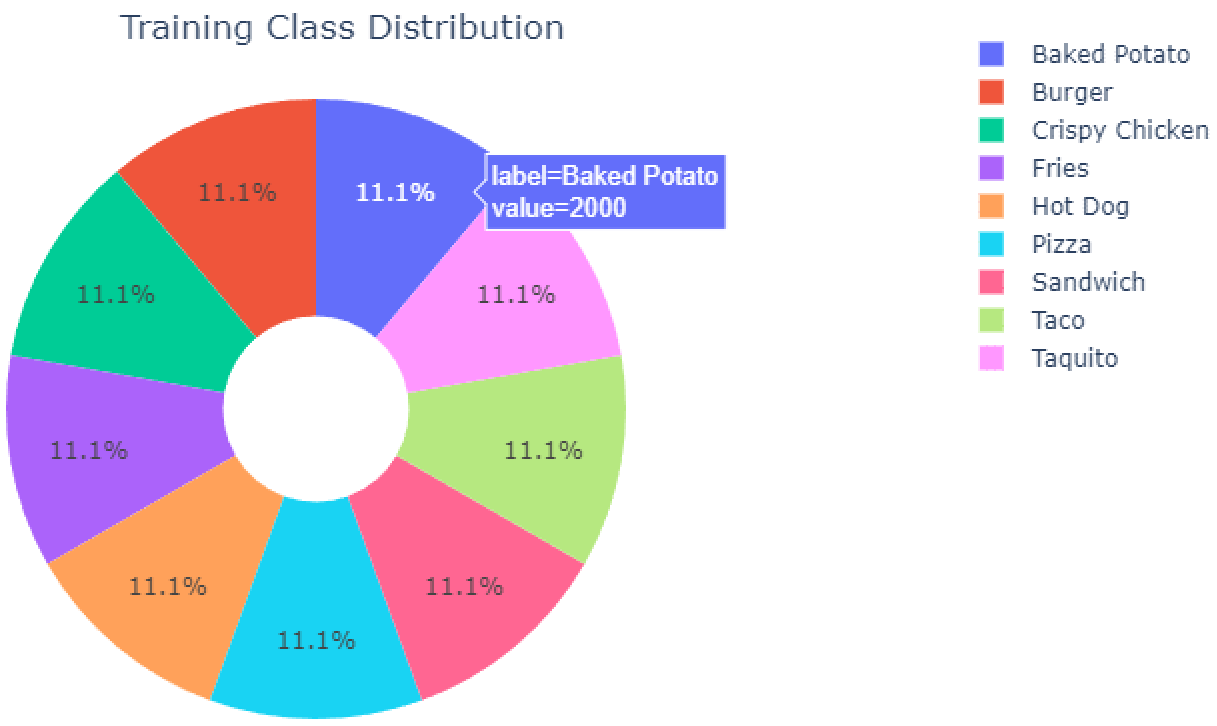
Abstract

This project focuses on classifying foods with the given images. For most of the food ordering companies like Getir, Yemeksepeti Grubhub etc. need a fast and reliable food classification model. The data consists of 2000 images of each of 9 different cuisines (Baked potato, burger, crispy chicken, donut, fries, hot dog, pizza, sandwich, taco, taquito) which were collected from Kaggle. Since we have enough images for each class, I did not do any data augmentation. For the pre-processing step, I applied scaling and label encoding. I have used cross-validation with stratifying. For the model part, first I tried the Sequential model from the Keras library. Layers for this model are Dropout, Maxpooling, Relu activation, 2D Convolutional layer, and since our problem is multiclass classification as a final layer softmax. Later I will explain why I have chosen these layers. Since this model did not propagate remarkable results, I want to use the transfer learning method to use the pre-trained model Resnet50. As a brief result, while working with enormous image data, we need higher processing power.



Introduction

In my problem, I wanted to classify foods by the images. This project can convert most of the frustrating work into automatically done. With the help of this project, visual impaired users can detect what kind of meal they are having. Another use is calorie tracking. By only taking a picture of the meal, you can keep the calorie that you have taken. Of course, this project can turn into other projects and can be integrated with mobile applications. For this project, I have used CNN neural networks and pre-trained model.

