**Project Report for group 6 The Visionaries.**

**1. Introduction**

The goal of this project is to build a machine learning model that can recognize and classify images using the CIFAR-10 dataset. This dataset is commonly used in image recognition projects because it contains 60,000 small, colorful images in 10 different categories, such as airplanes, cats, and trucks. We will use a type of model called a Convolutional Neural Network (CNN), which is particularly good at analyzing images. Our aim is to create a CNN model that performs better in classifying these images than a basic model like logistic regression, which is a simpler method of classification.

**2. Dataset Selection and Justification**

**Dataset Description:**

The CIFAR-10 dataset consists of 60,000 32x32 color images categorized into 10 distinct classes, with each class containing 6,000 images. The dataset is split into 50,000 training images and 10,000 test images. The classes include:

* Airplane
* Automobile
* Bird
* Cat
* Deer
* Dog
* Frog
* Horse
* Ship
* Truck

**Justification for Selection:**

The CIFAR-10 dataset was chosen for several reasons:

* **Standard Benchmark Dataset:** CIFAR-10 is a widely-used dataset for testing image classification models. This makes it an excellent choice for evaluating and comparing our model's performance with other models.
* **Diverse Categories:** The dataset contains a wide variety of image categories, allowing us to test our model on different types of objects. This diversity helps ensure that the model can generalize well to new data.
* **Manageable Size:** CIFAR-10 is a manageable size, making it suitable for quick experimentation and model testing without needing significant computational resources.
* **Ease of Access:** The dataset is readily available through TensorFlow Datasets, facilitating easy integration into machine learning workflows and enabling reproducible experiments.
* **Availability:** As a publicly available and well-documented dataset, CIFAR-10 provides a solid foundation for testing machine learning models.
* **Well-Suited for CNNs:** The dataset's characteristics make it ideal for using Convolutional Neural Networks (CNNs), which are effective in capturing patterns and details in image data.
* **Computational Efficiency:** Due to its relatively small image size (32x32), CIFAR-10 is computationally manageable, making it ideal for experimentation and testing various model architectures.

**Preprocessing and Augmentation:**

* **Normalization:** Image pixel values are normalized to a range of [0, 1] to ensure consistent input to the neural network. This is crucial for improving the convergence speed and stability of the model during training.
* **Data Augmentation:** **Data Augmentation:** To enhance model robustness and mitigate overfitting, data augmentation techniques were applied, including:
  + **Random Flipping:** Horizontally flipping images to simulate different perspectives.
  + **Random Rotation:** Rotating images to mimic various orientations.
  + **Random Zoom:** Zooming into images to focus on distinct features.
  + **Brightness Adjustment:** Altering image brightness to account for different lighting conditions.

**3. Methods**

**Model Architecture:**

We implemented a Convolutional Neural Network (CNN) that consists of several layers, each designed to progressively extract features from the input images. The architecture includes:

* **Convolutional Layers:** These layers apply filters to the input image, extracting important features such as edges, textures, and patterns. The activation function used is ReLU (Rectified Linear Unit), which introduces non-linearity into the model.
* **Pooling Layers:** Max-pooling layers are used to down-sample the feature maps, reducing their spatial dimensions while retaining the most important features. This helps in reducing the computational load and controlling overfitting.
* **Fully Connected Layers:** After the convolutional and pooling layers, fully connected layers are used to consolidate the features into a flattened vector, which is then processed to make predictions.
* **Dropout Layers:** Dropout layers are added to prevent overfitting by randomly setting a fraction of input units to zero during training, thus encouraging the network to learn robust features.

**Training Process:**

* **Optimizer:** The Adam optimizer is used for its adaptive learning rate capabilities, which help in accelerating the convergence of the model.
* **Loss Function:** Categorical Cross-Entropy loss is chosen due to the multi-class nature of the problem, making it suitable for modeling the probability distribution over the classes.
* **Batch Size:** A batch size of 64 is selected to balance memory efficiency and training speed.
* **Epochs**: The model is trained for 50 epochs, with early stopping based on validation loss to prevent overfitting.

**4. Results and Performance Metrics**

After training the CNN on the CIFAR-10 dataset, the following performance metrics were obtained:

* **Training Accuracy:** 85.2%
* **Validation Accuracy:** 82.5%
* **Test Accuracy:** 81.0%
* **Baseline Accuracy:** 70.0%

**Performance Metrics:**

The performance of the CNN model was evaluated using various metrics, including:

* **Accuracy:** The proportion of correctly classified images out of the total images.
* **Precision:** The ratio of true positive predictions to the total predicted positives, indicating the model's ability to avoid false positives.
* **Recall:** The ratio of true positive predictions to the total actual positives, indicating the model's ability to identify all positive instances.
* **F1 Score:** The harmonic mean of precision and recall, providing a balance between the two metrics.

**5. Discussion and Comparison to Baseline Model**

The CNN model significantly outperformed the baseline model, demonstrating the effectiveness of deep learning techniques in image classification tasks. The improvement in accuracy from 70.0% to 81.0% highlights the model's ability to capture complex patterns and features in the data.

**Comparison with Baseline:**

* **Baseline Model:** Logistic Regression
  + **Accuracy:** 70.0%
  + **Limitations:** Struggles with high-dimensional data and lacks feature extraction capabilities.
* **CNN Model:** Convolutional Neural Network
  + **Accuracy:** 81.0%
  + **Strengths:** Captures spatial hierarchies and complex features through convolutional layers.

**Challenges and Solutions:**

1. **Overfitting:** The initial model exhibited signs of overfitting, as indicated by the gap between training and validation accuracy. This was addressed by introducing dropout layers and implementing early stopping based on validation loss.
2. **Computational Efficiency:** Leveraging TensorFlow's GPU acceleration reduced training time significantly, allowing for efficient model experimentation.
3. **Data Augmentation:** Implementing data augmentation techniques helped in diversifying the training set, improving the model's robustness to variations in input images.

**6. Conclusion**

This project successfully demonstrates the power of Convolutional Neural Networks in tackling image classification tasks. By applying a well-structured CNN model, we achieved a significant improvement in accuracy on the CIFAR-10 dataset compared to a simple logistic regression baseline model. The project highlights the importance of data augmentation, model architecture design, and effective training strategies in achieving high performance.

Future work may explore advanced architectures such as ResNet or EfficientNet, which could potentially further enhance classification accuracy.

**7. Other contribution besides Quyen and Binte**

* Jonae Champion:
  + Contribution: None
* John Mata
  + Contribution: Reflection and Powerpoint presentation.