



# B21-CAP0454

Automated Self Checkout (ASC) with Object Detection for Better Shopping Experience



# Let's Begin SUMANGGA

*"Super Machine Learning Group  
Engagement in Action"*



# Meet our team



**Hanan Iqbal Alrahma /**  
***Mobile Development***  
***(Android)***



**Akmal Adnan Attamami /**  
***Machine Learning***

# Meet our team



**Aditheo Firman Saputra /**  
***Mobile Development***  
***(Android)***



**Mohammad Arif Ikhsanudin /**  
***Cloud computing***



**Mochamad Hakim Akbar**  
**Assidiq Maulana /**  
***Machine Learning***

## Content

01

Background

02

Comparison

03

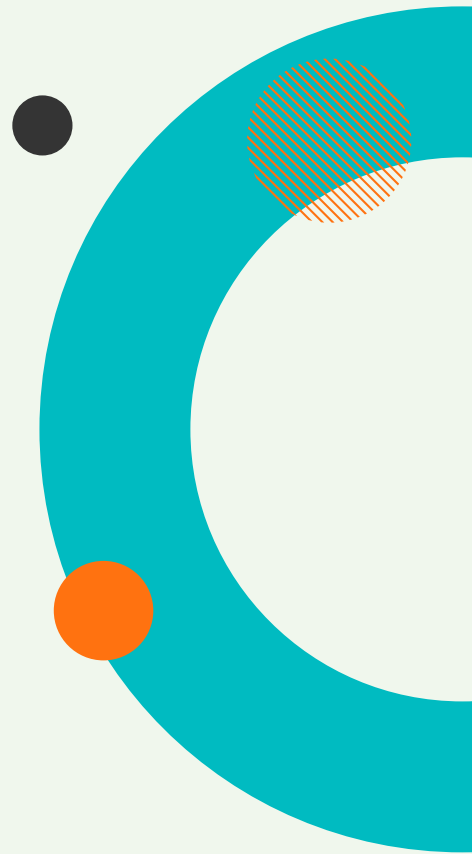
Evaluation

04

Documentation

05

Business Insider





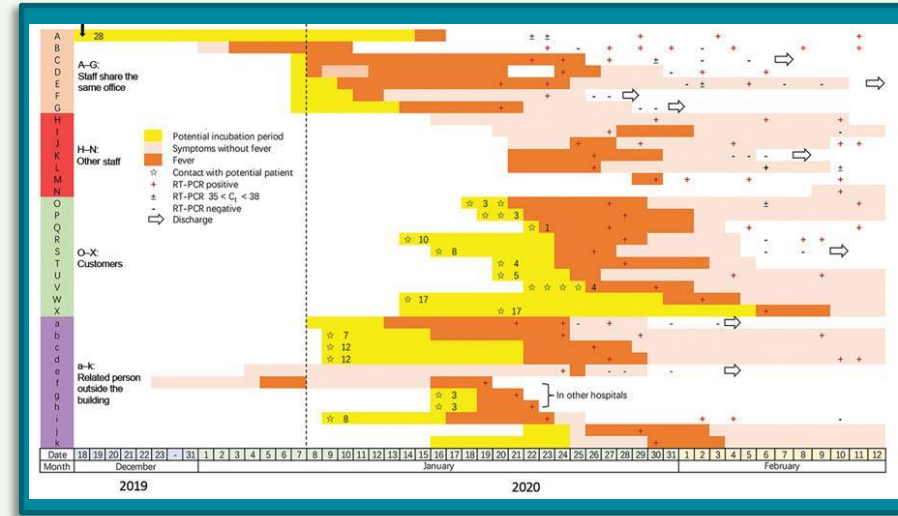
01

# Background

How could this happen? Let me tell you ...

## Background (1/3)

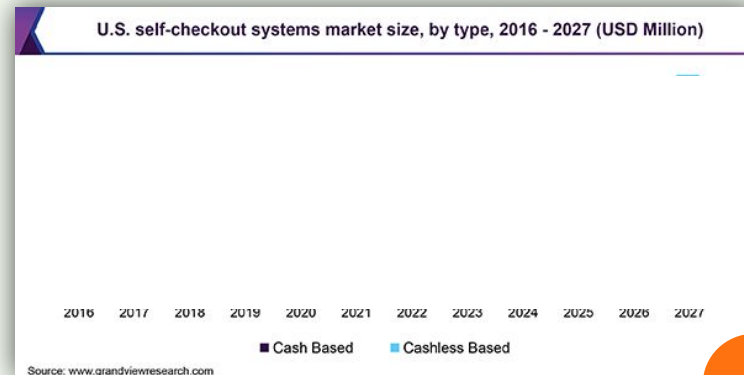
Many supermarkets rely on **manual cashier** systems to serve customers. This causes the queue of customers to **get longer** and **not suitable** for COVID-19. Person **not to stay in public** for long.



