

Dissimilarity-based nearest neighbor classifier for single-sample face recognition

Zhengqi Zhang, Li Zhang, Meng Zhang

کلاسبند نزدیکترین همسایگی مبتنی بر عدم شباهت برای شناسایی چهره تک نمونه

یاسمن صفاری

درس شناسایی الگو

استاد: دکتر ابراهیم پور

چکیده:

در تسک های شناسایی چهره با تک نمونه آموزشی، کلاسبند نزدیکترین همسایگی یکی از روشهای معروف از نظر سادگی پیاده سازی است. انتخاب معیار اندازه گیری فاصله مناسب در این مسایل به خاطر تاثیرات نور و سایر مسایل عکاسی باید با دقت بیشتری انجام شود چرا که روشهای سنتی مثل فاصله اقلیدسی دقت خوبی ندارند. در این مقاله یک معیار عدم شباهت یا فاصله معرفی شد. نمونه های آموزشی و تست به صورت پیچ هایی از سگمنتهای هر عکس بودند که معیار فاصله معرفی شده بر روی این پیچ ها اعمال شد. در نهایت دقت کلاسبند نزدیک ترین همسایگی محاسبه شد.

مقدمه و طرح مساله:

مساله شناسایی چهره، یکی از تسک های مهم به خصوص در مسایل امنیتی میباشد. در اغلب متدهای شناسایی چهره به این گونه است که از هر فرد تعدادی عکس در حالات مختلف موجود است و متد یادگیری بعد از استخراج ویژگی های مناسب و جداساز، میتواند هر فرد را تشخیص دهد. اما در اغلب مسایل امنیتی دنیای واقعی این امکان فراهم نیست و فقط یک عکس از هر فرد در دیتابیس موجود است و برای شناسایی افراد فقط به همان یک عکس میتوان رجوع کرد. متدهایی مثل LDA و SVM و NNC با ایجاد تغییراتی برای این تسک به کار میروند.

الگوریتم نزدیکترین همسایگی به خاطر سادگی پیاده سازی و دقت بالاتر و مقاومت بیشتر به نویز مورد توجه است. معیارهای فاصله سنجی معروف در نزدیکترین همسایگی، اقلیدسی و منهتن هستند. اما به خاطر شرایط مختلف عکاسی و ویژگی های پیچیده عکس، غالباً دقت خوبی ندارند. لذا روشهایی مبتنی بر سگمنت کردن عکس ها و ارایه معیارهایی برای اندازه گیری فاصله بین سگمنت های دو عکس، معرفی شدند. بخش بندی کردن عکس ها کمک میکند تا توجه به نقاط خاص و مشترک چهره ها بیشتر شود.

در این مقاله ابتدا هر عکس به پچ های غیر هم پوشا سگمنت شد و یک مجموعه مرتب از پچ ها برای هر عکس تهیه شد. سپس عدم شباهت یا همان فاصله بین هر مجموعه پچ های عکس آموزش و تست محاسبه شد. نهایتاً عکسی از آموزش نزدیکترین عکس به تست است که کمترین عدم شباهت را داشته باشد.

مطالعه پیشین

در ادامه به معرفی الگوریتم نزدیکترین همسایگی و معرفی روش کلی پیشنهادی برای معیار اندازه گیری پرداخته شده است.

کلاسبند نزدیکترین همسایگی

مساله نزدیکترین همسایگی درواقع همان کا نزدیکترین همسایگی با کا برابر یک است. برای این مسایل، یک مجموعه آموزشی شامل تعدادی نمونه که هر نمونه یک بردار ویژگی دارد میباشد. در کلاسبند باید به ازای هر نمونه تست که حاوی یک بردار ویژگی است را بر اساس معیار فاصله مناسب، فاصله اش با هر نمونه آموزش را محاسبه نهایتاً نمونه آموزشی با کمترین فاصله از نمونه تست مدنظر را به عنوان همکلاس نمونه تست در نظر گرفته و لیبلس را به نمونه تست نیز بدهد. معروفترین معیار فاصله در مسایل نزدیکترین همسایگی را میتوان فاصله اقلیدسی و منهتن نام برد که این فاصله ها روی بردارهای ویژگی دو نمونه ترین و تست اعمال میشوند.

