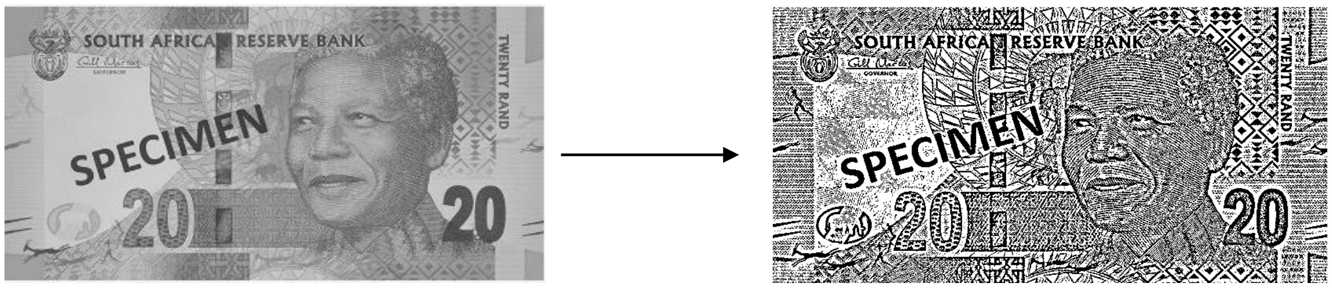
Figure 6: Original Image Converted to Grayscale



(c) Grayscale Image

(d) Histogram Equalization

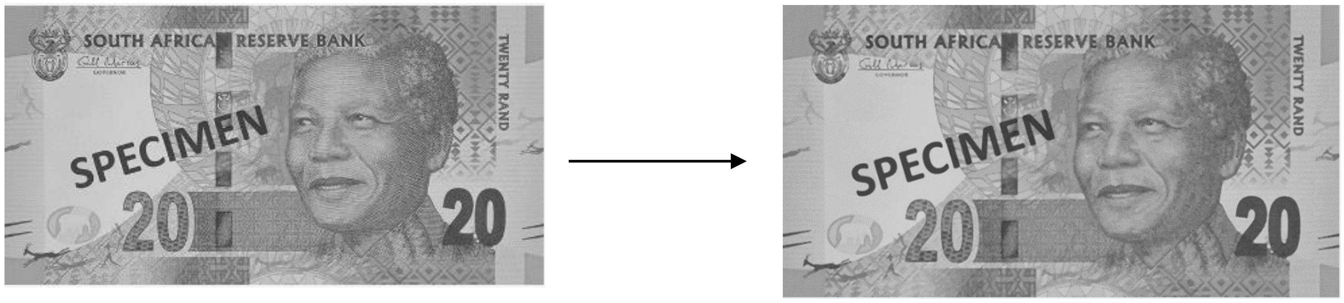
Figure 7: Result from Histogram Equalization



(e) Grayscale Image

(f) Adaptive Threshold

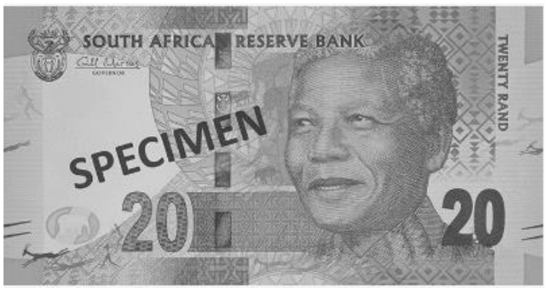
Figure 8: Result from Adaptive Thresholding

