***Documentation of the various Deep Learning modes used***

1)YOLO

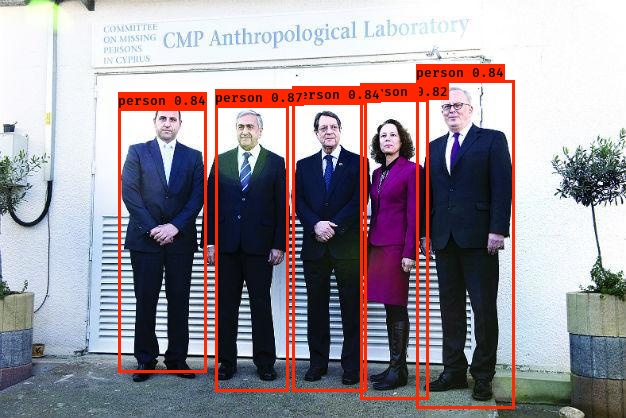
Yolo(You Only Look Once) is a state-of-the-art, real-time object detection system that can detect over 9000 object categories.

Yolo is better and faster than other state of the art object detection systems. Due to its speed of object detection it can be used in tasks requiring better results at greater speed.

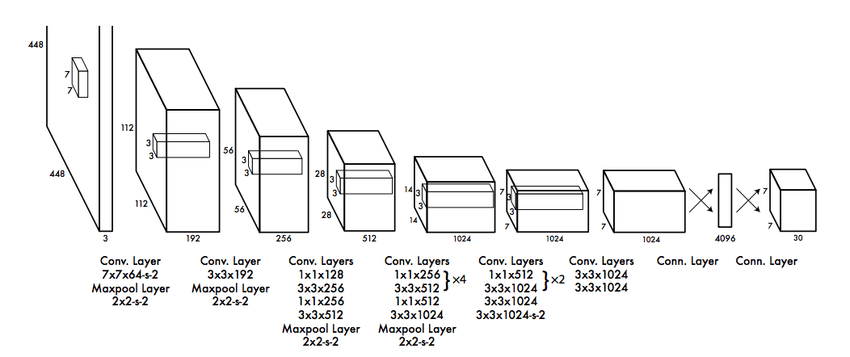
The output of the algorithm is a depth feature map. Each pixel in the depth feature map consists of the probabilities of all the classes along with the bounding box coordinates. Thus each pixel represents a particular region of the input image and gives the probabilities of different classes at that region in the image. Non maximum suppression is used to suppress all the class probabilities that are below a certain threshold. After this the output is further processed using Intersection Over Union technique that discards all the bounding boxes at a particular location in the image that have maximum overlap with the maximum probability bounding box.

YOLO in action:





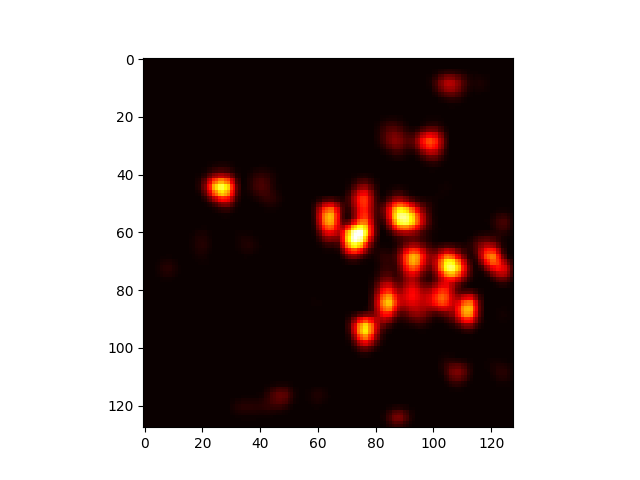
YOLO ARCHITECTURE:



2)HeatMaps

Heatmaps are used to predict the crowd density maps from images of crowded places. The ground truth crowd density maps in training images is produced by applying gaussian kernels at positions where humans are located. The convolutional neural network is a 20 layer deep convolutional neural network involving Batch Norm layers, Max Pooling layers.

HeatMaps in Action:

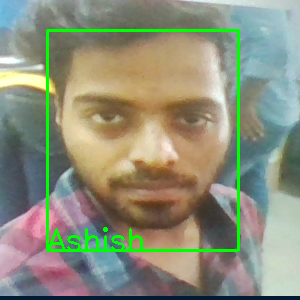


3) Facenet and MTCNN:

