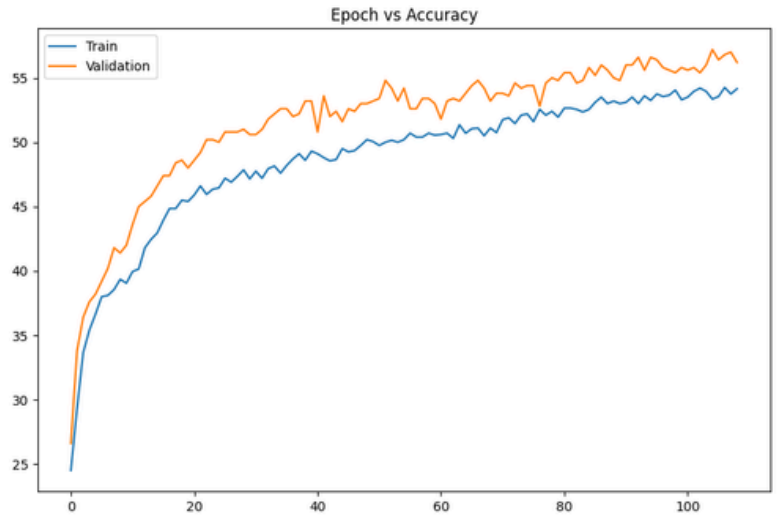
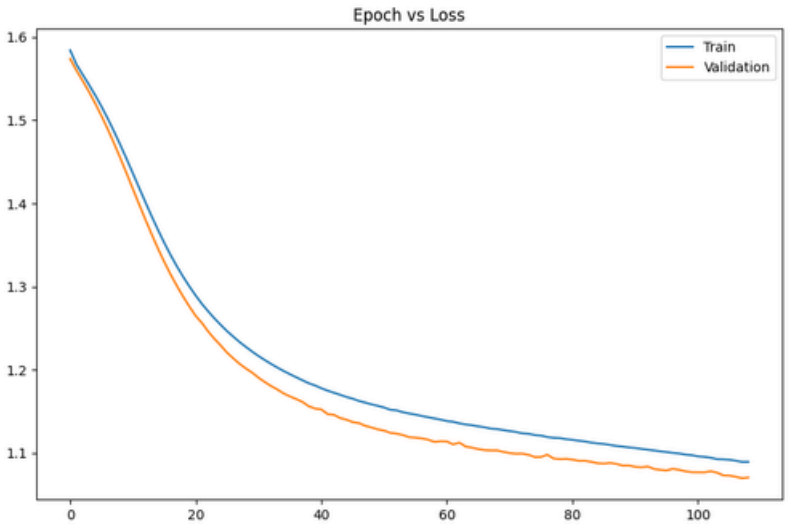
Task 1: Comparison of optimization methods for classification on Image dataset 1

Hyper parameters: Seed - 42 Learning rate - 0.0003 Threshold- 1e-5

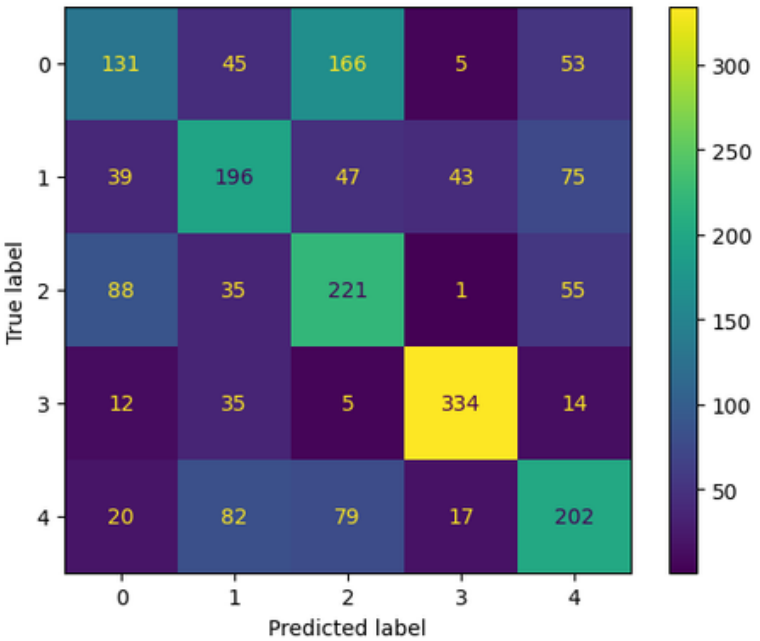
Delta Rule (Stochastic Gradient Descent - SGD)

- Updates parameters by moving in the opposite direction of the gradient of the loss function with respect to the parameters, scaled by a learning rate.
- Epochs to converge -109
- Training accuracy - 54.15%
- Validation accuracy - 56.2%

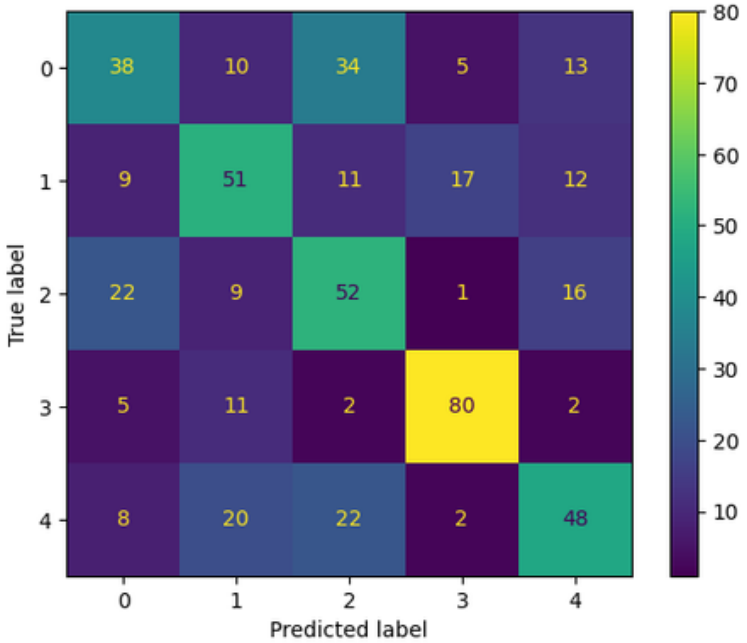




Confusion matrix for train data

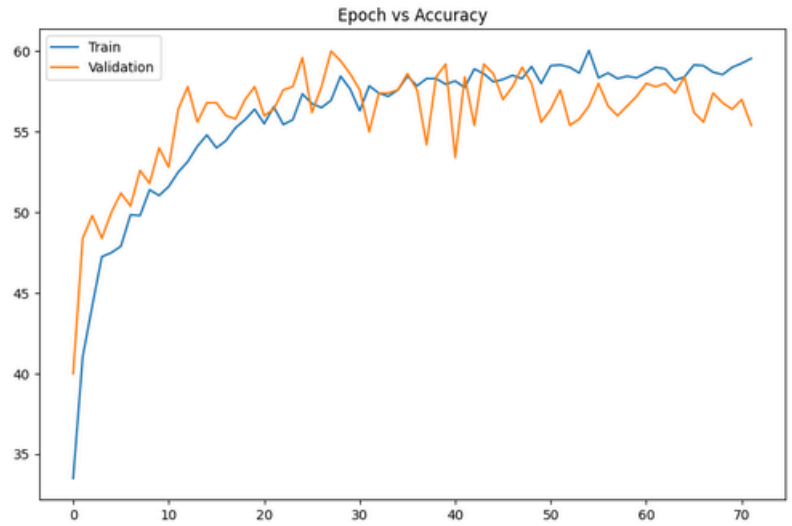


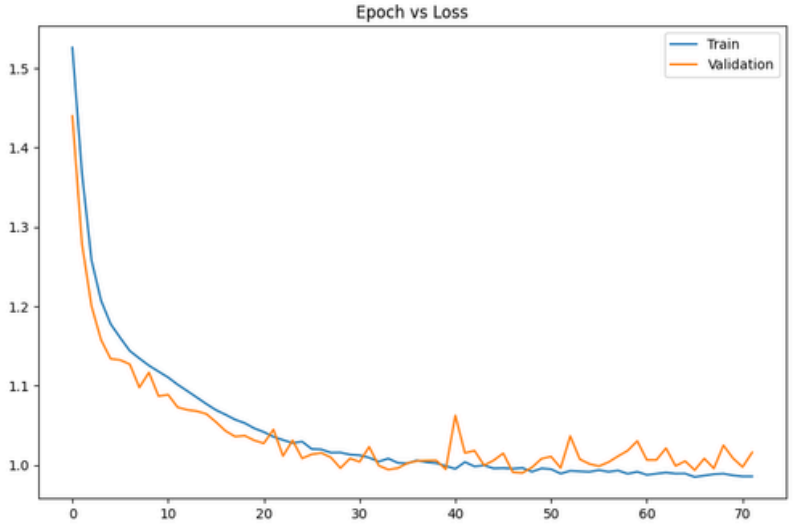
Confusion matriix for test data



