**AIFA**

**Course Assignment**

**- 19CS10042 Mahajan Rohan Raj**

**- 19CS10019 Battala Vamshi Krishna**

**Youtube Link:-**[**https://youtu.be/08yvQSAikG0**](https://youtu.be/08yvQSAikG0)

**OBJECT DETECTION USING PYTORCH**

**Detection of Anki Vector Robot**

**1. Problem Statement :**

* Developing an ML model in order to train Anki vector Robot to detect another Anki vector Robot through its camera feed.

**Problem Description:**

**Dataset used:** Anki Vector Robot Dataset.

**Dataset Description:**

The Anki Vector robot was first introduced in 2018. The Vector robot had been the cheapest fully functional autonomous robot. The Vector robot can be trained to recognize people; however, Vector does not have the ability to recognize another Vector. This dataset has been designed to allow one to train a model which can detect a Vector robot in the camera feed of another Vector robot.

Pictures were taken with Vector’s camera with another Vector facing it and had this other Vector could move freely. This allowed pictures to be captured from different angles. These pictures were then labeled by marking the rectangular regions around Vector in all the images. There is also a collection of pictures without Vector.

**Formal Problem Statement:**

INPUT: Dataset containing still images of Anki Vector Robot

OUTPUT: Bounding Box in images indicating the location of the robot present in it.

**2. AI Modelling :**

A typical object detection problem involves image localization and image classification. Here there is a single class, so the problem translates to localization of image. In other words, finding the Bounding Boxes. The problem of finding Bounding Boxes can be solved using deep learning techniques such as neural networks and convolutional neural networks. The performance of a model is evaluated using the distance between the expected and predicted bounding box for the expected class.

**3. Approach:**

Architecture:

There are two problems at hand which need to be solved. First, to tell whether an object is present or not, and second, to get the bounding boxes.

To solve the first problem we used an image classifier with two neurons at the output layer indicating the presence and absence of the object.

To solve the second problem we should have four output neurons emitting the values for the bounding boxes.

The challenge is that the first two neurons provide information about the classification whereas the other four provide the coordinates of bounding boxes.

The solution was to design the neural network with two branched outputs which share common feature extraction using convolutional layers and pooling layers.

Building Model:

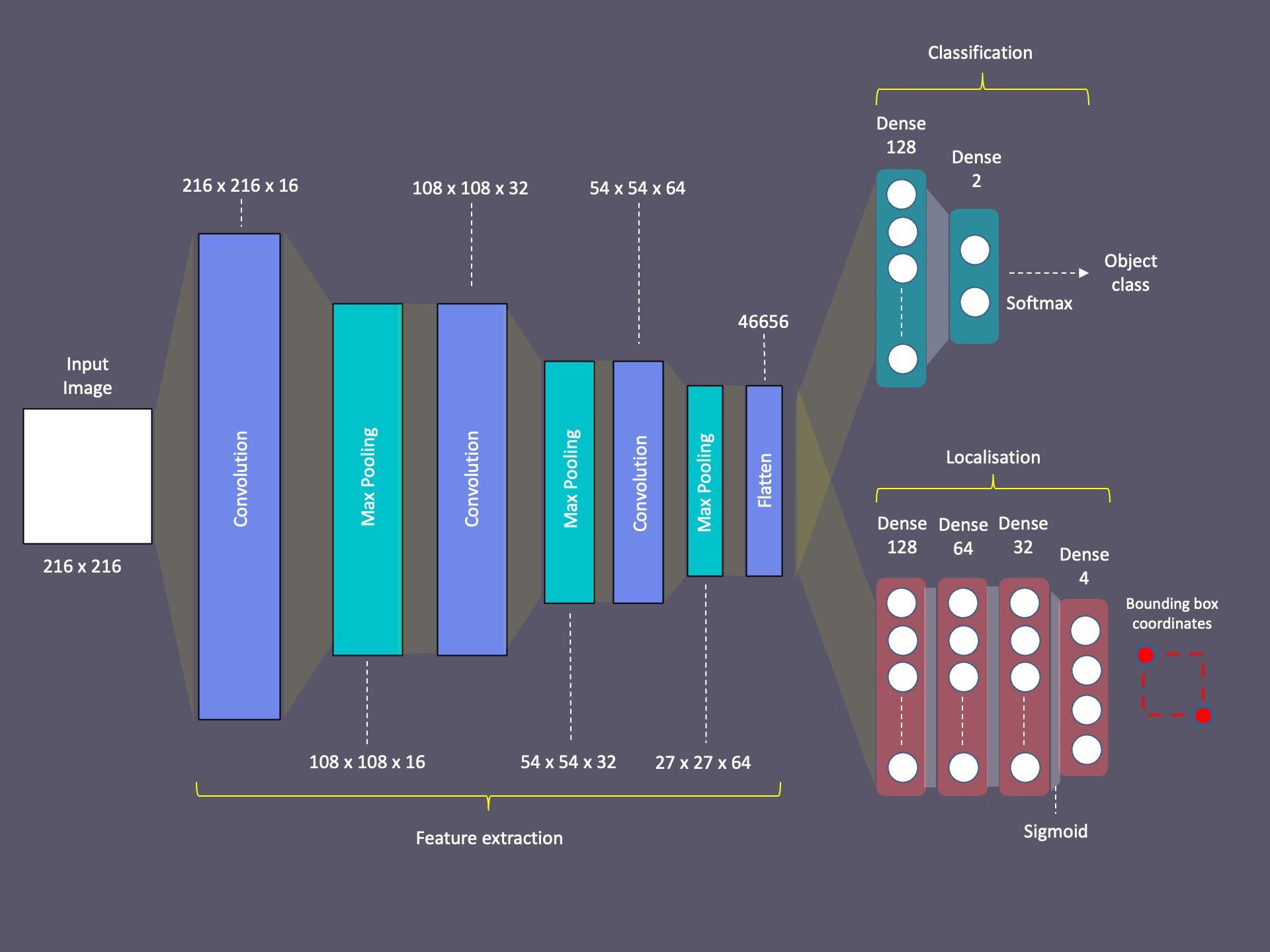
Defined the input layer by rescaling layer to transform data in the range [0,1]. Then created layers chaining the output of one layer to another. In total there are three convolutional layers and three Max pooling layers.

