Image classification using Deep learning on CIFAR-100 Dataset

Kazemi Mina | Sarpotdar Pranit | Patidar Shubham IE7615-NeuralNetworks and Deep Learning

Northeastern University, College of Engineering, Boston, MA, USA

# Abstract

Two groups from NYU and MIT put together a dataset of tiny images. In April 2009 Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton published a paper on classifying tiny images into different class (later subclass) and tried to prove that object recognition is significantly improved by pre-training a layer of features on a large set of unlabeled tiny images. We took forward the same task of classifying the images leveraging better processing speed hence improving accuracy and processing time.

# Introduction

The tiny images dataset on which we based all of our experiments was collected by colleagues at MIT and NYU over the span of six months; A rather revolutionary step was taken in the field of Deep Learning while classifying the images of image dataset by Alex Krizhevsky in His research paper *Learning Multiple Layers of Features from Tiny Images.* He introduced Deep Belief Networks to classify images. We took a step forward in this project and tried to label images using ResNet.

We used Residual Network (ResNet) and is a specific type of convolutional neural network (CNN) introduced in the 2015 paper “Deep Residual Learning for Image Recognition” by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. ResNet-50 is a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer). Residual neural networks are a type of artificial neural network (ANN) that forms networks by stacking residual blocks.

ResNet has many variants that run on the same concept but have different numbers of layers. Resnet50 is used to denote the variant that can work with 50 neural network layers. This model was immensely successful, as can be ascertained from the fact that its ensemble won the top position at the ILSVRC 2015 classification competition with an error of only 3.57%. Additionally, it also came first in the ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation in the ILSVRC & COCO competitions of 2015

We then combined the two extremely genius ideas and tried to label original image dataset by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton using ResNet, introduced by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian.

Overall performance was extremely worth noticing. Accuracy of the model was improved multiple folds with processing time much lower than the original work done by Alex Krizhevsky.

The task is to use image detection technique to classify images to the appropriate class and superclass based on identifying objects within the image. There is total 60000 images of pixel 32\*32 from the CFIAR dataset under the labeling of class and super class respectively. Images are divided into 100 classes.

We used stochastic gradient decent as the optimizing function and trained data on 50 epochs. The model shows interesting but slight predictable traits during whole learning process. Details of labelling is illustrated in the forward pages with final image proving the successful labelling with accuracy of around 72%.

# Background

The current state of the art neural networks models that are used to classify an image to respective classes are Big Transfer (BiT): General Visual Learning (Dec-2019) and EfficientNet (May 2019) with accuracy of 93.51% and 91.70%, respectively. We are considering reaching the base model performance with Representation maximum possible classification accuracy and eventually try to reach state of the art performance.

A second problematic aspect of the tiny images dataset is that there are no reliable class labels which makes it hard to use for object recognition experiments. We created two sets of reliable labels. The CIFAR-10 set has 6000 examples of each of

10 classes and the CIFAR-100 set has 600 examples of each of 100 non-overlapping classes. Using these labels, we show that object recognition is significantly improved by pre-training a layer of features on a large set of unlabeled tiny images.

Most of the previous work has been updated by a lot of different academics and scholars and hence the accuracy and processing time (Epochs) has been a huge matter of debate. Most of the models seem to have shown exceptional results but advancements in the technology and so many CNN models always leave some room for more improvement in a quest to reach a level all the labelling can be done with very low processing time and with highest accuracy.

We used ResNet model of CNN to increase the labelling accuracy as much as possible with the minimum computational time. Even though the model still shows some misclassification on the end results, Our model did an excellent job on labelling the images to their respective categories.

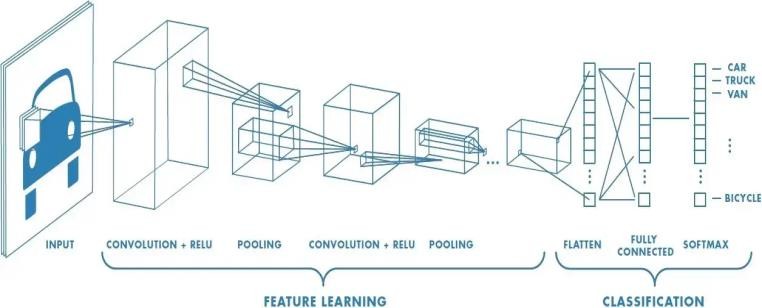
Even though a completely new task has been performed in the project, inspiration is taken from Alex Krizhevsky paper *Learning Multiple Layers of Features from Tiny Images, April 2009.*

