Gradient Hough Circle Transform vs. Watershed

A Comparison of 2D Feature Extraction for Coin Detection and MobileNet Classification

Sarrankan Pathmanathan
Institute for Aerospace Studies
University of Toronto

***Abstract*-- The automated extraction of geometric features is a common problem in the field of Computer Vision. One such application is determining the total monetary sum of coins that are provided in a 2D image. A traditional approach applies a 2-stage convolutional neural network such as Faster R-CNN to perform both detection of coins as well as classification based on face value. However, using large multi-network CNNs imposes complexity in in training, when a proper GPU resource is not available. By delegating the coin detection phase to an algorithm that accurately detects 2D hand engineered features, we can employ a lightweight CNN such as MobileNet for classification only. This project aims to determine which between the Gradient Hough Circle Transform and Watershed is the best candidate that will maximize true positive bounding boxes of coins which can be leveraged by MobileNet for a complete coin summation solution that relies less on deep learning and more on the image processing fundamentals of Computer Vision. Detections from both algorithms and classifications from MobileNet are compared to ground truth labeling from 2D images of Canadian coins. Due to its inherent robustness, which mitigates heavy input parameter hyper tuning, Watershed provides the best detection of coins on a uniform surface when coupled with a simple outlier rejection strategy like z-score thresholding. An average true positive rate of 97.07% over varying test cases, while being 38.58% more accurate than Hough as a standalone method.**

***Index Terms*—Feature Extraction, Image Preprocessing, Hough, Watershed, MobileNet**

I. Introduction

Coin detection is a well-known application to which computer vision based object detection and classification techniques can be applied. Given a 2D image with varying coins on a uniform surface, we can use convolutional neural networks to detect coins objects which can then be classified to obtain their monetary worth. Once summed, effectively we have created an application capable of determining the sum of coins present on a uniform surface. A candidate for such a neural network is Faster R-CNN, among other suitable candidates.

Faster R-CNN packages both the detection phase and classification phase into a single pipeline. The region proposal network is used to compute a predefined number of regions (bounding boxes) which may contain coins. Region of Interest Pooling (RoI) extracts those features which would correspond to the relevant objects, classify content in the bounding box, and regresses to produce more refined bounding boxes. Although Faster R-CNN would provide the most accurate prediction of coins, for applications that are memory constrained we must explore lighter CNNs or determine other means of coin (object) detection. A potential solution is to utilize the 2D hand engineered features present in the image and apply algorithms that can extract such features and enclose them in a bounding box, effectively replicating the region proposal network of a traditional CNN without the use of GPU intensive deep learning. Two such algorithms that can be used for this solution is the Gradient Hough Circle Transform, a variation of the traditional Hough Transform, as well as the topographical based watershed segmentation which can be used to detect coins in the foreground. Each of these 2D hand engineered feature-based detectors can be provided an image of coins and return the set of bounding circles that represent coins. Finding circular contours in an image seems trivial at first, however the attribution of background noise and improper lightning in the 2D image, can be a few of the many factors that cause false positive detections. Not only is background noise a factor, but coin faces also attribute noise that can cause false positive detections of the features within them. As such preprocessing is a very important step that must be applied to images for these techniques to mitigate false positives and is explored deeply as part of this project. Additionally, the techniques themselves must have a degree of robustness such that their input parameters do not require a high level of variability for differing images of coins.

This project examines whether the Gradient Hough Circle Transform, or the Watershed feature extractor is the most suitable in meeting the imposed criteria and acting as the most effective replacement of a CNN based region proposal network. Both models are tested against ground truth labeled images of coins to determine the true positive rate, accuracy and precision of each and how the consistent they are in determining features for any image of coins – without the need for intensive hyper tuning of input parameters. MobileNet is also explored as a potential lightweight classifier although the focus remains with the comparison of Hough vs Watershed.

II. Related work

There are several ways Hough and Watershed can be compared in terms of their efficiency in circular feature extraction. Milos and Luis [1] performed their analysis from a mathematical standpoint, drawing conclusions in the weakness

of input parameters to Hough, as a parameter set for an image is often not suitable for another. Their research also aligns with applying a strong set of preprocessing techniques and the dependency this poses on retrieving accurate detections using coin detection as a test method. The Gradient Hough Transform is one of the many existing variations of the Hough Transform. There have been several published works including [2] and [3] which explore other variations of Hough that aim to fix the weaknesses of the traditional model with handling varying radii circles and noise attribution. I employed Gradient Hough Transform discussed in [4] as there exists an implementation in the OpenCV library that provides a multitude of input parameters and includes the Canny Edge Detection for the detection of edges, as part of the function. It would be interesting to determine how well other variations of Hough perform for the case of coin detection against a watershed model. Although there are traditional approaches to watershed as well, and it is most used for segmentation [5], it can be derived into a detector when the foreground is composed solely of candidate objects. Kornilov [6] explores the implementation of watershed provided from open source libraries and describes the scikit-image implementation that is leveraged as part of this project. Although its performance is a bottleneck, as explored by Kornilov, when compared to other implementations for high resolution images containing larger objects, it is well suited for coin segmentation when preprocessing provides enough segmentation of the background to the foreground.

