

تشخیص حالت صورت با استفاده از هندسه‌ی نقاط کلیدی چهره

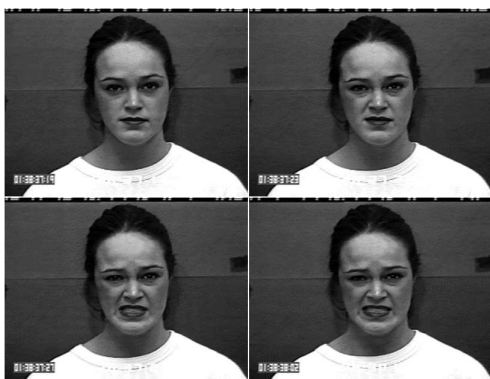
پرnian کسرائی

دانشکده برق - دانشگاه صنعتی شریف

تهران، ایران

kasraei_parnian@ee.sharif.edu

مسئله با استفاده از ویژگی‌های هندسی لندمارک‌های صورت و متد کلسیفیکیشن حل شده‌است. برای این آموزش دادن کلسیفایر از دیتاست کوهن-کنده^[1] استفاده شده است که حاوی بیش از 300 دنباله‌ی احساسات برچسب گذاری شده- است. یک نمونه را در شکل 1 مشاهده می‌کنید.



شکل 1 - نمونه دنباله‌ی دیتاست

چکیده: از کاربردهای بینایی کامپیوتری تشخیص چهره و نقاط کلیدی آن است. در این گزارش تلاش می‌شود که با داشتن ابزارهای تشخیص صورت، متدی برای تشخیص حالت چهره طراحی کنیم که بتواند به تصویری از فرد یکی از 7 احساس اصلی را نسبت بدهد. این الگوریتم می‌تواند بلوکی باشد در سیستم دستیارهای کامپیوتری (computer assistants) تا کامپیوتر بسته به احساس فرد پاسخ‌هایش را تنظیم کرده و ارتباط کاربر-کامپیوتر به روابط انسانی نزدیک‌تر شود. همچنین، پارامترهای یک خانه یا محیط هوشمند را با دانستن احساس کاربر می‌توان تنظیم کرد. برای چنین کاربردهایی سرعت عمل الگوریتم و دقت بالای آن برای تشخیص یک چهره-ی ثابت اهمیت زیادی دارد. در این متد تلاش کرده‌ایم به این مسائل توجه کنیم.

لغات کلیدی—تشخیص صورت ، SVM

1- مقدمه

در این گزارش می‌خواهیم سه روش برای تشخیص حالت صورت را بررسی و پیاده‌سازی کنیم و در نهایت تصمیم بگیریم کدام متد برای کاربردهای معمول در محیط‌های هوشمند نتایج بهتری به دنبال دارد. ورودی مسئله می‌تواند ویدئویی از صورت و یا یک تصویر باشد. برنامه‌ی ارائه شده به عنوان خروجی یکی از هفت احساس جدول 1 را به هر فریم از فیلم ورودی یا تصویر نسبت می‌دهد.

