Gesture Detection Models with VGG16 Base

Richeek Das

May 4, 2020

1 Introduction

I was playing around with **Transfer Learning** for my Hand Gesture Detection module, using pretrained convnets like *VGG16, ResNet50, GoogleNET, DenseNet121* while exploring and fine-tuning the hyperparameters to deliver dependable results in real-life applications. So here is a detailed specification of the models with a **VGG16** convolutional base that I had used, along with the *Accuracy and Loss Curves, Confusion Matrices, F1 scores and Error Rates.*

2 Dataset Description

I had *550* different images for each class spread over 5 classes :

Train Set : *375* images per class

Validation Set : *125* images per class

Test Set : *50* images per class (It's pretty less though)

2.1 Data Augmentation used :

```
rescale=1./255
rotation_range=45
width_shift_range=0.2
height_shift_range=0.2
shear_range=0.2
zoom_range=0.2
horizontal_flip=True
fill_mode='nearest'
```

A sample of the dataset(Just to check what it looks like) :

3 Models

Convolution Base has been taken to be the **VGG16** model with an input size of **(224,224,3)** with pretrained weights from **"imagenet"** training set.

3.1 Model 1

Just to get a working baseline with our dataset.

3.1.1 Specifications

Model Classifier Top :

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(5, activation='softmax'))
```

Model Compilation :

```
optimizer = Adam in default
loss = "categorical_crossentropy"
```

Model Fitting :

```
epochs = 50
steps_per_epoch = 100
batch_size = 15
validation_steps = 50
```

3.1.2 Results

Validation Accuracy : 90%

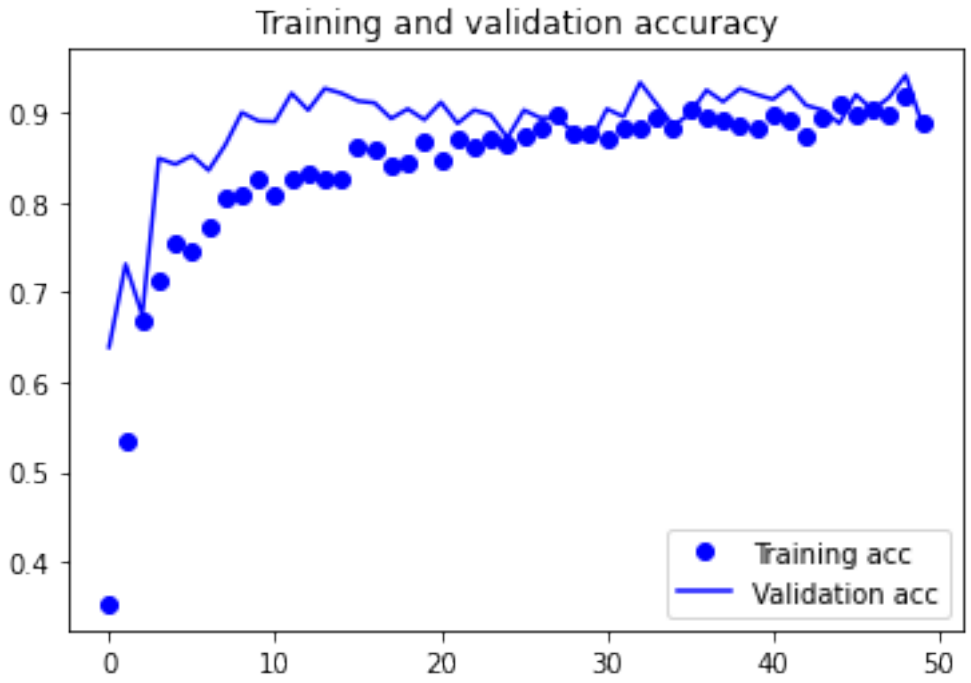
F1 Score : 0.8864

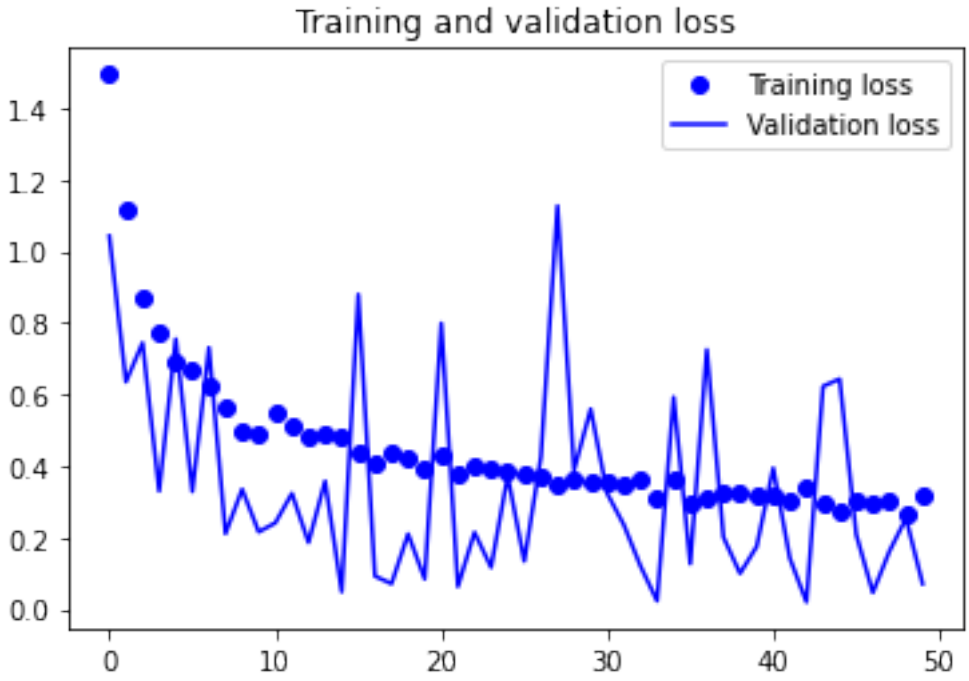
Confusion Matrix :