Related Work

Calorie tracking applications with the premium options. Bitesnap.

Approach

My data consists of 2000 images of each of 9 different cuisines (Baked potato, burger, crispy chicken, donut, fries, hot dog, pizza, sandwich, taco, taquito). Since our data consists of images, divided 255 to normalize all of them between 0 to 1. First, I created a Sequential model. Since I am working with image data, I have Convolutional layers. Additionally, Maxpooling, Drop-out and Flatten layers, as an activation function I used Relu. Since my goal is multiclass classification, at the output layer I have a layer with 9(number of class) nodes with a softmax activation function. I have used Adam optimizer and for the loss function from the Keras library, I have used SparseCategoricalCrossentropy. I picked this because for classification most of the other loss functions were giving errors. For monitoring purposes, I have looked at validation accuracy with patience 5. The number of epochs is 100 and the batch size was 32 because of the memory limitations. After I didn't get a great result from the model that I created, I have chosen to use transfer learning. With the ResNet-50 model, I have created a new sequential model. I merged the pre-trained model with the dropout layer and output layer. In the pre-trained model, I have used Piecewise Constant Decay as a learning rate scheduler from the Keras library. It makes it possible model to learn with different learning rates. For finding the optimal solutions, SGD is being used. With learning rate scheduler. This model was trained with 100 epochs. Since our total number of parameters was 23,518,793 needed more powerful computing power. Because of that I deployed my model into the MS Azure ML Studio compute instance with 2 core and 14 GB RAM. Training time was 10 hours for this model. For preventing overfitting I have used cross validation method.

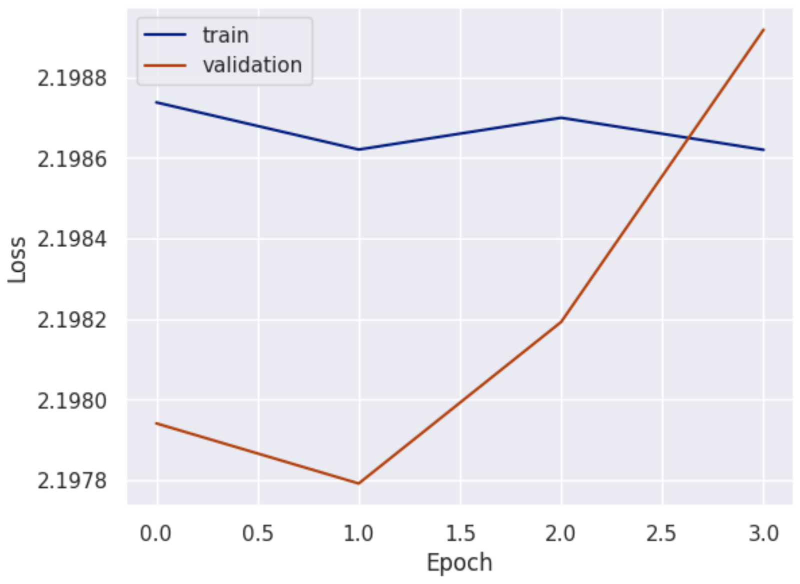
Implementation Details

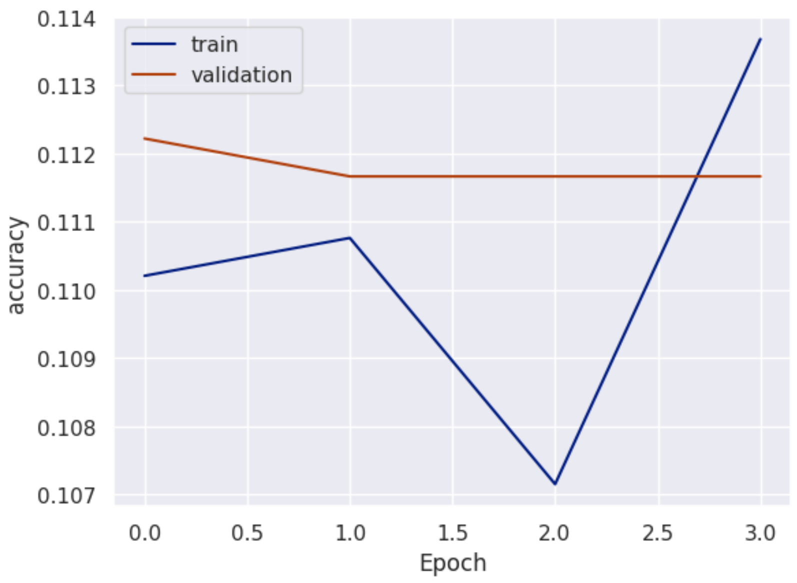
In this project, I have used numpy arrays to store image pixel information. I used sklearn library for preprocessing steps like label encoding while splitting data to train and test, matplotlib for visualizing images and accuracy change by epochs while learning, Keras for creating sequential model baseline, creating layers optimizers. ResNet-50 model for transfer learning. For computing I tried my first samples on my personal computer. For training model with the higher number of epochs I have used MS Azure ML Studio compute instance.

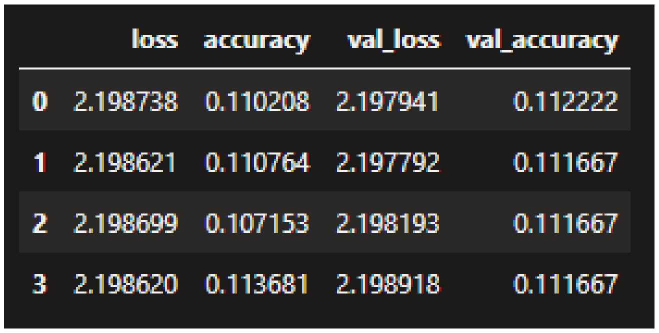
Evaluation Details, B. Results and Analysis/Discussion:

I have split my data by 80%-20% as train and test. For validation, I used cross-validation approach with stratify. For evaluating the model I choose to look validation accuracy score. In the CNN model, score was not good enough. It early stopped because of the weak accuracy. In the pre-trained model, validation accuracy increased continuously. After 100 epochs, validation accuracy of the Pre-trained model has minor changes but comparing to the CNN model, it created way much better result. As a result, we can say that if we are dealing with very complex problem, we need better networks and since pre-trained models have been trained with big datasets, there are powerful with the predicting image. Since they are trained with enormous data, with little epochs they create great results in the image data.

