علی آراسته 96101165

دید کامپیوتری – تمرین دوم

الف) تمارین کامپیوتری:

1) تابع cv2.HOGDescriptor آرگومان‌های زیادی دارد. در زیر به برخی از مهم ترین آن‌ها اشاره می‌کنیم:

blockSize: اندازه block را در واحد pixel مشخص می‌کند. این اندازه مطابق با اندازه cell است. مقدار پیش‌فرض آن اندازه 16*16 پیکسل است. هر چه مقدار این آرگومان کوچکتر باشد، تعداد ویژگی‌های استخراج شده توسط HOG بیشتر است.

blockStride: گام حرکت block را در واحد pixel مشخص می‌کند. باید مضربی از اندازه cell باشد. مقدار پیش‌فرض آن اندازه 8*8 پیکسل است. هر چه مقدار این آرگومان کوچکتر باشد، تعداد ویژگی‌های استخراج شده توسط HOG بیشتر است.

cellSize: اندازه cell را در واحد pixel مشخص می‌کند. مقدار پیش‌فرض آن اندازه 8*8 پیکسل است. هر چه مقدار این آرگومان کوچکتر باشد، تعداد ویژگی‌های استخراج شده توسط HOG بیشتر است.

L2HysThreshold: روش نرمالیزه‌کردن block را مشخص می‌کند.

nbins: تعداد bin در محاسبه هیستوگرام گرادیان‌ها را مشخص مشخص می‌کند. هر چه مقدار این آرگومان بزرگتر باشد، تعداد ویژگی‌های استخراج شده توسط HOG بیشتر است.

nlevels: ماکزیمم تعداد پنجره‌های تشخیص را مشخص می‌کند.

winSigma: اندازه پنجره نرم‌کننده گوسی را مشخص می‌کند.

winSize: اندازه پنجره تشخیص را مشخص می‌کند. این اندازه مطابق با اندازه‌های blockSize و blockStride است. مقدار پیش‌فرض آن اندازه 128*64 پیکسل است.

مقادیر این آرگومان‌ها در مقاله پیوست به شرح زیر است:

RGB color space with no gamma correction; [-1; 0; 1] gradient filter with no smoothing; linear gradient voting into 9 orientation bins in 0 degree to 180 degree; 16*16 pixel blocks of four 8*8 pixel cells; Gaussian spatial window with sigma = 8 pixel; L2-Hys (Lowe-style clipped L2 norm) block normalization; block spacing stride of 8 pixels (hence 4-fold coverage of each cell); 64*128 detection window; linear SVM classifier.

2) کد مربوط به این قسمت در فایل HOGDescriptor آمده است.

3) مراحل گفته‌شده به ترتیب انجام شده است و در آخر با استفاده از تابع features.hog از پکیج skimage، ویژگی Patch های دو کلاس Positive و Negative محاسبه شده است.

4) ابتدا به Patch های کلاس Positive برچسب یک و به Patch های کلاس Negative برچسب صفر اضافه شده است. سپس این همه Patch ها به همراه برچسب‌های خود در یک متغیر ذخیره شده است.

5) با استفاده از متغیری که در قسمت قبل ساخته شد، مدل SVM را آموزش می‌دهیم. همچنین با استفاده از GridSerchCV پارامترهای طبقه‌بند را رویداده‌های آموزشی بهینه می‌شود.

6) ابتدا هرم تصویر داده شده ساخته می‌شود. سپس با جاروب روی تصویرهایی از هرم که اندازه آن‌ها معقول است، تمام پنجره‌هایی با اندازه 47*62 در سایزهای مختلف تصویر که توسط مدل به عنوان صورت تشخیص داده می‌شوند، شناسایی و به پنجره متناظری در تصویر اصلی منتقل می‌شوند. سپس با استفاده از Non.Maximum.Suppression و ماکزیمم‌گیری محلی پنجره‌های نزدیک به هم به یک پنجره تبدیل می‌شوند. در آخر برای هر پنجره باقی‌مانده مستطیلی در تصویر اصلی رسم می‌شود.


Madeline Bracken
Student, United States
MBA

تمارین پژوهشی/تئوری:

1) Face alignment, also known as facial feature detection, refers to locating facial landmarks (such as brows, eyes, nose and corners of mouth) in a face image. In other words, the task of accurately localizing the set of landmark points that define the shape of the face is called face alignment.

The ability of understanding and interpreting facial structures is important for many image analyses tasks. For example, if we want to identify a person from a surveillance camera, a natural approach would be running the face image of the person through a database of known faces, examining the differences and identifying the best match.

There are different methods for face alignment, such as SDM (Supervised descent method), LBF (Local binary features regression) and HPO (Face alignment under poses and occlusion). SDM formulates the face alignment problem as a minimization problem and learns descent directions from training data LBF proposes to improve the accuracy of face alignment by learning local features from training data. By learning features from a local region, the method can take advantage of more discriminative features and save computational effort by learning each feature independently in a local region. The noises in the learned features are suppressed by using a global regression that maps all learned features together to give the optimal motion of landmarks. Both SDM and LBF treat all landmarks equally without considering occlusions and head poses explicitly. To handle occlusion, HPO proposes to take the visibility of landmarks into account and learns a model to predict the probability of landmark visibility. Such model is then used to assist the face alignment by adding the visibility information to local features and shape configuration around a landmark.

2) Edge-preserving smoothing is an image processing technique that smooths away noise or textures while retaining sharp edges. Examples are the median, bilateral, guided, and anisotropic diffusion filters. Edge-preserving filters are designed to automatically limit the smoothing at “edges” in images measured. Requirements of the strict edge preservation commonly limit the smoothing power of the filter, such that a single application of the filter still results in unacceptably large noise away from the edges. A repetitive application of the filter may be useful to reduce the noise, leading to the idea of combining the filter with an iterative method.

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution.

Anisotropic diffusion is a technique aiming at reducing image noise without removing significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image. Anisotropic diffusion resembles the process that creates a scale space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. Each of the resulting images in this family are given as a convolution between the image and a 2D isotropic Gaussian filter, where the width of the filter increases with the parameter.

3) با توجه به قسمت داخلی ماتریس داده شده (ماتریس 3*2 درونی) درفیلتر 3*3 به سورت زیر بدست می‌آید:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

سپس با بررسی حالت‌های مختلف Border Handling نوع آن را مشخص می‌کنیم. در این سوال از نوع wrap ابتدا به صورت عمودی وسپس به صورت افقی استفاده شده است.

4) ابتدا متغیرهای u و v را به صورت زیر تعریف می‌کنیم:

$$u = xsin\theta + ycos\theta$$

$$v = xcos\theta - ysin\theta$$

حال باید نشان دهیم رابطه زیر برای هر θبرقرار است:

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

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x} = \frac{\partial f}{\partial u}sin\theta + \frac{\partial f}{\partial v}cos\theta$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial u}sin\theta + \frac{\partial f}{\partial v}cos\theta\right) = \frac{\partial}{\partial x}\frac{\partial f}{\partial u}sin\theta + \frac{\partial}{\partial x}\frac{\partial f}{\partial v}cos\theta$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial u}\left(\frac{\partial f}{\partial u}sin\theta + \frac{\partial f}{\partial v}cos\theta\right)sin\theta + \frac{\partial}{\partial v}\left(\frac{\partial f}{\partial u}sin\theta + \frac{\partial f}{\partial v}cos\theta\right)cos\theta$$

$$\frac{\partial^2 f}{\partial x^2} = \left(\frac{\partial^2 f}{\partial u^2}sin\theta + \frac{\partial^2 f}{\partial u\partial v}cos\theta\right)sin\theta + \left(\frac{\partial^2 f}{\partial u\partial v}sin\theta + \frac{\partial^2 f}{\partial v^2}cos\theta\right)cos\theta$$

به طور مشابه داریم:

$$\frac{\partial^2 f}{\partial y^2} = \left(\frac{\partial^2 f}{\partial u^2}cos\theta - \frac{\partial^2 f}{\partial u\partial v}sin\theta\right)cos\theta - \left(\frac{\partial^2 f}{\partial u\partial v}cos\theta - \frac{\partial^2 f}{\partial v^2}sin\theta\right)sin\theta$$

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial u^2}(sin^2\theta + cos^2\theta) + \frac{\partial^2 f}{\partial v^2}(sin^2\theta + cos^2\theta) = \frac{\partial^2 f}{\partial u^2} + \frac{\partial^2 f}{\partial v^2}$$

5) The only time that the histogram of the images formed by the operations shown in the problem statement can be determined in terms of the original histograms is when one (both) of the images is (are) constant. In (d) we have the additional requirement that none of the pixels of g (x, y) can be 0. Assume for convenience that the histograms are not normalized, so that, for example, hf(rk) is the number of pixels in f (x, y) having intensity level rk. Assume also that all the pixels in g (x, y) have constant value c. The pixels of both images are assumed to be positive. Finally, let uk denote the intensity levels of the pixels of the images formed by any of the arithmetic operations given in the problem statement. Under the preceding set of conditions, the histograms are determined as follows:

a) We obtain the histogram hsum(uk) of the sum by letting uk = rk + c, and hsum(uk) = hf(rk) for all k. In other words, the values (height) of the components of hsum are the same as the components of hf, but their locations on the intensity axis are shifted right by an amount c.

b) Similarly, the histogram hdiff(uk) of the difference has the same components as hf but their locations are moved left by an amount c as a result of the subtraction operation.

c) Following the same reasoning, the values (heights) of the components of histogram hprod(uk) of the product are the same as hf, but their locations are at uk = c × rk. Note that while the spacing between components of the resulting histograms in (a) and (b) was not affected, the spacing between components of hprod(uk) will be spread out by an amount c.

d) Finally, if $c \neq 0$, the components of hdiv(uk) are the same as those of hf, but their locations will be at uk = rk / c. Thus, the spacing between components of hdiv(uk) will be compressed by an amount equal to 1/c.