

Project Development Phase

Sprint - 4

Date	20 November 2022
Team ID	PNT2022TMID35856
Project Name	A Gesture - Based Tool for Sterile Browsing of Radiology Images
Marks	4 Marks

IBM Watson Studio:

Sprint 1:

The screenshot displays the IBM Watson Studio interface for a project named 'sprint1'. The top navigation bar includes the IBM Watson Studio logo, a search bar, and user account information (ELAVARASAN P's Account, Frankfurt). The main workspace shows a Jupyter notebook with the following code:

```
In [1]: import os, types
import pandas as pd
from boto3.client import Config
import boto3

def __iter__(self): return 0

#@hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = boto3.client(service_name='s3',
    iam_api_key_id='x25zmx20eltzX2Uccw0ZFa5_14Fk13gX6Vfc42Vs1J-',
    iam_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

bucket = 'sprint1-donotdelete-pr-ry13csz9xkc2qe'
object_key = 'dataset.zip'

streaming_body_5 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a boto3.response.StreamingBody object.
# Please read the documentation of boto3 and pandas to learn more about the possibilities to load the data.
# boto3 documentation: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#python
# pandas documentation: http://pandas.pydata.org/
```

The notebook output shows the following:

```
# Your data file was loaded into a boto3.response.StreamingBody object.
# Please read the documentation of boto3 and pandas to learn more about the possibilities to load the data.
# boto3 documentation: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#python
# pandas documentation: http://pandas.pydata.org/

In [2]: from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_5.read()), 'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)

In [3]: ls

dataset/

In [ ]:
```

Sprint 2:

IBM Watson Studio Search in your workspaces Buy ? ? Ganapathi Subramanian R'... Frankfurt GS

Projects / A Gesture based tool for Sterile ... / handgesture

File Edit View Insert Cell Kernel Help No kernel Not Trusted | Python 3.9

In [3]: **#TRAIN DATA PREPROCESSING**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.25,zoom_range=0.2,horizontal_flip=True)
train_datagen=ImageDataGenerator(rescale=1./255)
train_gen=train_datagen.flow_from_directory(
    'dataset/train',
    target_size=(128,128),
    batch_size=198,
    color_mode='grayscale',
    class_mode='categorical'
)
```

Found 594 images belonging to 6 classes.

In [4]: **#VALIDATION DATA PREPROCESSING**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
valid_datagen=ImageDataGenerator(rescale=1./255)
valid_gen=valid_datagen.flow_from_directory(
    'dataset/test',
    target_size=(128,128),
    batch_size=10,
    color mode='grayscale',
    class_mode='categorical'
)
```

File Edit View Insert Cell Kernel Help Not Trusted | Python 3.9

In [3]: **#TRAIN DATA PREPROCESSING**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.25,zoom_range=0.2,horizontal_flip=True)
train_datagen=ImageDataGenerator(rescale=1./255)
train_gen=train_datagen.flow_from_directory(
    'dataset/train',
    target_size=(128,128),
    batch_size=198,
    color_mode='grayscale',
    class_mode='categorical'
)
```

Found 594 images belonging to 6 classes.

In [4]: **#VALIDATION DATA PREPROCESSING**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
valid_datagen=ImageDataGenerator(rescale=1./255)
valid_gen=valid_datagen.flow_from_directory(
    'dataset/test',
    target_size=(128,128),
    batch_size=10,
    color_mode='grayscale',
    class_mode='categorical'
)
```

