

# کاربرد دسته‌بندی کننده‌های مبتنی بر بازنمایی تنک در دسته‌بندی تصاویر

احمد اسدی - ۹۴۱۳۱۰۹۱

دی ماه ۱۳۹۵

## چکیده

دسته‌بندی کننده‌های مبتنی بر بازنمایی تنک عملکرد خوبی در زمینه دسته‌بندی تصاویر، مخصوصاً در زمینه دسته‌بندی تصاویر صورت افراد، از خود نشان داده‌اند. به دلیل اهمیت استفاده از چنین دسته‌بندی کننده‌هایی، در این گزارش به تفسیر و تبیین این دسته از دسته‌بندی کننده‌ها و معرفی برخی روش‌های بهبود کارایی آن‌ها خواهیم پرداخت. همان‌طور که خواهیم دید، چنین دسته‌بندی کننده‌هایی در مواقعی که داده‌های کلاس‌های مختلف روی یک بردار جهت یکسان توزیع شده باشند، دچار مشکل می‌شود. بنابراین با توسعه این روش با استفاده از یک توسعه غیر خطی، دسته‌بندی کننده‌ای تحت عنوان دسته‌بندی کننده بازنمایی تنک با تابع هسته را که در سال ۲۰۱۲ ارائه شده است، معرفی خواهیم نمود. استفاده از توسعه غیرخطی روش بازنمایی تنک، موجب افزایش کارایی مدل شده و امکان ترکیب آن با سایر مدل‌ها را فراهم می‌آورد. به عنوان نمونه مدل ترکیبی از بازنمایی تنک و تطبیق هرم مکانی ارائه شده است که در زمینه دسته‌بندی تصاویر، دقت بالایی را از خود نشان داده است. در انتها روشی را ارائه خواهیم داد که در آن با ترکیب روش یادگیری چند هسته‌ای و بازنمایی تنک، کدهای تنک و وزن‌های هسته در دو مرحله به مدل آموزش داده می‌شوند. این مدل روی مجموعه داده‌های مختلفی به منظور دسته‌بندی تصاویر استفاده شده و کارایی خوبی از خود نشان داده است.

## ۱ مقدمه

عموماً تنک‌بودن در حوزه یادگیری ماشین منجر به افزایش قدرت تعمیم‌پذیری<sup>۳</sup> مدل می‌شود. علاوه بر این، مدل‌های تنک را می‌توان با استفاده از الگوریتم‌هایی با پیچیدگی زمانی پایین‌تر آموزش داد و از آن‌ها استفاده نمود. از این رو وجود چنین خاصیتی در مدل‌ها، مطلوب است. مقارن با معنای فوق که لزوم توجه به تنک‌بودن مدل را مشخص می‌نماید، می‌توان میزان وابستگی تصمیم‌گیری به داده‌های آموزشی را به عنوان معیار ساده دیگری از تنک‌بودن در نظر گرفت. با فرض این معیار، شباهتی بین این مدل و ماشین‌های بردار پشتیبان<sup>۴</sup> وجود خواهد داشت.

مفهوم تنک‌بودن<sup>۱</sup> بسته به موضوع پژوهش و بحث می‌تواند معانی مختلفی به خود بگیرد. در زمینه یادگیری ماشین، تنک بودن عموماً به مواردی اطلاق می‌شود که در آن‌ها مدل ارائه شده دارای تعداد زیادی مقدار پوچ<sup>۲</sup> باشد. با بهره‌گیری از این تعریف، میزان تنک‌بودن را می‌توان با شمارش تعداد ضرایب صفر در بردار پارامترهای مدل، ارزیابی نمود به طوری که هر چه تعداد ضرایب صفر موجود در بردار پارامترهای یک مدل افزایش یابد، مدل تنک‌تر می‌شود.

<sup>۳</sup> Generalizability  
<sup>۴</sup> Support Vector Machine (SVM)

<sup>۱</sup> Sparsity  
<sup>۲</sup> Null Values

در مدل ماشین بردار پشتیبان، سعی می‌شود تا حد امکان از تعداد کمتری از داده‌های آموزشی، بردارهای پشتیبان<sup>۵</sup>، برای برچسب‌دهی و تصمیم‌گیری استفاده شود. در این‌جا نیز هر چه مدل به تعداد کمتری از داده‌های آموزشی وابسته شود، تنک‌تر و مطلوب‌تر است.

استفاده از بازنمایی تنک در دسته‌بندی، از ضرورت یافتن ویژگی‌های مناسب برای دسته‌بندی می‌کاهد. از آنجا که فضای داده‌ها در بازنمایی تنک، یک فضای با ابعاد بسیار بالا است، حتی استفاده از ویژگی‌های تصادفی، اطلاعات کافی را برای دسته‌بندی داده‌ها فراهم می‌آورد و از این طریق فرآیند استخراج ویژگی، که همواره از جمله حساس‌ترین و مهم‌ترین فرآیندها در پژوهش‌های مربوط به یادگیری ماشین بوده، تا حد بسیار خوب و قابل توجهی تسهیل می‌یابد.

