

Transfer Learning Approach : Apps For Image Classification on Dessert Problem

Farel Firman, @farelyue

Mikhael Adiputra, @pitydevil

Theresia Veronika Rampisela, @theresiavr

Tobias Ivandito Margogo Silalahi, @tobiassilalahi

Introduction

Knowledge about nutritional intake needed every day is important for everyone to know, because fulfillment of nutritional intake each person varies depending on internal factors and external factors. Lack of knowledge about nutritional intake will cause a nutritional problem. Nutritional problem occurs because nutritional intake is not enough or excessive. It will have an impact on one's productivity.

Dessert is food eaten after the main course and style from western cultures. Many Indonesian people now make dessert as a habit. Dessert usually is sweet courses such as fruits, ice cream, cake, etc. Dessert is used to digest the main course and acts as mouth freshener. Furthermore, dessert also increases one's mood. But, too many calories eaten will result in various health disorders

Dessert Classification Application can be an alternative solution to solve this problem. This application can be used to identify dessert types before being eaten and display information about calories with recipes or ways of making it. This information can be used to adjust dietary habit or planned diet and used to make the dessert

Goals

To deploy an application that can identify the type of dessert and display information about food calories and how to make the dessert from food that is being eaten

Methodology

The data used is image data Food-101 that comes from the Kaggle platform (<https://www.kaggle.com/kmader/food41>). This data consists of 101 types of food. Each food consists of 1000 images. We only used 22 types of food from the dataset, which is dessert. This approach used the Convolutional Neural Network (CNN) using pre-trained model from InceptionV3.

The stages in image classification are:

1. Splitting data into training and validation data with ratios of 80% and 20%
2. Train model CNN from scratch with predetermined architecture
3. Determine best pre-trained model by comparing performance from all three pre-trained models that are InceptionV3, VGG-16, and Resnet-50
4. Tuning hyperparameter and used data augmentation to improve model performance

Results

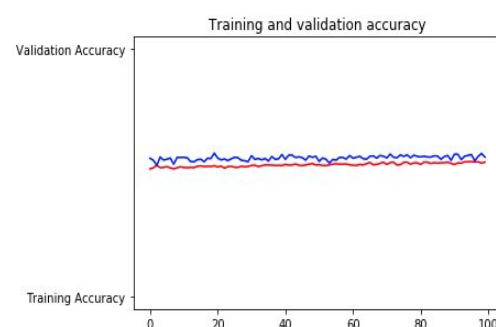
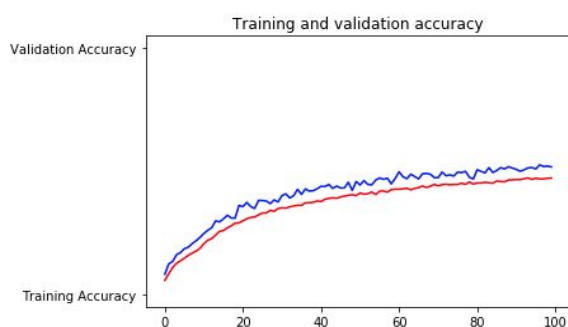
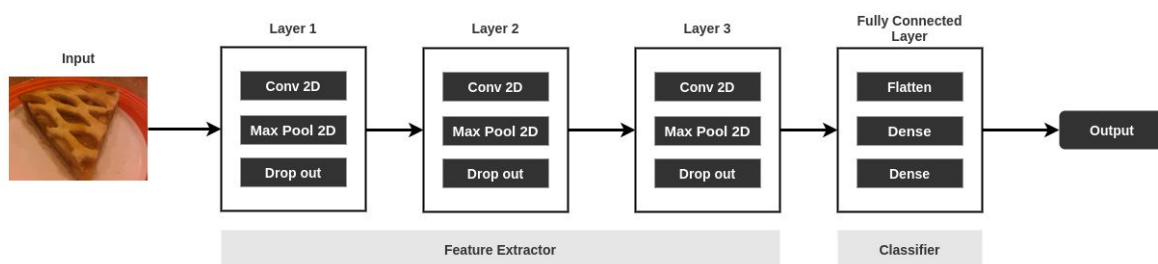
Splitting Data

Data is divided into two parts that are training and validation data with ratios are 80% and 20%. Training data is used to build a model and validation data is used to evaluate the model.