Found 30 images belonging to 6 classes.

```
File Edit View Insert Cell Kernel Help Not Trusted Python 3.9
In [5]: #CONVOLUTION NEURAL NETWORK MODEL
import tensorflow as tf
print(tf.__version__)
model=tf.keras.Sequential([
    tf.keras.layers.Conv2D(16,(3,3),activation='relu',input_shape=(128,128,1)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(32,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(16,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512,activation='relu'),
    tf.keras.layers.Dense(6,activation='softmax')
])

2.7.2

In [6]: #LOSS FUNCTION AND OPTIMIZER
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['Accuracy'])

In [7]: model.summary()

Model: "sequential"
Layer (type) Output Shape Param #
=====
conv2d (Conv2D) (None, 126, 126, 16) 160
max_pooling2d (MaxPooling2D) (None, 63, 63, 16) 0
conv2d_1 (Conv2D) (None, 61, 61, 32) 4640
max_pooling2d_1 (MaxPooling2D) (None, 30, 30, 32) 0
conv2d_2 (Conv2D) (None, 28, 28, 16) 4624
max_pooling2d_2 (MaxPooling2D) (None, 14, 14, 16) 0
flatten (Flatten) (None, 3136) 0
dense (Dense) (None, 512) 1606144
dense_1 (Dense) (None, 6) 3078
Total params: 1,618,646
Trainable params: 1,618,646
```

```
File Edit View Insert Cell Kernel Help Not Trusted Python 3.9
Model: "sequential"
Layer (type) Output Shape Param #
=====
conv2d (Conv2D) (None, 126, 126, 16) 160
max_pooling2d (MaxPooling2D) (None, 63, 63, 16) 0
conv2d_1 (Conv2D) (None, 61, 61, 32) 4640
max_pooling2d_1 (MaxPooling2D) (None, 30, 30, 32) 0
conv2d_2 (Conv2D) (None, 28, 28, 16) 4624
max_pooling2d_2 (MaxPooling2D) (None, 14, 14, 16) 0
flatten (Flatten) (None, 3136) 0
dense (Dense) (None, 512) 1606144
dense_1 (Dense) (None, 6) 3078
Total params: 1,618,646
Trainable params: 1,618,646
```

