**Databyte Inductions 2022**

**Topic 2**

**- Basic Image classifier CNN**

-  So Neural Nets are your go to approach for solving any problem.

-  Create a labeled dataset of Avengers images- Captain America, Iron Man,Black Widow, Hulk,Thor. (Try ​*scraping* ​images from internet , instead of manually creating the dataset)

-  Train a CNN that is able to classify an unseen image with reasonable accuracy.

-  You can use frameworks like ​PyTorch​ and ​Keras ​to simplify your workflow.

My approach to this problem is using one of the deep learning algorithms named CNN, also known as ConvNets. It is mainly used for image processing and object detection.

CNN's have multiple layers that process and extract features from data:

**Convolution Layer:** CNN has a convolution layer that has several filters to perform the convolution operation.

**Rectified Linear Unit (ReLU):** CNN's have a ReLU layer to perform operations on elements. The output is a rectified feature map.

**Pooling Layer:** The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map. The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.

**Fully Connected Layer**: A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.

I used tensorflow.keras which is used for creating deep models also used for distributed training of deep learning models. I used Adam Optimizers which used to change the attributes of deep learning model such as weights and learning rate to reduce the losses and then plotted graph between accuracy, loss of model and epochs using matplotlib.

The model was then tested on 30 images, and it predicted almost everything correctly trained using CNN.

I finally thank the Databyte team for this induction opportunity as I learned a lot in ML.