TEAM_ID : PNT2022TMID18715

A Gesture-Based Tool For Sterile Browsing Of Radiology Images

In [1]:
```
pwd
```

Out[1]:
```
'/home/wsuser/work'
```

In [2]:
```
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 cr
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='uB6f-I-uqznB_Fx8kPELA2HUEtTn9fiTSyy0_EqV9XQD',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'agesturebasedtoolforsterilebrowsi-donotdelete-pr-jbt66tuaggweyo'
object_key = 'Dataset.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilit
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [3]:
```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

Model Building

Importing libraries

In [4]:
```
# This library helps add support for large, multi-dimensional arrays and matrices
import numpy as np
#open source used for both ML and DL for computation
import tensorflow as tf
#it is a plain stack of layers
from tensorflow.keras.models import Sequential
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten, Dropout
#Faltten-used fot flattening the input or change the dimension, MaxPooling2D-for downsampl
```

```python
from tensorflow.keras.layers import Convolution2D,MaxPooling2D
#Its used for different augmentation of the image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Augmenting the data

In [5]:

```python
#setting parameter for Image Data agumentation to the traing data
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

Loading our data and performing data agumentation

In [7]:

```python
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/train',
                                            target_size=(64, 64),
                                            batch_size=3,
                                            color_mode='grayscale',
                                            class_mode='categorical')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(r'/home/wsuser/work/Dataset/test',
                                            target_size=(64, 64),
                                            batch_size=3,
                                            color_mode='grayscale',
                                            class_mode='categorical')
```

```
Found 594 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

In [8]:

```python
print(x_train.class_indices)#checking the number of classes
```

```
{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5}
```

Model Creation

In [9]:

```python
# Initializing the CNN
model = Sequential()
```

In [10]:

```python
# First convolution layer and pooling
model.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [11]:

```python
# Second convolution layer and pooling
model.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [12]:

```python
# Flattening the layers i.e. input layer
model.add(Flatten())
```

In [13]:

```python
# Adding a fully connected layer, i.e. Hidden Layer
model.add(Dense(units=512 , activation='relu'))
```

In [14]:

```python
# softmax for categorical analysis, Output Layer
model.add(Dense(units=6, activation='softmax'))
```

In [15]:

```python
model.summary()#summary of our model
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 6)                 3078

=================================================================
Total params: 3,224,422
Trainable params: 3,224,422
Non-trainable params: 0
_________________________________________________________________
```

Model Compilation

In [16]:

```
# Compiling the CNN
# categorical_crossentropy for more than 2
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Model fitting

In [17]:

```
# It will generate packets of train and test data for training
model.fit_generator(x_train,
                    steps_per_epoch = 594/3 ,
                    epochs = 25,
                    validation_data = x_test,
                    validation_steps = 30/3 )
```

```
/tmp/wsuser/ipykernel_229/804983804.py:2: UserWarning: `Model.fit_generator` is deprecat
ed and will be removed in a future version. Please use `Model.fit`, which supports gener
ators.
  model.fit_generator(x_train,
Epoch 1/25
198/198 [==============================] - 8s 37ms/step - loss: 1.3086 - accuracy: 0.463
0 - val_loss: 0.7836 - val_accuracy: 0.7000
Epoch 2/25
198/198 [==============================] - 7s 38ms/step - loss: 0.5869 - accuracy: 0.769
4 - val_loss: 0.7839 - val_accuracy: 0.7333
Epoch 3/25
198/198 [==============================] - 7s 37ms/step - loss: 0.3572 - accuracy: 0.856
9 - val_loss: 0.4616 - val_accuracy: 0.8667
Epoch 4/25
198/198 [==============================] - 7s 37ms/step - loss: 0.2665 - accuracy: 0.893
9 - val_loss: 0.4134 - val_accuracy: 0.9000
Epoch 5/25
198/198 [==============================] - 7s 37ms/step - loss: 0.2197 - accuracy: 0.929
3 - val_loss: 0.3990 - val_accuracy: 0.9333
Epoch 6/25
198/198 [==============================] - 7s 37ms/step - loss: 0.1428 - accuracy: 0.946
1 - val_loss: 0.4722 - val_accuracy: 0.8667
Epoch 7/25
198/198 [==============================] - 7s 37ms/step - loss: 0.1337 - accuracy: 0.952
9 - val_loss: 0.2617 - val_accuracy: 0.9333
Epoch 8/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0763 - accuracy: 0.979
8 - val_loss: 0.4079 - val_accuracy: 0.9000
Epoch 9/25
198/198 [==============================] - 7s 36ms/step - loss: 0.1163 - accuracy: 0.951
2 - val_loss: 0.5630 - val_accuracy: 0.8667
Epoch 10/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0754 - accuracy: 0.974
7 - val_loss: 0.2370 - val_accuracy: 0.9333
Epoch 11/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0931 - accuracy: 0.957
9 - val_loss: 0.1992 - val_accuracy: 0.9667
Epoch 12/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0183 - accuracy: 0.994
9 - val_loss: 0.1605 - val_accuracy: 0.9667
Epoch 13/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0321 - accuracy: 0.994
9 - val_loss: 0.3648 - val_accuracy: 0.9333
Epoch 14/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0156 - accuracy: 0.994
9 - val_loss: 0.2921 - val_accuracy: 0.9667
Epoch 15/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0380 - accuracy: 0.986
5 - val_loss: 0.3984 - val_accuracy: 0.9667
Epoch 16/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0933 - accuracy: 0.971
4 - val_loss: 0.2182 - val_accuracy: 0.9333
Epoch 17/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0457 - accuracy: 0.986
5 - val_loss: 0.3118 - val_accuracy: 0.9333
Epoch 18/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0405 - accuracy: 0.988
2 - val_loss: 0.2521 - val_accuracy: 0.9667
Epoch 19/25
198/198 [==============================] - 7s 38ms/step - loss: 0.0828 - accuracy: 0.978
1 - val_loss: 0.6236 - val_accuracy: 0.8333
Epoch 20/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0679 - accuracy: 0.979
8 - val_loss: 0.2793 - val_accuracy: 0.9333
Epoch 21/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0198 - accuracy: 0.993
3 - val_loss: 0.2393 - val_accuracy: 0.9667
Epoch 22/25
198/198 [==============================] - 7s 38ms/step - loss: 0.0532 - accuracy: 0.986
5 - val_loss: 0.2143 - val_accuracy: 0.9667
```

```
Epoch 23/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0388 - accuracy: 0.986
5 - val_loss: 0.2496 - val_accuracy: 0.9667
Epoch 24/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0311 - accuracy: 0.986
5 - val_loss: 0.2877 - val_accuracy: 0.9333
Epoch 25/25
198/198 [==============================] - 7s 37ms/step - loss: 0.0253 - accuracy: 0.994
9 - val_loss: 0.2049 - val_accuracy: 0.9667
```

