ONTARIO TECH UNIVERSITY
FACULTY OF SCIENCE, COMPUTER SCIENCE

Project Group: 8 April 6, 2025

Computer Vision Course Project: Exploring Image Classification

by

Malhar Singh (100829423) Johnny Liang (100832977)

CSCI 4220U (Winter 2025) Computer Vision

Faisal Z. Qureshi

GitHub Project Link

1 Introduction

In this Project we plan to explore and analyze image classification. More specifically, we plan to test different machine learning models on the CIFAR-10 dataset (primarily) and slightly with STL-10 to see how different models perform for detecting and classifying images correctly and efficiently. These analyses include: dataset wide and class-wise accuracies (explained more in Background). After comparing accuracies for the primary dataset, we will explore how these trained models work on a new, but similar dataset STL-10.

2 Background

The objective of these analyses is to compute and compare the accuracies of different models trained for 10 and 20 epochs, whilst keeping the complexity of each model similar and only varying one variable at a time. The models analyzed include: Linear, MLP (Multi-Layer Perceptron), CNN (Convolutional Neural Network), CNN with Batch Normalization, ResNet, as well as more advanced models such as deep CNN and deep ResNet.

CIFAR-10 Classes:

- Plane
- Car
- Bird
- Cat
- Deer
- Dog
- Frog
- Horse
- Ship
- Truck

STL-10 Classes:

- Plane
- Car
- Bird
- Cat
- Deer
- Dog
- Monkey
- Horse
- Ship
- Truck

Information about the datasets: Both the datasets are collections of coloured images, 32x32 for CIFAR [4] and 96x96 for STL-10 [1]. Here we can see that most of the classes are the same in both datasets, except ”Monkey” and ”Frog”, which differ in both datasets.

3 Models Explored

We implemented and trained multiple models of varying complexity whilst keeping them as simple and similar as possible.

3.1 Linear Model

Simple network that takes an image of size H, W and channels C. Flattens it and feeds it into a fully connected layer that also acts as the output layer.

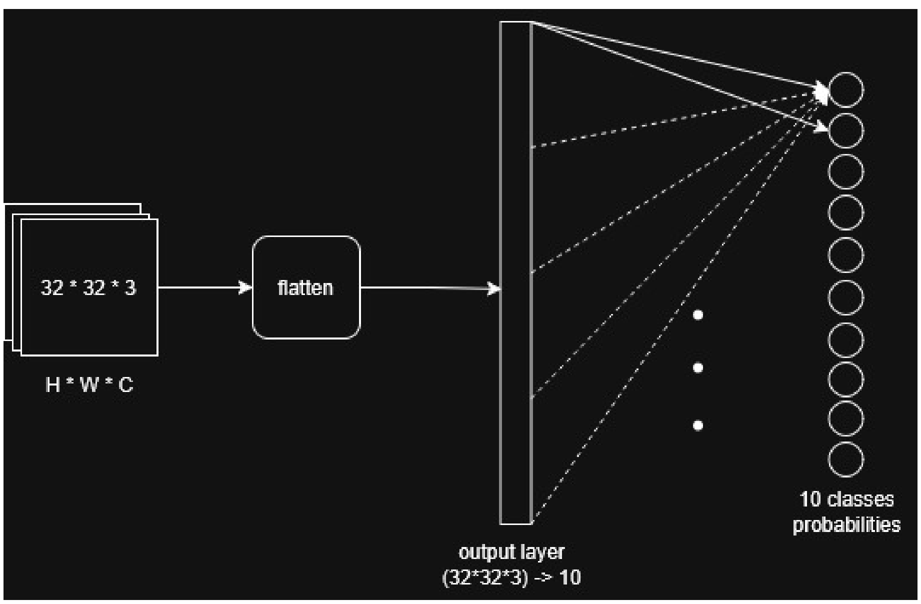

Figure 1: **Linear Model Architecture**

3.2 MLP Model

Network same as the linear model with the addition of a ReLu activation function and a linear layer.