```
Class Indices : {'cool': 0, 'fist': 1, 'ok': 2, 'stop': 3, 'yo': 4}

array([[26,  0,  4,  3, 17],
       [ 0, 48,  0,  2,  0],
       [ 0,  0, 50,  0,  0],
       [ 0,  0,  0, 50,  0],
       [ 0,  0,  0,  0, 50]], dtype=int64)
```

Seems like it is pretty heavily confusing, “cool” sign with “yo” which is kind of apparently obvious, since both of them have *2* fingers

3.1.3 Accuracy and Loss Curves





3.2 Model 2

Decreased the *learning rate* of the optimizer, increased the *train batch size* from *15* to *32*, set the *validation steps* and *steps per epoch* to something that seemed optimal and increased the *no of epochs* to analyse it further.

3.2.1 Specifications

Model Classifier Top :

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(5, activation='softmax'))
```

Model Compilation :

```
optimizer = Adam(lr=2e-5)
loss = "categorical_crossentropy"
```

Model Fitting :

```
epochs = 100
steps_per_epoch = 120
batch_size = 32
validation_steps = None
```

3.2.2 Results

Validation Accuracy : 93%

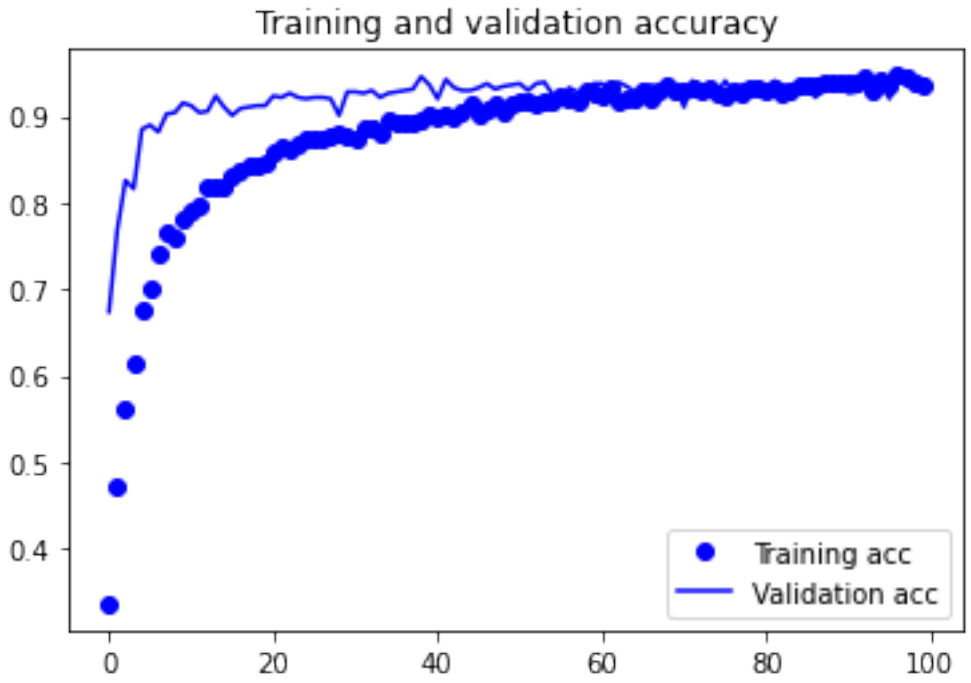
F1 Score : 0.9207

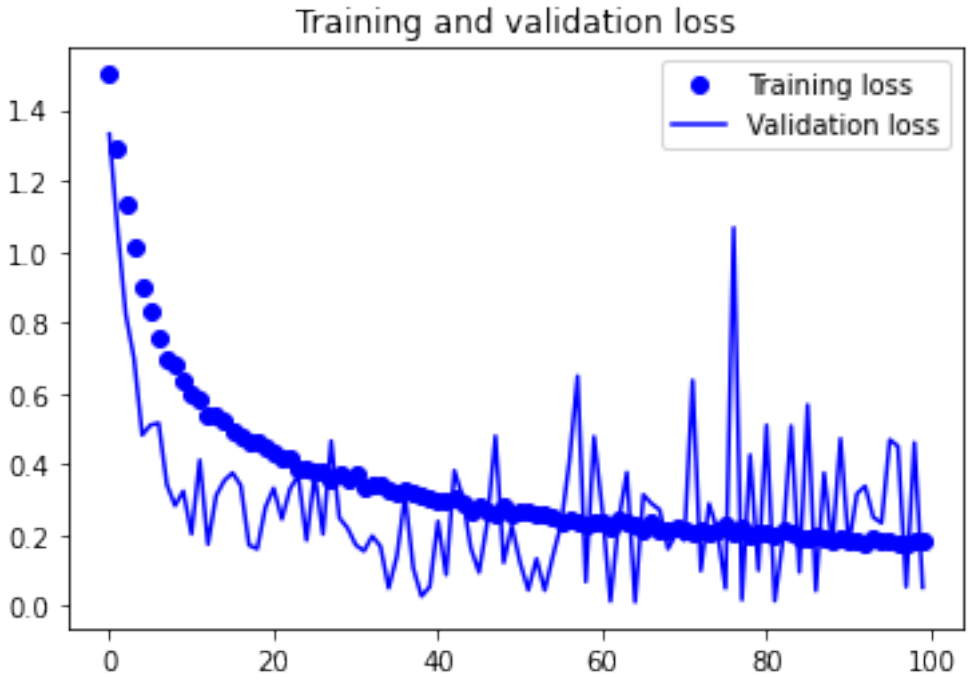
Confusion Matrix :

```
Class Indices : {'cool': 0, 'fist': 1, 'ok': 2, 'stop': 3, 'yo': 4}

array([[35,  0,  4,  4,  7],
       [ 0, 48,  0,  2,  0],
       [ 0,  0, 50,  0,  0],
       [ 0,  0,  0, 50,  0],
       [ 0,  0,  0,  2, 48]], dtype=int64)
```

Seems like it is still heavily confusing, “cool” sign with “yo”.