CNN MODEL EVALUATION

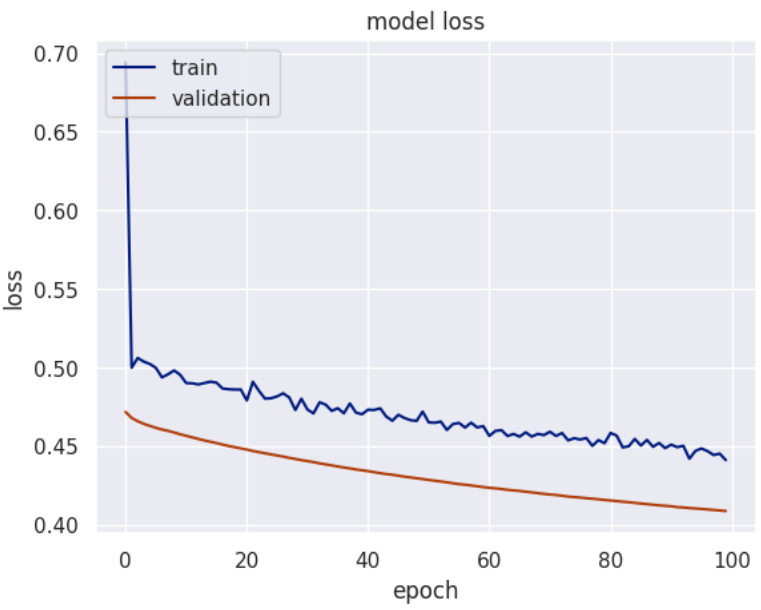


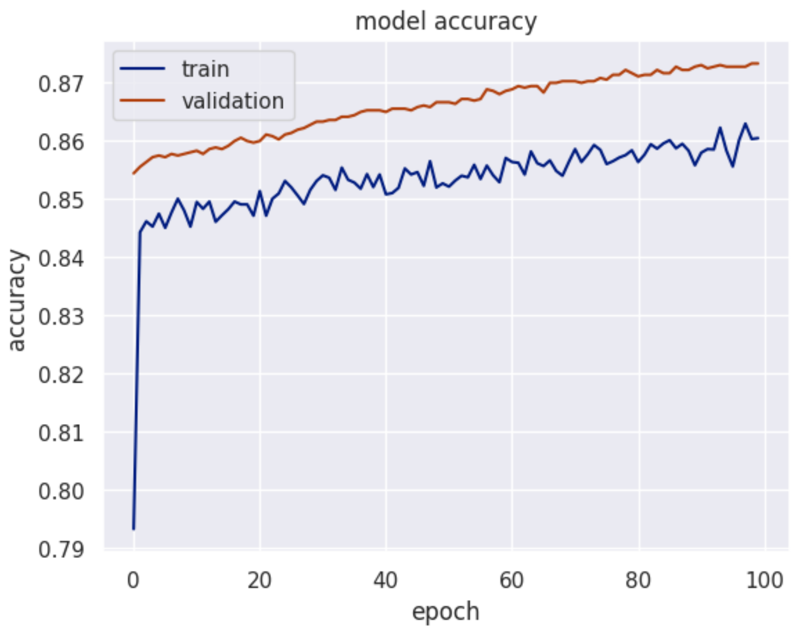


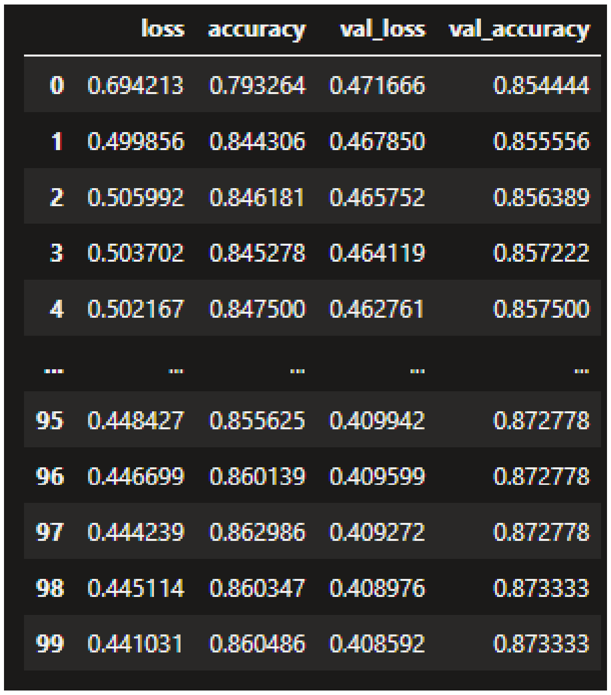


	loss	accuracy	val_loss	val_accuracy
0	2.198738	0.110208	2.197941	0.112222
1	2.198621	0.110764	2.197792	0.111667
2	2.198699	0.107153	2.198193	0.111667
3	2.198620	0.113681	2.198918	0.111667

PRE-TRAINED MODEL (RESNET-50) EVALUATION







	loss	accuracy	val_loss	val_accuracy
0	0.694213	0.793264	0.471666	0.854444
1	0.499856	0.844306	0.467850	0.855556
2	0.505992	0.846181	0.465752	0.856389
3	0.503702	0.845278	0.464119	0.857222
4	0.502167	0.847500	0.462761	0.857500
...	...	...	...	...
95	0.448427	0.855625	0.409942	0.872778
96	0.446699	0.860139	0.409599	0.872778
97	0.444239	0.862986	0.409272	0.872778
98	0.445114	0.860347	0.408976	0.873333
99	0.441031	0.860486	0.408592	0.873333

Conclusion

While training the model with more than 100 epochs my computer crashed, because of that I used Azure ML Studio compute instance with 4 cores and 28 GB RAM. For predicting images, basic CNN models are not powerful enough. In this kind of situation, best practices are using the pre-trained model. Also, need powerful computing power since the number of parameters can increase exponentially according to your network structure.

References

Fast Food Classification | BiT | Acc : 94% | Kaggle

Dataset source: Fast Food Classification Dataset - V2 | 20k Images | Kaggle