Overall, the detector component of this project could be easily replaced with any algorithm that provides well formed boundaries of the coins in the images. Nonetheless by comparing two fundamentally different models like Hough and Watershed, we can build a spectrum with regards how well they fair as coin detectors.

III. Methodology

A. Overview

Although Gradient Hough Circle Transforms and Watershed are the primary feature extraction models to compare and form the core of this project, I also employ the use of a lightweight MobileNet CNN through Google's Teachable Machine [7] to classify the true positive coin detections. This network is used to determine the total sum of coins presented in a 2D static image.

The implemented coin detector leverages a 3-stage pipeline which includes Image Preprocessing, Candidate Detection, and Classification. Given a set of preprocessing steps, and the MobileNet classifier, the goal is to determine which of Hough and Watershed is best suited as the Candidate Detector. The algorithm that best leverages image preprocessing, handles image variability with the least amount of input hyper tuning, provides the least number of false positive coin detections and feeds the most optimal bounding box enclosure of coins resulting in successful MobileNet classifications is the winner. This section details each of the different steps within the proposed 3 stage pipeline as well as the overall design of the Coin Detector application.

B. Design and Constraints

When first determining how the environment within which the coin detector would function, an Android app was used as the initial design. However due to the constraint of time, I was unable to explore the portability of the desktop application to an Android app. However, the current coin detector can easily be ported to an app given the proper use of the dependent SDKs for android applications. The input it requires is a 2D static image of coins and the output it provides are the enclosing contours of each coin as well as a sentence applied on the image letting the user know of the total sum. I chose to only allow the classification of Canadian coins in the system as they were available to me, and there was a lack of test data with groupings of coins with ground truth labels and summation. Since my project is more aligned with comparing Hough and Watershed Feature Extraction, and less with determining the effectiveness of MobileNet classification, I felt this constraint to be valid. Regardless, if it is was a requirement to classify and detect other coins, one would just need to include the other coins as part of training the network and leave the detector untouched as it is adjusted to detect coins of varying radii with the assumption that there is no coin that is twice as large as the two dollar Canadian coin.

The input image fed into the application requires that coins have minimal overlaps. Although the detector should be able to detect overlapping coins to a fair degree, the classifier requires as much of the faces to be visible to determine their monetary value. The input image must have a uniform background, with a black background providing the best results, however considerable effort was applied to the preprocessing steps to accept non-uniform backgrounds as well.

C. Image Preprocessing

The goal of image preprocessing is to remove as much noise from the 2D input image of coins as possible such that the resultant provides accurate segmentation between the non-relevant background and the relevant foreground where it is assumed that the coins lie. Noise attribution is not only present in the background of the image, but also within the faces of the coins themselves. These features are the primary cause of false positive detections within the Hough and Watershed.

Figure 1 demonstrates the Gradient Hough Transform (red circles) applied to an image with a diverse variety of coin faces, without the use of preprocessing. Watershed requires a binarized image that separates the foreground from the background. However non-optimal segmentation results in false positive detections as illustrated in Figure 1 (green circles). However, compared to Hough it's false positives lie within the features extracted in the faces of the coins which can be easily optimized as opposed to Hough's false positives which is only able to detect 1 or 2 coins that fit an input radius range. Therefore, preprocessing is heavily relied upon to remove these false positives.

When applying preprocessing techniques, they must be resilient enough to handle any degree of noise variability and complement the Hough and Watershed accordingly. As such,

an ample amount of time was dedicated towards determining a subset of techniques that could be leveraged by the candidate detector.

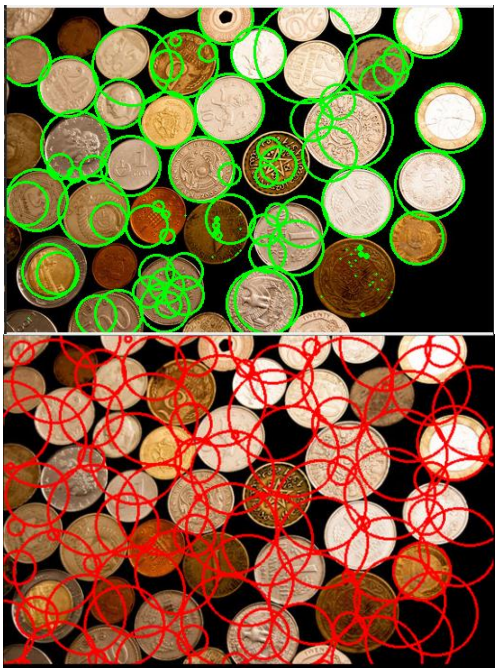

Fig. 1. Green depicts Watershed Detection with prequisite non-optimal preprocessing. Red depicts Gradient Hough Detection with no preprocessing.

1) Grayscaling

Gray scaling is often the first preprocessing step that is applied for candidate detection as color adds to noise attribution. Each pixel (x,y) in a 2D image have a value associated to each channel in RGB (red, green, blue). To grayscale an image, we must adjust the contribution of each channel through the luminosity method [8].

$$newRGB(x,y) = 0.21R_{xy} + 0.72G_{xy} + 0.07B_{xy} \quad (1)$$

This method effectively monochromes the input image by weighting green and blue on opposite spectrums and keeping red in between. The luminosity method applies the weights in (1) however other weights can also be used. For the purpose of this project we only require standard gray scaling. OpenCV provides **cv2.cvtColor** which when called with the flag **cv2.COLOR_BGR2GRAY** for an input image **I** outputs the gray scaled image. Many techniques often require a grayscale image; therefore, it was imperative to begin the preprocessing stage with this. Figure 3(a) provides an example of gray scaling.

2) Pyramid Mean Shift Filtering

An effective method of reducing noise attributed by both the background and the coin faces is to smooth the image to a degree that preserves the circular contours of the coins. To follow suit Pyramid, Mean Shift Filtering, which uses the method of Gaussian Pyramids [9], was leveraged for our needs. At any pixel $\overrightarrow{x_i}$ in an image, the RGB space converges toward a local maximal point adjacent to it. If each pixel is traversed and mean shift analysis of the form (2) is performed using kernel (3) that takes into consideration both a space and color window (h_s and h_r respectively), a new image can be constructed which replaces all points with corresponding local maximal points effectively smoothing the image [10].

$$\vec{m}(\vec{x}) = \frac{\sum_{i=1}^{n} K(\overrightarrow{x_i}; \vec{x}, h)\overrightarrow{x_i}}{\sum_{i=1}^{n} K(\overrightarrow{x_i}; \vec{x}, h)} \quad (2)$$

$$K(\overrightarrow{x_i}; \vec{x}, h) = K_{spatial}(\overrightarrow{x_i^s}; \vec{x}^s, h_s) K_{range}(\overrightarrow{x_i^r}; \vec{x}^r, h_r) \quad (3)$$

The output of the Pyramid Mean Shift Filtering is a filtered posterized image with color gradients and fine-grain textures flattened [10]. Figure 3 (b) provides an example of Pyramid Mean Shift Filtering, which proved to be the most optimal method based on my experimentation with direct Gaussian Blurs and bilateral filters. OpenCV provides **cv2.pyrMeanShiftFiltering** which when called with the thresholds for the special and color windows for an input image **I** outputs the smoothened image.

3) OTSU Thresholding

OTSU is a form of image thresholding where a threshold value T is computed given a distribution of pixels in a gray scaled image. The goal is to divide the image into foreground and background components using this single threshold. OTSU determines this threshold which minimizes the weighted within-class variance (4) of the bimodal distribution present in gray scale images [11]. As such, a requirement of OTSU is that the input image is grayscale. The components of (4) are detailed in [11].

$$\theta_w^2 = q1(t){\sigma_1}^2(t) + q2(t){\sigma_2}^2(t) \quad (4)$$

When applied to an image of coins with a uniform black background, OTSU can segment the foreground coins from the background. This is illustrated in Figure 3 (c). OpenCV provides **cv2.threshold** to which can separate foreground to white and background to blackthrough a binarization using **cv2.THRESH_BINARY** to set pixels above the threshold to white and below the threshold to black. Binarization is compounded with **cv2.THRESH_OTSU** to determine the appropriate threshold value.

4) Color Space Segmentation

Color space segmentation is a method that I explored but did not apply as part of the final solution due to lighting in the image causing color ambiguity. This technique attempts to segment the coins in the image based on a known range of coin colors. For Canadian coins, this includes copper, silver, and gold. By sampling the coin colors of several images, we can create a 3D color space of possible colors that should be segmented from an image of coins. Figure 2 provides such a graph over N samples. If for example a penny were to exist in

an image, its color space is most likely to lie within the boundaries of copper illustrated in Figure 2. Similarly, a mask for silver and gold can also be retrieved.

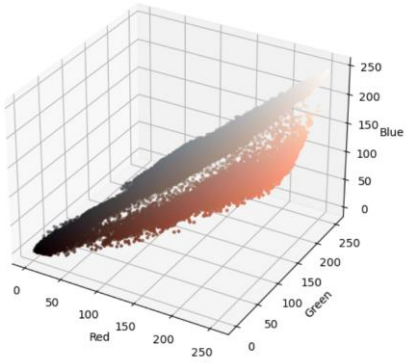

Fig. 2. Sampling of possible coin colors for silver and copper is performed. However due to lighting conditions, color is often skewed and varied.

This technique is promising if lighting conditions are kept consistent. It requires a large array of color samples to perform effective segmentation of coins. Figure 3 (d) provides an example of this segmentation. However, some parts of the coin remain as it is not successfully captured within the spectrum built in Figure 2.

By applying all preprocessing techniques in the sequence, we augment the true positive rate of detection by Hough and Watershed while decreasing false positives as well. An ablation study is performed as part of the experimental results section to determine the efficacy of each technique as they are compounded onto the image.