در پژوهش [۴] که در سال ۲۰۰۹ ارائه شده است، یک دسته‌بندی کننده مبتنی بر بازنمایی تنک ارائه شده است و از آن در تشخیص چهره استفاده شده است. دسته‌بندی کننده مبتنی بر بازنمایی تنک یک مدل ناپارامتری<sup>۶</sup> فاقد فاز یادگیری است. این مدل قادر است با استفاده از یک مجموعه داده آموزشی که به همراه برچسب صحیح، به مدل داده شده‌اند، مستقیماً برچسب مربوط برای داده‌های تست را مشخص نماید.

همان‌طور که در ادامه خواهیم دید، دسته‌بندی کننده مبتنی بر بازنمایی تنک ارائه شده توسط رایت و همکارانش در [۴] در مواردی که بردار توزیع داده‌های دو کلاس، یکسان یا به هم نزدیک باشند، با مشکل در تشخیص روبرو می‌شود. این مشکل به قدری جدی است که مدل ارائه شده حتی در مواردی که داده‌ها به طور خطی جداپذیر باشند در صورتی که شرط مذکور صادق باشد، عملکرد مناسبی از خود نشان نمی‌دهد.

ژنگ و همکارانش در سال ۲۰۱۲ با ارائه پژوهش [۱] و با ترکیب یک تابع هسته با دسته‌بندی کننده مبتنی بر بازنمایی تنک، مشکل مدل [۴] را مرتفع ساختند. استفاده از توابع هسته مختلف، این امکان را می‌دهد که با افزایش بعد داده‌ها و سپس استفاده از یک روش کاهش بعد

مناسب، داده‌ها به فضای جدیدی نگاشت شوند که در آن تفکیک پذیری داده‌های کلاس‌های مختلف، افزایش بیابد و با ایت کار بتوان مشکل دسته‌بندی کننده مبتنی بر بازنمایی تنک را مرتفع ساخت.

ترکیب دسته‌بندی کننده مبتنی بر بازنمایی تنک با توابع هسته، امکان انجام فعالیت‌ها و پژوهش‌های مختلف و متنوعی را برای پژوهش‌گران در این زمینه ایجاد کرده است. از جمله پژوهش‌هایی که با اتکا بر روش‌های پیشین و به منظور ارتقا آن‌ها ارائه شده است می‌توان به پژوهش شریواستاوا و همکارانش در [۳] اشاره کرد. در این پژوهش، روشی شامل دو مرحله ارائه شده است که در آن ابتدا با استفاده از یک روش یادگیری، وزن‌های تابع هسته آموزش داده می‌شوند و سپس با استفاده از کدهای تنک محاسبه شده توسط کرنل آموزش دیده، عمل دسته‌بندی انجام می‌شود.

از نمونه‌های دیگر فعالیت‌های مشابه که با اتکا به ترکیب توابع هسته و دسته‌بندی کننده‌های مبتنی بر بازنمایی تنک ارائه شده است، پژوهشی است که گائو و همکارانش در سال ۲۰۱۳ در [۲] ارائه داده‌اند. در این پژوهش با ترکیب روش تطبیق هرم مکانی و دسته‌بندی کننده مبتنی بر بازنمایی تنک با تابع هسته، اقدام به دسته‌بندی تصاویر نموده‌اند.

در ادامه این گزارش، در بخش دوم، ایده اصلی دسته‌بندی کننده مبتنی بر بازنمایی تنک را ارائه می‌دهیم. در بخش سوم این گزارش، ایده استفاده از توابع هسته برای افزایش بعد و ترکیب آن با دسته‌بندی کننده مبتنی بر بازنمایی تنک را مورد بررسی قرار می‌دهیم. بخش‌های چهارم و پنجم گزارش، به ترتیب به بررسی روش مبتنی بر یادگیری و روش مبتنی بر تطبیق هرم مکانی می‌پردازند. در نهایت یک جمع‌بندی از مطالب ارائه شده در این گزارش و مقایسه نقاط ضعف و قوت روش‌های ارائه شده خواهیم پرداخت.

