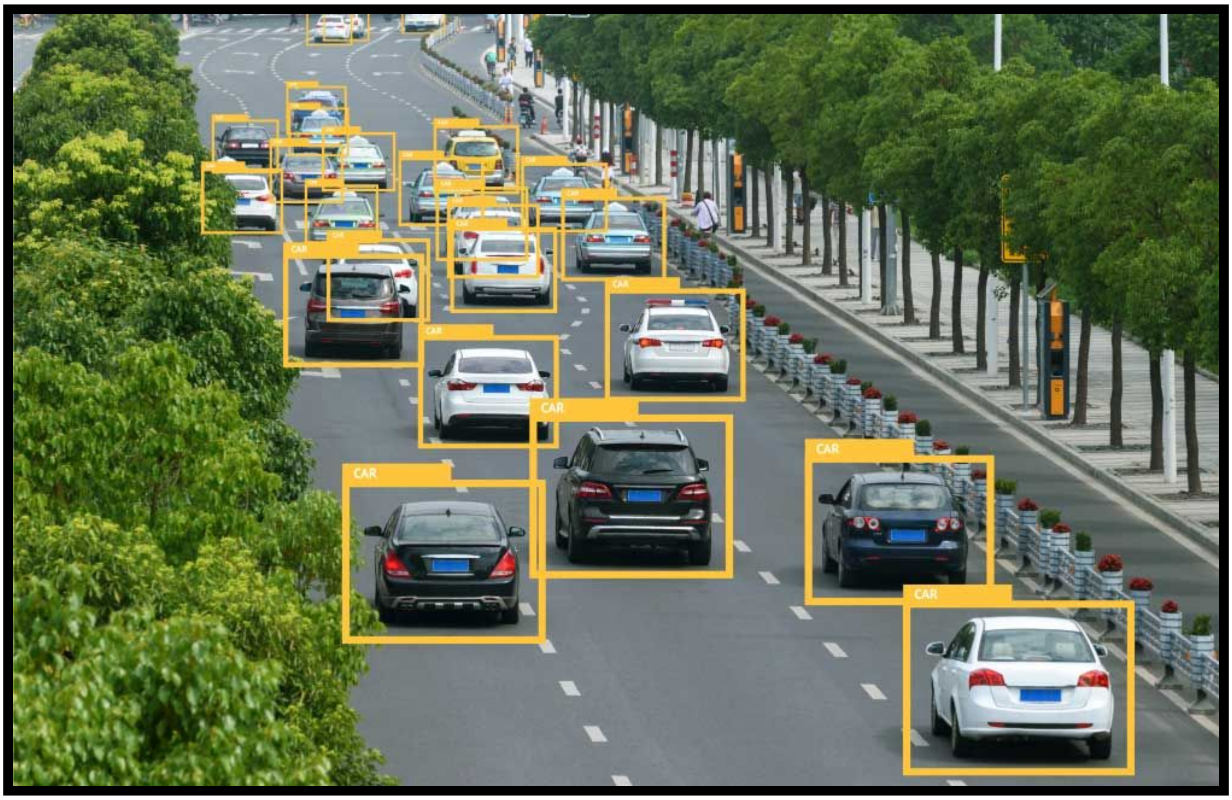
MULTICLASS CLASSIFICATION OF IMAGES



Project Report

EKLAVYA MENTORSHIP PROGRAMME

At

**SOCIETY OF ROBOTICS AND AUTOMATION,
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
MUMBAI**

JUNE 2020

ACKNOWLEDGMENT

We would like to sincerely thank our Mentors especially Rahul Singh and Abhinav for helping us out and answering our every query. Even though we slacked off a little bit during the project, Rahul made sure we are updating our daily spreadsheet. Abhinav made sure we understood the 'why' of the concepts and it was a great time spent with these guys.

All the group members were also supportive and encouraged each other during the project.