CNN Model From Scratch

CNN Model is one of the main categories to do image classification tasks. CNN takes an input image and through the convolution layers used to extract features from the image and finally with activation function used to determine class from the image. The first step from building a CNN model is the determined model architecture. As for the architecture used is:

1. Convolutional Layer 2D with 32 filters each then followed by Max Pooling 2D 2x2 and Dropout is 20%. Output shape: 149x149x16
2. Convolutional Layer 2D with 64 filters each then followed by Max Pooling 2D 2x2 and Dropout is 20%. Output shape: 73x73x32
3. Convolutional Layer 2D with 128 filters each then followed by Max Pooling 2D 2x2 and Dropout is 20%. Output shape: 35x35x64
4. Fully Connected Layer 1. Output shape: 1x1x256
5. Fully Connected Layer 2. Output shape: 1x1x512
6. Output Layer. Output shape: 1x1x22

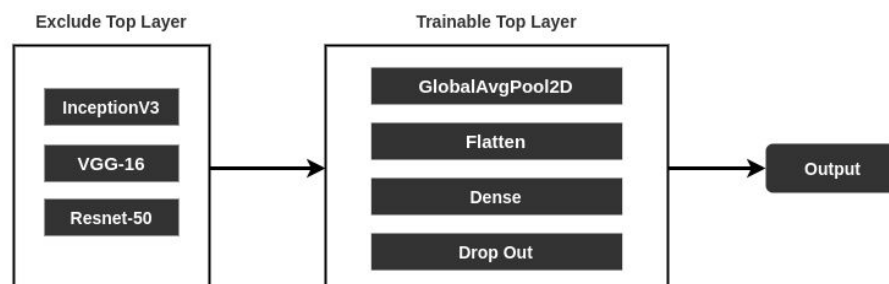


Added epoch from 100 to 300 improve accuracy model performance from the baseline model is 51.80% to 59.80%. This improvement is fairly small and tends to be stable on 50% to 60%.

Improved CNN Model

The improved performance model used the Transfer Learning approach. Transfer Learning is an approach used model that has been trained before called a pre-trained model. Then, the feature from this model is fine-tuned to adjust to the case that will be implemented. There are many pre-trained models. But in this case, we only use 3 pre-trained such as InceptionV3, VGG-16, and Resnet-50 to compare the model performances. The top layer for each pre-trained model adjusts to the case Dessert problem with 22 classes. As for the architecture, we used:

1. Exclude top layer for each pre-trained model such as InceptionV3, VGG-16, and Resnet-50
2. Global Average Pooling 2D layer followed by Flatten layer used to convert featured map into a single column vector as an input to Fully Connected Layer
3. Dense Layer with 512 filters each and used ReLU activation function. Then followed by Dropout layer with 50% parameter
4. Output layer consists of 22 classes and used Softmax activation function



Model	Validation Accuracy (%) (Epoch = 5)	Validation Accuracy (%) (Epoch = 10)
InceptionV3	55.09	58.84
VGG-16	44.93	48.68
Resnet-50	4.55	4.55

Based on the result of comparing all three pre-trained models, we found that InceptionV3 has a much better performance model compared to the others. For epoch = 5 validation accuracy is 55.09% and for epoch = 10 validation accuracy is 58.84%. Hence, InceptionV3 is chosen as the pre-trained model

The resulting model suspected of having an overfitting problem. It is seen from the accuracy in training and validation data is quite different. Overfitting problem occurs because the model is too fitted on training data and result in poor accuracy in validation data. Hence, we used data augmentation to solve this problem. Data augmentation is used to create a variation of the images that can improve the ability of the fit models to generalize.

As for the hyperparameter final model used are:

1. Epoch: 140
2. Batch Size :
Training data: 800
Validation data: 200
3. Optimizer: Adam
4. Loss Function: Categorical Cross-Entropy
5. Data augmentation :
Rescale: 1/255
Rotation Range: 40
Width Shift Range : 0.2
Height Shift Range: 0.2
Shear Range: 0.2
Zoom Range: 0.2
Horizontal Flip: True

Based on the result from the final model we get training accuracy is 72.87% and validation accuracy is 72.52%. The results obtained are good considering the model from scratch only scored 55% accuracy and now it improved to about 17%. This model also does not have overfitting problem, as there is not too much difference between the accuracy of training data and validation data.