## ۲ دسته‌بندی کننده مبتنی بر بازنمایی تنک

که در آن  $N = \sum_{i=1}^k n_i$  تعداد کل داده‌های آموزشی موجود است. با استفاده از ماتریس  $A$  می‌توان رابطه (۱) را بر اساس داده‌های آموزشی کلاس  $i$ ام به شکل رابطه (۲) بازنویسی کرد.

$$y = Ax_i \quad (2)$$

که در آن:

$$x_i = [0, \dots, 0, \alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,n_i}, 0, \dots, 0]$$

برداری از تمام ضرایب کلاس  $i$ ام است. با استفاده از بازنمایی ارائه شده در (۲) داده تست  $y$  را به ازای  $x_i$ های مختلف بازنمایی کرده و از خطای بازسازی<sup>۷</sup> برای تصمیم‌گیری استفاده می‌نماییم. به این معنی که تصویر  $y$  را به کلاسی نسبت می‌دهیم که  $y - Ax_i$  در آن مقدار کمتری داشته باشد.

از آنجا که بردار  $x_i$  مستقیماً در برچسب‌دهی تصاویر تست تاثیر دارد، می‌توان مقدار مولفه‌های آن را با استفاده از بهینه‌سازی مساله خطی  $y = Ax_i$  محاسبه نمود. بدون وارد آمدن خللی به مساله می‌توانیم به جای آن که مقادیر مولفه‌های بردار  $x_i$  را به ازای  $i$ های مختلف با بهینه‌سازی بیابیم، مستقیماً بردار  $x = [x_1, \dots, x_k]$  را با بهینه‌سازی مساله خطی (۳) به دست بیاوریم.

$$\text{minimize } y = Ax \quad (3)$$

در صورتی که بدانیم  $m > n$  است، مساله (۳) به یک دستگاه معادلات فرامعین<sup>۸</sup> تبدیل می‌شود که می‌توان به طور تحلیلی آن را حل کرد. همین‌طور در صورتی که بدانیم  $m < n$  است، با یک دستگاه معادلات<sup>۹</sup> مواجه می‌شویم که پاسخ آن منحصر به فرد نیست. به طور معمول در چنین مواردی با حل مساله نرم<sup>۱۰</sup> مطابق با رابطه (۴) و یافتن

در این بخش به بررسی مدل دسته‌بندی کننده مبتنی بر بازنمایی تنک که رایت و همکارانش در [۴] ارائه داده‌اند، می‌پردازیم. همان‌طور که قبلاً گفته شد، ایده اصلی در این پژوهش، استفاده از ترکیب خطی داده‌های آموزشی برای به دست آوردن یک بازنمایی تنک از فضای داده‌ها است. بازنمایی مذکور با استفاده از بهینه‌سازی ریاضیاتی، طوری تعیین می‌شود که کمترین داده‌های آموزشی مورد استفاده قرار بگیرند تا مدل تا حد ممکن تنک شود و از طرف دیگر خللی در دسته‌بندی داده‌ها ایجاد نشود.

### ۱.۲ تئوری

با فرض این که داده‌های آموزشی از  $k$  کلاس مختلف فراهم شده‌اند و تعداد داده‌های آموزشی فراهم شده از کلاس  $i$ ام را با  $n_i$  نمایش دهیم، مجموعه داده‌های آموزشی کلاس  $i$ ام را با  $A_i = \{\nu_{i,1}, \nu_{i,2}, \dots, \nu_{i,n_i}\}$  نمایش می‌دهیم که  $A \in R^{m \times n_i}$ . از آنجا که قصد داریم از این روش در تشخیص چهره استفاده نماییم، داده‌های آموزشی را که تصویر چهره‌ها هستند به شکل یک بردار  $\nu \in R^m$  در نظر می‌گیریم که در آن  $m = w * h$  و  $w$  و  $h$  به ترتیب عرض و ارتفاع تصویر را نمایش می‌دهند. به عبارت ساده‌تر تمام پیکسل‌های تصویر را در یک بردار قرار می‌دهیم و از بردارهای حاصل در بقیه فرایندها استفاده می‌نماییم.

با استفاده از ایده اصلی که مطرح شد، می‌توانیم یک تصویر با تست  $y \in R^m$  را بر اساس داده‌های آموزشی کلاس  $i$ ام به شکل رابطه (۱) مدل کنیم که در آن  $\alpha_{i,j}$ ها همگی اعداد حقیقی هستند.

$$y = \alpha_{i,1}\nu_{i,1} + \alpha_{i,2}\nu_{i,2} + \dots + \alpha_{i,n_i}\nu_{i,n_i} \quad (1)$$

برای این که بتوانیم تصویر تست  $y$  را بر اساس تمام داده‌های آموزشی مدل کنیم و میزان تعلق آن را به تمام کلاس‌های موجود محاسبه نماییم، ماتریس  $A$  را از حاصل کنارهم قرار دادن ماتریس داده‌های آموزشی تمام کلاس‌ها ایجاد می‌نماییم.

$$A = [A_1, A_2, \dots, A_k] \in R^{m \times N}$$

<sup>۷</sup> Reconstruction Error  
<sup>۸</sup> Overdetermined Equation System  
<sup>۹</sup> Underdetermined Equation System  
<sup>۱۰</sup>  $l^2$ -norm

