How to Build Social Distancing Tool

Poonam Yadav
Computer Science and Engineering, Dronacharya College of Engineering, Farukhnagar, Haryana

***Abstract*- Social Distancing – the term that has taken the planet by storm and is transforming the way we live. Social distancing has become a mantra round the world, transcending languages and cultures. This way of living has been forced upon us by the fastest growing pandemic the planet has ever seen – COVID-19. As per the planet Health Organization (WHO), COVID-19 has thus far infected almost 4 million people and claimed over 230K lives globally. Around 213 countries are affected thus far by the deadly virus. The biggest explanation for concern is that COVID-19 spreads from person to person through contact or if you're within close proximity of an infected person. Given how densely populated some areas are, this has been quite challenge. So this got me thinking – i would like to create a tool which will potentially detect where everyone is in real-time, and return a bounding box that turns red if the space between two people is dangerously close. This will be employed by governments to research the movement of individuals and alert them if things turn serious.**

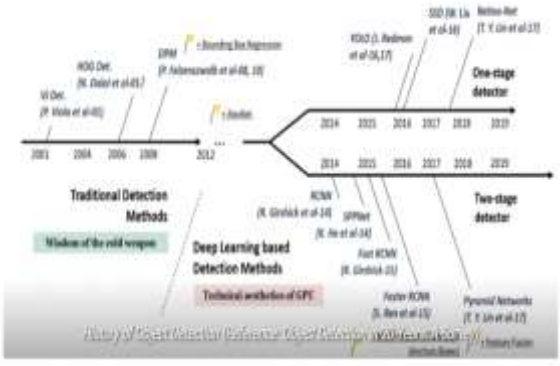
1. INTRODUCTION

I can vividly recall my initial days learning computer vision. I often got confused between these two terms – Image Classification and Object Detection. I used both of those terms interchangeably assuming the thought behind them was similar. And, as a result, I kept getting confused between deep learning projects
Object Detection and Image Classification are very fashionable tasks in computer vision. They need a good range of applications in defense, healthcare, sports, and therefore the space industry.
The fundamental difference between these two tasks is that image classification identifies an object in a picture whereas object detection identifies the thing also as its location in a picture. Here's a classic example to know this difference:

Well, then how is Object Tracking different from Object Detection?
Object Tracking and Object Detection are similar in terms of functionality. These two tasks involve identifying the thing and its location. But, the sole difference between them is that the sort of data that you simply are using. Object Detection deals with images whereas Object Tracking deals with videos.
As a video may be a collection of fast-moving frames, Object Tracking identifies an object and its location from each and each frame of a video.

2. THOERY

"Evolution of State-of-the-Art (SOTA) for Object Detection"
Object Detection is one among the foremost challenging problems in computer vision. Having said that, there has been an immense improvement over the past 20 years during this field. we will broadly divide this into two generations – before and after deep learning:



"Sliding Window for Object Detection"
The simplest approach to create an Object Detection model is thru a window approach. Because the name suggests, a picture is split into regions of a specific size then every region is assessed into the respective classes. The regions are often overlapping and ranging in size also. It all depends on the way you would like to formulate the matter.

'Model Workflow'

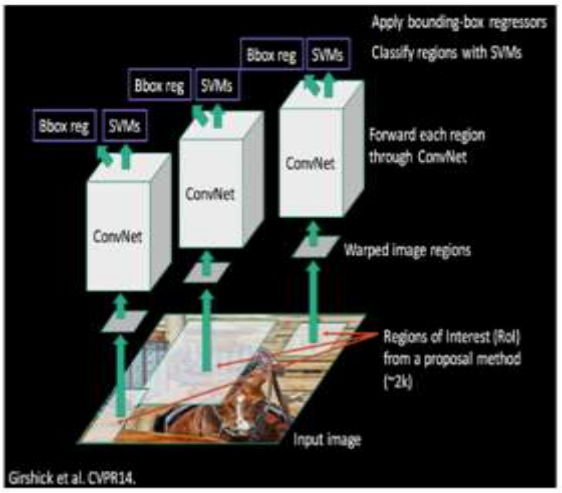
1 Consider a picture
2 Divide a picture into regions (assume 10 * 10 region)
3 for every region:

1 Pass a neighborhood to Convolutional Neural Network (CNN)
2 Extract features from CNN
3 Pass features to a classifier & regressor

This method is basically simple and efficient. But it's a time-consuming process because it considers the large number of regions for classification. Now, we'll see how we will reduce the amount of regions for classification within the next approach.

"R-CNN for Object Detection"

So how can we make this a non-time consuming task? This will be brought down by discarding the regions that aren't likely to contain the thing. This process of extracting the regions that are likely to contain the thing is understood as Region Proposals. Region proposals have a better probability of containing an object. Many Region Proposal algorithms are proposed to pick a neighborhood of Interest (ROI). a number of the favored ones are objectness, selective search, category-independent object proposals, etc.



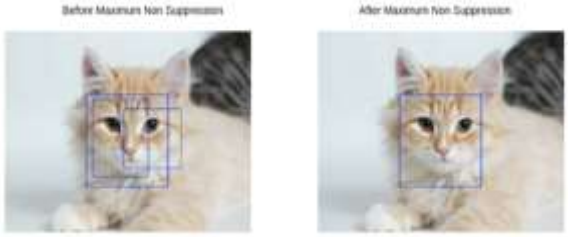
So, R-CNN was proposed with a thought of using the outside region proposal algorithm. R-CNN stands for Region-based Convolutional Neural Network. It uses one among the external region proposal algorithms to pick the region of interest (ROI).

'Model Workflow'

1 Consider a picture
2 Select ROI using exterior region proposal algorithm
3 for every region:

1 Pass a neighborhood to CNN
2 Extract features from CNN
3 Pass features to a classifier & regressor

The predicted regions are often overlapping and ranging in size also. So, Maximum Non Suppression is employed to ignore the bounding boxes depending upon the Intersection Over Union (IOU) score:



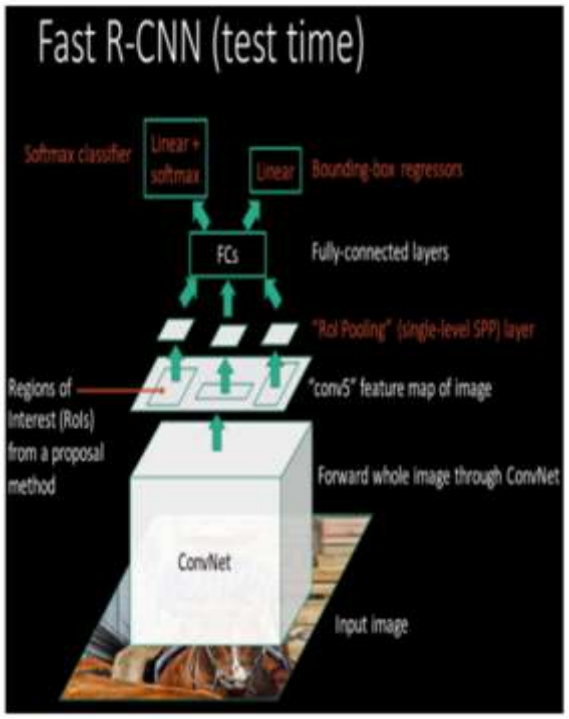
Certainly, R-CNN's architecture was the State of the Art (SOTA) at the time of the proposal. But it consumes nearly 50 seconds for each test image during inference due to the amount of forward passes to a CNN for feature extraction. As you'll observe under the model workflow, every region proposal is passed to a CNN for feature extraction.

For example, if a picture has 2000 regions of proposals, then the amount of forward passes to the CNN is around 2000. This inevitably led to a different model architecture referred to as Fast R-CNN.

"Fast R-CNN for Object Detection"

In order to scale back the inference speed, a small change within the R-CNN workflow was made and proposed, referred to as Fast R-CNN. The modification was wiped out the feature extraction of region proposals. In R-CNN, feature extraction takes place for every region proposal whereas, in Fast R-CNN, feature extraction takes place just one occasion for an ingenious image. Then the relevant ROI features are chosen supported the situation of the region proposals. These region proposals are constructed before passing a picture to CNN.

The input to CNN is that the actual image with none ROI:



'Model Workflow'

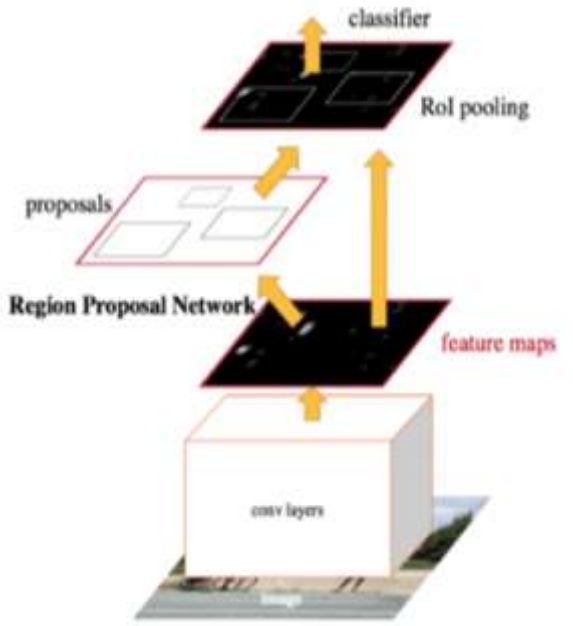
1 Consider a picture
2 Select Regions of Interest (ROI) using exterior region proposal algorithm
3 Pass a picture to the CNN
4 Extract the features of a picture
5 Choose relevant ROI features using the situation of ROI
6 for every ROI feature, pass features to a classifier & regressor

During inference, Fast R-CNN consumes nearly 2 seconds for every test image and is about 25 times faster than R-CNN. the rationale being the change within the feature extraction of ROI. for instance , if a picture features a 2000 region of proposals, then the amount of forward passes to the CNN is around 1.

