

به نام خدا



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

## پروژه پایانی تصویربرداری رقمی

بازنمایی چهره با استفاده از فیلتر گابور مبتنی بر کمی سازی فاز محلی

و الگوی باینری محلی

استاد: دکتر رحمتی

دانشجو: حلیمه رحیمی

شماره دانشجویی: ۹۹۱۳۱۰۴۳

بهار ۱۴۰۰



## چکیده

شناسایی چهره نقش مهمی را در بسیاری از کاربردها همچون رابط کامپیوتر-انسان، نظارت بصری، کنترل دسترسی و غیره دارد. گاهی مواقع تصاویر کم وضوح یا محو از چهره مانع از بازنمایی واقع‌گرا از چهره می‌شود. علاوه بر این تغییرات حالت و روشنایی، و انسداد جزئی مانع از تشخیص مناسب می‌گردد. در نتیجه لازم است روش‌های دسته‌بندی ارائه شود که در برابر چنین مشکلاتی مقاوم باشند. در این گزارش به بازنمایی چهره با استفاده از ترکیب فیلتر گابور، الگوی باینری محلی و کمی‌سازی فاز محلی می‌پردازیم. در فرآیند فیلتر گابور، از توابع موجک گابور با دو درجه و چهار جهت مختلف استفاده می‌کنیم تا ویژگی‌های بصری برجسته در تصویر بدست آید. روش‌های مختلف از جمله روش ارائه شده را مورد بررسی قرار می‌دهیم. روش پیشنهادی شامل اعمال LBP بر تصاویر حاصل از فیلترهایی گابور و اعمال LPQ بر تصاویر اصلی می‌باشد؛ به این ترتیب به طور کامل خاصیت‌های مقاوم در برابر محوشدگی و اطلاعات در دامنه‌ی مکانی را برای درجات و جهات مختلف می‌کاویم.

## کلمات کلیدی:

موجک‌های گابور، الگوی باینری محلی، کمی‌سازی فاز محلی، بازنمایی چهره، شناسایی چهره

## فهرست مطالب

۱- مقدمه .....	۱
۲- بازنمایی چهره .....	۱
۲-۱- موجک‌های گابور .....	۱
۲-۲- الگوی باینری محلی .....	۳
۲-۳- کمی‌سازی فاز محلی .....	۴
۲-۴- بازنمایی و توصیف چهره .....	۶
۳- پیاده سازی .....	۶
۴- آزمایشات .....	۱۱
۴-۱- آزمایش بر مجموعه داده Yale Face .....	۱۱
۴-۲- آزمایش بر مجموعه داده CMU-Multi-Pie .....	۱۵
۵- نتیجه گیری .....	۱۹
منابع .....	۲۰

## فهرست اشکال

- شکل ۱- فیلترهای گابور  $4 \times 4$  و  $16 \times 16$  ..... ۲
- شکل ۲- روند اعمال الگوی باینری محلی بر تصویر ..... ۴
- شکل ۳- روند اعمال کمی‌سازی فاز محلی بر تصویر ..... ۶
- شکل ۴- کتابخانه‌های بکاررفته ..... ۶
- شکل ۵- کد ساخت فیلترهای گابور ..... ۷
- شکل ۶- کد اعمال فیلترهای گابور ..... ۸
- شکل ۷- کد محاسبه برچسب الگوی باینری محلی برای یک پیکسل ..... ۸
- شکل ۸- کد محاسبه برچسب‌های الگوی باینری محلی برای یک تصویر ..... ۹
- شکل ۹- کد محاسبه برچسب‌های کمی‌سازی فاز محلی برای یک تصویر ..... ۱۰
- شکل ۱۰- کد دسته‌بندی یک-نزدیک‌ترین همسایه ..... ۱۱
- شکل ۱۱- نمونه‌های از مجموعه داده‌ی Yale Face ..... ۱۱
- شکل ۱۲- نمونه‌هایی از تصاویر حاصل از الگوی باینری محلی بر مجموعه داده Yale Face ..... ۱۳
- شکل ۱۳- نمونه‌هایی از تصاویر حاصل از کمی‌سازی فاز محلی بر مجموعه داده Yale Face ..... ۱۳
- شکل ۱۴- نمونه‌هایی از تصاویر حاصل از فیلترهای گابور برای دو تصویر از شخص اول ..... ۱۳
- شکل ۱۵- نمونه‌هایی از تصاویر حاصل از فیلترهای گابور برای دو تصویر از شخص دوم ..... ۱۴
- شکل ۱۶- فیلترهای گابور  $16 \times 16$  بکار رفته برای مجموعه داده Yale Face ..... ۱۴
- شکل ۱۷- نمونه‌هایی از مجموعه داده‌ی CMU-Multi-Pie ..... ۱۶
- شکل ۱۸- نمونه‌هایی از تصاویر حاصل از الگوی باینری محلی بر مجموعه داده CMU-Multi-Pie ..... ۱۶

شکل ۱۹- نمونه‌هایی از تصاویر حاصل از کمی‌سازی فاز محلی بر مجموعه داده CMU-Multi-Pie ..... ۱۶

شکل ۲۰- فیلترهای گابور ۸x۸ بکار رفته برای مجموعه داده CMU-Multi-Pie ..... ۱۷

## فهرست جداول

جدول ۱- مقایسه صحت روش‌های مختلف بر مجموعه داده‌ی Yale Face ..... ۱۲

جدول ۲- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Pie ..... ۱۵

جدول ۳- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با وجود حالت متفاوت

چهره ..... ۱۷

جدول ۴- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با وجود روشنایی

متفاوت برای دو حالت چهره ..... ۱۷

جدول ۵- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با انتخاب تصادفی

مجموعه آموزش و آزمون ..... ۱۸

## ۱- مقدمه

بازنمایی چهره علاوه بر کاربردهای مختلف، به دلیل پویایی و پیچیدگی صورت، معیاری برای ارزیابی الگوریتم‌های شناسایی الگو شده است.

الگوریتم‌های بسیاری برای بازنمایی چهره معرفی و آزموده شده‌اند. در این میان الگوی باینری محلی<sup>۱</sup> [1] برای حالات چهره [2] و شناسایی چهره [3]، و کمی‌سازی فاز محلی<sup>۲</sup> [4] نیز برای شناسایی چهره‌های محوشده بکار رفته است. LPQ براساس کمی‌سازی فاز انتقال فوریه در همسایگی محلی است. در تحلیل تصویر چهره، مشابه روش LBP، هیستوگرام‌های برچسب‌های LPQ در مناطق محلی بدست آمده به عنوان یک توصیف کننده‌ی چهره بکار می‌رود. از طرفی موجک‌های گابور<sup>۳</sup> از نقاط توجه برخی تحقیقات بوده است [5] [7]. موجک‌های گابور ساختار محلی مرتبط با فرکانس مکانی، موقعیت مکانی و جهت‌گیری انتخابی بخصوص را که نشان داده شده متمایز کننده بوده و در برابر روشنایی و تغییرات حالت، مقاوم‌اند، بدست می‌آورد. از این جهت، لی و همکاران [5] برای شناسایی و بازنمایی چهره از روشی با ترکیب فیلترهای گابور با چند درجه و چند جهت و تحلیل الگوی باینری محلی استفاده می‌کنند.