Source :  
Cluster of COVID-19 at shopping mall in Wenzhou, China

## Background (2/3)

**Increase** of using Self-Checkout Systems around the world because of this **effectively** and **efficiency**.

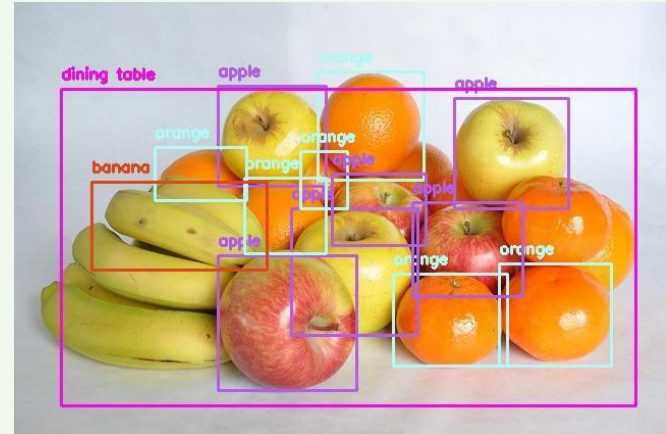
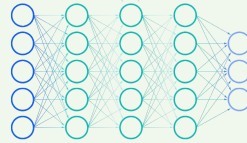


Source :  
US, Global



## Background (3/3)

Scanning the barcode to identify the price product is **inefficient**. Machine Learning make shopping checkout process **easily and quickly**



02

# Comparison

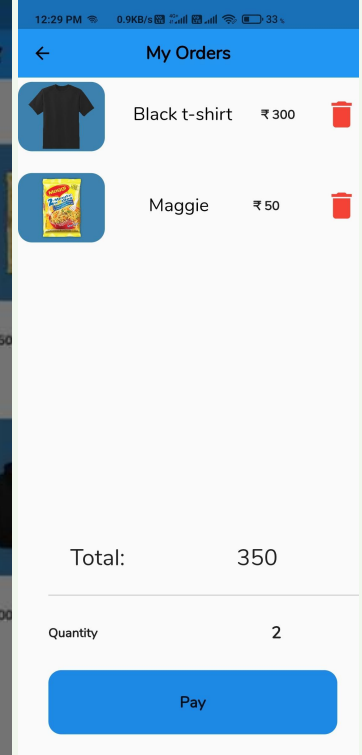
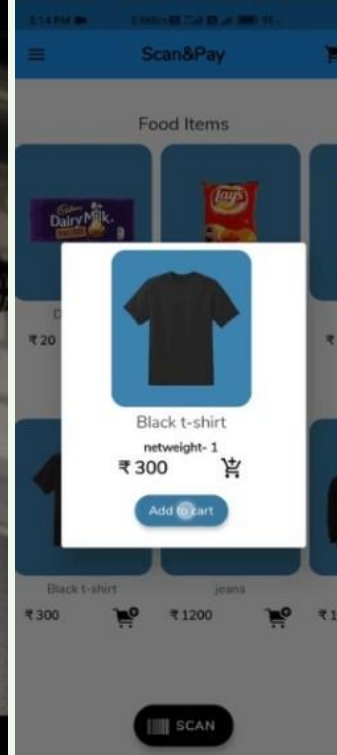
There are other product vs Our solution



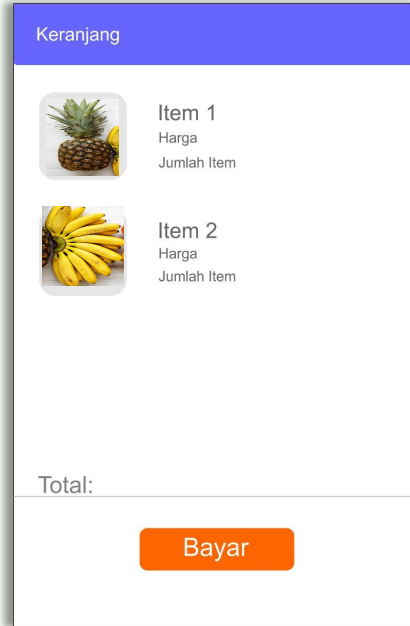
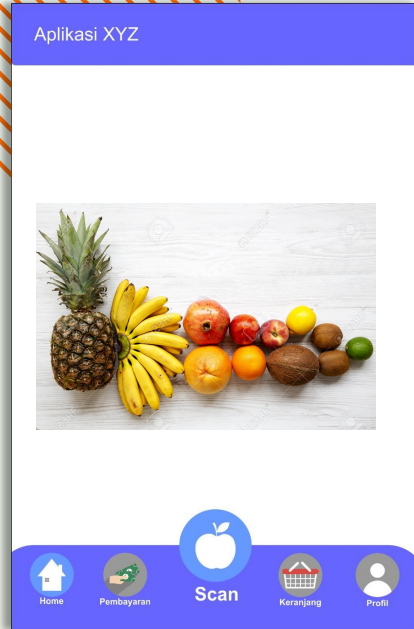
# Comparison