Out[17]:

```
<keras.callbacks.History at 0x7fe20402cfd0>
```

Saving model

In [18]:

```python
# Save the model
model.save('gesture.h5')
```

In [19]:

```
!tar -zcvf Gesture-based-Radiology-Images.tgz gesture.h5
```

```
gesture.h5
```

In [20]:

```python
model_json = model.to_json()
with open("model-bw.json", "w") as json_file:
    json_file.write(model_json)
```

IBM Deployment

In [21]:

```
!pip install ibm_watson_machine_learning
```

```
Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/l
ib/python3.9/site-packages (1.0.257)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-pa
ckages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-
packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-pa
ckages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/pyt
hon3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python
3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/pytho
n3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-pac
kages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/py
thon3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)
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==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.
9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.1
1.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/l
ib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson
_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/si
te-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/s
ite-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->
ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/si
te-packages (from requests->ibm_watson_machine_learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/li
b/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-
packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/
python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
```

In [23]:

```
from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"ExiP7ZwnKnNTejcaqe-eP5uyIuWbgoc2OBNSnsTDm7Kp"
}

client=APIClient(wml_credentials)
```

In [24]:

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

In [25]:

```
space_uid=guid_space_name(client,'models')
print("Space UID "+space_uid)
```

```
Space UID 0df2e4a6-9f00-423f-9618-5de134f5771c
```

In [26]:

```
client.set.default_space(space_uid)
```

Out[26]:

```
'SUCCESS'
```

In [27]:

```
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
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9             12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6        154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6         1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6           1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt  1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6           1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda        26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                       295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8            2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6          2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9          2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6              2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt     32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37           36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6             39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt     40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9         4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                 42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                   493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6          49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6             4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6         50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8            52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11     55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                 5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1              5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                     5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc         5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  base
-----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [32]:

```
software_spec_uid = client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid
```

Out[32]:

```
'12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

In [33]:

```
model_details = client.repository.store_model(model="Gesture-based-Radiology-Images.tgz",
    client.repository.ModelMetaNames.NAME: "CNN",
    client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
})

model_id = client.repository.get_model_id(model_details)
```

In [34]:
```
model_id
```

Out[34]:
```
'90d61dc3-9e76-4c54-b578-8d6f4a5017b5'
```

In [36]:
```
client.repository.download(model_id,"my_model.tar.gz")
```

```
File with name: 'my_model.tar.gz' already exists.
---------------------------------------------------------------------------
WMLClientError                            Traceback (most recent call last)
/tmp/wsuser/ipykernel_229/3354860318.py in <module>
----> 1 client.repository.download(model_id,"my_model.tar.gz")

/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm_watson_machine_learning/reposit
ory.py in download(self, artifact_uid, filename, rev_uid, format)
    901
    902         if res['model'] is True:
--> 903             return self._client._models.download(artifact_uid, filename, rev_uid,f
ormat)
    904         elif res['function'] is True:
    905             return self._client._functions.download(artifact_uid, filename, rev_ui
d)

/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm_watson_machine_learning/models.
py in download(self, model_uid, filename, rev_uid, format)
   1803         """
   1804         if os.path.isfile(filename):
-> 1805             raise WMLClientError(u'File with name: \'{}\' already exists.'.format(
filename))
   1806         if rev_uid is not None and self._client.ICP_30 is None and not self._clien
t.CLOUD_PLATFORM_SPACES\
   1807                 and self._client.ICP_35 is None and self._client.ICP_40 is None an
d self._client.ICP_45 is None \

WMLClientError: File with name: 'my_model.tar.gz' already exists.
```

In [ ]:

In [ ]: