**Name: Israr Qayyum**

**Bytewise Limited**

**Project Proposal**

Project Title: **Image Classification using Convolutional Neural Networks (CNNs)**

Objective:

The objective of this project is to build a Convolutional Neural Network (CNN) model to classify images from a dataset. The model will be trained on an open-source dataset and will aim to correctly identify categories based on image features. This project will demonstrate the ability to work with CNNs and deep learning in solving image classification problems.

Problem Statement:

Image classification is a common task in the field of computer vision. With the increasing use of images in various fields such as healthcare, retail, and technology, it is important to develop efficient models that can correctly classify images into relevant categories. This project will focus on training a CNN model to classify images from a given dataset into predefined classes.

Dataset:

The dataset to be used will be sourced from Kaggle. A suitable dataset for beginners is the CIFAR-10 dataset, which contains 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images.

Alternatively, we can use a smaller or more specific dataset like Fashion MNIST or another publicly available dataset based on the student's preference.

Proposed Methodology:

1. Data Preprocessing:

* Load the dataset (either CIFAR-10, Fashion MNIST, or another simple dataset).
* Perform image normalization and reshape the images if needed.
* Split the data into training and testing sets.

2. Model Building:

* Build a basic CNN architecture consisting of convolutional layers, pooling layers, and fully connected layers.
* Use activation functions like ReLU and softmax for classification.
* Compile the model using categorical cross-entropy loss and an optimizer like Adam.

3. Model Training:

* Train the CNN model using the training data.
* Use appropriate evaluation metrics such as accuracy to assess model performance.
* Perform validation using the test set to measure the generalization of the model.

4. Results and Evaluation:

* Evaluate the model on the test dataset.
* Visualize the training process with loss and accuracy graphs.
* Display the confusion matrix to show the classification performance.

5. Conclusion:

* + Summarize the model's performance and discuss potential improvements.

Tools & Libraries:

* + Programming Language: Python
  + Libraries: TensorFlow, Keras, NumPy, Matplotlib, and Pandas