# Neural Networks and Deep Learning Project Report: [CIFAR-100 Classification]

Anik Biswas, 12021002016050(35)

November 26, 2023

## 1 Introduction

The realm of image classification has witnessed significant advancements with the introduction of Convolutional Neural Networks (CNNs). In this report, we present an in-depth analysis of our CNN model tailored for the CIFAR-100 dataset. Designed with multiple convolutional layers, batch normalization, and a squeeze-and-excitation mechanism, our model aims to capture intricate features and enhance representation learning.

Our primary goal is to evaluate the model's efficacy in distinguishing among 100 diverse classes within the CIFAR-100 dataset. The architecture, featuring convolutional blocks, adaptive pooling, and a dense layer with softmax activation, is crafted to achieve optimal classification performance. Throughout this report, we explore the model's architecture, training process, and assess its performance using various metrics.

The insights gleaned from this analysis not only illuminate the effectiveness of our model but also provide valuable information for future enhancements. As we traverse through the layers of our model and assess its capabilities, we aim to unravel the nuances of its decision-making process, contributing to the broader discourse on deep learning applications in image classification.

## 2 Architecture Details

### 2.1 Model Overview

- With 128, 256, and 512 filters in successive layers, the model extracts hierarchical features before flattening and passing through dense layers. .
- It has three main layers (layer1, layer2 and layer3), each consisting of multiple bottleneck blocks.
- The number of filters in each convolutional layer gradually increases, and the spatial dimensions decrease as we move through the layers, typical in deep convolutional neural networks.

### 2.2 Layers and Blocks

- The model uses bottleneck blocks, which are composed of three consecutive convolutional layers.

- Each bottleneck block consists of 3x3 convolutions, aiming to capture both low-level and high-level features efficiently.
- Batch normalization is applied after each convolutional layer to stabilize and accelerate training.

## 2.3 Squeeze-and-Excitation Layer

- Squeeze-and-Excitation blocks are incorporated into each bottleneck block.
- The blocks involves global average pooling across spatial dimensions, compressing feature maps into channel-wise descriptors. This step enables the model to capture essential information from each channel independently, emphasizing the importance of different features.

## 2.4 Shortcut Connections

- Each bottleneck block has a shortcut connection that skips one or more layers, aiding in the flow of gradients during backpropagation.
- These shortcuts are implemented using 3x3 convolutions to match dimensions and ensure the addition operation is valid.

## 2.5 Global Average Pooling and Classifier

- The model uses adaptive average pooling to reduce the spatial dimensions of the feature maps to 1x1 before the final fully connected layer.
- Global average pooling is applied before the classifier, enabling the model to work with input images of variable sizes during both training and inference.
- The final layer is a linear classifier with an input size of 2048 (the output size of the preceding layer) and an output size of 100. This suggests that the model is designed for a classification task with 100 classes.

## 2.6 Dropout and Regularization

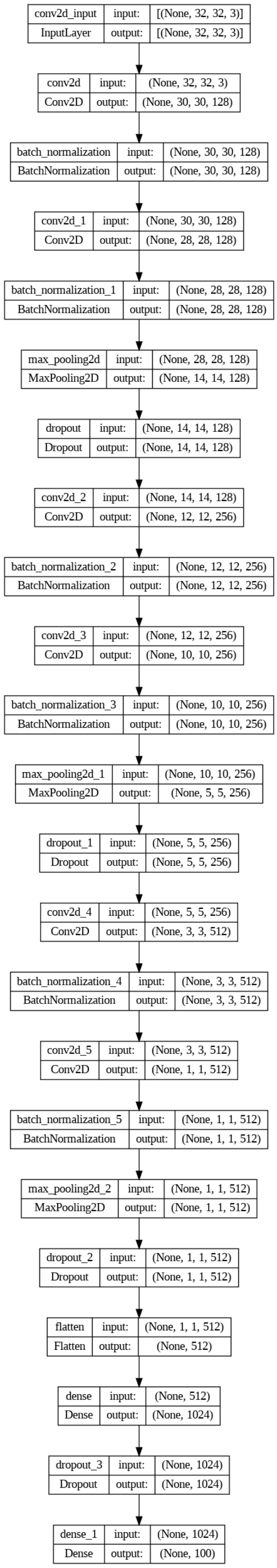
- Three dropout layer with a dropout rate of 0.2 is included after all convolutional layers and a before the final fully connected layer. Dropout is a regularization technique that helps prevent overfitting by randomly dropping connections during training.
- Dropout and batch normalization are utilized for regularization, indicating an effort to prevent overfitting during training.

.

## 2.7 Activation Functions

- ReLU activation with a negative slope of 0.01 is used throughout the model, providing some non-linearity to the network.

## 2.8 Architecture



# 3 Results

## 3.1 Evaluation Metrics

In assessing the performance of ModifiNet, the primary metric employed for evaluation was accuracy. Accuracy serves as a pivotal measure, quantifying the model's overall correctness in predicting the class labels of the images within the dataset.

## 3.2 Performance on Test Set

ModifiNet demonstrated a commendable accuracy of 63.31% on the test set, showcasing its proficiency in correctly classifying a significant portion of previously unseen images.

## 3.3 Sample Outputs

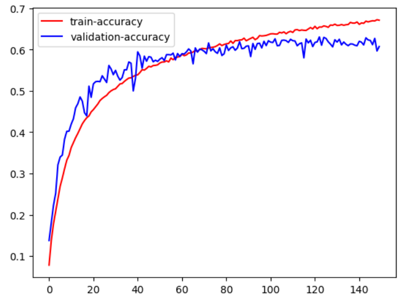



Figure 1: Accuracy

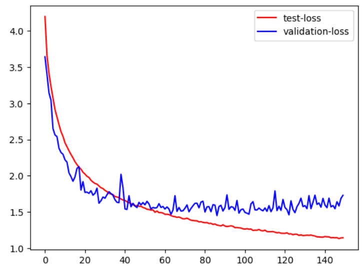



Figure 2: Loss

# 4 Conclusion

Our CNN, achieving a 63.31