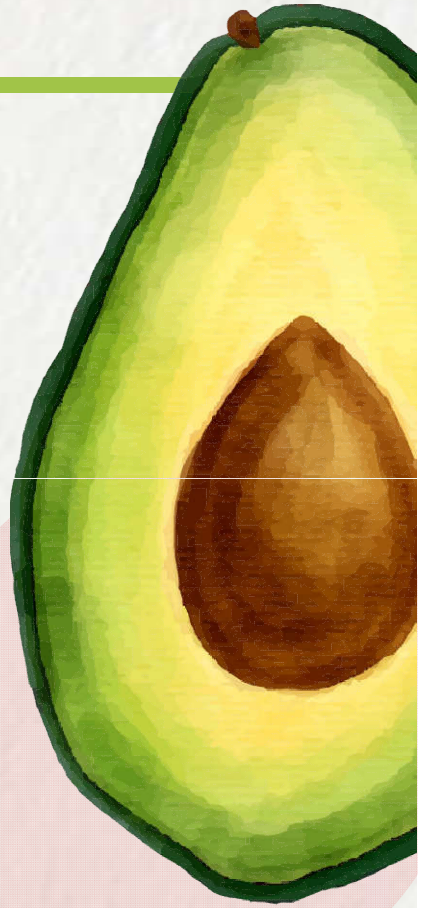
# FRUIT AND VEGETABLE CLASSIFICATION

Google Bangkit Tensor Flow's Assignment

Maharani Devira Pramita
Tia Dwi Setiani

# TABLE OF CONTENTS





# METADATA

Fruits 360 dataset by Horea Muresan and Mihai Oltean downloaded from Kaggle

10 types of fruits and vegetables (10 classes) are: ***apple red 1, avocado, banana, blueberry, cauliflower, grape white, onion red, pepper yellow, potato sweet, tomato 3.***

7162 total number of images,

5367 training set size,

750 validaton set size

1045 tets set size.

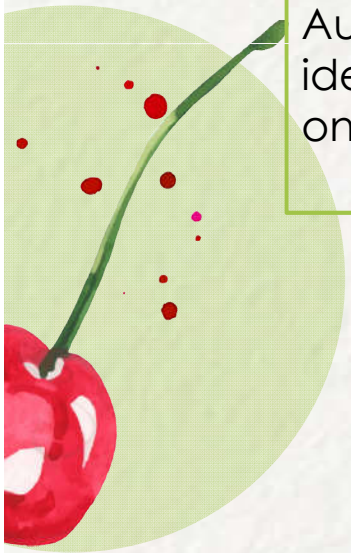
Image dimension: 100, 100, 3.

format in jpg.

# PROBLEM FRAMING and HYPOTHESIS (1)

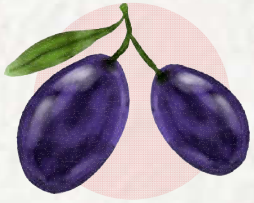
Agriculture-related works include sorting and packaging requires lots of time and resources for manual execution.

Automation may be the solution to this problem. And the identification of the fruit and vegetable class accurately is one of the main challenges for this automation.



The hypothesis for our ML model is prediction function that classify fruits and vegetables based on the corresponding label.

# PROBLEM FRAMING and HYPOTHESIS (2)



## GOAL, OUTCOME, and OUTPUT

✔ We want our ML model to classifiy image of fruits and vegetables

✔ Our ideal outcome is to perform a classification of fruits and vegetables

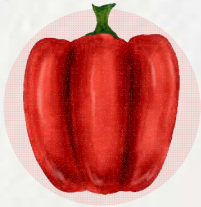
✔ The output from our ML model will be a prediction of a fruit and vegetable label

## SUCCESS METRICS

✔ Our success metrics are: the number of fruits & vegetables are acurately classified.

✔ Our key result is: to acurately predict the images of fruits and vegetables with 95% of accuracy

✔ Our model is deemed a failure if accuracy is < 95%



## USING THE OUTPUT

✔ The Output from the ML model will be made as soon as a new image is inputted to the model

✔ The outcome will be used for the implementation of fruit and vegetable identification on agricultural works.

# PREPARATIONS

## Data Preparations

✔ Download the **fruit 360 dataset** from **Kaggle**

✔ Choose 10 classes from the dataset, which is 10 types of fruits and vegetable, 5 each.