$$D_{man} = \sum_{i=1}^p |x_i - y_i| \quad D_{euc} = \left(\sum_{i=1}^p (x_i - y_i)^2 \right)^{\frac{1}{2}}$$

$$\hat{u} = u_{i^*}$$

where

$$i^* = \arg \min_{i=1, \dots, l} d(u', u_i)$$

معیار فاصله یا عدم شباهت

در این مقاله با استفاده از روش بیان شده در مقاله ای، برای محاسبه فاصله بین دو دیتاست بر اساس داده های اصلی، معیار فاصله نزدیکترین همسایگی تعیین شد. لذا در این قسمت به معرفی متد مربوطه پرداخته میشود.

فرض کنید که دو دیتاست در غالب دو ماتریس $l \times d$ داریم. که سطرها نمونه ها هستند و ستونها ویژگیها یا درواقع بعد هر نمونه.

ابتدا یک ماتریس مشترک بر حسب دو دیتاست میسازیم.

$$\mathbf{R} = \frac{l-1}{2l-1} \mathbf{R}_1 + \frac{l-1}{2l-1} \mathbf{R}_2$$

where

$$\mathbf{R}_t = \frac{1}{l-1} \mathbf{a}_t^T \mathbf{a}_t, \quad t = 1, 2$$

حالا هدف اینست که با حذف همبستگی ویژگی ها، کار آموزش شبکه را بهبود ببخشیم. لذا از تبدیلی به نام whitening یا بردار نویز سفید استفاده میکنیم. هدف اینست تا ماتریس دیتاها را با یک تبدیلی به فضای جدیدی ببریم که همبستگی متغیرها صفر شود. برای این امر ابتدا ماتریس دیتا را برای هر دیتاست بریداریم. هر ستون یا هر ویژگی در هر ماتریس را zero-center میکنیم یا به عبارتی هر مقدار ویژگی برای هر نمونه را منهای میانگی مقادیر همان ویژگی میکنیم. حال واریانس هر ستون را محاسبه میکنیم. سپس بین هر دو ستون یا هر دو ویژگی کواریانس میگیریم. همین محاسبات را در حالتی که تعداد نمونه ها را به علاوه یک (بدون بایاس) است میکنیم. حال بردار و مقادیر ویژه ماتریس کواریانس هر ویژگی را میابیم. سپس یکی از روش های pca یا zca را برای تجزیه مولفه های ویژگی ها استفاده میکنیم. (توضیحات whitening در مقاله نبود و صرفا برای فهم بهتر خودم تحقیق کردم)

$$\mathbf{R}\mathbf{P}_0 = \mathbf{P}_0\mathbf{A}$$

$$\mathbf{P} = \mathbf{P}_0\mathbf{A}^{-1/2}$$

خب پس تا اینجا بردارها و مقادیر ویژه ماتریس مشترک را محاسبه و ماتریس تبدیل را به دست میآوریم. و ماتریس دیتاست ها را بر حسب تبدیل به دست آمده به فضای با غیرهمبستگی ویژگی ها مپ میکنیم.

حال دو ماتریس بر اساس مپ های جدید دیتا ست ها محاسبه میکنیم.

$$\mathbf{S}_l = \frac{1}{l-1} \bar{\mathbf{a}}_l^T \bar{\mathbf{a}}_l, \quad l = 1, 2$$

سپس مقادیر ویژه ماتریس های جدید را محاسبه میکنیم.

از آنجایی که مجموع دو ماتریس برابر همانی میشود (نفهمیدم چرا و در مقاله اصلا ذکر نشده که چرا) لذا پس مقادیر ویژه دو ماتریس مجموعشان در هر سطر برابر یک خواهد بود.

در نهایت عدم شباهت دو دیتا ست محاسبه میشود (d=بعد) (لاندا میتواند لانداهای هر کدام از ماتریسها باشد).

$$d(\mathbf{a}_1, \mathbf{a}_2) = \frac{4}{D} \sum_{j=1}^D (\lambda_j - 0.5)^2$$

پیاده سازی

تمام پیاده سازی ها در محیط کولب به زبان پایتون نوشته شد.

دیتاست

دیتا ست اصلی مقاله AR میبود ولی از آنجایی که حداقل دو هفته برای دریافت تایید پروفسور مسیول دیتاست زمان نیاز بود و این زمان برای بنده مقدر نبود لذا دیتاستی مشابه با حجم کمتر و مناسبتر برای کولب و همینطور پیشفرض کتابخانه scikit learn مورد استفاده قرار گرفت.

```
[42] 1 data=np.load("../content/olivetti_faces.npy")
      2 target=np.load("../content/olivetti_faces_target.npy")
```

Figure 1: لود دیتاست

دیتاست مورد استفاده ما شامل 400 عکس از ده فرد متفاوت است که هر فرد در حالات متفاوت چهره عکس انداخته است. تمام عکس های gray هستند و سائز پیکسلها نرمال شده و تقسیم بر 255 شدند. سائز عکسها نیز 64 در 64 است. لذا ماتریس دیتا شامل 400 عکس یا سطر و 4096 پیکسل یا عکس است. هر عکس با یک عدد لیبل گذاری شده که شماره افراد داخل دیتاست است که از 0 تا 39 میباشد.

```
[10] 1 print("There are {} images in the dataset".format(len(data)))
      2 print("There are {} unique targets in the dataset".format(len(np.unique(target))))
      3 print("Size of each image is {}x{}".format(data.shape[1],data.shape[2]))
      4 print("Pixel values were scaled to [0,1] interval. e.g:{}".format(data[0][0,:4]))

There are 400 images in the dataset
There are 40 unique targets in the dataset
Size of each image is 64x64
Pixel values were scaled to [0,1] interval. e.g:[0.30991736 0.3677686 0.41735536 0.44214877]
```

```
[11] 1 print("unique target number:",np.unique(target))

unique target number: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]
```

Figure 2: مشخصات دیتاست

برای تقسیم بندی ترین و تست، از آنجا که عکسها به طور مرتب ده تا دهتا پشت سرهم از هر فرد قرار گرفته بودند، و ما نیاز به یک عکس از هر فرد در ترین داریم و از آنجا که در مقاله اصلی و دیتاست اصلی، حالت چهره در عکس اول هر فرد نرمال است و در دیتاست مذکور نیز تقریبا چنین است، لذا تمام عکسهای اول هر ست دهتایی افراد را به عنوان ترین که مجموعا 40 عکس میشد گرفتیم و مابقی را در مجموعه تست گذاشتیم که مجموعا 360 عکس میشد.

There are 40 distinct people in the dataset



Figure 3: عکس اول هر 40 نفر

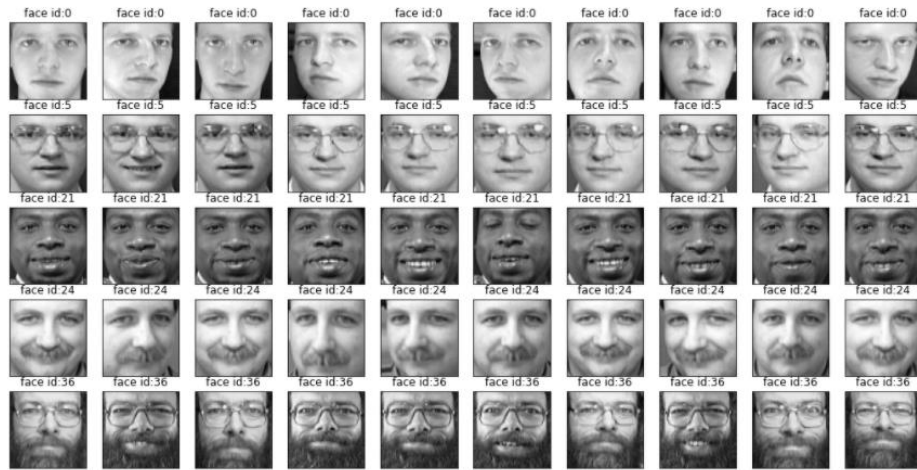


Figure 4: همه 10 عکس 5 نفر

```
[44] 1 t=[]
      2 tr=[]
      3 for i in range(0,399):
      4     if(i%10 ==0):
      5         tr.append(i)
      6     elif(i%10 !=0):
      7         t.append(i)
      8
      9
     10 trImages=X[tr]
     11 trLabels=y[tr]
     12 tImages=X[t]
     13 tLabels=y[t]
     14 print(type(trImages[0]))
```