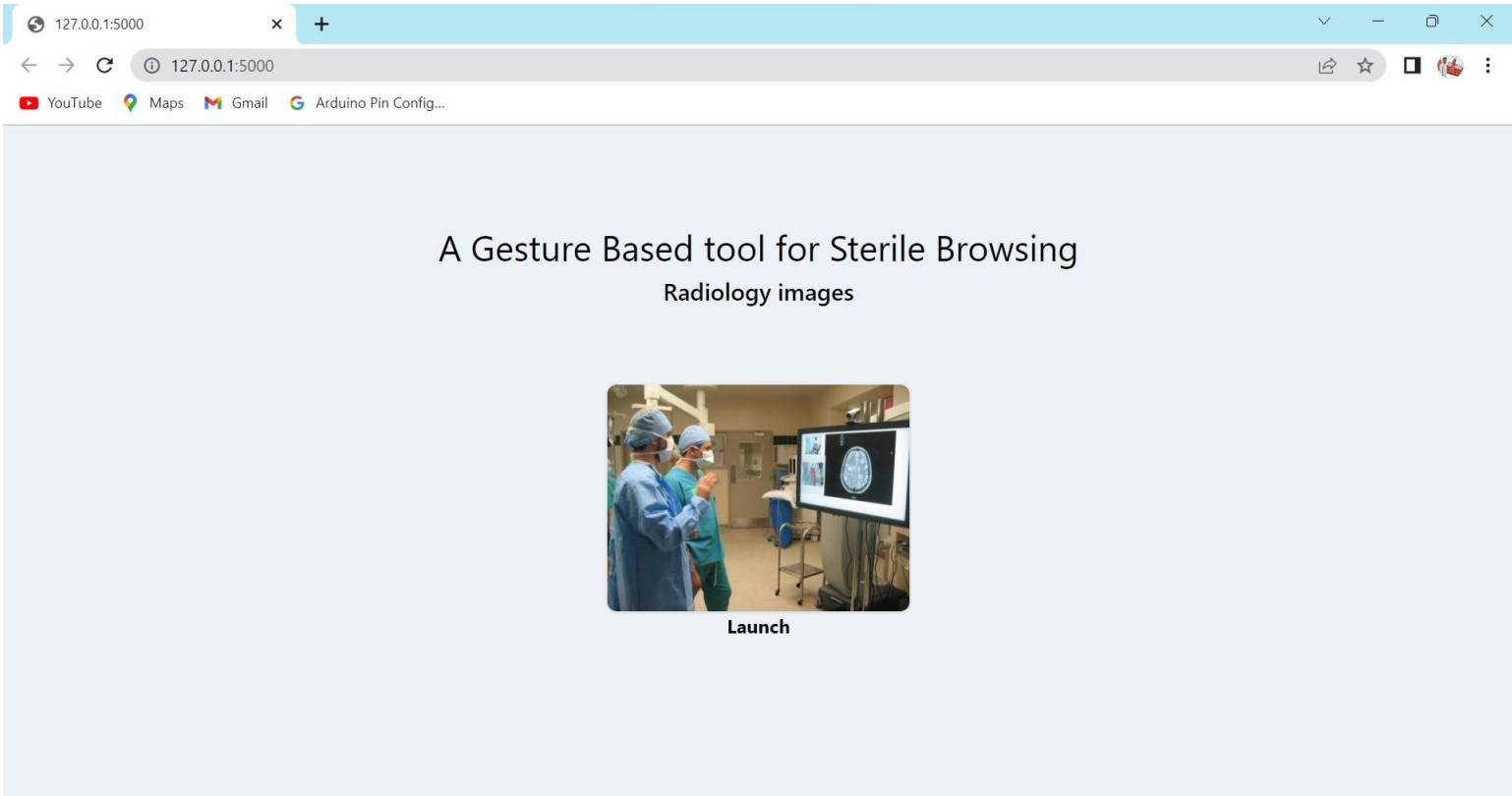
```
In [8]: #TRAINING THE MODEL
trainmodel=model.fit(
    train_gen,
    steps_per_epoch=3,
    epochs=20,
    validation_data=valid_gen,
    validation_steps=3,
)
```

```
Epoch 1/20
3/3 [=====] - 6s 2s/step - loss: 1.7762 - Accuracy: 0.2003 - val_loss: 1.6875 - val_Accuracy: 0.2667
Epoch 2/20
3/3 [=====] - 6s 2s/step - loss: 1.6189 - Accuracy: 0.4327 - val_loss: 1.4747 - val_Accuracy: 0.4000
Epoch 3/20
3/3 [=====] - 6s 2s/step - loss: 1.3585 - Accuracy: 0.6515 - val_loss: 1.2633 - val_Accuracy: 0.5667
Epoch 4/20
3/3 [=====] - 6s 2s/step - loss: 1.0766 - Accuracy: 0.6515 - val_loss: 0.9468 - val_Accuracy: 0.8333
Epoch 5/20
3/3 [=====] - 6s 2s/step - loss: 0.8148 - Accuracy: 0.7273 - val_loss: 0.9268 - val_Accuracy: 0.7000
Epoch 6/20
3/3 [=====] - 6s 2s/step - loss: 0.6414 - Accuracy: 0.7694 - val_loss: 0.6724 - val_Accuracy: 0.7667
Epoch 7/20
3/3 [=====] - 6s 2s/step - loss: 0.4723 - Accuracy: 0.8519 - val_loss: 0.6585 - val_Accuracy: 0.6667
Epoch 8/20
3/3 [=====] - 6s 2s/step - loss: 0.3796 - Accuracy: 0.8636 - val_loss: 0.5784 - val_Accuracy: 0.8667
Epoch 9/20
3/3 [=====] - 6s 2s/step - loss: 0.2983 - Accuracy: 0.9226 - val_loss: 0.7214 - val_Accuracy: 0.7333
```