✔ Upload the chosen dataset using sklearn.datasets.load_files

## Hyperparameters Tuning

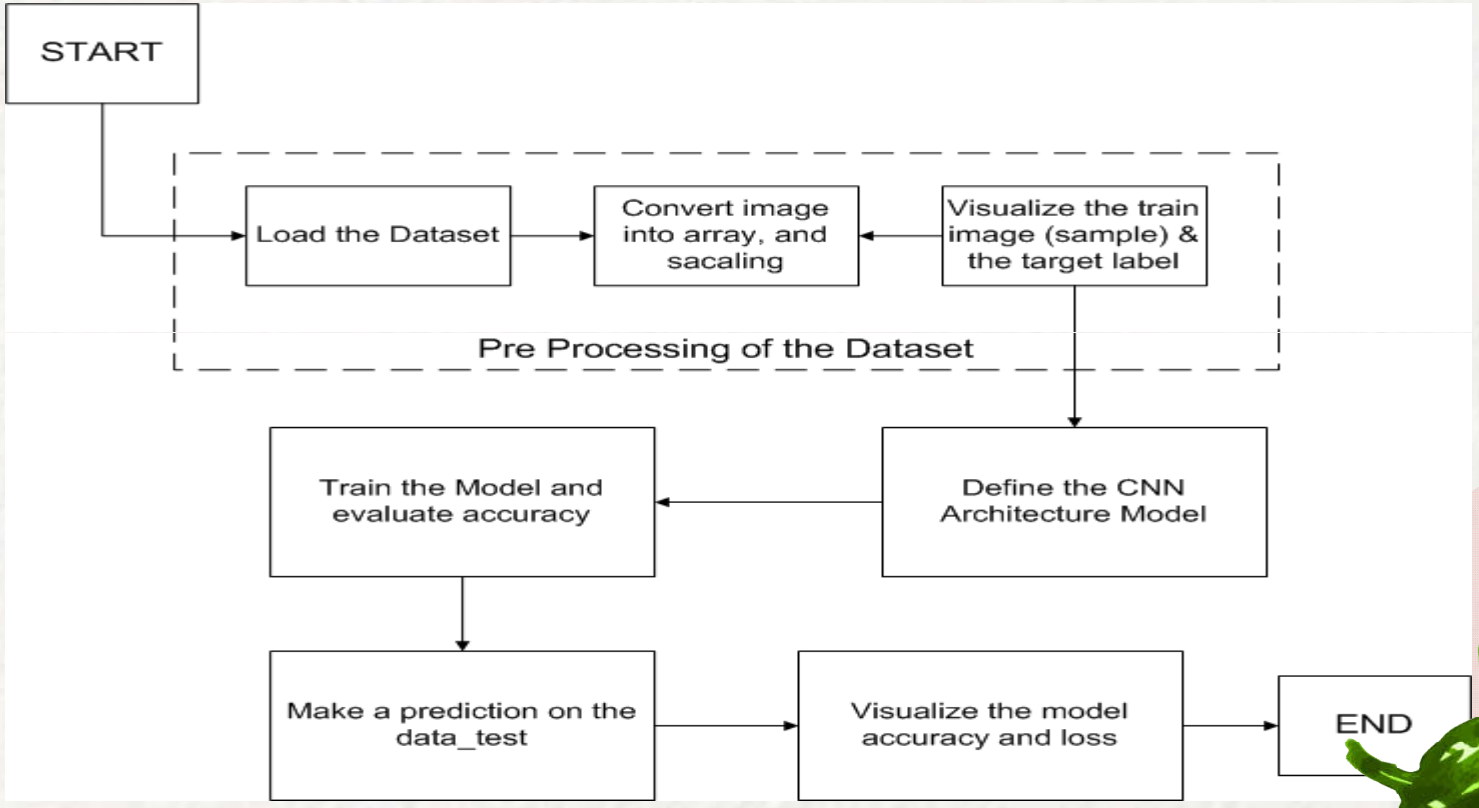
We did hyperparameters tuning to get the best accuracy and minimal loss

✔ Paremeter dropout from 0.1 - 0.5

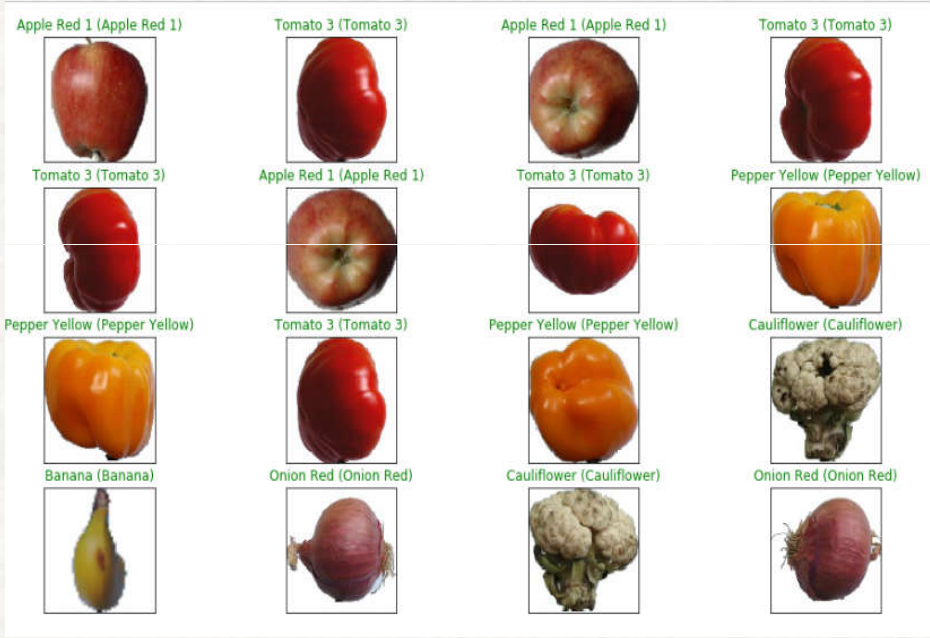
✔ Epoch setting up to 50.

