### NAME: Ashish Dnyaneshvar Parulekar

### STUDENT ID: 1017913240

### COURSE: Advanced Computer Vision (CPS-667)

### Assignment 2

### TITLE: CIFAR-100 Classification Using VGG16 with PyTorch

## Introduction

The objective of this project is to manually implement the VGG16 deep learning architecture using PyTorch and apply it to the CIFAR-100 dataset, which contains 100 image classes. The implementation includes building the network from scratch, training it on CIFAR-100, evaluating its performance, visualizing the training progress using loss and accuracy curves, and displaying sample predictions.

## Steps Implemented

### Step 1: Model Architecture (VGG16)

We manually built the VGG16 architecture, consisting of:

* **5 Convolutional Blocks:** Each block contains Conv2D layers (with increasing channels: 64, 128, 256, 512, 512) followed by Batch Normalization, ReLU activation, and MaxPooling layers for downsampling.
* **Fully Connected Layers:** A flattened layer followed by two FC layers with 4096 units each (including Dropout), and an output layer with 100 units (for CIFAR-100 classes).

The model is initialized using **Xavier Uniform initialization** to improve learning stability.

### Step 2: Dataset Preparation (CIFAR-100)

* The CIFAR-100 dataset is downloaded and loaded via **torchvision.datasets.CIFAR100**.
* **Training augmentations:** Random crop, horizontal flip, random rotation, and normalization.
* **Test transformations:** Resize to 128x128, followed by normalization.

### Step 3: Model Training Setup

* **Loss Function:** CrossEntropyLoss with label smoothing (0.1).
* **Optimizer:** AdamW with learning rate = 0.0001 and weight decay = 1e-4.
* **Scheduler:** Cosine Annealing LR scheduler with T\_max = 50 epochs.
* **AMP:** Mixed Precision Training using PyTorch’s torch.amp for faster training on CUDA.

Checkpoints are saved every 5 epochs to allow for training to resume seamlessly.

### Step 4: Training Progress (Loss & Accuracy Curves)

The model was trained for **71 epochs** (resumed from checkpoints after epoch 50). Below is the plot showing the loss and accuracy curves:

A graph with a line graph and a line graph

AI-generated content may be incorrect.

Figure 1: Loss Curve and Accuracy

### **Training Loss and Accuracy Curve Explanation**

The above plot shows the **Training Loss** and **Test Accuracy** curves over epochs **50 to 89** for the VGG16 model trained on the CIFAR-100 dataset.

* The **blue curve** represents the **training loss**, which shows a gradual and consistent decrease as the number of epochs increases. This indicates that the model is learning and fitting the training data over time.
* The **orange curve** shows the **test accuracy (%)**, which steadily improves and stabilizes around **65%** after approximately **65 epochs**. This suggests that the model has started generalizing well to unseen test data.

### **Key Insights:**

* From epoch **50 to 60**, both loss and accuracy show notable improvement.
* After **epoch 65**, test accuracy plateaus, indicating the model has reached a performance ceiling under the current settings (optimizer, augmentation, architecture).
* The training loss continues to decline slightly after the accuracy plateaus, which is expected in most deep learning models.
* **Training loss:** Reduced progressively from ~2.1 to ~1.1.
* **Test accuracy:** Increased and stabilized around **65.5%**.

### Step 5: Evaluation and Performance

* **Final Test Accuracy:** ~65.5% on the CIFAR-100 test set.
* The model was able to generalize reasonably well, considering a basic VGG16 implementation on CIFAR-100.

### Step 6: Sample Image Predictions

Below is an example of model predictions on a test image:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Output Image

A blurry image of a raccoon

AI-generated content may be incorrect.

Figure 3: Predicted Image 2

* **True Label:** raccoon
* **Predicted Label:** raccoon (correct)

A blurry image of a wooden chest of drawers

AI-generated content may be incorrect.

Figure 4: Predicted Image 3

* **True Label:** wardrobe
* **Predicted Label:** wardrobe (correct)

The model performs well on various samples from the CIFAR-100 dataset.

## Conclusion

This project successfully demonstrates a manual implementation of VGG16 using PyTorch on the CIFAR-100 dataset. The model achieves over **65% test accuracy** using proper data augmentation, regularization, mixed precision training, and cosine annealing scheduling. The process highlights best practices in training deep convolutional networks from scratch.