احساس	خشم	نفرت	ترس	شادی	غم	هیجان	خنثی
شماره	1	2	3	4	5	6	7

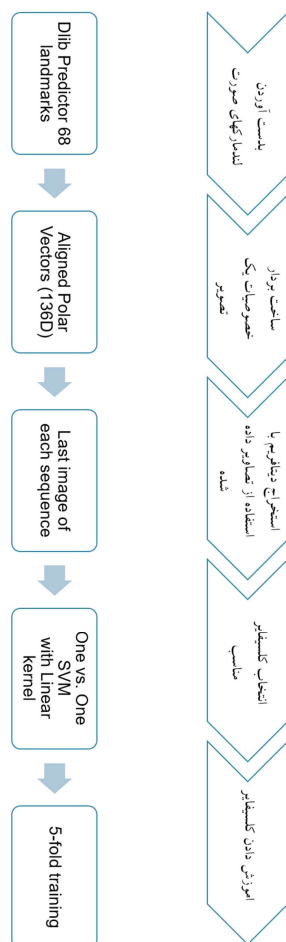
جدول شماره 1 - لیست احساسات

1-2- مروی بر ادبیات موضوع

برای حل این مسئله تا کنون متدهای مختلفی ارائه شده‌اند. بعضی بر پایه‌ی شبکه‌های عصبی بوده^[2,3] و بعضاً با استفاده از روش‌های کلسیفیکیشن مانند SVM^[4] با کرنل‌های مختلف، Decision Tree^[5] و غیره. بعلت در دسترس نبودن دیتاست مناسب و بزرگ برای روش‌های یادگیری عمیق، در این پروژه از کلسیفیکیشن استفاده شده‌است، هرچند برداشت می‌شود که برای مسائل بینایی ماشین در صورت در اختیار داشتن حجم-های بزرگ از داده، شبکه‌های عصبی عملکرد قوی‌تری دارند. همچنین، انتخاب‌های گوناگونی برای استخراج بردار ویژگی‌ها صورت گرفته است. که به تعدادی از آن‌ها اشاره می‌کنیم.

در ادامه نشان می‌دهیم که ابعاد بالاتر بردارها و یا آموزش دادن کلسیفایرهای پیچیده‌تر تاثیر بزرگی بر دقت پروژه نمی‌گذارند و تنها زمان اجرا را بالا می‌برند.

2-1- بلوک‌های الگوریتم



شکل 2- فلوچارت روند آموزش و به کارگیری کلسیفایر

2-2- دریافت تصویر، استخراج صورت و لندمارک‌ها

با استفاده از کتابخانه‌ی dlib از صورت 68 نقطه‌ی کلیدی استخراج می‌شود.



شکل 3 - لندمارک‌های صورت

- استفاده از خصوصیات هندسی لندمارک‌ها
- استفاده از EigenFace ها [6]
- استفاده‌ی مستقیم از تصویر چهره و استخراج بردار خصوصياتی همچون خروجی فیلترهای Gabor, SIFT و غیره

بعلت در دسترس داشتن کتابخانه‌ی dlib در پایتان و توابع کارآمد آن در تشخیص نقاط کلیدی صورت، همچنین بالاتر بودن سرعت استخراج خصوصیات هندسی در مقایسه با استخراج ویژگی‌های بصری تصویر مثل اعماق فیلتر یا توصیفگر سیفت، در این پروژه بردار ویژگی‌ها از نوع هندسی انتخاب شده‌اند تا با نیازهای محیط‌های هوشمند همخوانی داشته باشند.

2- ارائه مدل تشخیص صورت و حالت آن

2-0- پارامترهای موثر در عملکرد متد

در ابتدا پارامترهای مهم در حل این مسئله را بررسی می‌کنیم.

- حفظ واریانس معنی‌دار در دیتا
باید توجه داشته باشیم که واریانس در میان تصاویر دیتاست دو منشا دارد. تفاوت میان صورت دو فرد متمایز و تفاوت میان دو دسته حالت صورت. هدف این است که با نرمال کردن دیتا به نحوی واریانس مربوط به تمایز صورت انسان‌ها، تفاوت ناشی از تصویربرداری، نورپردازی، اندازه‌ی صورت و کیفیت تصویر را از بین ببریم. در عین حال این تلاش‌ها برای یکنواخت کردن تصاویر دیتاست نباید منجر به از بین رفتن واریانس میان حالت صورت در هنگام داشتن احساس خاصی بشود.
بعنوان مثال، ممکن است نرمال کردن اندازه‌ی صورت بر حسب فاصله‌ی لندمارک‌های حاشیه‌ای منجر به از بین رفتن واریانس در بالا رفتن ابروها به هنگام تعجب شود.

- برقراری توازن میان دقت و زمان اجرا
به وضوح می‌توان بردار خصوصياتی با چند هزار عضو ساخت و تصاویر را با دقت بالایی توصیف کرد و از طرفی کلسیفایر بهتری آموزش داد. اما در مقابل باید به زمان اجرای متد در حالت تست و Over fit کردن کلسیفایر به دیتاست نیز توجه داشت. با دقت به این مسئله که دیتاست ما تعداد محدودی تصویر دارد، بیشتر بودن ابعاد بردار خصوصیات از اندازه‌ی دیتاست می‌تواند نتایج ناگواری به دنبال بیاورد.

