WEEK 3

Date: 13.10.2023 – 20.10.2023

Project Team: Esma Zeynep Sayılı – Esra Cesur

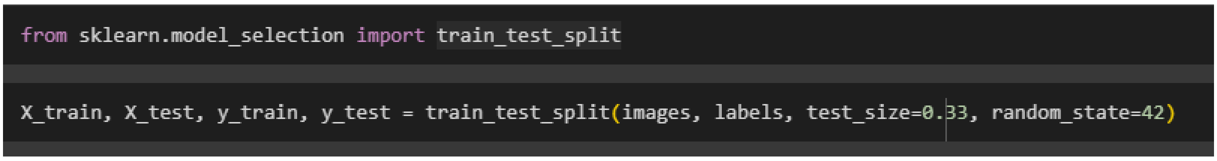
Project Name: Neckline Classifier

Objectives:

1. Preprocessing
2. Model Setup

Preprocessing

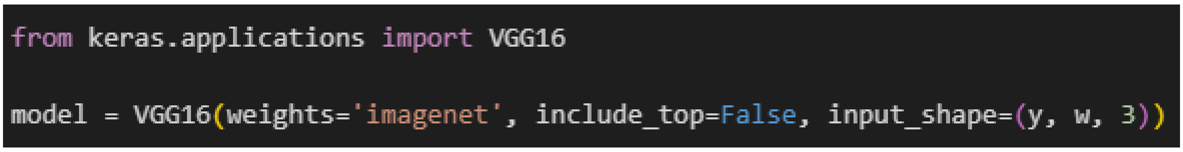
After the preparation of the dataset, we divided our dataset into two different arrays, one containing the images and one containing the labels. Using the “train_test_split” function from the “sklearn.model_selection” module we split our images and labels with 0.33 test split.



```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.33, random_state=42)
```

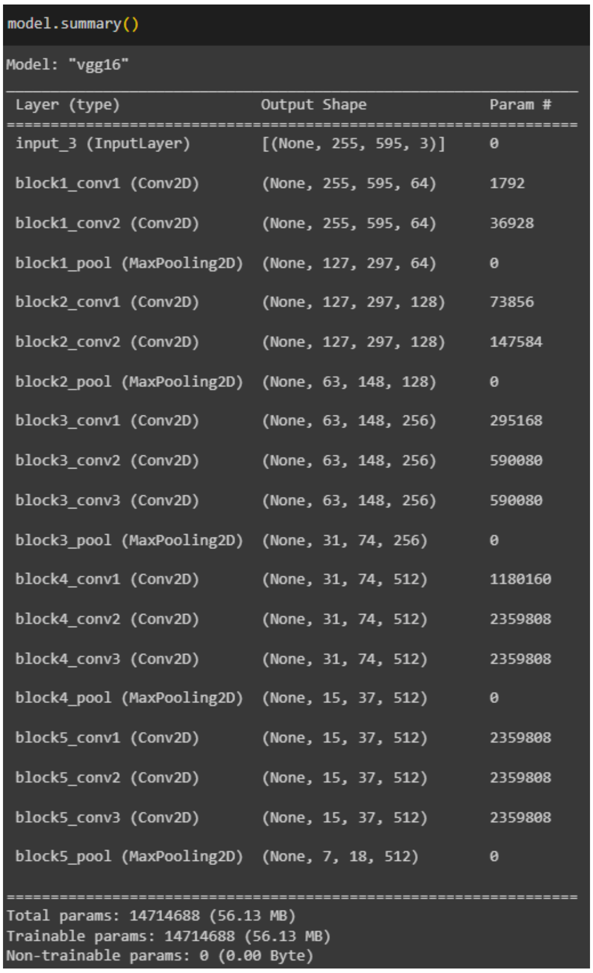
Model Setup

We implemented a VGG16 model, initializing it with pre-trained weights and specifying the input shape with three color channels.



```
from keras.applications import VGG16

model = VGG16(weights='imagenet', include_top=False, input_shape=(y, w, 3))
```

