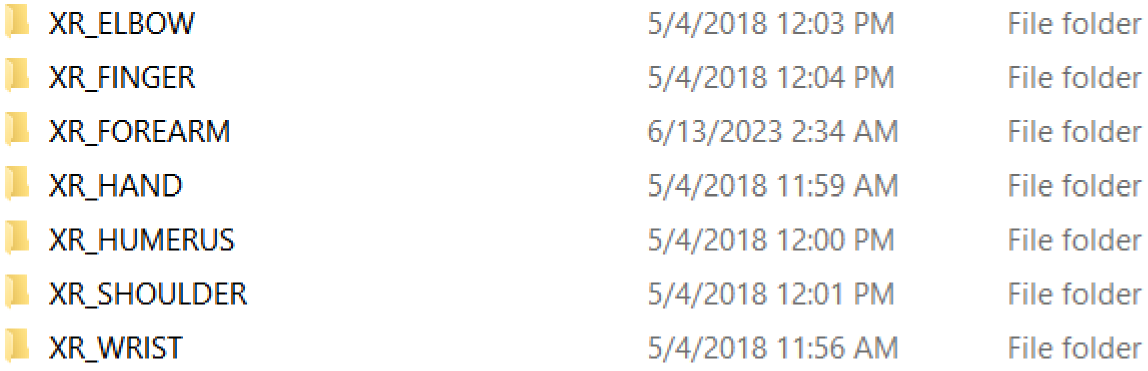
سروش جهانیان

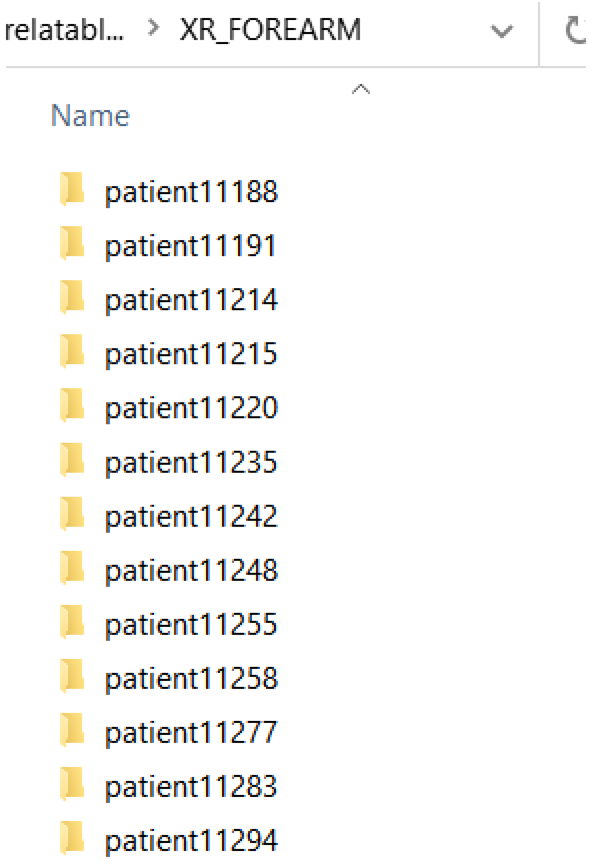
983112034

گزارش پروژه پایانی مبانی بینایی ماشین

در مرحله اول باید دیتاستی که به صورت زیر است



XR_ELBOW	5/4/2018 12:03 PM	File folder
XR_FINGER	5/4/2018 12:04 PM	File folder
XR_FOREARM	6/13/2023 2:34 AM	File folder
XR_HAND	5/4/2018 11:59 AM	File folder
XR_HUMERUS	5/4/2018 12:00 PM	File folder
XR_SHOULDER	5/4/2018 12:01 PM	File folder
XR_WRIST	5/4/2018 11:56 AM	File folder



relatabl... > XR_FOREARM

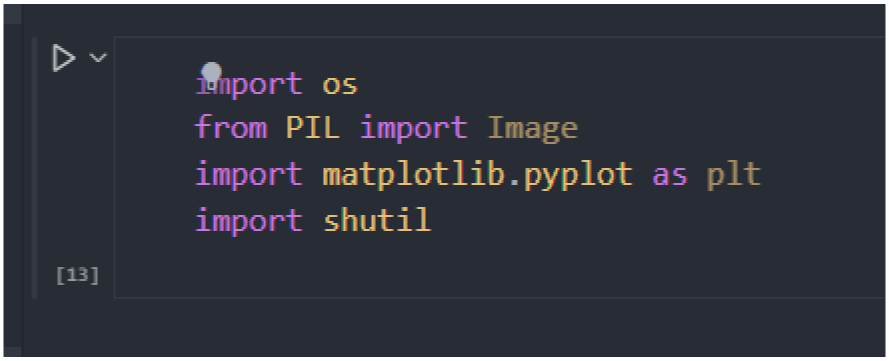
Name

- patient11188
- patient11191
- patient11214
- patient11215
- patient11220
- patient11235
- patient11242
- patient11248
- patient11255
- patient11258
- patient11277
- patient11283
- patient11294

را به شکلی دربیاوریم که کل عکس ها در پوشه test و train قرار بگیرند.

برای همین کدی را می نویسیم که (چون پوشه عکس های با نتیجه مثبت شامل "positive" است) عکس های داخل این پوشه های رو پیدا کند و در قسمت smiling قرار دهد.

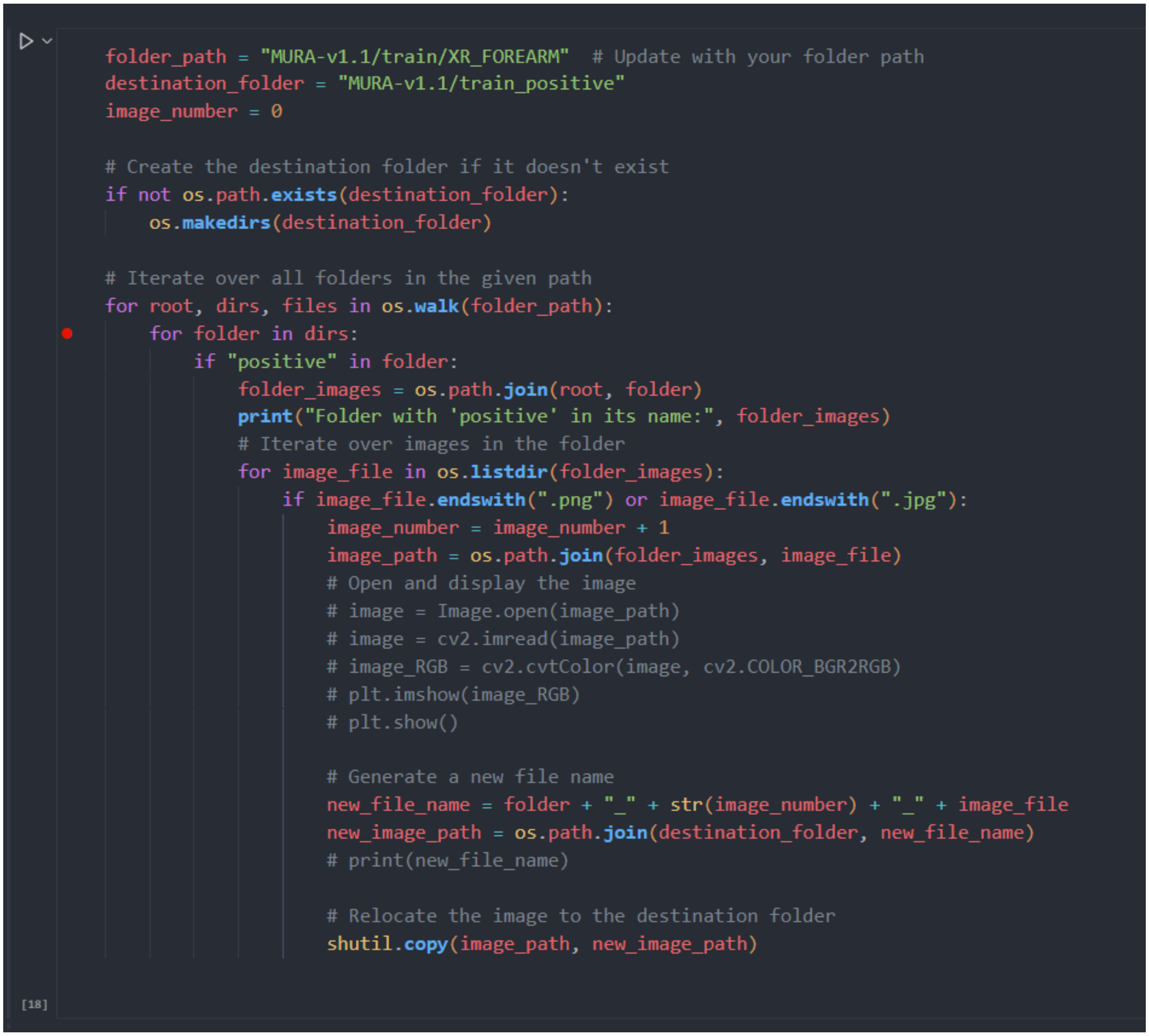
پکیج های استفاده شده برای این کار :



```
import os
from PIL import Image
import matplotlib.pyplot as plt
import shutil
```

[13]

کد :



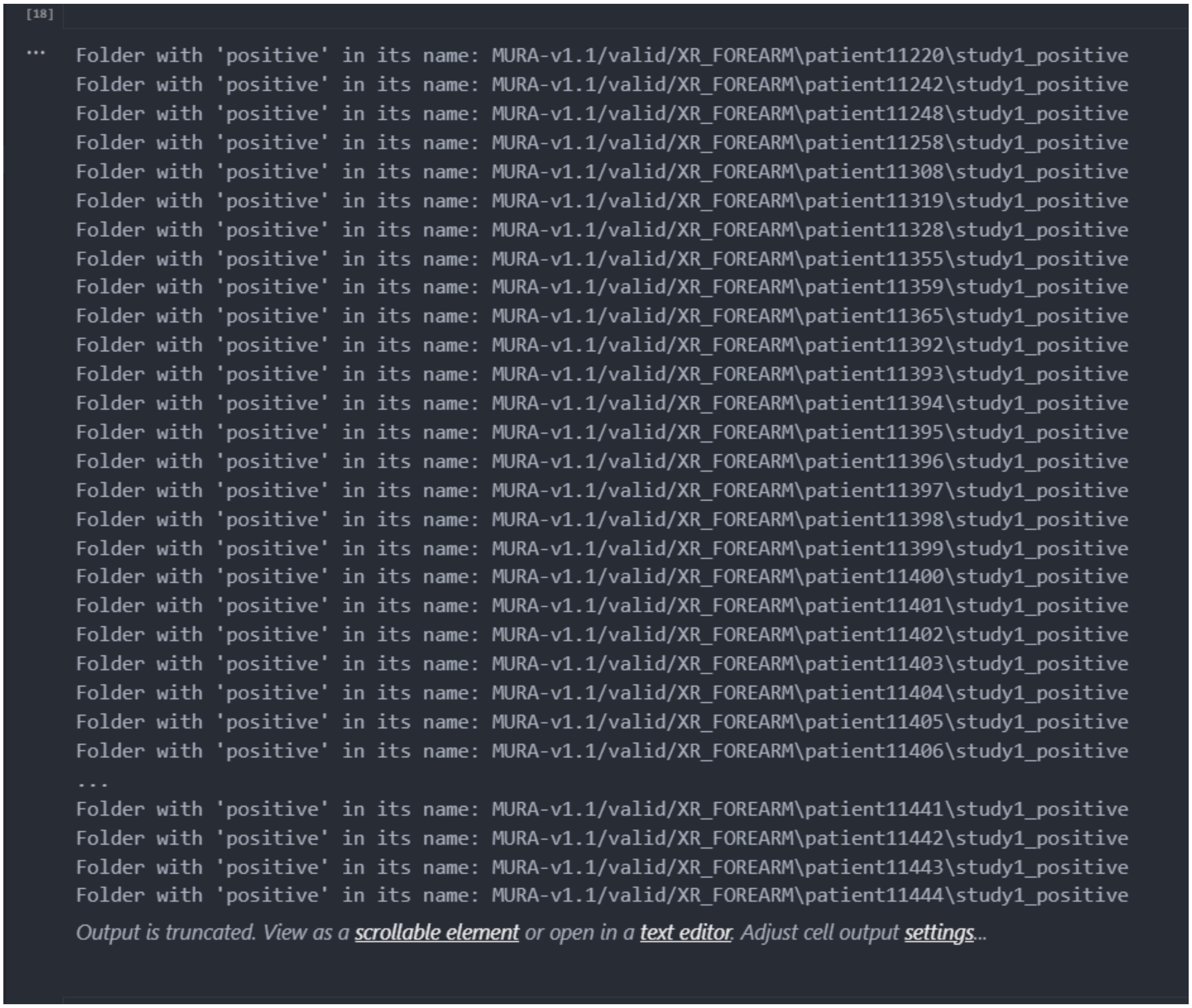
```python
folder_path = "MURA-v1.1/train/XR_FOREARM"  # Update with your folder path
destination_folder = "MURA-v1.1/train_positive"
image_number = 0

# Create the destination folder if it doesn't exist
if not os.path.exists(destination_folder):
    os.makedirs(destination_folder)

# Iterate over all folders in the given path
for root, dirs, files in os.walk(folder_path):
    for folder in dirs:
        if "positive" in folder:
            folder_images = os.path.join(root, folder)
            print("Folder with 'positive' in its name:", folder_images)
            # Iterate over images in the folder
            for image_file in os.listdir(folder_images):
                if image_file.endswith(".png") or image_file.endswith(".jpg"):
                    image_number = image_number + 1
                    image_path = os.path.join(folder_images, image_file)
                    # Open and display the image
                    # image = Image.open(image_path)
                    # image = cv2.imread(image_path)
                    # image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    # plt.imshow(image_RGB)
                    # plt.show()

                    # Generate a new file name
                    new_file_name = folder + "_" + str(image_number) + "_" + image_file
                    new_image_path = os.path.join(destination_folder, new_file_name)
                    # print(new_file_name)

                    # Relocate the image to the destination folder
                    shutil.copy(image_path, new_image_path)
```

برای این کار path پوشه فعلی و پوشه ای که قراره منتقل بشه بهش داده میشود. چک میکنه اگه پوشه وجود نداشت درست میکنه. سپس پوشه های دارای "positive" keyword رو پیدا میکنه و عکساشو پیدا میکنه. سپس اسم عکسا رو عوض میکنه که تو پوشه مقصد overwrite نداشته باشیم.