## Existing Solution

Other solution only use Barcode to analyze the description of Product



# Comparison



## Our Solution

Using Object Detection  
Algorithm to extract  
image feature to get  
Product description

03

# Evaluation

Analyzing our model performance



# IMPLEMENTATION

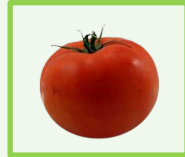
Tomato



Banana



Tomato



Orange

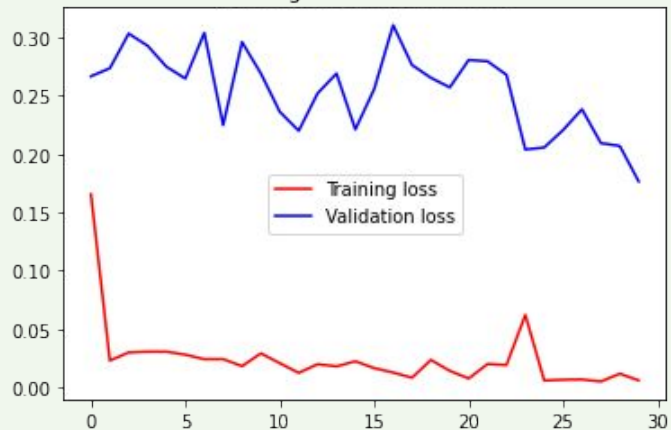
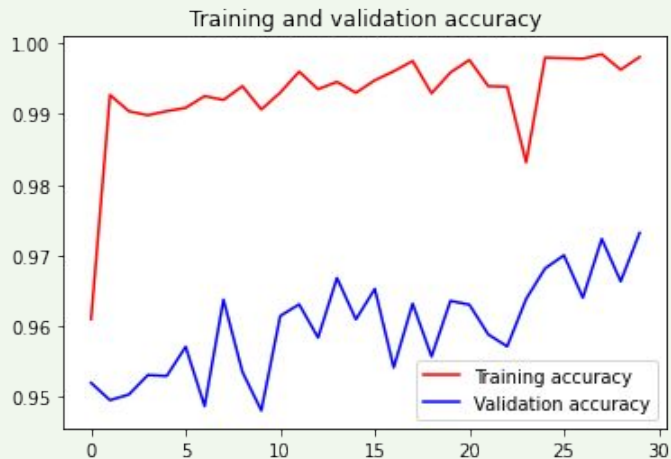


Check out List :

Fruits	Quantity	Price (Rp)	Total (Rp)
Tomato	2	500	1000
Banana	3	5000	15000
Orange	1	1000	1000

