

بسمه تعالی

سید محمدرضا حسینی

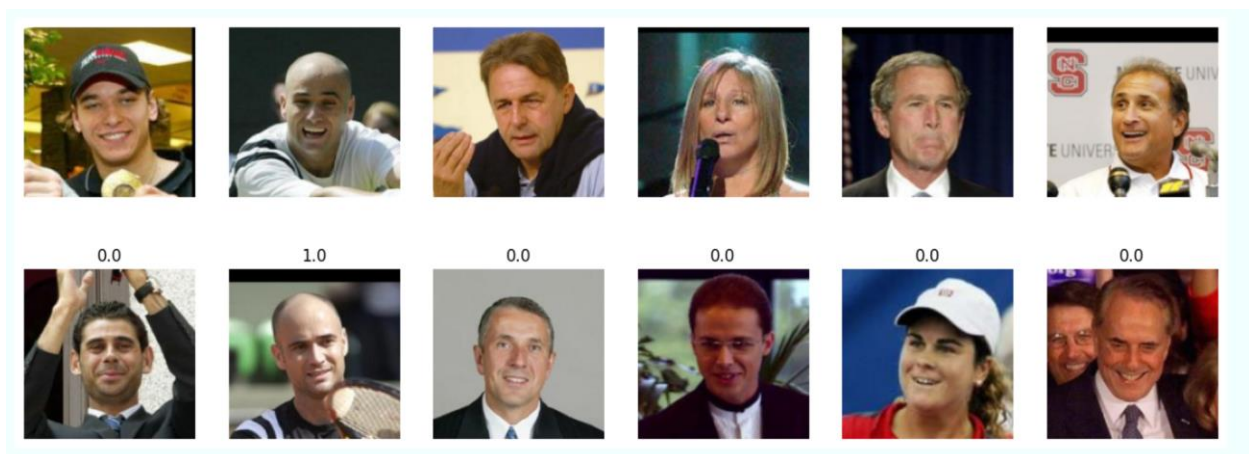
۹۷۲۴۳۱۲۹

تمامی موارد در Jupyter notebook وجود دارد

در این پروژه از Transfer learning استفاده میکنیم . ابتدا مدل از پیش ترین شده Facenet که با استفاده از دیتاست MS-celeb ترین شده را استفاده کرده ایم . هدف تشکیل یک شبکه Siamese سات که با لاس فانکشن Contrastive Loss ترین میشود . مدل Face_verifier به این شکل است که دو عکس داده شده به شبکه هر کدام از یک شبکه Facenet عبور کرده و پس از گرفتن خروجی embedded vector فاصله اقلیدسی بین آن ها محاسبه شده و پس از عبور از یک لایه Normalization ، نتیجه از لایه خروجی با یک نورون با اکتویشن فانکشن sigmoid میزان شباهت را محاسبه میکند.

تشکیل دیتاست

در ابتدا ۲۵۰۰ نفر از افراد برای تشکیل دیتاست به صورت تصادفی انتخاب میکند . پس از آن باید داده ها به صورت Positive pair و Negative pair تشکیل شود . در Positive pair دو تصویر از یک فرد و در Negative Pair دو تصویر از دو فرد متفاوت وجود دارد . سپس batch های ۳۲ تایی از داده ها را تشکیل میدهیم.



در تصویر بالا نمونه ای زوج های تشکیل شده مشخص است . از تمامی زوج های تشکیل شده نیمی NEGATIVE PAIR و نیمی دیگر Pos pair میباشد.

```
def Face_verifier():
    input1 = keras.Input(shape = (160,160,3),name="input1")
    input2 = keras.Input(shape =(160,160,3),name="input2")
    embedding1 =myFacenet (input1)
    embedding2 =myFacenet (input2)
    merge_layer = layers.Lambda(euclidean_distance)([embedding1,embedding2])
    normal_layer = tf.keras.layers.BatchNormalization()(merge_layer)
    outputs = layers.Dense(1, activation="sigmoid")(normal_layer)
    model = Model(inputs=[input1, input2], outputs=outputs)
    return model
```

هدف کلی در این مدل Fine tune کردن Facenet و ترین کردن لایه های انتهایی اضافه شده توسط ماست که در آن مدل بتواند با استفاده از عکس های دیتاست LFW ، وریفای کند که آیا تصاویر متعلق به یک فرد میباشد یا خیر .

```
val_accuracy_sim: 0.7827 - val_f1_m: 0.6864 - val_precision_m: 0.6612 - val_recall_m: 0.7664 - val_accuracy: 0.7817
Epoch 7/10
323/323 [=====] - 77s 239ms/step - loss: 0.0125 - accuracy_sim: 0.9672 - f1_m: 0.9652 - precision_m: 0.9424 - recall_m: 0.9928 - accuracy: 0.9672 - val_loss: 0.0778 -
val_accuracy_sim: 0.8307 - val_f1_m: 0.7216 - val_precision_m: 0.6940 - val_recall_m: 0.7925 - val_accuracy: 0.8301
Epoch 8/10
323/323 [=====] - 78s 240ms/step - loss: 0.0102 - accuracy_sim: 0.9734 - f1_m: 0.9722 - precision_m: 0.9557 - recall_m: 0.9916 - accuracy: 0.9734 - val_loss: 0.0795 -
val_accuracy_sim: 0.8194 - val_f1_m: 0.7177 - val_precision_m: 0.6859 - val_recall_m: 0.8065 - val_accuracy: 0.8196
Epoch 9/10
323/323 [=====] - 77s 238ms/step - loss: 0.0088 - accuracy_sim: 0.9768 - f1_m: 0.9752 - precision_m: 0.9580 - recall_m: 0.9951 - accuracy: 0.9768 - val_loss: 0.1051 -
val_accuracy_sim: 0.8077 - val_f1_m: 0.7009 - val_precision_m: 0.7128 - val_recall_m: 0.7369 - val_accuracy: 0.8061
Epoch 10/10
323/323 [=====] - 78s 243ms/step - loss: 0.0086 - accuracy_sim: 0.9777 - f1_m: 0.9762 - precision_m: 0.9618 - recall_m: 0.9930 - accuracy: 0.9777 - val_loss: 0.0831 -
val_accuracy_sim: 0.8343 - val_f1_m: 0.7161 - val_precision_m: 0.7090 - val_recall_m: 0.7640 - val_accuracy: 0.8346
```

در نهایت پس از ده epoch ، تمامی معیار ها به خوبی نشان میدهد که مدل توانسته به خوبی بین افراد تمایز قائل شود.



```
def load_img(image_path):  
    img = tf.keras.preprocessing.image.load_img(image_path, target_size=(160, 160))  
    x_train = np.expand_dims(img, axis=0)  
    return x_train  
  
pred=Verifier_model.predict([load_img("image/messi/6.jpeg"), load_img("image/messi/5.jpeg")])  
if np.squeeze(pred)<0.8:  
    print("Not the same person")  
    print(pred)  
else:  
    print("Same Person")
```

Same Person

شبکه به درستی توانست تشخیص دهد هر دو عکس متعلق به یک فرد میباشد.



```
def load_img(image_path):  
    img = tf.keras.preprocessing.image.load_img(image_path, target_size=(160, 160))  
    x_train = np.expand_dims(img, axis=0)  
    return x_train  
pred=Verifier_model.predict([load_img("image/lfw/Abdoulaye_Wade/Abdoulaye_Wade_0001.jpg"), load_img("image/lfw/Beth_Blough/Beth_Blough_0001.jpg")])  
if np.squeeze(pred)<0.8:  
    print("Not the same person")  
    print(pred)  
else:  
    print("Same Person")
```

Not the same person
[[0.0011983]]

شبکه به درستی تشخیص داد که این دو عکس برای یک فرد نمیباشد.