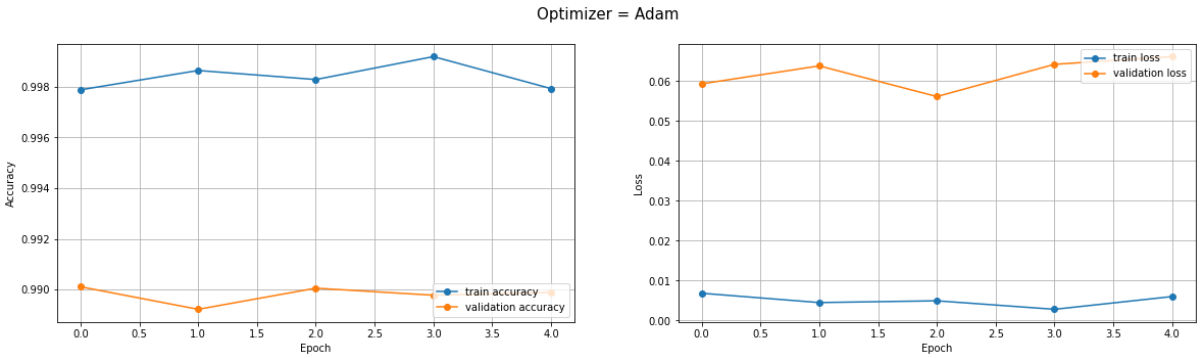
**Name: Nuhan Islam**
**ID: 18-37898-2**
**Course: CVPR [B]**

**Project report**
**Title: CNN architecture to classify the MNIST handwritten dataset**

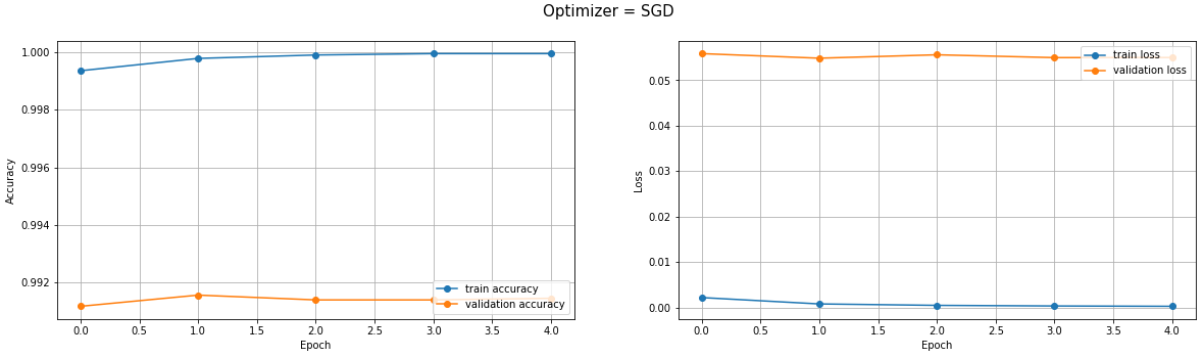
**Abstract**: MNIST data collection contains a large number of handwritten text data sets and is often used for CNN deep model training, testing, and validation. We've built an efficient model with many relu and pooling layers in this post. Which of the following is evaluated with 98.45 percent accuracy on the MNIST data set? Furthermore, this model is evaluated on a similar type of random picture data set, which yields considerable accuracy results.

**Introduction**: Recognizing handwritten numbers is crucial and has a variety of applications in online handwriting recognition. When attempting to address this problem, there are a variety of obstacles to overcome. Handwritten digits have a variety of strokes, sizes, thicknesses, orientations, and distances from the margins, making identification more difficult. The aim was to build a model that could categorize a digit based on its pattern by using CNN to detect a handwritten digit with a similar pattern. A Convolutional Neural Network is a type of artificial neural network used in deep learning to interpret visual data. In this report, we have shown a novel CNN model to achieve the best performance on the handwritten digit recognition task from random images and the MNIST dataset. Images are of numerical handwritten digits taken from a resource. It was tried to get the accuracy over 98% by modifying the model architecture. It is also tested with different optimizer (Adam, SGD, RSMProp) to identify actual accuracy.

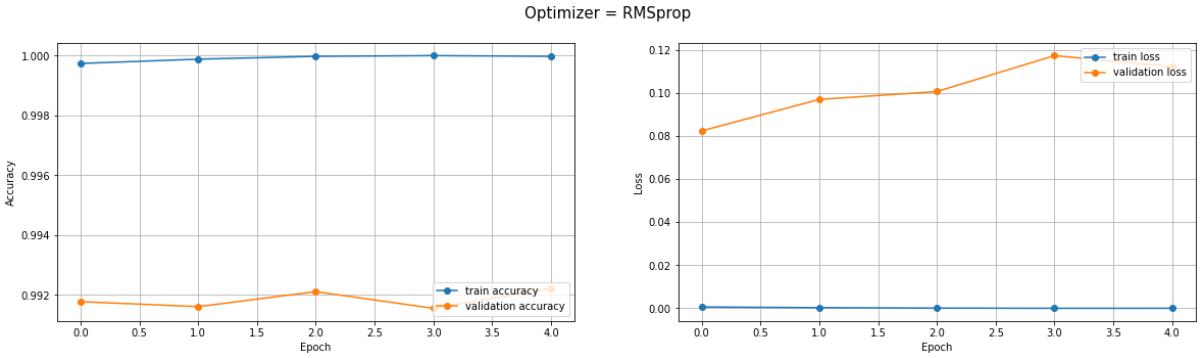
**Result**:
Test accuracy using Adam is 99.08%.




Test accuracy using SGD is 99.35 %.



Test accuracy using RMSprop is 99.36 %.




**Discussions**: According to the report result, RMSprop optimizer is preferable due to its accuracy. This optimizer lost ratio is 0.79%. It is a very robust optimizer which has pseudo-curvature information. It can deal with stochastic objectives very nicely, making it applicable to mini batch learning. It converges faster than momentum.