Akshay Bakshi
Email-ID: apbakshi@hotmail.com
Contact No. 99677 09649

Tushar Bauskar
Email-ID: tsbauskar96@gmail.com
Contact No. 91679 99883

Mohak Chandani
Email-ID: mohakc10@gmail.com
Contact No. 77381 48191

Purvank Bhiwgade
Email-ID: bhiwgadepurvank@gmail.com
Contact No. 95452 53853

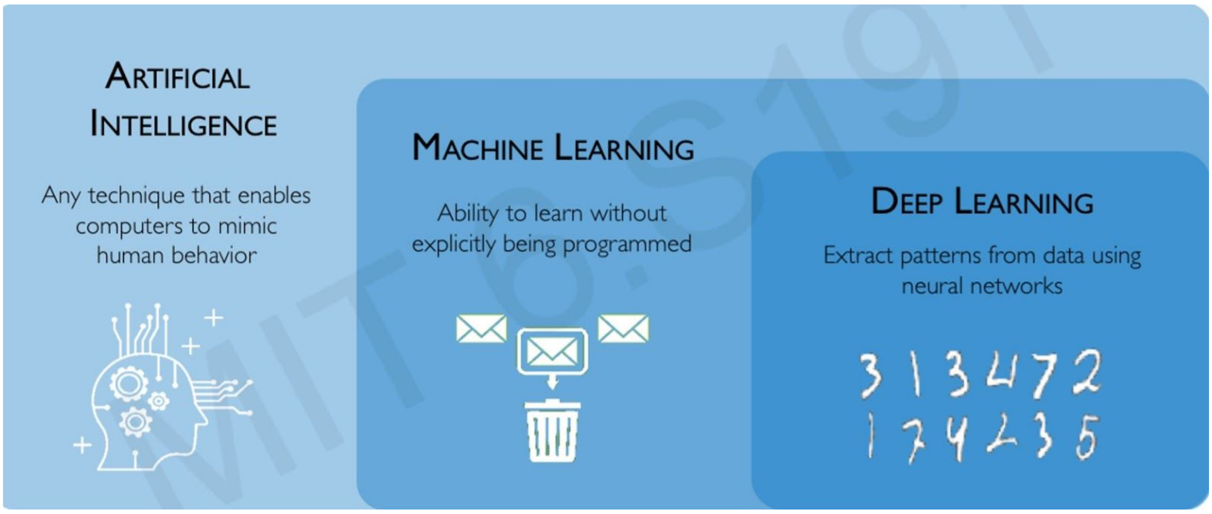
TABLE OF CONTENTS

1 Introduction

1.1 Deep Learning

"Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but nonlinear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. [...] The key aspect of deep learning is that these layers are not designed by human engineers: they are learned from data using a general-purpose learning procedure" – Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, Nature 2015. [9]

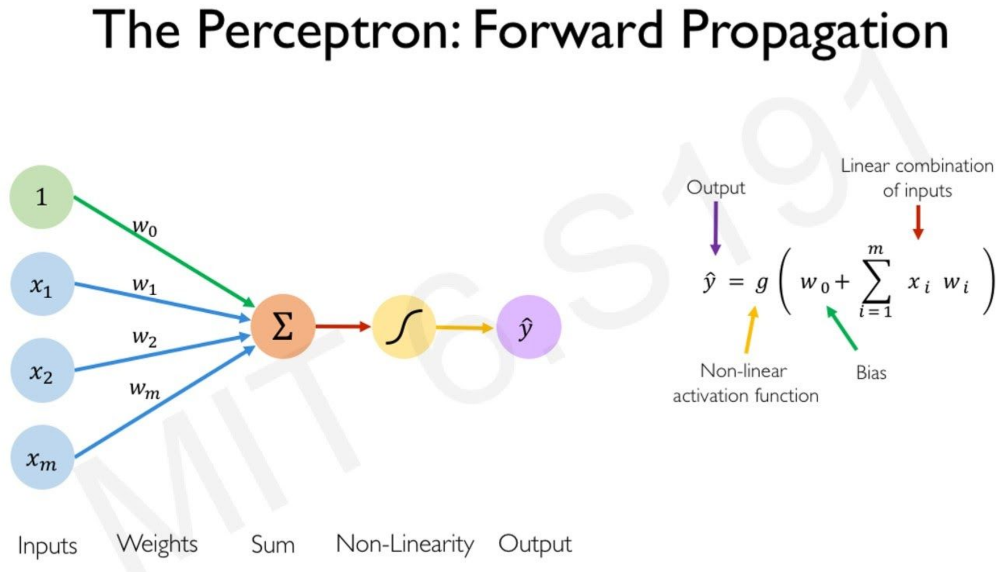
Deep learning is a subfield of machine learning, which is, in turn, a subfield of artificial intelligence (AI).



1.2 Mathematical Formulation

- Deep Learning consists of multiple Neural Networks which are connected to each other. Here, we are using Keras library with a Tensorflow backend.
- Each Neural Network consists of a Tensor(a Neuron).
- A Single Neuron is called the Perceptron.

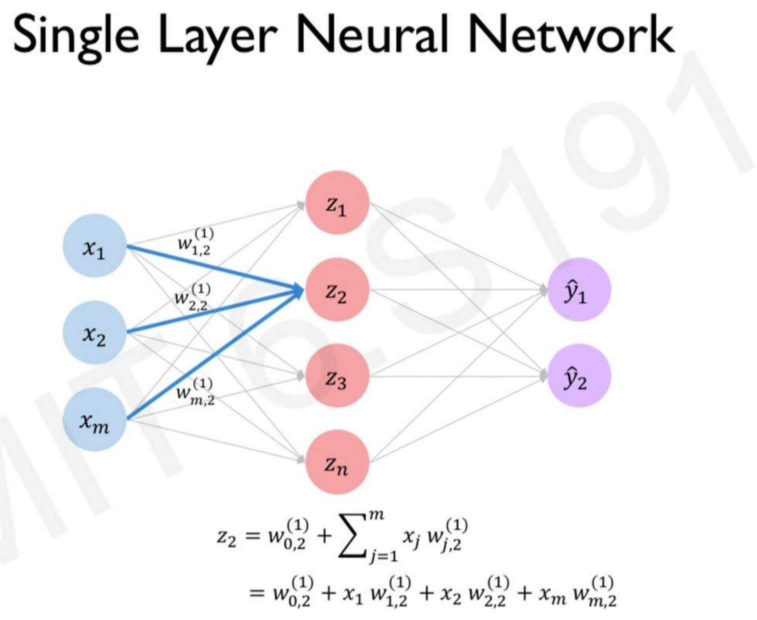
A Perceptron is given in the following figure.