پاسخ آن، دستگاه معادلات، حل می‌شود.

$$\begin{aligned} & \text{minimize } \|x\|_2 \\ & \text{subject to } y = Ax \end{aligned} \quad (4)$$

روش‌های مختلفی برای حل مساله (4) در زمینه بهینه‌سازی ریاضیاتی وجود دارد. با این وجود پاسخ این مساله، عموماً پاسخ مناسبی برای ما نیست به این دلیل که این پاسخ توانایی زیادی در تفکیک داده کلاس‌ها از یک‌دیگر ندارد. آزمایشات نشان می‌دهد پاسخ این مساله، دارای تعداد زیادی ضریب غیر صفر بزرگ از تمام کلاس‌ها است که باعث کاهش قدرت تفکیک داده‌ها توسط مدل می‌شود.

ایده اصلی روش در حل این مشکل بسیار کارساز است. در شرایطی که تعداد کلاس‌های مساله زیاد باشد، بازنمایی ارائه شده در این پژوهش، یک بازنمایی تنک است که باعث می‌شود تراکم ضرایب غیر صفر در یکی از کلاس‌ها زیاد و در کلاس‌های دیگر نزدیک به صفر باشد که قدرت تفکیک خوبی به مدل می‌دهد. به همین منظور به جای استفاده از تابع نرم 2، از تابع نرم صفر<sup>۱۱</sup> مطابق با (5) استفاده می‌شود.

$$\begin{aligned} & \text{minimize } \|x\|_0 \\ & \text{subject to } y = Ax \end{aligned} \quad (5)$$

تابع نرم صفر برابر است با تعداد مولفه‌های غیر صفر یک بردار. با وجود این که مساله (5) می‌تواند مشکلات موجود را حل کند، این مساله از جمله مسائل ان پی سخت<sup>۱۲</sup> است و راه حل سریعی برای حل آن در حالت کلی وجود ندارد. با این حال، می‌توان با در نظر گرفتن شرایط خاص مسائل، راه‌حل‌های خوبی برای آن پیدا کرد.

از طرفی می‌توان پاسخ مساله (5) را در حالتی که به دنبال تنک‌ترین پاسخ آن هستیم با مساله بهینه‌سازی نرم<sup>۱۳</sup> مطابق با (6) تخمین بزنیم. همان‌طور که مشخص است در این مساله برای در نظر گرفتن نویزی که به طور معمول در داده‌ها وجود دارد، قید تساوی  $y = Ax$  را به  $\|Ax - y\|_2 < \epsilon$  تغییر دادیم که در آن  $\epsilon > 0$  یک مقدار بسیار کوچک است.

$$\begin{aligned} & \text{minimize } \|x\|_1 \\ & \text{subject to } \|Ax - y\|_2 < \epsilon \end{aligned} \quad (6)$$

مساله (6) با استفاده از برنامه‌ریزی مخروطی مرتبه دوم<sup>۱۴</sup> قابل حل است و اثبات می‌شود که پاسخ آن به پاسخ (5) همگرا می‌شود به طوری که ثابت‌های  $\rho$  و  $\zeta$  وجود خواهند داشت طوری که رابطه (7) را ارضا نمایند.

$$\|\hat{x}_1 - x_0\|_2 \leq \zeta \epsilon \quad (7)$$

که در آن  $x_0$  و  $x_1$  به ترتیب مقادیر بهینه مسائل (5) و (6) هستند.

## ۲.۲ الگوریتم

با ورود یک تصویر تست جدید  $y$  ابتدا باید بردار ضرایب را برای این تصویر به دست بیاوریم. این کار را می‌توان با بهینه‌سازی مساله (6) در حالت نویزی یا بدون نویز انجام داد. در مرحله بعد کفایت بردار  $x^*$  به دست آمده را توسط یک دسته‌بندی کننده، دسته‌بندی نماییم. این کار می‌تواند در ساده‌ترین حالت به این شکل انجام شود که تصویر ورودی را به کلاس مولفه‌ای از  $x^*$  که بیشترین مقدار را دارد تخصیص دهیم.