# VERSION 1

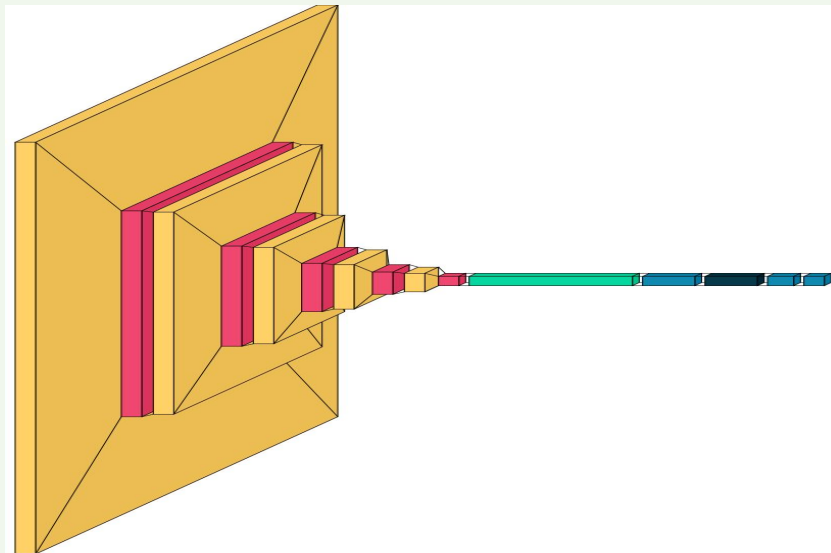
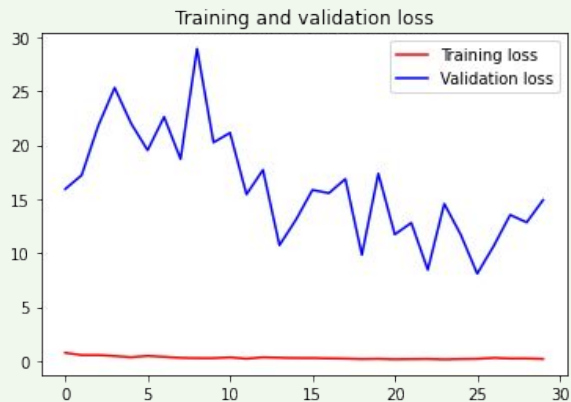
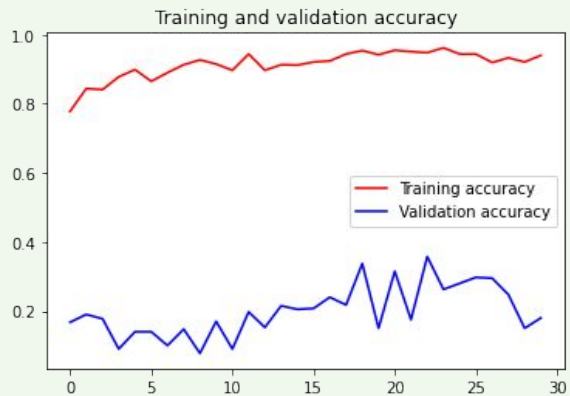
Using CNN and adding  
depthwise\_conv2d layer



Model: "sequential"

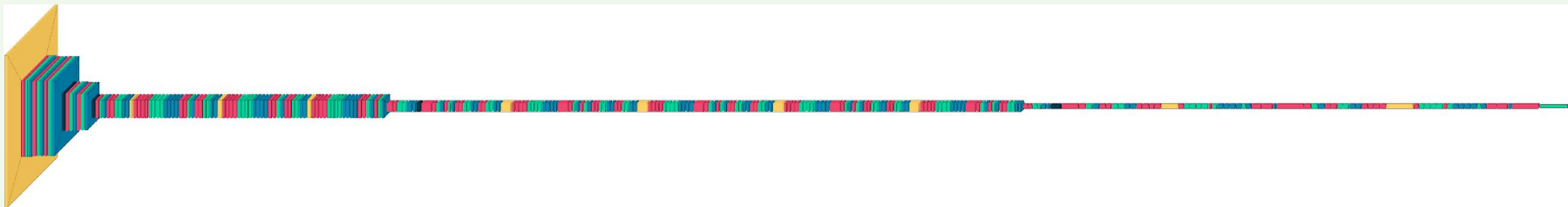
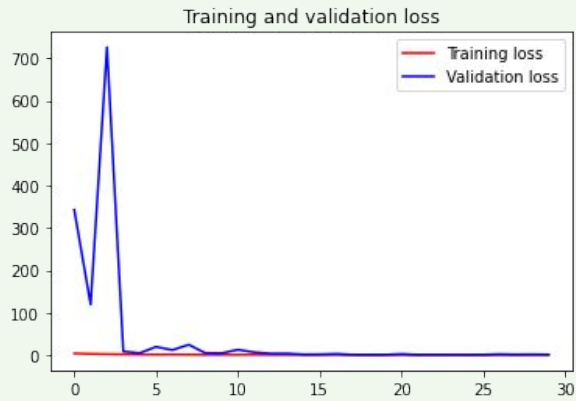
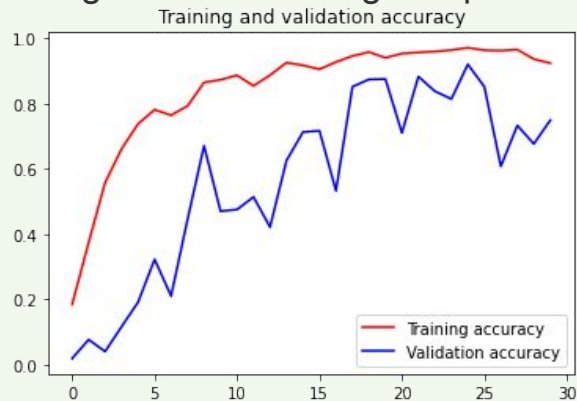
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 16)	448
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
depthwise_conv2d (DepthwiseC	(None, 27, 27, 16)	16400
max_pooling2d_1 (MaxPooling2	(None, 13, 13, 16)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	9280
conv2d_2 (Conv2D)	(None, 13, 13, 64)	36928
conv2d_3 (Conv2D)	(None, 13, 13, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 512)	2359808
dropout (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dense_3 (Dense)	(None, 67)	17219
Total params: 2,645,267		
Trainable params: 2,645,267		
Non-trainable params: 0		