همچنین برای نرمال کردن اندازه‌ی بردارها نقاط پیرامون صورت را با کمک همین کتابخانه تشخیص داده می‌شوند.



شکل 4 - تشخیص صورت

نسبت به مختصات کارترین و می‌توان نشان داد که نسبت به تغییرات ناخواسته در دیتاست مقاومت بهتری دارد.

تنها عملی که برای نرمال‌سازی در این بخش انجام می‌شود چرخاندن صورت با توجه به بایاس زاویه‌ی بینی است.

- حذف بایاس صورت از زوایا

$$\bar{\theta} = \theta - \theta_{nose}$$

که زاویه‌ی بینی با استفاده از زاویه‌ی خط گذرنده از لندمارک-های ثابت روی بینی محاسبه شده است.

$$\theta_{nose} = \tan^{-1} \frac{y'_{30} - y'_{28}}{x'_{30} - x'_{28}} \pm \frac{\pi}{2}$$

اندازه‌ی این بردارها برای از بین رفتن واریانس میان حالات مختلف صورت نرمال نشده‌اند.

بدین ترتیب یک بردار ویژگی 136 بعدی (68 بعد اندازه و 68 بعد زاویه) از مختصات قطبی دوران داده شده بدست می‌آید.

2-3-2 بردار جابه‌جایی نرمال مختصات لندمارک‌ها

مراحل زیر برای استخراج این بردار ویژگی بر روی اولین و آخرین تصویر هر دنباله انجام می‌شوند.

- تغییر مبدا تصویر به مرکز جرم لندمارک‌های ثابت یعنی شماره‌ی [1,16] و محاسبه‌ی مختصات جدید لندمارک‌ها در هر دو تصویر به صورت جداگانه

$$\vec{R}_{avg_j} = \frac{1}{16} \sum_{i=1}^{16} \vec{r}_{ij}$$

$$\vec{r}'_{ij} = \vec{r}_{ij} - \vec{R}_{avg_j}$$

که بردار \vec{r}'_{ij} مختصات اولیه‌ی لندمارک i در تصویر j است.

- با استفاده از تشخیص گر صورت ابعاد صورت محاسبه شده و بردارهای \vec{r}'_{ij} نرمال می‌شوند.

$$X_{face_j} = \Delta x_{detector_j}$$

$$Y_{face_j} = \Delta y_{detector_j}$$

$$\vec{r}''_{ij} = \frac{\vec{r}'_{ij} \cdot \hat{x}}{X_{face_j}} \hat{x} + \frac{\vec{r}'_{ij} \cdot \hat{y}}{Y_{face_j}} \hat{y}$$

2-3 استخراج بردار ویژگی‌های صورت

در این بخش با توجه به پارامترهای نام برده شده در 2-0 سه بردار ویژگی مختلف استخراج می‌کنیم و در نهایت کلسیفایر را با هر سه آموزش می‌دهیم تا مشخص شود کدامیک از عملکرد بهتری برخوردار هستند.

2-3-1 مختصات نرمال نشده‌ی لندمارک‌ها

مراحل زیر برای استخراج این بردار ویژگی بر روی آخرین تصویر هر دنباله از دیتاست انجام می‌شوند.

- تغییر مبدا تصویر به مرکز جرم \vec{R}_{avg} و محاسبه‌ی مختصات جدید لندمارک‌ها

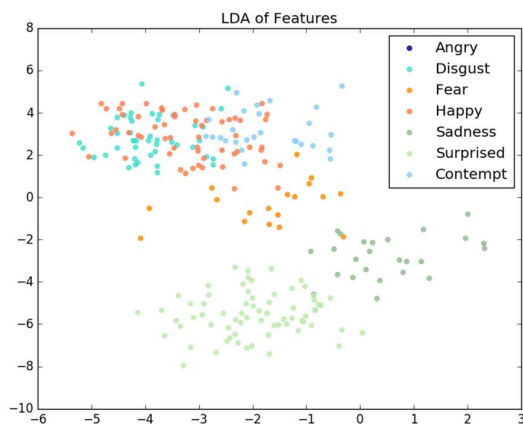
$$\vec{R}_{avg} = \frac{1}{68} \sum_{i=1}^{68} \vec{r}_i$$

$$\vec{r}'_i = \vec{r}_i - \vec{R}_{avg}$$

که بردار \vec{r}'_i مختصات اولیه‌ی لندمارک‌ها در تصویر است.