3.2.3 Accuracy and Loss Curves





3.3 Model 3

Increased the *learning rate* of the optimizer, did major changes to the model classifier(as listed below) and increased the *no of epochs* to analyse it further.

3.3.1 Specifications

Model Classifier Top :

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(5, activation='softmax'))
```

Model Compilation :

```
optimizer = Adam(lr=1e-4)
loss = "categorical_crossentropy"
```

Model Fitting :

```
epochs = 200
steps_per_epoch = 120
batch_size = 32
validation_steps = None
```

3.3.2 Results

Model is **UNDERFITTING** and the *loss* is still pretty high :(

Validation Accuracy : 95%

F1 Score : 0.9429

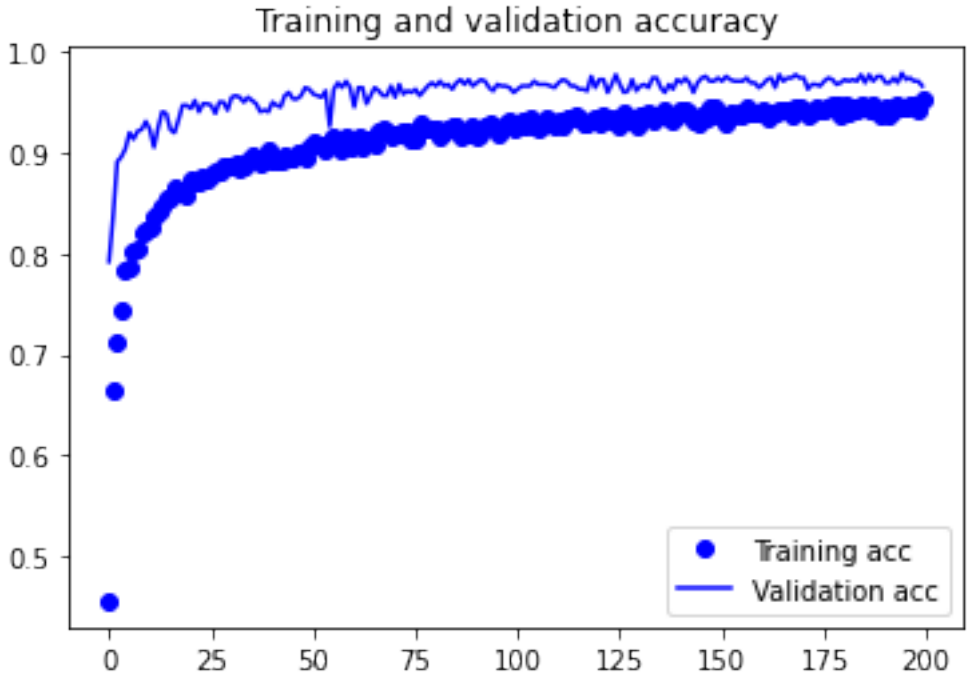
Confusion Matrix :

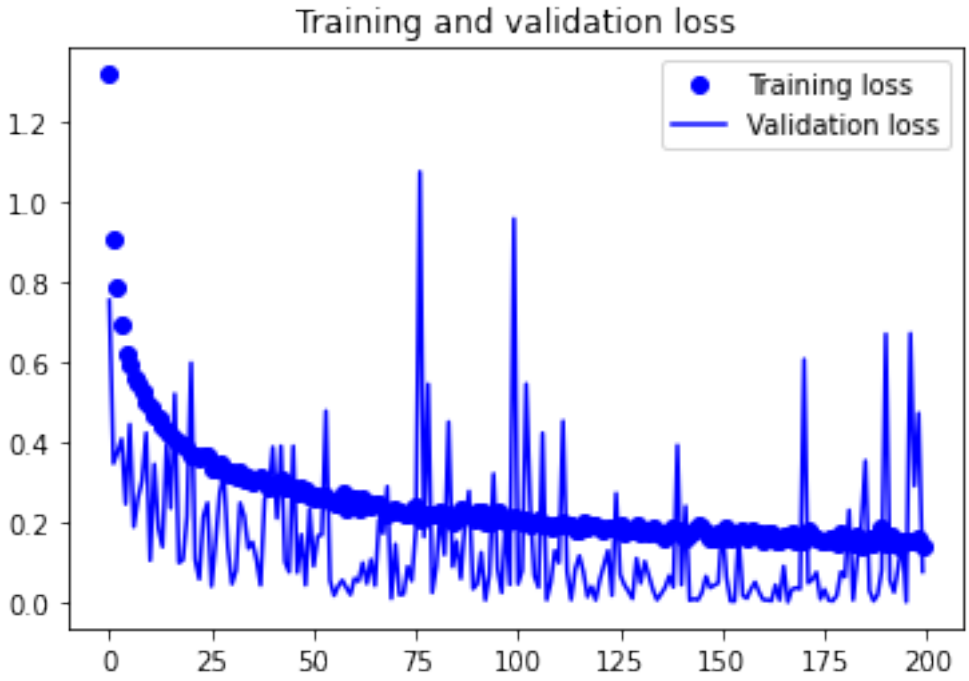
```
Class Indices : {'cool': 0, 'fist': 1, 'ok': 2, 'stop': 3, 'yo': 4}