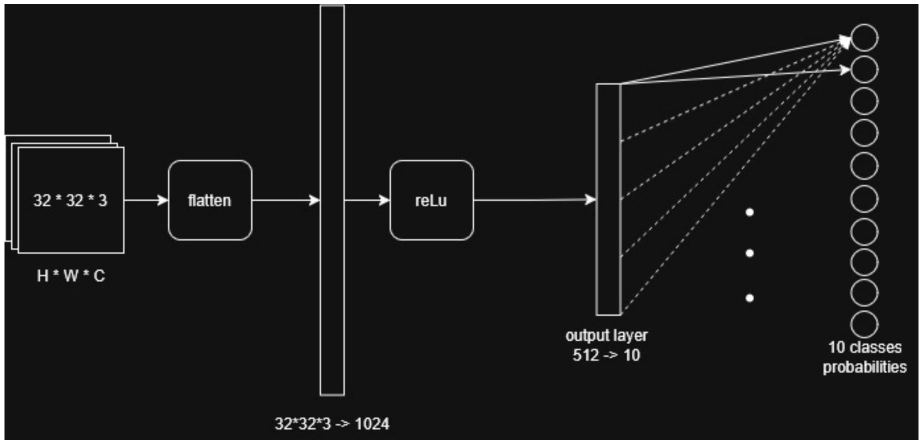

Figure 2: **MLP Architecture**

3.3 Convolutional Neural Network

The Convolutional Neural Network uses the mathematical operation Convolution to produce feature maps that capture patterns, shapes, or edges in an image, and the spatial coherence of the image. The implemented model has two convolutional blocks that include the convolutional operation, a ReLu activation, and a maxpool layer. The maximum number of channels in a feature map is 24.

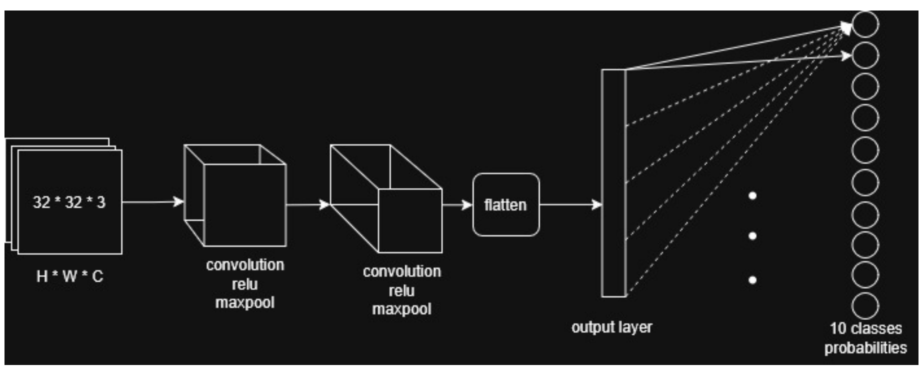

Figure 3: **CNN Architecture**

3.4 CNN with Batch Normalization

This Neural Network repeats the architecture of the ***Convolutional Neural Network*** 3.3, but implements Batch Normalization to speed up and normalize the training of our model [3].

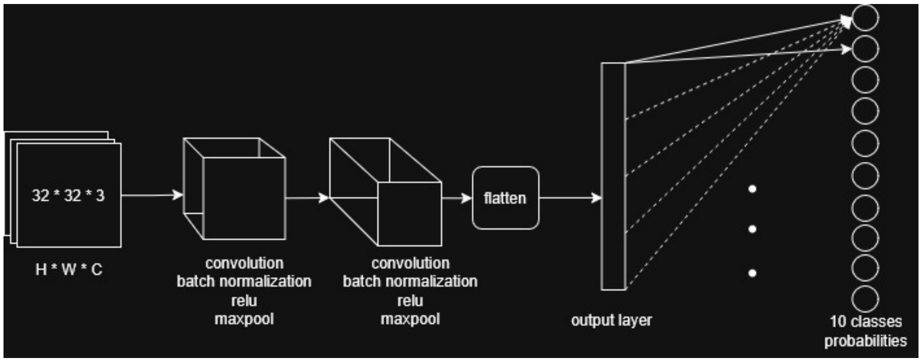

Figure 4: **CNN + Batch Normalization Architecture**

3.5 CNN with Residual Shortcut

This neural network is an iteration of the ***CNN + Batch Normalization Network*** 3.4 with the addition of a residual shortcut to simulate the main mechanic used in Deep Resnet models [2].

