Nutrition from detected food classes : CS 7643

Alek Francescangeli*, Sohaib Ishtiaq Malik*, Hoang Xuan Phu*, Rachael Rho*
Georgia Institute of Technology
{alek1,smalik,phx3,rrho3}@gatech.edu

Abstract

In this project we aim to address and discuss the ability to estimate nutritional value from food images. We created and assessed several model(s) that can make predictions on food information based on the Fruits360 image dataset and transposing to USDA FoodData information. Ultimately, we found that fine-tuned, pre-trained ShuffleNet model performed the best at identifying the underlying fruit images with 99.8 percent train and 95.7 percent test accuracy. Finally, we discuss additional areas for further research and exploration regarding estimating nutritional value.

1. Introduction/Background/Motivation

We are aiming to estimate nutritional value from food images, specifically fruit and vegetable classes then mapped to nutritional information. The subproblems to our project is as follows:

- Detecting the type of food from an image across multiple deep learning methods
- Getting the nutritional value of that food (in a standardized form)

There are a couple of key papers in which the research teams have been able to address different aspects of the problem we are trying to solve. In "Implementation of Fruits Recognition Classifier using Convolutional Neural Network Algorithm for Observation of Accuracies for Various Hidden Layers," [5], the team used Deep Convolutional Neural Networks to create a fruit class recognition system. While this resulted in an impressive 100 percent test and 99.8 percent training accuracy, the output is somewhat limited since it does not go the additional step to connect the classification results to nutritional information. Going further, the paper "Food calorie measurement using deep learning neural network," [4] aimed to both classify food images and estimate subsequent caloric content. This proposed calorie measurement system is less limited in its design, but still does not address the greater need for nutritional info beyond just caloric content.

These previous limitations lead to our next point as to why this project matters and the difference it could potentially make in individuals' health outcomes. Given we are aiming to detect nutritional information based on the image of a fruit or vegetable, users of this information would be able to make optimal food choices based on their day-to-day nutritional needs. For instance, a beet which is high in folic acid is known to be essential for pregnant women, but not so much for elderly men. The Fruits 360 dataset contains 90k images of fruits and vegetables with 131 class labels [2]. While there are different varieties of technically the same fruit (i.e. Golden vs. Granny Smith apple), they are considered to be separate classes in this original dataset. The images were created by each item being placed on the shaft of a low-speed motor (3 rpm) and a Logitech camera recorded a 20-second video. Therefore, each fruit and vegetable class has multiple images taken from various angles of the item at hand. Additionally, a flood-fill algorithm was used to preprocess the images, extracting the fruit from backgrounds of non-uniform lighting conditions. A sample of the Fruit360 image classes and angles are shown in Figure 1. Finally, the dataset is already divided into training and test sets, with training at 68k and test at 23k images, the latter taken from a Nexus 5x phone.

The second data set we have used was the FoodData Central database provided by the United States Department of Agriculture (USDA) via a REST API [3]. The FoodData Central API provides a "Food Details" endpoint, that gives nutritional information based on the food item sent in the request.

2. Approach

Our initial approach to this project was to experiment with developing different fully connected, feed-forward Neural Networks in order to evaluate how well that these networks could identify fruits and vegetables in the Fruits 360 dataset that we selected to test on. When developing these Neural Networks we first built off of vanilla networks focusing on using linear layers and activation functions such as ReLU. Once this was established further work was done on the model to allow for pooling of layers along

Figure 1. Sample batch (size=128) of fruit classes across various camera angles

with dropout regularization in order to reduce over fitting of the model. Tuning was done on the learning rate, dropout, and number of epochs in order to optimize the accuracy of the models. In addition to these models we tested the accuracy of four existing models: SqueezeNet, ShuffleNet, MnasNEt, and MobileNet. These network comparisons allowed us to compare what kind of architecture would work best with our dataset.

At least for the fully connected Neural Network, we anticipated a wide range of performance given the inherent limitations of model design as well as its potential sensitivity to hyperparameter settings. This hypothesis was correct given that the first few models tried did not perform too well until the learning parameter was tuned, which will be further discussed in the Experiment and Results section.

