MASK CLASSIFICATION

แนะนำสมาชิก

1.นาย กรวรรษ พวงแก้ว 07610567

2.นาย นิธิศ ดวงแสงเหล็ก 07610613

Preprocess

```python
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,  #sแบ่งข้อมูล เพื่อ training 80% และ validate 20%
  subset="training",
  seed=123,
  image_size=(img_rows, img_cols),
  batch_size=batch_size)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="validation",
  seed=123,
  image_size=(img_rows, img_cols),
  batch_size=batch_size)
```

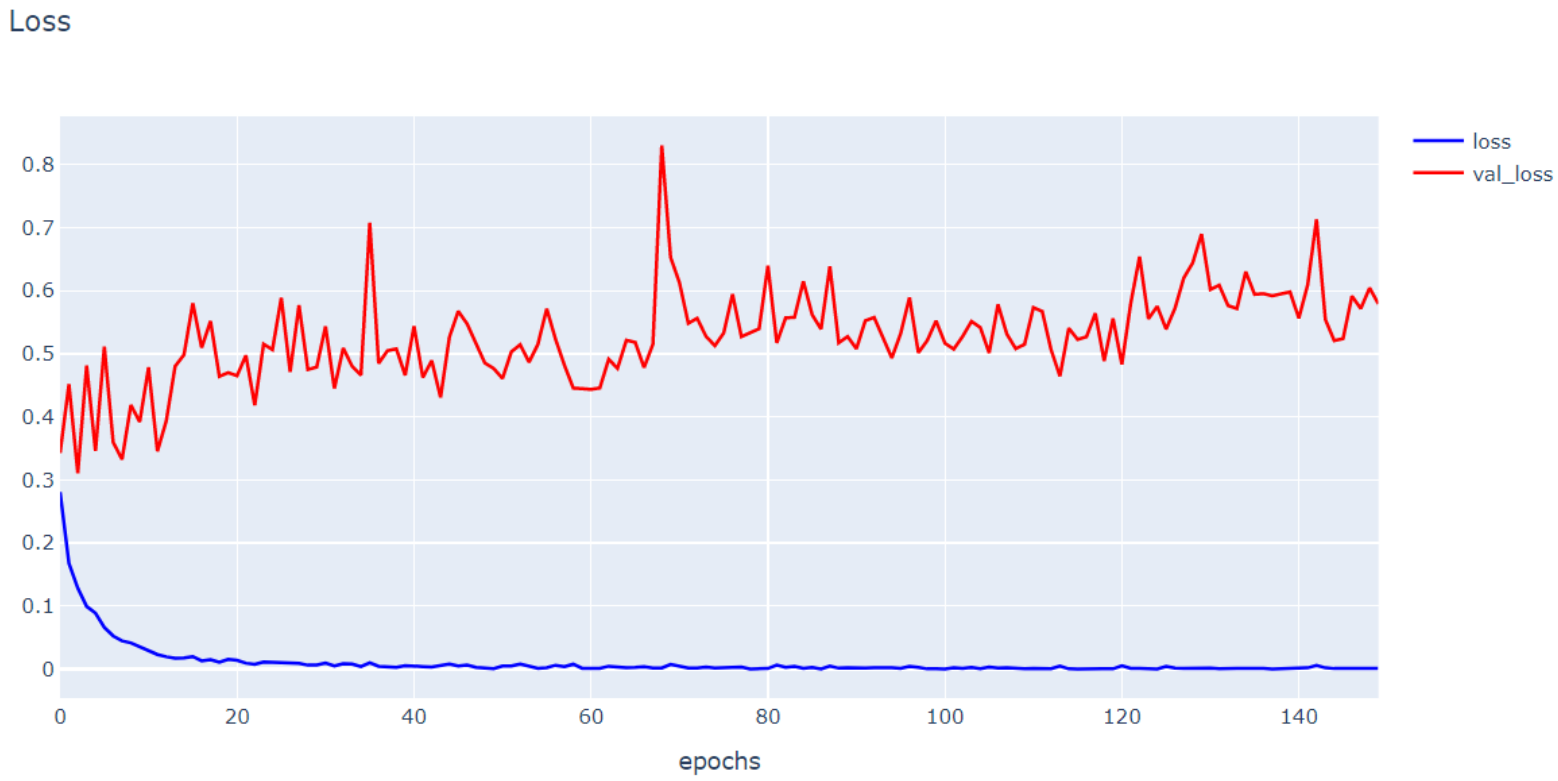
```
Found 11391 files belonging to 2 classes.
Using 9113 files for training.
Found 11391 files belonging to 2 classes.
Using 2278 files for validation.
```

Preprocess คือจัดเตรียมข้อมูลก่อนนำไปใช้จริง โดยจะทำการแบ่งข้อมูลที่นำไป training 80% และ validate 20%

```
num_classes =  len(list(class_names))
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_rows, img_cols, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.BatchNormalization(),
  layers.Dropout(0.3),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.BatchNormalization(),
  layers.Dropout(0.3),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.BatchNormalization(),
  layers.Dropout(0.3),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
```

การใช้ฟังก์ชั่นต่างๆ ในการเพิ่มประสิทธิภาพ ในการเทรน กรณีถ้าผลของ model ไม่เป็นที่พอใจ โดยการใช้เทคนิค ต่างๆ เช่น **MaxPooling** คือ การปรับขนาดภาพให้มีขนาดเล็กลง **BatchNormalization** คือ การ **Normalzation** โดยการกำหนดขอบเขตของข้อมูล **Dropout** คือ การตัดข้อมูลบางส่วนที่ไม่ต้องการออก

