



گزارش پروژه‌ی پایانی مبانی بینایی کامپیوتر  
استاد نظری

حمید الهی نراقی - ۹۷۲۴۳۰۰۵

## خواسته‌ی پروژه:

در این پروژه می‌بایست سیستمی پیاده‌سازی می‌شد که دو عکس را به عنوان ورودی بگیرد و فارغ از تشخیص هویت افراد، تشخیص دهد که آیا دو تصویر مربوط به یک نفر هست یا خیر.

## راهبرد:

برای ساخت چنین سیستمی از یک مدل از پیش آموزش داده شده (pre-trained) به نام vggFace استفاده کردم. در این شبکه‌ی عصبی که از نوع پیچشی (CNN) است به ازای هر عکس، یک بردار متناظر در فضای n بعدی تولید می‌کند که با مقایسه فاصله‌ی این نقاط می‌توان تشخیص داد که تا چه حد شبیه به یکدیگر هستند. به این شکل که هرچه فاصله دو نقطه در فضای n بعدی کمتر و نزدیک‌تر به صفر باشد، تصاویر متناظر با این نقاط نیز شبیه‌تر به هم هستند.

## پیاده‌سازی:

با استفاده از زبان پایتون و با کمک کتابخانه‌ی keras یک مدل ترتیبی طبق معماری مدل معروف vggFace ساختم و لایه‌های کانولوشنی و max-pooling را به ترتیب در نظر گرفتم. معماری کلی مدل به شرح زیر است:

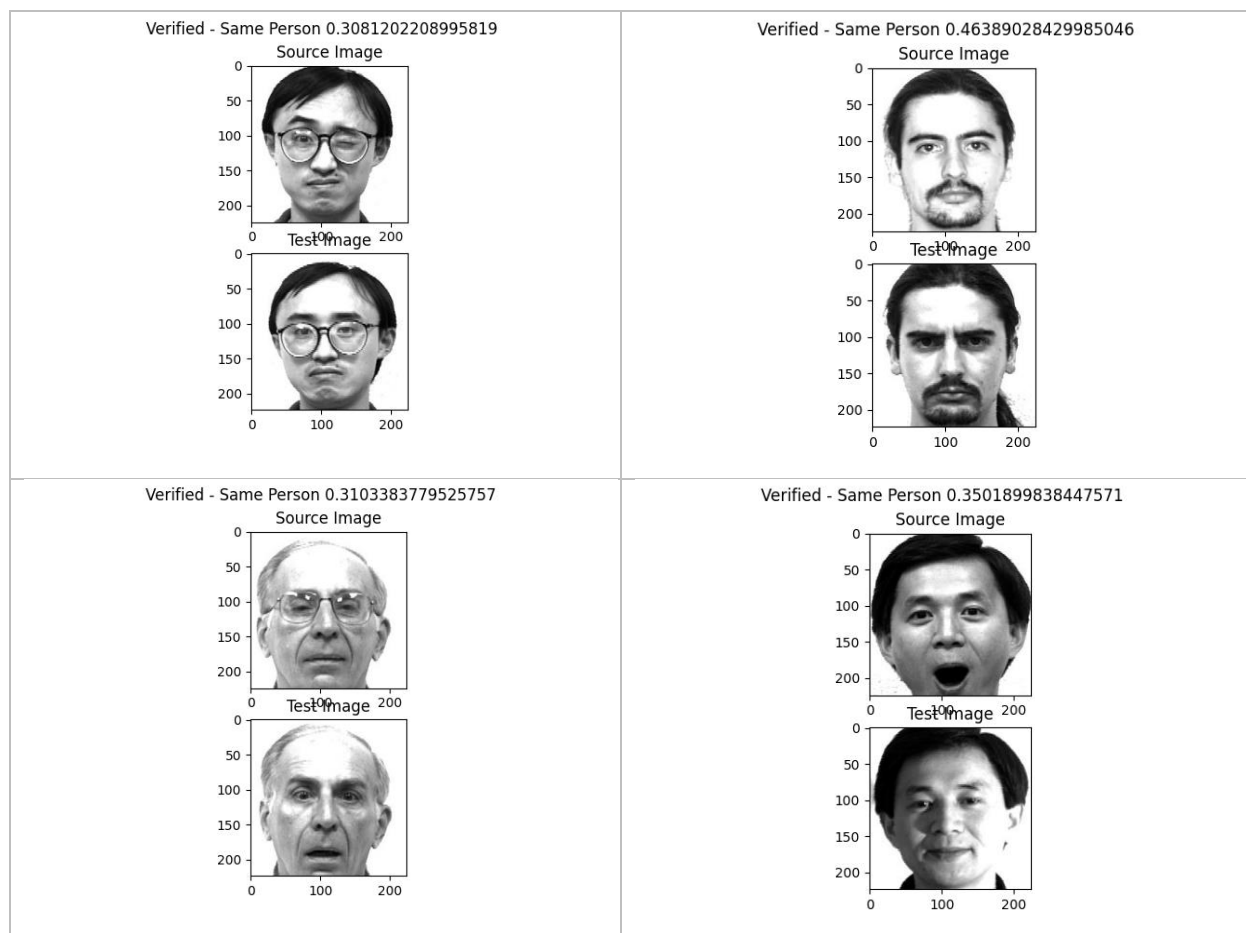
Layer (type)	Output Shape	Param #
zero_padding2d_input (Input Layer)	[(None, 224, 224, 3)]	0
zero_padding2d (ZeroPadding2D)	(None, 226, 226, 3)	0
conv2d (Conv2D)	(None, 224, 224, 64)	1792
zero_padding2d_1 (ZeroPadding2D)	(None, 226, 226, 64)	0
conv2d_1 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
zero_padding2d_2 (ZeroPadding2D)	(None, 114, 114, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 128)	73856
zero_padding2d_3 (ZeroPadding2D)	(None, 114, 114, 128)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 128)	0
zero_padding2d_4 (ZeroPadding2D)	(None, 58, 58, 128)	0
conv2d_4 (Conv2D)	(None, 56, 56, 256)	295168
zero_padding2d_5 (ZeroPadding2D)	(None, 58, 58, 256)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
zero_padding2d_6 (ZeroPadding2D)	(None, 58, 58, 256)	0
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
zero_padding2d_5 (ZeroPadding2D)	(None, 58, 58, 256)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	590080
zero_padding2d_6 (ZeroPadding2D)	(None, 58, 58, 256)	0
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 256)	0
zero_padding2d_7 (ZeroPadding2D)	(None, 30, 30, 256)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	1180160
zero_padding2d_8 (ZeroPadding2D)	(None, 30, 30, 512)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	2359808
zero_padding2d_9 (ZeroPadding2D)	(None, 30, 30, 512)	0
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 512)	0
zero_padding2d_10 (ZeroPadding2D)	(None, 16, 16, 512)	0
conv2d_10 (Conv2D)	(None, 14, 14, 512)	2359808
dropout_1 (Dropout)	(None, 1, 1, 4096)	0
conv2d_15 (Conv2D)	(None, 1, 1, 2622)	10742334
flatten (Flatten)	(None, 2622)	0
Total params: 145,002,878		
Trainable params: 145,002,878		
Non-trainable params: 0		