## Using transfer learning mobilenet v2





## Using transfer learning Inception v3



# VERSION 4

Using transfer learning mobilenet v4

Model: "sequential"

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 1280)	2257984
dense (Dense)	(None, 512)	655872
dropout (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dense_3 (Dense)	(None, 67)	17219
Total params: 3,062,403		
Trainable params: 804,419		
Non-trainable params: 2,257,984		

```

132/132 [=====] - 119s 903ms/step - loss: 0.0050 - accuracy: 0.9984 - val_loss: 0.0999 - val_accuracy: 0.9849
Epoch 00022: val_loss did not improve from 0.06573
Epoch 23/30
132/132 [=====] - 119s 903ms/step - loss: 0.0082 - accuracy: 0.9975 - val_loss: 0.0835 - val_accuracy: 0.9881
Epoch 00023: val_loss did not improve from 0.06573
Epoch 24/30
132/132 [=====] - 119s 905ms/step - loss: 0.0074 - accuracy: 0.9975 - val_loss: 0.1070 - val_accuracy: 0.9869
Epoch 00024: val_loss did not improve from 0.06573
Epoch 25/30
132/132 [=====] - 119s 902ms/step - loss: 0.0065 - accuracy: 0.9982 - val_loss: 0.1038 - val_accuracy: 0.9858
Epoch 00025: val_loss did not improve from 0.06573
Epoch 26/30
132/132 [=====] - 121s 913ms/step - loss: 0.0073 - accuracy: 0.9977 - val_loss: 0.0919 - val_accuracy: 0.9847
Epoch 00026: val_loss did not improve from 0.06573
Epoch 27/30
132/132 [=====] - 120s 909ms/step - loss: 0.0080 - accuracy: 0.9974 - val_loss: 0.1307 - val_accuracy: 0.9846
Epoch 00027: val_loss did not improve from 0.06573
Epoch 28/30
132/132 [=====] - 120s 905ms/step - loss: 0.0095 - accuracy: 0.9969 - val_loss: 0.0999 - val_accuracy: 0.9862
Epoch 00028: val_loss did not improve from 0.06573
Epoch 29/30
132/132 [=====] - 120s 908ms/step - loss: 0.0070 - accuracy: 0.9978 - val_loss: 0.1451 - val_accuracy: 0.9818
Epoch 00029: val_loss did not improve from 0.06573
Epoch 30/30
132/132 [=====] - 120s 907ms/step - loss: 0.0068 - accuracy: 0.9978 - val_loss: 0.1386 - val_accuracy: 0.9833
Epoch 00030: val_loss did not improve from 0.06573

```



# How to Improve?

- **Add** more data and change augmentation parameters
- **Tuning** training parameters
- **Reduce** Similar classes