در اینجا ابتدا از موجک‌های گابور با جهات و درجات مختلف برای بدست آوردن ویژگی‌های چهره استفاده می‌شود و سپس LBP بر روی تصاویر حاصل اعمال می‌گردد. همچنین برچسب‌های LPQ را برای تصاویر اصلی بدست می‌آوریم. به این ترتیب اطلاعات بافت و اطلاعات محلی چهره را نه تنها در تصاویر اصلی بلکه در چهره‌های حاصل از فیلترهای گابور در جهات و درجات مختلف نیز کسب می‌کنیم. در نهایت بردار ویژگی‌ها، ترکیب هیستوگرام تصاویر حاصل شده‌ی نهایی خواهد بود.

(این گزارش مبتنی بر [8] می‌باشد.)

## ۲- بازنمایی چهره

### ۲-۱- موجک‌های گابور

هسته‌های گابور مشابه پروفایل‌های میدان دید در سلول‌های ساده‌ی چشم است که با ویژگی محلی بودن، انتخاب کننده‌ی جهت، و انتخاب کننده‌ی فرکانس شناخته می‌شوند. یک خانواده‌ی هسته‌ی گابور، حاصل ضرب یک پوش گوسی و موج ساده است.

---

<sup>1</sup> Local Binary Pattern

<sup>2</sup> Local Phase Quantization

<sup>3</sup> Gabor Wavelets

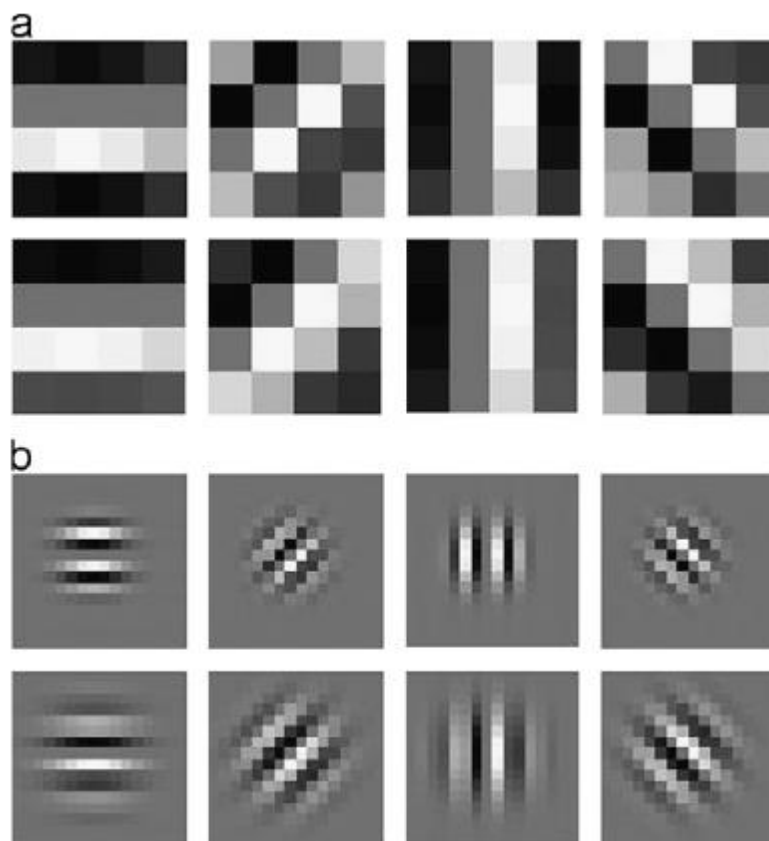
$$\psi_k(z) = \frac{k^2}{\sigma^2} \exp\left(-\frac{k^2}{2\sigma^2} z^2\right) \left(\exp(ikz) - \exp\left(-\frac{\sigma^2}{2}\right)\right) \quad \text{رابطه (۱)}$$

که در آن  $z = (x, y)$  متغیری در دامنه‌ی مکانی است و  $k$  بردار فرکانس است که تعیین کننده‌ی جهت  $(\mu)$  و درجات  $(\nu)$  هسته‌های گابور می‌باشد. در این مقاله،  $\mu$  و  $\nu$  ضریب جهت و درجه است. انتخاب متفاوتی از این دو متغیر هسته‌های گابور متفاوتی می‌دهد. در اینجا مقادیر  $\mu = 0, 1, 2, 3$  و  $\nu = 0, 1$  را انتخاب می‌کنیم. بنابراین در کل هشت هسته‌ی گابور خواهیم داشت.

برای هر تصویر، تبدیل گابور آن در یک محل خاص از کانولوشن با هسته‌های گابور بدست می‌آید:

$$G_k(x, y) = I(x, y) * \psi_k(x, y) \quad \text{رابطه (۲)}$$

فیلترهای بکاررفته را در شکل (۱) می‌بینید.



شکل ۲۱- فیلترهای گابور ۴x۴ و ۱۶x۱۶ (تصویر از [8])



## ۲-۲- الگوی باینری محلی

الگوی باینری محلی [1] روشی قدرتمند برای توصیف بافت براساس تحلیل آماری است. با آنکه انواع LBP بدلیل کارآمدی قابل قبول در دسته‌بندی برای تحلیل چهره بکار می‌روند، هنوز مناسب و کافی بودن آن اثبات نشده است. بنابراین ترکیب ویژگی روشی مفید بنظر می‌رسد. LBP به همراه بیشینه کردن اطلاعات مشترک عملکرد بهتری را در تحلیل چهره نمایش می‌دهد. عملگر پیکسل‌های تصویر را در یک همسایگی هر پیکسل بررسی می‌کند و اگر مقدار آن بیشتر از مقدار مرکز باشد، آن خانه را یک و در غیر این صورت صفر در نظر می‌گیرد. سپس مقادیر صفر و یک را در جهت عقربه‌های ساعت کنار یکدیگر گذاشته و مقدار دهمی آن را محاسبه می‌کند.

$$LBP_{P,R}(x_c, y_c) = \sum_{n=0}^{P-1} \delta(g_n - g_c) 2^n \quad \text{رابطه (۳)}$$