✔ Batch size from 32 – 64.

# TECHNIQUES


START
Load the Dataset
Convert image into array, and sacaling
Visualize the train image (sample) & the target label
Pre Processing of the Dataset
Train the Model and evaluate accuracy
Define the CNN Architecture Model
Make a prediction on the data_test
Visualize the model accuracy and loss
END

# RESULTS and CONCLUSSION (1)



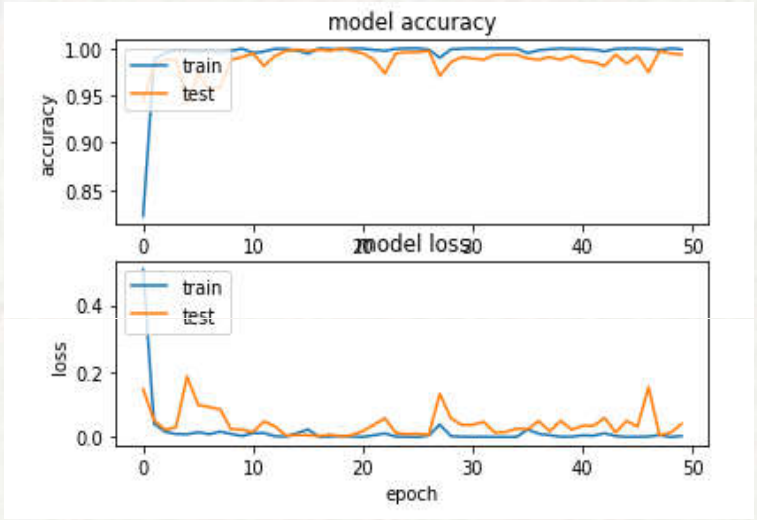

## Result Visualization of the images and the corresonding label

A group of **images, prediction label, and actual label (in parenthesis)** are shown in the result picture.

**The prediction label matches the actual label.**

Means that **our model can predict the label of an image properly.**

# RESULTS and CONCLUSSION (2)




```
Epoch 00019: val_loss improved from 0.00334 to 0.00228, saving model to cnn_basic_fruit.hdf5
Epoch 20/50
5367/5367 [==============================] - 28s 5ms/step - loss: 0.0017 - acc: 0.9998 - val_loss: 0.0057
- val_acc: 0.9973

Epoch 00020: val_loss did not improve from 0.00228
Epoch 21/50
5367/5367 [==============================] - 27s 5ms/step - loss: 4.1016e-04 - acc: 0.9998 - val_loss: 0.0
183 - val_acc: 0.9947

Epoch 00021: val_loss did not improve from 0.00228
Epoch 22/50
5367/5367 [==============================] - 27s 5ms/step - loss: 0.0055 - acc: 0.9983 - val_loss: 0.0377
- val_acc: 0.9880
```

## Model Accuracy and Model Loss

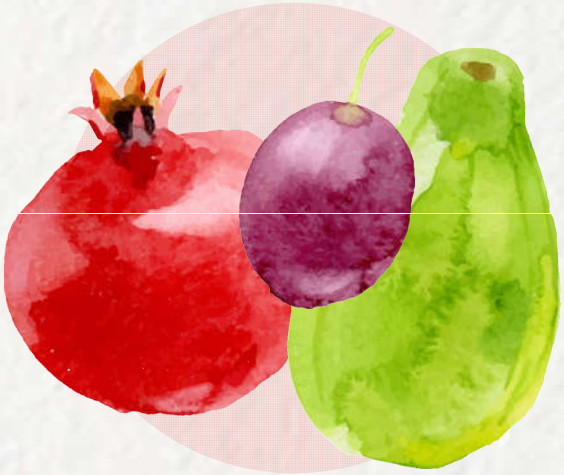
Model accuracy, model loss function and the step of training results are shown in the pictures.

**Model accuracy and model loss started to convergence in epoch size: 3**

**The accuracy for both the train and test set are above 95 %. Our model can achieve 99% of accuracy.**

**Best valuation loss of our model is 0.00228 and was achieved in epoch size: 19.**

# THANKS!



MAHARANI DEVIRA PRAMITA
devipramita

TIA DWI SETIANI
tiadwi



Github repository for the project:

**https://github.com/devipramita/GoogleBangkit-FruitClassification**