ประสิทธิภาพ


Loss
loss
val_loss
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
epochs

ประสิทธิภาพ

In [25]:
```
import requests
from tensorflow import keras
from IPython.display import Image
from io import BytesIO
test_path = 'C:/Users/jane_/Desktop/ai_cpsu/PicForTrainMo/download (7).jpg'
img = keras.preprocessing.image.load_img(
test_path, target_size=(img_rows, img_cols)
)
img_array = keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch
predictions = predict_model.predict(img_array)
score = tf.nn.softmax(predictions[0])
print("ใส่หน้ากาก",score[0],"\n ไม่ใส่หน้ากาก",score[1])
display(Image(filename=test_path,width=150, height=150))
if score[0]==np.max(score) :
    m = "ใส่หน้ากาก"
elif score[1]==np.max(score) :
    m = "ไม่ใส่หน้ากาก"
print(
    "ภาพนี้ {} {:.2f}%."
    .format(m, 100 * np.max(score))
)
```

```
ใส่หน้ากาก tf.Tensor(0.99999654, shape=(), dtype=float32)
 ไม่ใส่หน้ากาก tf.Tensor(3.4618738e-06, shape=(), dtype=float32)
```

ตรงส่วนนี่จะเป็นการนำรูปภาพมาทำนายผลกับ **model** ที่เราสร้างว่ามีความถูกต้องของรูปภาพนั้นกี่เปอร์เซ็นต์

ประสิทธิภาพ

ใส่หน้ากาก tf.Tensor(0.99999654, shape=(), dtype=float32)
ไม่ใส่หน้ากาก tf.Tensor(3.4618738e-06, shape=(), dtype=float32)

ภาพนี้ ใส่หน้ากาก 100.00%.

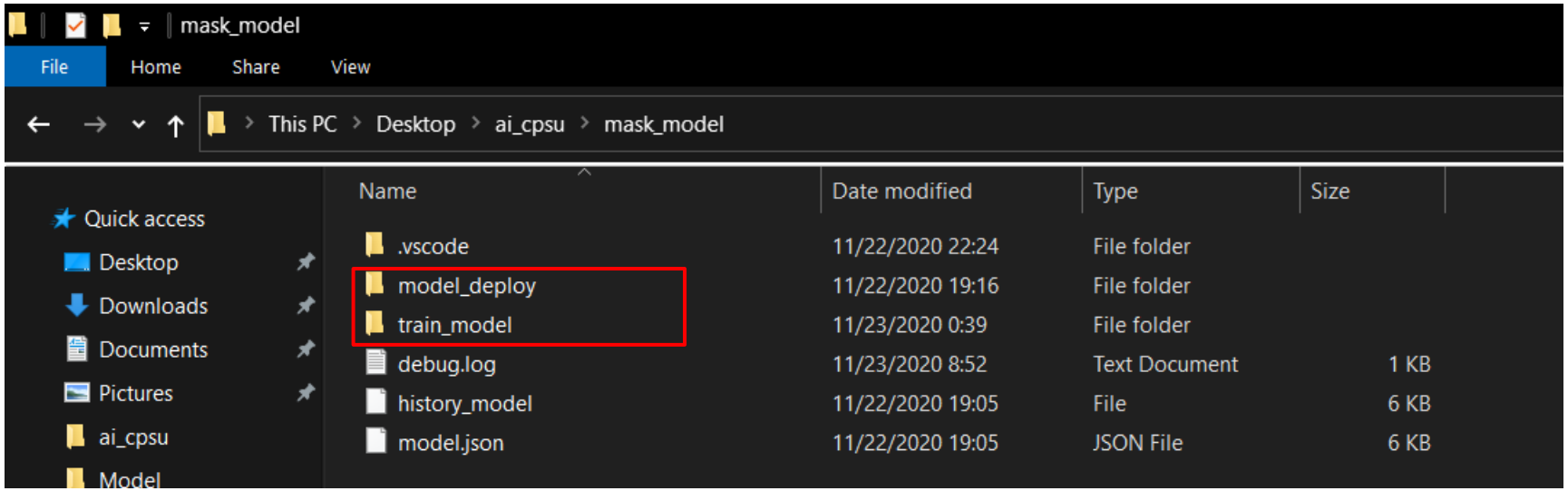
การ Deploy

model_deploy > python > Dockerfile > ...

```
1  FROM python:3.6.8-slim-stretch
2  RUN apt-get update && apt-get install -y python-pip \
3      && apt-get clean
4  WORKDIR /app
5  COPY api.py .env model.h5 requirements.txt ./
6  RUN pip install --no-cache-dir -r requirements.txt
7  CMD uvicorn api:app --host 0.0.0.0 --port 80 --workers 6
```

อันนี้จะเป็นการ Deploy ขึ้นไปตัวของ Docker

โดยจะใช้คำสั่ง docker-compose up –d --build


mask_model
File
Home
Share
View
This PC
Desktop
ai_cpsu
mask_model
Quick access
Desktop
Downloads
Documents
Pictures
ai_cpsu
Model
Name
Date modified
Type
Size
.vscode
11/22/2020 22:24
File folder
model_deploy
11/22/2020 19:16
File folder
train_model
11/23/2020 0:39
File folder
debug.log
11/23/2020 8:52
Text Document
1 KB
history_model
11/22/2020 19:05
File
6 KB
model.json
11/22/2020 19:05
JSON File
6 KB

การเรียกใช้งาน

โดยเราจะเรียกผ่าน Localhost ของตัว Docker