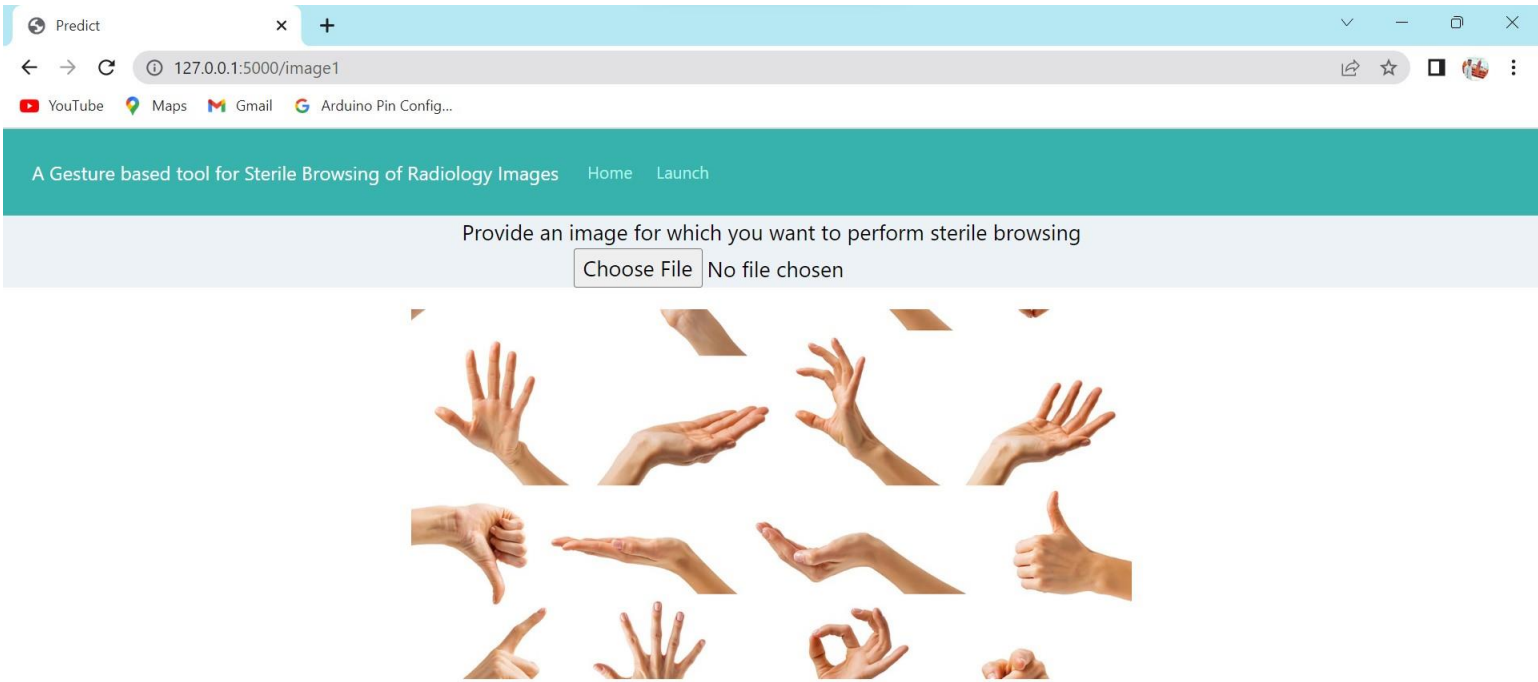
```
Epoch 10/20
3/3 [=====] - 6s 2s/step - loss: 0.2561 - Accuracy: 0.9242 - val_loss: 0.5249 - val_Accuracy: 0.8667
Epoch 11/20
3/3 [=====] - 6s 2s/step - loss: 0.2160 - Accuracy: 0.9276 - val_loss: 0.6461 - val_Accuracy: 0.7667
Epoch 12/20
3/3 [=====] - 6s 2s/step - loss: 0.1949 - Accuracy: 0.9293 - val_loss: 0.7166 - val_Accuracy: 0.7667
Epoch 13/20
3/3 [=====] - 6s 2s/step - loss: 0.1745 - Accuracy: 0.9461 - val_loss: 0.6339 - val_Accuracy: 0.8333
Epoch 14/20
3/3 [=====] - 6s 2s/step - loss: 0.1319 - Accuracy: 0.9613 - val_loss: 0.5087 - val_Accuracy: 0.9000
Epoch 15/20
3/3 [=====] - 6s 2s/step - loss: 0.1055 - Accuracy: 0.9663 - val_loss: 0.7516 - val_Accuracy: 0.8000
Epoch 16/20
3/3 [=====] - 6s 2s/step - loss: 0.0914 - Accuracy: 0.9731 - val_loss: 0.5083 - val_Accuracy: 0.9000
Epoch 17/20
3/3 [=====] - 6s 2s/step - loss: 0.0810 - Accuracy: 0.9764 - val_loss: 0.7712 - val_Accuracy: 0.8333
Epoch 18/20
3/3 [=====] - 6s 2s/step - loss: 0.0508 - Accuracy: 0.9899 - val_loss: 0.5598 - val_Accuracy: 0.9000
```

