

```
In [1]: import os

print(os.getcwd())
```

C:\Users\afm

```
In [3]: import os

print(os.listdir())
```

```
['.anaconda', '.conda', '.condarc', '.config', '.continuum', '.ipynb_checkpoints',
'.ipython', '.jupyter', '.matplotlib', '.ms-ad', '.streamlit', '.virtual_documents',
'anaconda3', 'anaconda_projects', 'AppData', 'Application Data', 'CarPrice_Assignmen
t.csv', 'carprice_dataset_dictionary.csv', 'Contacts', 'Cookies', 'Data-Melbourne_F_
fixed.csv', 'Diabetes Missing Data.csv', 'Documents', 'Downloads', 'Favorites', 'gym
1', 'gym1.csv', 'Gym_Exercise_Classification.ipynb', 'Gym_Report.docx', 'heart.csv',
'level_count.png', 'Links', 'Local Settings', 'megaGymDataset.csv', 'Music', 'My Doc
uments', 'NetHood', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2', 'NTUSER.DAT
{e4763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.TxR.0.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c
-11f0-92d9-fcb3aad5f9d9}.TxR.1.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb
3aad5f9d9}.TxR.2.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.Tx
R.blf', 'NTUSER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TM.blf', 'NTUSER.DAT{e4763
a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer000000000000000001.regtrans-ms', 'NTUS
ER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer000000000000000002.regtran
s-ms', 'ntuser.ini', 'OneDrive', 'PCManager', 'PrintHood', 'Project 1', 'Project 1.ip
ynb', 'rating_box.png', 'rating_hist.png', 'Recent', 'Saved Games', 'Searches', 'Sen
dTo', 'Start Menu', 'Templates', 'type_pie.png', 'Untitled.ipynb', 'Untitled1.ipyn
b', 'Untitled2.ipynb', 'Untitled3.ipynb', 'Untitled4.ipynb', 'Untitled5.ipynb', 'Unt
itled6.ipynb', 'Untitled7.ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb', 'Videos']
```

```
In [4]: import os

os.makedirs("dataset", exist_ok=True)
print(os.listdir())
```

```
['.anaconda', '.conda', '.condarc', '.config', '.continuum', '.ipynb_checkpoints',
'.ipython', '.jupyter', '.matplotlib', '.ms-ad', '.streamlit', '.virtual_documents',
'anaconda3', 'anaconda_projects', 'AppData', 'Application Data', 'CarPrice_Assignmen
t.csv', 'carprice_dataset_dictionary.csv', 'Contacts', 'Cookies', 'Data-Melbourne_F_
fixed.csv', 'dataset', 'Diabetes Missing Data.csv', 'Documents', 'Downloads', 'Favor
ites', 'gym1', 'gym1.csv', 'Gym_Exercise_Classification.ipynb', 'Gym_Report.docx',
'heart.csv', 'level_count.png', 'Links', 'Local Settings', 'megaGymDataset.csv', 'Mu
sic', 'My Documents', 'NetHood', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2',
'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.TxR.0.regtrans-ms', 'NTUSER.DAT{e4
763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.TxR.1.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11
f0-92d9-fcb3aad5f9d9}.TxR.2.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb3aa
d5f9d9}.TxR.blf', 'NTUSER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TM.blf', 'NTUSE
R.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer000000000000000001.regtrans
-ms', 'NTUSER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer000000000000000
002.regtrans-ms', 'ntuser.ini', 'OneDrive', 'PCManager', 'PrintHood', 'Project 1', 'P
roject 1.ipynb', 'rating_box.png', 'rating_hist.png', 'Recent', 'Saved Games', 'Sear
ches', 'SendTo', 'Start Menu', 'Templates', 'type_pie.png', 'Untitled.ipynb', 'Untit
led1.ipynb', 'Untitled2.ipynb', 'Untitled3.ipynb', 'Untitled4.ipynb', 'Untitled5.ipyn
b', 'Untitled6.ipynb', 'Untitled7.ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb', 'Vi
deos']
```

