VISUAL RECOGNITION

MINI PROJECT

By:
IMT2018002 Abhigna Banda
IMT2018026 Gayathri Venkatesh
IMT2018046 Mundla Aarthi Sree
IMT2018047 Nachiappan S K

Date: 21-03-2021

Problem Statement

To build a real-time vision based automatic door entry system that allows a person to enter only if he/she is wearing a mask.

Since the system has to be real time, there are 3 modules to this problem:

- Identifying the person - human detection
- Identifying region of interest of the face - face detection
- Identifying the presence of a mask

Module 1 - Human Detection

For this problem, we would like to draw a bounding box around every human within each frame of the video. At a given point in time, there can be any number of people walking in, and hence, the number of bounding boxes is not fixed. This is why a regular CNN will not work - because the length of the output layer cannot be determined.

Another approach is to consider several regions of interest in the image and use a CNN in each of them to detect the object. This again, leads to a problem since each object in the frame might have different orientations and aspect ratios. Algorithms like R-CNN and YOLO solves the problem of having to choose a large number of regions and reduces the overall time complexity of the model.

Algorithms:

1. **R-CNN**

   R-CNN stands for Regions with CNN. It uses an algorithm called Selective Search to extract only 2000 regions from every image. Each of these regions are acted upon by a CNN that acts as a feature. Then SVM is used for object classification.

Disadvantages:

- High training time because it has to create 2000 regions for every image.
- Cannot be made real time as it takes a long time to process and classify each image.
- The 2000 chosen regions may not be the best candidates in the image, leading to incorrect classification.

2. **Faster R-CNN**

This algorithm has an approach similar to R-CNN, but instead of feeding 2000 region proposals to the CNN, we directly input the entire image. From this image, the CNN generates a convolutional feature map which allows us to find the region proposals. They are warped into a square, reshaped and fed into a fully connected layer which classifies the object.

Since the convolutional neural network is applied only once per image instead of 2000, it is much faster than R-CNN and can be used for real-time applications.

3. **YOLO**

YOLO stands for You Look Only Once. It aims to eliminate the time-consuming selective search algorithm that finds multiple region proposals within a single image.

Unlike R-CNN which uses regions to localize objects, YOLO uses a single convolutional neural network to look at the entire image, draw the bounding boxes and predict the class probabilities for these boxes.

The algorithm takes in an image and draws a grid, and within each square we take m bounding boxes. For each bounding box the CNN outputs a class probability. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image.

Advantages:

- Much faster than R-CNN or Faster R-CNN as it does not rely on the selective search algorithm for region proposals.
- Better than Faster R-CNN for real-time purposes.
- It has a simpler architecture than faster R-CNN.

Disadvantages:

- Does not work well with small objects / objects that are very close to each other.
- Does not generalize well when objects in the image show rare aspects of ratio.

Results:

Algorithm	Accuracy	Training time	Testing time
R-CNN	57.03%	> 3 hours	~13 seconds
Faster R-CNN	88.12%	44.6 mins	1.2 seconds
YOLO	90.26%	19.0 mins	< 0.5 seconds

Comparison of YOLO and Faster R-CNN

1. Speed (training and testing):
   YOLO gave us faster results than Faster R-CNN.

2. Detection under low lighting conditions:
   Both YOLO and Faster R-CNN worked equally well under low lighting.

3. Detection of small objects:
   YOLO detected smaller objects / objects that are far away better than Faster R-CNN.

4. Detection when multiple objects are present within a small space:
   Faster R-CNN gave us a slightly better accuracy in detecting crowded objects.

Overall, YOLO gives a better accuracy with a shorter testing time. Hence, we have gone ahead with **YOLO** instead of Faster R-CNN.

Module 2 - Face Detection

For face detection, there are several algorithms like Viola Jones, Kanade-Lucas-Tomasi (KLT) and Histogram of Oriented Gradients (HoG).

For this problem, we have experimented with two such algorithms and noted down their accuracies in detecting faces real-time.

Algorithms:

1. **Viola-Jones Algorithm**

Viola-Jones is an object-recognition framework that allows the detection of image features in real-time. The Viola-Jones Object Detection Framework uses Haar-like features, AdaBoost algorithm, and Cascade Classifier to create a system for object detection that is fast and accurate. It acts on a grayscale image and marks out the face in the original image.

- Haar-like features are represented as adjacent black and white rectangles. It produces a single value by taking the sum of the intensities of the light regions and subtracting that by the sum of the intensities of dark regions. Haar-like features let us extract features such as edges and lines that we can use to detect a face.

- AdaBoost (Adaptive Boosting) is an algorithm for selecting the best subset of features among all available features. It outputs a prediction function / strong classifier, which is a linear combination of weak classifiers (also known as best features).

- The Cascade classifier has multiple stages used to perform fast and accurate object detection. Each stage has a strong classifier produced by the above

AdaBoost algorithm. The input is evaluated sequentially, and if at any stage, the output of the classifier is negative, the input is discarded immediately. Only if the classifier output is positive, it moves to the next stage. This allows us to immediately discard ‘non-faces’ and spend more time on images that are face-like, which helps us reduce the time complexity.

- During the training phase, we use the input dataset to construct integral images, from which we can extract the Haar-like features. The Cascade Classifier is then created by using a modified AdaBoost Algorithm on that training data.

- For testing, the algorithm uses a sliding window approach where a window of a particular size is slid across the entire image. The window size and the shifting step size are to be decided by us. Each subwindow passes through the cascade classifier, and a face is detected only if it passes through all the stages.