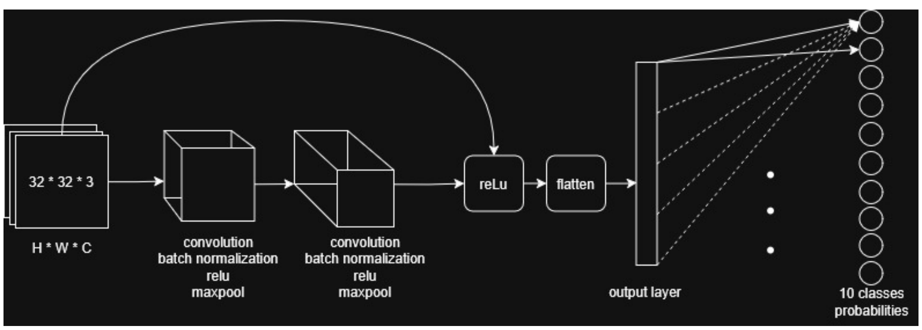

Figure 5: **CNN with Residual Shortcut Architecture**

3.6 Deeper Convolutional Neural Network

An improvement of the previously implemented convolutional neural networks. It uses 5 blocks of Convolution which includes a convolution layer, batch normalization layer, a ReLu activation, and fitfully, a maxpool layer. The feature maps produced have more channels, with the maximum number of channels in a feature map being 180. Additionally, we implement the use of dropouts to reduce overfitting by randomly dropping weights.

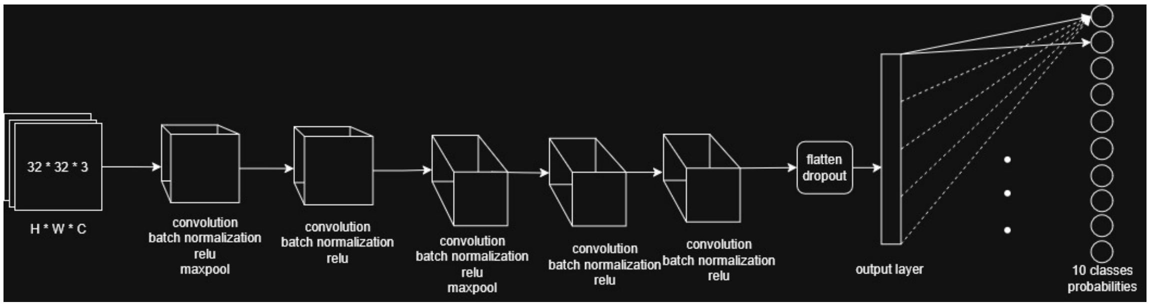

Figure 6: **Deep CNN Architecture**

3.7 Residual Neural Network

An iteration of our ***Deeper Convolutional Neural Network*** 3.6, keeping the same number of blocks of Convolution and channels per feature map, but it also implements shortcuts from a convolution block to another.