The next major component to solve this problem was the ability to relate these images that we trained on to dietary guidelines based on a person's age and gender. This was done by condensing dietary guidelines into a readable file. At that point it could be integrated into our system so that the model could identify the value of each fruit or vegetable according to an individuals specific nutritional information.

Some issues that we ran into were difficulties in data loading into our first choice of tool, Google CoLab. To combat this we used individual project spaces in order to test each model correctly. Another issue that came up during our experiments was the problem of properly categorizing the data in our dataset. The solution was to aggregate the 113 subclasses found in our dataset into 67 classes that described each type of fruit.

The Github repository for our final project is linked here.

3. Experiments and Results

The core of our experiments is transfer learning: we want to take advantage of proven models that are already trained with large datasets. To this end we selected 4 models that focus on keeping the network small while still achieving decent performance: SqueezeNet, ShuffleNet, MnasNet, MobileNet. This fits very well with our goal, because the small models are usually less demanding computationally, in case we want to use our nutrition analyzer on a mobile phone. We theorizes that performance won't be a problem because our dataset is much smaller than ImageNet, and this turned out to be correct.

Additionally, we also select 4 larger models with state of the art ImageNet performance: ResNet, GoogLeNet, ResNext, Wide ResNet. The idea is to compare and contrast them against the smaller models. We found that they perform reasonably well in the training set given some hyper parameter tuning. However they take a lot more resources to train, as expected, and their performance in the test set is questionable. These large models may be too powerful for the data that we have. Adding regularization techniques like dropout and L1/L2 norms may help with this generalization.

As mentioned in the approach, we also created a fully-connected, feed forward Network outside of the pre-trained models mentioned above. Varying the learning rates, this gave us a baseline on basic non-convolution network performance across this parameter. We also created a Convolutional Neural Network from scratch as also another baseline reference for the pre-trained models.

Overall, the goal is to train a model to classify an image of a fruit with maximum accuracy. Our original plan is to first try to use the pretrained models as a fixed feature extractor, and then to try finetuning the whole model.

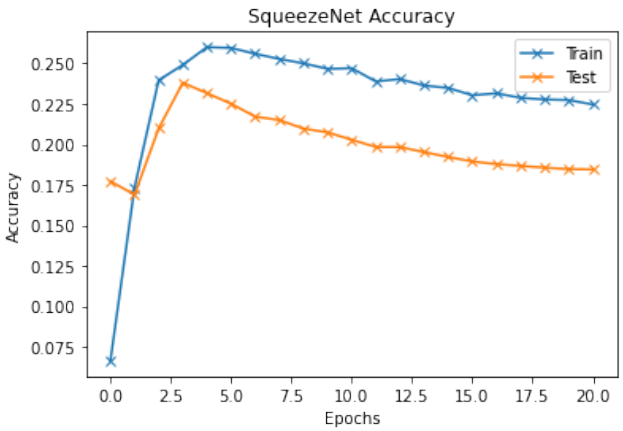
This way we only need to train the last classification layer. The expectation is that it should be faster to train using the feature extractor given we only need to care about the classification layer. And we can see how much performance improves when switching to training the full network. On the off chance that performance is not good enough, we planned to use larger networks such as ResNeXt as well.

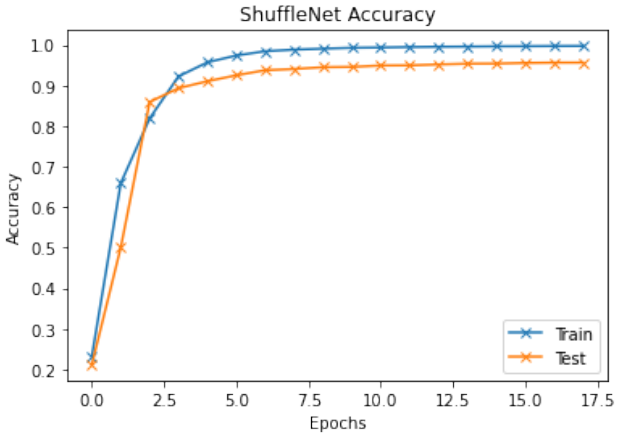
In reality, we found that just training the classification layer achieves very strong accuracy, that there is not much room for improvement by training the full network at all. In the best case, the network based on ShuffleNet achieves accuracy above 99 percent in the train set, and above 95 percent in the test set.

