U.S. CURRENCY DETECTION AND CLASSIFICATION

Graham Fuller
Jacob Oakes
Sydney Satchwill
Payden Beyer

Rose-Hulman Institute of Technology
Email: fullerga@rose-hulman.edu
oakesja@rose-hulman.edu
satchwsm@rose-hulman.edu
beyerpc@rose-hulman.edu

ABSTRACT

For our system, our objective was to take an overhead view image of money on a table and determine the value of it. We will discuss our approach to the problem which uses Hough transforms and hue to segment the image. To classify individual bills and coins we used multiple convolution neural nets as well as MATLAB's pattern net. We were unable to reach our goal of being able to determine the value of both coins and bills in the same image, but we were able to distinguish between different coins with an accuracy of 85.03%. If we were able to properly segment bills, we could classify them with 75.00% accuracy.

1. INTRODUCTION

There are times when people need to be able to count money quickly, which has been solved using machines that use weight and size to determine the total. A cheaper way of doing this would be using image recognition. The aspect of saving companies money and the challenges that it brings to image recognition is what makes this problem interesting.

The most challenging part of this problem is that most US currency looks very similar. For example, all US bills are the same color and size, and most US coins with the exception of pennies are the same color. There are also multiple versions of each currency including fifty states quarters. Rotated and overlapping bills and coins also adds complexity.

Our proposed solution is interesting because it uses multiple classifiers that use different features including edges, histograms of distances, and RGB color to classify coins. For both coins and bills we explored convolution neural nets, which are an emerging technology that helped us boost our accuracy.

2. LITERATURE REVIEW

Marco, Ronneberger, and Burkhardt used Hough transforms to segment their images to easily find coins [1]. We used this idea and obtained good results by implementing a coin finder that searched for circles of arbitrary size using Hough transforms. For paper currency detection Chakraborty, Basumatary, Dasgupta, Kalita, and Mukherjee used distinguishing features such as serial numbers, size, and texture to classify paper currency [2]. Most of these features were not applicable to US paper currency since all bills are the same size and generally the same color. However, this led us to using the upper left hand corner which shows the denomination as features. Wang and Lin used color and the fact that Swedish currency they were classifying had specific patterns for each denomination [3]. As stated before, using color was not applicable for distinguishing US paper currency, however each US paper denomination has a unique pattern to it. This is something that could be investigated in future work. Van der Maaten and Boon used statistical features such as distance distribution, angle distribution, and a combination of both as features in their classifier [4]. We used these ideas to build a neural net classifier that uses a histogram of distances from the origin as features.

3. PROCESS

3.1 Preprocessing

For preprocessing the image, we resize it so that the minimum of the height and width becomes 300 pixels while maintaining its aspect ratio. This resizing is done to help speed up the time taken to segment the image because larger pictures can take over a minute to segment. After the image is resized, it is segmented into coins and bills, which are then classified.

3.2 Coins

With the processed image, possible coins are found using a Hough transform. First edge pixels are found by using a Sobel filter. Each edge pixel casts a vote for all possible circles it could be a part of. We cast votes for circles with a radius range from 10 pixels to half the size of the image. The direction of the gradient is used to speed this up by only looking at angles from the plus or minus ten degrees from this direction. Next, the local maxima of the voting space are found. Local maxima that are not above a threshold are removed. Finally, circles that have nearby centers are combined. This process is the most time consuming part our system and is the main reason why we resize the image.

After this process, we have the center and radius of all the circles in the image that are possibly coins. These circles are extracted from the image similar to Figure 1. The circles are then processed through a number of classifiers. The first classifier we use is based off of RGB color. This classifier can distinguish if a coin is a penny or not since pennies have a unique copper color. The rest of the coins are then classified using a classifier built specifically for dimes, nickels, and quarters. This classifier is comprised of three sub-classifiers. The first of these is a convolution neural network that takes a 28 by 28 pixel image. The second is a pattern net that takes a histogram of distances of the edges of the coin from the center using 140 bins. Figure 2 shows examples of histograms produced from a dime, nickel, and quarter. The outer ten percent of the coin is not used in the histogram because the outer parts of most coins are the same. The third is a pattern net that takes a 40 by 40 pixel edge mask using a Sobel filter as a feature. Our classifier used the results from all three of these to make a decision on what to classify the coin. An overview of this can be seen in Figure 3.

Fig. 1: Example extracted coin

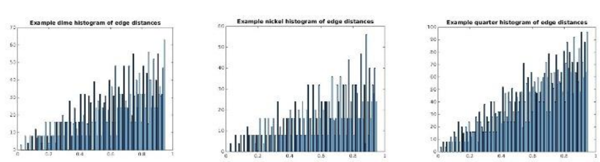

Fig. 2: Example histograms used as features with 140 bins

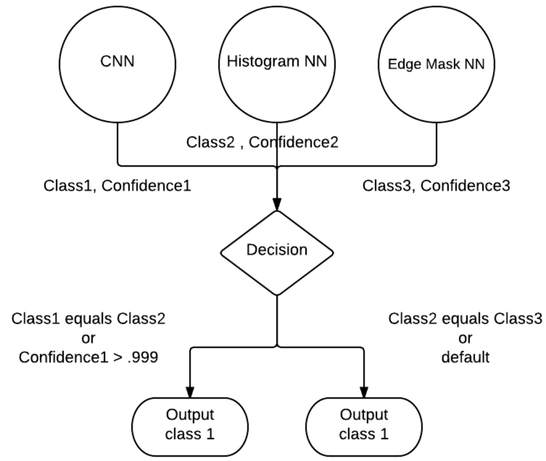

Fig. 3: Description of coin classifier

Once every coin is classified, we look to see if any coins were classified as pennies. If at least one penny is found, we go through a validation phase; otherwise we return the found classifications. Since pennies are easy to spot due to their color and we assume that pictures are taken directly above the coins, we can validate the other classifications based on size. We find the average penny size and compare this size to the other coins. If a coin is smaller than the penny size we ensure that it was classified as a dime and if not label it as such. For coins larger than the penny size, we reclassify them with another classifier built to handle only nickels and quarters. This classifier is another pattern net that uses the same histogram features previously described. We chose to reclassify since this classifier had a higher accuracy when not including dimes. See Table 2 for a comparison of accuracies. After this is complete we have the classes for all the coins in the picture. This entire process can be seen in Figure 4.

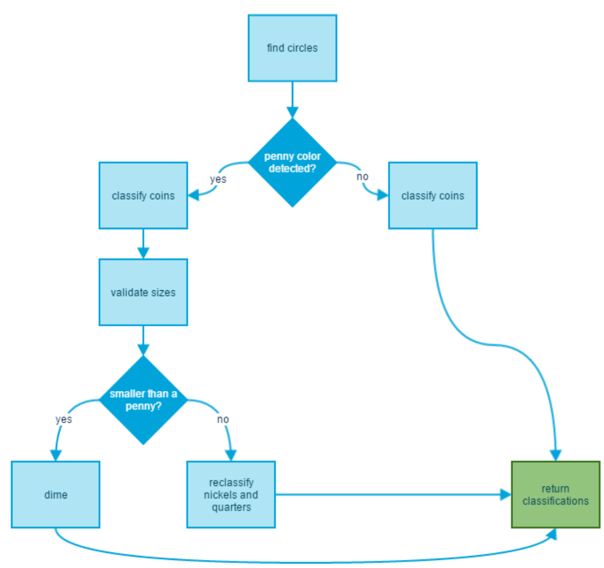

Fig. 4: Process of coin segmentation and classification

3.3 Bills

The bills component of our classification system relies on isolating the bills, rotating them to a horizontal position, and then cropping the upper left hand corner of the bill, resizing it to 28 by 28 pixels, and transforming it into a column vector as the final extracted feature. The bills are isolated by using hue thresholding, selecting for green color, and then performing a small dilation on the connected components obtained.

Min	Max
0.08	0.4

Table 1: Threshold Values for Hue in Bills

The structure element used is a 5 by 5 matrix of ones to smooth out rough edges and obtain a more rectangular shape. For each connected component the centroid is found using MATLAB's regionprops function. The four corners of the component are identified by finding the four minimum and maximum points in the x and y direction of it. The two points representing the top long edge of the bill are found, and the angle to rotate the bill is calculated from their slope.

The new centroid and corners of the component are calculated after rotation, and these measurements are used to obtain the submatrix representing the upper half and leftmost one sixth of the area. The selected region is then sized down to 28 by 28 pixels which is used as a feature for a convolution neural network. This process can be seen in Figure 5.

Fig. 5: Process of rotating and extracting features of bills

4. RESULTS

Our coin training data included 621 images consisting of 109 dimes, 166 nickels, 152 pennies, and 194 quarters. For each image we rotated it 36 times incrementing by 10 degrees each time. This results in 22,356 training images used to train our convolution neural net and our pattern nets.

For bill training data we had 75 images consisting of 9 ones, 13 twos, 11 fives, 13 tens, 14 twenties, and 15 fifties. Similarly rotated it 12 times incrementing by 30 degrees each time. This results in 900 training images used to train our bill convolution neural net.

To test our classifiers we used 188 coin and 35 bill images distributed as follows: 21 pennies, 44 nickels, 39 dimes, 84 quarters, 4 ones, 4 fives, 7 tens, 6 twenties, and 6 fifties. The accuracy for each of our classifiers can be seen in Table 2.

Classifier	Accuracy
Coin Convolution Neural Network	75.78%
Coin Histogram Neural Network	77.84%
Coin Edge Mask Neural Network	61.80%
Coin Combined (above) classifier	85.03%
Nickel/Quarter Histogram Neural Network	94.53%
Bills Convolution Neural Network	75.00%

Table 2: Accuracies of Classifiers.

