In [1]:

```
#import required packages
import os
import cv2
import numpy as np
import pandas as pd
import seaborn as sn
import tensorflow as tf
import matplotlib.pyplot as plt
```

In [2]:

```
#Set all directories with different test conditions
baseDir = "C:/Users/abhir/Desktop/Fall_2020/ECE_578_Intelligent_Robotics_1/Project/Fina
l1/ConfusionMat"
testDir1 = "Set_Similar_to_Train"
testDir2 = "Abhiraj_Simple_Test"
testDir3 = "Abhiraj_Different_Background_Test"
testDir4 = "Abhiraj_Extreme_Translation_Test"
testDir= [testDir1,testDir2,testDir3,testDir4]
```

In [3]:

```
os.chdir(baseDir)
```

In [4]:

```
#prediction syntax when using tflite model
def getPredict(img, interpreter, input_details, output_details):
    interpreter.set_tensor(input_details[0]['index'], img)
    interpreter.invoke()
    outputData = interpreter.get_tensor(output_details[0]['index'])
    return outputData
    pass
```

In [5]:

```
#preprocessing the image, Same as done while training the model
def preProcess(image):
    reszImg = cv2.resize(image, (224, 224))
    reszImg= cv2.cvtColor(reszImg, cv2.COLOR_BGR2RGB)
    preProsImg = tf.keras.applications.mobilenet.preprocess_input(reszImg)
    reshapImg = preProsImg.reshape(-1,224,224,3)
    return reshapImg
    pass
```

In [6]:

```
#function to create confusion matrix by passing CM array and its labels
def confMat(preds, lables):
    df_cm = pd.DataFrame(np.array(preds), lables, lables)
    plt.figure(figsize=(20,10))
    sn.set(font_scale=1.7) # for label size
    sn.heatmap(df_cm, annot=True, annot_kws={"size": 16}, cmap='Blues', fmt='g') # font
size
    plt.show()
```

In [7]:

```
#Get interpreter details for tflite model
def getIntr(modName):
    interpreter = tf.lite.Interpreter(model_path=modName)
    interpreter.allocate_tensors()
    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()
    return interpreter, input_details, output_details
```

In [8]:

```
#load tflite model to be used for prediction
interMain, inputMain, outputMain = getIntr("mainModel.tflite")
print('Main Model Loaded')
```

```
Main Model Loaded
```

In [9]:

```
#dictionary corresponding to numerical value of the gesture
dectFolder= {"five":5, "four":4, "ok":7, "one":1, "rock":9, "spider_man":8, "three":3,
"thumbsup":6, "two":2, "zero":0}
```

In [10]:

```
#Initialize Comfussion Matrix for all tests with 0 value
predArray= np.zeros((4,10,10))
```

In [11]:

```
#Start predicting the test folders and update corresponding Confusion matrix
#Load Test Folder
for i in range(0,4):
    print(testDir[i])
    #gesture in corresponding test directory
    for subs in os.listdir(testDir[i]):
        print(subs)
        #load images in gesture folder and predict and update confussion matrix
        for images in os.listdir(os.path.join(testDir[i], subs)):
            img = cv2.imread(os.path.join(testDir[i], subs, images))
            prosimg = preProcess(img)
            pred = getPredict(prosimg, interMain, inputMain, outputMain)
            predArray[i][dectFolder[subs]][np.argmax(pred[0])] = predArray[i][dectFolde
r[subs]][np.argmax(pred[0])] + 1
    print("")
```

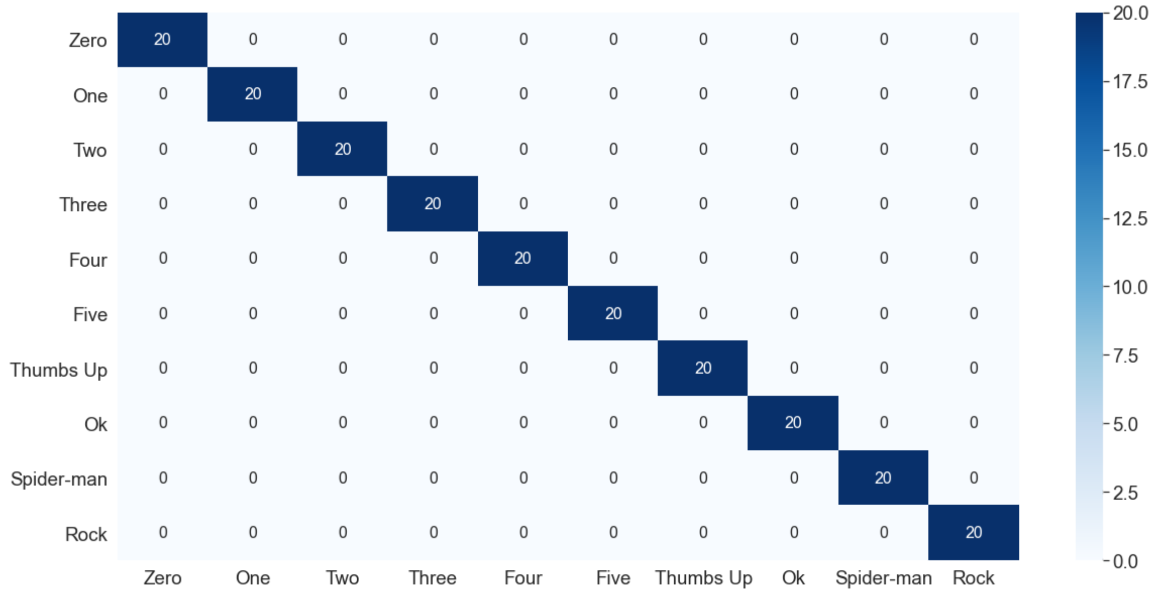
```
Set_Similar_to_Train
five
four
ok
one
rock
spider_man
three
thumbsup
two
zero

Abhiraj_Simple_Test
five
four
ok
one
rock
spider_man
three
thumbsup
two
zero

Abhiraj_Different_Background_Test
five
four
ok
one
rock
spider_man
three
thumbsup
two
zero

Abhiraj_Extreme_Translation_Test
five
four
ok
one
rock
spider_man
three
thumbsup
two
zero
```

In [12]:

```
#label for confussion matrix
lables = ["Zero","One","Two","Three","Four","Five","Thumbs Up","Ok","Spider-man","Rock"
]
```

In [13]:

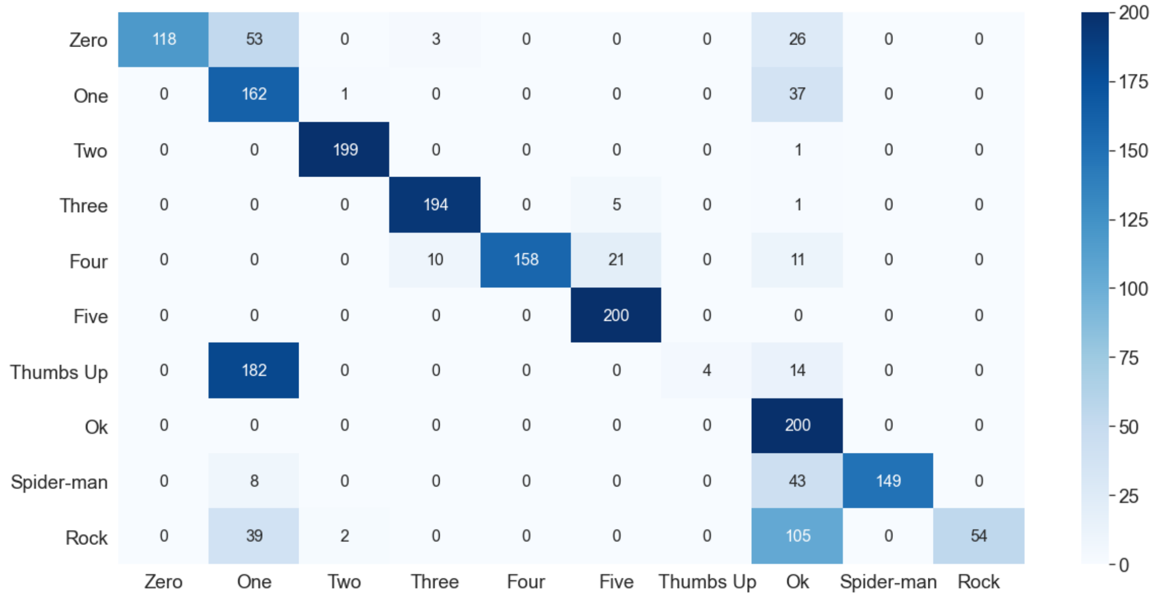
```
#Confussion Matrix for test set 1
confMat(predArray[0], lables)
```



	Zero	One	Two	Three	Four	Five	Thumbs Up	Ok	Spider-man	Rock
Zero	20	0	0	0	0	0	0	0	0	0
One	0	20	0	0	0	0	0	0	0	0
Two	0	0	20	0	0	0	0	0	0	0
Three	0	0	0	20	0	0	0	0	0	0
Four	0	0	0	0	20	0	0	0	0	0
Five	0	0	0	0	0	20	0	0	0	0
Thumbs Up	0	0	0	0	0	0	20	0	0	0
Ok	0	0	0	0	0	0	0	20	0	0
Spider-man	0	0	0	0	0	0	0	0	20	0
Rock	0	0	0	0	0	0	0	0	0	20

In [14]:

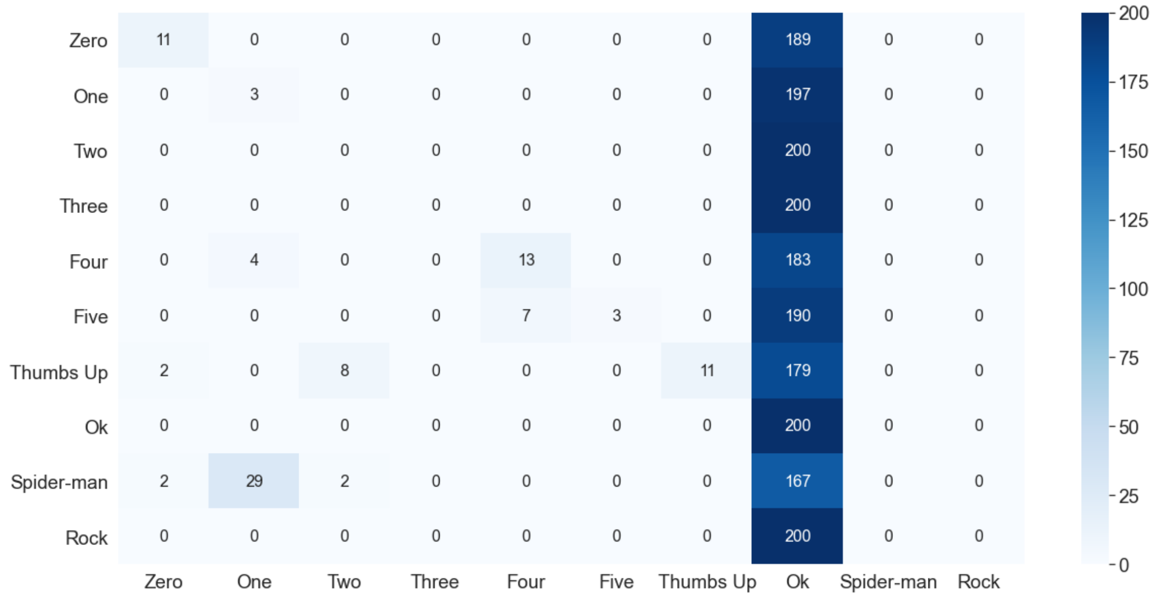
```
#Confussion Matrix for test set 2
confMat(predArray[1], lables)
```



	Zero	One	Two	Three	Four	Five	Thumbs Up	Ok	Spider-man	Rock
Zero	118	53	0	3	0	0	0	26	0	0
One	0	162	1	0	0	0	0	37	0	0
Two	0	0	199	0	0	0	0	1	0	0
Three	0	0	0	194	0	5	0	1	0	0
Four	0	0	0	10	158	21	0	11	0	0
Five	0	0	0	0	0	200	0	0	0	0
Thumbs Up	0	182	0	0	0	0	4	14	0	0
Ok	0	0	0	0	0	0	0	200	0	0
Spider-man	0	8	0	0	0	0	0	43	149	0
Rock	0	39	2	0	0	0	0	105	0	54

In [15]:

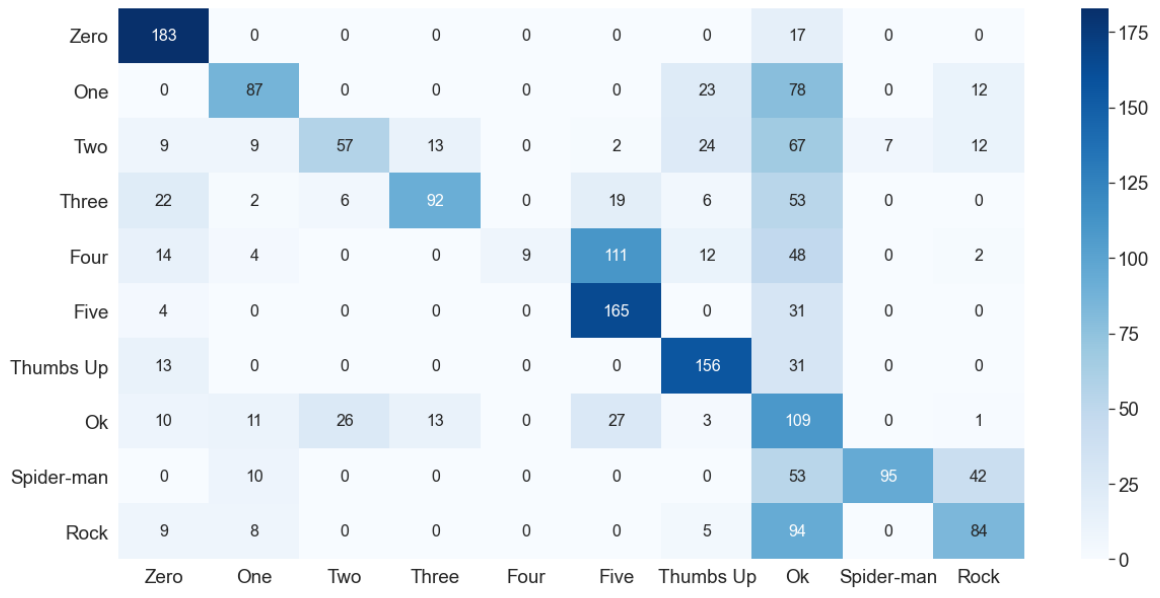
```
#Confussion Matrix for test set 3
confMat(predArray[2], lables)
```



	Zero	One	Two	Three	Four	Five	Thumbs Up	Ok	Spider-man	Rock
Zero	11	0	0	0	0	0	0	189	0	0
One	0	3	0	0	0	0	0	197	0	0
Two	0	0	0	0	0	0	0	200	0	0
Three	0	0	0	0	0	0	0	200	0	0
Four	0	4	0	0	13	0	0	183	0	0
Five	0	0	0	0	7	3	0	190	0	0
Thumbs Up	2	0	8	0	0	0	11	179	0	0
Ok	0	0	0	0	0	0	0	200	0	0
Spider-man	2	29	2	0	0	0	0	167	0	0
Rock	0	0	0	0	0	0	0	200	0	0

In [16]:

```
#Confussion Matrix for test set 4
confMat(predArray[3], lables)
```



	Zero	One	Two	Three	Four	Five	Thumbs Up	Ok	Spider-man	Rock
Zero	183	0	0	0	0	0	0	17	0	0
One	0	87	0	0	0	0	23	78	0	12
Two	9	9	57	13	0	2	24	67	7	12
Three	22	2	6	92	0	19	6	53	0	0
Four	14	4	0	0	9	111	12	48	0	2
Five	4	0	0	0	0	165	0	31	0	0
Thumbs Up	13	0	0	0	0	0	156	31	0	0
Ok	10	11	26	13	0	27	3	109	0	1
Spider-man	0	10	0	0	0	0	0	53	95	42
Rock	9	8	0	0	0	0	5	94	0	84

In [ ]:

```

```