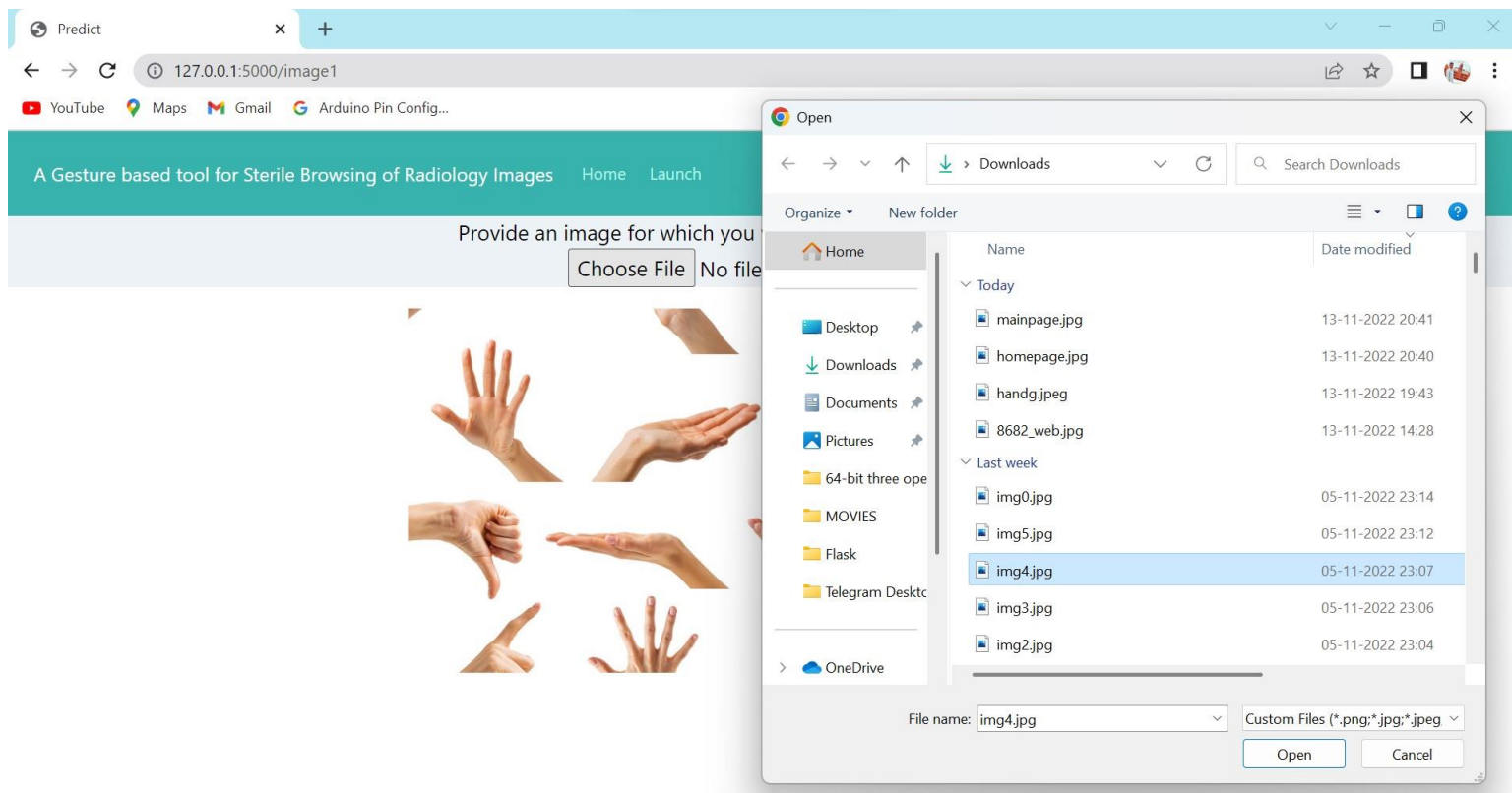
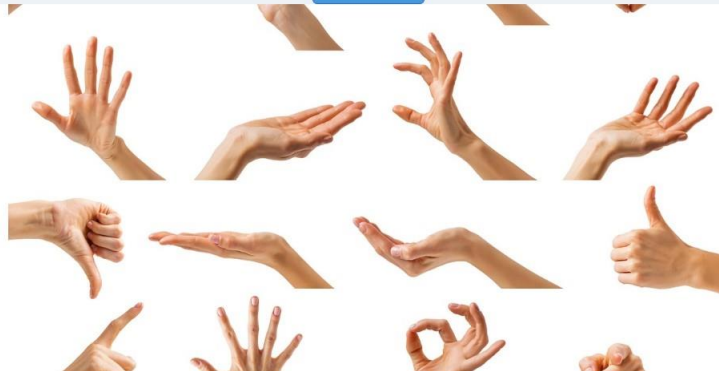
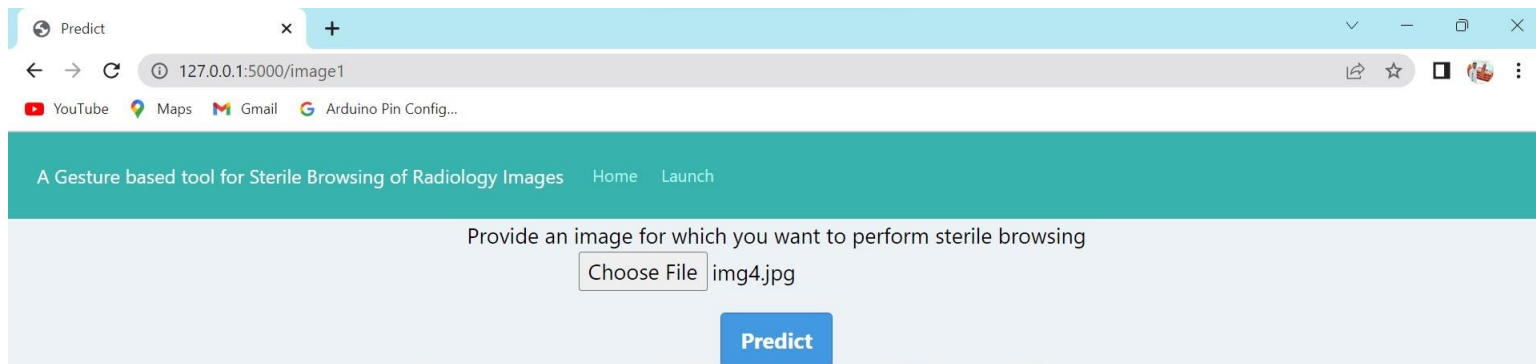
Sprint 3:

Home Page:



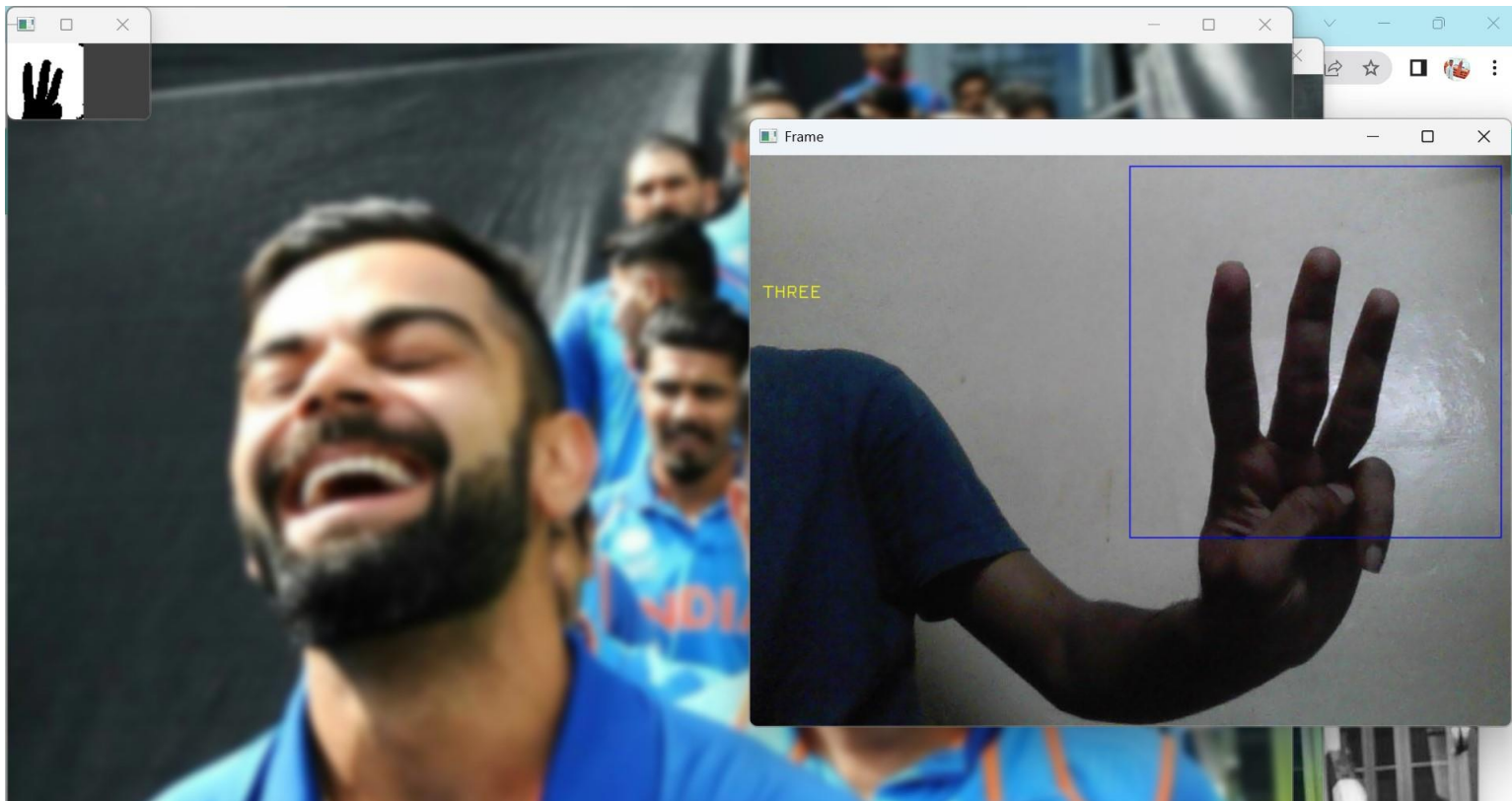
Main page:





Prediction:

For checking correctness, "Three" gesture is directed to blur the input image



Sprint 4:

Increasing Dataset:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img
datagen = ImageDataGenerator(rotation_range=30, horizontal_flip=True, fill_mode='nearest')

for ix in range(1, 100):
    img = load_img('/content/drive/MyDrive/handgesturedataset/dataset/train/5/' + str(ix) + '.jpg')
    x = img_to_array(img)
    x = x.reshape((1,) + x.shape)
    i = 1;
    for batch in datagen.flow(x, save_to_dir='/content/drive/MyDrive/handgesturedataset/dataset/train/tr_5', save_prefix=str(ix), save_format='jpg'):
        i += 1;
        if i > 4:
            break
```

```

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255)#Normalisation
#Preprocessing the Training dataset
train_gen=train_datagen.flow_from_directory(
    'dataset/train',
    #Image size:128*128
    target_size=(128,128),
    batch_size=198,
    #Train Dataset has Grayscale images
    color_mode='grayscale',
    #Since output has classification class_mode='categorical'
    class_mode='categorical'
)

```

Found 2376 images belonging to 6 classes.

Since the dataset is increased, there is a considerable increase in accuracy

Cloud Deployment:

```
In [27]: model.save('ibmhandgesture.h5')
```

```
In [28]: !tar -zcvf hand_gesture-model_new.tgz ibmhandgesture.h5
        ibmhandgesture.h5
```

```
In [29]: ls
        dataset/  hand_gesture-model_new.tgz  ibmhandgesture.h5
```

```
!pip install watson-machine-learning-client --upgrade
```

```

Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)

```



```
#https://{instance_ID}.eu-de.secrets-manager.appdomain.cloud
from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://eu-de.ml.cloud.ibm.com",
    "apikey": "wdHAVMdZd2o-0w7rFdGYfDffZi71oX49zHdBfxEu2SgV"
}
client=APIClient(wml_credentials)
```

```
In [32]: def guid_from_space_name(client, space_name):
         space=client.spaces.get_details()
         return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])
```

```
In [33]: space_uid=guid_from_space_name(client,'handgesture')
         print("Space UID=",space_uid)
```

Space UID= c0fab0c-57ce-4797-9ee9-2466ed198029

```
In [34]: client.set.default_space(space_uid)
```

Out[34]: 'SUCCESS'