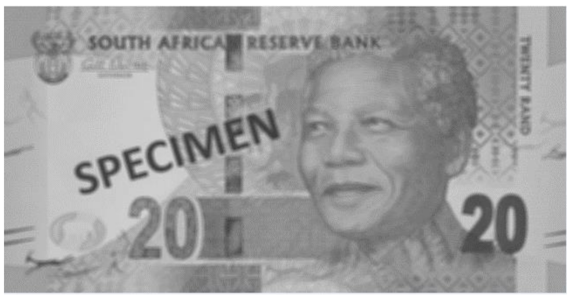


(g) Grayscale Image

(h) Median Filter

Figure 9: Result from Median filtering



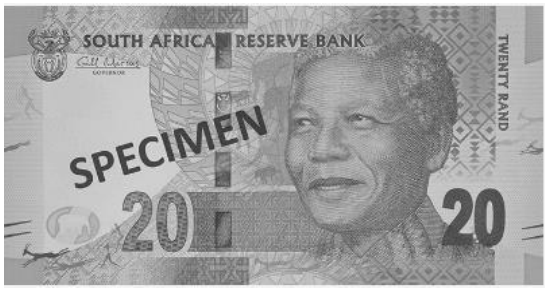


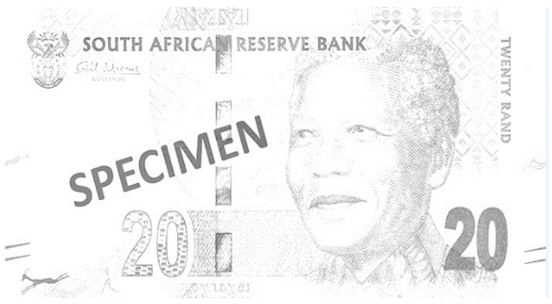
(i) Grayscale Image

(j) Blur Filter

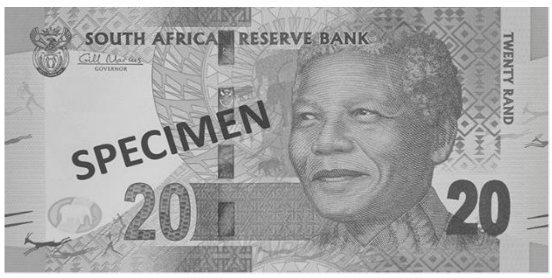
Figure 10: Result from Blur filtering

(k) Grayscale Image





(l) Brightened Image (alpha: 2 beta: 0.5



(j) Brightened Image (alpha: 1 beta: 5)

Figure 11: Result from Enhancing Brightness of the image

Bank Note Segmentation

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Edge detection is a fundamental tool used in most image processing applications to obtain information for feature extraction and object segmentation. Methods in edge detection transform original images into edge images which benefits from the changes of grey tones in the image.

Two edge detection techniques were investigated i.e. Canny Edge Detection and Sobel Edge Detection.

Canny Edge Detection:

The Canny Edge Detector is one of the most commonly used image processing tools, detecting edges in a very forceful manner. The steps in the Canny edge detector are as follows:

1. Smooth the image with a two dimensional Gaussian.

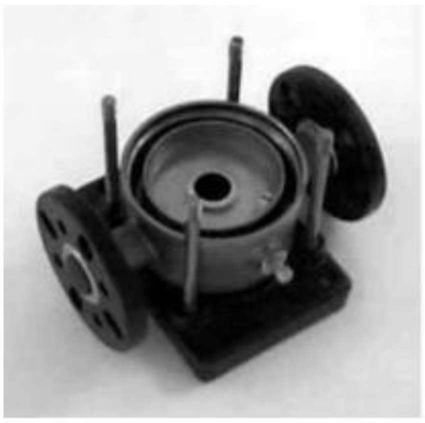
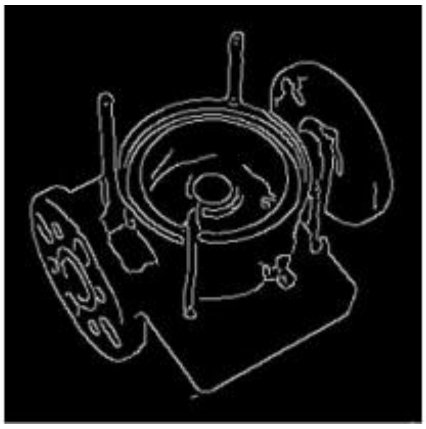
$$\boldsymbol{g(m,n) = G\sigma(m,n) * f(m,n)}$$

Where
$$G\sigma = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{m^2+n^2}{2\sigma^2})$$

2. Take the gradient of the image. This demonstrates changes in intensity, which indicates the occurrence of edges. This genuinely gives two consequences, the gradient in the x direction and the gradient in the y direction.

3. Non-maximal suppression- Edges will occur at points the where the gradient is at a maximum. Hence, all points at a maximum should not be suppressed. To facilitate this, the magnitude and direction of the gradient is computed at each pixel. After that for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not larger than both, suppress it.