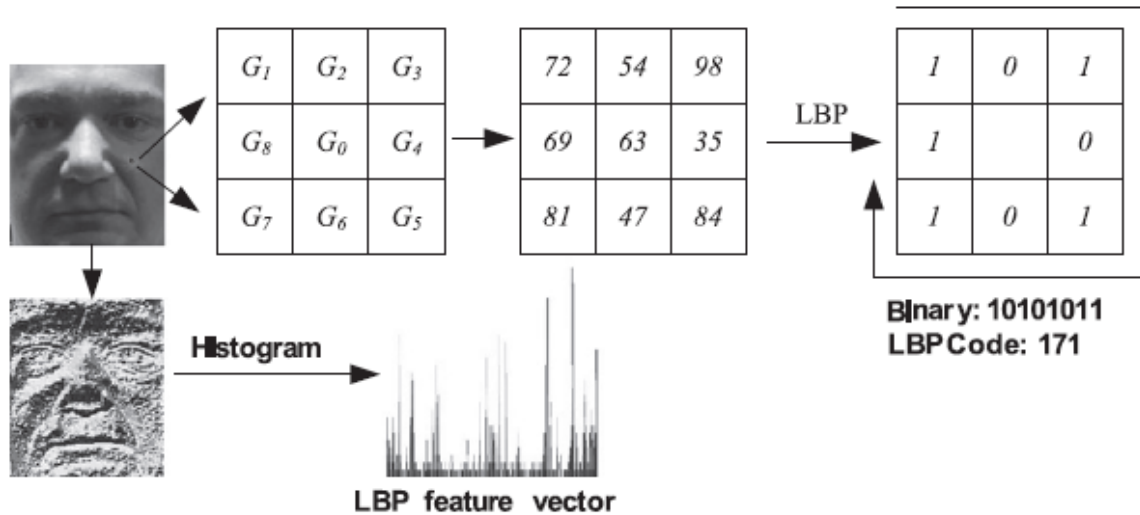
در این رابطه  $g_c$  مقدار پیکسل مرکزی است که در  $(x_c, y_c)$  قرار گرفته و  $g_n$  مقدار یکی از خانه‌های همسایه در شعاع همسایگی  $R$  می‌باشد.  $P$  نیز تعداد کل همسایگی‌ها بوده (در اینجا  $P=8$  در نظر می‌گیریم) و  $\delta(\cdot)$  تابع علامت است که به شکل زیر تعریف می‌شود.

$$\delta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{رابطه (۴)}$$

هیستوگرام برچسب‌های LBP می‌تواند به عنوان توصیف کننده‌ی ویژگی استفاده شود.

LBP با  $P$  همسایگی، دارای  $2^P$  الگوی باینری متفاوت می‌باشد. با این حال [1] مشاهده کرد تصاویر طبیعی شامل تعداد کدهای LBP کمتری است که با عنوان LBP یکنواخت شناخته می‌شوند. LBP یکنواخت شامل بیشینه‌ی دو انتقال باینری از صفر به یک یا برعکس است در حالیکه الگوی بیت دورانی در نظر گرفته شده است. این الگوهای یکنواخت ارائه دهنده‌ی بسیاری از ریزساختارهای بافت می‌باشد.

شکل (۲) روند اعمال LBP را به نمایش می‌کشد.



شکل ۲۲- روند اعمال الگوی باینری محلی بر تصویر (تصویر از [8])

## ۲-۳- کمی سازی فاز محلی

اوجانسیوو و همکاران [4] برای محو کردن تصویر، از اعمال کانولوشن بین تصویر و یک تابع گسترش نقطه استفاده کرد. در صورتی که  $f(x)$  را تصویر اصلی که در آن  $x = [x_0, x_1]^T$  است و  $h(x)$  را PSF بدانیم، تصویر حاصل  $g(x)$  خواهد بود.

$$g(x) = f(x) \otimes h(x) \quad \text{رابطه (۵)}$$

این رابطه در دامنه ی فوریه، به شکل  $G(u) = F(u) \cdot H(u)$  در می آید که می توان بزرگی و فاز آن را به صورت زیر بدست آورد:

$$|G(u)| = |F(u)| \cdot |H(u)|, \quad \angle G(u) = \angle F(u) + \angle H(u) \quad \text{رابطه (۶)}$$

اگر فرض کنیم تابع محوکننده ی  $h(x)$  نسبت به مرکز متقارن بوده، انتقال فوریه ی آن همواره مقداری حقیقی خواهد داشت و بنابراین فاز آن تنها دو مقدار خواهد گرفت که به شکل تابع زیر بیان می گردد:

$$\angle H(u) = \begin{cases} 0 & \text{if } H(u) \geq 0 \\ \pi & \text{if } H(u) < 0 \end{cases} \quad \text{رابطه (۷)}$$

چنین چیزی این معنی را به همراه خواهد داشت که:

$$\angle G(u) = \angle F(u) \quad \text{for all } H(u) \geq 0 \quad \text{رابطه (۸)}$$

به عبارتی فاز تصویر  $\angle G(u)$  در فرکانس‌هایی که  $H(u)$  مثبت است، نسبت به محوشدگی متقارن پایدار است.

در LPQ فاز در همسایگی‌های محلی  $\mathcal{N}_x$  هر پیکسل  $x$  بررسی می‌شود. این طیف محلی از طریق تبدیل فوریه کوتاه مدت محاسبه می‌گردد:

$$F(u, x) = \sum_{y \in \mathcal{N}_x} f(x - y) e^{-j2\pi u^T y} \quad \text{رابطه (۹)}$$

ضرایب فوریه محلی در چهار نقطه‌ی فرکانسی محاسبه می‌شوند:  $u_3 = u_2 = [0, a]^T$ ,  $u_1 = [a, 0]^T$  و  $u_4 = [a, -a]^T$  که در آنها  $a$  مقدار یست به اندازه‌ی کافی کوچک تا شرط  $H(u_i) > 0$  را ارضا کند. برای هر پیکسل  $x$  بردار زیر حاصل می‌شود:

$$F(x) = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)] \quad \text{رابطه (۱۰)}$$

با بررسی علامت بخش حقیقی و مجازی مولفه‌های  $F(x)$ ، اطلاعات فاز را می‌توان بر اساس تابع زیر بدست آورد:

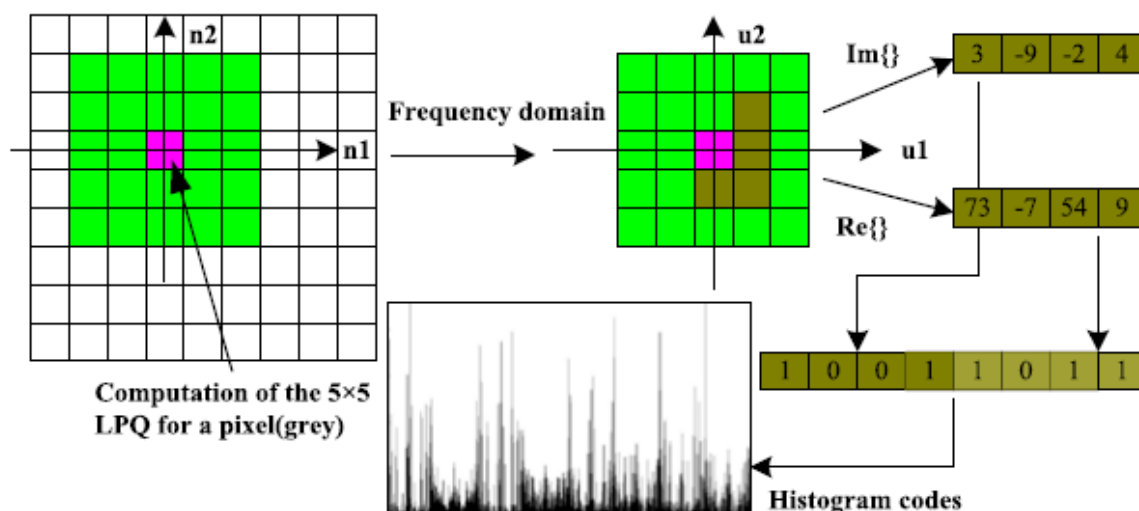
$$q_i = \begin{cases} 1 & \text{if } g_j \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad \text{رابطه (۱۱)}$$