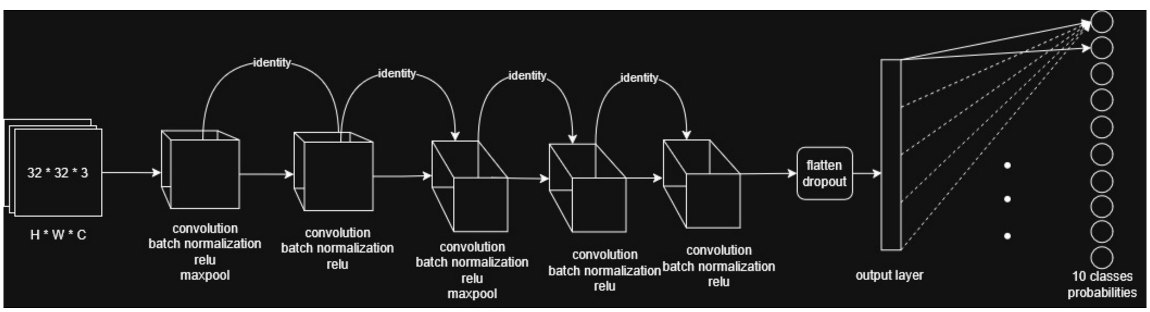

Figure 7: **Deep Resnet Architecture**

4 Results

We begin by examining the model accuracies (%) on the CIFAR-10 dataset, followed by a brief comparison with the STL-10 dataset.

4.1 CIFAR-10

Model	Accuracy (10 epochs)	Accuracy (20 epochs)
Linear	40.13	40.10
MLP (multi-layer perceptron)	52.31	56.22
CNN (convolutional neural network)	53.55	62.49
CNN using Batch Normalization	63.90	67.57
Resnet	62.39	66.75
Deep CNN	77.56	N/A
Deep Resnet	75.41	N/A

Figure 8: **CIFAR-10 - Dataset Accuracy (10 and 20 Epochs)**

From Figure 8, the general trend seems to be an increase of accuracy from 10 epochs to 20 epochs. Since we wanted to see the trend rather than exact values, we decided to exclude 20 epochs for deep models. This supports the idea that allowing the model more time to train helps it better refine its parameters. We also see that the more complex the model, the better the accuracy. Throughout this exploration, a consistent observation is that with the more complex models such as deep CNN and deep ResNet, performance tends to be more balanced and one does not completely outperform the other like for some simpler models.

Class	Linear	MLP	CNN	Batch CNN	Resnet		Deep Cnn	Deep Resnet
Plane	41.5	55.1	61.0	70.0	71.4		80.4	83.2
Car	43.6	69.5	73.2	86.3	81.6		82.9	91.5
Bird	28.1	39.7	27.7	39.6	48.7		62.5	70.2
Cat	27.2	32.7	30.1	32.1	29.2		55.4	59.1
Deer	31.6	39.8	38.3	50.3	49.1		78.2	68.0
Dog	30.7	38.9	47.0	58.9	45.9		62.3	66.9
Frog	47.2	62.8	69.3	82.7	89.4		92.8	81.9
Horse	43.8	59.8	64.1	76.5	69.1		78.8	84.7
Ship	59.4	67.2	59.2	73.5	74.3		90.8	87.9
Truck	48.2	57.6	65.6	69.1	65.2		91.5	80.7

Figure 9: **CIFAR-10 - Class-wise Accuracy (10 Epochs)**

Figure 9 shows us the class-wise accuracies for each model for the CIFAR-10 dataset for 10 epochs. For all subsequent tables: light gray represents the lowest accuracy for each class (row) across all models, while orange highlights the highest accuracy. Here, we observe a clear trend where the Linear model consistently underperforms compared to all other models, while deep CNN and deep ResNet achieve the highest accuracies, with deep ResNet slightly outperforming in certain classes. We also observe some inconsistencies with CNN, where it ended up having the lowest accuracy for classes Bird and Ship. Just goes to show, these models are constantly learning and not perfect in any way.

