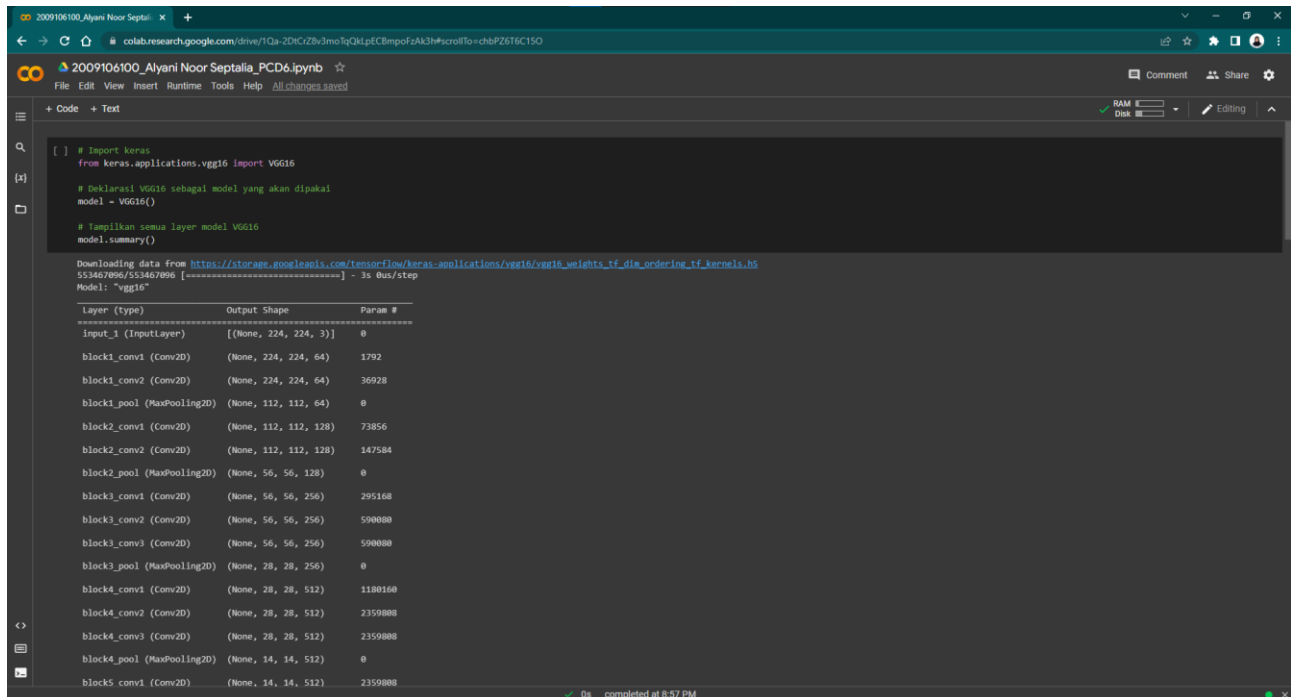


Screenshot Projek 6 dan Deskripsi Tiap Tahapannya

1. Import Library Model



```
[ ] # Import keras
from keras.applications.vgg16 import VGG16

# Deklarasi VGG16 sebagai model yang akan dipakai
model = VGG16()

# Tampilkan semua layer model VGG16
model.summary()
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467896/553467896 [=====] - 1s Dns/step
Model: "vgg16"

| Layer (type) | Output Shape | Param # |
|----------------------------|-----------------------|---------|
| input_1 (InputLayer) | [(None, 224, 224, 3)] | 0 |
| block1_conv1 (Conv2D) | (None, 224, 224, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, 224, 224, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, 112, 112, 64) | 0 |
| block2_conv1 (Conv2D) | (None, 112, 112, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, 112, 112, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, 56, 56, 128) | 0 |
| block3_conv1 (Conv2D) | (None, 56, 56, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, 28, 28, 256) | 0 |
| block4_conv1 (Conv2D) | (None, 28, 28, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, 14, 14, 512) | 0 |
| block5_conv1 (Conv2D) | (None, 14, 14, 512) | 2359808 |

completed at 8:57 PM

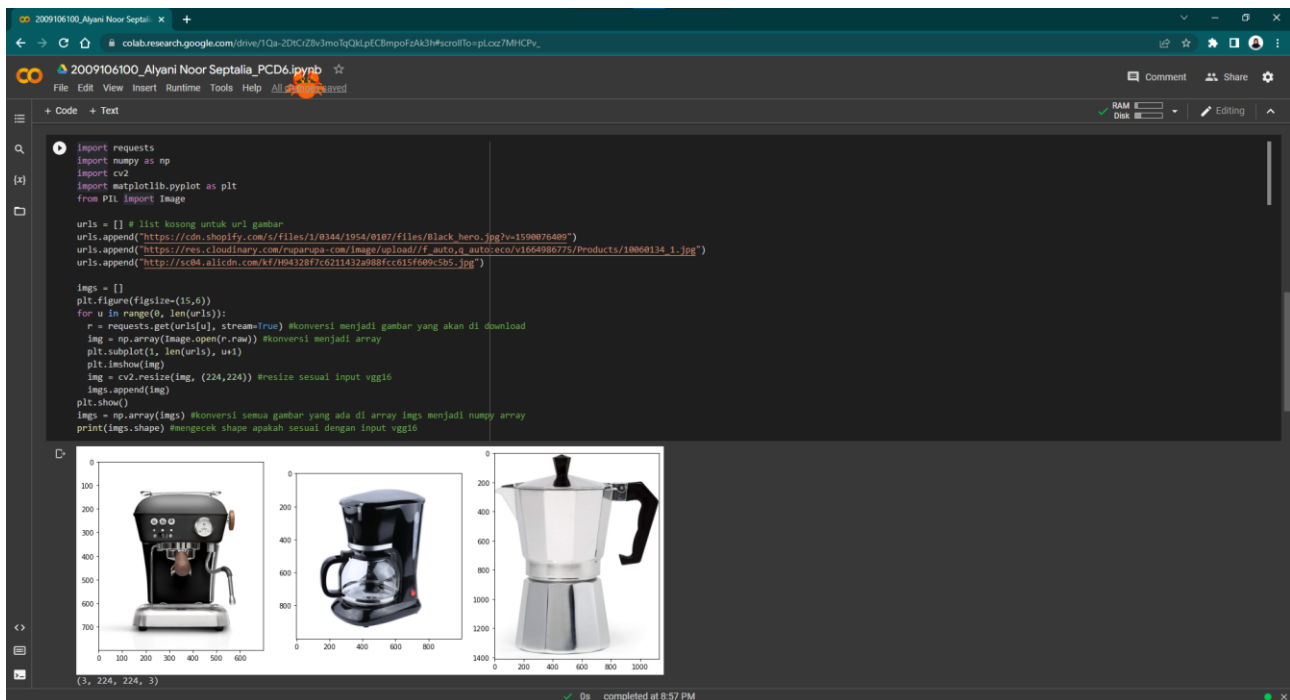