که در آن  $g_j$  مولفه‌ی  $j$ ام از بردار  $G(x) = [Re\{F(x)\}, Im\{F(x)\}]$  است.

سپس تصویر برچسب‌های LPQ به شکل زیر خواهد بود:

$$f_{LPQ}(x) = \sum_{j=1}^8 q_j(x) 2^{j-1} \quad \text{رابطه (۱۲)}$$

شکل (۳) روند بدست آوردن برچسب‌های LPQ را به تصویر می‌کشد.



شکل ۲۳- روند اعمال کمی‌سازی فاز محلی بر تصویر (تصویر از [8])

## ۲-۴- بازنمایی و توصیف چهره

در ابتدا هشت تصویر تبدیل یافته توسط فیلترهای گابور را بدست می‌آوریم و سپس LPQ را بر روی تصاویر اصلی و LBP را بر تصاویر فیلتر شده اعمال می‌کنیم. پس از آن مقادیر هیستوگرام تصاویر حاصل شده را در یک بردار ویژگی قرار می‌دهیم. در نهایت از این بردار برای دسته‌بندی به وسیله‌ی دسته‌بند یک-نزدیکترین همسایه بهره می‌بریم. فاصله اقلیدسی به عنوان معیار فاصله برای دسته‌بند انتخاب شده است.

## ۳- پیاده‌سازی

برای پیاده‌سازی از کتابخانه‌های زیر استفاده شده است:

```
import numpy as np
import scipy.io
from scipy.signal import convolve2d
import cv2
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
```

شکل ۲۴- کتابخانه‌های بکاررفته

تابع `build_filters` هشت فیلتر ذکر شده در بخش (۲-۱) را با استفاده از فراخوانی تابع `gabor_wavelet` ایجاد می‌کند. این هشت فیلتر را در اندازه‌ی متناسب با مجموعه داده ایجاد کردم که در بخش بعدی شکل (۱۵) و (۱۹) مشخص است.

ویژگی‌های حاصل از فیلترهای گابور از طریق تابع `extract_features` بدست می‌آید.

```
def gabor_wavelet(rows, cols, kmax, f, orientation, scale, delt2):

    k = (kmax / (f ** scale)) * np.exp(1j * orientation * np.pi / 4)
    kn2 = np.abs(k) ** 2

    gw = np.zeros((rows, cols), np.complex128)

    for m in range(int(-rows/2) + 1, int(rows / 2) + 1):
        for n in range(int(-cols/2) + 1, int(cols / 2) + 1):
            t1 = np.exp(-0.5 * kn2 * (m**2 + n**2) / delt2)
            t2 = np.exp(1j * (np.real(k) * m + np.imag(k) * n))
            t3 = np.exp(-0.5 * delt2)
            gw[int(m + rows/2 - 1), int(n + cols/2 - 1)] = (kn2 / delt2) * t1 * (t2 - t3)

    return gw

def build_filters(num_or, num_sc, rows=4, cols=4, plot=False):

    # num_or as number of orientations
    # num_sc as number of scales

    filters = []
    kmax = np.pi / 2
    f = np.sqrt(2)
    delt2 = (np.pi) ** 2

    for v in range(0, num_sc):
        for u in range(0, num_or):
            gw = gabor_wavelet(rows, cols, kmax, f, u, v, delt2)
            filters.append(gw)

    if plot:
        fig = plt.figure(figsize=(10,5))
        for v in range(0, num_sc):
            for u in range(0, num_or):
                fig.add_subplot(2, 4, 4*(v) + (u+1), xticks=[], yticks=[])
                plt.imshow(np.real(filters[(v*num_or)+u]), cmap='gray')

    plt.show()

    return filters
```

شکل ۲۵- کد ساخت فیلترهای گابور

```
def extract_features(img, filters, plot=False):

    results = []

    for i in range(8):
        resultR = cv2.filter2D(img, cv2.CV_32F, np.real(filters[i]))
        resultI = cv2.filter2D(img, cv2.CV_32F, np.imag(filters[i]))
        result = np.hypot(resultR, resultI)
        results.append(result)

    if plot:
        fig = plt.figure(figsize=(10,5))
        for i in range(8):
            fig.add_subplot(2, 4, (i+1), xticks=[], yticks=[])
            plt.imshow(np.real(results[i]), cmap='gray')
        plt.show()

    return results
```

شکل ۲۶- کد اعمال فیلترهای گابور

سپس برای بخش محاسبه برجسب‌های الگوهای باینری محلی از تابع `lbp` استفاده می‌شود که برجسب‌ها را از طریق تابع `lbp_pixel` برای هر پیکسل بدست می‌آورد.

```
def get_pixel(img, center, x, y):

    new_value = 0
    try:
        if img[x,y] >= center:
            new_value = 1
    except:
        pass

    return new_value

def lbp_pixel(img, x, y):

    center = img[x,y]
    # Neighbors in a clockwise direction
    xplace = [-1,-1,-1,0,1,1,1,0]
    yplace = [-1,0,1,1,1,0,-1,-1]

    val = 0
    # Compute LBP codewords
    for k in range(len(xplace)):
        val += (get_pixel(img, center, x+xplace[k], y+yplace[k]))*(2**k)

    return val
```

شکل ۲۷- کد محاسبه برجسب الگوی باینری محلی برای یک پیکسل

این تابع از هیستوگرام تصویر حاصل استفاده می‌کند و پس از تبدیل آن به بردار، نتیجه‌ی نرمال‌سازی شده را ارائه می‌دهد.

```
def lbp(img, width, height, normalize=True, plot=False):  
  
    LBPdesc = np.zeros((width, height), np.uint8)  
  
    for i in range(0, width):  
        for j in range(0, height):  
            LBPdesc[i, j] = lbp_pixel(img, i, j)  
    if plot:  
        plt.imshow(LBPdesc, cmap="gray")  
        plt.xticks([])  
        plt.yticks([])  
        plt.show()  
  
    LBPdesc = np.histogram(LBPdesc.ravel(), range(256)) [0]  
  
    if normalize:  
        LBPdesc=LBPdesc/LBPdesc.sum()  
  
    return LBPdesc
```

