**Computer Vision TAE 2 Report**

***Shyam Nair***

***Cse Aiml - 7th Sem - 56***

## **1. Introduction**

In computer vision, **Convolutional Neural Networks (CNNs)** have become a vital tool for analyzing and classifying images. This report focuses on implementing CNN architecture from scratch using TensorFlow. The activity involved working with popular datasets like MNIST (handwritten digits) and CIFAR-10 (object images), building a CNN model with essential components such as convolutional layers, activation functions (ReLU), pooling layers, and fully connected layers. This task provides practical insights into how CNNs extract and process visual features, which enables accurate image classification.

## **2. Objective**

The objective of this task was to:

1. Understand the fundamental components of CNN architecture.
2. Build a CNN from scratch using TensorFlow or PyTorch.
3. Train the CNN model on the MNIST and CIFAR-10 datasets.
4. Evaluate the model’s performance by analyzing accuracy and loss.

## **3. CNN Architecture Overview**

A CNN consists of multiple layers that help in the extraction of visual features from images. Below are the essential layers of a CNN:

* **Convolutional Layer**: Extracts features from the input image by applying convolution filters.
* **ReLU Activation**: Introduces non-linearity to the model, making it capable of learning complex patterns.
* **Pooling Layer**: Reduces the dimensionality of the data while retaining the important features.
* **Fully Connected Layer**: Processes the features extracted by the convolutional and pooling layers to perform classification.

## **4. Dataset and Preprocessing**

For this activity, we worked with two datasets:

* **MNIST**: A collection of 60,000 training images and 10,000 test images of handwritten digits (0–9).
* **CIFAR-10**: A dataset containing 60,000 images categorized into 10 classes, such as airplanes, cars, birds, cats, etc.

The images were preprocessed by normalizing their pixel values between 0 and 1 for faster convergence during training.

## **5. CNN Model Implementation**

### **a. Model for MNIST Dataset**

The model was built using the **TensorFlow** library. The architecture included:

* Two convolutional layers, each followed by a ReLU activation function.
* Max-pooling layers for dimensionality reduction.
* A flattening layer to convert 2D feature maps into a 1D vector.
* A fully connected layer for classification into 10 output categories.

### **b. Model Training**

The model was trained on the MNIST dataset for 10 epochs, using the **Adam optimizer** and **categorical cross-entropy** as the loss function. The training achieved an accuracy of over 98% on the test data, highlighting the power of CNNs in recognizing handwritten digits.

## **6. CIFAR-10 Model Implementation**

The CIFAR-10 dataset presented a more complex classification task, requiring a deeper CNN architecture to handle the greater variety of image categories.

### **a. Model for CIFAR-10 Dataset**

### **b. Model Training and Evaluation**

The CNN for CIFAR-10 was trained for 10 epochs. Due to the dataset's complexity, the model achieved a classification accuracy of 70%, showcasing its ability to differentiate between diverse objects like planes, cars, and animals.

## **7. Performance Evaluation**

**MNIST** **Accuracy**: 99.4%

**CIFAR-10** **Accuracy**: 70%

The performance of the CNN models on both datasets demonstrates their efficacy in feature extraction and image classification. The use of convolutional layers helped the model in recognizing important patterns in images, while the pooling layers ensured computational efficiency without losing essential information.

## **8. Conclusion**

This TAE-II activity allowed for hands-on experience with CNN architecture, providing an in-depth understanding of how these networks work for image classification tasks. By implementing models for MNIST and CIFAR-10 datasets, the project showed how CNNs can effectively learn and classify images from diverse datasets. Further enhancements, such as adding more layers or employing data augmentation, can improve the model's accuracy and generalization.