برای این کار یک counter درست کردم و به اسم عکسا اضافه کردم.

در آخر هم با دستور shutil عکس ها را کپی می کنیم.

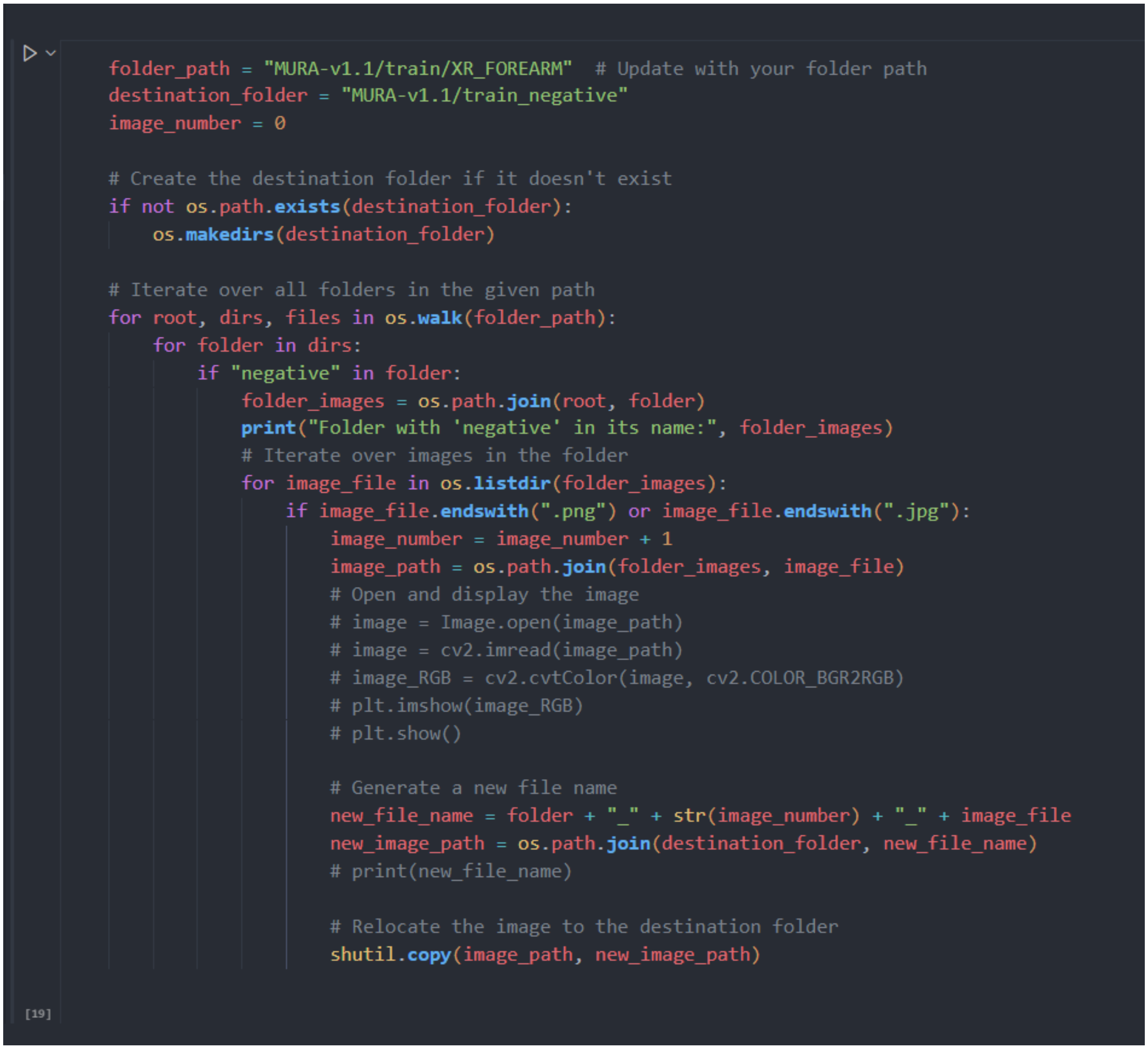
نتیجه :



```
[18]
...  Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11220\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11242\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11248\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11258\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11308\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11319\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11328\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11355\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11359\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11365\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11392\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11393\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11394\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11395\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11396\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11397\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11398\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11399\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11400\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11401\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11402\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11403\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11404\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11405\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11406\study1_positive
     ...
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11441\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11442\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11443\study1_positive
     Folder with 'positive' in its name: MURA-v1.1/valid/XR_FOREARM\patient11444\study1_positive
     Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

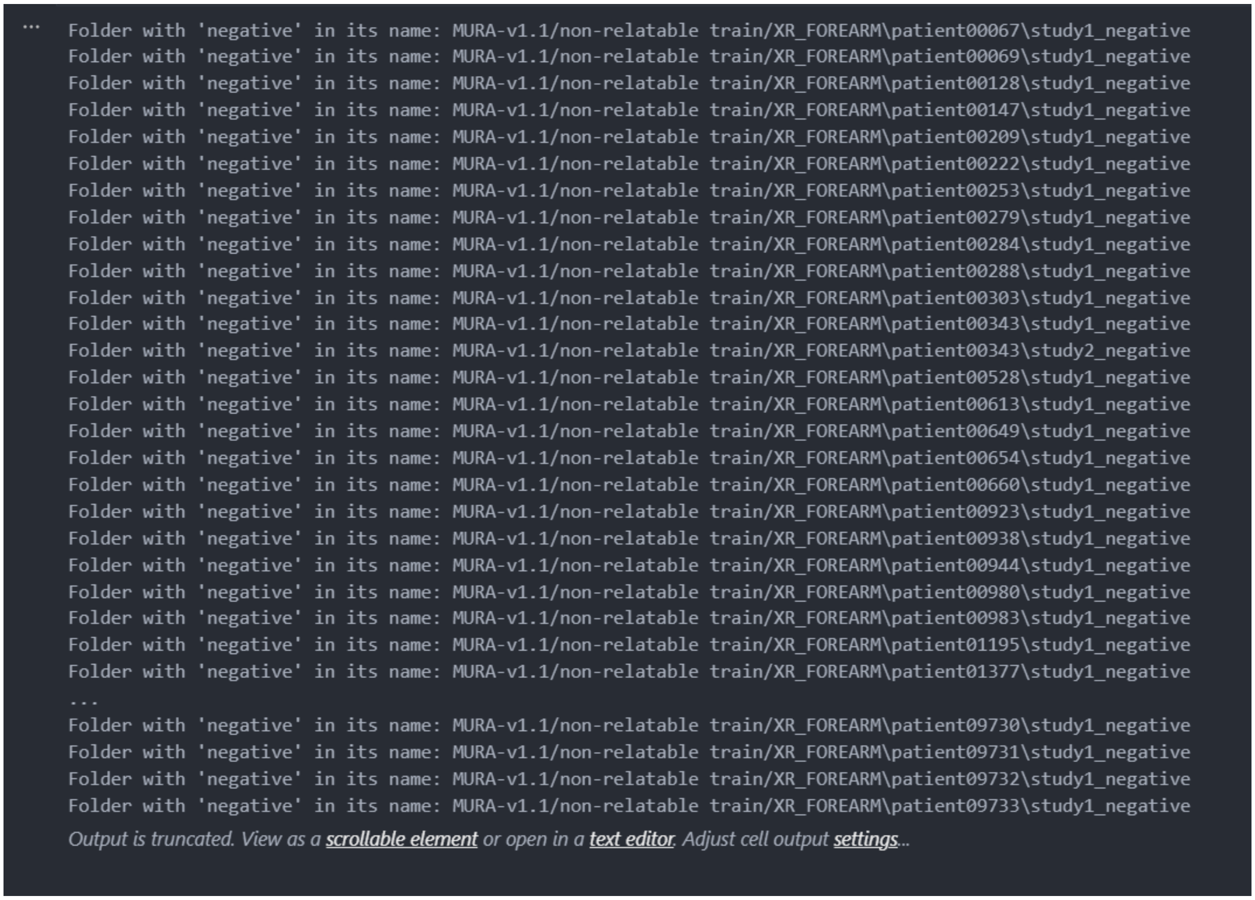
در مرحله بعد همین کار را برای عکس های با keyword "negative" انجام میدهیم. اما آنها را در پوشه non-smiling قرار می دهیم.

کد :



```python
folder_path = "MURA-v1.1/train/XR_FOREARM"  # Update with your folder path
destination_folder = "MURA-v1.1/train_negative"
image_number = 0

# Create the destination folder if it doesn't exist
if not os.path.exists(destination_folder):
    os.makedirs(destination_folder)

# Iterate over all folders in the given path
for root, dirs, files in os.walk(folder_path):
    for folder in dirs:
        if "negative" in folder:
            folder_images = os.path.join(root, folder)
            print("Folder with 'negative' in its name:", folder_images)
            # Iterate over images in the folder
            for image_file in os.listdir(folder_images):
                if image_file.endswith(".png") or image_file.endswith(".jpg"):
                    image_number = image_number + 1
                    image_path = os.path.join(folder_images, image_file)
                    # Open and display the image
                    # image = Image.open(image_path)
                    # image = cv2.imread(image_path)
                    # image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    # plt.imshow(image_RGB)
                    # plt.show()

                    # Generate a new file name
                    new_file_name = folder + "_" + str(image_number) + "_" + image_file
                    new_image_path = os.path.join(destination_folder, new_file_name)
                    # print(new_file_name)

                    # Relocate the image to the destination folder
                    shutil.copy(image_path, new_image_path)
```

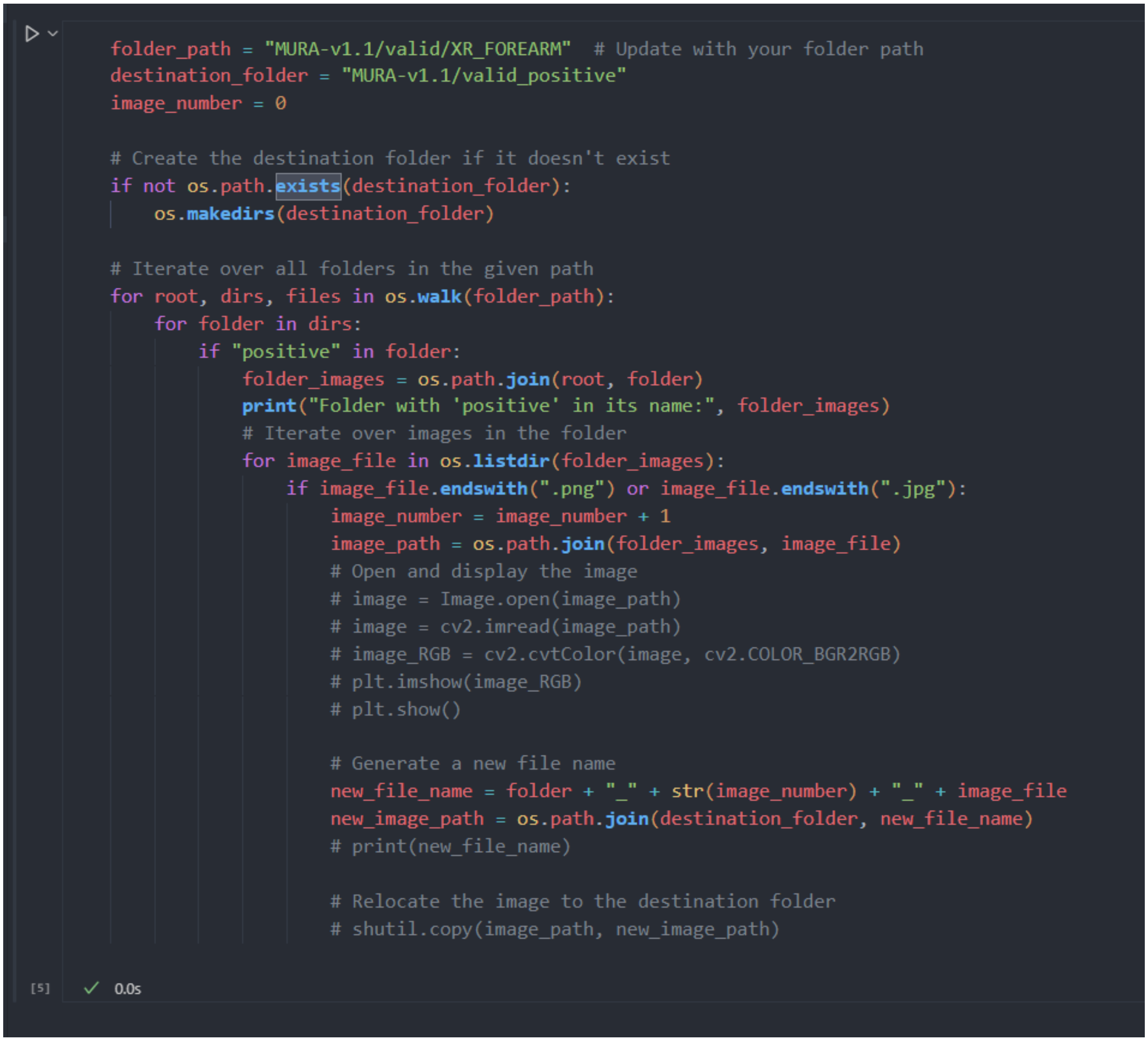
نتیجه :