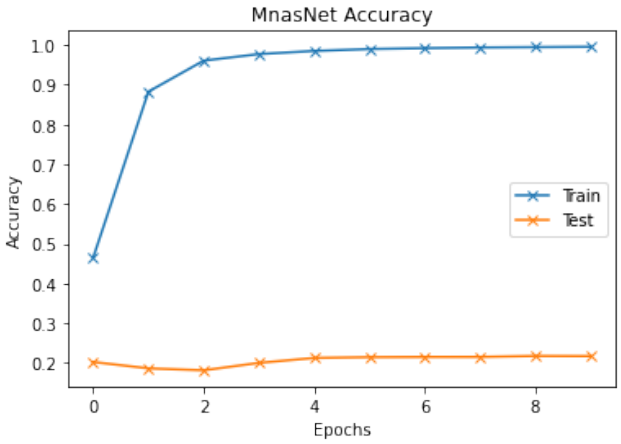
Note that in the base, fully-connected network results, that the higher learning rate of l = 0.05 (Fully-Connected in Table 1) performed better than l = 0.02 (Fully-Connected 1 in Table 1) and 0.001 (Fully-Connected 2 in Table 1), seemingly due to a complete lack of convergence for both training and test for at least the latter learning rate.

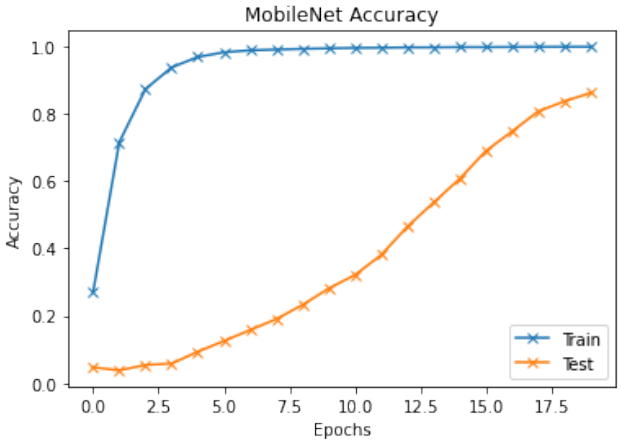
Beside the numbers, we found that SqueezeNet is really unstable during training. There were some runs with very high accuracy, but on average its performance has been abysmal with our dataset, even on training data. We also notice that its loss keeps decreasing monotonically, but its accuracy peaks pretty early during training and then slowly decreases over time. One possible explanation is its classifier involves a convolution layer, a non linear activation, and an adaptive pooling layer. This makes it the most complex classifier in the experiment, and thus may require more hyper parameter tuning or training time. We may test this hypothesis by changing this classifier to a simple fully connected layer.

Overall, we find success in ShuffleNet. It has very high accuracy in the training set, only beaten narrowly by MobileNet. However its performance in the test set closely follows its training performance, which we interpret as more reliable.