- تغییر دستگاه مختصات از کارترین به قطبی

می‌دانیم که مختصات قطبی نسبت به دکارتی نسبت به تبدیلات خطی و نویز و واریانس کاذب در دیتا (تفاوت) صورت‌ها و غیره) مقاوم‌تر است چرا که با استفاده از r, θ توصیف می‌شود. با توجه به اینکه مولفه‌ها را با دقت مشخصی (مثلا 2 رقم اعشار) نگه می‌داریم، فضای توصیف شده توسط مختصات قطبی کاردینالیتهی کمتری دارد



شکل 6 – LDA

واریانس جمعی توصیف شده توصیف این 10 بعد برابر:

$$\sigma_{10} = 0.9717 \times \sigma_{2278 \times}$$

است. پس با فیچر وکتوری به طول 10 (به جای 4556). 97% واریانس داده توصیف می‌شود.

2-4- انتخاب کلسیفایر متناسب با ویژگی‌ها

2-4-0 نحوه‌ی آموزش و تست کلسیفایرها

هر کدام از کلسیفایرها با روش k-fold به ازای $k = 5$ آموزش داده شده‌اند. دیتای تست و آموزش هر بار به صورت تصادفی از میان دیتاست به نسبت 20-80 انتخاب می‌شوند.

نمره‌ای که به هر کلسیفایر نسبت داده شده با امتحان آن بر روی تمامی تصاویر دیتاست بدست آمده است. هر کلسیفایر 3 بار آموزش دیده و تست شده و نمره‌ی ثبت شده میانگین این مقادیر است.

2-4-1 کلسیفایر مناسب برای روش 2-3-1

کلسیفایر SVM با کرنل‌های متفاوت و AdaBoost بررسی شده‌اند که نتایج در جدول 2 آمده‌اند.

Classifier	Score
Linear SVC (One vs. One)	87.45%.
Linear SVC (One vs. All)	90.53%.
RBF SVC	66.7%
Polynomial SVC (d=5)	86.2%
Polynomial SVC (d=7)	84.7%
AdaBoost (no adjustments)	49.02%

اکنون مختصات دکارتی نرمال شده‌ی لندمارک‌ها نسبت به مرکز صورت را در هر دو تصویر داریم.

- جابه‌جایی هر لندمارک نسبت به حالت اولیه را محاسبه می‌شوند.

$$\Delta r_i = \vec{r'_{i2}} - \vec{r'_{i1}}$$

- بردارهای جابه‌جایی در دستگاه مختصات قطبی بازنویسی می‌شوند، بنا به دلایل ذکر شده.
- تفاضل بایاس زوایا حذف می‌شود.

بایاس زاویه‌ی صورت مانند روش 2-3-1 محاسبه شده و

بر بردار جابه‌جایی در مختصات قطبی اعمال می‌شود.

$$\bar{\Delta \theta} = \Delta \theta - (\theta_{nose2} - \theta_{nose1})$$

بدین ترتیب یک بردار ویژگی 136 بعدی از مختصات

قطبی بردار جابه‌جایی بدست می‌آید که نسبت به تمامی تبدیل‌های affine مقاوم است.

2-3-3 بردارهای میان هر دو لندمارک‌ها با کاهش بعد

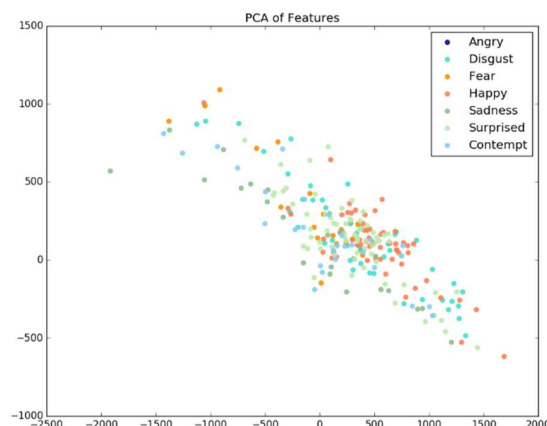
مراحل زیر برای استخراج این بردار ویژگی بر روی آخرین تصویر هر دنباله از دیتاست انجام می‌شوند.

- بردار میان هر دو لندمارک محاسبه می‌شود.

$$\forall i \neq j, \quad d_{ij} = r_i - r_j$$

که i, j در بازه‌ی [1,68] هستند. پس 2278 بردار دو بعدی خواهیم داشت که نیاز به کاهش بعد دارند.

- با استفاده از PCA و LDA بردار فیچرها را به 10 بعد کاهش می‌دهیم. نمودار پراکندگی دیتا در دو بعد به ازای هر دو نوع کاهش در زیر آمده است.

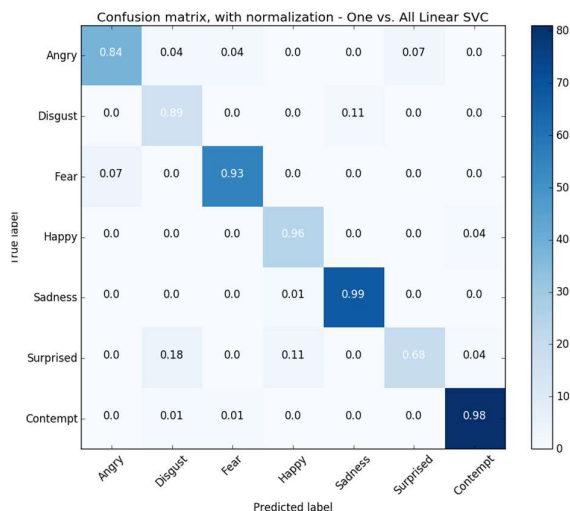


شکل 5 – PCA

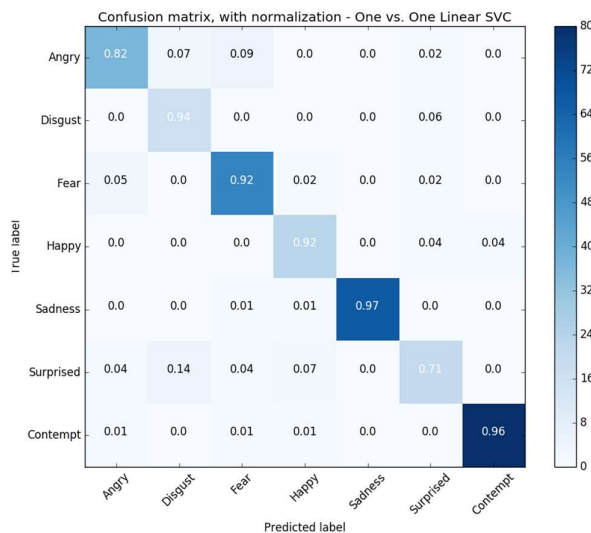
(شکل 13) کلسیفایر آموزش دیده با خروجی الگوریتم LDA در اصل به دیتا Over fit شده است و در عمل بر عکس‌های خارج از این دیتاست عملکرد دقیقی نخواهد داشت.

3- تعیین عملکرد و ماتریس Confusion

برای هر سه دسته بردار ویژگی، ماتریس confusion برای دو کلسیفایر بهتر در ادامه آمده‌اند.



شکل 7 - ماتریس با مختصات نرمال نشده‌ی لندمارک‌ها



شکل 8 - ماتریس با مختصات نرمال نشده‌ی لندمارک‌ها

می‌توان دید که کلسیفایر خطی برای این بردار ویژگی بهترین عملکرد را دارد. کلسیفایرهای چندجمله‌ای هم عملکرد مناسبی دارند، هرچند با زیاد کردن درجه Over fitting اتفاق افتاده و قابل اطمینان نیستند.

2-4-2 کلسیفایر مناسب برای روش 2-3-2

مشابه بخش قبل نتایج زیر بدست می‌آیند. می‌توان دید که روش پیشین عملکرد بهتری دارد.