در این پژوهش به جای استفاده از دسته‌بندی کننده مذکور، همان‌طور که قبلاً ذکر شد، از خطای بازسازی داده برای دسته‌بندی استفاده می‌نماییم به این معنی که تصویر تست را با ضرایب داده‌های آموزشی همه  $k$  کلاس به طور جداگانه بازسازی نموده و خطای بازسازی را محاسبه می‌نماییم و در انتها، تصویر را به کلاسی که کم‌ترین خطای بازسازی را ایجاد نموده است تخصیص می‌دهیم. بازسازی تصویر می‌تواند با استفاده از رابطه (8) برای کلاس  $i$ ام انجام شود.

$$\hat{y}_i = Ax_i^* \quad (8)$$

دسته‌بندی کننده مذکور را می‌توان مطابق با (9) تعریف نمود.

$$\min_i r_i(y) = \|y - Ax_i^*\|_2 \quad (9)$$

Second Order Cone Programming<sup>۱۴</sup>

$l^0$ -norm<sup>۱۱</sup>  
NP-hard<sup>۱۲</sup>  
 $l^1$ -norm<sup>۱۳</sup>

**Data:** a matrix of training samples

$A = [A_1, A_2, \dots, A_k] \in R^{m \times n}$  for  $k$  classes, test sample  $y \in R^m$ , error tolerance  $\epsilon > 0$

1. Normalize the columns of  $A$  to have unit  $l^2$ -norm.
2. Solve (6) to find  $\hat{x}$ .
3. Compute the residuals  $r_i(y) = \|y - Ax_i\|_2$  for  $i = 1, 2, \dots, k$ .

**Result:**  $identity(y) = argmin_i r_i(y)$

**Algorithm 1:** الگوریتم دسته‌بندی با استفاده از [4] دسته‌بندی‌کننده مبتنی بر بازنمایی تنک

با توجه به این مطلب می‌توانیم مساله (۶) را به فرم (۱۰) بازنویسی نماییم.

$$\begin{aligned} & \text{minimize } \|x\|_1 \\ & \text{subject to } \|RAx - y\|_2 < \epsilon \end{aligned} \quad (10)$$

مساله‌ای که از اهمیت بالایی برخوردار است، انتخاب مناسب ماتریس  $R$  برای دست‌یافتن به دقت‌های دسته‌بندی بالاتر است. نتایج آزمایشات و بررسی‌های ریاضیاتی نشان داده‌اند که در صورتی که تعداد ابعاد فضای ویژگی به طور قابل قبولی زیاد باشد، مطابق با رابطه (۱۱)، می‌توان از یک ماتریس تصادفی برای  $R$  استفاده کرد و بهینه‌سازی نرم ۱ می‌تواند مقدار بهینه  $x^*$  را به خوبی بیابد.

$$d \geq 2t \log\left(\frac{n}{d}\right) \quad (11)$$

یکی از نکات بسیار مهم و کاربردی که در این پژوهش می‌توان به آن اشاره کرد این است که:

«می‌توان از ویژگی‌های تصادفی به منظور دسته‌بندی استفاده کرد که باعث کاهش پیچیدگی فرآیند استخراج ویژگی می‌شود.»

فرآیند توضیح داده شده را می‌توانید در الگوریتم ۱ مشاهده نمایید. در این الگوریتم ابتدا تمام تصاویر ورودی را که به شکل یک بردار  $m$  مولفه‌ای تبدیل شده‌اند، استانداردسازی<sup>۱۵</sup> می‌نماییم به نحوی که نرم ۲ همگی آن‌ها برابر با واحد شود. از آن‌جا که خطای نرمال‌سازی را به شکل  $\|Ax - y\|_2$  تعریف نموده‌ایم، این کار باعث می‌شود ضرب داخلی ماتریس  $A$  و بردار  $x$  دچار مقیاس اندازه نشود. اگر اندازه تصاویر با هم برابر نباشد، این مقیاس اندازه خطای بازسازی را تحت تاثیر قرار داده و موجب می‌شود کلاسی که خطای بازسازی کم‌تری را تولید می‌کند، لزوماً بهترین و نزدیک‌ترین کلاس برای تصویر ورودی  $y$  نباشد. در قدم شماره ۳ از الگوریتم ۱ مساله نویزی (۶) را حل می‌نماییم. برای حل این مساله می‌توان از ابزارهای آماده استفاده نمود. در این پژوهش از کتابخانه حل مسائل برنامه‌ریزی خطی نرم‌افزار مت‌لب<sup>۱۶</sup> استفاده شده است. در انتها با انتخاب ضرایب هریک از کلاس‌ها و صفر کردن مابقی ضرایب در بردار پاسخ، تصویر ورودی با ترکیب خطی داده‌های هر کلاس متناسب با ضرایب موجود، بازسازی شده و خطای بازسازی آن برای هر کلاس محاسبه می‌شود و کلاسی که کم‌ترین خطای بازسازی را داشته باشد به عنوان برچسب تصویر، انتخاب می‌شود.