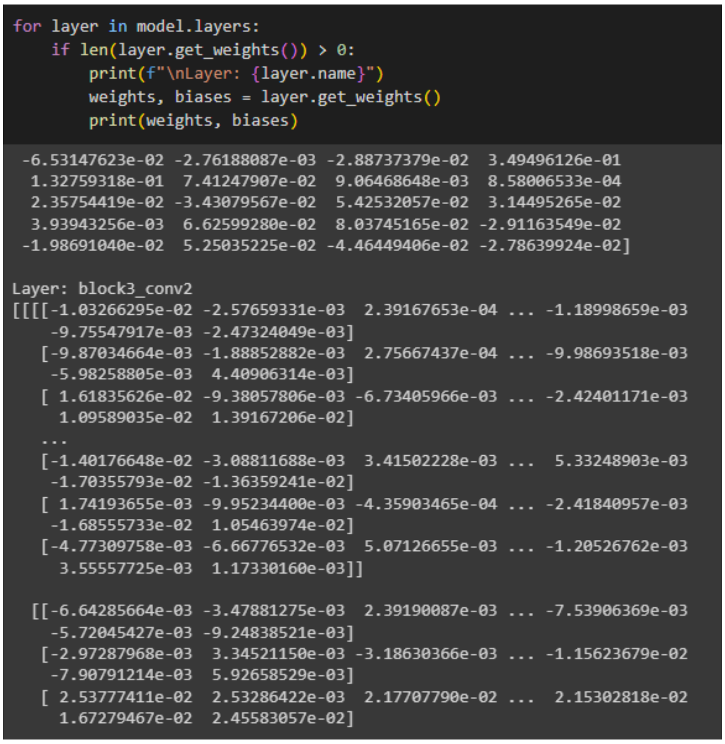
The model summary, revealing its architecture and parameter counts, was displayed to gain insights into its structure.



```
model.summary()
```

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, 255, 595, 3)]     0

 block1_conv1 (Conv2D)       (None, 255, 595, 64)      1792

 block1_conv2 (Conv2D)       (None, 255, 595, 64)      36928

 block1_pool (MaxPooling2D)  (None, 127, 297, 64)      0

 block2_conv1 (Conv2D)       (None, 127, 297, 128)     73856

 block2_conv2 (Conv2D)       (None, 127, 297, 128)     147584

 block2_pool (MaxPooling2D)  (None, 63, 148, 128)      0

 block3_conv1 (Conv2D)       (None, 63, 148, 256)      295168

 block3_conv2 (Conv2D)       (None, 63, 148, 256)      590080

 block3_conv3 (Conv2D)       (None, 63, 148, 256)      590080

 block3_pool (MaxPooling2D)  (None, 31, 74, 256)       0

 block4_conv1 (Conv2D)       (None, 31, 74, 512)       1180160

 block4_conv2 (Conv2D)       (None, 31, 74, 512)       2359808

 block4_conv3 (Conv2D)       (None, 31, 74, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 15, 37, 512)       0

 block5_conv1 (Conv2D)       (None, 15, 37, 512)       2359808

 block5_conv2 (Conv2D)       (None, 15, 37, 512)       2359808

 block5_conv3 (Conv2D)       (None, 15, 37, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 18, 512)        0

=================================================================
Total params: 14714688 (56.13 MB)
Trainable params: 14714688 (56.13 MB)
Non-trainable params: 0 (0.00 Byte)
```

In addition, we meticulously examined the weights and biases of each layer in the model, providing a comprehensive understanding of the learned parameters. This exploration contributes to our comprehension of the model's internal representations and its ability to capture complex features.



```
for layer in model.layers:
    if len(layer.get_weights()) > 0:
        print(f"\nLayer: {layer.name}")
        weights, biases = layer.get_weights()
        print(weights, biases)
```

```
 -6.53147623e-02 -2.76188087e-03 -2.88737379e-02  3.49496126e-01
  1.32759318e-01  7.41247907e-02  9.06468648e-03  8.58006533e-04
  2.35754419e-02 -3.43079567e-02  5.42532057e-02  3.14495265e-02
  3.93943256e-03  6.62599280e-02  8.03745165e-02 -2.91163549e-02
 -1.98691040e-02  5.25035225e-02 -4.46449406e-02 -2.78639924e-02]

Layer: block3_conv2
[[[[-1.03266295e-02 -2.57659331e-03  2.39167653e-04 ... -1.18998659e-03
    -9.75547917e-03 -2.47324049e-03]
   [-9.87034664e-03 -1.88852882e-03  2.75667437e-04 ... -9.98693518e-03
    -5.98258805e-03  4.40906314e-03]
   [ 1.61835626e-02 -9.38057806e-03 -6.73405966e-03 ... -2.42401171e-03
     1.09589035e-02  1.39167206e-02]
   ...
   [-1.40176648e-02 -3.08811688e-03  3.41502228e-03 ...  5.33248903e-03
    -1.70355793e-02 -1.36359241e-02]
   [ 1.74193655e-03 -9.95234400e-03 -4.35903465e-04 ... -2.41840957e-03
    -1.68555733e-02  1.05463974e-02]
   [-4.77309758e-03 -6.66776532e-03  5.07126655e-03 ... -1.20526762e-03
     3.55557725e-03  1.17330160e-03]]

  [[-6.64285664e-03 -3.47881275e-03  2.39190087e-03 ... -7.53906369e-03
    -5.72045427e-03 -9.24838521e-03]
   [-2.97287968e-03  3.34521150e-03 -3.18630366e-03 ... -1.15623679e-02
    -7.90791214e-03  5.92658529e-03]
   [ 2.53777411e-02  2.53286422e-03  2.17707790e-02 ...  2.15302818e-02
     1.67279467e-02  2.45583057e-02]
```