Figure 5: نحوه تقسیم دیتاست به ترین و تست

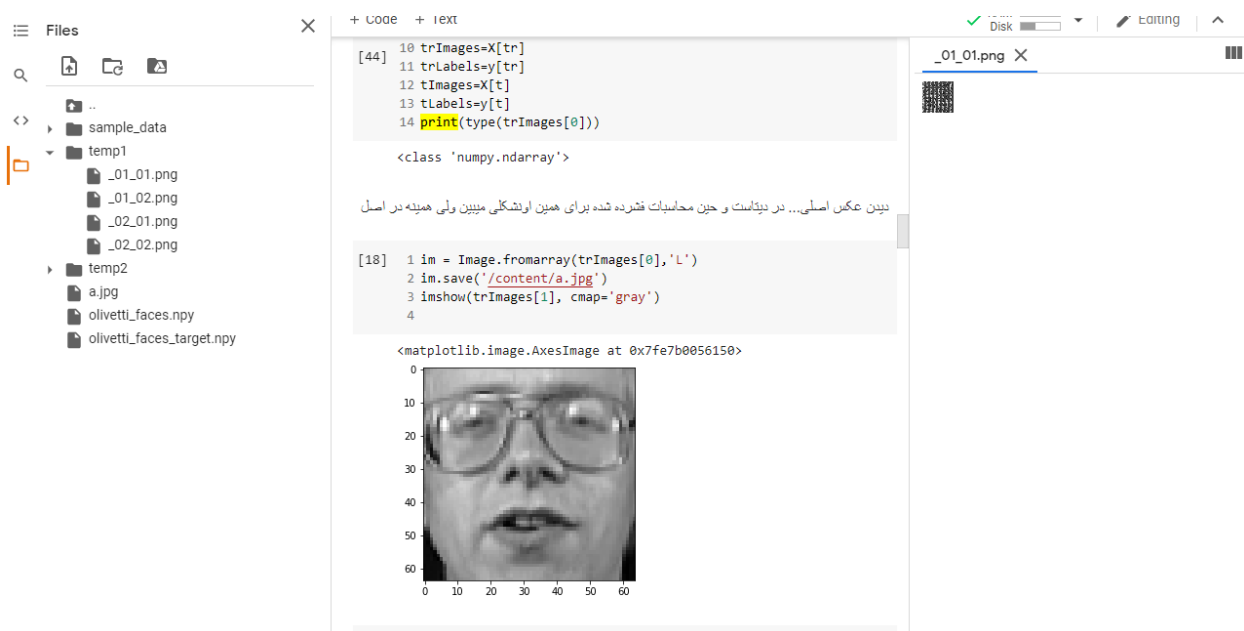


Figure 6: به دلیل فشرده شدن و نرمال شدن پیکسلها، عکس در فولدر به شکل فشرده دیده میشود اما در صورت نرمالایز کردن میتوان شکل اصلی را دید.

الگوریتم

ابتدا توابع لازم برای محاسبه معیار فاصله پیاده شدند. همانطور که گفته شد برای مقایسه هر دو عکس ابتدا هر دو تا عکس سگمنت میشوند که در این پیاده سازی به طور ثابت همه سگمنت ها را 2×2 در نظر گرفتیم. یعنی

هر عکس چهار قسمت مساوی میشد. سپس به طور موقت در فایل‌های حافظه موقت قرار می‌گرفتند و فاصله بین هر دو پچ مرتبط با هم محاسبه میشد. این نحوه محاسبه فاصله در ادامه به طور مفصل توضیح داده میشود. تا اینجا فرض کنید فاصله هر پچ دو عکس با هم را داریم. سه روش کلی برای محاسبه فاصله کلی دو عکس اصلی است. یا فاصله پچ‌ها را می‌گیریم یا کمینه یا بیشینه فاصله را برگردانیم به عنوان فاصله دو عکس اصلی. که طبق پیشنهاد مقاله بر اساس آزمایشاتشان روش میانگین بهترین دقت را داشت که ما هم همان روش را برگزیدیم.

حال برای پیاده سازی نزدیک ترین همسایگی، به ازای هر عکس تست، فاصله اش را با تمام نمونه های ترین محاسبه میکنیم. سپس لیبل عکس با نزدیکترین فاصله با تست را برمی‌داریم و به عکس تست میدهیم.