# Approach

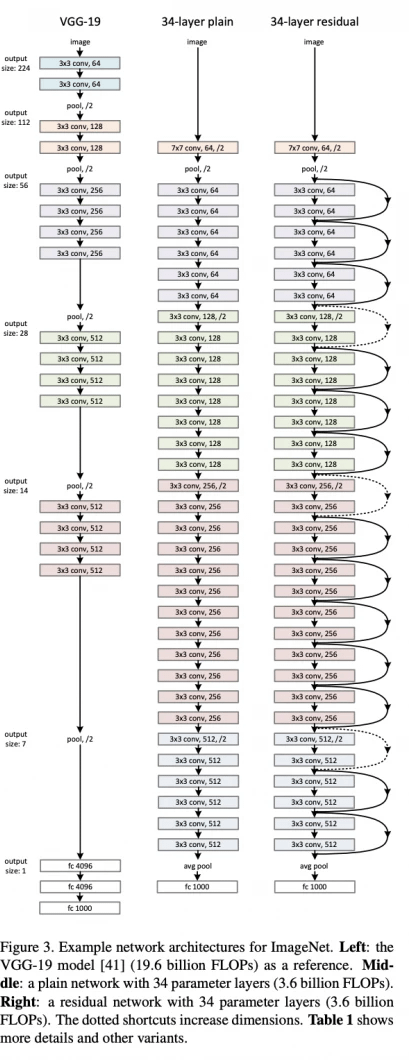
CIFAR-100 is a very popular dataset first put together by Alex Krizhevsky and the team. Since then numerous scholars, academics and even student teams tired to label these images in most innovative ways possible. There has been numerous examples of good accuracy but slightly more processing time or easy computing but compromising on the accuracy.

We have limited the methods used for classification. In particular, we have used CNNs , ResNets architecture to train the model on the CIFAR-100 dataset. Hyperparameters were decided after trial and error methods and as for parameters they were assigned by the model during training phase.

A **Convolutional Neural Network** (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

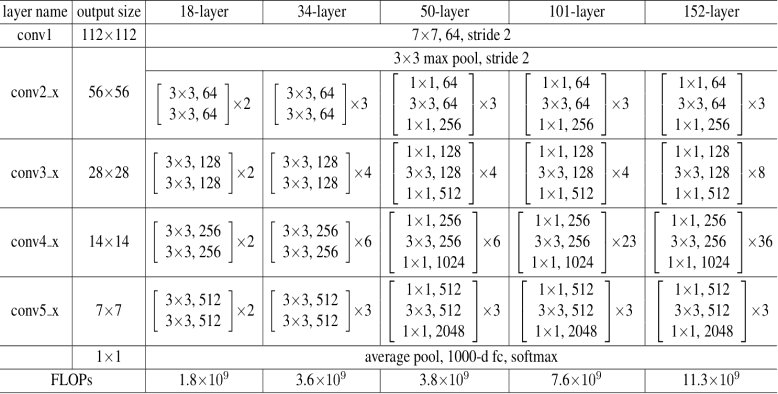
The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

**ResNet50**:

The original *ResNet* architecture was ResNet-34, which comprised 34 weighted layers. It provided a novel way to add more convolutional layers to a CNN, without running into the vanishing gradient problem, using the concept of shortcut connections. A shortcut connection “skips over” some layers, converting a regular network to a residual network.

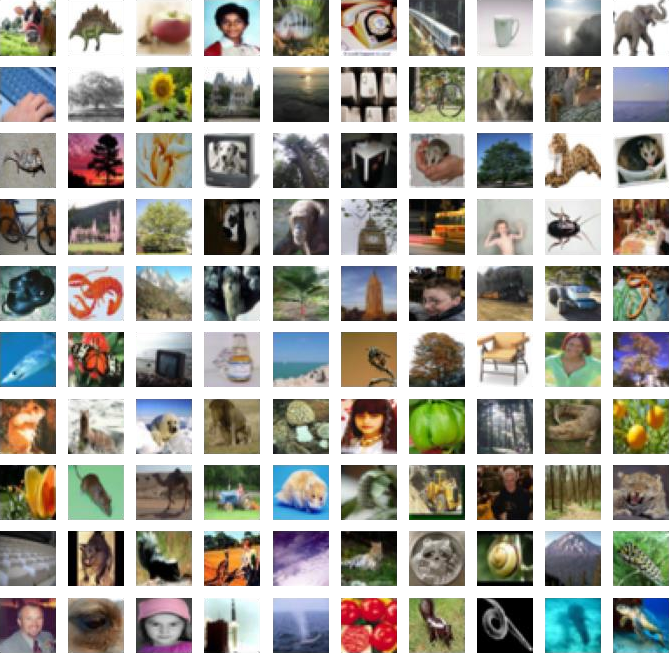
ResNet-50 has an architecture based on the model depicted above, but with one important difference. The 50-layer ResNet uses a bottleneck design for the building block. A bottleneck residual block uses 1×1 convolutions, known as a “bottleneck”, which reduces the number of parameters and matrix multiplications. This enables much faster training of each layer. It uses a stack of three layers rather than two layers. We have limited the methods used for classification.