Class	Linear	MLP	CNN	Batch CNN	Resnet
Plane	53.5	62.6	68.0	75.9	73.3
Car	43.9	64.2	76.8	86.1	82.5
Bird	23.5	45.3	37.8	47.3	49.2
Cat	23.4	31.5	32.1	37.9	27.8
Deer	33.4	49.0	45.1	51.1	63.5
Dog	31.8	46.8	59.2	58.0	56.1
Frog	47.5	67.0	75.1	90.0	88.6
Horse	41.1	64.2	75.4	82.8	74.2
Ship	59.2	64.4	78.5	72.1	82.6
Truck	43.7	67.2	76.9	74.5	69.7

Figure 10: **CIFAR-10 - Class-wise Accuracy (20 Epochs)**

Similarly for 20 epochs, we can see similar performance patterns. We decided to exclude the deep models to try and get a better look at the simpler models. Similar trend: Linear still worst model. However, at higher epochs, Batch CNN and ResNet show more balanced performance overall. Interestingly, the standard CNN also achieves high accuracies in specific classes such as Dog and Truck.

4.2 CIFAR-10 Summary

All of these trends align with our understanding of spatial coherence. Since Linear and MLP models lack the ability to capture spatial relationships in images, they perform poorly, while CNN, Batch CNN, ResNet, and their deeper versions perform better by leveraging spatial relationships.

4.3 STL-10

Model	Accuracy (10 epochs)
Linear	18.41
MLP (multi-layer perceptron)	20.96
CNN (convolutional neural network)	18.01
CNN using Batch Normalization	24.99
Resnet	21.32
Deep CNN	27.61
Deep Resnet	29.62

Figure 11: **STL-10 - Dataset Accuracy (10 Epochs)**

Surprisingly, When testing these models (10 epochs) with the STL-10 dataset, we can see a heavy drop off of accuracies across all models. Where CNN is even slightly worse than MLP.

Class	Linear	MLP	CNN	Batch CNN	Resnet		Deep Cnn	Deep Resnet
Plane	30.0	42.6	35.9	57.1	53.6		57.0	74.6
Car	31.2	12.1	15.0	12.9	16.2		7.2	9.4
Bird	2.0	1.0	0.0	0.1	0.0		0.0	0.0
Cat	30.1	9.2	3.6	9.1	4.2		15.1	17.6
Deer	29.6	53.6	73.5	76.4	68.9		89.8	80.5
Dog	13.9	13.4	5.5	12.0	3.5		4.0	13.9
Monkey	2.4	4.8	2.5	1.1	0.6		0.6	0.6
Horse	11.8	15.2	13.2	3.8	8.4		5.1	1.4
Ship	21.9	46.4	25.6	64.8	51.4		71.6	71.2
Truck	11.2	11.2	5.2	12.6	6.4		25.6	27.0

Figure 12: **STL-10 - Class-wise Accuracy (10 Epochs)**

In Figure 12 it is very difficult to find a pattern. We can see that the max/min accuracies are very distributed. The lowest accuracies are now with the ”better” models, while the highest accuracies are with the ”worse” models. Also, we can see that bird, monkey and horse have really low accuracies across all models.

4.4 STL-10 Summary

These results are quite surprising and highlight the uncertainty of these models. There are several potential factors contributing to these inconsistencies, such as brightness, resolution, overfitting, underfitting, and more. This highlights the sensitivity of these models to even slight variations in images, emphasizing the need for careful fine-tuning to make sure the models work well across different datasets.

5 Conclusion/Summary

In conclusion, we observed that when models were trained and tested on the same dataset (CIFAR-10), the accuracies followed an order: Linear, MLP, CNN, CNN with Batch Normalization, ResNet, with deep CNN and deep ResNet models showing more balanced performance. We also discussed the uncertainty associated with models when working with different datasets. Specifically, models must be fine-tuned to handle diverse datasets that vary in properties such as brightness, resolution, skew, scale, and more.

References

[1] Adam Coates, Honglak Lee, and Andrew Y. Ng. Stl-10 dataset. https://cs.stanford.edu/~acoates/stl10/, 2025.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[4] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html, 2025.

[5] OpenAI. Chatgpt. https://chat.openai.com, 2025.

[6] Faisal Z. Qureshi. Class resources. https://csundergrad.science.uoit.ca/courses/csci4220u/latest/, 2025.