Langkah pertama yang dilakukan kita perlu mengimpor library keras dan model VGG16 untuk pengklasifikasian citra, karena VGG16 merupakan model yang sudah di train 1000 kelas untuk pengklasifikasian gambar, kelas-kelasnya dapat dilihat di <https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a>



```
block5_conv1 (Conv2D)      (None, 14, 14, 512)      2359808
block5_pool (MaxPooling2D) (None, 7, 7, 512)        0
flatten (Flatten)          (None, 25088)            0
fc1 (Dense)                (None, 4096)             102764544
fc2 (Dense)                (None, 4096)             16781312
predictions (Dense)        (None, 1000)             4097000
-----
Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0
```

Dapat dilihat model VGG16 memiliki 138 juta weights yang sudah di train.

2. Import Library



```
import requests
import numpy as np
import cv2
import matplotlib.pyplot as plt
from PIL import Image

urls = [] # list kosong untuk url gambar
urls.append("https://cdn.shopify.com/s/files/1/0344/1954/0187/files/Black_hero.jpg?v=1598076489")
urls.append("https://res.cloudinary.com/ruparupa-com/image/upload/ff_auto,q_auto:eco/v1664986775/Products/10060134-1.jpg")
urls.append("http://sc04.alicdn.com/kf/H0432877c6211432a988fcc615f609c5b5.jpg")