در تمامی لایه‌ها از تابع فعال‌سازی relu و در لایه‌ی آخر پس از لایه‌ی faltten از تابع فعال‌سازی softmax استفاده شده است. در نهایت وزن‌های مدل را به کمک دستور زیر در مدل ساخته شده load کردم:

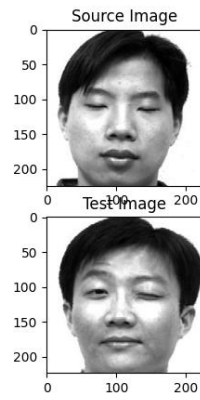
```
model.load_weights('data/vgg_face_weights.h5')
```

### سنجش:

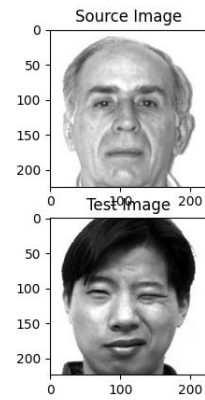
برای سنجش صحت مدل، توابع کمکی random\_pairs و Distance را به ترتیب برای ساخت جفت عکس‌های ورودی و محاسبه فاصله‌ی اقلیدسی دو نقطه نوشتم. در نهایت در کلاس verification اقدام به ساخت شیء مدل کردم و ۱۰ جفت عکس را برای دادن ورودی به مدل به صورت رندوم انتخاب کردم. با هر جفت عکسی که به مدل داده می‌شود، می‌توان فاصله‌ی بین نقاط متناظر آن‌ها را محاسبه کرد و میزان شباهت دو چهره را محاسبه کرد. مرز 0.55 به عنوان threshold در نظر گرفته شده و فاصله‌ی کمتر از آن به معنای شباهت دو چهره است. در ادامه تشخیص مدل برای شباهت یا عدم شباهت برخی از جفت عکس‌های ورودی آورده شده است:



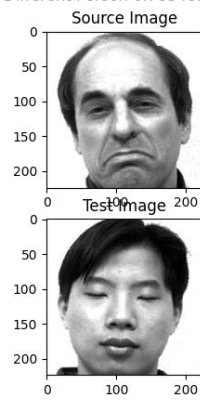
Verified - Different Person 0.5508731603622437



Verified - Different Person 0.8495126366615295



Verified - Different Person 0.7834818363189697



Verified - Different Person 0.8246088624000549