4. Edge Thresholding- The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis". It makes utilize of both a high threshold and a low threshold. If a pixel has a value above the high threshold, it is set as an edge pixel. If a pixel has a value above the low threshold and is the neighbour of an edge pixel, it is set as an edge pixel as well. If a pixel has a value above the low threshold but is not the neighbour of an edge pixel, it is not set as an edge pixel. If a pixel has a value below the low threshold, it is never set as an edge pixel.

(a) (b)

Figure 13 : (a) Original Image. The resulting image after applying Canny Edge Detection(b).

Sobel Edge Detection:

Sobel method is applied to perform edge detection. The Sobel edge detector uses two masks with 3x3 sizes, one estimating the gradient in the x-direction and the other estimating the gradient in the y-direction. The mask is slid over the image, manipulating a square of pixels at a time. The algorithm calculates the gradient of the image intensity at each point, and then gives the direction to increase the image intensity at each point from light to dark. Edges areas represent strong intensity contrasts which are darker or brighter. Sobel algorithms work using a mathematical procedure called convolution and commonly analyse derivatives or second derivatives of the digital numbers over space. We implement the Sobel method for edges detection, which is based on a 3 by 3 array that is moved over the main image. The Sobel convolution kernels are designed to respond to edges vertically and horizontally. These masks are each convolved with the image. It calculates horizontal and vertical gradient (Gx and Gy), then combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. These numbers are used to compute the edge magnitude which given by:

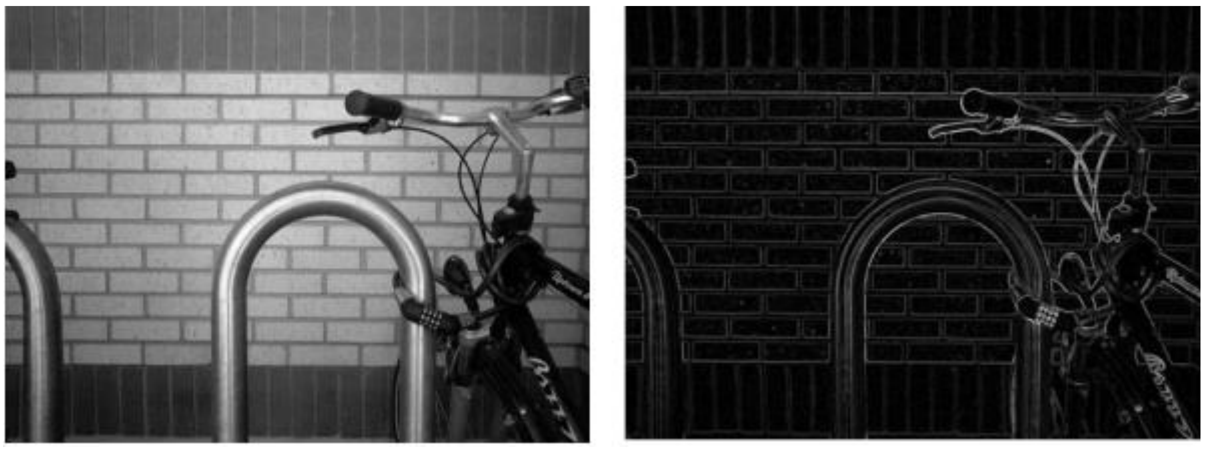
$$|G| = \sqrt{Gx^2 + Gy^2}$$

The vertical Mask of Sobel Operator

-1	0	1
-2	0	2
-1	0	1

The horizontal Mask of Sobel Operator

-1	-2	-1
0	0	0
1	2	-1



(a) (b)

Figure 13: (a) Original Image. The resulting image after applying Sobel method is shown in (b)

Comparison of Sobel and Canny Edge Detection:

1. Sobel Edge Detection:

The primary advantages of the Sobel operator lie in its simplicity. The Sobel method provides an approximation to the gradient magnitude. Another advantage of the Sobel operator is it can detect edges and their orientations. In this cross operator, the detection of edges and their orientations is said to be simple due to the approximation of the gradient magnitude.

On the other hand, there exist some disadvantages of the Sobel method. That is, it is sensitive to the noise. The magnitude of the edges will degrade as the level of noise present in image increases. As a result, Sobel operator accuracy suffers as the magnitude of the edges decreases. Overall, the Sobel method cannot produce accurate edge detection with thin and smooth edge.