In [5]:

```
import os

os.makedirs("dataset", exist_ok=True)
print(os.listdir())
```

```
['.anaconda', '.conda', '.condarc', '.config', '.continuum', '.ipynb_checkpoints',
'.ipython', '.jupyter', '.matplotlib', '.ms-ad', '.streamlit', '.virtual_documents',
'anaconda3', 'anaconda_projects', 'AppData', 'Application Data', 'CarPrice_Assignmen
t.csv', 'carprice_dataset_dictionary.csv', 'Contacts', 'Cookies', 'Data-Melbourne_F_
fixed.csv', 'dataset', 'Diabetes Missing Data.csv', 'Documents', 'Downloads', 'Favor
ites', 'gym1', 'gym1.csv', 'Gym_Exercise_Classification.ipynb', 'Gym_Report.docx',
'heart.csv', 'level_count.png', 'Links', 'Local Settings', 'megaGymDataset.csv', 'Mu
sic', 'My Documents', 'NetHood', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2',
'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.TxR.0.regtrans-ms', 'NTUSER.DAT{e4
763a78-7a7c-11f0-92d9-fcb3aad5f9d9}.TxR.1.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11
f0-92d9-fcb3aad5f9d9}.TxR.2.regtrans-ms', 'NTUSER.DAT{e4763a78-7a7c-11f0-92d9-fcb3aa
d5f9d9}.TxR.blf', 'NTUSER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TM.blf', 'NTUSE
R.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer000000000000000001.regtrans
-ms', 'NTUSER.DAT{e4763a79-7a7c-11f0-92d9-fcb3aad5f9d9}.TMContainer0000000000000000
002.regtrans-ms', 'ntuser.ini', 'OneDrive', 'PCManager', 'PrintHood', 'Project 1', 'P
roject 1.ipynb', 'rating_box.png', 'rating_hist.png', 'Recent', 'Saved Games', 'Sear
ches', 'SendTo', 'Start Menu', 'Templates', 'type_pie.png', 'Untitled.ipynb', 'Untit
led1.ipynb', 'Untitled2.ipynb', 'Untitled3.ipynb', 'Untitled4.ipynb', 'Untitled5.ipyn
b', 'Untitled6.ipynb', 'Untitled7.ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb', 'Vi
deos']
```

In [6]:

```
import os

dataset_path = "dataset"

for folder in os.listdir(dataset_path):
    print(folder)
```

In []:

In [7]:

```
import os

dataset_path = "dataset"

for folder in os.listdir(dataset_path):
    print(folder)
```

In [10]:

```
import os

dataset_path = "dataset"

for folder in os.listdir(dataset_path):
    print(folder)
```

```
EfficientNetB0-100-(224 X 224)- 98.40.h5
sports.csv
test
train
valid
```

```
In [11]: import os

train_path = "dataset/train"

classes = os.listdir(train_path)

print("Number of Classes:", len(classes))
print(classes)
```