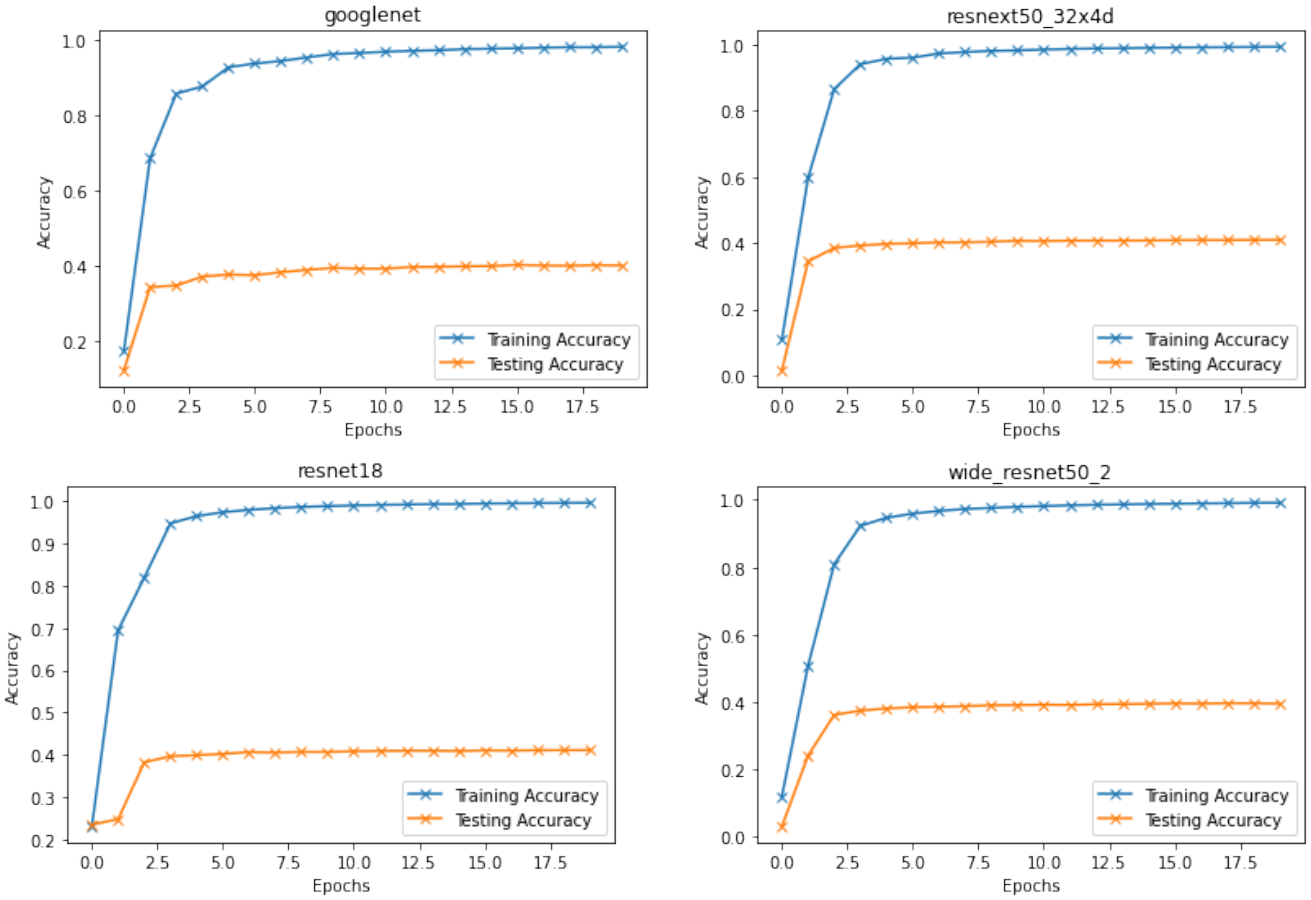






Model	Params	Train Accuracy	Test Accuracy
Fully-Connected	9068868	99.92%	94.42%
Fully-Connected1	9068868	98.65%	93.45%
Fully-Connected2	9068868	33.78%	71.03%
Convolutional	1525568	99.82%	94.94%
SqueezeNet	1235496	22.46%	18.45%
ShuffleNet	1366792	99.76%	95.66%
MnasNet	2218512	99.58%	21.71%
MobileNet	2542856	99.79%	86.12%
ResNet	11689512	99.54%	41.15%
GoogLeNet	13004888	98.29%	40.23%
ResNext	25028904	99.43%	40.96%
Wide ResNet	68883240	99.14%	39.57%

Table 1. Model Results



3.1. Getting Nutritional Data

Once we were able to put together a reliable model to predict the class of fruit, the next step was to return the nutritional information of a particular class, similar to Figure 2. We brainstormed multiple ways of doing this. We initially thought of training a separate model for this, but ultimately felt that given the FoodData Central Database