2. Canny Edge Detection:

With Gaussian filter, any noise present in an image can be removed. The next advantage is enhancing the signal with respect to the noise ratio. This is done by non-maxima suppression method as it results in one pixel wide ridges as the output. The third advantage is better detection of edges especially in noisy state by applying thresholding method. The effectiveness of Canny method is affected by adjustable parameters. Small filters are desirable for detection of small, sharp lines, since it causes fewer instances of blurring. Large filters are desirable for detecting larger, smoother edges. However, it causes higher instances of blurring.

The main disadvantage of Canny edge detector is that it is time consuming, due to its complex computation

Conclusion: The Sobel operator is simple, but its accuracy suffers in noisy conditions. On the contrary, Canny edge detector has many favourable features such as smoothing effect to remove noise, and improving signal to noise ratio through a process known as non-maximal suppression. Since Canny edge detection has more favourable strengths as opposed to Sobel, Canny edge detection is used in this project for bank note image segmentation.

Bank Note Feature Extraction

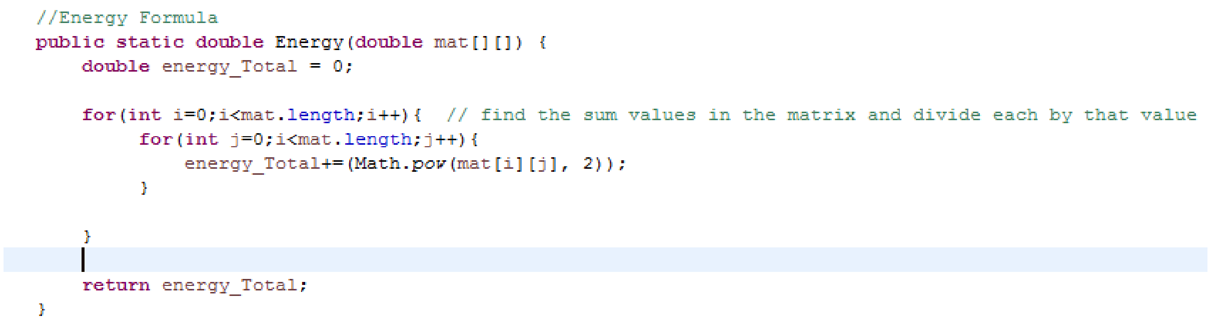
For feature extraction the following techniques were investigated and tested:

- GLCM (5 features selected for the feature vector)
- Hu 7 invariant moments

GLCM:

A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM. The GLCM functions characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix. GLCM directions of Analysis include 0 degrees,90 degrees,135 degrees and 45 degrees. Denoted as $P\phi, d(i, j)$ where ϕ is the direction (Degree) ,d is the distance and (i,j) is the row and column. In this project the GLCM was tested using a direction of 0 degrees and a distance of 1. Matrix was then normalized and five texture features were selected for use in the feature vector i.e.: energy, entropy, max probability, contrast and Homogeneity.

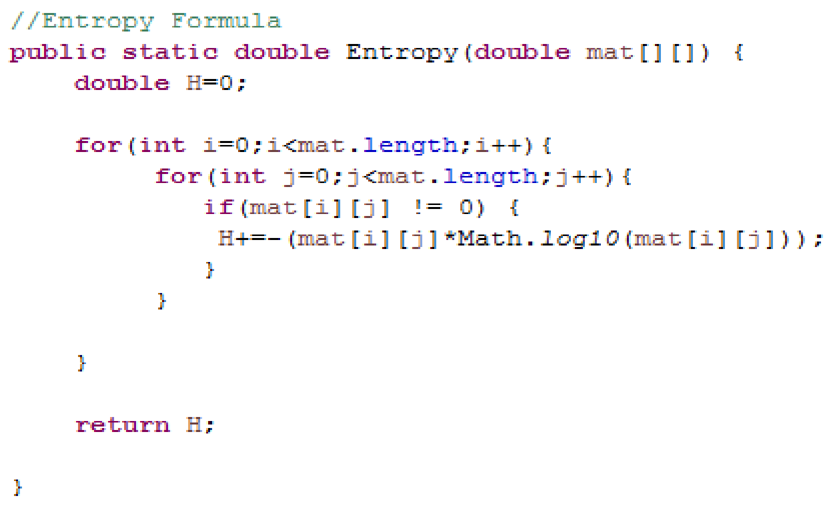
Energy: $\sum_{i,j} p(i,j)^2$



```
//Energy Formula
public static double Energy(double mat[][]) {
    double energy_Total = 0;

    for(int i=0;i<mat.length;i++){  // find the sum values in the matrix and divide each by that value
         for(int j=0;i<mat.length;j++){
             energy_Total+=(Math.pow(mat[i][j], 2));
         }

    }

    return energy_Total;
}
```