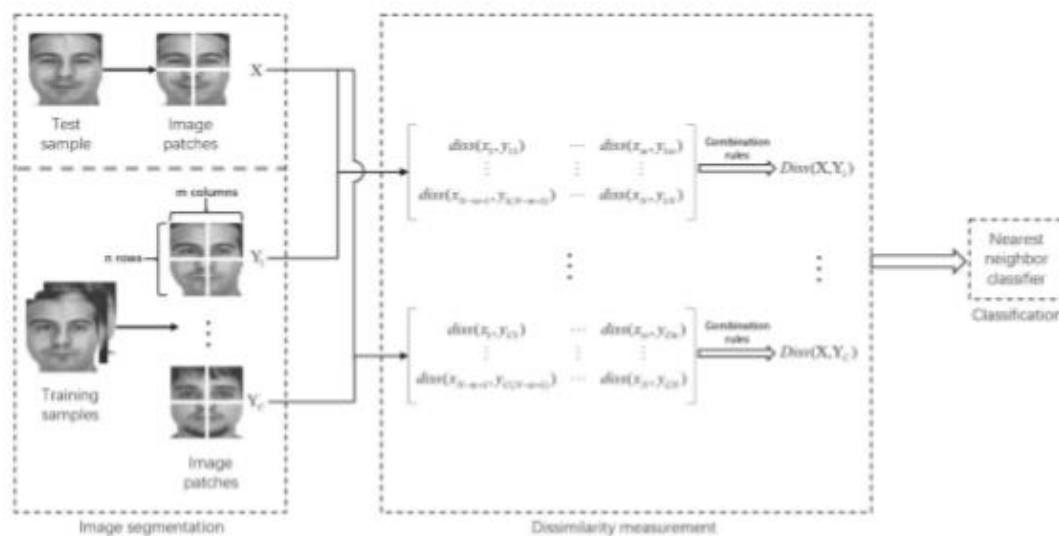


Figure 7: فاصله دو عکس

محاسبه فاصله دو عکس

هر دو عکس مد نظر را ابتدا به چهار قسمت تقسیم میکنیم. هر قسمت یک عکس 32×32 خواهد بود. قرار است برای محاسبه فاصله از روش قبلا گفته شده استفاده کنیم. اما در اینجا دیگر معلوم نیست که در هر سگمنت، معنای سطر و ستون چیست و چه نسبتی با تعداد نمونه و بعد دارد لذا در و حالت که هر سطر و ستون میتواند تعداد نمونه یا بعد باشد را در نظر میگیریم. لذا دو ماتریس مشترک بر اساس هر دو سگمنت مرتبط از دو عکس محاسبه میکنیم. سپس فاصله بین دو پچ را در هر دو حالت به دست می آوریم. نهایتا میانگین فاصله های هر دو حالت را محاسبه و به عنوان فاصله نهایی دو سگمنت ارایه مدهیم. برای محاسبه فاصله کلی دو عکس نیز میانگین فواصل سگمنتها ارایه میشود. ($z=32 \times 32$ segment, $t=1,2$ (train and test)).

$$\mathbf{R} = \frac{1}{2p-1} \mathbf{z}_1^T \mathbf{z}_1 + \frac{1}{2p-1} \mathbf{z}_2^T \mathbf{z}_2$$

and

$$\mathbf{R}' = \frac{1}{2q-1} \mathbf{z}_1 \mathbf{z}_1^T + \frac{1}{2q-1} \mathbf{z}_2 \mathbf{z}_2^T$$

Figure 8: محاسبه دو حالت ماتریس مشترک دو بچ عکس

$$d(\mathbf{z}_1^T, \mathbf{z}_2^T) = \frac{4}{p} \sum_{j=1}^p (\lambda'_j - 0.5)^2$$

: فرمول 14 9 Figure

$$d(\mathbf{z}_1, \mathbf{z}_2) = \frac{4}{q} \sum_{j=1}^q (\lambda_j - 0.5)^2$$

: فرمول 15 10 Figure

$$diss(\mathbf{z}_1, \mathbf{z}_2) = \frac{d(\mathbf{z}_1, \mathbf{z}_2) + d(\mathbf{z}_1^T, \mathbf{z}_2^T)}{2}$$