array([[40,  0,  3,  2,  5],
       [ 0, 49,  0,  1,  0],
       [ 0,  0, 50,  0,  0],
       [ 0,  0,  0, 50,  0],
       [ 0,  0,  0,  3, 47]], dtype=int64)
```

Seems like it is still confusing, “cool” sign with “yo” :(

But first, we need to fight the Underfit problem !

3.3.3 Accuracy and Loss Curves





3.4 Model 4

Removed the *Dropout layer* from the classifier top and further increased the *no of epochs* to analyse it further.

3.4.1 Specifications

Model Classifier Top :

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(5, activation='softmax'))
```

Model Compilation :

```
optimizer = Adam(lr=1e-4)
loss = "categorical_crossentropy"
```

Model Fitting :

```
epochs = 200
steps_per_epoch = 120
batch_size = 32
validation_steps = None
```

3.4.2 Results

Validation Accuracy : 96.8%

F1 Score : 0.9344

Confusion Matrix :

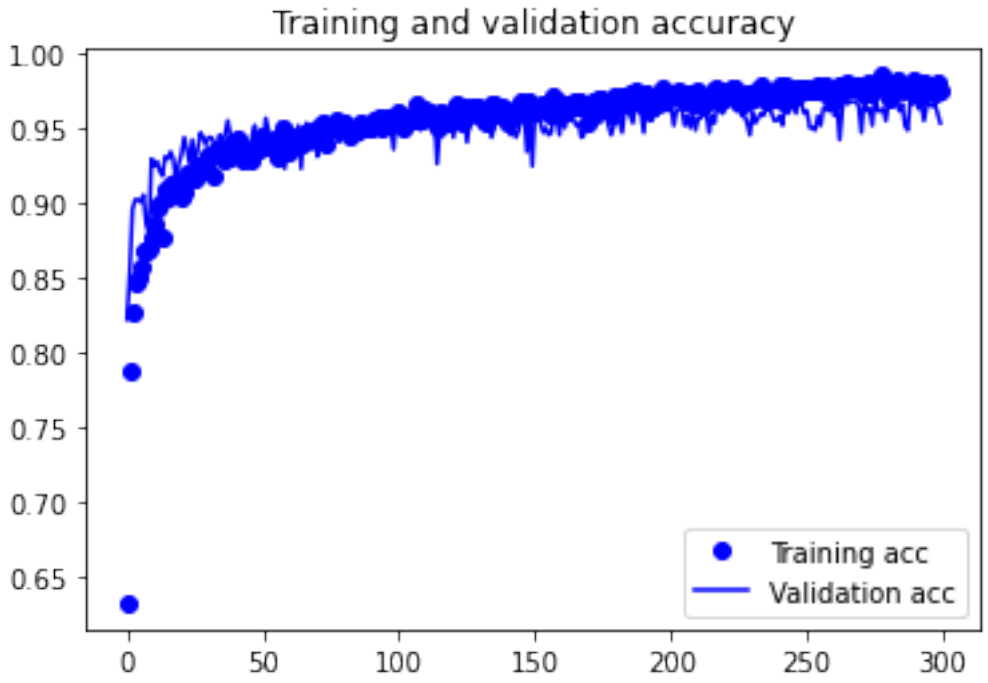
```
Class Indices : {'cool': 0, 'fist': 1, 'ok': 2, 'stop': 3, 'yo': 4}

array([[38,  0,  2,  4,  6],
       [ 0, 47,  0,  3,  0],
       [ 0,  0, 50,  0,  0],
       [ 0,  0,  0, 50,  0],
       [ 0,  0,  0,  1, 49]], dtype=int64)
```

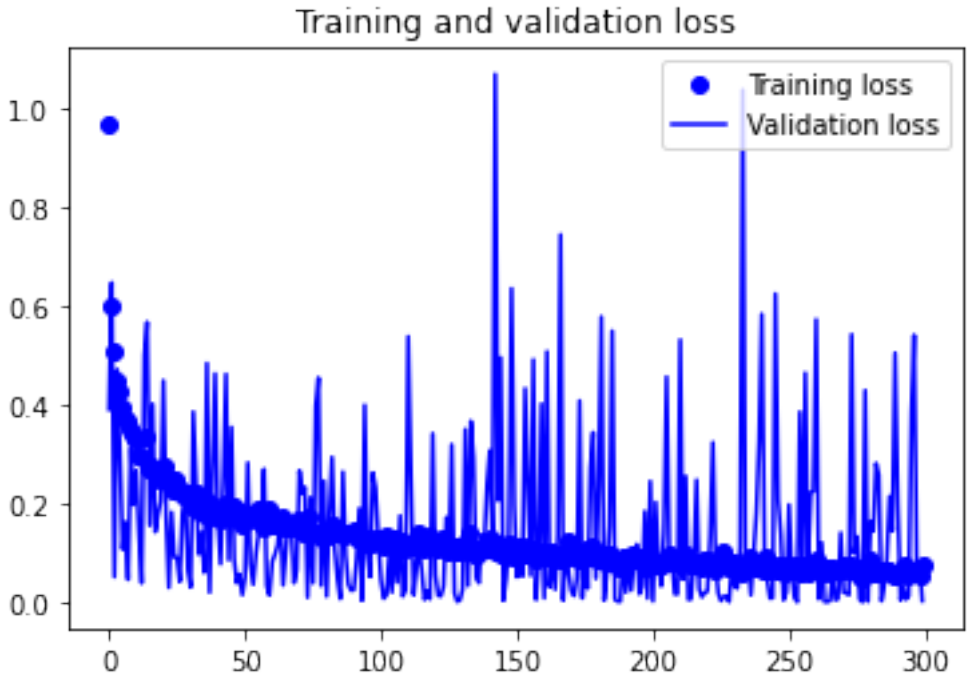
Seems like it is still confusing, "cool" sign with "yo" -_-

But, as we see the Underfit problem has been solved !

Now let's fine tune the final layers of the freezed **VGG16** base to gain a performance boost !

3.4.3 Accuracy and Loss Curves





3.5 Model 4 Fine Tuned

Unfreezed `block5_conv1` layer of `vgg16` and retrained it with a reduced learning rate :

```
epochs = 100
optimizer = optimizers.Adam(lr=1e-4)
loss = "categorical_crossentropy"
```

3.5.1 Results

Validation Accuracy : 98.08%

F1 Score = 0.9959

Confusion Matrix :

```
Class Indices : {'cool': 0, 'fist': 1, 'ok': 2, 'stop': 3, 'yo': 4}

array([[50,  0,  0,  0,  0],
       [ 0, 50,  0,  0,  0],
       [ 0,  0, 50,  0,  0],
       [ 0,  1,  0, 49,  0],
       [ 0,  0,  0,  0, 50]], dtype=int64)
```

Seems like that fixed its confusion :)

3.5.2 Accuracy and Loss Curves