Figure 14 : Code snippet of energy calculation

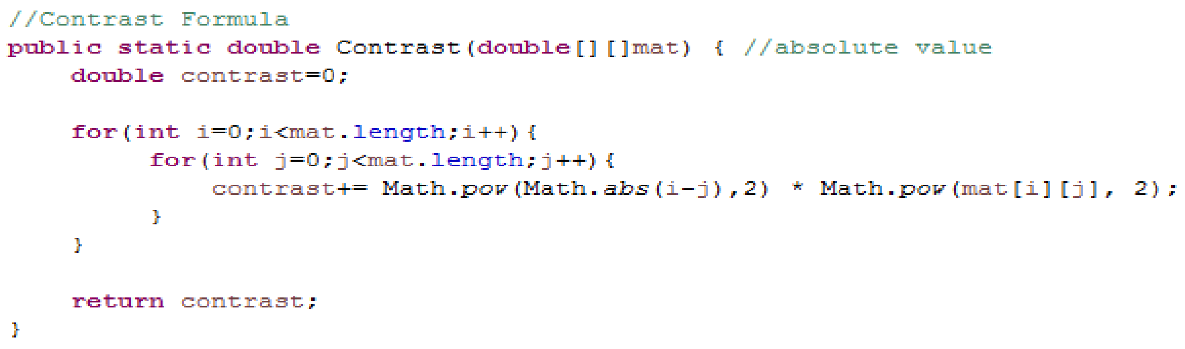
Entropy: $H = -\sum_{K=0}^{L-1} P(rk) * log_{10}\ (P(rk))$



```
//Entropy Formula
public static double Entropy(double mat[][]) {
    double H=0;

    for(int i=0;i<mat.length;i++){
         for(int j=0;j<mat.length;j++){
             if(mat[i][j] != 0) {
              H+=-(mat[i][j]*Math.log10(mat[i][j]));
             }
         }

    }

    return H;

}
```

Figure 15: Code snippet of entropy calculation

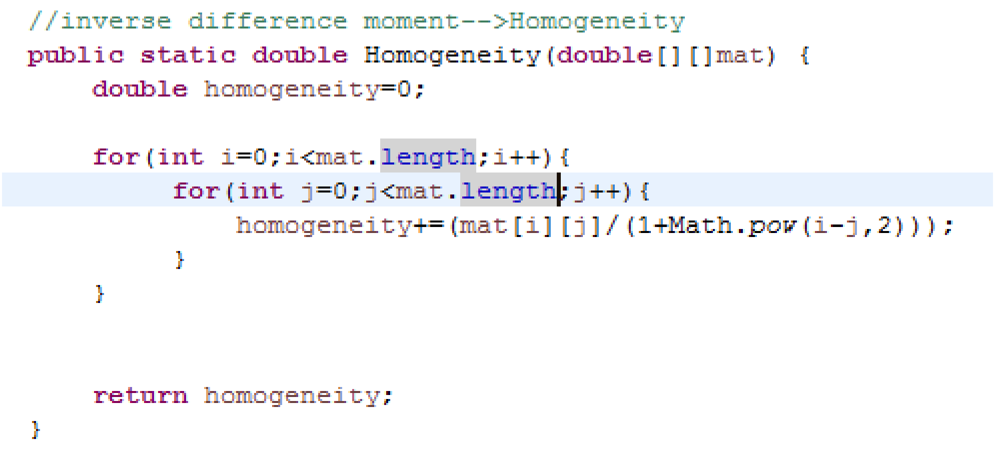
Contrast : $\sum_{i,j} |i-j|^2\ p(i,j)$



```
//Contrast Formula
public static double Contrast(double[][]mat) { //absolute value
    double contrast=0;

    for(int i=0;i<mat.length;i++){
         for(int j=0;j<mat.length;j++){
             contrast+= Math.pow(Math.abs(i-j),2) * Math.pow(mat[i][j], 2);
         }
    }

    return contrast;
}
```