```
...  Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00067\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00069\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00128\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00147\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00209\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00222\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00253\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00279\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00284\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00288\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00303\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00343\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00343\study2_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00528\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00613\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00649\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00654\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00660\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00923\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00938\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00944\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00980\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient00983\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient01195\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient01377\study1_negative
     ...
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient09730\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient09731\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient09732\study1_negative
     Folder with 'negative' in its name: MURA-v1.1/non-relatable train/XR_FOREARM\patient09733\study1_negative
     Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

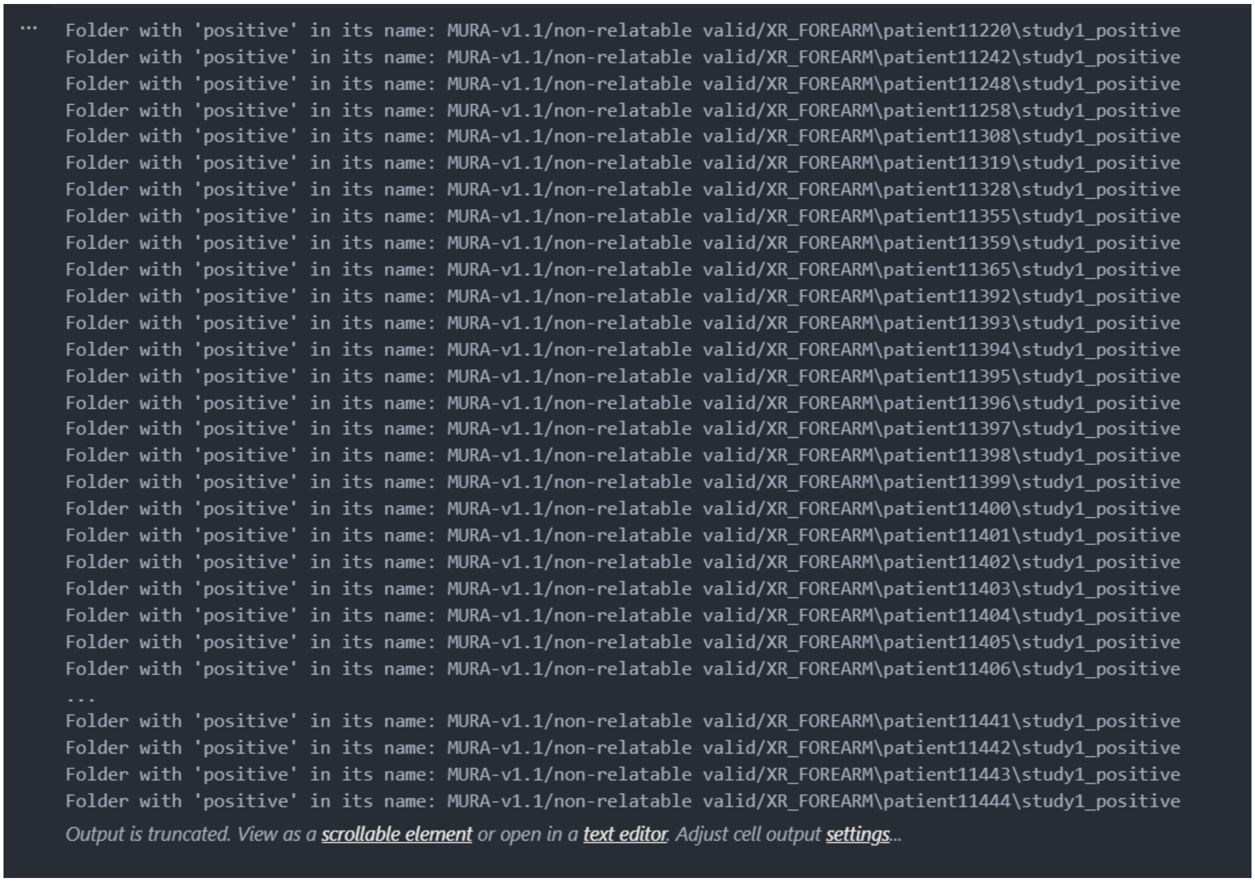
چون دیتاست خودش پوشه های test و valid رو جدا کرده بود منم به ترتیب پوشه بندی دست نزدم و همین کاری که برای train انجام دادیم را برای valid انجام داده و آنها را در پوشه valid در زیرپوشه های smiling و non-smiling قرار می دهیم.

کد :



```python
folder_path = "MURA-v1.1/valid/XR_FOREARM"  # Update with your folder path
destination_folder = "MURA-v1.1/valid_positive"
image_number = 0

# Create the destination folder if it doesn't exist
if not os.path.exists(destination_folder):
    os.makedirs(destination_folder)

# Iterate over all folders in the given path
for root, dirs, files in os.walk(folder_path):
    for folder in dirs:
        if "positive" in folder:
            folder_images = os.path.join(root, folder)
            print("Folder with 'positive' in its name:", folder_images)
            # Iterate over images in the folder
            for image_file in os.listdir(folder_images):
                if image_file.endswith(".png") or image_file.endswith(".jpg"):
                    image_number = image_number + 1
                    image_path = os.path.join(folder_images, image_file)
                    # Open and display the image
                    # image = Image.open(image_path)
                    # image = cv2.imread(image_path)
                    # image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    # plt.imshow(image_RGB)
                    # plt.show()

                    # Generate a new file name
                    new_file_name = folder + "_" + str(image_number) + "_" + image_file
                    new_image_path = os.path.join(destination_folder, new_file_name)
                    # print(new_file_name)

                    # Relocate the image to the destination folder
                    # shutil.copy(image_path, new_image_path)
```

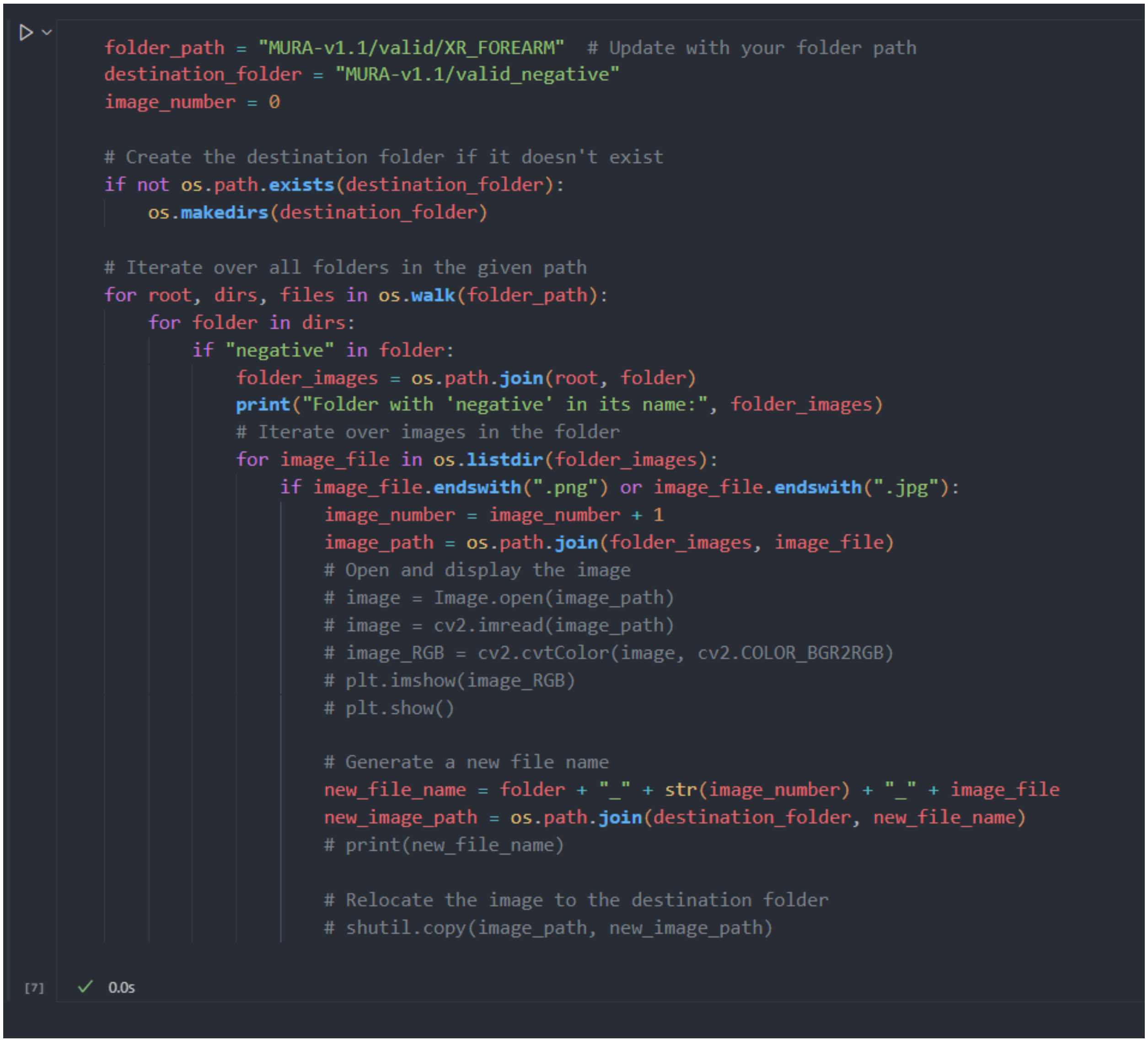
نتیجه :



```
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11220\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11242\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11248\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11258\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11308\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11319\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11328\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11355\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11359\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11365\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11392\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11393\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11394\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11395\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11396\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11397\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11398\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11399\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11400\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11401\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11402\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11403\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11404\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11405\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11406\study1_positive
...
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11441\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11442\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11443\study1_positive
Folder with 'positive' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11444\study1_positive
```

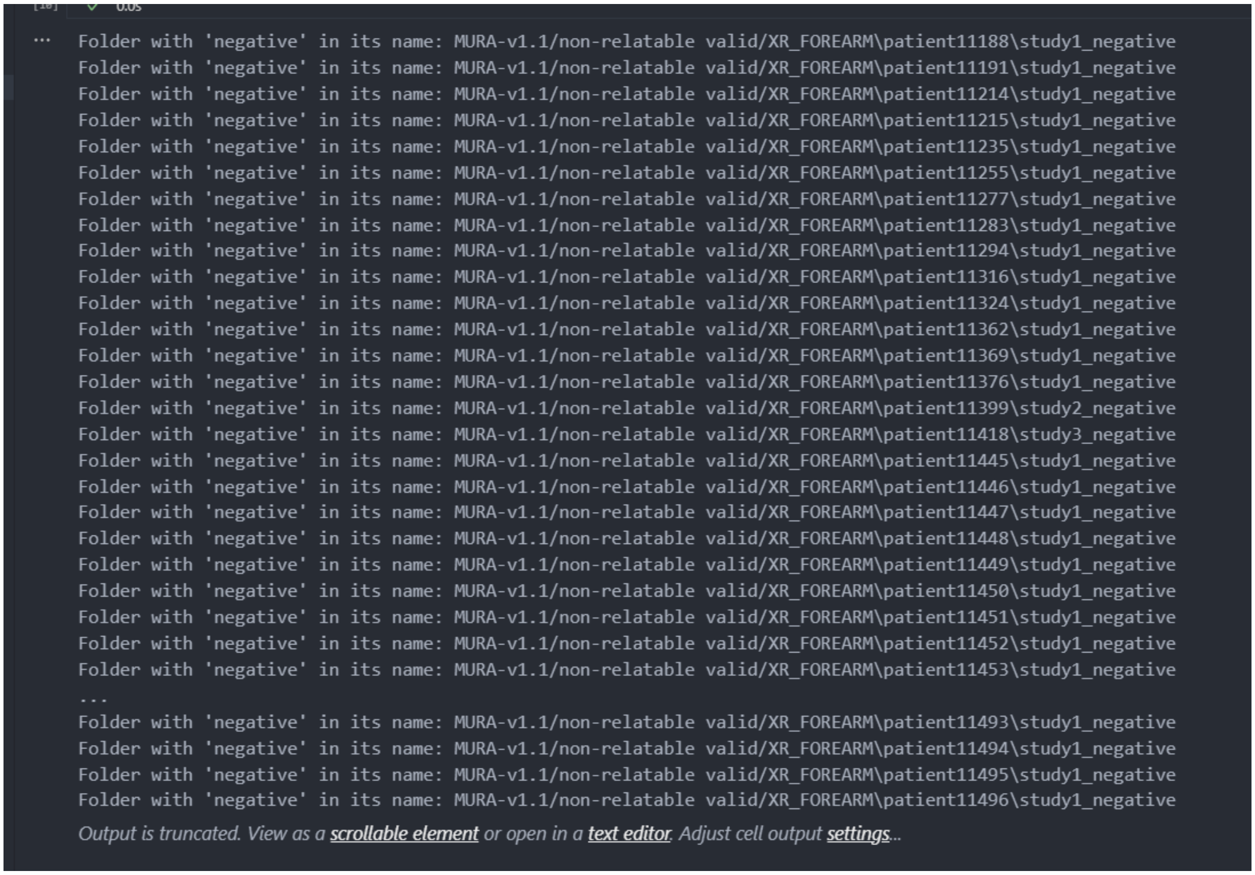
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