Figure 11: محاسبه فاصله دو بچ

در ادامه الگوریتم کلی نیز آورده شده است:

Algorithm 1: DNNC

Input: The set of training images

$Y = \{Y_1, Y_2, \dots, Y_C\}$, $i = 1, 2, \dots, C$; a test image X ; the number of patches N ;

Output: The estimated label k for X ;

- 1: Partition training images into N non-overlapping patches and then generate training image patch sets $Y_i = \{y_{i1}, y_{i2}, \dots, y_{iN}\}$, $i = 1, \dots, C$.
 - 2: Partition the test image into N non-overlapping patches and then generate the test image patch set $X = \{x_1, x_2, \dots, x_N\}$;
 - 3: **for** $i = 1$ to C **do**
 - 4: **for** $j = 1$ to N **do**
 - 5: Calculate the difference $d(x_j, y_{ij})$ using (14);
 - 6: Calculate the difference $d(x_j^T, y_{ij}^T)$ using (15);
 - 7: Calculate the dissimilarity $diss(x_j, y_{ij})$ using (16);
 - 8: **end for**
 - 9: Calculate the dissimilarity $Diss(X, Y_i)$ according to the average rule (17) or the maximum rule (18) or the minimum rule (19);
 - 10: **end for**
 - 11: Assign the estimated label k to X using (20);
 - 12: **return** k ;
-

Figure 12: الگوریتم کلی مقاله

ارزیابی

با توجه به فرصت شدیداً محدود بنده، فرصت نشد تا انواع ارزیابی های داخل مقاله را اعمال کنم. ولی دو ارزیابی انجام شد:

مقایسه مدل مقاله با متریک اقلیدسی

مقایسه تعداد سگمنت $2*2$ با $4*4$

اما قبل ارایه نتایج لازم به ذکر است که مدل پیاده شده در هیچ شرایطی نتیجه مطلوب نداشته و میتوان گفت اصلاً کار نکرد. بنده خیلی تلاش کردم تا ایراد کار را در نحوه پیاده سازی متریک فاصله بررسی کنم ولی موفق نشدم. حدس بنده در این است که شاید دیتاست مشکل دارد اما متأسفانه در عکس های خدوم که متریک مورد

نظر نتوانست اصلا فاصله را عددی بین 0 و 1 دهد و همواره عددی بیش از یک است که خب خلاف حرف مقاله است. ولی پیاده سازی الگوریتم نزدیکترین همسایگی بر روی معیار اقلیدسی تانست به دقت 59 درصدی دست یابد که با توجه به 40 کلاسه بودن مساله میتوان این دقت را خوب دانست.

از آنجا که در اصل متریک مقاله مشکل وجود داشت لذا تعداد سگمنتها تفاوتی ایجاد نکرد. تمام نتایج در فایل پیاده سازی قابل مشاهده هستند.

دقت		دقت	
مدل ما (2*2)	2.50	مدل ما (4*4)	2.50
اقلیدسی	59.88		

$$100 * (numTestImages - numErrs) / numTestImages$$
 معیار محاسبه دقت:

تلاش های بنده برای پیدا کردن مشکل:

چیزی که متوجه شدم این بود که دایما تمام نمونه ها را به نفر اول یا ای دی صفر متعلق میدانست. با تغییر شماره شروع لوپ ترینها، یعنی مقایه هر تست فقط از نفر هشتم به بعد (به طور مثال) این حالت تعلق به ای دی صفر تغییر میکرد ولی باز هم دایما به نمونه ها ثابتی تعلق پیدا میکرد. نتیجه نهایی اینکه کمی دقت بهتر میشد ولی مثلا فقط میشد: 3.62 درصد!

نتیجه گیری:

در این مقاله سعی شد متریکی جدید برای محاسبه فاصله دو عکس برای الگوریتم نزدیکترین همسایگی ارایه شود که به دلیل خروجی نامناسب متریک پیاده شده، نتایج اصلا خوب نبود. اما الگوریتم نزدیکترین همسایگی برای مساله مورد نظر با متریک اقلیدسی خوب نتیجه داد.

کار آینده: سعی خواهد شد تا دیتاست اصلی تهیه و از طریق مکاتبه با نویسندگان مقاله ایراد کار به دست آید..