Image Recognition with ResNet 50

Alexji Gaston Sean Fulbright

[agast002@odu.edu](mailto:agast002@odu.edu) [sfulb001@odu.edu](mailto:sfulb001@odu.edu)

# Abstract

Machines have reached a point where they can recognize and classify objects in images from one another. For this term project, we experimented with convolutional neural network ResNet-50 and various sizes of training data. In this work, we reveal how resNet-50 can increase accuracy once the size of training data is increased.

# Introduction

Image recognition and image classification are two very different terms. Image recognition focuses on identifying objects in an image, while image classification analyzes the objects inside an image and categorizes the picture.

In the 1960s, computer vision started as a way to see computers try and mimic human vision with the hope of understanding what computers see. Once computing power and deep learning techniques grew over time, computers were able to train themselves and improve over time. One thing to note was that computers needed human input in order to recognize and classify the images in the photos.

Through this paper, we will reveal how the labels and size of images can affect the accuracy of image recognition and classification.

The paper is organized as follows:

* In section 2, we introduce the model of the experiment, Res-Net-50, and the experiment
* In section 3, we discuss the conclusion of the experiment and discuss the significance of the results produced

# Model and Experiment

In recent years, a popular trend online was taking photos of food and posting them to social media. One thing that interested us was the different ways people took photos of food, and how some people could not identify the object in the photos. Our hypothesis was that once we feed enough training data, specifically photos around categories of food, a computer would be able to accurately recognize and classify a photo.

Originally, a one-vs-all linear regression model was planned as the model. The training photos were to be broken by pixels and fed into the model, using the RGB binary value as input for the linear regression. However, since photos have varying sizes, this would mean a large amount of pixels being broken as well as the three dimensions of RGB. Another issue with using this method would be the large number of pixels per image and the 3 color channels. This would result in using large amounts of memory. A feature extraction model was chosen to help find the most valuable features in a photo and then feed the binary values of those features into the linear regression model was suggested instead but neither was used. Instead a residual neural network was chosen.

Once we looked at neural networks for the feature extractor, we realized that the linear regression model we chose was too complex and went with neural networks instead. ResNet-50 is a convolutional neural network with 50 hidden layers, including the input and output layers created for image classification. However, to save time since this is a semester project, we chose ResNet-50, which was trained on more than 14 million images from the ImageNet database, where the model came from OVHCloud’s ai-training-examples repository on github. In the OVHCloud’s example, ResNet-50 was also pretrained from ImageNet, but the focus was on images of flowers. For this purpose, transfer learning was used where ResNet-50 was already trained on images, but for the experiment it would solely focus on images of food.



(a): 512 x 512 (b): 512 x 343

Label: ‘Bread’ Label: ‘Bread’

Figure 1: Training data taken from Food-101 LabelBox

ResNet-50 training parameters, other than training data, are epochs, optimizer, validation data and batch size. The batch size is the number of images in each batch of the training and validation data. The data is loaded in batches to reduce memory use. The epoch is the number of passes with the training data while training the model. During each epoch it feeds all the training data in to train the model before using the validation data to adjust the model. The optimizer has two options to choose from, SGD and Adam. The Adam optimizer converges faster than SGD and is considered to be the default optimization regardless of field, while SGD generalizes better than Adam.

The experiment performed was to see how the number of training images impacted the accuracy of the image recognition of a trained model. The language that was used is python with the library tensorflow to use ResNet-50. To perform this experiment we set up functions to make and train the model and to create training data of a specific size. After those functions were set up a loop was made to go through the desired sizes, create the training data, train the model with that data and predict using that model on the test data. The validation data used in training and the testing data was the same each time. During each loop, the accuracy of the training data, validation data, and testing data was stored for use in a graph.

# Conclusion and Discussion

The results of the experiment were similar to what was expected but left some considerations for if the experiment was repeated. As the training data increased in size, the accuracy increased as shown in the following graph.

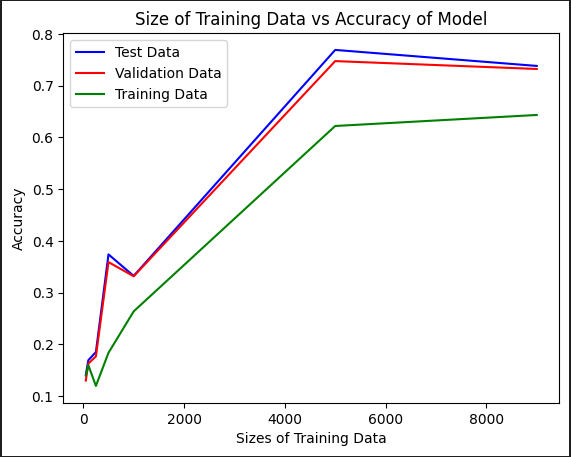


Figure 2: Results from one run of ResNet-50

There were not enough points plotted in order to make an informed prediction on the trend after a training data size of 5000 and between 1000 and 5000. The decrease shown between the sizes 0 and 1000 is likely an outlier and would require further runs to determine one way or another. When it comes to overfitting and underfitting, the graph shows no signs of overfitting at all. The reason for this is thought to be that the algorithm automatically corrects for overfitting through use of the validation data fed in with the training data. The only concern for the ResNet-50 model is underfitting. The maximum accuracy was with a training data size of 5000 with an accuracy of 76.9%.

Ways to improve the experiment performed would be, to perform more runs and increase the number of points tested. This experiment only had one run and there was not enough data to eliminate outliers and to draw definitive conclusions. If more runs were performed, the accuracies could be averaged across runs to generate a graph that could be used to draw a more accurate and definitive conclusion. When it comes to increasing the number of training sizes tested, ranges to increase the number of points would be between 1000 and 5000 and beyond 5000. There is only one point after 5000 used and thus is not enough information to determine the end behavior of the line. There are no points between 1000 and 5000 and thus there is a large section in the graph that is a straight line. With more points in between, there would be a more defined curve and a trend could be better concluded.

## **Acknowledgements**

## We would like to thank Dr. Jiangwen Sun for his advice and much needed feedback during both the beginning and final presentations of this project.

## **References**

Antonov, A. (2019, December 2). *Food-11 Image Dataset*. Food-11 image dataset. <https://www.kaggle.com/datasets/trolukovich/food11-image-dataset>

Bossard, Guillaumin, & Van Gool. (n.d.). *Food-101*. Labelbox. <https://labelbox.com/datasets/food-101/>

OVHCloud. (2022). Ai-training-examples. Github. <https://github.com/ovh/ai-training-examples/blob/main/notebooks/computer-vision/image-classification/tensorflow/resnet50/notebook-resnet-transfer-learning-image-classification.ipynb>