کد :



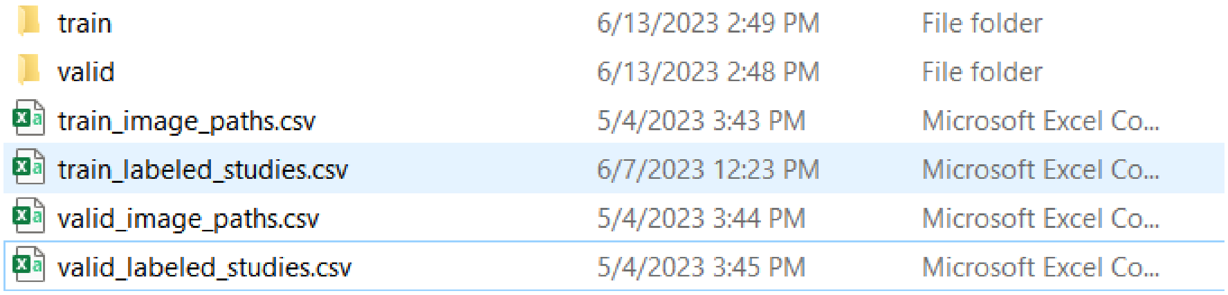
```python
folder_path = "MURA-v1.1/valid/XR_FOREARM"  # Update with your folder path
destination_folder = "MURA-v1.1/valid_negative"
image_number = 0

# Create the destination folder if it doesn't exist
if not os.path.exists(destination_folder):
    os.makedirs(destination_folder)

# Iterate over all folders in the given path
for root, dirs, files in os.walk(folder_path):
    for folder in dirs:
        if "negative" in folder:
            folder_images = os.path.join(root, folder)
            print("Folder with 'negative' in its name:", folder_images)
            # Iterate over images in the folder
            for image_file in os.listdir(folder_images):
                if image_file.endswith(".png") or image_file.endswith(".jpg"):
                    image_number = image_number + 1
                    image_path = os.path.join(folder_images, image_file)
                    # Open and display the image
                    # image = Image.open(image_path)
                    # image = cv2.imread(image_path)
                    # image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    # plt.imshow(image_RGB)
                    # plt.show()

                    # Generate a new file name
                    new_file_name = folder + "_" + str(image_number) + "_" + image_file
                    new_image_path = os.path.join(destination_folder, new_file_name)
                    # print(new_file_name)

                    # Relocate the image to the destination folder
                    # shutil.copy(image_path, new_image_path)
```

نتیجه :

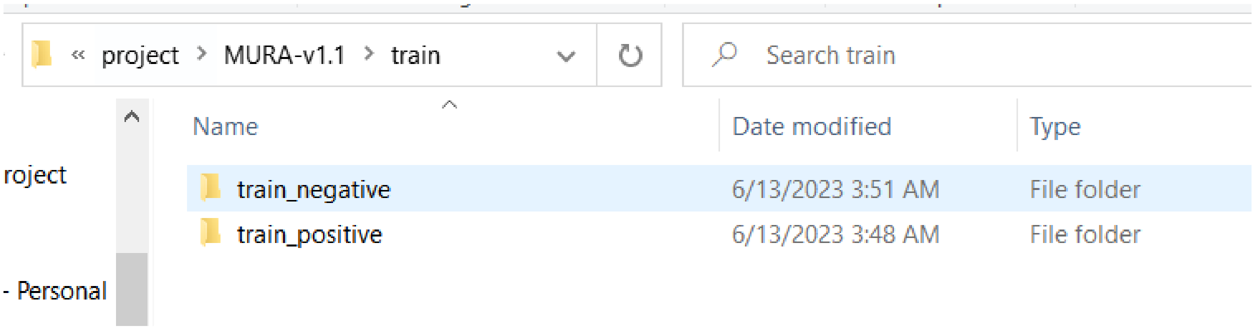


```
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11188\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11191\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11214\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11215\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11235\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11255\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11277\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11283\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11294\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11316\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11324\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11362\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11369\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11376\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11399\study2_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11418\study3_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11445\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11446\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11447\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11448\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11449\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11450\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11451\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11452\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11453\study1_negative
...
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11493\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11494\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11495\study1_negative
Folder with 'negative' in its name: MURA-v1.1/non-relatable valid/XR_FOREARM\patient11496\study1_negative
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

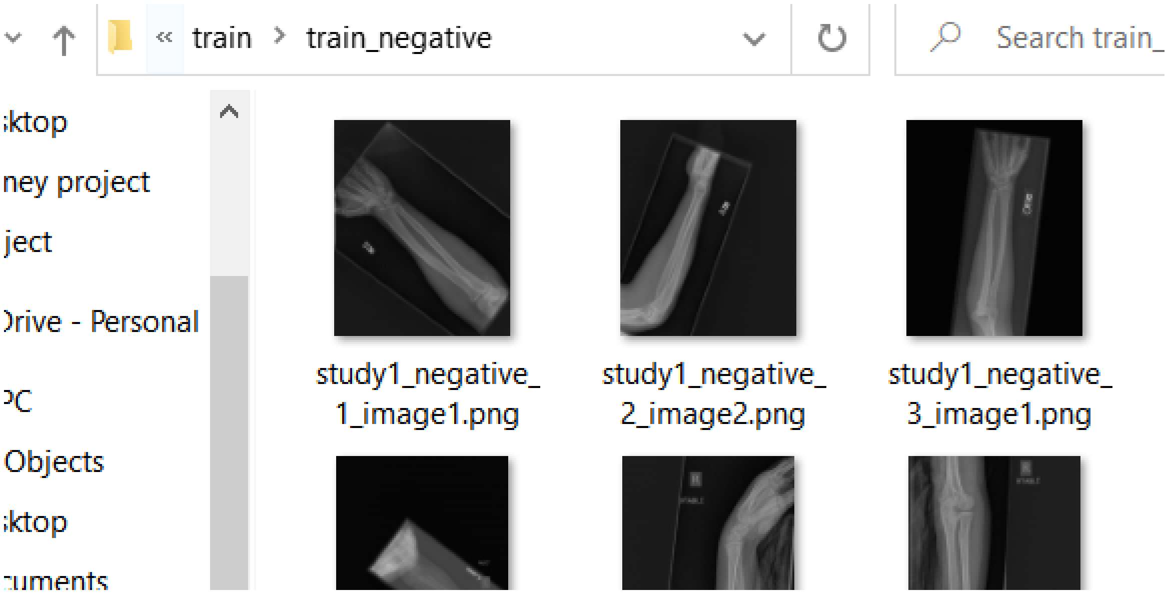
نتیجه نهایی:



train	6/13/2023 2:49 PM	File folder
valid	6/13/2023 2:48 PM	File folder
train_image_paths.csv	5/4/2023 3:43 PM	Microsoft Excel Co...
train_labeled_studies.csv	6/7/2023 12:23 PM	Microsoft Excel Co...
valid_image_paths.csv	5/4/2023 3:44 PM	Microsoft Excel Co...
valid_labeled_studies.csv	5/4/2023 3:45 PM	Microsoft Excel Co...



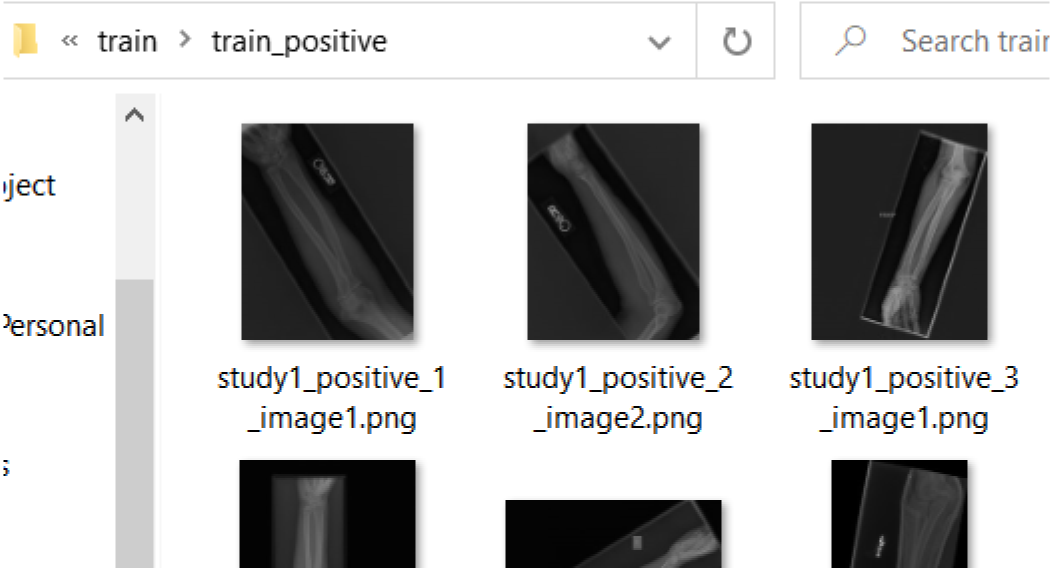
Name	Date modified	Type
train_negative	6/13/2023 3:51 AM	File folder
train_positive	6/13/2023 3:48 AM	File folder



study1_negative_1_image1.png

study1_negative_2_image2.png

study1_negative_3_image1.png



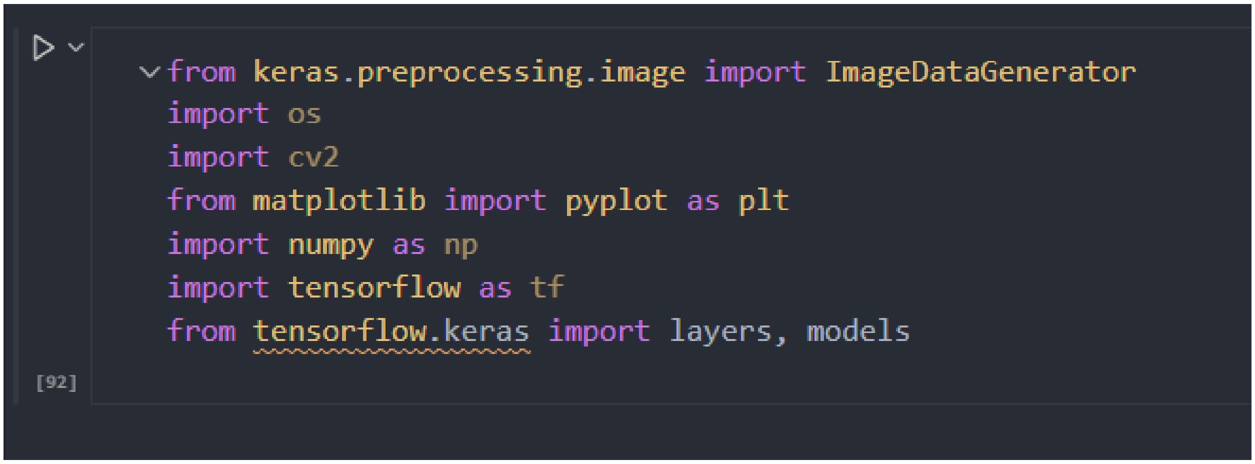
study1_positive_1_image1.png

study1_positive_2_image2.png

study1_positive_3_image1.png

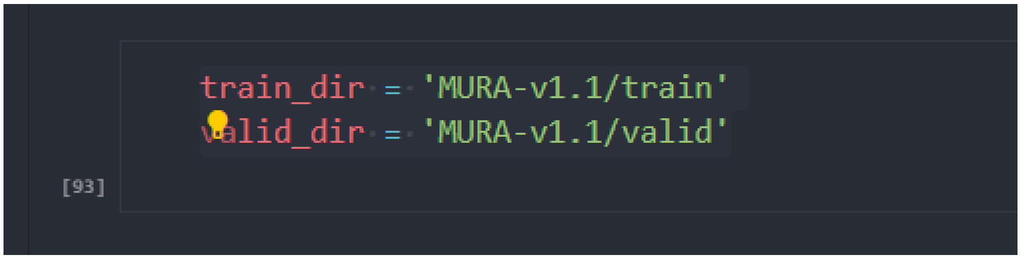
در مرحله بعد در فایل forearm_abnormality_detection :

پکیج های استفاده شده در این فایل :



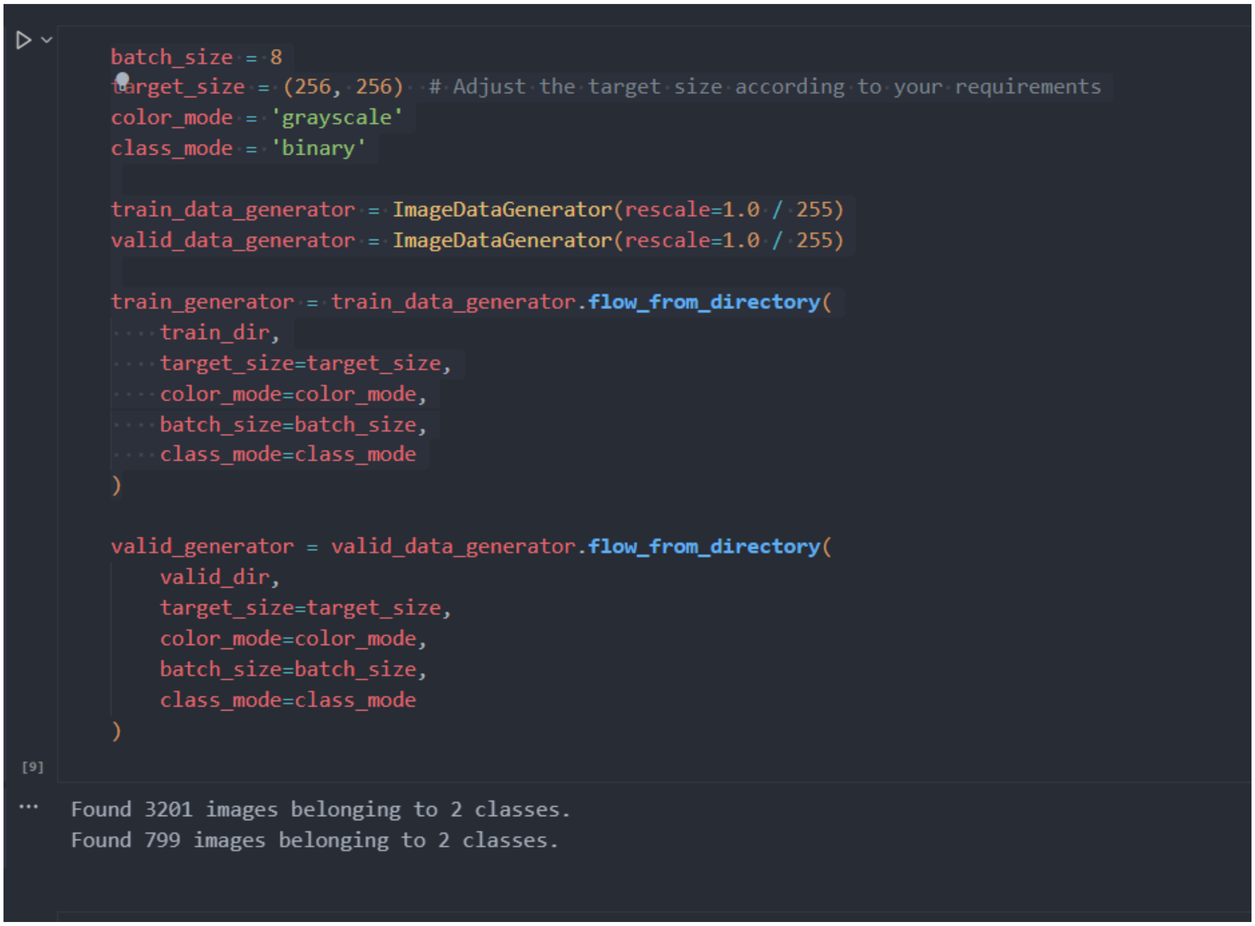
```
from keras.preprocessing.image import ImageDataGenerator
import os
import cv2
from matplotlib import pyplot as plt
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
```

سپس مسیر های پوشه ها را تعیین می کنیم.