Figure 16: Code snippet of contrast calculation

Homogeneity: $\sum_{i,j} \frac{p(i,j)}{1+|i-j|}$



```
//inverse difference moment-->Homogeneity
public static double Homogeneity(double[][]mat) {
    double homogeneity=0;

    for(int i=0;i<mat.length;i++){
        for(int j=0;j<mat.length;j++){
            homogeneity+=(mat[i][j]/(1+Math.pow(i-j,2)));
        }
    }


    return homogeneity;
}
```

Figure 17: Code snippet of Homogeneity calculation

Hu 7 invariant moments:

Based on normalized central moments, Hu introduced seven moment invariants (shown in **figure 18** below):

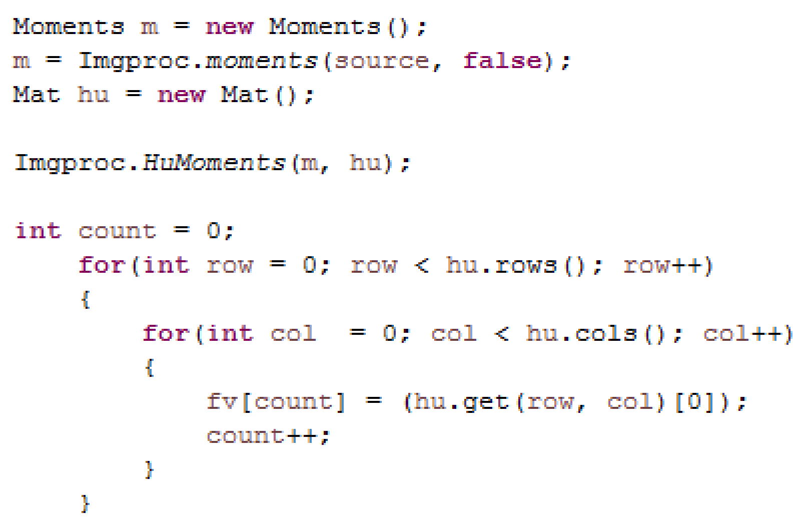
$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

Figure 18: Image of Hu 7 invariant moments

The moments are calculated which are invariant under:

- Sides
- Scale
- Rotation

The following figure (**figure 19**) shows how this was performed and used as a feature vector for classification and detection:



```
Moments m = new Moments();
m = Imgproc.moments(source, false);
Mat hu = new Mat();

Imgproc.HuMoments(m, hu);

int count = 0;
    for(int row = 0; row < hu.rows(); row++)
    {
        for(int col  = 0; col < hu.cols(); col++)
        {
            fv[count] = (hu.get(row, col)[0]);
            count++;
        }
    }
```

Figure 19: Code snippet of Hu 7 invariant moments

Conclusion: After processing a bank note with the use of GLCM, there were instances in which had atie between determing the note e.g. : 5 nearest neighbours (R20,R50,R100,R100,R20),the results would sometimes output the incorrect note value. After testing the bank note using Hu 7 invariant moments the above mentioned problem was resolved .Thus, the selected method for feature extraction is with the use of Hu 7 invariant moments.

Bank Note Classification:

Two techniques were investigated for classification:

- K-Nearest Neighbour (KNN)
- Support Vector Machine (SVM)

KNN :

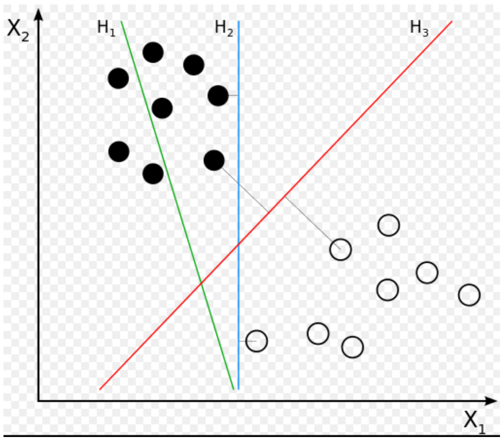
The classification technique used in this project is K-nearest neighbour. KNN assumes that the data is in a feature space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance –Euclidean distance used for this project. Each of the training data consists of a set of vectors and class label associated with each vector. Also single number "k" is given. This number decides how many neighbours (where neighbours is defined based on the distance metric) influence the classification, in this project KNN-5 is used.

Advantages of KNN	Disadvantages of KNN
Robust to noisy training data	Computation cost is high since each the distance of each query instance to all training samples.
Effective if the training data is large	Need to determine the value of parameter K(number of nearest neighbours)

SVM:

This technique was investigated but not implemented in the project.