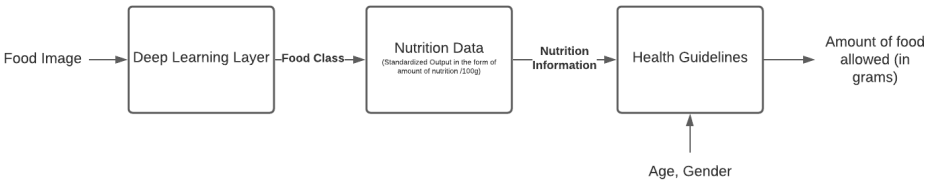


Figure 2. Food image to nutrition diagram

(provided by the US Department of Agriculture) had a very extensive database of nutritional information about foods, it was better for us to simply lookup the information in the database rather than over-engineering the problem. We therefore used the FoodData Central Database to fetch and store nutritional information for all the classes of data that we had. We stored this data in the form of a CSV file that was then read into our code.

3.2. Dietary guidelines

As a stretch goal to the project we intended to complete the following:

- Stretch Goal 1: Estimating the quantity of the food that a particular person should be having given their information (age, gender, weight, etc.)

To get this information together, we used the USDA Dietary guidelines to build a Dataset that allowed us to query daily nutritional requirements for a given age and gender. [1]

3.3. Calculating quantity

To calculate the quantity of food we implemented the following algorithm:

```
def getMaxAllowed(person_guidelines, food_nutrients):
    allowed = []
    for nutrient, allowed_value in person_guidelines.items():
            if nutrient in food_nutrients:
                allowed.append(allowed_value/food_nutrients[nutrient])
    return min(allowed)*100
```

The algorithm looks at each nutrient in the person's daily guidelines, and calculates the quantity of food based on that specific nutrient. At the end it takes a min of all of the quantities (across all nutrients) and returns it.

4. Further Research

There are several further areas of research that were both mentioned in the previous section (i.e. nutrition-related dimension) and listed below. Note that of these points were a part of our strech goals in the original project proposal.

- Detect multiple fruits per image
- Detect fruits' mass and volume (i.e. grams)
- Detect other food classes
- Predict nutrition information directly, rather than fruit classes
- Recommend foods based on individuals' nutritional needs

5. Work Division

Details of each team members' contributions are in Table 2.

Student Name	Contributed Aspects	Details
Alek Francescangeli	Implementation and Analysis	Convolutional Neural Network from scratch, see dl_project/alek folder in GitHub
Hoang Xuan Phu	Data Creation, Implementation and Analysis	Finetuned small pretrained models (SqueezeNet, MobileNet, ShuffleNet, MnasNet), Nutritional Data extraction and ingestion, see dl_project/phu folder in GitHub
Sohaib Ishtiaq Malik	Data Creation, Implementation and Analysis	Finetuned large pre-trained models (ResNet, GoogLeNet, ResNext, Wide ResNet), Extracting aggregate fruit classes, Calculating food quantity based on USDA guidelines, see dl_project/Sohaib folder in GitHub
Rachael Rho	Implementation and Analysis	Fully connected Neural Network from scratch, see dl_project/rho folder in GitHub

Table 2. Contributions of team members.

References

[1] DietaryGuidelines.gov. Dietary guidelines for americans 2020-2025. https://www.dietaryguidelines.gov/sites/default/files/2021-03/Dietary_Guidelines_for_Americans-2020-2025.pdf/, 2020. Dietary Guidelines for Americans 2020-2025. 5