```
train_dir = 'MURA-v1.1/train'
valid_dir = 'MURA-v1.1/valid'
```

سپس توسط image generator تمام عکس ها رو خونده. عکس های train در train generator و عکس های test در test generator ذخیره می شوند



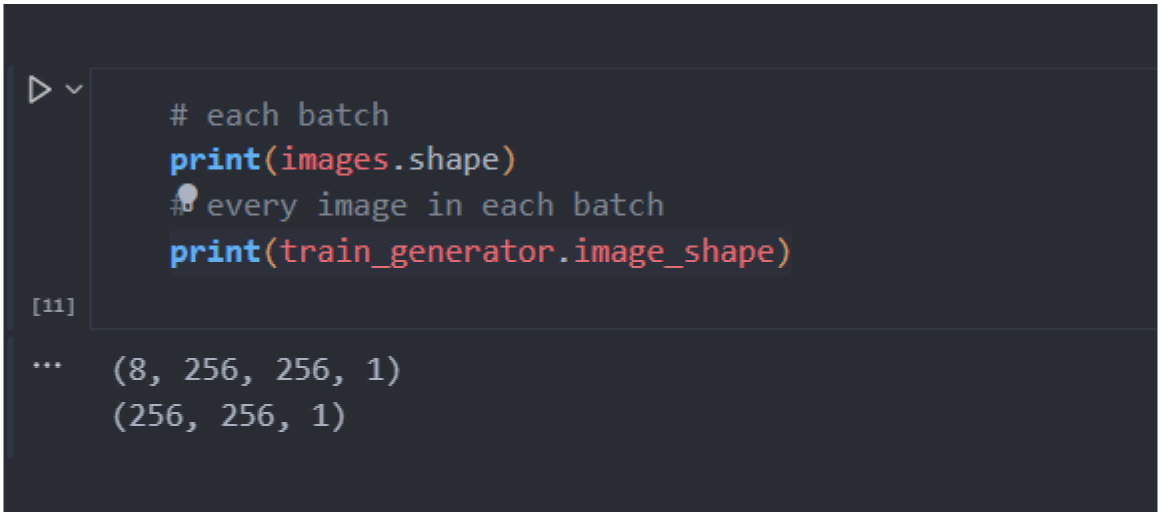
```
batch_size = 8
target_size = (256, 256)  # Adjust the target size according to your requirements
color_mode = 'grayscale'
class_mode = 'binary'

train_data_generator = ImageDataGenerator(rescale=1.0 / 255)
valid_data_generator = ImageDataGenerator(rescale=1.0 / 255)

train_generator = train_data_generator.flow_from_directory(
    train_dir,
    target_size=target_size,
    color_mode=color_mode,
    batch_size=batch_size,
    class_mode=class_mode
)

valid_generator = valid_data_generator.flow_from_directory(
    valid_dir,
    target_size=target_size,
    color_mode=color_mode,
    batch_size=batch_size,
    class_mode=class_mode
)
```

[9]

```
Found 3201 images belonging to 2 classes.
Found 799 images belonging to 2 classes.
```

مزیت استفاده از image generator این است که همه عکسای خوانده شده را به فرمت (256, 256) در میاورد و grayscale می کند بنابراین فرمت هر عکس به صورت (256, 256, 1) می شود.

و در batch های 8تایی به مدل داده می شود.

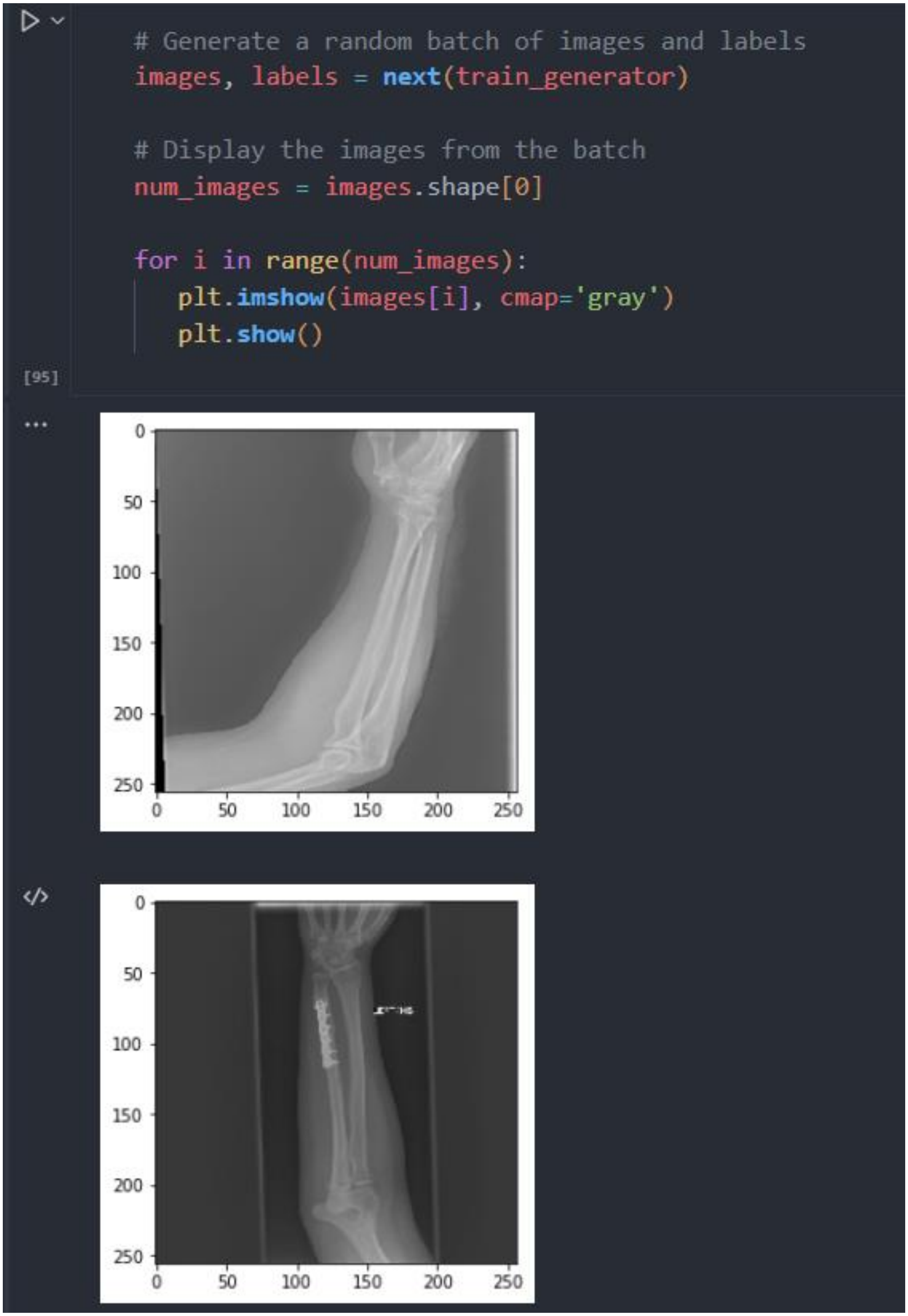


```
# each batch
print(images.shape)
# every image in each batch
print(train_generator.image_shape)
```

[11]

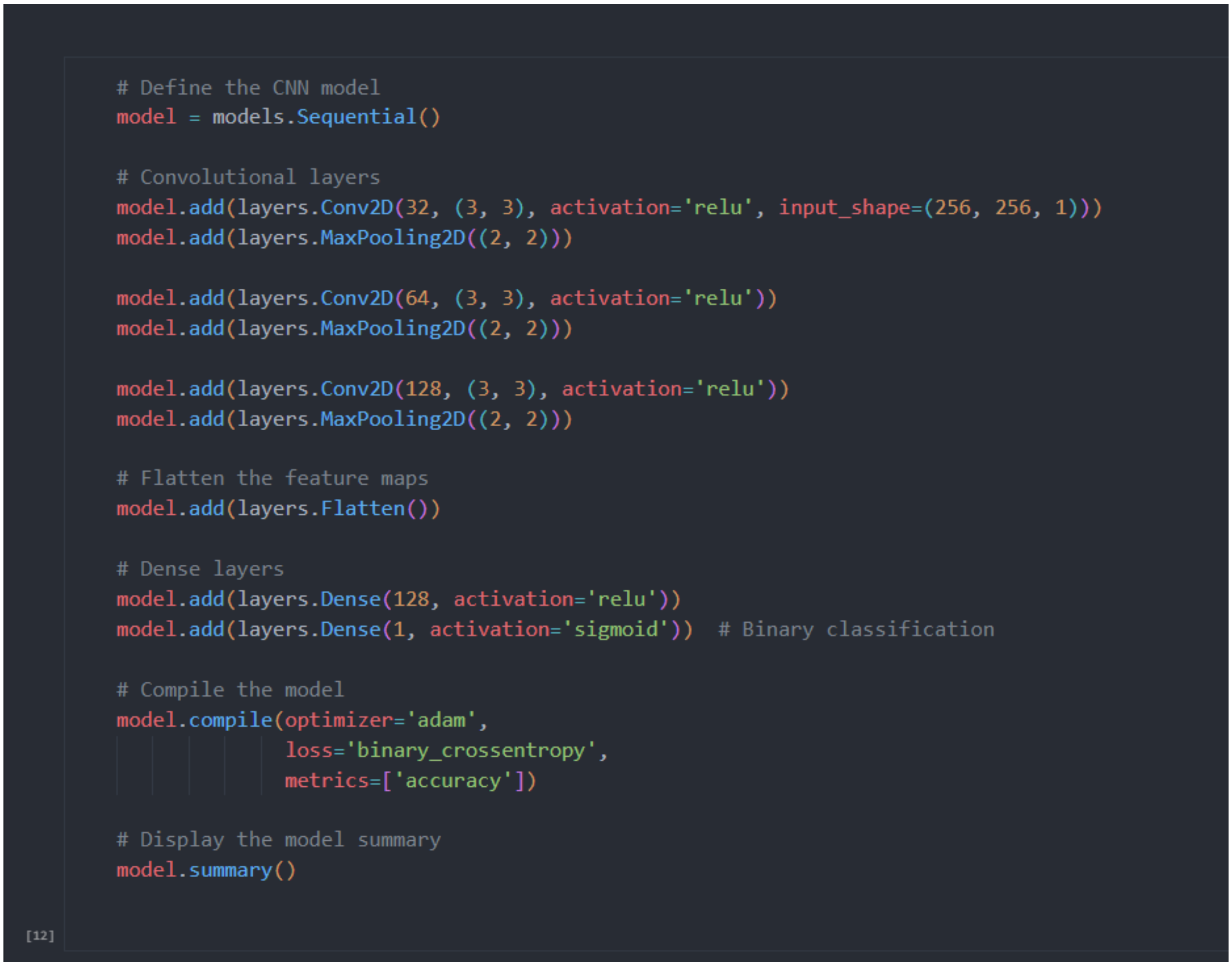
```
(8, 256, 256, 1)
(256, 256, 1)
```

نمایش یک batch رندوم 8تایی:



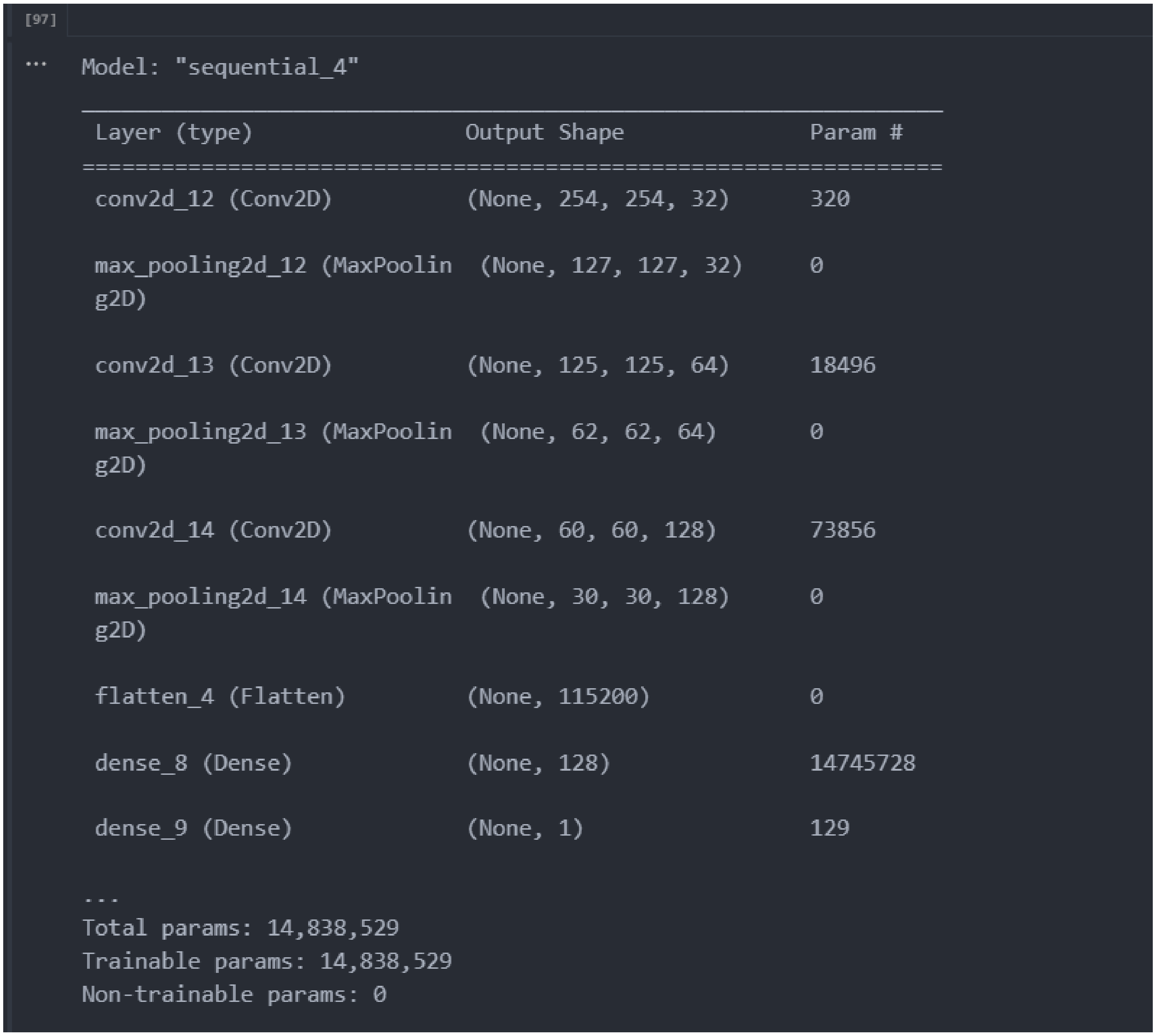
```
# Generate a random batch of images and labels
images, labels = next(train_generator)

# Display the images from the batch
num_images = images.shape[0]

for i in range(num_images):
    plt.imshow(images[i], cmap='gray')
    plt.show()
```

در مرحله بعد مدل CNN رو میسازیم:



```
# Define the CNN model
model = models.Sequential()

# Convolutional layers
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 1)))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

# Flatten the feature maps
model.add(layers.Flatten())

# Dense layers
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))  # Binary classification

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Display the model summary
model.summary()
```

مدل ما دارای 9 لایه است که به ترتیب convolution, max pooling, convolution, max pooling, convolution, max pooling, flatten, dense, dense است

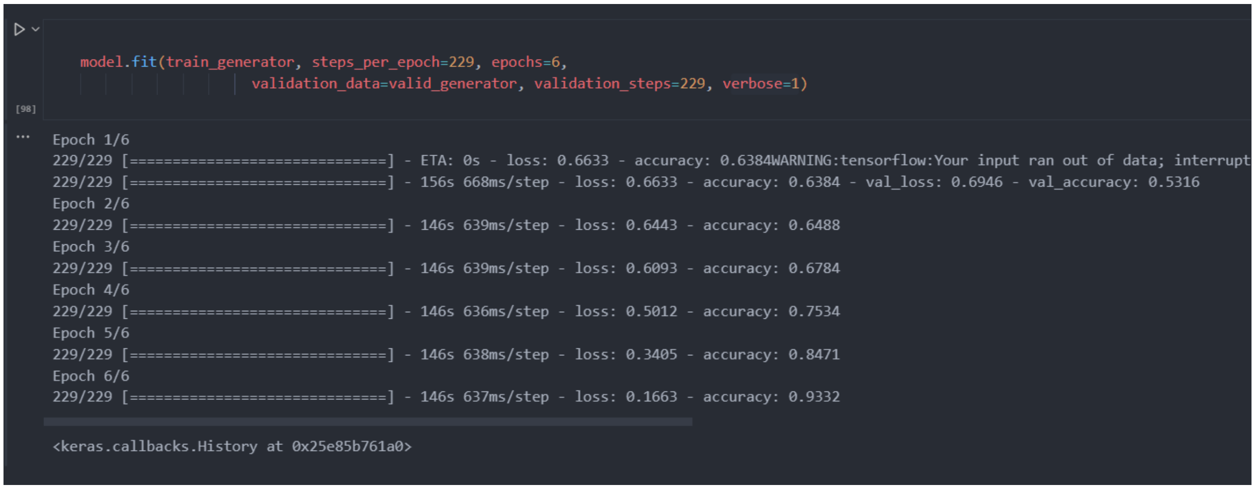


```
[97]

...  Model: "sequential_4"
     _________________________________________________________________
      Layer (type)                Output Shape              Param #
     =================================================================
      conv2d_12 (Conv2D)          (None, 254, 254, 32)      320

      max_pooling2d_12 (MaxPoolin  (None, 127, 127, 32)     0
      g2D)

      conv2d_13 (Conv2D)          (None, 125, 125, 64)      18496

      max_pooling2d_13 (MaxPoolin  (None, 62, 62, 64)       0
      g2D)

      conv2d_14 (Conv2D)          (None, 60, 60, 128)       73856

      max_pooling2d_14 (MaxPoolin  (None, 30, 30, 128)      0
      g2D)

      flatten_4 (Flatten)         (None, 115200)            0

      dense_8 (Dense)             (None, 128)               14745728

      dense_9 (Dense)             (None, 1)                 129

     ...
     Total params: 14,838,529
     Trainable params: 14,838,529
     Non-trainable params: 0
```

سپس باید مدل را fit کنیم.

برای اینکه overfit نداشته باشیم ابتدا 6 epoch می زنیم و دقت را اندازه گیری می کنیم.

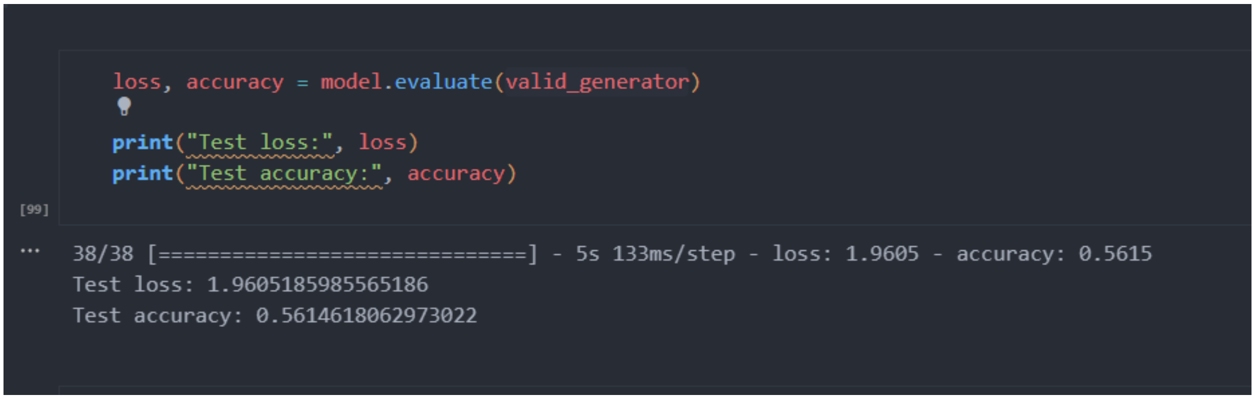
با قرار دادن verbose دقت در epoch را مشاهده می کنیم.



```
model.fit(train_generator, steps_per_epoch=229, epochs=6,
                           validation_data=valid_generator, validation_steps=229, verbose=1)
```

```
Epoch 1/6
229/229 [==============================] - ETA: 0s - loss: 0.6633 - accuracy: 0.6384WARNING:tensorflow:Your input ran out of data; interrupt
229/229 [==============================] - 156s 668ms/step - loss: 0.6633 - accuracy: 0.6384 - val_loss: 0.6946 - val_accuracy: 0.5316
Epoch 2/6
229/229 [==============================] - 146s 639ms/step - loss: 0.6443 - accuracy: 0.6488
Epoch 3/6
229/229 [==============================] - 146s 639ms/step - loss: 0.6093 - accuracy: 0.6784
Epoch 4/6
229/229 [==============================] - 146s 636ms/step - loss: 0.5012 - accuracy: 0.7534
Epoch 5/6
229/229 [==============================] - 146s 638ms/step - loss: 0.3405 - accuracy: 0.8471
Epoch 6/6
229/229 [==============================] - 146s 637ms/step - loss: 0.1663 - accuracy: 0.9332

<keras.callbacks.History at 0x25e85b761a0>
```

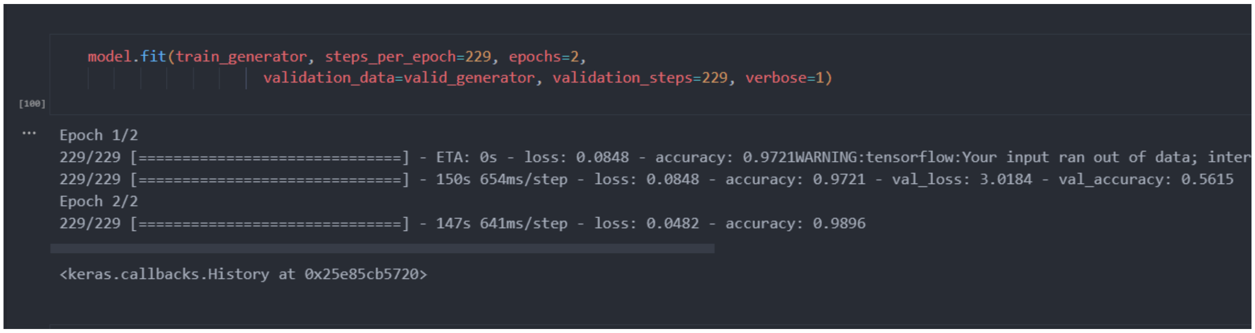
سپس دیتای test را به مدل می دهیم و loss و accuracy را اندازه گیری می کنیم.