Then defined classification branch layers by inputting the flattened outputs and adding two dense layers.

Localization branch layers inputs from flattened outputs and adding 4 dense layers

Training:

Since there are two different types of outputs, we defined two types of error losses using the Sparse Categorical Cross-Entropy loss function for the classification head and the Mean Squared Error (MSE) for the localizer.



Results:

Accuracy-

**4. Analysis and Explanations:**

Why Machine Learning:

If the problem is to be worked out as a search problem then, for a one-megapixel image (means that each input to the network has one million dimensions) the search space with states as bounding boxes, the number of states itself crosses which makes searching difficult.

The problem can’t be formulated as a CSP as the constraints are heavily math involved which makes their computation costly and hard to represent.

With the help of Machine learning, we create models which have the ability to infer and make predictions based upon past data. Due to heavy complexity in representation and computation of object detection problems as a CSP or Search problem we use machine learning as the suitable and practical approach, which uses data and uses less structural and computing complexity,

Why Neural Networks:

All machine learning models remain appropriate when we deal with data whose features are specified. We might anticipate finding patterns and correlations among the features. But with images the case is different. There are no explicitly declared features and the amount of data represented by a single image is huge.

The statistical models need to become more memory efficient while simultaneously being able to spend more time on optimizing these parameters, due to an increased computational budget. Consequently, the sweet spot in machine learning and statistics moved from linear models to deep neural networks.

A neural network works similarly to the human brain’s neural network. A “neuron” in a neural network is a mathematical function that collects and classifies information according to a specific architecture. A neural network contains layers of interconnected nodes. In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications. Hidden layers fine-tune the input weightings until the neural network’s margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis

Convolutional Neural Networks:

For a collection of megapixel images, even an aggressive reduction to one thousand hidden dimensions would require a fully-connected layer characterized by , which is not feasible. While we might be able to get away with one hundred thousand pixels, our hidden layer of size 1000 grossly underestimates the number of hidden units that it takes to learn good representations of images, so a practical system will still require billions of parameters. Moreover, learning a classifier by fitting so many parameters might require collecting an enormous dataset

Images exhibit two important structural properties: *Translation Invariance* and *locality*.

Translation Invariance implies that a shift in the input should simply shift the immediately hidden layer. This is possible only when the weights for a pixel do not depend on the location of the pixel. The locality principle believes that we should not look far away from the present pixel location in order to specify the output of the immediately hidden layer. Which implies that the weights should become 0 after a width . Convolutional Neural Networks exploit these rich structures of the image.

Improvements: