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 detect 2D hand engineered features, we can employ a lightweight MobileNet CNN just for classification. This project aims to determine which of 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, with an average true positive rate of []%.**

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

I. Introduction

Coin detection is one of many way computer vision based object detection and classification 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 those of us who require a model to apply to a memory constrained system, we must either employ a lightweight CNN 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. Two such algorithms that can be used for our 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 foreground coins . 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 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 and accuracy 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.

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 these 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, it can be derived into a detector when the foreground is composed solely of candidate objects. Kornilov [5] 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 compared to other implementations for high resolution images containing larger objects, it is well suited for coin segmentation when preprocessing provides enough preprocessed 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, as opposed to aligned algorithms, we can build a spectrum with regards how well they fair against a CNN based detector which is sure to provide accurate results.

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 to determine the quality of the bounding boxes that are perceived as true positive coin detections. This network provides the total sum of coins presented in a 2D static image and is the use case of the application.

Coin Detection using 2D hand engineered features can be described as 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, can handle 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. Application 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 OpenCV SDK 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.

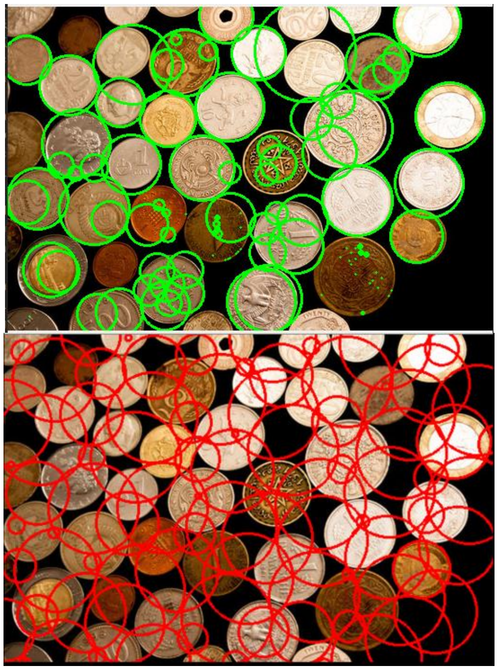
The input image fed into the application requires that coins have minimal overlapping. Although the detector should be able to detect overlapping coins to a fair degree, the classifier requires as much of the faces to be visible to determine their monetary value. The input image must have a uniform background, 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 detectors as often, they are mistaken for coins.

Figure N demonstrates Gradient Hough Transform (red circles) applied to an image with a diverse variety of coin faces, without the use of preprocessing. Although watershed requires preprocessing to function properly, the figure demonstrates a poor amount of preprocessing being applied. 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.

When applying preprocessing we need to apply techniques that are resilient enough to handle any degree of noise variability and complements the Hough and Watershed accordingly. If preprocessing results in a bottleneck for coin detection, it can greatly affect the results of detection and introduce bias when evaluating Hough and Watershed. As such, an ample amount of time was dedicated towards determining a subset of techniques that could be leveraged by the candidate detector.



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 [].

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

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. However, 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 N(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 [N], was leveraged for our needs. At any pixel $\vec{x_i}$ 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 [N]

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

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

The output of the Pyramid Mean Shift Filtering is a filtered posterized image with color gradients and fine-grain textures flattened [N]. Figure N (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

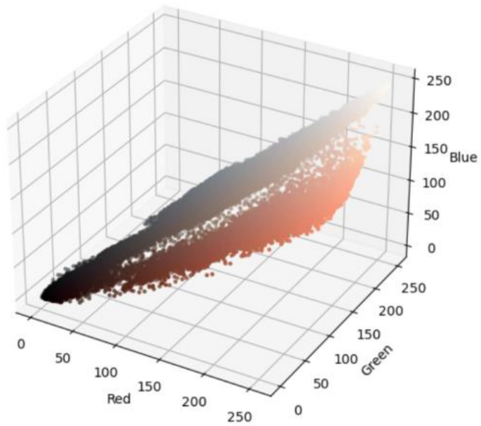
OTSU is a form of Image Thresholding where a threshold value T is computed given a distribution of pixels in a gray scaled image. The goal is to divide the image into foreground and background components using this single threshold. OTSU determines this threshold which minimizes the weighted within-class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

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

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 N (c). OpenCV provides **cv2.threshold** to which can separate foreground to white and background to black while through 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 N 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 N. Similarly, a mask for silver and gold can also be retrieved.



This technique is promising if lighting conditions are kept consistent. It requires a large array of color samples to perform effective segmentation of coins.

By applying all preprocessing techniques in the sequence, we augment the true positive rate of detection by Hough and Watershed. 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.

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. Such a circle in 2D space can be described by (5).

In the application of coin detection, suppose a coin of known radius r is present in a 512 x 512 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 [N]. 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 N (a) retrieved from [N] illustrates the Canny Edge Detection phase as well as the completed tally of the accumulator which defines the coin center.

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 (Figure [N](b)] 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.

There have been several attempts to improve the Hough model through variations of the traditional algorithm including [N] and [N]. However, one 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[N]. This reduces the search space greatly which improves run-time. OpenCV provides **cv2.HoughCircles** which includes the **cv2.HOUGH_GRADIENT** flag [N]. 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 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 5 stage pipeline of intuitive 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 [N]. 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 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 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, to determine pixels that are part of the same component map, creating markers on the pixel centers of the each coin and retrieving the radius of each component. Finally, we utilize the radius and centres to create minimum enclosing circles which can then be translated to bounding boxes.

As we can see, 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 through OTSU. If coin faces are not smoothened optimally through pyramid mean shift filtering, then OTSU may create sub valleys through the faces of the coins 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. OpenCV provides a function for each of the stages required as part of 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 robust the outlier rejection strategy is for a given technique to retrieve a result set of accurate coin detections the better the standalone algorithms.

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. The parameters of the preprocessing techniques may not be as effective for varying image. 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 (5).

Determining (P, F) dynamically is non-trivial as there have been several works including [N] 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 detected radius. Outliers are then defined as those who are not within a defined threshold. When a minimum number of rejections is determined, it is assumed that all current detections are those of coins. Much of the dependency on true positive detections lie in parameter P. However, it is much easier to tune these techniques as their variability space is smaller than that of the radius thresholds of Hough. Since Watershed also relies on P, and does not have inherent parameters to optimize, it works best standalone.

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

Although the focus of this project is not to analyze the effectiveness of MobileNet as a coin classier, a coin detection solution that sums coins requires the classification of coins by value. If the

IV. Experimental results

Appendixes, if needed, appear before the acknowledgment.

V. Conclusion

The following is an example of an acknowledgment. (Please note that financial support should be acknowledged in the unnumbered footnote on the title page.)

The authors gratefully acknowledge the contributions of I. X. Austan, A. H. Burgmeyer, C. J. Essel, and S. H. Gold for their work on the original version of this document.

VI. References

References are important to the reader; therefore, each citation must be complete and correct. There is no editorial check on references; therefore, an incomplete or wrong reference will be published unless caught by a reviewer or discusser and will detract from the authority and value of the paper. References should be readily available publications.

List only one reference per reference number. If a reference is available from two sources, each should be listed as a separate reference. Give all authors' names; do not use *et al.*

Samples of the correct formats for various types of references are given below.

Basic format for books:

[1] https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/
[2] https://www.irjet.net/archives/V4/i11/IRJET-V4I11103.pdf

by 3.18 centimeters (1.25 inches) high. The head and shoulders should be centered, and the photo should be flush with the left margin. The space required for the biographies and photos is included in the eight-page limit.