```
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
```

```
38/38 [==============================] - 5s 133ms/step - loss: 1.9605 - accuracy: 0.5615
Test loss: 1.9605185985565186
Test accuracy: 0.5614618062973022
```

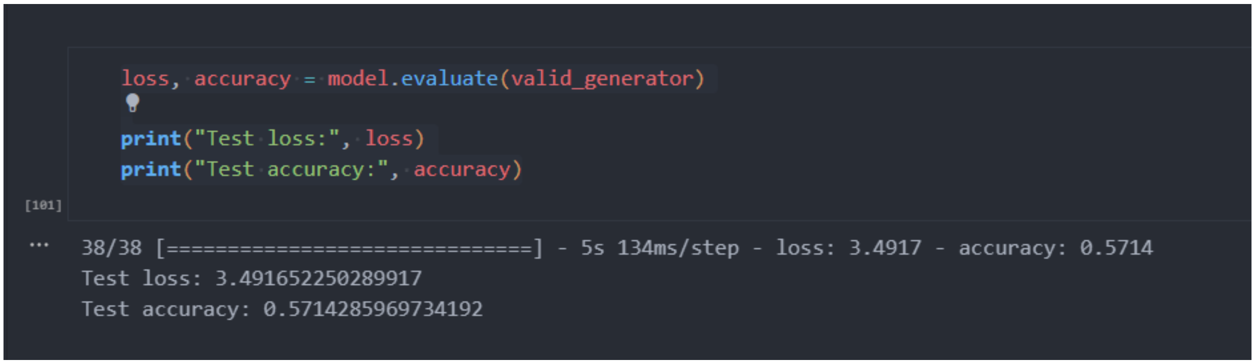
سپس دو epoch دیگر می زنیم.



```
model.fit(train_generator, steps_per_epoch=229, epochs=2,
                           validation_data=valid_generator, validation_steps=229, verbose=1)
```

```
Epoch 1/2
229/229 [==============================] - ETA: 0s - loss: 0.0848 - accuracy: 0.9721WARNING:tensorflow:Your input ran out of data; inter
229/229 [==============================] - 150s 654ms/step - loss: 0.0848 - accuracy: 0.9721 - val_loss: 3.0184 - val_accuracy: 0.5615
Epoch 2/2
229/229 [==============================] - 147s 641ms/step - loss: 0.0482 - accuracy: 0.9896

<keras.callbacks.History at 0x25e85cb5720>
```

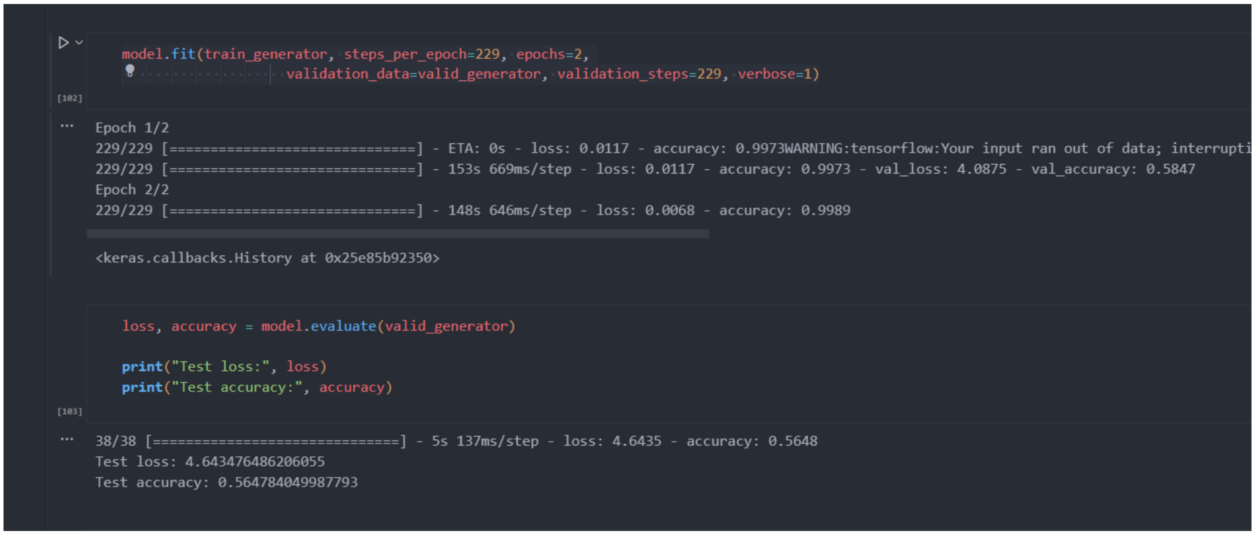
دوباره دیتای test را به مدل می دهیم و loss و accuracy را اندازه گیری می کنیم.



```
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
```

```
38/38 [==============================] - 5s 134ms/step - loss: 3.4917 - accuracy: 0.5714
Test loss: 3.491652250289917
Test accuracy: 0.5714285969734192
```

هم دقت test و هم دقت train افزایش پیدا کرده. بنابراین دو epoch دیگه هم می زنیم.



```
model.fit(train_generator, steps_per_epoch=229, epochs=2,
                    validation_data=valid_generator, validation_steps=229, verbose=1)
```

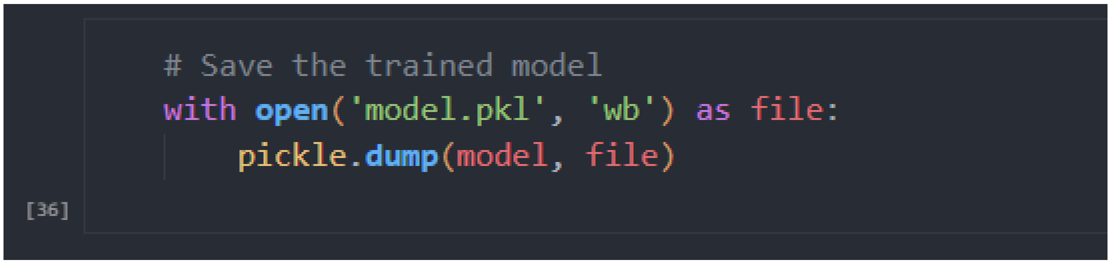
```
Epoch 1/2
229/229 [==============================] - ETA: 0s - loss: 0.0117 - accuracy: 0.9973WARNING:tensorflow:Your input ran out of data; interrupti
229/229 [==============================] - 153s 669ms/step - loss: 0.0117 - accuracy: 0.9973 - val_loss: 4.0875 - val_accuracy: 0.5847
Epoch 2/2
229/229 [==============================] - 148s 646ms/step - loss: 0.0068 - accuracy: 0.9989

<keras.callbacks.History at 0x25e85b92350>
```

```
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
```

```
38/38 [==============================] - 5s 137ms/step - loss: 4.6435 - accuracy: 0.5648
Test loss: 4.643476486206055
Test accuracy: 0.564784049987793
```

مشاهده می کنیم که بعد از epoch اول دقت روی دیتا تست (val_accuracy) برابر با 0.5847 شده ولی در epoch بعدی 0.5647 شده است.

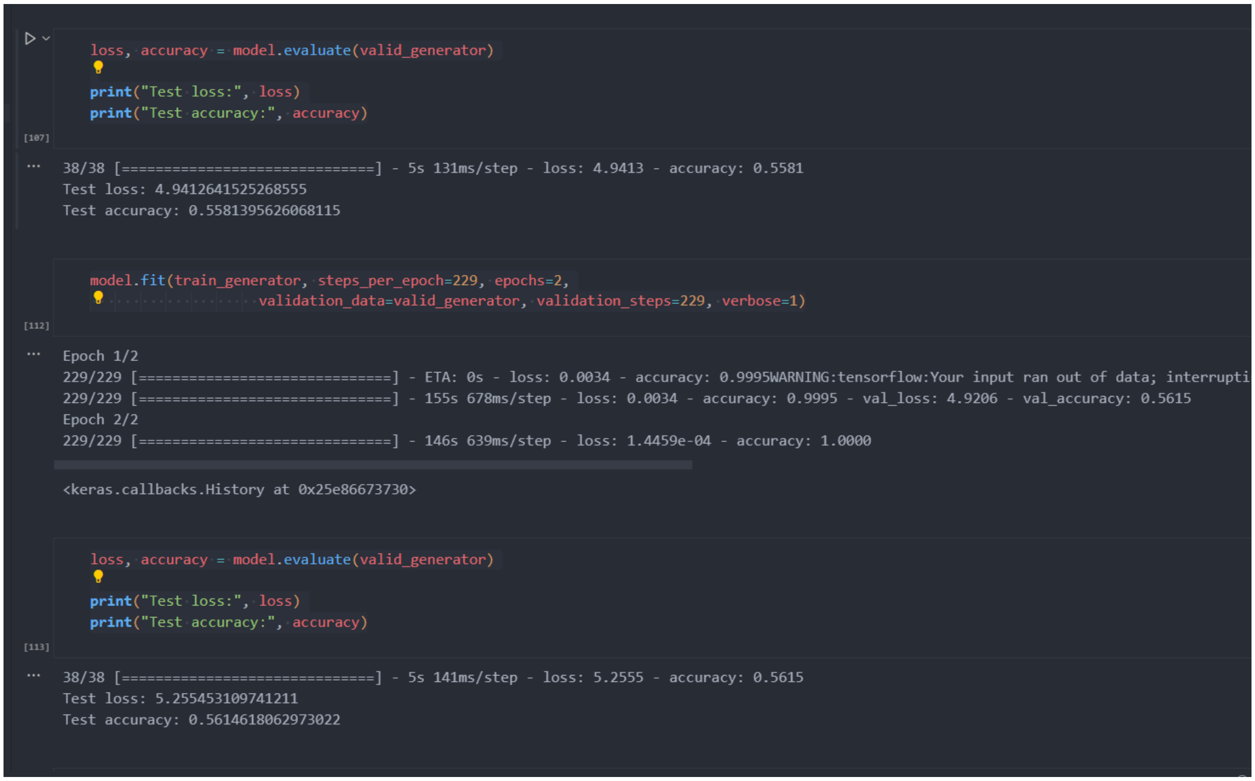
احتمالا اگر ترین کردن مدل را ادامه دهیم دچار overfit خواهیم شد بنابراین دیگر ادامه نمی دهیم و مدل به دست آمده را توسط کتابخونه pickle سیو می کنیم.



```
# Save the trained model
with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)
[36]
```

اما در کد همین کار (زدن دو epoch و بررسی نتیجه را انجام می دهیم) را انجام می دهیم تا اگر نتیجه بهتری حاصل شد مدل را overwrite کنیم.

ولی نتیجه بهتری روی test حاصل نمی شود.



```
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
[107]
```

```
38/38 [==============================] - 5s 131ms/step - loss: 4.9413 - accuracy: 0.5581
Test loss: 4.9412641525268555
Test accuracy: 0.5581395626068115
```

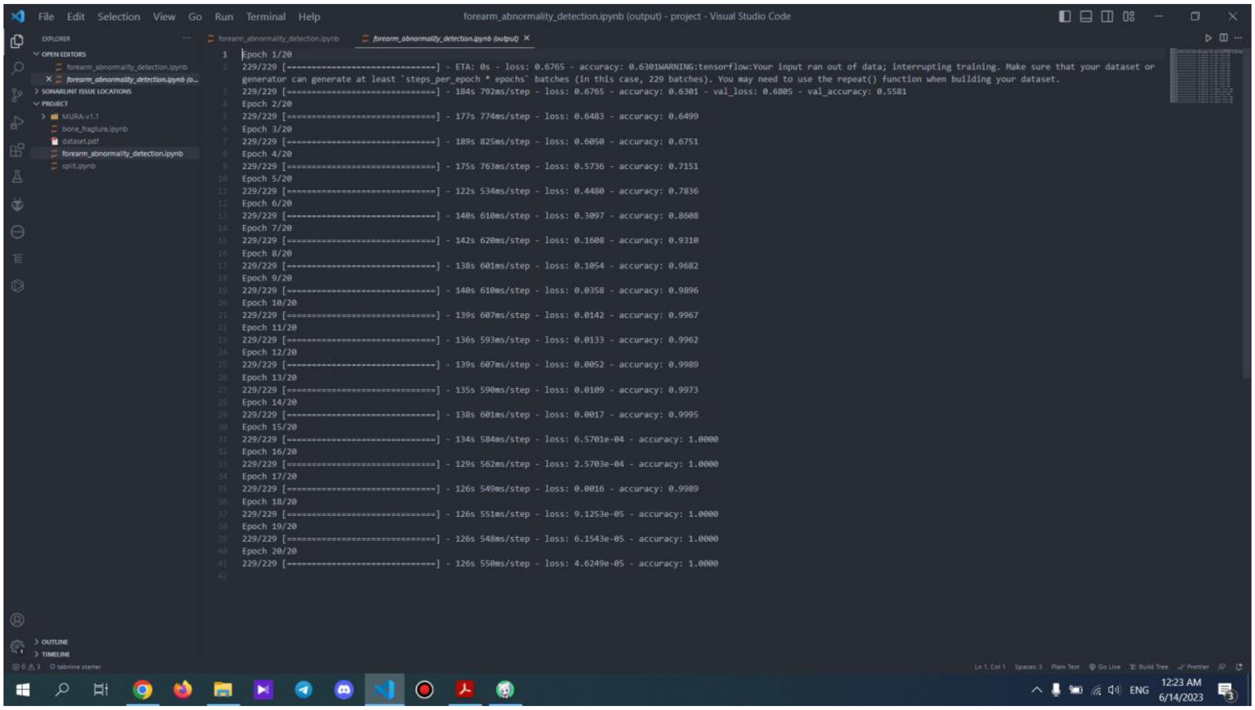
```
model.fit(train_generator, steps_per_epoch=229, epochs=2,
                    validation_data=valid_generator, validation_steps=229, verbose=1)
[112]
```