```
In [35]: client.software_specifications.list()
```

NAME	ASSET_ID	TYPE
default_py3.6	0062b8c9-8b7d-44a0-a9b9-46c416adcbd9	base
kernel-spark3.2-scala2.12	020d69ce-7ac1-5e68-ac1a-31189867356a	base
pytorch-onnx_1.3-py3.7-edt	069ea134-3346-5748-b513-49120e15d288	base
scikit-learn_0.20-py3.6	09c5a1d0-9c1e-4473-a344-eb7b665ff687	base
spark-mllib_3.0-scala_2.12	09f4cff0-90a7-5899-b9ed-1ef348aebdee	base
pytorch-onnx_rt22.1-py3.9	0b848dd4-e681-5599-be41-b5f6fccc6471	base
ai-function_0.1-py3.6	0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda	base
shiny-r3.6	0e6e79df-875e-4f24-8ae9-62dcc2148306	base
tensorflow_2.4-py3.7-horovod	1092590a-307d-563d-9b62-4eb7d64b3f22	base

```

In [36]: software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid

Out[36]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [38]: model_details=client.repository.store_model(model="hand_gesture-model_new.tgz",meta_props={
    client.repository.ModelMetaNames.NAME:"ibmhandgesture",
    client.repository.ModelMetaNames.TYPE:"tensorflow_rt22.1",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})

model_id=client.repository.get_model_id(model_details)

In [40]: client.repository.download(model_id, "hand_gesture-model_new_cloud.tgz")

Successfully saved model content to file: 'hand_gesture-model_new_cloud.tgz'

Out[40]: '/home/wsuser/work/hand_gesture-model_new_cloud.tgz'

In [41]: ls

dataset/                                hand_gesture-model_new.tgz
hand_gesture-model_new_cloud.tgz      ibmhandgesture.h5

In [42]: from tensorflow.python.keras.models import load_model
model_body = load_model('ibmhandgesture.h5')

In [45]: from keras.models import load_model
from keras.preprocessing import image

In [47]: model=load_model('ibmhandgesture.h5')

In [50]: import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='x25zmx20eltzX2Ucocw0ZFaS_14Fkl3gX6Vfc42Vs1J-',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

bucket = 'sprint1-donotdelete-pr-ryi3csz9xkc2qe'
object_key = '1.zip'

streaming_body_6 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

```

```
In [52]: from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_6.read()), 'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```
In [53]: ls

1.jpg      hand_gesture-model_new_cloud.tgz  ibmhandgesture.h5
dataset/   hand_gesture-model_new.tgz
```

```
In [55]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs=mpimg.imread(path)
imgplot=plt.imshow(imgs)
plt.show()
```



```
In [68]: img=image.load_img(path,color_mode='grayscale',target_size=(128,128))
         #image to array
         x=image.img_to_array(img)
```

```
In [69]: x.shape
```

```
Out[69]: (128, 128, 1)
```

```
In [70]: import numpy as np
         x=np.expand_dims(x,axis=0)
         x.shape
```

```
Out[70]: (1, 128, 128, 1)
```

```
In [71]: pred=np.argmax(model.predict(x),axis=-1)
         #predicting the class
         index=['0','1','2','3','4','5']
```

```
In [72]: pred
```

```
Out[72]: array([2])
```

```
In [75]: result=str(index[pred[0]])
```

```
In [76]: result
```

```
Out[76]: '2'
```

```
In [78]: test_img = []

         for i in range(0,6):
             for j in range(0,5):
                 path = "dataset/test/"+str(i)+"/"+str(j)+".jpg"
                 img = image.load_img(path,color_mode = "grayscale",target_size= (128,128))
                 x = image.img_to_array(img)
                 x = np.expand_dims(x,axis = 0)
                 pred = np.argmax(model.predict(x), axis=-1)
                 test_img.append(pred)
         print(test_img)

         [array([0]), array([0]), array([0]), array([0]), array([0]), array([1]), array([1]), array([1]), array([1]), array([1]), array([2]), array([2]), array
         ([1]), array([2]), array([2]), array([3]), array([2]), array([3]), array([4]), array([3]), array([4]), array([4]), array([4]), array([4]), array([4]),
         array([5]), array([5]), array([5]), array([5]), array([5])]
```

```
In [80]: test_res = []
         index=['0','1','2','3','4','5']
         for i in test_img:
             test_res.append(index[i[0]])
         print(test_res)

         ['0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '2', '2', '1', '2', '2', '3', '2', '3', '4', '3', '4', '4', '4', '4', '4', '4', '5', '5', '5', '5', '5']
```

```
In [ ]:
```