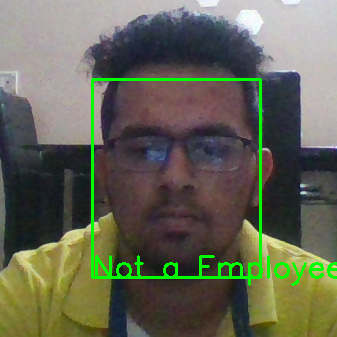
Facenet is the state of the art deep convolution neural network(CNN) based on the Inception Resnet Architecture. It is the state of the art technique for facial recognition. One single instance of data for training the facenet contains three images one known as anochor ,one as positive and one as negative. The facenet model is so trained that it can produce embeddings of any face. Its loss function include making the mean squared error of embeddings of anchor and positive image/face(ie the same person) and the mean squared error of anchor and negative image/face to be large(ie different persons) to be large. This is done with the help of a method known as Triplet loss and using a single CNN. Training examples are so choosen that the anchor and false image are hard to distinguish. During training Embeddings of all different faces is saved. So during testing , given a new face the face embeddings are first calculated using the same CNN. Then embeddings are classified based on the machine learning model which was trained on embeddings from training dataset

MTCNN stands for multitask CNN it is usually used for face detection and proper alignment of face. The frames from input are first given to MTCNN and then to facenet.

Facenet In Action: The Facenet was trained on 4 classes Ashish ,Viplav, Swapnil, Rajnessh. The third image from down is the image it was’t trained on:







4)ChatBot:

The chatbot presented in this project combines traditional rule-based chatbot models with advanced deep learning models.

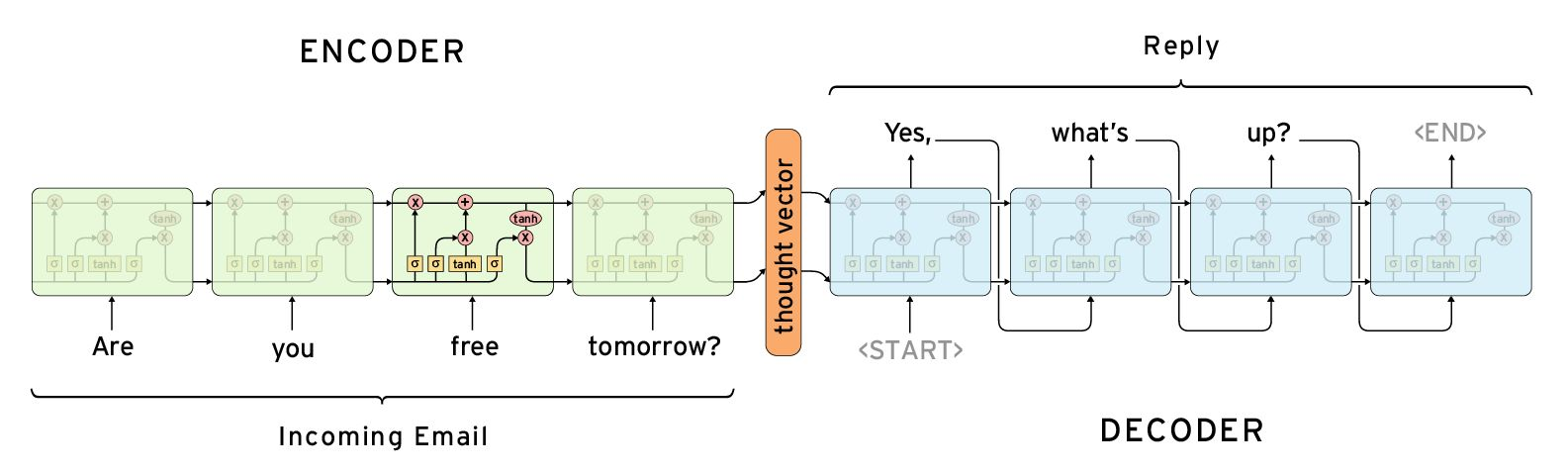
Deep Learning models, as amazing as they are, have trouble doing basic tasks like

interpretations of calculation-based questions, or telling novel data that it has not been trained on, like telling a joke for example, or even telling the time or remembering a name or context for extended sessions of time.

To overcome these shortcomings, we have integrated a rule-based approach with the deep-learning models, to create a novel architecture.

The deep-learning model used in this project is sequence-to-sequence model which consists of two RNNs - an encoder and a decoder. The encoder reads the input sequence, word by word and emits a context (a function of final hidden state of encoder), which would ideally capture the essence (semantic summary) of the input sequence. Based on this context, the decoder generates the output sequence, one word at a time while looking at the context and the previous word during each timestep.

Recurrent Neural Networks(RNNs), are a special kind of neural network that are capable of dealing with sequential data, like music, videos, text sequences or basically any sequence of symbols. Without knowing what the symbols are or their meaning, they infer the underlying semantics by analysing the sequence and the relative positions of the symbols. They accept an input sequence x and give you an output sequence y. However, crucially this output vector’s contents are influenced not only by the input you just fed in, but also on the entire history of inputs you’ve fed in in the past. Thus it enables the RNN to hold a hidden state that it gets to update every time step is called. This hidden state, or memory is the essence of the input sequence in numeric form.

The RNNs used are Dynamic RNNs which allow for variable sequence lengths, ideal for a chatbot setting.