شکل ۲۸- کد محاسبه برجسب‌های الگوی باینری محلی برای یک تصویر

تصاویر برای کمی‌سازی فاز محلی نیز از طریق تابع LPQ بدست می‌آید.

```

def lpq(img, winSize=3, normalize=True, plot=False):

    STFTalpha=1/winSize

    convmode='same'

    img=np.float32(img)
    r=(winSize-1)/2 # Get radius from window size
    x=np.arange(-r,r+1)[np.newaxis] # Form spatial coordinates in window

    # Gaussian STFT filters
    w0=np.ones_like(x)
    w1=np.exp(-2*np.pi*x*STFTalpha*1j)
    w2=np.conj(w1)

    sigmaS=(winSize-1)/4
    gs=np.exp(-0.5*(x/sigmaS)**2)/(np.sqrt(2*np.pi)*sigmaS);

    w0=gs*w0;
    w1=gs*w1;
    w2=gs*w2;

    w1=w1-np.mean(w1);
    w2=w2-np.mean(w2);
    # Run filters to compute the frequency response in the four points
    filterResp1=convolve2d(convolve2d(img,w1.T,convmode),w0,convmode)
    filterResp2=convolve2d(convolve2d(img,w0.T,convmode),w1,convmode)
    filterResp3=convolve2d(convolve2d(img,w1.T,convmode),w1,convmode)
    filterResp4=convolve2d(convolve2d(img,w1.T,convmode),w2,convmode)

    # Initilize frequency domain matrix for four frequency coordinates
    freqResp=np.dstack([filterResp1.real, filterResp2.real, filterResp3.real, filterResp4.real,
                        filterResp1.imag, filterResp2.imag, filterResp3.imag, filterResp4.imag])

    # Perform quantization and compute LPQ codewords
    inds = np.arange(freqResp.shape[2])[np.newaxis,np.newaxis,: ]
    LPQdesc=((freqResp>0)*(2**inds)).sum(2)

    if plot:
        plt.imshow(np.uint8(LPQdesc),cmap='gray')
        plt.xticks([])
        plt.yticks([])
        plt.show()

    # Histogram
    LPQdesc=np.histogram(LPQdesc.flatten(),range(256))[0]

    if normalize:
        LPQdesc=LPQdesc/LPQdesc.sum()

    return LPQdesc

```

شکل ۲۹- کد محاسبه برجسب‌های کمی‌سازی فاز محلی برای یک تصویر

دسته‌بند بکار رفته نیز همچنان یک-نزدیک‌ترین همسایه می‌باشد.



```
def classify(train_data, test_data, ytrain, ytest):

    classifier = KNeighborsClassifier(n_neighbors=1, metric='euclidean')
    classifier.fit(train_data, np.squeeze(ytrain))
    ypred = classifier.predict(test_data)

    confmat = confusion_matrix(ytest, ypred, range(1,16))
    print('Confusion Matrix:\n', confmat)
    acc = np.sum(np.diag(confmat)) / np.sum(confmat) * 100
    print('Accuracy: ', acc)
```

شکل ۳۰- کد دسته‌بندی یک-نزدیک‌ترین همسایه

## ۴- آزمایشات

نتایج حاصل به دلیل کاملاً یکسان نبودن مجموعه داده‌ها و ذکر نشدن جزئیات پیاده‌سازی از نتایج مقاله متفاوت خواهد بود. در تمامی بخش‌ها دلیل آن را بیان خواهیم کرد.

### ۴-۱- آزمایش بر مجموعه داده Yale Face

مجموعه داده‌ی Yale Face شامل ۱۵ شخص هر یک با ۱۱ تصویر مختلف می‌باشد. این مجموعه داده به صورت کراپ شده در مقاله بکار رفته است اما از آن جهت که این نسخه از آن موجود نبود، سعی داشتم با کراپ تصاویر به نتایج یکسان برسم که حاصل نشد. در نهایت تصمیم گرفتم این مجموعه داده را به صورت کراپ شده و در ابعاد ۶۴x۶۴ از صفحه‌ی Deng Cai<sup>۴</sup> دریافت کنم.

مطابق با مقاله، تصاویر ۱ تا ۶ هر شخص را در مجموعه آموزش و ۷ تا ۱۱ را در مجموعه آزمون قرار دادم.

برخی از تصاویر در این مجموعه داده را در شکل (۱۰) می‌بینید.



شکل ۳۱- نمونه‌های از مجموعه داده‌ی Yale Face

<sup>۴</sup> <http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html>

در اینجا ابتدا از مقادیر شدت خود پیکسل‌های تصاویر برای دسته‌بندی استفاده کردم، سپس به ترتیب برچسب‌های LBP و LPQ بر روی خود تصاویر، LBP و LPQ بر روی تصاویر حاصل از گابور (Gabor+LBP) و (Gabor+LPQ)، ترکیب LPQ و Gabor+LBP همانطور که در مقاله بیان شده، و ترکیب Gabor+LBP و Gabor+LPQ را برای دسته‌بندی بکار گرفتم.

جدول (۱) نتایج را برای مجموعه داده‌ی Yale Face حاصل شده توسط مولفین و نتایج حاصل از پیاده‌سازی اینجانب را نشان می‌دهد.

جدول ۶- مقایسه صحت روش‌های مختلف بر مجموعه داده‌ی Yale Face. (a) نتایج بدست آمده در مقاله (جدول از [8])، (b) نتایج حاصل از پیاده‌سازی مجدد.

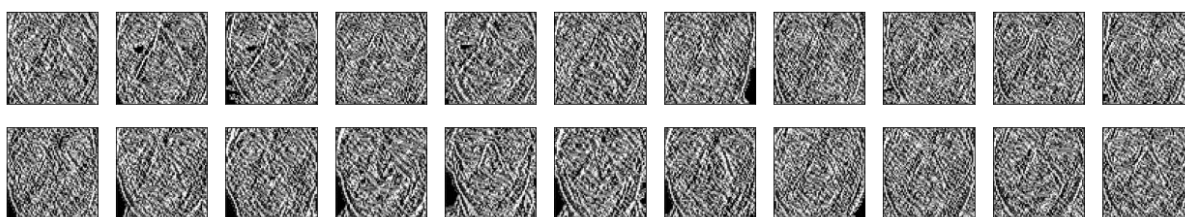
a)		b)	
Method	Accuracy	Method	Accuracy
Image Intensities	---	Image Intensities	77.3
LBP	72	LBP	61.3
LPQ	81.3	LPQ	69.3
Gabor + LBP	74.7	Gabor + LBP	80.0
Gabor + LPQ	88.0	Gabor + LPQ	88.0
(Gabor + LBP) + LPQ	90.7	(Gabor + LBP) + LPQ	82.0
(Gabor + LBP) + (Gabor + LPQ)	---	(Gabor + LBP) + (Gabor + LPQ)	90.7