Support vector machines are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. The SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.



- H1 does not separate the classes
- H2 does but only with a small margin
- H3 separates classes with a maximum margin gap

Figure 20: Illustration of different Gaps

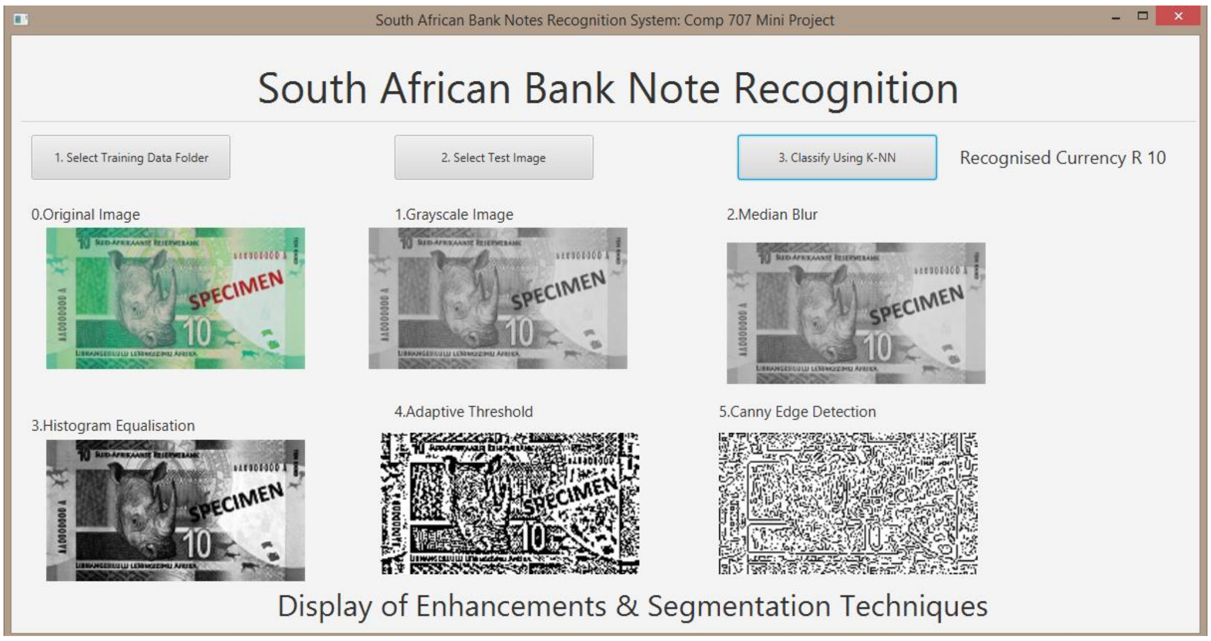
Advantages of SVM	Disadvantages of SVM
Over fitting is not common	Retraining is difficult
Fast Classification	No explanation capability
Good generalization capacity	Slow training
Works well with high dimensional data	

Results and Discussions:

Bank Note Classification Results: (Sample of 10 images tested below)