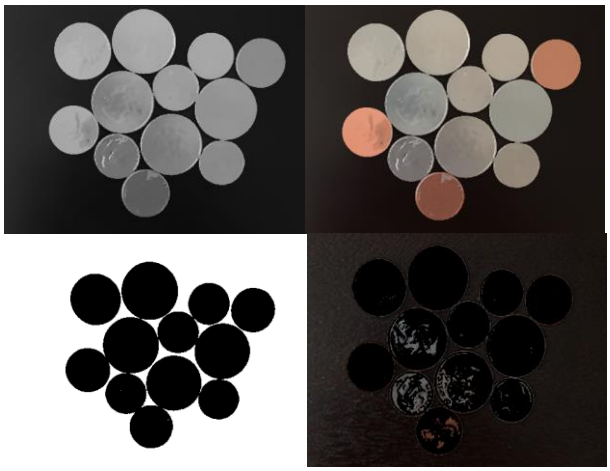

Fig. 3. Top left (a) Grayscaling with pyramid mean shift filtering . **Top Right (b)** Pyramid Mean Shift Filtering. **Bottom Left (c)** OTSU Thresholding. **Bottom Right (d)** Colour Space Segmentation

D. Gradient Hough Circle Transform

The traditional Hough Circle Transform determines circles of the form (x, y, r) where (x, y) is the circle center and r is the radius.

Suppose a coin of known radius r is present in a 700 x 700 pixel image against a uniform black background with another coin of smaller radius overlapping it. Canny Edge Detection is first performed to determine the circular contours of the coin in the image. For each pixel that defines the "edges" of the coin, a circle of the known radius R is drawn. Since (x,y) are the unknowns in this situation, we are effectively searching a 2D space for these coordinates which is often named the 2D Hough search space [12]. Each pixel that forms the circle can be thought of as a vote. To keep track of votes, Hough employs an accumulator matrix of the same dimension as the search space. As circles are drawn at each edge pixel, several votes converge with the global maxima (the pixel with the most votes) being defined as the circle center and the local maxima as the circle edge. Since R is known, many false detections can be discarded, with the resulting coin center and radius retrieved. By drawing a circle at the provided coordinate of center R and enclosing this within a bounding box, we have effectively used Hough to detect a coin. Figure 4a retrieved from [12] illustrates the Canny Edge Detection phase as well as the completed tally of the accumulator which defines the coin center.

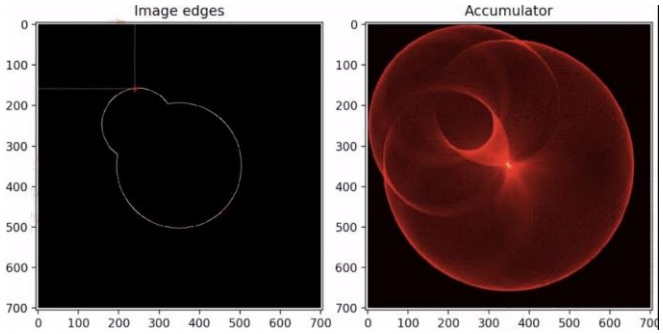

Fig.4 a. An example of a two overlapping coins in an image and Hough detecting coins of known radius R within the 2D Hough search space.

Fig. 4b. **Below** demonstrates varying the HoughCircles parameters to obtain optimal detections as previous parameter sets do not work with this image.

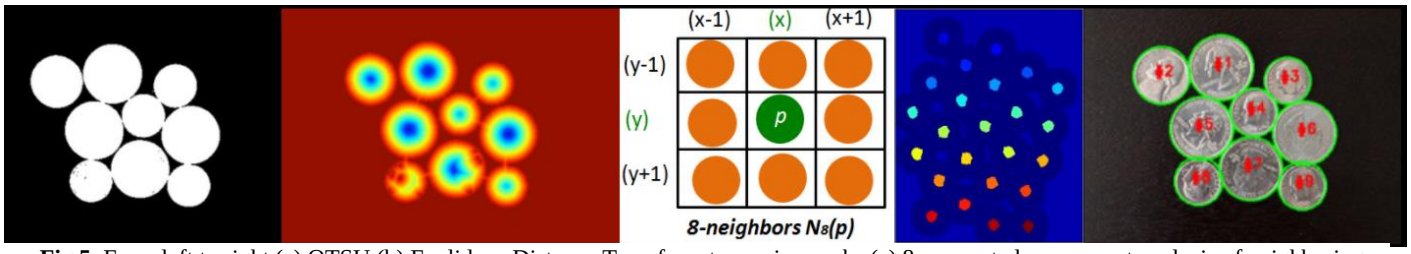

Fig.5. From left to right (a) OTSU (b) Euclidean Distance Transform to receive peaks (c) 8-connected component analysis of neighboring intensities (d) watershed markers to which we retrieve coin center and radius

Although Hough is extremely effective when the radius that is meant to be detected is known, this is often not the case in real-world applications, including for the coin detector. Images of coins can be taken from any angle as well as from near or far. A penny which has a certain radius from the viewpoint of one image, can have any other radius from the viewpoint of another. As R is now unknown, a 3D Hough search space is now required to determine candidate coins. This is computationally expensive and forms one of the many weaknesses of the Hough Circle Transform – input parameter hyper tuning.