imgs = []
plt.figure(figsize=(15,6))
for u in range(0, len(urls)):
    r = requests.get(urls[u], stream=True) #konversi menjadi gambar yang akan di download
    img = np.array(Image.open(r.raw)) #konversi menjadi array
    plt.subplot(1, len(urls), u+1)
    plt.imshow(img)
    img = cv2.resize(img, (224,224)) #resize sesuai input vgg16
    imgs.append(img)
plt.show()
imgs = np.array(imgs) #konversi semua gambar yang ada di array imgs menjadi numpy array
print(imgs.shape) #mengecek shape apakah sesuai dengan input vgg16
```

(3, 224, 224, 3)

completed at 8:57 PM

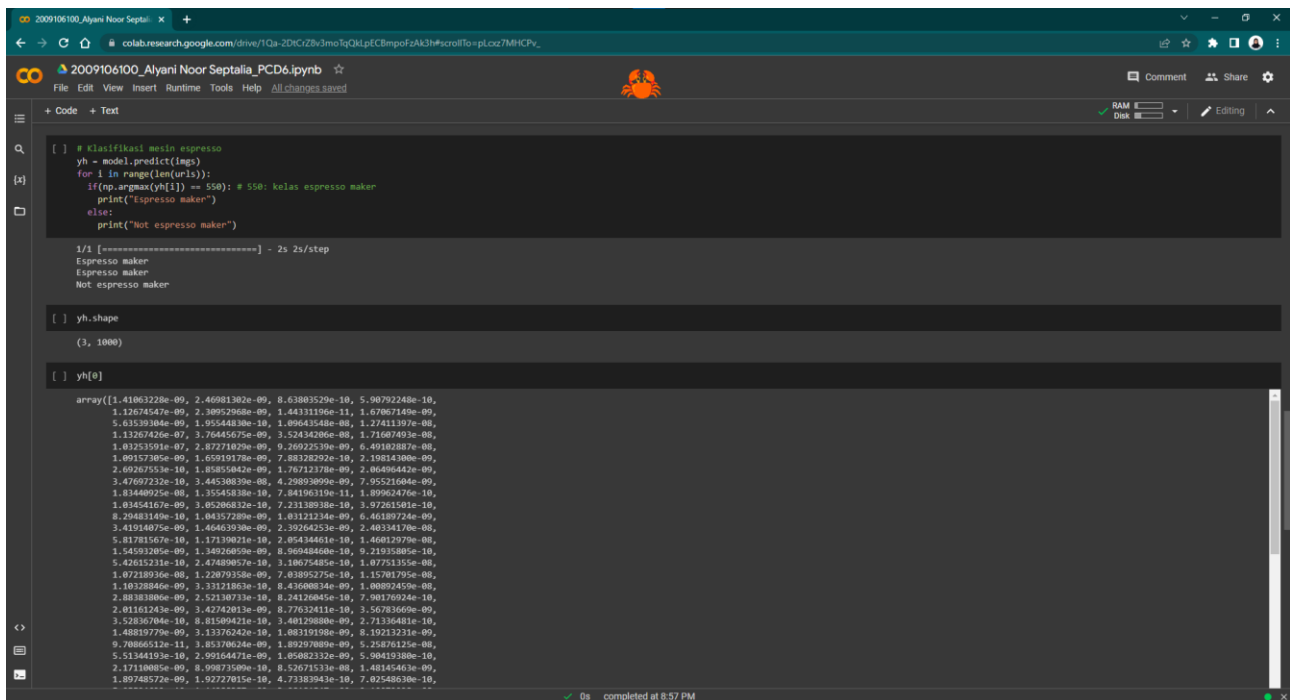
Langkah kedua, kita perlu mengimpor beberapa library tambahan:

- requests : berguna untuk mengambil dan mengunduh gambar dari URL
- numpy : berguna untuk proses array multidimensi
- cv2 : berguna untuk proses resizing
- matplotlib.pyplot : berguna untuk menampilkan gambar
- PIL : berguna untuk operasi gambar

Setelah melakukan impor library, kita membuat array yang akan diisi oleh URL gambar yang digunakan untuk klasifikasi. Setiap URL akan ditambahkan kedalam array menggunakan fungsi append.

Array yang dibuat ada 2 buah, 1 dengan nama variabel urls untuk menampung URL, dan imgs untuk menampung gambar yang sudah dikonversi ke bentuk array. Gambar yang dimasukkan kedalam array imgs merupakan gambar yang sudah di resize sesuai dengan input VGG16.

3. Klasifikasi Citra



```
[ ] # klasifikasi mesin espresso
yh = model.predict(imgs)
for i in range(len(urls)):
    if(np.argmax(yh[i]) == 550): # 550: kelas espresso maker
        print("Espresso maker")
    else:
        print("Not espresso maker")

1/1 [=====] - 2s 2s/step
Espresso maker
Espresso maker
Not espresso maker

[ ] yh.shape
(3, 1000)

