Computer Vision Project  
CSE 483

Spring 2023

Prepared By:

1. Tasneem Hisham Sayed 19P4152
2. Elsaeed Ahmed Elsaeed 19P1087
3. Ahmed Sameh Mourad 19P5861

Table of Contents

[1. PHASE 1: 3](#_Toc135241402)

[1.1 Pipeline 3](#_Toc135241403)

[1.2 Pipeline Steps 3](#_Toc135241404)

[1.3 Output Screenshots: 5](#_Toc135241405)

[2. PHASE 2 7](#_Toc135241406)

# 1. PHASE 1:

## 1.1 Pipeline

In phase 1, we’ve worked on the first format of the “train” dataset only, which has 33,402 images in total.

All the functions used in this phase are provided by the “OpenCV” python library.

## 1.2 Pipeline Steps

1. Read all images from the “train” dataset, using the “imread” function. The “imread” function takes the path to an image file and reads it into a NumPy array.
2. Convert each color image to its corresponding grayscale format. Grayscale images only have one channel, as opposed to RGB images which have three channels for red, green, and blue. This makes them easier to process and analyze for certain computer vision tasks. The "cvtColor" function is used here, which converts an image from one color space to another. In this case, we would use it to convert each color image to grayscale.
3. Dilating and blurring the image: Dilating using the dilate function with a 7x7 kernel, then applying a median filter using the medianBlur function with a kernel size of 21. This is done to remove any small, unwanted details and to create a background image.
4. Calculating the difference image then normalizing it: Calculating the difference between the background image using the absdiff function. It subtracts the result from 255 to invert the image, making the objects in the foreground white. Then normalizes the result using the normalize function with the NORM\_MINMAX normalization type and a dtype of CV\_8UC1.
5. Apply histogram equalization to enhance the contrast or sharpness of the grayscale image. Histogram equalization is a method used to stretch the brightness levels of an image so that the entire range of brightness is utilized. This can improve the contrast of the image, making it easier to identify features. The “equalizeHist” function in OpenCV is used to perform histogram equalization on the image.
6. Apply median filtering, which is a type of noise reduction technique that works by replacing each pixel value in an image with the median value of its neighboring pixels. This helps to remove random noise from the image while preserving edges and other important features. In this code, OpenCV's fastNlMeansDenoising function is used here to apply median filtering to the image.
7. Apply a binary threshold on the equalized image, creating a binary image with white regions representing areas of the image with high intensity values, and black regions representing areas with low intensity values. The “threshold” function in OpenCV is used to create a binary image, where all pixel values greater than 200 are set to 255, and all others are set to 0.
8. Find the contours of the output threshold image. Contours are the boundaries of objects in an image. They can be used to identify and locate objects in an image. The “findContours” function in OpenCV is used to find all the contours in an image. This function takes the output of the previous step, which is the histogram equalized grayscale image, and applies a threshold to it to convert it into a binary image. Then it identifies all the contours in the binary image.
9. Draw the contours found in the previous step onto the copy of the original image. This will help visualize where the contours are in the image. The “drawContours” function in OpenCV is used to draw contours on an image. This function takes the copy to the original image, the contours found in the previous step, the index of the contour to draw (or -1 to draw all contours), and the color and thickness of the contour.
10. Draw bounding boxes around each of the found contours, that are between a minimum & a maximum threshold that is proportionate to the size of each image. Bounding boxes are rectangles that enclose an object in an image. They can be used to crop or isolate specific objects in an image. The “rectangle” function in OpenCV is used to draw bounding boxes around the contours.
11. The final step is to calculate the accuracy of the localization by using the “intersection over union” method, by measuring the overlap between the real & predicted bounding boxes of each image. The “Intersection over Union” (IoU) is a method used to evaluate the performance of the object detection algorithms. The second method used is to calculate the accuracy uses the IoU but with a small modification that uses a black image and intersecting the found boundaries with the real boundaries.

---------------------------------------------------------------------------------------

## 1.3 Output Screenshots:

The Blue bounding boxes represent the predicted drawn boxes.

The Red bounding boxes represent the real boxes, read from the .mat file.

The Green outliers represent the contours drawn on the processed image.

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A picture containing screenshot, graphics, design

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing screenshot, text, graphics, design

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, multimedia software, line

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

# 2. PHASE 2

GitHub Repo Link: <https://github.com/Ahmed-Sameh-MM/Computer-Vision-Project>