Advantages:

- Fast detection
- Requires lesser training data than other models.
- The input image does not need to be scaled because of the scale-invariant detector.

Disadvantages:

- Viola-Jones works well with frontal faces. But its accuracy reduces while detecting faces tilted or turned in another direction. For real-time detection, it is unlikely that all people will look directly at the camera while walking.
- The algorithm is very sensitive to lighting. The face needs to be well illuminated to be detected accurately.

2. KLT Algorithm (Kanade-Lucas-Tomasi)

Kanade-Lucas-Tomasi (KLT) algorithm is used for tracking human faces continuously in a video frame. This method is accomplished by finding the parameters that allow the reduction in dissimilarity measurements between feature points that are related to the original translational model.

- First, the displacement of the tracked points is calculated from one frame to another. From this, the movement of the head is computed.

- For the initial step of face detection, we use the Viola-Jones algorithm along with a classification model.

- To track the face over time, we use the Kanade-Lucas-Tomasi (KLT) algorithm. We cannot repeatedly use Viola-Jones and the cascade classifier as it would blow up the time complexity. Moreover, it would fail when the face gets tilted or the lighting shows variation. Therefore, we use KLT to track the detected face across all frames.

Results:

Face direction	Viola-Jones	Kanade-Lucas-Tomasi
Frontal	95.34%	90.18%
Sideways	89.05%	82.67%
Up / Down	79.89%	80.24%

On average, Viola-Jones gives an accuracy of **88.09%** while KLT gives **84.36%**. Therefore, we have gone ahead with the **Viola-Jones algorithm** for face detection.

Module 3 - Mask Detection

For the above problem we have used convolutional neural networks.

For testing, we use the cascade classifier created above and obtain the video from the webcam. The region of interest is identified and data preprocessing is done as described below.

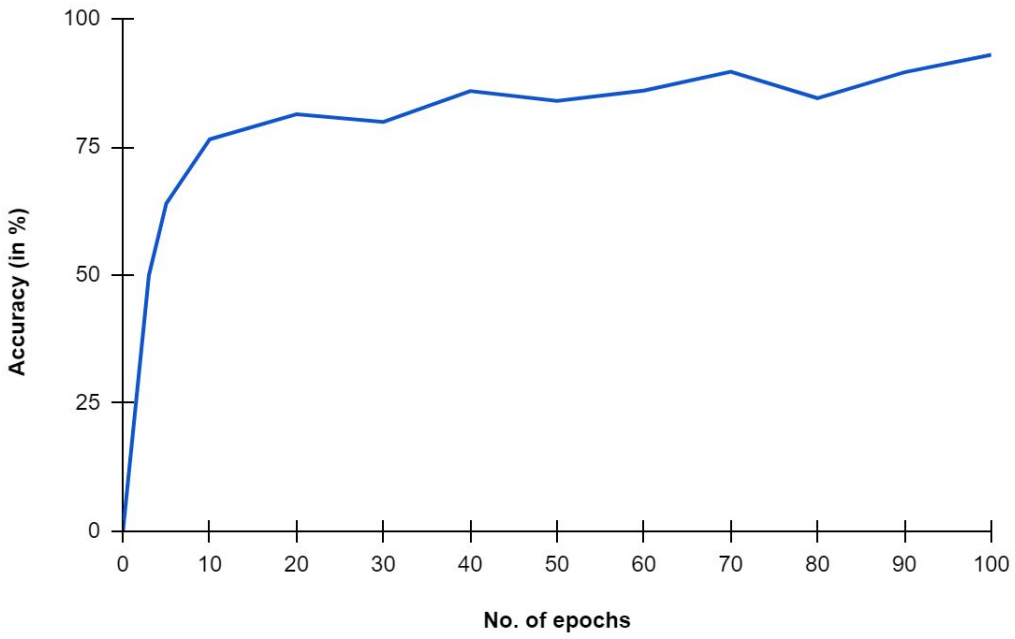
Steps:

- Data preprocessing
    - Convert the image to grayscale and resize it to 100x100.
    - Construct a label set. The model outputs two labels - 0 (with mask) and 1 (without mask).
    - Normalize the image by dividing it by 255. This will ensure that the pixel value is within the range 0 to 1.

- Architecture of the Neural Network
    - The CNN consists of two convolutional layers - the first one with 200 3x3 kernels and the second, consisting of 100 3x3 kernels.
    - Subsequently, we have added layers for ReLU activation and max pooling to reduce the size of the data and determine the output of the network.
    - After this we add a flattening layer that converts the data into a 1-dimensional array that serves as input to the fully connected layer.
    - We have added a dropout of 50% to prevent overfitting of the model.
    - This is followed by two fully connected layers. The first one has 50 neurons and the second one has 2 - representing the two categories (with and without mask).
    - The Adam Optimizer has been used as the learning algorithm since it gives a better accuracy then SGD.

Algorithm	Accuracy
Stochastic Gradient Descent	86.19%
Adam Optimizer	93.28%

Plot of accuracy vs no. of epochs for Adam Optimizer



Testing Accuracy of the CNN	0.9328
Testing Loss	0.0317

How real-time and accurate is your overall system and the individual modules?

The model works fairly well real-time, with an average testing time of **0.1 seconds** and an accuracy of **93.28%**.

Drawbacks:

- Since most of the dataset consists of people wearing surgical masks, the detection accuracy on masks with colours / patterns is slightly weak.
- Accuracy slightly reduces when the person is facing sideways or looking down.
- Accuracy depends on lighting and occlusion. In our case, a person walking from a dark room is always flagged as ‘no mask’, even when he is wearing a mask. As he walks closer, the mask gets detected.