Number of Classes: 100

```
['air hockey', 'ampute football', 'archery', 'arm wrestling', 'axe throwing', 'balance beam', 'barell racing', 'baseball', 'basketball', 'baton twirling', 'bike polo', 'billiards', 'bmx', 'bobsled', 'bowling', 'boxing', 'bull riding', 'bungee jumping', 'canoe slamon', 'cheerleading', 'chuckwagon racing', 'cricket', 'croquet', 'curling', 'disc golf', 'fencing', 'field hockey', 'figure skating men', 'figure skating pairs', 'figure skating women', 'fly fishing', 'football', 'formula 1 racing', 'frisbee', 'gaga', 'giant slalom', 'golf', 'hammer throw', 'hang gliding', 'harness racing', 'high jump', 'hockey', 'horse jumping', 'horse racing', 'horseshoe pitching', 'hurdles', 'hydroplane racing', 'ice climbing', 'ice yachting', 'jai alai', 'javelin', 'jousting', 'judo', 'lacrosse', 'log rolling', 'luge', 'motorcycle racing', 'mushin', 'nascar racing', 'olympic wrestling', 'parallel bar', 'pole climbing', 'pole dancing', 'pole vault', 'polo', 'pommel horse', 'rings', 'rock climbing', 'roller derby', 'rollerblade racing', 'rowing', 'rugby', 'sailboat racing', 'shot put', 'shuffleboard', 'sidecar racing', 'ski jumping', 'sky surfing', 'skydiving', 'snowboarding', 'snowmobile racing', 'speed skating', 'steer wrestling', 'sumo wrestling', 'surfing', 'swimming', 'table tennis', 'tennis', 'track bicycle', 'trapeze', 'tug of war', 'ultimate', 'uneven bars', 'volleyball', 'water cycling', 'water polo', 'weightlifting', 'wheelchair basketball', 'wheelchair racing', 'wingsuit flying']
```

```
In [12]: import os

train_path = "dataset/train"

for cls in os.listdir(train_path)[:10]:
    cls_path = os.path.join(train_path, cls)

    if os.path.isdir(cls_path):
        print(cls, len(os.listdir(cls_path)))
```

```
air hockey 112
ampute football 112
archery 132
arm wrestling 99
axe throwing 113
balance beam 147
barell racing 123
baseball 174
basketball 169
baton twirling 108
```

```
In [13]: import matplotlib.pyplot as plt
from PIL import Image
import random
import os

train_path = "dataset/train"
```

```
class_name = random.choice(os.listdir(train_path))

image_name = random.choice(
    os.listdir(os.path.join(train_path, class_name))
)

img_path = os.path.join(
    train_path,
    class_name,
    image_name
)

img = Image.open(img_path)

plt.imshow(img)
plt.title(class_name)
plt.axis("off")
plt.show()
```

wheelchair basketball



```
In [14]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [15]: train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    horizontal_flip=True,
    width_shift_range=0.2,
    height_shift_range=0.2
)
```

```
valid_datagen = ImageDataGenerator(
    rescale=1./255
)

test_datagen = ImageDataGenerator(
    rescale=1./255
)
```

```
In [16]: IMG_SIZE = (224,224)
        BATCH_SIZE = 32
```

```
In [17]: train_generator = train_datagen.flow_from_directory(
        "dataset/train",
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode="categorical"
    )

    valid_generator = valid_datagen.flow_from_directory(
        "dataset/valid",
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode="categorical"
    )

    test_generator = test_datagen.flow_from_directory(
        "dataset/test",
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode="categorical",
        shuffle=False
    )
```

Found 13492 images belonging to 100 classes.

Found 500 images belonging to 100 classes.

Found 500 images belonging to 100 classes.

```
In [18]: from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Conv2D, MaxPooling2D
        from tensorflow.keras.layers import Flatten, Dense, Dropout

        model = Sequential([

            Conv2D(32, (3,3), activation='relu',
                input_shape=(224,224,3)),
            MaxPooling2D(2,2),

            Conv2D(64, (3,3), activation='relu'),
            MaxPooling2D(2,2),

            Conv2D(128, (3,3), activation='relu'),
            MaxPooling2D(2,2),

            Flatten(),

            Dense(256, activation='relu'),
```

```
Dropout(0.5),  
  
Dense(100, activation='softmax')  
)  
  
model.compile(  
    optimizer='adam',  
    loss='categorical_crossentropy',  
    metrics=['accuracy']  
)  
  
model.summary()
```

C:\Users\afm\anaconda3\envs\gymcnn\lib\site-packages\keras\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an `input\_shape`/`input\_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

WARNING:tensorflow:TensorFlow GPU support is not available on native Windows for TensorFlow >= 2.11. Even if CUDA/cuDNN are installed, GPU will not be used. Please use WSL2 or the TensorFlow-DirectML plugin.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 256)	22,151,424
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 100)	25,700

Total params: 22,270,372 (84.95 MB)

Trainable params: 22,270,372 (84.95 MB)