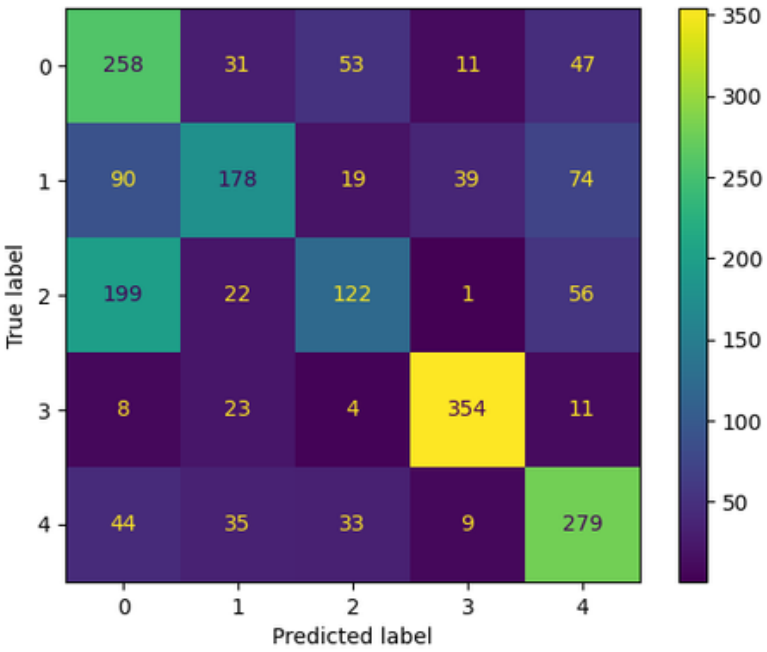
Generalized Delta Rule(SGD with Momentum)

- Adds a fraction of the previous weight update to the current update, helping accelerate convergence in Stochastic Gradient Descent (SGD) by maintaining velocity along relevant directions while reducing oscillations in noisy or high-curvature regions.
- Momentum -0.9
- Epochs to converge -72
- Training accuracy - 59.55%
- Validation accuracy - 57%