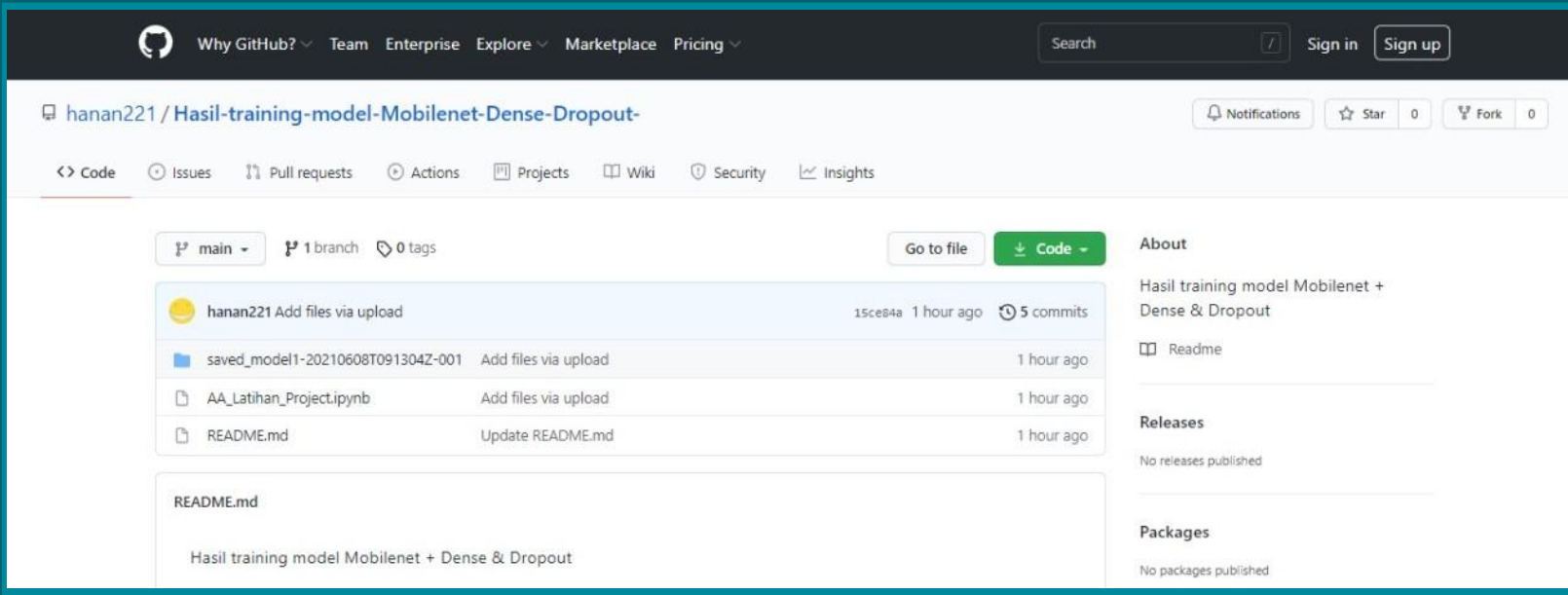
04

# Documentation

From our work



# Our Repository at **Git Hub**



The screenshot shows the GitHub interface for the repository 'hanan221/Hasil-training-model-Mobilenet-Dense-Dropout'. The repository is in the 'main' branch and has 1 branch and 0 tags. The commit history shows 5 commits, with the latest commit 'hanan221 Add files via upload' made 1 hour ago. The commit details show a folder 'saved\_model1-20210608T091304Z-001' and two files: 'AA\_Latihan\_Project.ipynb' and 'README.md', all added 1 hour ago. The README.md file is visible, showing the title 'Hasil training model Mobilenet + Dense & Dropout'. The right sidebar contains sections for 'About' (Hasil training model Mobilenet + Dense & Dropout), 'Readme', 'Releases' (No releases published), and 'Packages' (No packages published).

hanan221 / Hasil-training-model-Mobilenet-Dense-Dropout

Notifications Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Code

hanan221 Add files via upload 15ce84a 1 hour ago 5 commits

saved_model1-20210608T091304Z-001	Add files via upload	1 hour ago
AA_Latihan_Project.ipynb	Add files via upload	1 hour ago
README.md	Update README.md	1 hour ago