به دلیل رزولوشن پایین تصاویر نتایج دسته‌بندی با برچسب‌های LBP و LPQ بر روی خود تصاویر نه تنها نتیجه را بهتر نمی‌کند بلکه بدتر نیز می‌کند. در واقع با اعمال LBP قادر هستیم بافت چهره را کد کنیم. با توجه به رزولوشن پایین جزئیات خاصی (تفاوت مقادیر در هیستوگرام) از بافت چهره در تصویر به چشم نمی‌خورد و بیشتر، کلیات هستند که قابل مشاهده‌اند. همچنین این مسئله که شخص جلوتر/عقب‌تر ایستاده است، ممکن است بر مقادیر هیستوگرام تصاویر کراپ شده تاثیر بگذارد؛ چرا که در تصاویر کامل، لباس‌ها و موها تقریباً به یک اندازه دیده می‌شدند، علاوه بر این LBP را بر روی تصاویر کامل نیز اعمال کرده و دسته‌بند را آزموده بودم و نتیجه بدتر از نتیجه حاصل از شدت روشنایی پیکسل‌ها نشده بود.



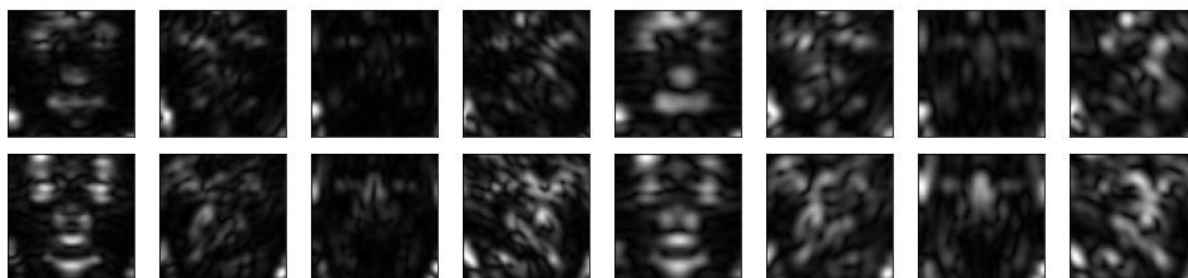
شکل ۳۲- نمونه‌هایی از تصاویر حاصل از الگوی باینری محلی بر مجموعه داده Yale Face

مقاومت LPQ در برابر محوشدگی برای حالتی که رزولوشن عکس پایین است به کار می‌آید؛ چرا که در چنین حالتی هم به نوعی محوشدگی حس می‌شود. همچنین در برابر تغییرات بسیار کوچک ایجاد شده در چهره‌ی شخص مقاوم خواهد بود.



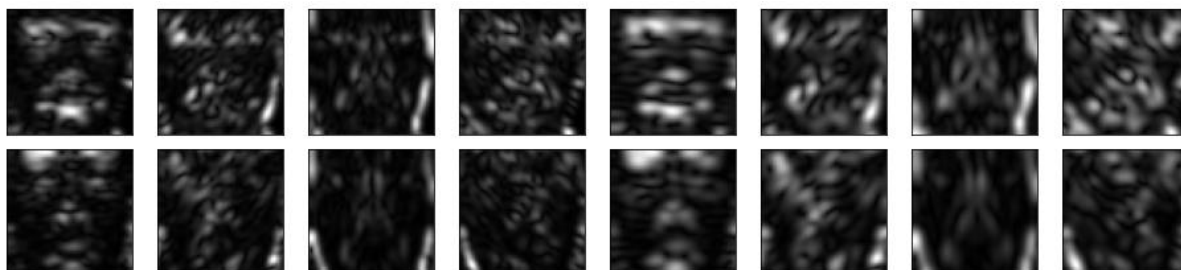
شکل ۳۳- نمونه‌هایی از تصاویر حاصل از کمی‌سازی فاز محلی بر مجموعه داده Yale Face

بهبود عملکرد دسته‌بندها با فیلترهای گابور را می‌توان با دیدن تصاویر حاصل از آنها انتظار داشت. در شکل (۱۳) به ترتیب هشت تصویر بالایی حاصل فیلترها بر روی یک تصویر و هشت تصویر پایینی بر تصویر دیگر از یک شخص اند. شکل (۱۴) نیز تصاویر حاصل را برای دو تصویر از یک شخص دیگر نمایش می‌دهد.

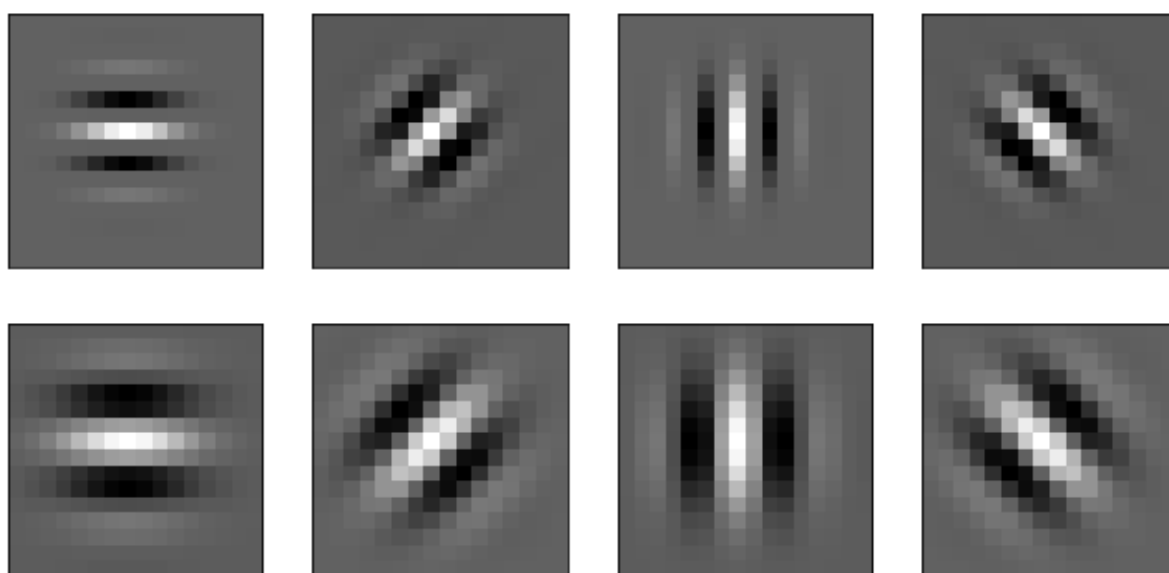


شکل ۳۴- نمونه‌هایی از تصاویر حاصل از فیلترهای گابور برای دو تصویر از شخص اول. (ردیف بالا یک تصویر و ردیف پایین یک تصویر دیگر)

تفاوت میان افراد با وجود فیلترها و حالات یکسان چهره همچنان روشن است و در عین حال شباهت تصاویر چهره‌ی هر فرد از بین نرفته است.



شکل ۳۵- نمونه‌هایی از تصاویر حاصل از فیلترهای گابور برای دو تصویر از شخص دوم. (ردیف بالا یک تصویر و ردیف پایین یک تصویر دیگر)



شکل ۳۶- فیلترهای گابور  $16 \times 16$  بکار رفته برای مجموعه داده Yale Face

علت انتخاب فیلترهای  $16 \times 16$  تغییرات محدود به دلیل رزولوشن پایین تصاویر می‌باشد. تصاویر حاصل از فیلتر  $4 \times 4$  تفاوت بسیاری با تصویر اصلی نخواهند داشت. از آن جهت که هر چه تصویر مشابه به تصویر اصلی باشد، LBP بیشتر مثل قبل عمل می‌کند، بنابراین می‌توان انتظار داشت که با فیلترهای بزرگتر گابور (البته نه آنچنان که نصف تصویر را برای محاسبه‌ی برچسب یک نقطه بکار گیرند)، نتایج از حالت قبلی متفاوت باشد و به علت وجود تفاوت در چندین تصویر که هیستوگرام‌هایشان با هم پیوند می‌خورند (افزایش تعداد ویژگی‌های متفاوت از هم) احتمال می‌رود دسته‌بند بهتر از قبل پیشبینی کند که چنین اتفاقی نیز می‌افتد.

نکته‌ی دیگر نیز در انتخاب اندازه‌ی فیلتر، بزرگی الگوهای تصویر می‌باشد. هر چه به نسبت تصویر، این الگوها بزرگتر، نیازمند فیلتر بزرگتر برای آنها هستیم.

فیلتر  $8 \times 8$  به دلیل محوشدگی که حاصل می‌دهد (در عین حفظ ساختار تصویر اصلی) عملکرد LPQ را بهتر می‌کند اما فیلتر  $16 \times 16$  با آنکه عملکرد آن را کاهش می‌دهد، عملکرد LBP را بهبود می‌بخشد.

تا کنون باید علت آنکه علاوه بر روش ارائه شده در مقاله از ترکیب دیگری نیز استفاده کردم روشن شده باشد. ویژگی‌های LPQ مبتنی بر تصویر اصلی به دلیل رزولوشن پایین آنچنان مناسب نبوده‌اند درحالی‌که ویژگی‌های LPQ مبتنی بر چهره‌های گابور عملکرد بهتری نسبت به آن داشته‌اند.

#### ۴-۲- آزمایش بر مجموعه داده CMU-Multi-Pie

متأسفانه مجموعه داده CMU-PIE در دسترس نبود. نتایج مربوط به آن را در جدول (۲) گرفته شده از مقاله قابل مشاهده است. به جای آن ۴۰ تصویر ۵۱ درجه از ۲۵ شخص را از مجموعه داده‌ی CMU-Multi-Pie که با حالت خنثی و خندان به جلو نگاه می‌کنند، بکار بردم. این مجموعه داده را به شکل کراپ شده  $128 \times 128$  از صفحه‌ی گیت‌هاب [9]<sup>۵</sup> دریافت کرده‌ام. نتایج حاصل در اینجا نیز قابل بحث است.

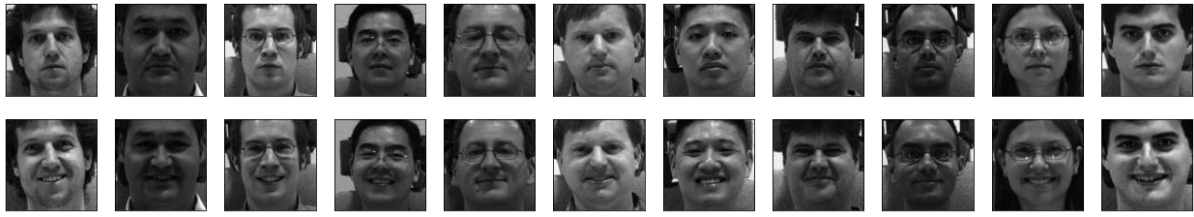
جدول ۷- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Pie (جدول از [8])

Method	Accuracy
Image Intensities	---
LBP	47.5
LPQ	57.7
Gabor + LBP	57.4
Gabor + LPQ	61.3
(Gabor + LBP) + LPQ	71.9
(Gabor + LBP) + (Gabor + LPQ)	---

چند نمونه از تصاویر این مجموعه داده را در شکل (۱۶) می‌بینید.

<sup>5</sup> <https://github.com/bluer555/CR-GAN/blob/master/README.md>





شکل ۳۷- نمونه‌هایی از مجموعه داده‌ی CMU-Multi-Pie

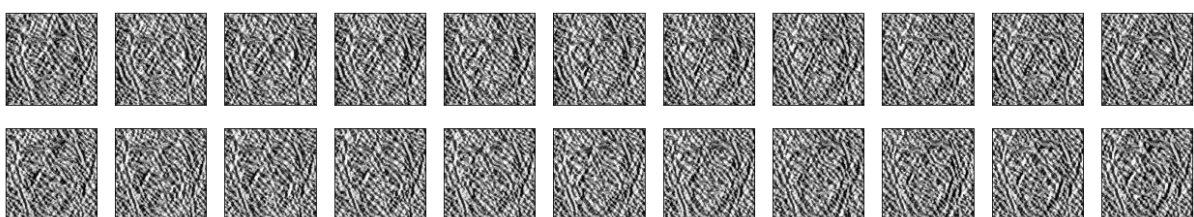
نکته قابل توجه‌ای در انتخاب داده‌ها وجود داشت؛ در صورتی که تصاویر با یک حالت خاص را هم در آموزش و آزمون داشته باشیم، به علت شباهت تصاویر یک شخص، عملکرد شدت روشنایی پیکسل‌ها نیز بسیار بالا خواهد بود (برای ۵۰ شخص با تنها یک حالت چهره‌ی خنثی، ۹۹٫۵٪، درحالی‌که عملکرد LBP به دلیل تفاوت روشنایی بسیار پایین خواهد آمد (۵۹٪)، هیستوگرام LBP معیار مناسبی برای تمایز نخواهد بود). Gabor+LPQ حتی در این شرایط هم عملکرد عالی داشت (۱۰۰٪).

در ابتدا از ۴۰ تصویر، بیست تصویر حالت خنثی را برای آموزش و بیست تصویر حالت خندان را برای آزمون در نظر گرفتیم. نتایج را در این حالت در جدول (۳) می‌بینید.

LBP و LPQ دیگر مشکل قبل را بر روی تصاویر ندارند و بنابراین نتایج نسبت به شدت روشنایی پیکسل‌های تصویر بهبود یافته است.