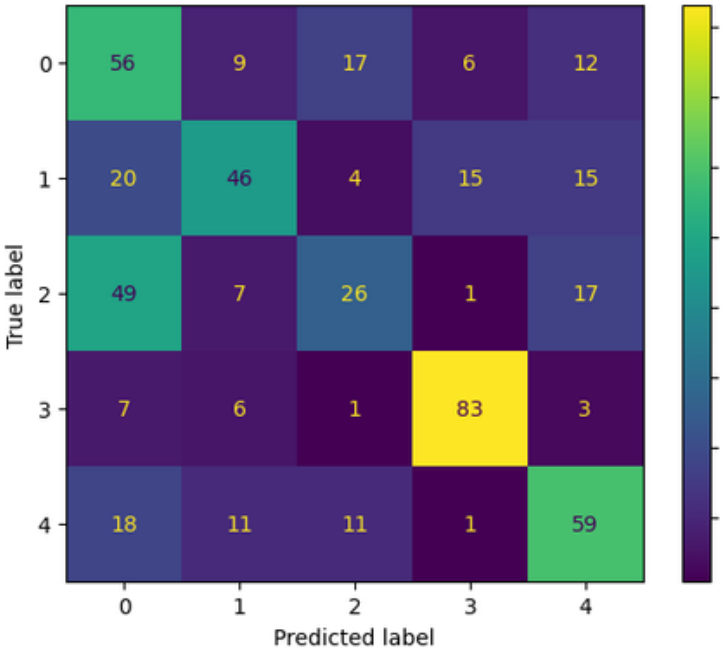




Confusion matrix for train data

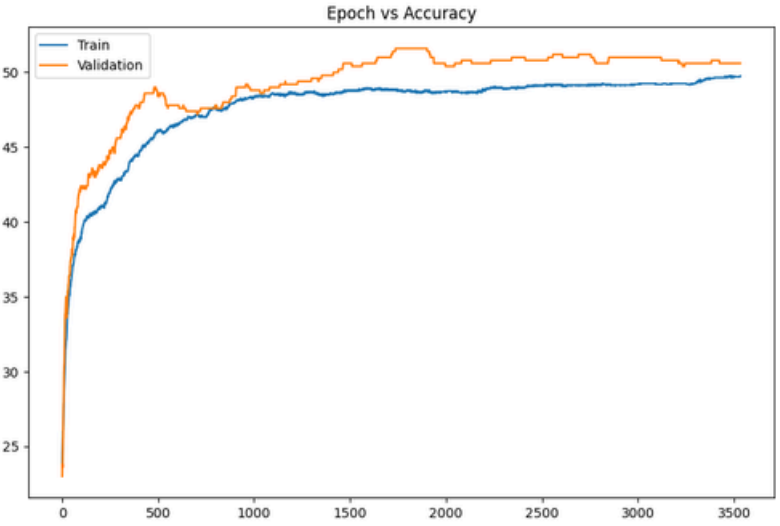


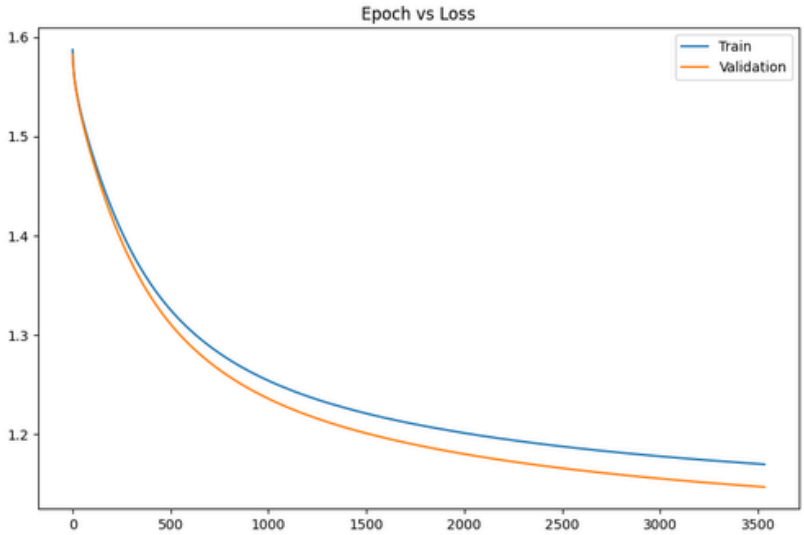
Confusion matriix for test data



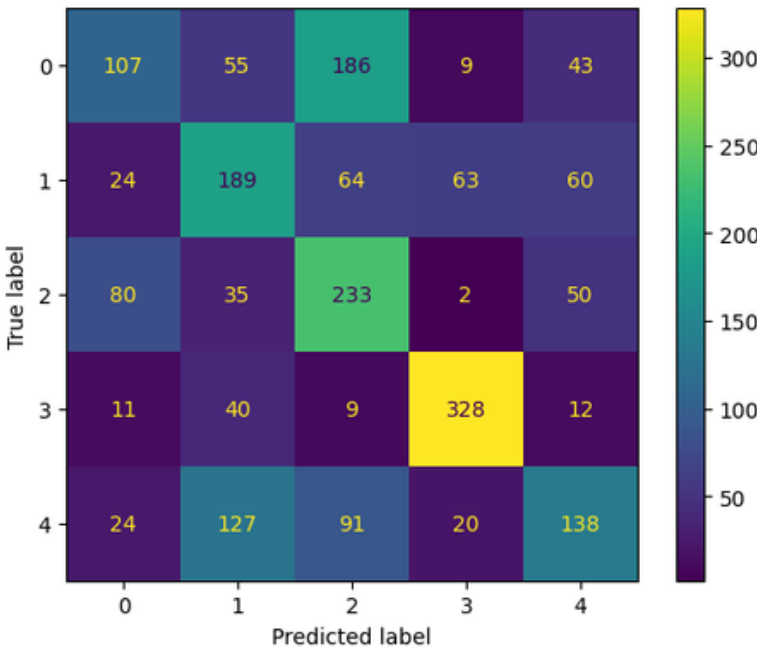
Adagrad- Adaptive Gradient

- Dynamically adjusts the learning rate for each parameter, making larger updates for rarely occurring parameters and smaller updates for frequently occurring ones. This helps in handling sparse data effectively.
- Epochs to converge -3534
- Training accuracy - 49.75%
- Validation accuracy - 50.6%

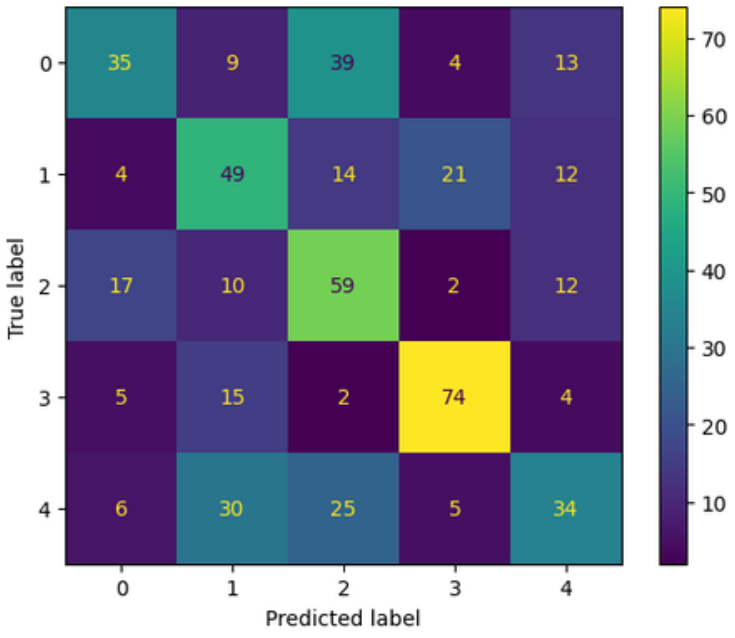




Confusion matrix for train data

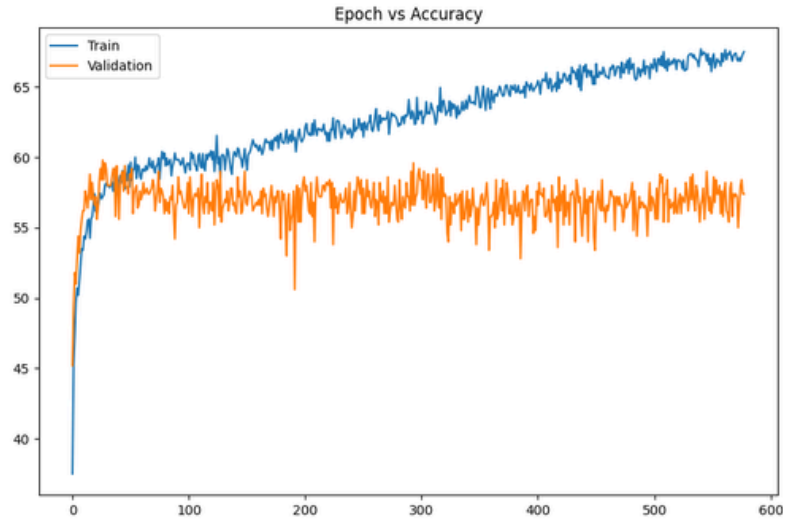


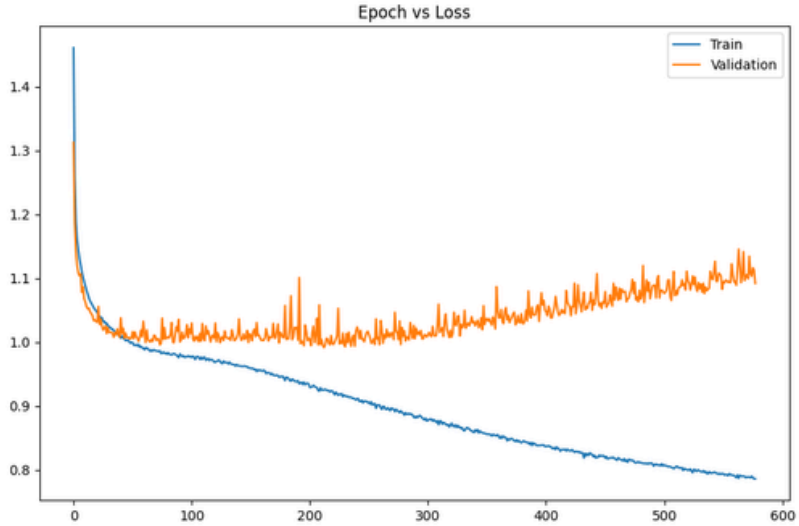
Confusion matriix for test data



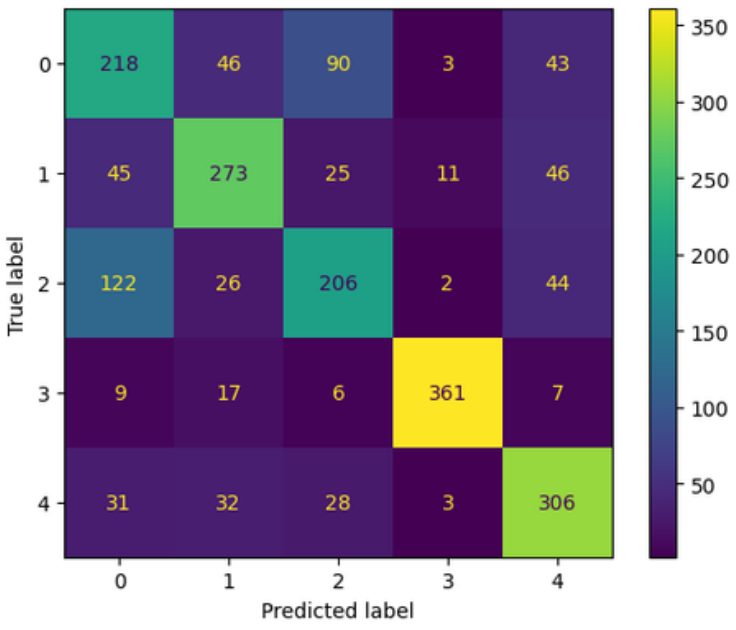
RMSProp - Root Mean Squared Propagation

- Improves upon Adagrad by using an exponentially decaying moving average of squared gradients instead of accumulating all past gradients. It prevents the learning rate from shrinking too much over time
- Epochs to converge -578
- Training accuracy - 67.5%
- Validation accuracy - 58%