The 50-layer ResNet architecture includes the following elements, as shown in the table below:



The CIFAR-100 dataset consists of 60000 32x32 color images in 100 classes, with 6000 images per class. There are 40000 training images and 10000 validation images and 10000 test images.

A sample of the images in shown below:



One hot encoding is performed on all the categorical inputs

and normalization of the pixels is performed by dividing each set with 255.

The dataset is spited into training, validation and t est sets by [(40000, 32, 32, 3) (10000, 32, 32, 3) (1

0000, 32, 32, 3)] for input and (40000, 100) (1000

0, 100) (10000, 100) for target variable.

In the next step an image augmentation is using im age Data generator function from Keras.

**Image augmentation** artificially creates training i mages through different ways of processing or co mbination of multiple processing, such as random rotation, shifts, shear and flips, etc.

Reduce learning rate when a metric has stopped im proving. Models often benefit from reducing the le arning rate by a factor of 2-10 once learning stagn ates. This scheduler reads a metrics quantity and if no improvement is seen for a ‘patience’ number of epochs, the learning rate is reduced.

In the given dataset patience (number of epochs before which model stops training) is set to be 3, After which model will automatically reduce the learning rate by the factor of 0.6.

One of the most challenging parts was to train the dataset on ResNet50. Since The Cifar images are of the shape 32,32,3 and resnet model is trained on images of 224,224,3. To deal with the challenge we used Keras 2D upsampling method to upscale our image dataset to fit in the ResNet model.

As for the optimizer we used gradient decent since it has been proven widely successful with Neural Network because of its property of descending quickly towards optimized solution and also not overfitting the model.

Gradient Descent Algorithm iteratively calculates the next point using gradient at the current position, scales it (by a learning rate) and subtracts obtained value from the current position (makes a step). It subtracts the value because we want to minimize the function (to maximize it would be adding). This process can be written as:



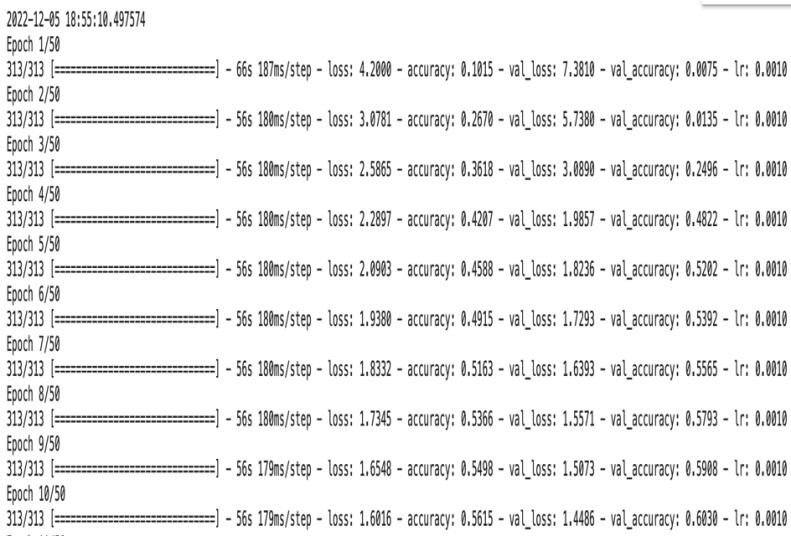
parameter η which scales the gradient and thus controls the step size. In machine learning, it is called learning rate and have a strong influence on performance.

# Results

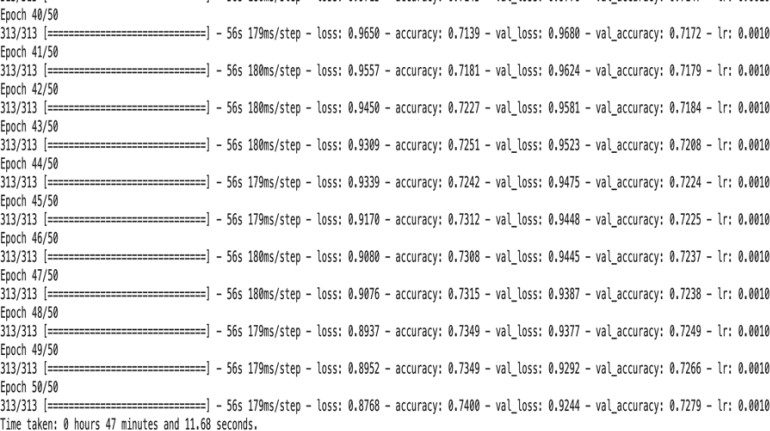
This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 super classes. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).