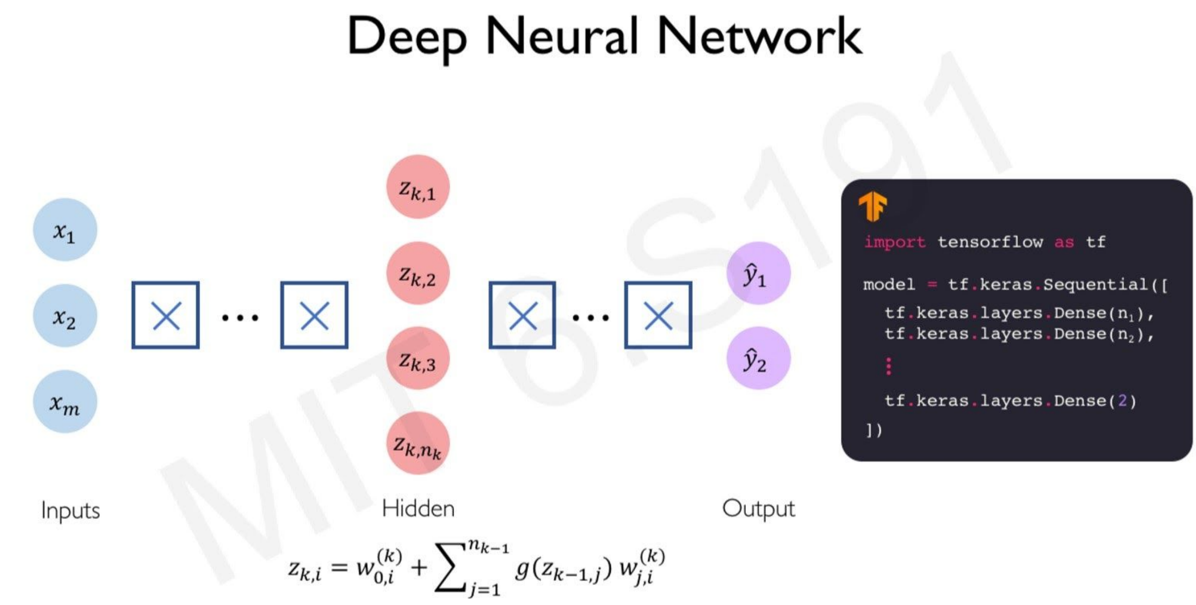


Mathematics behind a Perceptron

1. We take the inputs from the image in the form of an array.
2. Multiply the inputs with a randomly initiated weight vector and take a sum of the outputs. (A dot product between an input vector and a weight vector).
3. A bias may or may not be added.
4. It is then passed through a Non-linear function
5. Thus, we get the output

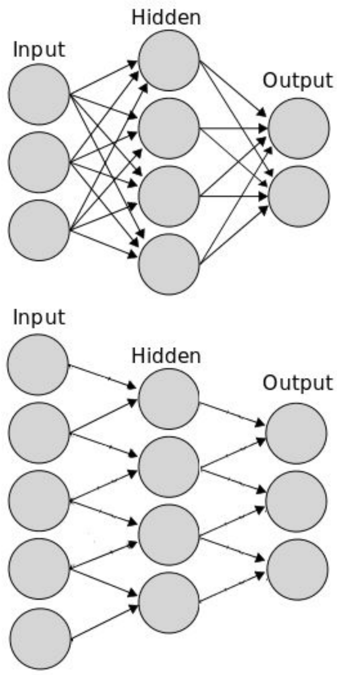
A number of perceptrons form a single layer Neural Network.





1.3 Convolutional Neural Networks (CNN)

A Convolutional Neural Network is similar to a Neural Network but not densely connected. CNN has dropout layers which makes the calculation a little bit less hectic.



In the above figure, the first diagram is NN and the second diagram is CNN.

2 Proposed Methods

- We used Supervised Learning to train our model.
- We were supposed to use YOLO to train our model and apply it in real life and video feed.

3 Implied Process

- We did use OpenCV to read the images(cv2.imread()) and the real time(frame by frame).
- We didn't use YOLO to train our models because we couldn't manage to get to that level of application of deep learning in a month (+ we didn't have any experience in AI before doing this).
- To train the model, we used CNN (LeNet Architecture), and we could get to the accuracy of 93% for classification of images among Bikes, Cars and Busses (model name in GitHub repo: hell.model).

4 Experimental Results and Analysis

- Testing on the deep learning model was done by about 30 unseen images of bikes, cars and busses that weren't included in the dataset. Once the testing image is imported using OpenCV, the image is resized to 28 by 28 pixels. The resized image was then converted to a NumPy array. After that, we add one more dimension to it to make it a 4th rank tensor with our NumPy array.
- Finally, we make the model predict the result by giving it the NumPy array and getting the zeroth element from it. We then use simple logic to figure out which vehicle the model is predicting.
- The experimental results were fairly accurate by achieving an accuracy of above 90%.

5 Conclusion and Future Work

- We have successfully tested this model on a realistic dataset of vehicle images from multiple angles. The LeNet architecture obtains better results, particularly in differentiating between cars and busses. Our recognition rate is above 90%. Furthermore, this project will allow us to combine it with other processes like license plate recognition systems to reject vehicles that are not recognized.

References

1. Sentdex OpenCV tutorial - https://youtu.be/Z78zbnLlPUA
2. MIT Deep Learning - https://youtu.be/njKP3FqW3Sk
3. Pyimagesearch - https://www.pyimagesearch.com
4. Car Dataset - https://ai.stanford.edu/~jkrause/cars/car_dataset.html
5. Bike Dataset - https://www.kaggle.com/phucbb/motorbike-zaloai
6. Bus Dataset - https://www.facebook.com/pg/davecarrtruckphotos/photos/?tab=album&album_id=1165367943495631&ref=page_internal