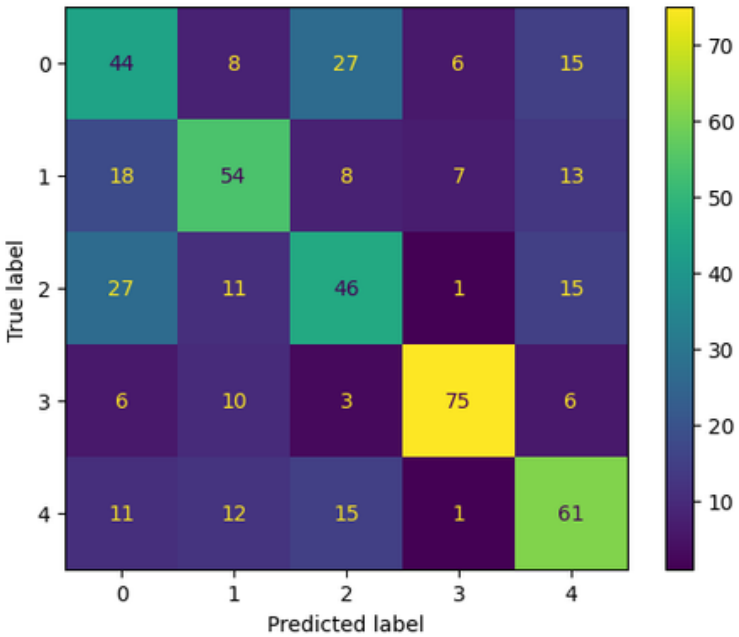




Confusion matrix for train data

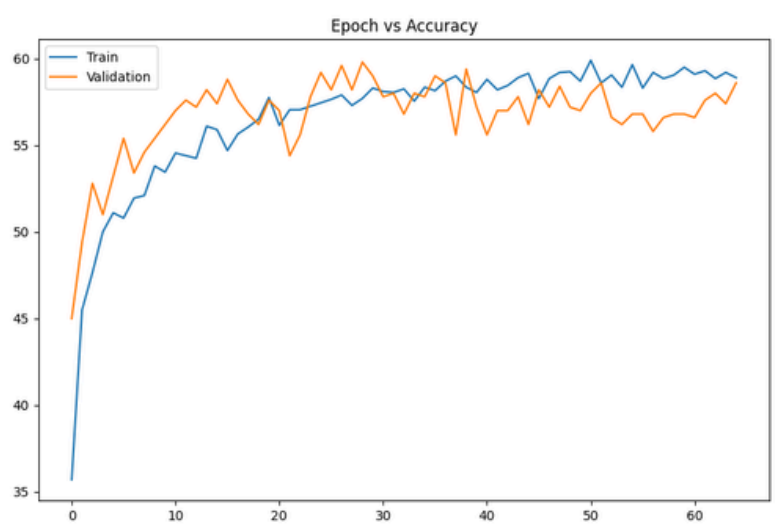


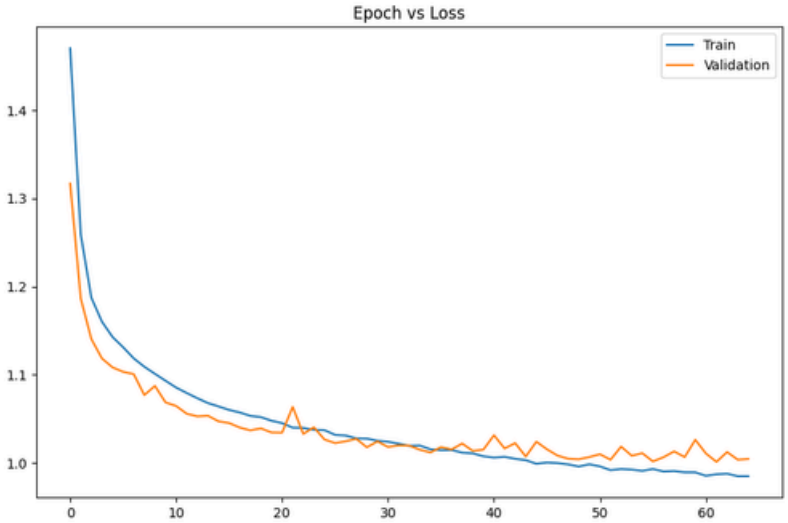
Confusion matrix for test data



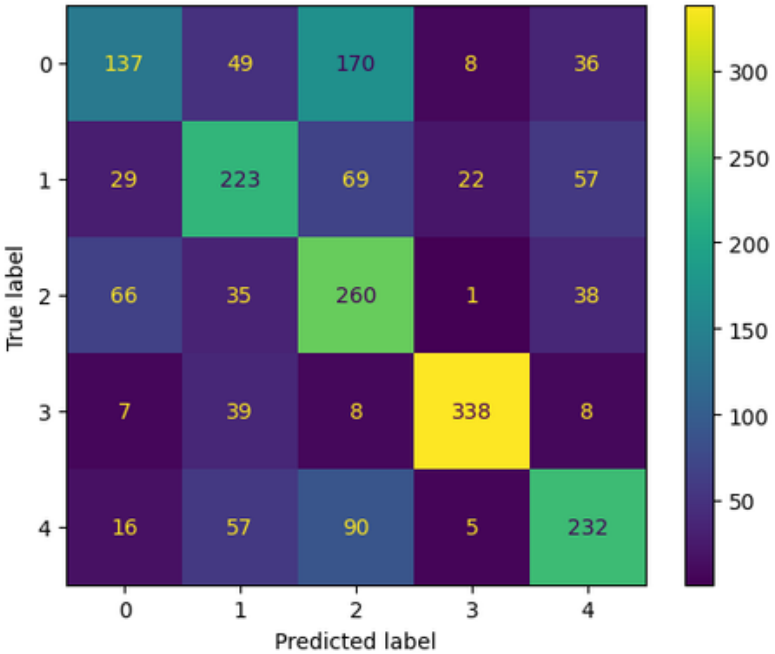
AdaM - Adaptive Movement Estimation

- Combines the strengths of RMSprop and SGD with momentum. It maintains individual learning rates for each parameter and estimates both the first moment (mean of gradients) and second moment (uncentered variance of gradients) to adaptively adjust updates. Faster convergence and automatic learning rate adjustments.
- Epochs to converge -65
- Training accuracy - 59.2%
- Validation accuracy - 58%

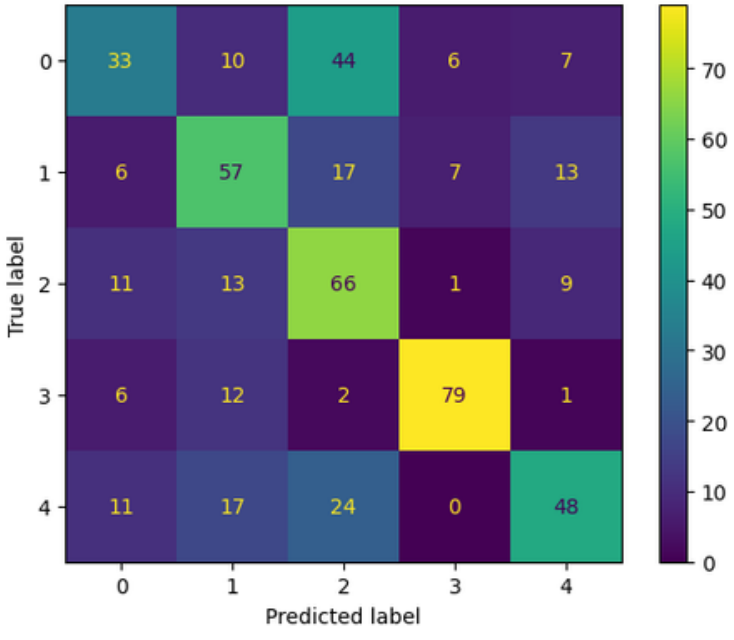




Confusion matrix for train data



Confusion matrix for test data



OBSERVATIONS

- **Delta Rule** is the most basic optimization method, often requiring many iterations to achieve convergence.
- **SGD** with Momentum improves upon standard SGD by incorporating moving averages of past updates, making optimization faster and more stable, though it may not always yield the best convergence. It is better than delta rule as seen from the accuracy values and took less iterations than Delta rule
- **Adagrad** adapts learning rates for individual parameters, making it effective for sparse data, but it tends to slow down significantly over time. Took more than 3000 epochs to converge and each epoch was generally slow. There was no significant improvement in the accuracy as well.
- **RMSProp** mitigates Adagrad's issue of diminishing learning rates by using an exponentially decaying average of squared gradients, making it more suitable for various scenarios. Took one- sixth of epochs taken by Adagrad but was able to give a significantly higher accuracy as seen from the data and the confusion matrix
- **Adam** combines the benefits of RMSProp and momentum, leading to an adaptive, efficient, and robust optimization method widely used in deep learning. It just took 65 epochs to converge, was the quickest training of all. It was able to give an accuracy near to that of RMSProp with just a fraction of epochs making it the most efficient optimization technique of all