README.md

Hasil training model Mobilenet + Dense & Dropout

About

Hasil training model Mobilenet + Dense & Dropout

Readme

Releases

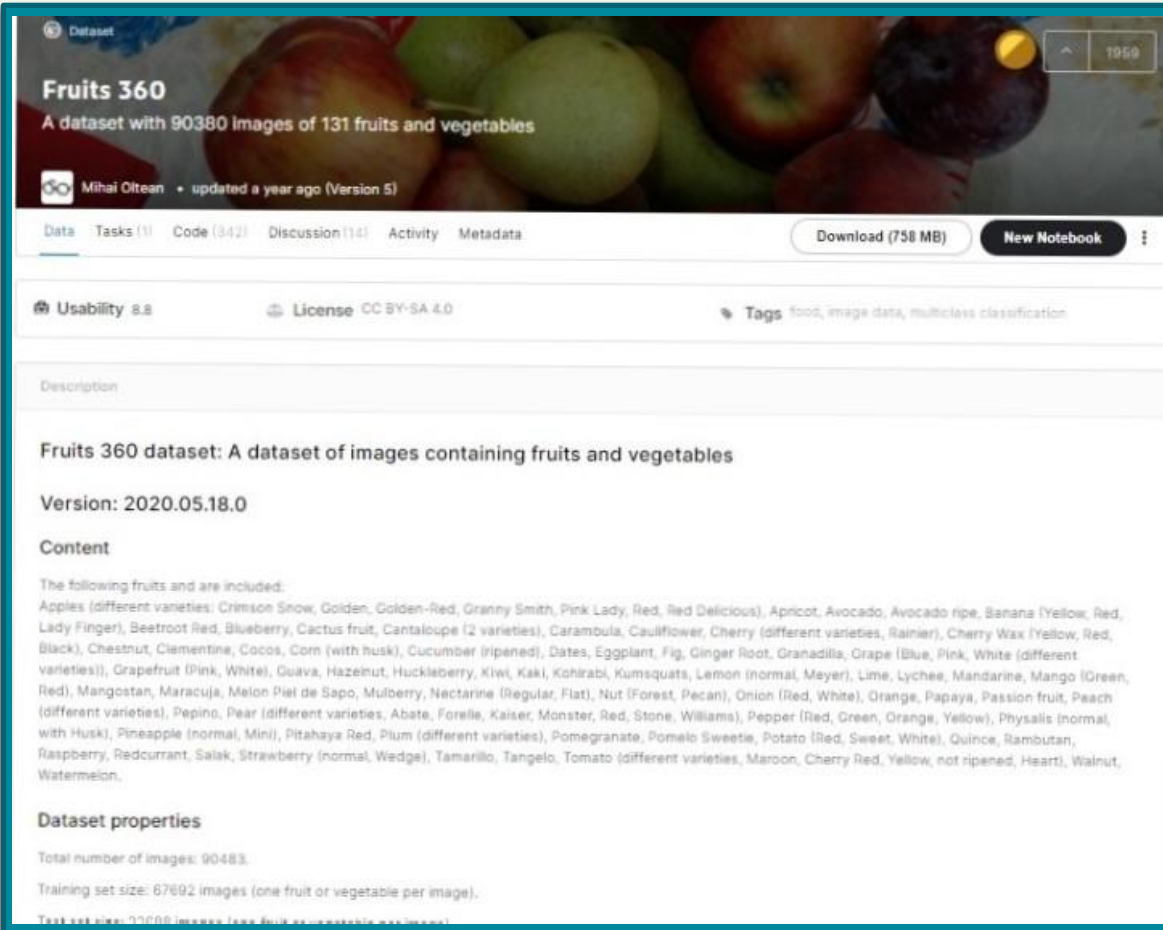
No releases published

Packages

No packages published

Source :

<https://github.com/hanan221/Hasil-training-model-Mobilenet-Dense-Dropout.git>



**Fruits 360**  
A dataset with 90380 images of 131 fruits and vegetables

Mihai Oltean • updated a year ago (Version 5)

Data Tasks (1) Code (342) Discussion (14) Activity Metadata

Download (758 MB) New Notebook

Usability 8.8 License CC BY-SA 4.0 Tags food, image data, multiclass classification

**Description**

**Fruits 360 dataset: A dataset of images containing fruits and vegetables**

**Version: 2020.05.18.0**

**Content**

The following fruits and are included:

Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambola, Cauliflower, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Corn (with husk), Cucumber (ripened), Dates, Eggplant, Fig, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Nut (Forest, Pecan), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Stone, Williams), Pepper (Red, Green, Orange, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow, not ripened, Heart), Walnut, Watermelon.

**Dataset properties**

Total number of images: 90483.

Training set size: 67692 images (one fruit or vegetable per image).

Test set size: 22791 images (one fruit or vegetable per image).