Classifier	Score
Linear SVC (One vs. One)	65.11%
Linear SVC (One vs. All)	80.53%
RBF SVC	25.38%
Polynomial SVC (d=3)	25.38%
Polynomial SVC (d=5)	25.38%
AdaBoost (no adjustments)	58.81%

3-4-2 کلسیفایر مناسب برای روش 3-3-2

- نتایج با بردار خصوصیات 4456 بعدی:

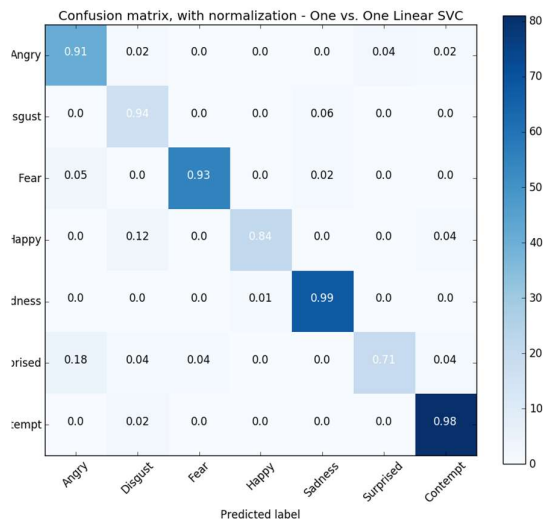
Classifier	Score
Linear SVC (One vs. One)	93.38%
Linear SVC (One vs. All)	91.59%
RBF SVC	25.38%
Polynomial SVC (d=5)	91.94%
Polynomial SVC (d=7)	90.92%
AdaBoost (no adjustments)	49.23%

- نتایج با بردار خصوصیات 10 بعدی:

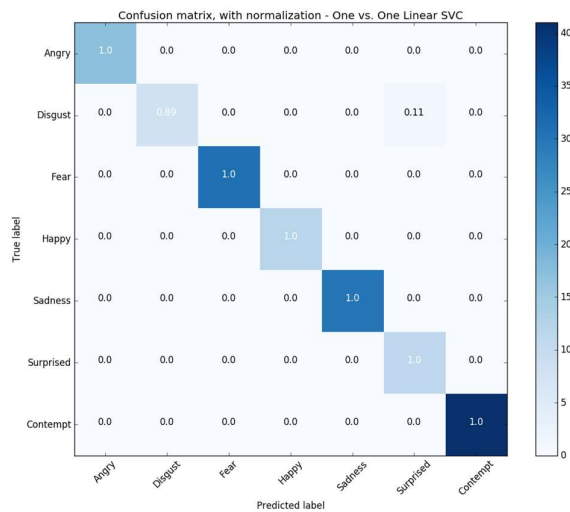
Classifier	Score
Linear SVC (One vs. One)	58.30%
Linear SVC (One vs. All)	81.19%
RBF SVC	26.3%
Polynomial SVC (d=5)	60.84%
Polynomial SVC (d=7)	76.54%
AdaBoost (no adjustments)	33.23%

دیده می‌شود که کاهش بعد در عملکرد کلسیفایرها تاثیر منفی زیادی داشته است. پس در عمل با بردار خصوصیات اصلی کلسیفایرها را آموزش خواهیم داد.

توجه داشته باشید، اگر از روش LDA استفاده کنیم، چون کلسیفیکیشن LDA بر تمام دیتا انجام شده، هنگام آزمون کلسیفایر SVM به دقت بسیار بالایی می‌رسیم که کاذب است.

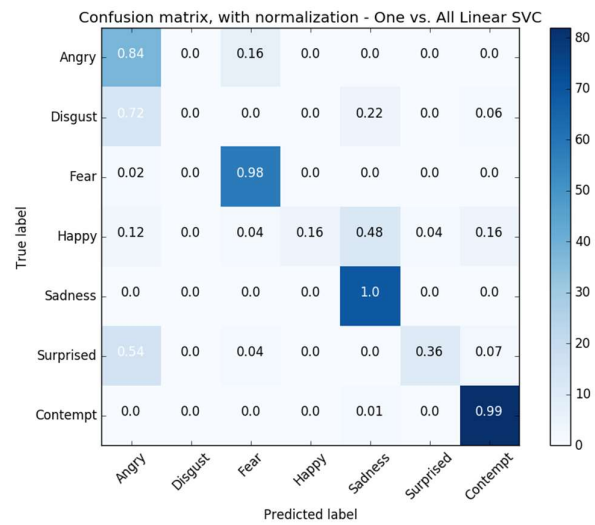


شکل 12 – ماتریس با بردارهای میان هر دو لندمارک‌ها با کاهش بعد

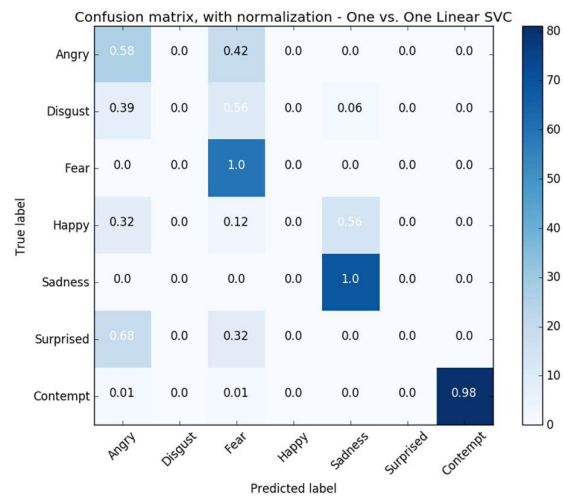


شکل 13 – کلسیفایر Over fit شده، آموزش دیده با خروجی LDA

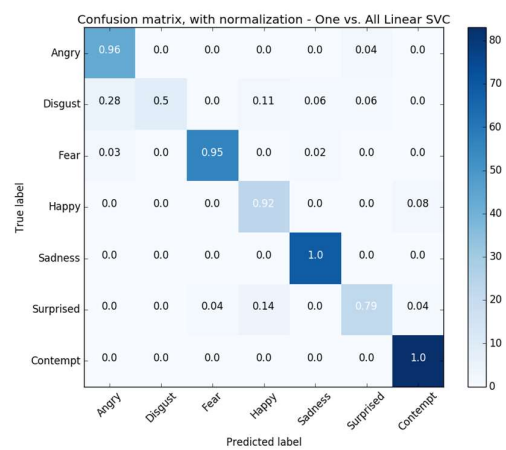
4- نتایج بر داده‌های خارج و داخل دیتاست
در ادامه کلسیفایر مربوط به شکل 12 بر روی چند نمونه تصویر آزموده شده‌است.



شکل 9 – ماتریس با بردار جابه‌جایی نرمال مختصات لندمارک‌ها

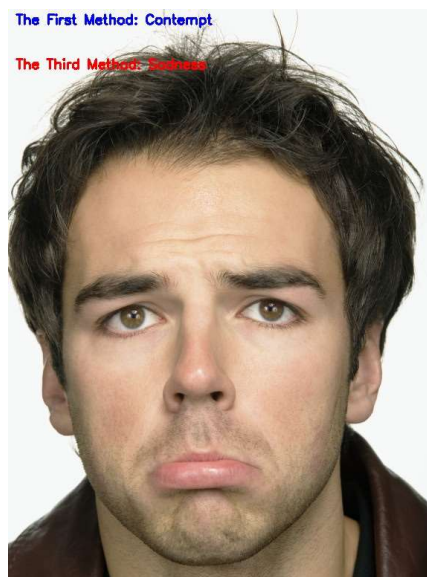
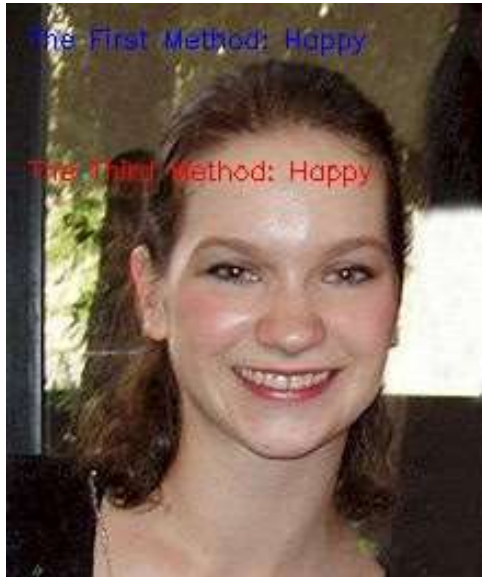