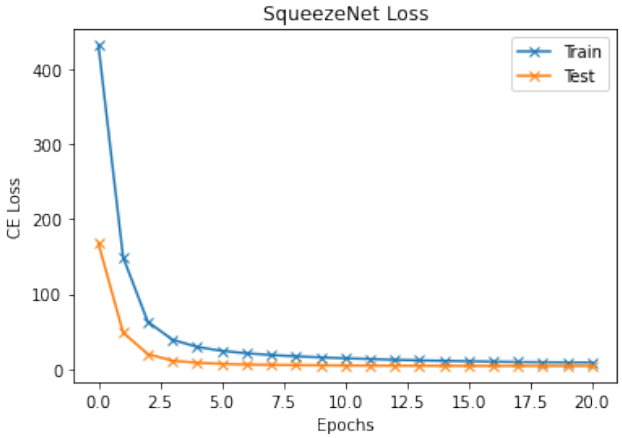
[2] Mihai Oltean Horea Muresan. Fruits 360 dataset, 2018. Fruits 360 dataset: A dataset of images containing fruits and vegetables. 1

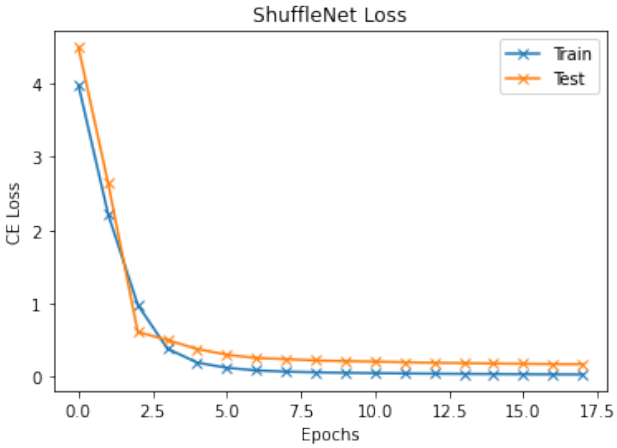
[3] United States Department of Agriculture. Food data central., 2021. Food Data Central Nutrition API. 1

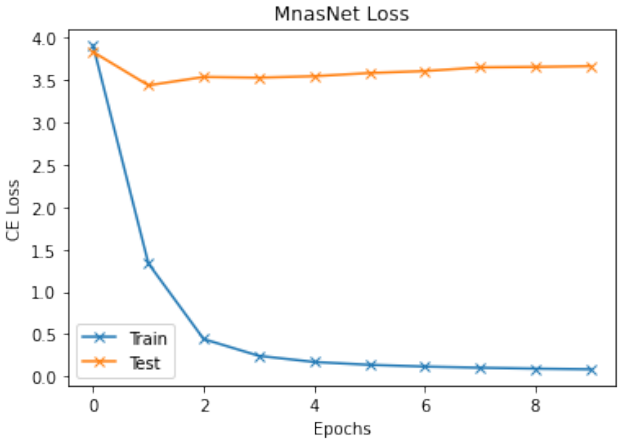
[4] Parisa et al. Pouladzadeh. Food calorie measurement using deep learning neural network. 2016. 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings. IEEE,. 1

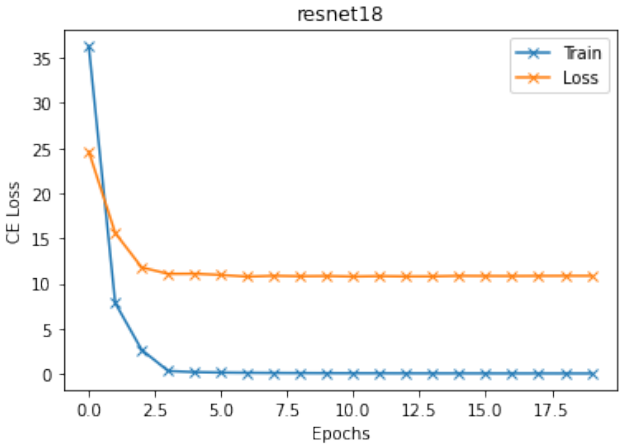
[5] Shadman et al. Sakib. Frobnication tutorial. 2019. Implementation of Fruits Recognition Classifier using Convolutional Neural Network Algorithm for Observation of Accuracies for Various Hidden Layers. 1

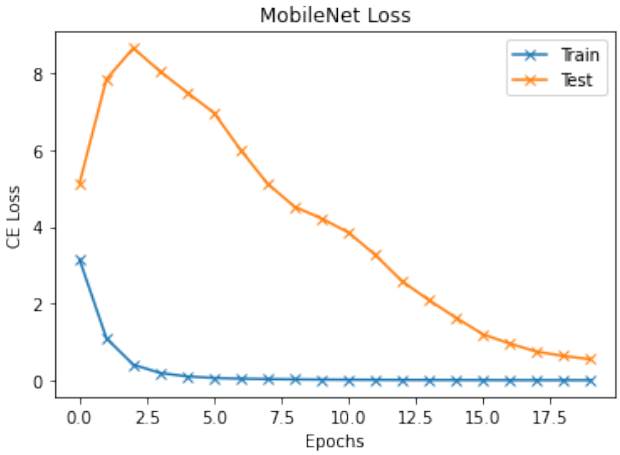
6. Additional Loss Charts

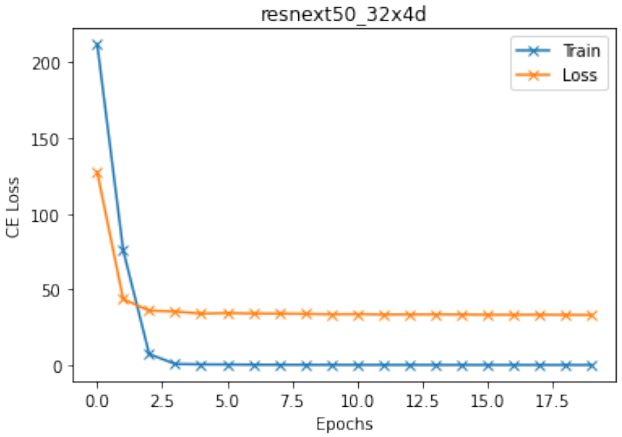


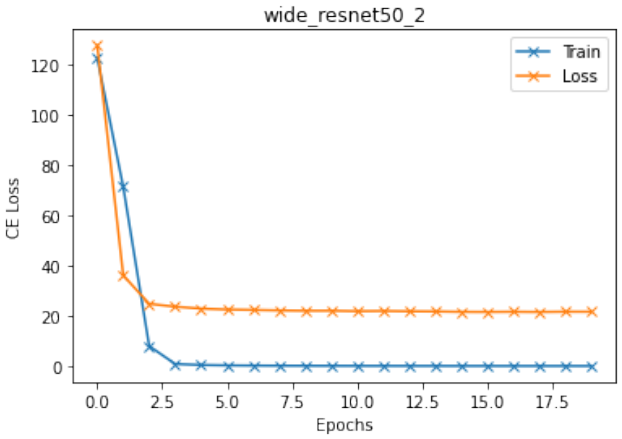


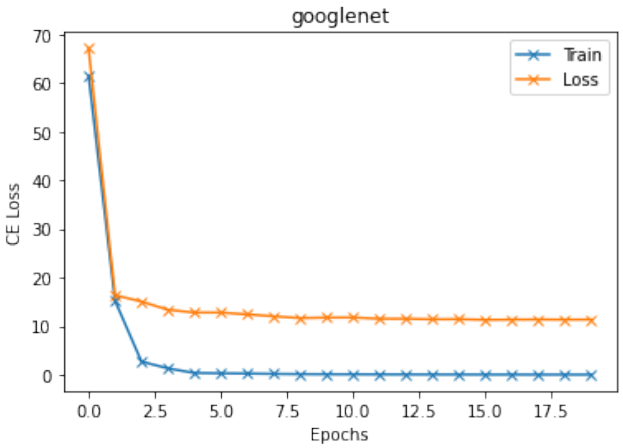

MnasNet Loss
CE Loss
Epochs
Train
Test


resnet18
CE Loss
Epochs
Train
Loss


MobileNet Loss
CE Loss
Epochs
Train
Test


resnext50_32x4d
CE Loss
Epochs
Train
Loss


wide_resnet50_2
CE Loss
Epochs
Train
Loss


googlenet
CE Loss
Epochs
Train
Loss