Aim: To Detecting and Recognizing Objects/Text

Objective: Object Detection and recognition techniques HOG descriptor The Scale issues The location issue Non-maximum (or non-maxima) suppression vector machine people detection

Theory:

Object detection and recognition Techniques

Object recognition is a computer vision technique used to identify, locate, and classify objects in digital images or real-life scenarios. It is an applied artificial intelligence approach that repurposes a computer as an object detector so it can scan an image or video from the real world. It understands the object's features and interprets its purpose just like humans do. Object recognition combines four techniques: image recognition, object localization, object detection, and image segmentation. Object recognition decodes the features and predicts the category or class of image through a classifier, for example, supervised machine learning models like Support Vector Machine (SVM), Adaboost, Boosting, or Decision Tree. Object recognition algorithms are coded in Darknet, an open-source neural network framework written in C, Cuda, or Python.

HOG descriptors

HOG is a feature descriptor, so it belongs to the same family of algorithms as scale invariant feature transform (SIFT), speeded-up robust features (SURF), and Oriented FAST and rotated BRIEF (ORB). . Like other feature descriptors, HOG is capable of delivering the type of information that is vital for feature matching, as well as for object detection and recognition.Most commonly, HOG is used for object detection. The algorithm – and, in particular, its use as a people detector – was popularized by Navneet Dalal and Bill Triggs in

their paper Histograms of Oriented Gradients for Human Detection (INRIA, 2005). HOG's internal mechanism is really clever; an image is divided into cells and a set of gradients is calculated for each cell. Each gradient describes the change in pixel intensities in a given direction. Together, these gradients form a histogram representation of the cell.

The Location issue

Like other detectors, a HOG-based detector needs to cope with variations in objects' location and scale. The need to search in various locations is addressed by moving a fixed size sliding window across an image. The need to search at various scales is addressed by scaling the image to various sizes, forming a so-called image pyramid Suppose we are using a sliding window to perform people detection on an image. We slide our window in small steps, just a few pixels at a time, so we expect that it will frame any given person multiple times. Assuming that overlapping detections are indeed one person, we do not want to report multiple locations but, rather, only one location that we believe to be correct. In other words, even if a detection at a given location has a good confidence score, we might reject it if an overlapping detection has a better confidence score; thus, from a set of overlapping detections, we would choose the one with the best confidence score.

Non-maximum(or Non-maxima)Suppression

A typical implementation of NMS takes the following approach:

1. Construct an image pyramid.

2. Scan each level of the pyramid with the sliding window approach, for object detection. For each window that yields a positive detection (beyond a certain arbitrary confidence threshold), convert the window back to the original image's scale. Add the window and its confidence score to a list of positive detections.

3. Sort the list of positive detections by order of descending confidence score so that the best detections come first in the list.

4. For each window, W, in the list of positive detections, remove all subsequent windows that significantly overlap with W.

We are left with a list of positive detections that satisfy the criterion of NMS. Besides NMS, another way to filter the positive detections is to eliminate any subwindows. When we speak of a subwindow (or subregion), we mean a window (or region in an image) that is entirely contained inside another window (or region). To check for subwindows, we simply need to

compare the corner coordinates of various window rectangles. We will take this simple approach in our first practical example, in the Detecting people with HOG descriptors section. Optionally, NMS and suppression of subwindows can be combined

Support vector machines

Given labeled training data, an SVM learns to classify the same kind of data by finding an optimal hyperplane, which, in plain English, is the plane that divides differently labeled data by the largest possible margin
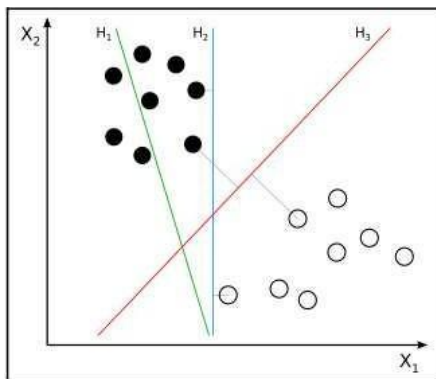
Hyperplane H1 (shown as a green line) does not divide the two classes (the black dots versus the white dots). Hyperplanes H2 (shown as a blue line) and H3 (shown as a red line) both divide the classes; however, only hyperplane H3 divides the classes by a maximal margin.

Code

```
!pip install easyocr
import cv2
import easyocr
import matplotlib.pyplot as plt
import numpy as np
image_path = '/content/images.jpg'
img = cv2.imread(image_path)
# instance text detector
reader = easyocr.Reader(['en'], gpu=False)
# detect text on image
```

```
text_ = reader.readtext(img)

threshold = 0.25

# draw bbox and text

for t_, t in enumerate(text_): print(t)

bbox, text, score = t

if score > threshold:

    cv2.rectangle(img, bbox[0], bbox[2], (0, 255, 0), 5)

    cv2.putText(img, text, bbox[0], cv2.FONT_HERSHEY_COMPLEX, 0.65, (255, 0, 0), 2)

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

plt.show()
```

Output
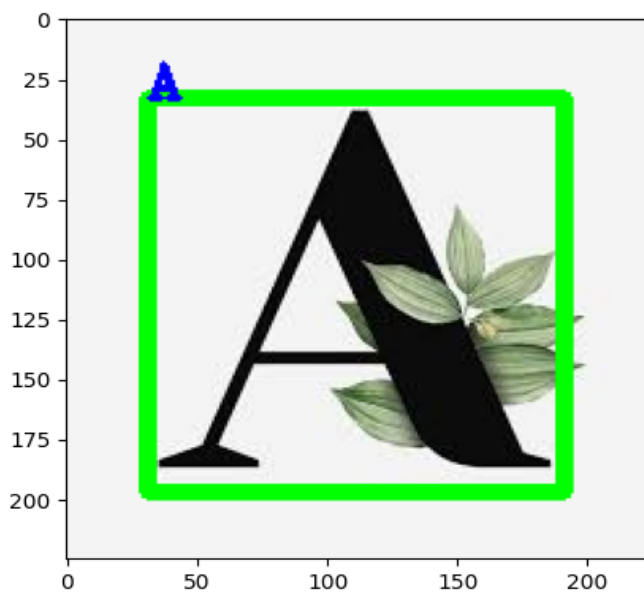
Conclusion:

Our primary objective was to employ object recognition methods, specifically Histograms of Oriented Gradients (HOG) and text recognition, to identify and interpret objects and text present in an image. HOG descriptors were employed for the purpose of object detection, while we utilized the Non-Maximum Suppression (NMS) algorithm to eliminate redundant detections, ensuring that only the most reliable results were retained. For the text recognition task, we harnessed the capabilities of the EasyOCR library to locate and decipher text within the image. This experiment proved effective in accurately identifying and marking text regions with bounding boxes, thus offering a valuable tool for the extraction of textual information from images.