The dataset is further divided into training, validation and testing sets. The training set consists of 40000 images as training, 10000 images as validation and rest of the 10000 images as test sets. Total time taken by the model is 46 minutes 11.68 seconds to process 50 epochs. Below is the complete detail of accuracy of first 10 and last 10 epochs:

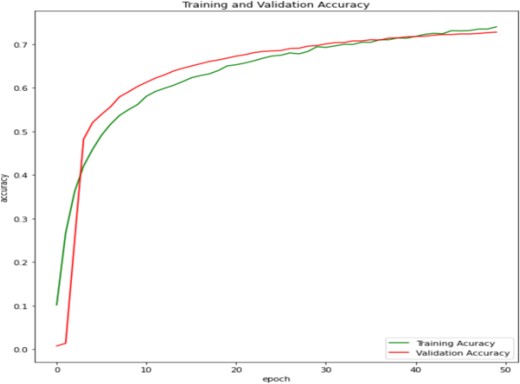
Fist 10 epochs are as follows:



We can notice sudden decrease in the loss for the first 20 epochs and later the loss decreases gradually. Mainly because Stochastic gradient decent performs worse when they come closer to optimized point.



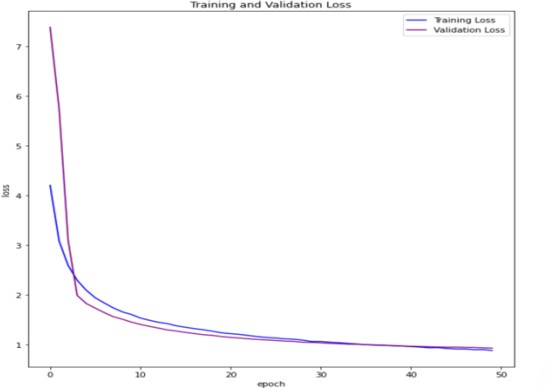
increases.



*Accuracy graph (training and validation set)*

We can see that the accuracy of training and validation graph starts to rise from somewhere around 10% and keeps on increasing with number of epochs. But once the accuracy reaches more than 50%, the increment of the accuracy starts to slow down and is almost constant after 50 epochs.

Also in the Loss function



*Loss function graph (training and validation set)*

There is sudden and then gradual drop in loss function mainly because of the stochastic gradient decent optimization function.

The medium accuracy, recall and precision details of the model are summarized in the below table:

Text  Description automatically generated with medium confidence

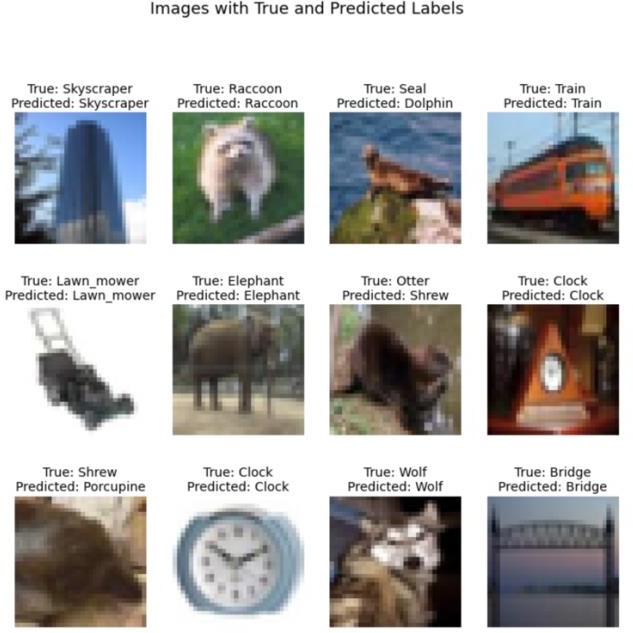
# Conclusion

The results obtained from the following project were promising, and the performance seems to be efficient in using neural networks for classification.

This paper was mainly focused on

using the basic CNN model and Res-net without pre-trained them on large datasets. The model was successful in labeling images with an accuracy of around 72%. We can see that some of the images are misclassified and there is still room for better accuracy. In the future developments on the project, accuracy and processing time will always have room for improvement, that being said, the model still is performing good and can be used in real life image detection successfully.

Below is the sample of classified images:



# References

Alex Krizhevsky, Learning Multiple Layers of Features from Tiny Images, April 2009: [https://www.cs.toronto.edu/~kriz/learning-](https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf) [features-2009-TR.pdf](https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf)

CIFAR-10 and CIFAR-100 dataset (a sample collection by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton: <https://www.cs.toronto.edu/~kriz/cifar.html>

ResNet-50: The Basics and a Quick Tutorial:

<https://datagen.tech/guides/computer-vision/resnet-50/>

A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way by Sumit Saha:

[https://towardsdatascience.com/a-comprehensive-guide-to-](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53) [convolutional-neural-networks-the-eli5-way-3bd2b1164a53](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)