Pytorch Implementing Neural networks

In this project, a neural network model was created and implemented using PyTorch to classify images. The model consisted of two convolutional layers and two fully connected layers. The model was trained on a dataset of images and their respective labels. The implementation of the model involved using the PyTorch library to define the architecture and parameters of the network.

To train the model, the Adam optimizer was used, which performed better compared to the stochastic gradient descent optimizer. The learning rate was set to 0.001, and the momentum was set to 0.9.

The model was evaluated on the training and testing datasets, and the F1\_score was used as the evaluation metric. The model performed relatively poorly, with an F1\_score of 27% for the training set and 24% for the testing set.

In conclusion, this project showed that the implementation of a neural network for image classification is a complex task that requires careful consideration of hyperparameters and optimization algorithms. The Adam optimizer was found to be more effective than stochastic gradient descent in this project. However, the model's performance was not satisfactory, suggesting that further optimization and model refinement are necessary to improve performance.

Based on these findings, it is recommended that future work focuses on exploring additional hyperparameters and architectures that can enhance the model's performance. Additionally, it may be beneficial to collect a more extensive and diverse dataset of images to increase the model's robustness and accuracy.