

# **CSC 421**

## **Introduction to AI**

### **Final Project**

**Topic: CIFAR 100 Image Classification using Deep Learning**

**GITHUB :** <https://github.com/abhishek-924/CIFAR100-Image-Classification>

**By:**

Abhishek Kathuria (V00959831)

Dhruvraj Singh (V00970352)

Anushka Halder (V00961967)

CIFAR-100 classification using various models such as Linear Classifier, Convolutional Neural Net (CNNs) from scratch, Residual Neural Nets (ResNet) from scratch, Pretrained ResNet-18 and Pretrained ResNet-50.

## Files

The usage of the following python files is given as follows:

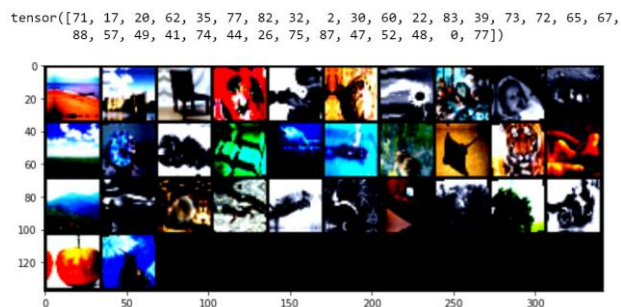
1. **Linear\_Classifier.pdf (BASIC)** - contains an Artificial Neural Net with two hidden layers.
2. **CNN.pdf (EXPECTED)** - contains a four layered CNN built from scratch.
3. **RESNET.pdf (EXPECTED)** - contains a custom ResNet with two residual blocks.
4. **PRETRAINED\_RESNET.pdf (ADVANCED)** - contains pretrained ResNet-18 and ResNet-50 architectures fine tuned on the given dataset.

## Grading Rubric:

### Basic (40%)

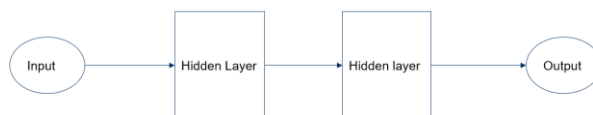
#### 1. EDA (Exploratory Data Analysis)

Included in Linear\_Classifier.pdf



#### 2. Linear Model (a simple ANN from scratch)

Included in Linear\_Classifier.pdf



Parameters:

- Loss function - CrossEntropyLoss()
- Optimizer- SGD

- Batches = 32

### 3. Find Max Learning Rate

We compared training loss and validation loss for each model to find the MAX learning Rate out of the following list of learning rates:

LR = [ 1e-1, 1e-2, 1e-3, 1e-4]

### 4. Different Optimizers

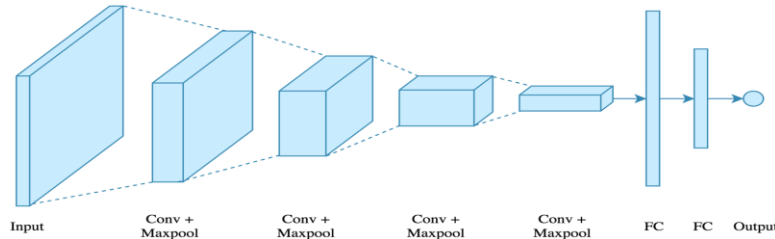
- Adam or Adaptive Moment Estimation
- SGD

## Expected (40%)

### 1. Data visualization

Result graphs, training accuracy vs validation accuracy graphs, training and validation loss graphs, graphs with and without data augmentation are all included in all the models

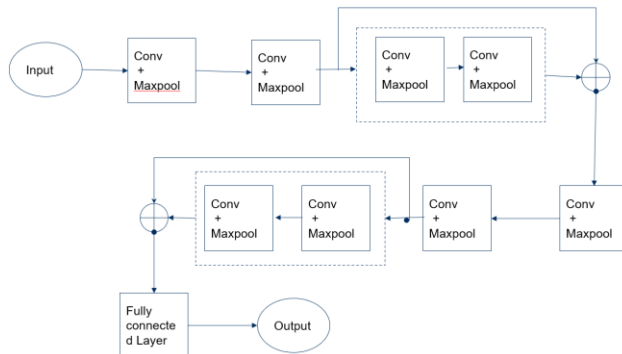
### 2. A CNN from scratch (4 Layers)



Code included in CNN.pdf

Number of CNN Layers – 4, Nodes in FC layer - 206 nodes, Filters - Different number of filters of values 16,32 and 64, and 128, Loss - Cross Entropy, Optimizer- SGD, Activation function used – Relu

### 3. Custom ResNet



Code included in RESNET.pdf

### 4. Compare model with and without Data Augmentation

All the graphs comparing accuracies with and without data augmentation are included in all the models separately. The data augmentation techniques used are Random Crop and Random Horizontal Flip.

## 5. Regularization

We have used two types of regularizations:

- L2 Regularization
- Dropout

## Advanced (20%)

### 1. Pretrained ResNet-18

Included in PRETRAINED\_RESNET.pdf

### 2. Pretrained ResNet-50

PRETRAINED\_RESNET.pdf

### 3. Learning Rate Scheduling

We used the following learning rate scheduling:

- OneCycleLR
- COSINE ANNEALINGLR

### 4. Result and analysis

Model Architecture	Optimizer	Learning Rate	Learning Rate Scheduler	Accuracy (in %)
Linear Classifier	SGD	1e-2	-	7.9
Small Conv Net	SGD	1e-1	OneCycleLR	33.4
Custom ResNet	Adam	1e-2	CosineAnnealingLR	55.7
Pretrained ResNet 18	Adam	1e-2	OneCycleLR	50.3
Pretrained ResNet 50	Adam	1e-2	OneCycleLR	54.7