"Faster R-CNN for Object Detection"

Faster R-CNN replaces the outside region proposal algorithm with a neighborhood Proposal Network (RPN). RPN learns to propose the region of interests which successively saves tons of your time and computation as compared to a quick R-CNN.

Faster R-CNN = Fast R-CNN + RPN



'Model Workflow'

1 Consider a picture
2 Pass a picture to CNN
3 Extract the features of a picture
4 Select ROI features using Region Proposal Network (RPN)
5 for every ROI feature, pass features to a classifier & regressor

Faster R-CNN takes on the brink of 0.2 seconds for each test image during inference and is about 250 times faster than Fast R-CNN.

"Social Distancing Tool Use Case of Object Detection & Tracking"

Social Distancing is that the only thanks to prevent the spread of COVID-19 immediately.

"Understanding Detectron 2"

Detectron 2 is an open-source library for object detection and segmentation created by the Facebook AI Research team, popularly referred to as FAIR. Detectron 2 implements state of the art architectures like Faster R CNN, Mask R CNN, and RetinaNet for solving different computer vision tasks, such as:

1 Object Detection
2 Instance Segmentation
3 Keypoint Detection
4 Panoptic Segmentation

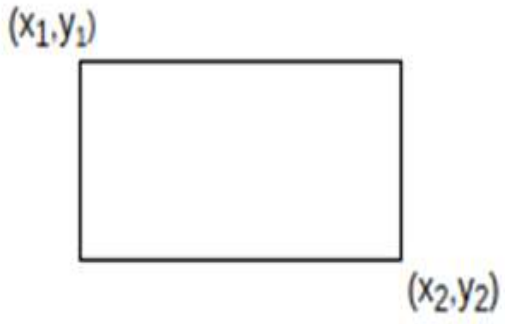
The baseline models of Faster R-CNN and Mask R-CNN are available with 3 different backbone combinations.

3. ALGORITHM

1 Install Dependencies.
2 Import Libraries.
3 Reading a video: Read a video and save frames to a folder.
4 Check the frame rate of a video.
5 Read a picture and pass it to the model for predictions.
6 Output of the model is Objects and Locations, since its an object detection model. we will use Visualizer to draw the predictions on the image.

As you'll see here, multiple objects are present in a picture, sort of a person, bicycle, and so on. We are well on our thanks to building the social distancing detection tool!

7 Next, understand the objects present in a picture.
8 Have a look at the bounding boxes of every object.
9 As different objects are present in a picture, identify classes and bounding boxes associated with only the people.
10 Understand the format of the bounding box.



11 Draw a bounding box for one among the people.

Our ultimate goal is to compute the space between two people in a picture . Once we all know the bounding box for every person, we will easily compute the space between any two people. But the challenge here is to pick the proper coordinate for representing an individual as a bounding box is within the sort of a rectangle. I have chosen rock bottom center of a rectangle for representing everyone to live the space accurately and also this measure is unaffected by the peak of an individual .

12 Define a function that returns rock bottom center of each bounding box.
13 Compute rock bottom center for each bounding box and draw the points on the image.

14 Define a function to compute the Euclidean distance between every two points in a picture.
15 Compute the space between every pair of points.
16 Define a function that returns the closest people supported the given proximity distance. Here, proximity distance refers to the minimum distance between two people

17 Set the edge for the proximity distance. Let's find the people that are within the proximity distance.

18 Define a function to vary the colour of the closest people to red.

19 Perform similar steps on each and each frame of the video.

20 After identifying the closest people in each frame, convert the frames back to a video.

4. LIMITATIONS AND FUTURE WORK

The projection of the camera also matters tons while computing the space between the objects in a picture In our case, I even have not taken under consideration the projection of the camera since the impact of the camera's projection on the estimated distance is minimum. However, the universal approach is to convert a video into a top view or birds' eye view then compute the space between two objects in a picture. This process is understood as camera calibration.

5. CONCLUSION

The handiest manner to prevent the unfold of covid-19 is social distancing. keeping a secure distance from each other is the final manner to save you the unfold of this disorder (as a minimum till a vaccine is discovered).

REFRENCES

[1] https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md

[2] https://courses.analyticsvidhya.com/courses/fundamentals-of-deep-learning

[3] https://courses.analyticsvidhya.com/courses/computer-vision-using-deep-learning-version

[4] https://analyticsindiamag.com/landing-ais-ai-enabled-social-distancing-detection-tool/

[5] https://github.com/zishansami102/CNN-from-Scratch