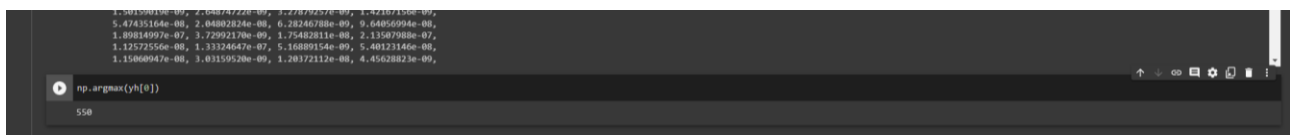
[ ] yh[0]
array([1.41063228e-09, 2.46981302e-09, 8.63803529e-10, 5.96792248e-10,
1.12674547e-09, 2.30952968e-09, 1.44131196e-11, 1.67067149e-09,
5.43539384e-09, 1.95544830e-10, 1.09641548e-08, 1.27411397e-08,
1.13287425e-07, 3.76445675e-09, 3.52834286e-09, 1.71807493e-08,
1.8323591e-07, 2.87271829e-09, 9.26922139e-09, 4.49182887e-08,
1.09157385e-09, 1.65919178e-09, 7.88328292e-10, 1.19814300e-09,
2.09267553e-10, 1.8585842e-09, 1.76712178e-09, 2.06496442e-09,
3.47697232e-10, 3.44538039e-08, 4.29893099e-09, 2.95521604e-09,
1.83480925e-08, 1.35545838e-10, 7.04106319e-11, 1.89962476e-10,
1.03454167e-09, 3.05206832e-10, 7.23138938e-10, 3.97261501e-10,
8.29481140e-10, 1.04357280e-09, 1.8312134e-09, 4.46189724e-09,
3.43514075e-09, 1.46463939e-09, 2.30264535e-09, 2.40334179e-08,
5.81781567e-10, 1.17139821e-10, 2.05434461e-10, 1.46812979e-08,
1.54593285e-09, 1.34926859e-09, 8.96948460e-10, 9.21935805e-10,
5.4055231e-10, 2.47409057e-10, 3.18975485e-10, 1.07751355e-09,
1.07218936e-08, 1.22879358e-09, 7.03895275e-10, 1.15701795e-08,
1.10328846e-09, 3.33121863e-10, 8.43600834e-09, 1.00892459e-08,
2.88181806e-09, 2.52130733e-10, 8.24126045e-10, 7.98176924e-10,
2.01161243e-09, 3.42242033e-09, 8.77032411e-10, 2.56782669e-09,
3.52836704e-10, 8.81509421e-10, 3.40129880e-09, 2.71336481e-10,
1.48819779e-09, 3.13376242e-10, 1.08319198e-09, 8.19213231e-09,
9.70860512e-11, 3.85170624e-09, 1.09207089e-09, 2.23876125e-08,
5.1346153e-10, 2.99164471e-09, 1.05082132e-09, 5.98415388e-10,
2.17110085e-09, 8.99873509e-10, 8.52671533e-08, 1.48145463e-09,
1.89748572e-09, 1.92772015e-10, 4.73383943e-10, 7.02548638e-10,
```

Langkah ketiga, kita mulai membuat kriteria untuk mengklasifikasikan citra. Saya ingin mengklasifikasikan mesin espresso, berdasarkan kelas model VGG16 yang sudah disebutkan di langkah pertama mesin espresso berada di kelas 550. Kelas yang terdeteksi arraynya selain 550 akan dideteksi sebagai bukan mesin espresso.

Setelah melakukan pembuatan kriteria, kita cek lagi shapenya untuk memastikan data yang sudah dicek di variabel yh sama dengan jumlah gambar yang kita masukkan di array urls.

yh[0] berfungsi untuk menampilkan array dari setiap kelas yang ada di model VGG16, array yang memiliki nilai tertinggi merupakan kelas deteksi dari model.

4. Hasil Akhir



```
np.argmax(yh[0])
550
```

Untuk memastikan hasil deteksi, kita gunakan fungsi argmax. Dapat dilihat bahwa nilai array tertinggi berada di kelas 550 yang berarti kelas mesin espresso. Pengklasifikasian citra menggunakan model VGG16 dapat dikatakan akurat.

```
548: 547: 'electric locomotive',
549: 548: 'entertainment center',
550: 549: 'envelope',
551: 550: 'espresso maker',
552: 551: 'face powder',
553: 552: 'feather boa, boa',
554: 553: 'file, file cabinet, filing cabinet',
```