## ۳.۲ استخراج ویژگی

برای مدل‌سازی استخراج ویژگی می‌توان از یک ماتریس تبدیل  $R$  استفاده کرد. قید تساوی در این حالت به شکل زیر تبدیل می‌شود:

$$y = RAx, \quad y \in R^d$$

که در آن  $d < n$  است. این تبدیل، تصاویر را از فضای تصاویر به فضای ویژگی‌ها نگاشت می‌کند. پژوهش‌ها نشان‌دهنده است افزایش بعد فضای ویژگی می‌تواند تاثیر مثبتی در بهبود عملکرد دسته‌بندی‌کننده و افزایش دقت دسته‌بندی داشته باشد.

<sup>۱۵</sup> Normalize

<sup>۱۶</sup> <https://en.mathworks.com/help/optim/ug/linprog.html>

## ۴.۲ آزمایشات

بردار پشتیبان خطی به ترتیب برابر با ۹۰.۷، ۹۴.۱ و ۹۷.۷ درصد بوده است.

در این بخش آزمایشات مختلفی را که در این پژوهش انجام شده است، مورد بررسی قرار می‌دهیم.

### ۱.۴.۲ تاثیر استخراج ویژگی

در این بخش آزمایشاتی انجام شده است تا تاثیر روش‌های مختلف استخراج ویژگی بر روی عملکرد نهایی دسته‌بندی‌کننده مورد بررسی قرار داده شود.

در آزمایشات این پژوهش از مجموعه داده‌ای<sup>۱۷</sup> شامل ۲۴۱۴ تصویر از چهره ۳۸ نفر در شرایط مختلف، انجام شده است. ابعاد همه تصاویر ۱۶۸ \* ۱۹۲ است. نیمی از تصاویر به طور تصادفی انتخاب و به عنوان تصاویر آموزشی مورد استفاده قرار گرفته‌اند و مابقی تصاویر به عنوان تصاویر تست استفاده شده‌اند. نتایج آزمایشات روی روش ارائه شده تحت عنوان اس‌آرسی<sup>۱۸</sup> و روش‌های نزدیک‌ترین همسایه<sup>۱۹</sup>، نزدیک‌ترین فضا<sup>۲۰</sup> و ماشین بردار پشتیبان خطی<sup>۲۱</sup> اندازه‌گیری و با یکدیگر مقایسه شده است.

در این آزمایش، الگوریتم‌های مختلف با تعداد ابعاد مختلفی از فضای ویژگی مورد استفاده قرار گرفته‌اند. تعداد ابعاد مورد آزمایش در این بخش به ترتیب ۳۰، ۵۶، ۱۲۰ و ۵۴۰ بعد بوده است.

شکل ۱ نتایج مقایسه آزمایشات را نمایش می‌دهد. همان‌طور که در این شکل مشخص است، الگوریتم ارائه شده در مقایسه با روش‌های دیگر به جز روش نزدیک‌ترین فضا، مقاومت بسیار خوبی در برابر تنوع روش‌های انتخاب و استخراج ویژگی از خود نشان داده است. همین‌طور دقت نهایی که الگوریتم به آن دست‌یافته است نسبت به بقیه روش‌ها بالاتر است. از طرفی عمل‌کرد الگوریتم بسیار شبیه به عمل‌کرد الگوریتم نزدیک‌ترین فضا است.

دقت دسته‌بندی کسب‌شده توسط روش اس‌آرسی بین ۹۲.۱ تا ۹۵.۶ درصد و بیشترین دقت کسب‌شده توسط روش‌های نزدیک‌ترین همسایه، نزدیک‌ترین فضا و ماشین

### ۲.۴.۲ استفاده از ویژگی‌های جزئی چهره

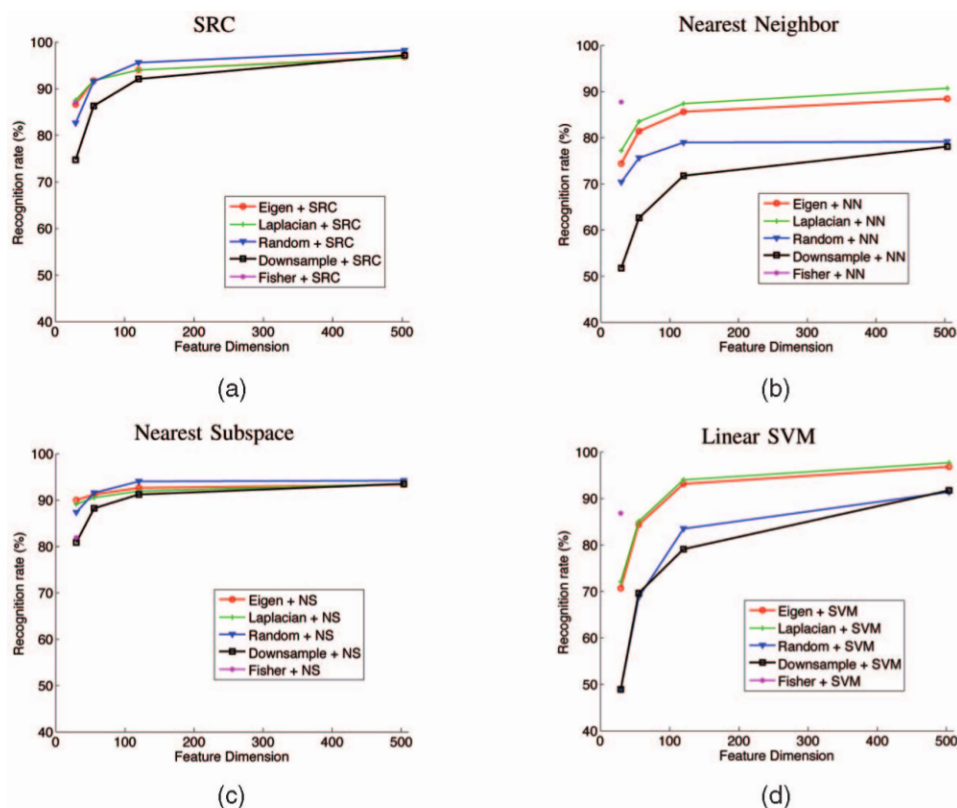
در بسیاری از پژوهش‌ها برای تشخیص چهره از ویژگی‌های جزئی چهره شامل گوش، چشم، بینی، دهان، چانه و انواع اندام‌های دیگر موجود در تصویر چهره استفاده شده است. همین‌طور پژوهش‌های زیادی بر روی نحوه انجام فرآیند تشخیص چهره در مغز انسان با استفاده از همین ویژگی‌های جزئی انجام شده است که باعث بالارفتن اهمیت این موضوع می‌شود.

در این آزمایش هدف، بررسی میزان صحت عملکرد الگوریتم در دسته‌بندی تصاویر و تشخیص چهره با استفاده از ۳ ویژگی بینی، گوش راست و دهان و چانه است. شکل ۲ سه ویژگی استخراج شده را برای یک نمونه از تصویر نمایش می‌دهد. نتایج عملکرد الگوریتم‌ها در جدول ۱ ارائه شده است. همان‌طور که در جدول مشاهده می‌شود، عملکرد الگوریتم اس‌آرسی در استفاده از هر ۳ ویژگی نسبت به سایر روش‌ها بهتر بوده است.

جدول ۱: جدول نتایج استفاده از الگوریتم‌ها در تشخیص چهره با استفاده از سه ویژگی جزئی چهره [۴]

ویژگی‌ها تعداد ابعاد	بینی ۴۲۷۰	گوش راست ۵۰۴۰	دهان و چانه ۱۲۹۳۶
SRC	۸۷.۳	۹۳.۷	۹۸.۳
NN	۴۹.۲	۶۸.۸	۷۲.۷
NS	۸۳.۷	۷۸.۶	۹۴.۴
SVM	۷۰.۸	۸۵.۸	۹۵.۳

Yale B Database<sup>۱۷</sup>  
SRC<sup>۱۸</sup>  
NN<sup>۱۹</sup>  
Nearest Space (NS)<sup>۲۰</sup>  
Linear SVM<sup>۲۱</sup>



شکل ۱: نتیجه اعمال روش‌های مختلف استخراج ویژگی بر عملکرد الگوریتم و مقایسه آن با الگوریتم‌های دیگر [۴].

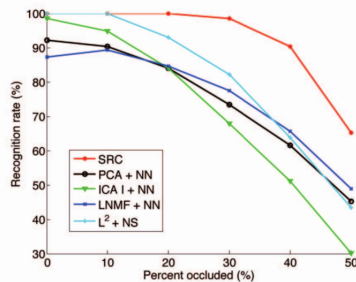
## ۵.۲ تاثیر خرابی تصویر و پوشش بخشی از چهره

در این آزمایش، میزان تاثیر خرابی بخشی از تصویر در عملکرد دسته‌بندی مورد بررسی قرار می‌گیرد. در مرحله اول از این آزمایش، نویز به بخشی از پیکسل‌های تصویر اعمال می‌شود و در مرحله دوم یک قطعه از تصویر به طور یکپارچه تخریب شده و عملکرد الگوریتم ارزیابی می‌گردد. جدول ۳ نتایج عملکرد الگوریتم را زمانی که درصدی از پیکسل‌های تصویر نویزی می‌شوند نمایش می‌دهد.