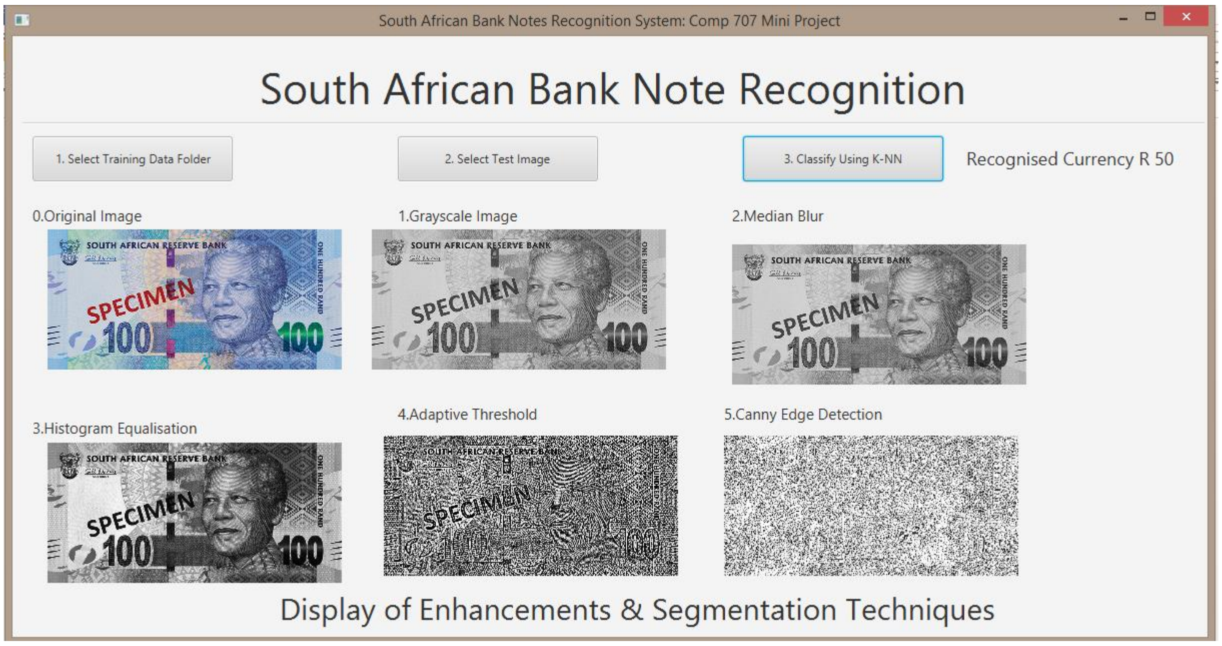
Image Tested	Classification Result
10back.jpg	Correct
10obverse.jpg	Correct
20back_large.jpg	Incorrect
050obverse-th.png	Correct
100front.jpg	Correct
200Front_large.jpg	Incorrect
t_r200n2.jpg	Correct

An Example of correctly classified bank note:



South African Bank Notes Recognition System: Comp 707 Mini Project
South African Bank Note Recognition
1. Select Training Data Folder
2. Select Test Image
3. Classify Using K-NN
Recognised Currency R 10
0.Original Image
1.Grayscale Image
2.Median Blur
3.Histogram Equalisation
4.Adaptive Threshold
5.Canny Edge Detection
Display of Enhancements & Segmentation Techniques

An Example of incorrectly classified bank note:


South African Bank Notes Recognition System: Comp 707 Mini Project
South African Bank Note Recognition
1. Select Training Data Folder
2. Select Test Image
3. Classify Using K-NN
Recognised Currency R 50
0.Original Image
1.Grayscale Image
2.Median Blur
3.Histogram Equalisation
4.Adaptive Threshold
5.Canny Edge Detection
Display of Enhancements & Segmentation Techniques

Total Results: (50 images in total)

- 43 Images identified correctly
- 7 Images incorrectly classified
- Accuracy = 43/50 * 100 → 86% (After testing all images)

Images that were incorrectly classified were some of the large new bank notes for the back and front side.

Conclusions:

As shown from the previous sections the best techniques to use were:

Pre-Processing:
Median filtering →Histogram Equalization → Adaptive Threshold

Segmentation:
Canny Edge Detection

Feature Extraction:
Hu 7 invariant moments

Classification:
K-nearest neighbour (K=5)

It can be seen that the success rate of the system is fairly high. Results can be improved by :

- Using a larger dataset for training
- Using a different classification technique such as SVM, neural networks etc.
- Using a different filters in pre-processing such Gaussian filtering

References:

- https://link.springer.com/chapter/10.1007/978-3-642-23199-5_39
  <Accessed 20-05-18>
- https://ieeexplore.ieee.org/document/6619121/
  <Accessed 20-05-18>
- Lecture Notes – slide 10
- https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/
  <Accessed 20-05-18>
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
  <Accessed 20-05-18>
- http://opencv-java-tutorials.readthedocs.io/en/latest/
  <Accessed 20-05-18>
- https://www.researchgate.net/publication/224146066_Analysis_of_Hu's_moment_invariants_on_image_scaling_and_rotation
- <Accessed 20-05-18>
- https://www.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glcm.html
  <Accessed 20-05-18>
- https://support.echoview.com/WebHelp/Windows_and_Dialog_Boxes/Dialog_Boxes/Variable_properties_dialog_box/Operator_pages/GLCM_Texture_Features.html
  <Accessed 20-05-18>
- https://en.wikipedia.org/wiki/Median_filter
  <Accessed 20-05-18>
- https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html
  <Accessed 20-05-18>
- https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html
  <Accessed 20-05-18>