

توجه: در انجام تمرین استفاده از اینترنت و مشورت مجاز می باشد اما کپی کردن تمرین حتی یک قسمت مجاز نمی باشد و در صورت مشاهده نمره کل تمرین صفر در نظر گرفته خواهد شد. برای همه تمرین نیاز است که گزارش کامل همراه کدها تحویل داده شود و گزارش نیز نمره قابل توجهی دارد. در کدها هم به میزان نیاز حتما کامنت گذاری انجام شود. در صورتی که سوالی در مورد تمرین داشتید، ترجیحا آن را در قسمت "پرسش و پاسخ" تمرین مطرح نمایید تا سوال دانشجویان دیگر نیز برطرف شود. در صورت نیاز با ایمیل های aref.einizade@yahoo.com یا parsa.rahimi.n@gmail.com نیز می توانید مکاتبه نمایید.

الف) تمرین کامپیوتری:

در این تمرین قصد داریم با استفاده از ویژگی های استخراج شده از HOG و طبقه بند SVM یک سیستم برای مکان یابی چهره انسان در یک تصویر پیاده سازی کنیم. برای این کار از پکیج OpenCV برای استخراج ویژگی های HOG و از پکیج Scikit-Learn برای طبقه بند SVM استفاده می کنیم.

(۱) آرگومان های تابع `cv2.HOGDescriptor` را مشاهده کنید و بیان کنید هر کدام بیانگر چه مشخصاتی هستند و چه رابطه ای با تعداد ویژگی های استخراج شده توسط HOG دارند. همچنین بیان کنید در مقاله پیوست هر کدام چه مقداری دارند؟

(۲) کدی بنویسید که ویژگی های HOG یک تصویر را استخراج کرده و در فایل ذخیره کند.

(۳) در این تمرین قصد داریم با استفاده از `HOGDescriptor` ویژگی های مؤثری برای مکان یابی چهره انسان در عکس استخراج کنیم (همچنین می توانید به جای `HOGDescriptor` از متد `feature.hog` از پکیج `skimage` هم استفاده کنید). به این منظور برای Patch های حاوی چهره انسان از دیتاست `fetch_lfw_people` از پکیج `sklearn.datasets` استفاده خواهیم کرد. ابتدا با استفاده از دستورات زیر پکیج `sklean` و `skimage` خود را `upgrade` کنید:

```
pip install --upgrade sklearn
```

```
pip install --upgrade skimage
```

نام کلاس حاوی Patch های چهره انسان را `Positive` می گذاریم. می توانید با اجرای خطوط زیر در محیط برنامه نویسی پایتون این دیتاست را دانلود کنید (توجه کنید که اتصال اینترنت هنگام اجرای این خطوط باید برقرار باشد، حجم تقریبی این دیتاست ۲۰۰ مگابایت است):

```
from sklearn.datasets import fetch_lfw_people
```

```
faces = fetch_lfw_people()
```

```
positive_patches = faces.images
```

برای تشکیل Patch های کلاس Negative (Patch های حاوی هر چیزی غیر از چهره انسان) از داده های بخش Data از پکیج Skimage استفاده می کنیم. با اجرای خطوط زیر نیز (مانند قسمت قبل) می توانید به این تصاویر دسترسی داشته باشید:

```
from skimage import data
imgs_to_use = ['camera', 'text', 'coins', 'moon',
               'page', 'clock', 'immunohistochemistry',
               'chelsea', 'coffee', 'hubble_deep_field']
images = [color.rgb2gray(getattr(data, name)())
           for name in imgs_to_use]
```

مشاهده می کنید که از کلاس های مختلفی برای تشکیل کلاس غیر از چهره انسان (Negative) استفاده شده است. برای اینکه عملکرد مدل شما بالاتر برود می توانید کلاس های بیشتری را نیز در اینجا فراخوانی کنید. توجه کنید که در دیتاست بالا (داده های Data از پکیج Skimage) شما به طور مستقیم به Patch های مفید دسترسی پیدا کرده اید، در حالی که در داده های کلاس Negative شما به تعدادی از عکس با سایزهای مختلف دسترسی دارید که برای طبقه بندی درست، نیاز است به میزان کافی از این تصاویر Patch های معنادار (هم سایز با Patch های کلاس Positive) استخراج کنید. به این منظور و برای استخراج این Patch های معنادار می توانید از متد PatchExtractor از پکیج زیر استفاده کنید:

```
from sklearn.feature_extraction.image import PatchExtractor
```

با توجه به اینکه سایز این تصاویر متغیر است برای اینکه مطمئن باشید در Scale مناسبی از این تصاویر این Patch ها را استخراج می کنید، پیشنهاد می شود از ۳ مورد 0.5, 1.0, 2.0 Scale استفاده کنید (می توانید Scale های متنوع تری را هم انتخاب کنید). به منظور پیاده سازی این توضیحات می توانید از خطوط زیر استفاده نمایید:

```
from sklearn.feature_extraction.image import PatchExtractor

def extract_patches(img, N, scale=1.0, patch_size=positive_patches[0].shape):
    extracted_patch_size = tuple((scale * np.array(patch_size)).astype(int))
    extractor = PatchExtractor(patch_size=extracted_patch_size,
                              max_patches=N, random_state=0)
    patches = extractor.transform(img[np.newaxis])
    if scale != 1:
        patches = np.array([transform.resize(patch, patch_size)
                             for patch in patches])
    return patches

negative_patches = np.vstack([extract_patches(im, 1000, scale)
                              for im in images for scale in [0.5, 1.0, 2.0]])
```

بنابراین تا حالا ۲ کلاس Positive و Negative حاوی Patch های معنادار داریم. در ادامه باید با استفاده از HOGDescriptor از این Patch ها ویژگی استخراج کرده و به این ترتیب دسته ویژگی برای داده های کلاس Positive و Negative ساخته می شود. در قسمت بعدی هدف آموزش طبقه بند SVM بر روی داده های این ۲ کلاس می باشد.

۴) طبقه بند SVM در اینجا یک طبقه بند باینری است که با استفاده از آن قصد داریم Patch های چهره انسان با غیر از چهره انسان را تشخیص دهیم. این طبقه بند بین نمونه های مثبت و منفی یک مرز بهینه به دست می آورد. نمونه های مثبت در این مساله داده ها شامل ویژگی های استخراج شده از Patch های چهره انسان و نمونه های منفی شامل Patch های غیر از چهره انسان است. در حقیقت نیاز است که ما در این روش مرز بین دو ناحیه را به خوبی پیدا کنیم. برای استفاده از طبقه بند نیاز است که هرکدام از داده های ما برچسب داشته باشند بنابراین برای هرکدام از داده های دارای چهره انسان برچسب ۱ و در غیر اینصورت برچسب صفر ذخیره نمایید.

۵) در این قسمت ۸۰ درصد کل داده ها را به عنوان داده آموزشی در نظر گرفته و با استفاده از متد SVC در پکیج sklearn مدل را آموزش دهید. سپس مدل خود را بر روی داده های تست (۲۰ درصد باقیمانده از داده ها) ارزیابی کنید و دقت را گزارش نمایید. برای بهتر شدن عملکرد مدل ابتدا با استفاده از متد GridSearchCV پارامترهای این طبقه بند را روی داده های آموزشی بهینه کنید و سپس مدل خود را بر روی داده های تست ارزیابی کنید.

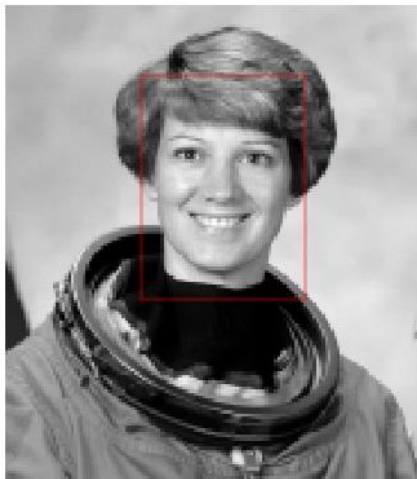
۶) در حال حاضر مدلی داریم که قادر است برای پنجره های با سایز 62×47 وجود یا عدم وجود چهره انسان را تشخیص دهد. برای تشخیص چهره انسان در تصویر در سایر سایزها (تصاویر با سایز دلخواه) لازم است ترتیبی اتخاذ کنیم که سایزشان به گونه ای تغییر کند که چهره ها در آن تصاویر نیز با سایز این پنجره ها همخوانی داشته باشد. سپس پنجره های اسکن شده موجود در تصویر را به مدل SVM آموزش دیده می دهیم. این مراحل را بر روی تصاویر موجود در پوشه Test_images انجام داده و کدها به همراه نتایج را به صورت زیر (تصویر نمونه) گزارش کنید. به این کار تشکیل Image Pyramid یا هرم عکس گفته می شود. با استفاده از این روش چهره انسان در داده های پوشه multiscale را شناسایی کنید و دور آنها را با مستطیل مشخص نمایید. (برای تشکیل هرم تصویر می توانید از متد pyramid_gaussian در پکیج skimage استفاده نمایید)

نکته مهم: هنگام اعمال طبقه بند SVM بر روی تصاویر با سایز دلخواه، ممکن است این طبقه بند چندین مستطیل را در تصویر اصلی حاوی چهره انسان تشخیص دهد. به مانند تصویر نمونه زیر:



در این حالت می بایست با استفاده از Score خروجی طبقه بند، با اعمال آستانه ای معقول، ابتدا Patch هایی که با احتمال کمی عضو کلاس Positive تشخیص داده شده اند را حذف کرد. سپس با استفاده از متد

Non.Maximum.Suppression از پکیج Tensorflow، مستطیل‌های همپوشان به دست آمده را به یک مستطیل با سائز معقول تبدیل کرد. البته برای رسیدن به این منظور راه‌های دیگری نظیر استفاده از متد heat map و ... نیز وجود دارد که استفاده از متدهایی علاوه بر Non.Maximum.Suppression دارای نمره امتیازی خواهد بود. در نهایت تصویر نمونه بالا باید به صورت زیر در آید:

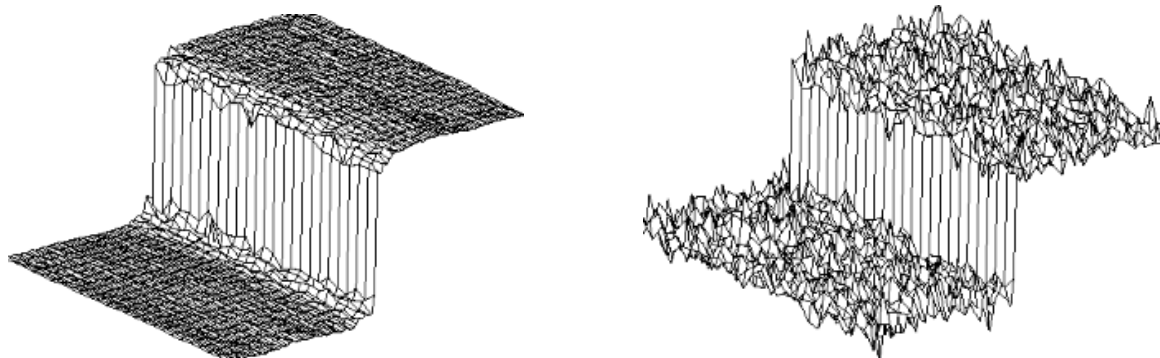


توجه مهم: با توجه به پارامترهای آزاد زیاد این مسئله، ممکن است به آوردن جواب ایده‌آل بر روی تصاویر تست سخت باشد. بنابراین تمرکز اصلی خود را بر روی اجرا کردن گام به گام مراحل بالا قرار دهید؛ زیرا نمره دهی اصلی بر اساس جواب ایده‌آل نهایی نخواهد بود بلکه قسمت عمده نمره را اجرای درست مراحل بالا تشکیل می‌دهد.

ب) تمرین پژوهشی / تئوری:

۱. با مراجعه به این مقاله توضیح دهید که Face Alignment چیست و چه کاربردی دارد. به طور کلی برای رسیدن به هدف Face Alignment چه روش‌هایی وجود دارد (دو تا سه مورد دیگر مانند مقاله اشاره شده را به طور خلاصه ذکر کنید).

۲. در مورد Edge Preserving Smoothing تحقیق کنید دو مورد از متدهای این روش را به طور خلاصه شرح دهید. (این روش به طور کلی باعث می شود که تصویر در مکان هایی که لبه نیستند نرم شود). عملکرد این روش را به طور کلی می توان در شکل های زیر دید:



۳. ماتریس سمت راست از کانوالو کردن یک فیلتر 3×3 (با مرکزیت وسط) با ماتریس سمت چپ حاصل شده است. مقادیر پارامترهای فیلتر و همچنین نوع Border Handling را مشخص کنید. (راهنمایی: Border Handling استفاده شده یکی از BorderType های این لینک است).

12	6	4	2	3
4	6	4	12	6
45	4	1	6	3
34	5	1	7	3

→

79	76	47	42	83
89	86	45	41	93
110	104	46	43	120
115	112	36	30	115

۴. نشان دهید عملگر لاپلاسین که به صورت زیر تعریف می شود ایزوتروپیک است (تغییر ناپذیر با دوران)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

۵. دو تصویر $f(x,y)$ و $g(x,y)$ دارای هیستوگرام h_f و h_g هستند. شرایطی را تعیین کنید که بتوان هیستوگرام عبارات زیر را بر حسب h_f و h_g بدست آورد:

الف) $f(x,y) + g(x,y)$

ب) $f(x,y) - g(x,y)$

ج) $f(x,y) * g(x,y)$

د) $f(x,y) / g(x,y)$

توضیح دهید در هر مورد هیستوگرام به چه صورت بدست می آید.