```
Epoch 1/2
229/229 [==============================] - ETA: 0s - loss: 0.0034 - accuracy: 0.9995WARNING:tensorflow:Your input ran out of data; interrupti
229/229 [==============================] - 155s 678ms/step - loss: 0.0034 - accuracy: 0.9995 - val_loss: 4.9206 - val_accuracy: 0.5615
Epoch 2/2
229/229 [==============================] - 146s 639ms/step - loss: 1.4459e-04 - accuracy: 1.0000

<keras.callbacks.History at 0x25e86673730>
```

```
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
[113]
```

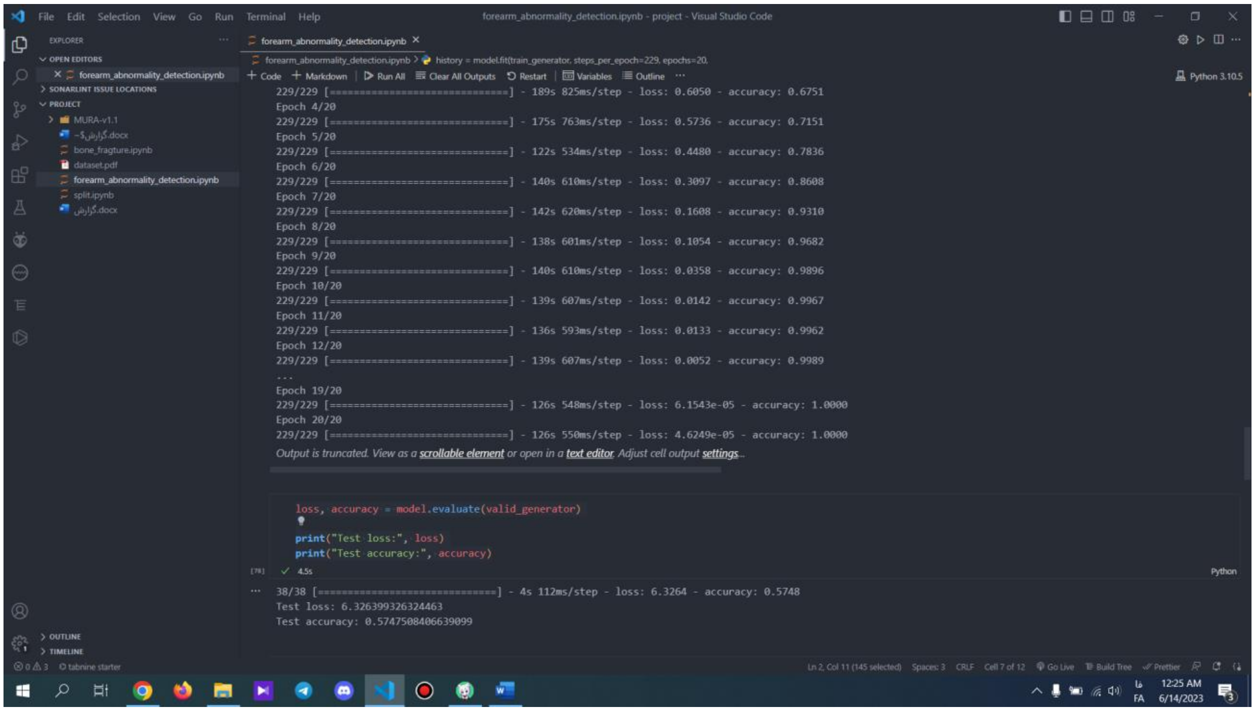
```
38/38 [==============================] - 5s 141ms/step - loss: 5.2555 - accuracy: 0.5615
Test loss: 5.255453109741211
Test accuracy: 0.5614618062973022
```

در تصاویر بعدی دقت در epoch تا epoch شماره 20 را مشاهده می کنیم.



forearm_abnormality_detection.ipynb (output) - project - Visual Studio Code
File
Edit
Selection
View
Go
Run
Terminal
Help
EXPLORER
OPEN EDITORS
forearm_abnormality_detection.ipynb
forearm_abnormality_detection.ipynb (output)
SONARLINT ISSUE LOCATIONS
PROJECT
MURA-v1.1
bone_fragture.ipynb
dataset.pdf
forearm_abnormality_detection.ipynb
split.ipynb
OUTLINE
TIMELINE
12:23 AM
6/14/2023

```
Epoch 1/20
229/229 [==============================] - ETA: 0s - loss: 0.6765 - accuracy: 0.6301WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or
generator can generate at least `steps_per_epoch * epochs` batches (in this case, 229 batches). You may need to use the repeat() function when building your dataset.
229/229 [==============================] - 184s 792ms/step - loss: 0.6765 - accuracy: 0.6301 - val_loss: 0.6805 - val_accuracy: 0.5581
Epoch 2/20
229/229 [==============================] - 177s 774ms/step - loss: 0.6483 - accuracy: 0.6499
Epoch 3/20
229/229 [==============================] - 189s 825ms/step - loss: 0.6050 - accuracy: 0.6751
Epoch 4/20
229/229 [==============================] - 175s 763ms/step - loss: 0.5736 - accuracy: 0.7151
Epoch 5/20
229/229 [==============================] - 122s 534ms/step - loss: 0.4480 - accuracy: 0.7836
Epoch 6/20
229/229 [==============================] - 140s 610ms/step - loss: 0.3097 - accuracy: 0.8608
Epoch 7/20
229/229 [==============================] - 142s 620ms/step - loss: 0.1608 - accuracy: 0.9310
Epoch 8/20
229/229 [==============================] - 138s 601ms/step - loss: 0.1054 - accuracy: 0.9682
Epoch 9/20
229/229 [==============================] - 140s 610ms/step - loss: 0.0358 - accuracy: 0.9896
Epoch 10/20
229/229 [==============================] - 139s 607ms/step - loss: 0.0142 - accuracy: 0.9967
Epoch 11/20
229/229 [==============================] - 136s 593ms/step - loss: 0.0133 - accuracy: 0.9962
Epoch 12/20
229/229 [==============================] - 139s 607ms/step - loss: 0.0052 - accuracy: 0.9989
Epoch 13/20
229/229 [==============================] - 135s 590ms/step - loss: 0.0109 - accuracy: 0.9973
Epoch 14/20
229/229 [==============================] - 138s 601ms/step - loss: 0.0017 - accuracy: 0.9995
Epoch 15/20
229/229 [==============================] - 134s 584ms/step - loss: 6.5701e-04 - accuracy: 1.0000
Epoch 16/20
229/229 [==============================] - 129s 562ms/step - loss: 2.5703e-04 - accuracy: 1.0000
Epoch 17/20
229/229 [==============================] - 126s 549ms/step - loss: 0.0016 - accuracy: 0.9989
Epoch 18/20
229/229 [==============================] - 126s 551ms/step - loss: 9.1253e-05 - accuracy: 1.0000
Epoch 19/20
229/229 [==============================] - 126s 548ms/step - loss: 6.1543e-05 - accuracy: 1.0000
Epoch 20/20
229/229 [==============================] - 126s 550ms/step - loss: 4.6249e-05 - accuracy: 1.0000
```



forearm_abnormality_detection.ipynb - project - Visual Studio Code
File
Edit
Selection
View
Go
Run
Terminal
Help
EXPLORER
OPEN EDITORS
forearm_abnormality_detection.ipynb
SONARLINT ISSUE LOCATIONS
PROJECT
MURA-v1.1
~$گزارش.docx
bone_fragture.ipynb
dataset.pdf
forearm_abnormality_detection.ipynb
split.ipynb
گزارش.docx
forearm_abnormality_detection.ipynb > history = model.fit(train_generator, steps_per_epoch=229, epochs=20,
+ Code
+ Markdown
Run All
Clear All Outputs
Restart
Variables
Outline
Python 3.10.5
Python
OUTLINE
TIMELINE
12:25 AM
6/14/2023

```
229/229 [==============================] - 189s 825ms/step - loss: 0.6050 - accuracy: 0.6751
Epoch 4/20
229/229 [==============================] - 175s 763ms/step - loss: 0.5736 - accuracy: 0.7151
Epoch 5/20
229/229 [==============================] - 122s 534ms/step - loss: 0.4480 - accuracy: 0.7836
Epoch 6/20
229/229 [==============================] - 140s 610ms/step - loss: 0.3097 - accuracy: 0.8608
Epoch 7/20
229/229 [==============================] - 142s 620ms/step - loss: 0.1608 - accuracy: 0.9310
Epoch 8/20
229/229 [==============================] - 138s 601ms/step - loss: 0.1054 - accuracy: 0.9682
Epoch 9/20
229/229 [==============================] - 140s 610ms/step - loss: 0.0358 - accuracy: 0.9896
Epoch 10/20
229/229 [==============================] - 139s 607ms/step - loss: 0.0142 - accuracy: 0.9967
Epoch 11/20
229/229 [==============================] - 136s 593ms/step - loss: 0.0133 - accuracy: 0.9962
Epoch 12/20
229/229 [==============================] - 139s 607ms/step - loss: 0.0052 - accuracy: 0.9989
...
Epoch 19/20
229/229 [==============================] - 126s 548ms/step - loss: 6.1543e-05 - accuracy: 1.0000
Epoch 20/20
229/229 [==============================] - 126s 550ms/step - loss: 4.6249e-05 - accuracy: 1.0000
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```python
loss, accuracy = model.evaluate(valid_generator)

print("Test loss:", loss)
print("Test accuracy:", accuracy)
```

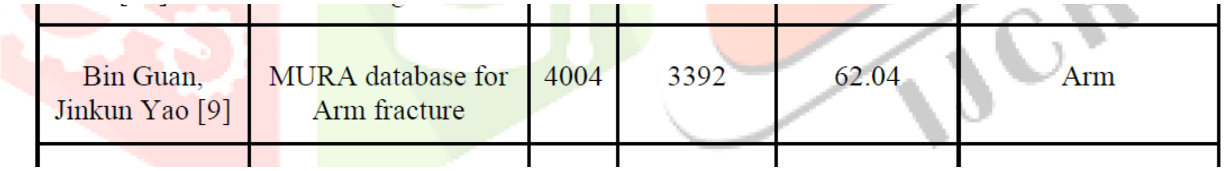
[78] ✓ 4.5s

```
38/38 [==============================] - 4s 112ms/step - loss: 6.3264 - accuracy: 0.5748
Test loss: 6.326399326324463
Test accuracy: 0.5747508406639099
```

بررسی دقت :

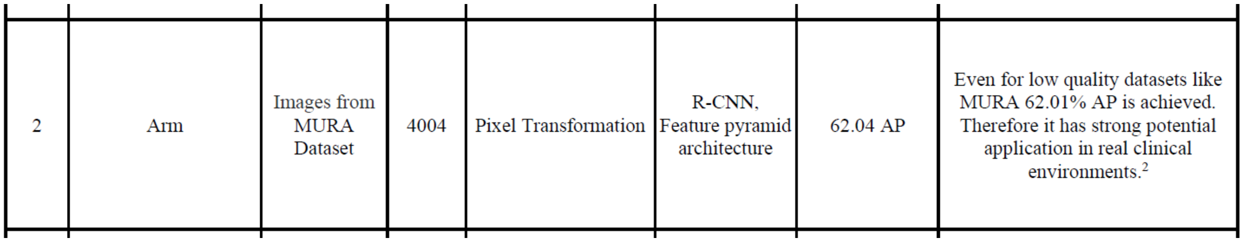
بهترین دقت به دست آمده برای این مدل CNN 9 لایه ای در epoch نهم به دست می آید که برابر با 58.47 است.

دقت در مقاله های مشابه :



Bin Guan, Jinkun Yao [9]	MURA database for Arm fracture	4004	3392	62.04	Arm

و



2	Arm	Images from MURA Dataset	4004	Pixel Transformation	R-CNN, Feature pyramid architecture	62.04 AP	Even for low quality datasets like MURA 62.01% AP is achieved. Therefore it has strong potential application in real clinical environments.[2]

همین طور که در مقاله دوم گفته شد با توجه به کیفیت پایین دیتاست نتیجه خوبی حاصل شده (مدل cnn قوی 169 لایه ای)