One variation of the Hough transform that was explored in this project, and OpenCV provides an implementation for, is the Gradient Hough Circle Transform. The traditional model is still respected, however only a sector of the circle depending on the edge direction provided by intensity gradient calculation needs to be incremented in the accumulator matrix [4]. This reduces the search space greatly which improves run-time. OpenCV provides **cv2.HoughCircles** which includes the **cv2.HOUGH_GRADIENT** flag [13]. This function takes several input parameters which provide thresholding for detected circles including the minimum distance between circle centers, the range of radii to look for, the gradient value used to handle Canny Edge Detection as well as the threshold limit of votes within the accumulator matrix.

The bottleneck with Hough remains with its variability in input parameters as seen in Figure 4b. It is rare that the same parameter set P can be applied to all input images, unless the image is specifically constrained to respect these parameters. Therefore, in addition to an optimal parameter set K for image processing, outlier rejection is also performed and is detailed in section F.

E. Watershed

Compared to Hough, Watershed is a far more robust algorithm as it composed of a 4 stage pipeline of computations in order to retrieve watershed markers which translate to labels of coin centre (x,y) of radius r.

Watershed follows a topological approach by flooding the valleys of different makers until they meet the markers that signify an exit of a foreground [14]. In our case these valleys are the coins present in an image. Suppose an image with several coins is provided where one or more coins could be overlapping against a uniform black background. OTSU thresholding (Figure 5(a)) is first done to segment these foreground coins from the background through a distinction of black for the background and white for the foreground. The effectively creates candidate white valleys in the black background.

The Euclidean Distance Transform (Figure 5(b)) is then used to determine the closest zero (background) from each of the foreground pixels which creates a distance map. Peaks in this distance map signify possible coin centers, to which the inverse provides defined valleys in the image. An advantage of leveraging this transform is that for overlapping coins, pseudo-boundaries can be created as peaks from each coin is captured.

Using these peaks, pseudo-boundaries and the distance map, we can apply a 8-connectivity connected components analysis (Figure 5(c)), to determine pixels that are part of the same component map, creating watershed markers on the pixel centers of the each coin. Finally, we utilize the markers and the mask to retrieve contours, and create minimum enclosing circles which can then be translated to bounding boxes (Figure 5(d)). This complete baseline method, including each method in the pipeline for watershed, was studied from [15].

This algorithm is not reliant on providing a range of radii to determine coins which makes it effective when images vary. Input parameter hyper tuning is not as prevalent with watershed in this case. However, the performance of watershed lies solely in the effectiveness of the preprocessing step in providing clear segmentation of the foreground and background before binarization and OTSU are applied. If coin faces are not smoothened optimally through pyramid mean shift filtering, then OTSU may create sub valleys through the faces of the coins (the smaller green circles in Figure 1) incurring false positive detections. However, such valleys can usually be rejected since their radii are often small enough that they cannot be considered a coin.

By compounding outlier rejection and parameter optimization, we can augment the standalone implementations of both Hough and Watershed.

F. Outlier Rejection and Parameter Optimization

Although the goal of this project is to determine how effective Gradient Hough Transform and Watershed are as standalone methods, the degree to which outlier rejection and parameter optimization is required in each is also a good measure in determining their effectiveness. That is, the less outliers (true negatives) are detected for a given technique to retrieve a

result set of accurate coin detections. the better the standalone algorithms are.

Parameter optimization is required heavily on the Hough Circle transform function as even a pixel variability in its input parameters (including the candidate circle radius and minimum distance between centers) can lead to false negatives. Additionally, the parameters of the preprocessing techniques may not be as effective for varying image conditions. Thus the coin detection problem can be formulated using the following statement: *There exists an image preprocessor parameter set (P) and a feature extraction parameter set (F) such that (P,F) provides the global optima of the number of true positives (coins) in the image.*

Determining (P, F) dynamically is non-trivial as there have been several works including [16] in formulating hyper tuning techniques. The simplest method I employed for Hough is to sample both a range of minimum distances between coins as well as the possible range of radii to detect in Hough. A Z-score threshold (6) is used to determine the mean and standard deviation of detected radii and reject those circles that are not within a provided threshold.

$$Z = \frac{x-\mu}{\sigma} \qquad (6)$$

Much of the dependency on true positive detections lie in parameter P, the preprocessing set. However, it is much easier to tune these techniques as their space for variability is smaller than that of the radius thresholds of Hough, as often a set P can be applied to several images.

Other rejection techniques that were compounded were removing false positives that were directly within the detection of a true positive as often these circles were too small to be that of coins and were a result of coin faces not being properly smoothed. By applying these techniques, we could ensure an optimal set of bounding boxes be provided to the MobileNet classifier.

G. MobileNet CNN

Since candidate detection is delegated to methods like Hough and Watershed that do not require deep learning, we can employ a lightweight convolutional neural network that is solely responsible for the classification of coins. This provides benefits in terms of memory usage, training when a proper GPU resource is not available, and portability. For this project, I explored the use of MobileNet which is based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks [17]

Google provides an implementation of a MobileNet CNN called Teachable Machine [7]. It uses transfer learning by leveraging a MobileNet base model trained on ImageNet weights and allowing the user to configure the output layer with their own training for object classification. Their model is comparable to the MobileNetV2 model in Figure 6 which includes a standard relu activation layer, a dropout layer to prevent overfitting, as well as a softmax activation output layer often used for multi-class prediction networks [18].