To visually show our classifications we outlined both bills and coins with color corresponding to a class. The key can be seen in Figure 6. Examples of classified images can be seen in Figure 7.

Fig. 6: Classification Key

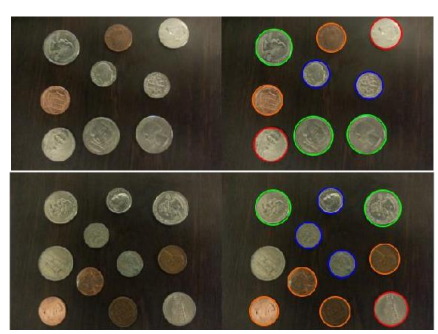

Fig. 7: Example classified images with original on the left and classified on the right

5. CONCLUSION

For identifying coins, we were able to meet our reasonable goal of distinguishing between multiple coins in a single image. The bill detecting portion of our program did not perform as well and only met our minimum goal.

5.1 Failures

Our system failed on complex images that included clumps of coins right next to each other. We believe that this is because there are so many edges close to each other. This can be seen in Figure 8. It is worth noting that MATLAB's built in imfindcircles does not find any circles at all for this image. Warm lighting also throws off our penny classifier because the warm lighting reflects off of dimes, nickels, and quarters making their color resemble that of a penny. This can be seen in Figure 9.

In general, our bill segmentation failed. This is because using hue to find bills was unsuccessful and either did not find the bill or found too much of the image to be useful. An example of this can be seen in Figure 9 where only part of the bill is found. We are confident that if we were able to segment bills correctly using another technique we would have better results.

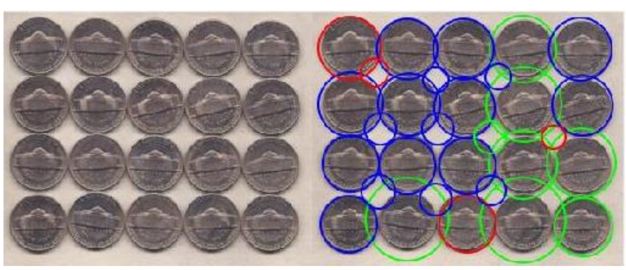

Fig. 8: Example of failed finding circles

Fig. 9: Example of failed bill segmentation and effects of warm lighting

5.2 Successes

We found success in finding coins in our images. Our classifiers were able to distinguish between each coin and classify them well. Our system was very successful with images that contained pennies in them since we were able to validate by size. As seen in Figure 7, after identifying the penny, it was easy for our algorithm to identify the smaller dimes and classify the nickels and quarters. We found that most lighting conditions work other than warm lighting.

5.3 Challenges

Initially, we had trouble classifying coins with high accuracy. To fix this, we investigated three different types of classifiers, including a convolution neural net and two pattern nets. We learned that through combining the outputs of each we could achieve a higher accuracy.

We also had trouble segmenting bills so we could properly extract features from them. This is because we used hue to try and distinguish bills from other objects. We obtained extremely poor results from this and should have used another techniques such as a Hough transform to find rectangles.

5.4 Future work

If given a little more time, we would explore more ways to find bills within the image using a Hough transform for rectangles. We would also explore a more efficient way to perform the Hough transforms. This is because most of the time it takes our system to segment and classify an image is spent finding circles using a Hough transform. We think that we can more efficiently find the local maxima and vote.

For the long term we would like to be able to handle images taken at an angle. We think that we can find and apply a perspective projection to adjust the image so that it can be classified. Another long term goal would be to handle overlapping bills and coins. This would require a robust classification system. Finally, we would like to expand our classifiers to handle more US currency such as fifty cent pieces and foreign currency as well.

6. REFERENCES

[1] Reisert, Marco, Olaf Ronneberger, and Hans Burkhardt. "A Fast and Reliable Coin Recognition System." Pattern Recognition. Berlin: Springer, 2007. 415-24. Print.

[2] Chakraborty, Kishan, Jordan Basumatary, Debasmita Dasgupta, Jagadish Chandra Chandra Kalita, and Subra Mukherjee. "Recent Development in Paper Currency Recognition System." IJRET: International Journal of Research in Engineering and Technology Nov-2013 02.11 (2013): 222-26. IJRET: International Journal of

Research in Engineering and Technology. ESAT Publishing House Pvt. Ltd. Web. 25 Jan. 2015.

[3] Yaojia Wang, Siyuan Lin. Currency recognition system using image processing. Bachelor thesis. Högskolan I Gävle, Gävle, 2010. Digitala Vetenskapliga Arkivet. Web. 24 January 2015.

[4] Van der Maaten, L. J. P. & Boon, P. J. (2006). COIN-O-MATIC: A fast system for reliable coin classification. In M. Nölle & M. Rubik (eds.), Proceedings of the MUSCLE CIS Coin Competition Workshop (p./pp. 7--17), .