شکل 10 – ماتریس با بردار جابه‌جایی نرمال مختصات لندمارک‌ها

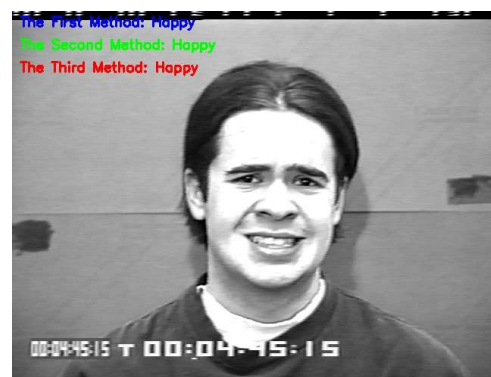
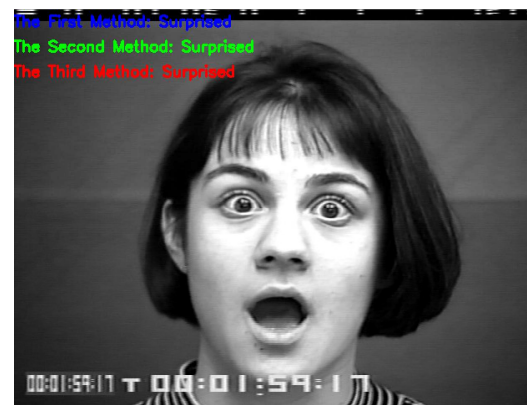
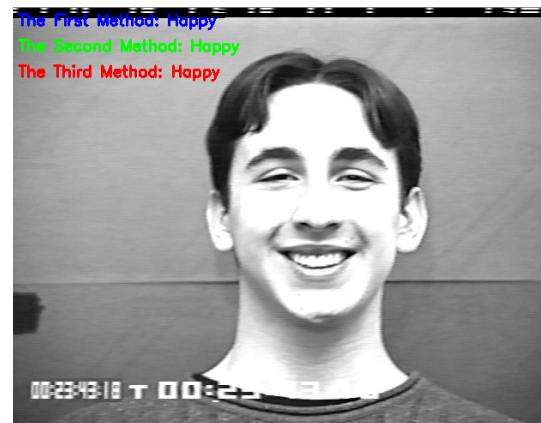


شکل 11 – ماتریس با بردارهای میان هر دو لندمارک‌ها با کاهش بعد

• تصاویر خارج دیتاست



• تصاویر داخل دیتاست



5- نتیجه گیری و کارهای آینده

با مقایسه‌ی عملکرد کلسیفایرها به این نتیجه می‌رسیم که دسته‌بند خطی SVM در حالت One vs. One با فیچر وکتوری با بعد 4456 بهترین عملکرد را دارد. دقت تشخیص هر کدام از احساس‌ها در زیر آمده‌است.

احساس	خشم	نفرت	ترس	شادی	غم	هیجان	خنثی
دقت	0.91	0.94	0.93	0.84	0.99	0.71	0.96

حالت های شادی-خنثی و عصبانیت-نفرت به خوبی از یکدیگر تفکیک نشده اند. در آینده می‌توان با کمک متدهای زیر الگوریتم را بهبود بخشید.

- استفاده از Eigen Face ها
- دسته بندی به صورت Cascade برای جداسازی این سه حالت
- استفاده از چند فریم و رای گیری میان آنها
- تغییر روش یادگیری به شبکه های عصبی

6- سپاسگزاری

در انتها از تمامی زحمات استاد ارجمند دکتر کربلایی آقاچان و همینطور دستیاران آموزشی آقایان آیین رضایی، افشین بزرگپور و دیگران تشکر می‌شود.

7- منابع

1. Facial gesture recognition in face image sequences: A study on facial gestures typical for speech articulation, M. Pantic and L.J.M. Rothkrantz, TUDelft
2. Emotion Recognition Using Facial Landmarks, Van Gent, P. (2016).
3. Facial Expression Recognition, 2015, Rustem Bekmukhametov
4. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, 2010, Lucey et al.
5. EigenfaceBased Recognition of Emotion Variant Faces, 2014, Thuseethan et al.
6. Facial Expression Recognition Using Eigenspaces, 2013, D. Chakrabarti et al.