Non-trainable params: 0 (0.00 B)

```
In [22]: history = model.fit(
          train_generator,
          validation_data=valid_generator,
```

```

    epochs=3,
    steps_per_epoch=50,
    validation_steps=10
)

```

Epoch 1/3

50/50 ————— 25s 490ms/step - accuracy: 0.2031 - loss: 3.1934 - val_accuracy: 0.3406 - val_loss: 2.6763

Epoch 2/3

50/50 ————— 24s 490ms/step - accuracy: 0.1956 - loss: 3.2283 - val_accuracy: 0.3250 - val_loss: 2.7274

Epoch 3/3

50/50 ————— 23s 465ms/step - accuracy: 0.2294 - loss: 3.1934 - val_accuracy: 0.2969 - val_loss: 2.7305

```

In [23]: test_loss, test_accuracy = model.evaluate(
        test_generator,
        steps=10
    )

```

```

print("Test Accuracy:", test_accuracy)
print("Test Loss:", test_loss)

```

10/10 ————— 4s 425ms/step - accuracy: 0.3625 - loss: 2.6954

Test Accuracy: 0.36250001192092896

Test Loss: 2.6953916549682617

```

In [24]: import matplotlib.pyplot as plt

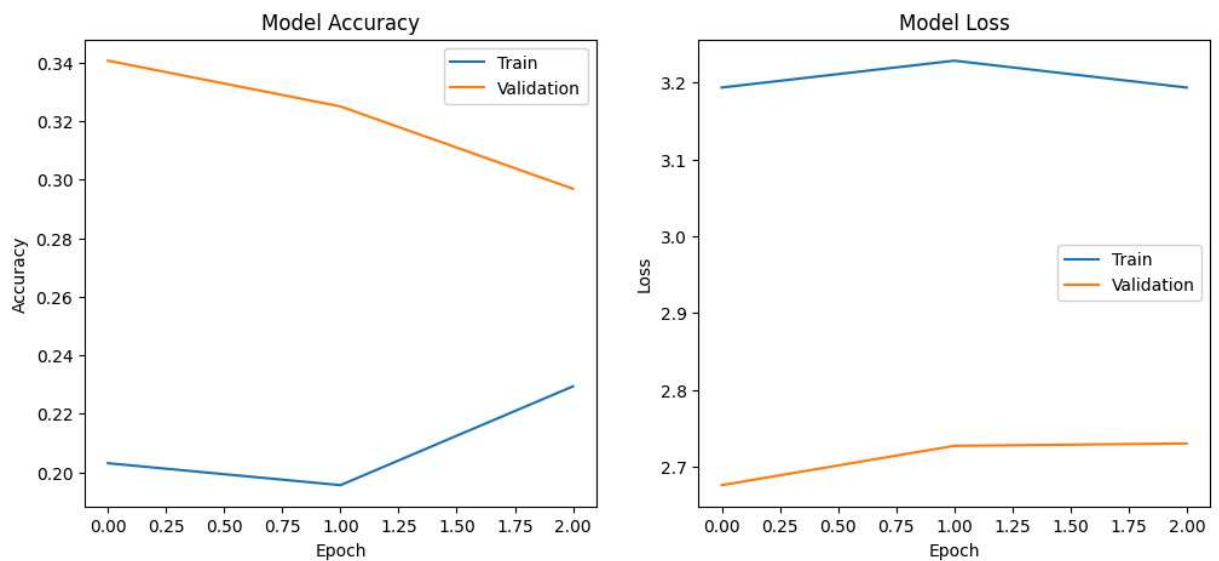
plt.figure(figsize=(12,5))

plt.subplot(1,2,1)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'])

plt.subplot(1,2,2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'])

plt.show()

```



```
In [25]: test_loss, test_accuracy = model.evaluate(test_generator)

print("Test Accuracy:", test_accuracy)
print("Test Loss:", test_loss)
```

16/16 ————— 3s 161ms/step - accuracy: 0.3360 - loss: 2.7090
 Test Accuracy: 0.335999995470047
 Test Loss: 2.709047317504883

```
In [26]: model.save("sports_cnn_model.h5")
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.