شکل ۳۸- نمونه‌هایی از تصاویر حاصل از الگوی باینری محلی بر مجموعه داده CMU-Multi-Pie



شکل ۳۹- نمونه‌هایی از تصاویر حاصل از کمی‌سازی فاز محلی بر مجموعه داده CMU-Multi-Pie

جدول ۸- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با وجود حالت متفاوت چهره

Method	Accuracy
Image Intensities	76.4
LBP	79.8
LPQ	95.6
Gabor + LBP	92.2
Gabor + LPQ	98.8
(Gabor + LBP) + LPQ	95.8
(Gabor + LBP) + (Gabor + LPQ)	98.6

از آن جهت که ممکن است LBP تا حدودی از یکسان بودن روشنایی برخی تصاویر بهره برده باشد، این حالت را هم آزمودم که ده تصویر اول از حالت خنثی، و ده تصویر آخر از حالت خندان هر شخص را در آموزش و باقی را در آزمون قرار دادم. در صورتی که LBP در آزمایش کنونی از طریق شباهت روشنایی به نتیجه‌ی بهتری می‌رسید، باید تفاوت روشنایی موجب تشخیص‌های اشتباه می‌شد درحالی‌که نتیجه‌ی بهتری پیدا کرد (۸۴/۶٪)؛ به این معنی که حالت این افراد هم در تشخیص صحیح با استفاده از LBP تاثیر گذار است. از طرفی چون آموزش، حالت خندان فرد را در خود داشت، موجب عملکرد بهتر شدت روشنایی پیکسل‌ها گردیده است. نتایج را برای این حالت از داده‌ها در جدول (۴) مشاهده می‌کنید.

جدول ۹- مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با وجود روشنایی متفاوت برای دو حالت چهره

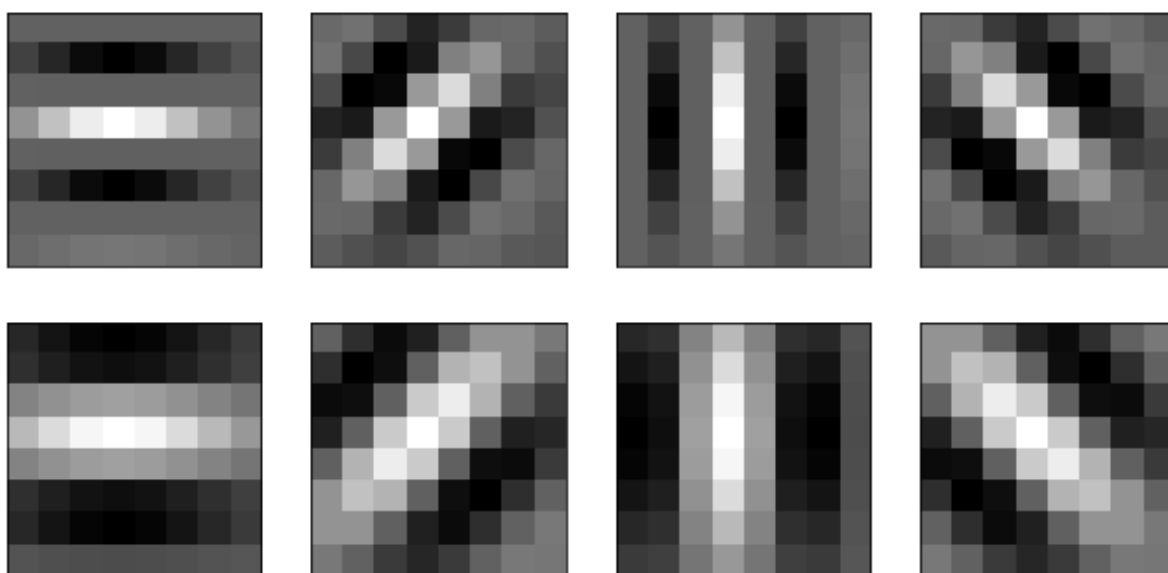
Method	Accuracy
Image Intensities	95.2
LBP	84.6
LPQ	98.6
Gabor + LBP	96
Gabor + LPQ	99.6
(Gabor + LBP) + LPQ	97.8
(Gabor + LBP) + (Gabor + LPQ)	99

اگر داده‌ها را به طور تصادفی انتخاب کنیم، نتایج جدول (۵) حاصل می‌گردد.

جدول ۱۰ - مقایسه صحت برای روش‌های مختلف بر مجموعه داده CMU-Multi-Pie با انتخاب تصادفی مجموعه آموزش و آزمون

Method	Accuracy
Image Intensities	91.36
LBP	81.8
LPQ	94.12
Gabor + LBP	93.2
Gabor + LPQ	98.71
(Gabor + LBP) + LPQ	94.18
(Gabor + LBP) + (Gabor + LPQ)	95.4

برای هر سه از پارامترهای یکسان و فیلترهای گابور یکسان ۸×۸ استفاده شده است.



شکل ۴۰ - فیلترهای گابور ۸×۸ بکار رفته برای مجموعه داده CMU-Multi-Pie

طبق نتایج بدست آمده بنظر می‌رسد تاثیر Gabor+LBP بیشتر از Gabor+LPQ بسته به داده باشد و نتیجه‌ی روش ارائه شده را تحت تاثیر قرار می‌دهد.



## ۵- نتیجه گیری

در این گزارش به بررسی بازنمایی چهره بر اساس فیلترهای گابور، الگوی باینری محلی و کمی سازی فاز محلی پرداختیم. مشاهده شد که روش ارائه شده بهبود عملکرد مناسبی را در رابطه با شناسایی چهره دارد. علاوه بر این عملکرد روش های مختلفی را در برابر داده های متفاوت مورد بررسی قرار دادیم و نکاتی را برای رسیدن به عملکرد بهتر حتی با وجود تصاویر با کیفیت پایین ارائه کردیم.

- [1] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.
- [2] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, 2007.
- [3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [4] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *International conference on image and signal processing*, 2008, pp. 236–243.
- [5] Z. Lei, S. Liao, M. Pietikäinen, and S. Z. Li, "Face recognition by exploring information jointly in space, scale and orientation," *IEEE Trans. image Process.*, vol. 20, no. 1, pp. 247–256, 2010.
- [6] W. Jiang, K.-M. Lam, and T.-Z. Shen, "Efficient edge detection using simplified Gabor wavelets," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 4, pp. 1036–1047, 2009.
- [7] H. Zhou and A. H. Sadka, "Combining perceptual features with diffusion distance for face recognition," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 41, no. 5, pp. 577–588, 2010.
- [8] S.-R. Zhou, J.-P. Yin, and J.-M. Zhang, "Local binary pattern (LBP) and local phase quantization (LBQ) based on Gabor filter for face representation," *Neurocomputing*, vol. 116, pp. 260–264, 2013.
- [9] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas, "CR-GAN: learning complete representations for multi-view generation," *arXiv Prepr. arXiv1806.11191*, 2018.