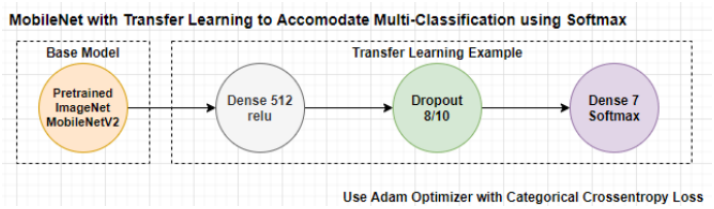

Fig. 6. A MobileNetV2 CNN which is comparable to the network used by Google's Teachable Machine. Due to a lack of proper GPU resources, Teachable Machine was leveraged for classification.

Such a model could be trained and optimized on Google ML compute or Google Colab under a GPU or TPU runtime, however I did not have proper GPU resources to train this model, therefore I chose to use the MobileNet weights provided by Teachable Machine which met the requirements of classifying Canadian coins. The winner of Hough or Watershed should be able to provide candidate boxes that can be used by the MobileNet classifier. Experimental Method The method that was used to compare Hough and Watershed was to create a test set of images that included Canadian coins of varying values on a somewhat uniform black background as lighting conditions is a variable factor in each image. Pictures were either retrieved online or were taken using a Samsung Galaxy S10 camera. Ground truth labels and sums were retrieved manually in each case as I was unable to find existing test sets meant for such an application. Implementation of the application was done in Python using the OpenCV library of functions for preprocessing as well as candidate detection through Hough and Watershed. A single pass of each algorithm was done on each image in the test set where one version includes the parameter optimization and outlier rejection techniques, and one does not and is the standalone implementation of each of Hough and Watershed.

Candidate detections were then passed to the MobileNet classifier to determine the quality of bounding boxes of coins retrieved from Hough and Watershed.

IV. Experimental Results

A. MobileNet Training

To train the MobileNet network provided by Google's Teachable Machine, the World Coins dataset from Kaggle world coins was used [19]. In order to simplify the training of the neural network, as there was no dependency in the classifying all coins to determine how well Hough and Watershed performed, I reduced the training, validation and test set to encompass only 6 classes of Canadian Coins including 1 cent (penny), 5 cent (nickel), 10 cent (dime), 25 cent (quarter), 1 dollar (loonie), and 2 dollar (toonie). The network was trained for 200 epochs with 16 batches per epoch. Given the reduced dataset with 360 total images, the validation and test set made 14% of the set with the remaining 86% being used for training. This subset of the world coins dataset A detailed list of parameters for training is provided in Table I. Loss and accuracy plateau quickly as shown in Figure 7 due to the reduced subset of inputs.

The greatest challenge with coin classification is distinguishing between coins when the queen's head is used. For real-world scenarios where dimes, nickels and quarters are often mistaken for the same thing, this poses a limitation as shown in Table II as well as the confusion matrix in Figure 8. Regardless, our concern is with the efficacy of the candidate coin detector derived from Hough and Watershed in being able to provide refined bounding boxes for classification. MobileNet is still capable of providing over 80% accuracy for all classes.

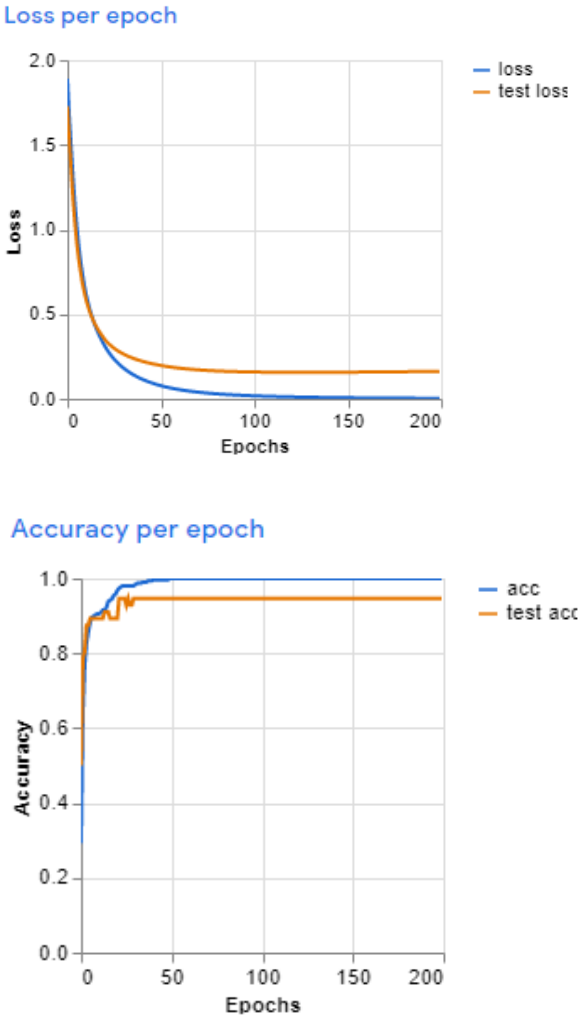

Fig. 7. Loss and Accuracy per epoch. The main outliers are the often-indistinguishable nickel, dime, and quarter due to similar color.

Table I. Network Training Hyperparameters

Hyperparameter	Value
Epochs	200
%validation data	14
Batches/epoch	16
# of input points	360
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Categorical Cross entropy

Accuracy per class

CLASS	ACCURACY	# SAMPLES
1	1.00	8
5	0.88	8
10	0.88	8
25	0.88	8
100	1.00	8
200	1.00	8

Table II. Sampled accuracy for each class.

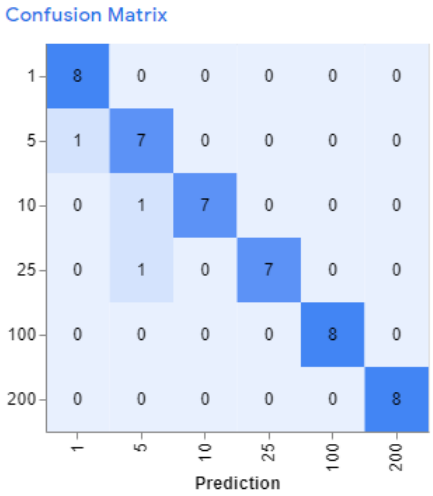

Fig. 8. The confusion matrix for test set detections. The network has trouble with the similar nickel, dime, and quarter, especially when the queen's head is used. However other classes are detected accurately.

B. Comparison of Hough and Watershed

1) Preprocessing Ablation Study and Precision of Detection

An ablation study is provided in Table III to determine how well Hough and Watershed perform when parts of the preprocessing is removed from the overall application. This to determine the efficacy of the algorithms as standalone implementations as well as their true dependencies to these preprocessors in a complete solution.

In the case of Watershed, OTSU threshold and binarization of the foreground and background is a prerequisite of the algorithm. For both Watershed and Hough, gray scaling is also a prerequisite. Regardless, how well this technique is applied in both Hough and Watershed when coupled with gray scaling and pyramid mean shift filtering provides insight in their inherent abilities to handle noise.

True positive detection is a major criterion for an object detector. Given our test set of images with Canadian coins, whichever of Hough and Watershed maximizes the true positive rate and provides accurate detections is a clear candidate replacement of a region proposal network in a CNN implementation of the coin detector.

Features	Accuracy	Precision	Recall
Watershed Complete	93.92%	93.76%	97.07%
Remove Pyramid Filter	57.24%	47.65%	75.73%
Gradient Hough Complete	86.97%	86.40%	86.83%
Remove OTSU	64.78%	42.51%	89.58%
Remove OTSU, Pyramid Filter	47.59%	6.92%	28.58%

Table III. Ablation study to determine the effectiveness of preprocessing as well as the overall implementation of Gradient Hough and Watershed,

2) Parameter Optimization and Outlier Rejection

Table IV. provides insights into how well the compounded parameter optimization and outlier rejection techniques augment the overall models. When parameter optimization is used, several false positives that are detected become true negatives. When not used, all true negatives become false positives. Although both algorithms can detect coins in the image, and bound them, the false positive detections cause an additional number of incorrect bounding boxes which negatively impact the classification step, as they are not part of the true summation of coins present in the image. Therefore, parameter optimization and outlier rejection aim to discard as many false positives as possible. Doing so in Watershed increases precision and accuracy by 14.02% and 15.29% respectively.

Features	Accuracy	Precision	Recall
Watershed	78.63%	79.74%	97.07%
Watershed with Optimization	93.92%	93.76%	97.07%
Gradient Hough	40.05%	43.07%	86.83%
Gradient Hough with Optimization	86.97%	86.40%	86.83%

Table IV. Analysis of Gradient Hough and Watershed with and without parameter optimization and outlier rejection.

3) MobileNet Summation of Coins

As we can see in Figure N, the weakness in the MobileNet model lied in classifying the very similar nickel, dime, and quarter. The quality of the pictures taken by the Samsung Galaxy S10 camera were also factor, as sometimes coins face was not as defined. Nevertheless, for images where coins are defined with minimal noise caused by lighting and bounding boxes contain only the candidate coin and no overlapping ones (done by reducing the proposed radius to the coin face), the MobileNet is able to produce accurate results for most cases. A complete example of the application of Watershed detection, the winner of the comparison, and MobileNet classification, is shown if Figure 9.

Fig. 9. Combined Watershed Detection with MobileNet classification.

C. Discussion and Final Analysis

When comparing the standalone methods without any form of parameter optimization in Table N, Watershed is on average 38.58% more accurate especially in cases where coins overlap, and radii varies. The fundamental issue with Hough is that its parameter set P is often not applicable to more than 1 image in the test cases. For example, if an image A is provided as input where the radius of all coins is R, Hough requires that the parameter set P is within a range of R, denoted as range(R), to mitigate false positives. However, once this range is set, and image B has coins of R > range(R), Hough is unable to retrieve any true positives. To avoid this issue, the initial Hough radius range can be set to large enough range that true positives are always detected. However, this greatly reduced precision and accuracy as several false positive detections are obtained. We can see that parameter optimization is primarily used to reduce false positive detections and the z-score threshold technique coupled with sampling ranges based on the obtain minimum and maximum radius of previous iterations almost doubles precision and accuracy.

Parameter optimization does aid Watershed in removing false positives detected from non-optimal segmentation of foreground coins for the background, however due to its anti-dependency on the coin radius present in the image, it work very well as is. Watershed is highly dependent on preprocessing techniques, as it requires OTSU and gray scaling to function. Pyramid mean shift filtering can aid the OTSU step in retrieving better binarization between the coins and background as in our tests it increased the accuracy, precision and recall of watershed by 36.68%, 46.11%, 21.34% respectively. Therefore, the better preprocessing is done to the input image, which is easier to tune then unpredictable radii in an image, the better watershed performs overall. Therefore, due to its robust qualities, its ability to mitigate false positive

detections compared to Hough, and being able to noise efficiently, it is the clear winner as the candidate detector to a MobileNet coin Classifier.

V. Conclusion

This report presents a comparison between the Gradient Hough Circle Transform and Watershed object detectors to obtain bounding boxes of coins by analyzing its 2D hand engineered features. By utilizing these detectors, a lightweight MobileNet classifier can be leveraged to provide better memory efficiency and portability of the system. Several preprocessing steps were explored to augment the capabilities of Hough and Watershed, while coupling parameter optimization and outlier rejection to further discard false positive detection - a major bottleneck of Hough. Hough requires heavy optimization to discard the large number of false positives retrieved. Its dependency on an input range of radii is often a weakness as radii vary per image. Nonetheless, Watershed proved to be the better of the two as it was 38.58% more accurate as a standalone method than Hough as was the better candidate to accompany Mobile Net and provide optimal bounding boxes for the classification and summation of coins.

VI. References

[1] T. Milos and M.Luis “Hough and Watershed Transforms Algorithms Brief Review,” Mechanics and Mechatronics Department, Universidad Nacional de Colombia, 2017

[2] Yadav, Virendra & Batham, Saumya & Acharya, Anuja & Paul, Rahul. (2014). Approach to accurate circle detection: Circular Hough Transform and Local Maxima concept. 2014 International Conference on Electronics and Communication Systems, ICECS 2014. 1-5. 10.1109/ECS.2014.6892577.

[3] Roushdy, Mohamed. (2007). Detecting Coins with Different Radii based on Hough Transform in Noisy and Deformed Image. ICGST International Journal of Graphics, Vision, and Image Processing GVIP. 7. 25-29.

[4] Petkovic and Loncaric. An Extension to Hough Tranform Based on Gradient Orientation. Proceedings of the Croation Computer Vision Workshop, Year 3.

[5] Beucher, S. Watershed, hierarchical segmentation and waterfall algorithm. In Mathematical Morphology and its Applications to Image Processing, J. Serra and P. Soille, Eds. Kluwer Acad. Publ., Dordrecht, 1994, pp. 69-76.

[6] Kornilov, A.S.; Safonov, I.V. An Overview of Watershed Algorithm Implementations in Open Source Libraries. J. Imaging 2018, 4, 123.

[7] “Google Teachable Machine,” Google. [Online]. Available: https://teachablemachine.withgoogle.com/.

[8] Kanan C, Cottrell GW (2012) Color-to-Grayscale: Does the Method Matter in Image Recognition?. PLOS ONE 7(1): e29740

[9] “OpenCV - Image Pyramids,” Tutorialspoint. [Online]. Available: https://www.tutorialspoint.com/opencv/opencv_image_pyramids.htm.

[10] Seiya.kumada, Mean Shift Filtering ～Practice by OpenCV～, 01-Jan-1970.[Online].Available:http://seiya-kumada.blogspot.com/2013/05/mean-shift-filtering-practice-by-opencv.html.

[11] Liu W, Shi H, He X, Pan S, Ye Z, Wang Y. An application of optimized Otsu multi-threshold segmentation based on fireworks algorithm in cement SEM image. Journal of Algorithms & Computational Technology. January 2019. doi:10.1177/1748301818797025

[12] tkorting, “How Circle Hough Transform works,” YouTube, 25-Feb-2020. [Online]. Available: https://www.youtube.com/watch?v=Ltqt24SQQoI.

[13] “Hough Circle Transform,” OpenCV. [Online]. Available: https://docs.opencv.org/master/d3/de5/tutorial_js_houghcircles.html.

[14] Beucher, S., Lantuejoul, C.: Use of Watersheds in Contour Detection. In: International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France (September 1979)

[15] A. Rosebrock, “Watershed OpenCV,” PyImageSearch, 18-Apr-2020. [Online].Available:https://www.pyimagesearch.com/2015/11/02/watershed-opencv/.

[16] Hassanein , Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab. A Survey on Hough Transform, Theory, Techniques and Applications. Informatics Department, Electronics Research Institute,El-Dokki, Giza,12622, Egypt.

[17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, & Hartwig Adam. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[18] Sebastian Bruch, Xuanhui Wang, Mike Bendersky, & Marc Najork (2019). An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance. In Proceedings of the 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR 2019) (pp. 75-78).

[19] Santori, P. (2018). World Coins: A collection of coin mages from 32 different currencies, Version 1. from https://www.kaggle.com/wanderdust/coin-images.