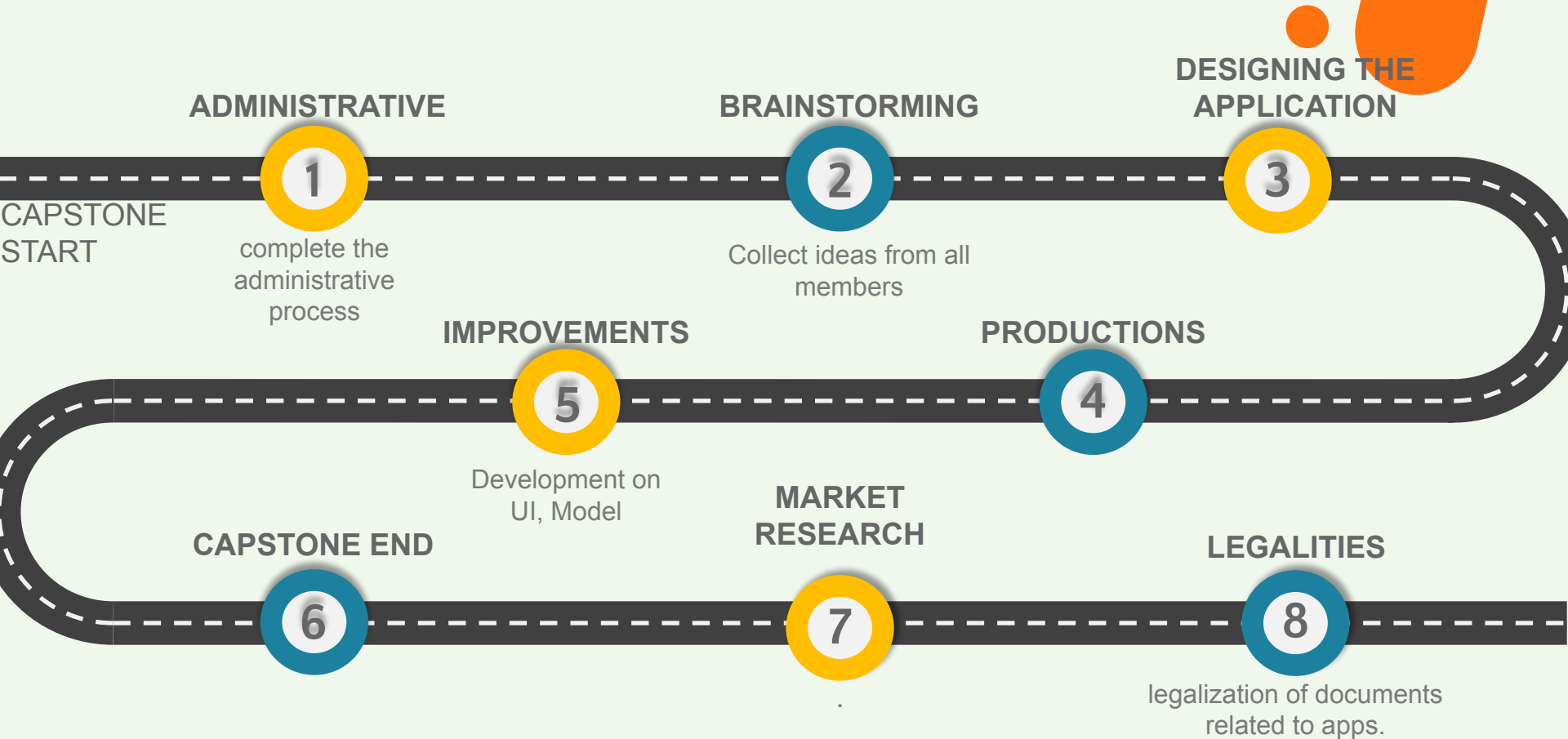
# Dataset Source

## FRUIT-360

Source :

<https://www.kaggle.com/moltean/fruits>

# Our *Milestone*





05

# Business Insider

Go-to-market Proposal





# Target Market

people who need us

Age Range	Profession	Roles	Spesific
25-60	Owner of the grocery	Regulator & Stimulus	Who own and makes the rules of the grocerys

## Local Deployment

Timeline	Roles	Budget (max) in USD
Early July - End of July	Market Research & Analysis (Observation ,Questionnaire & Market Analysis )	\$1.000 / Rp14.000.000
Early August – End of September	Research & Development, Production (Other Design, Improvements, and API Services)	\$1.000 / Rp14.000.000
Early October – End of December	Advertising, Operations and Maintenance with Cloud Services	\$1.500 / Rp21.000.000
July - December	Other (Team Salary, Electricity, Internet, Transportation, and Taxes & Legalities)	\$2.500 / Rp35.000.000
TOTAL		\$6000 / Rp84.000.000

## Budgeting @5000 USD

Category	Propotions	Budget (max) in USD
Team Salary	20%	\$1.000 / Rp14.000.000
Operational		
1. Renting Cloud Service		
2. Productions		
3. Reaserching		
4. Surveys		
5. Transportation, Electricity and Internet	80%	\$4.000 / Rp56.000.000
6. Advertising		
7. Paid Experts, Observer and Respondens		
8. Legalities Regristation		
9. Taxes		

## Budgeting @10.000 USD

Category	Propotions	Budget (max) in USD
All aspect covered in Budgeting	50%	\$5.000 / Rp70.000.000
Additional for Team Salary	15%	\$1.500 / Rp21.000.000
Additional for Research/Ops	20%	\$2.000 / Rp28.000.000
Marketing and Sales	20%	\$2.000 / Rp28.000.000
Market research competitive analysis	10%	\$1.000 / Rp 14.000.000
Future Development/R&D	20%	\$2.000 / Rp28.000.000
Other Expenses (taxes, reserves)	10%	\$1.000 / Rp14.000.000

“This is a quote, words full of wisdom that someone important said and can make the reader get inspired.”

—Someone Famous