همین‌طور شکل ۳ عملکرد الگوریتم‌های مختلف را بر اساس درصد تخریب تصاویر مورد بررسی قرار می‌دهد. همان‌طور که مشاهده می‌شود الگوریتم اس‌آرسی مقاومت



شکل ۲: نمونه‌ای از استخراج سه ویژگی برای یک نمونه از تصاویر [۴]



شکل ۴: مقایسه میزان مقاومت الگوریتم‌ها در برابر تخریب قطعه‌ای از تصاویر [۴]

### ۳ دسته‌بندی کننده مبتنی بر بازنمایی تنک با تابع هسته

همان‌طور که قبلاً ذکر شد، دسته‌بندی‌کننده مبتنی بر بازنمایی تنک که در پژوهش [۴] ارائه شده است، هر تصویر تست را با استفاده از یک ترکیب خطی از داده‌های کلاس‌های موجود بازنمایی کرده و تنک کردن این ترکیب خطی قصد دارد تا شرایطی را فراهم آورد که در آن ضرایب مربوط به داده‌های موجود در نزدیک‌ترین کلاس موجود به تصویر ورودی غیر صفر و مابقی ضرایب صفر شوند. با توجه به ایده اصلی روش، می‌توان دریافت که در صورتی که داده‌های موجود در مجموعه داده آموزشی در دو کلاس مختلف، روی یک بردار جهت یکسان توزیع شده باشند، به طوری که بتوان هر یک از داده‌های موجود در یک کلاس را با مقیاس کردن یکی از داده‌های موجود در کلاس دیگر به دست آورد، دسته‌بندی‌کننده مبتنی بر بازنمایی تنک، قدرت تفکیک کردن داده‌های این دو کلاس را نخواهد داشت.

برای رفع این مشکل، می‌توان از ایده توابع هسته<sup>۲۲</sup> در یادگیری ماشین استفاده نمود. یک تابع هسته را می‌توان با توجه به قضیه مرسر<sup>۲۳</sup> به شکل رابطه (؟؟) نمایش داد.

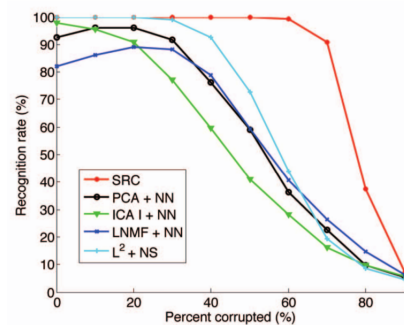
$$k(x, x') = \Phi(x)^T \phi(x') \quad (12)$$

Kernel Functions<sup>۲۲</sup>  
Mercer's Theorem<sup>۲۳</sup>

جدول ۲: عملکرد الگوریتم SRC با نویزی کردن درصدی از پیکسل‌های هر تصویر [۴]

درصد تخریب	دقت	درصد تخریب	دقت
۰	۱۰۰	۵۰	۱۰۰
۱۰	۱۰۰	۶۰	۹۹.۳
۲۰	۱۰۰	۷۰	۹۰.۷
۳۰	۱۰۰	۸۰	۳۷.۵
۴۰	۱۰۰	۹۰	۷.۱

بسیار بالایی نسبت به تخریب تصاویر در مقایسه با روش‌های دیگر از خود نشان می‌دهد.



شکل ۳: مقایسه میزان مقاومت الگوریتم‌ها در برابر تخریب تصاویر [۴]

به علاوه جدول؟؟ دقت الگوریتم را در برابر خرابی قطعه‌ای از تصویر گزارش می‌کند.

جدول ۳: عملکرد الگوریتم SRC با نویزی کردن قطعه‌ای از هر تصویر [۴]

تخریب	۱۰	۲۰	۳۰	۴۰	۵۰
دقت	۱۰۰	۹۹.۸	۹۸.۵	۹۰.۳	۶۵.۳

شکل ۴ عملکرد الگوریتم را در برابر تخریب قطعه‌ای از تصویر در مقایسه با روش‌های دیگر نمایش می‌دهد که مطابق با آنچه در این شکل به نمایش درآمده، مقاومت الگوریتم ارائه شده به مراتب نسبت به الگوریتم‌های دیگر در برابر تخریب قطعه‌ای از تصویر بیشتر است.



همین طور می دانیم  $m \gg D$  تعداد ابعاد فضای ویژگی ها را مشخص می نماید.

با استفاده از ایده اصلی در روش دسته بندی کننده مبتنی بر بازنمایی تنک که در آن هر تصویر ورودی را می توان با ترکیب خطی از تصاویر آموزشی موجود بازنمایی کرد، در این جا هم تصاویر ورودی را می توانیم بر اساس یک ترکیب خطی از نگاشت تصاویر آموزشی در فضای ویژگی ها بازنمایی بنماییم. این بازنمایی مطابق با رابطه (۱۴) انجام می شود.

$$\Phi(y) = \sum_{i=1}^n \alpha_i \Phi(x_i) = \Phi \alpha \quad (14)$$

مطابق با روش دسته بندی کننده مبتنی بر بازنمایی تنک، می توان یک مساله بهینه سازی ایجاد کرد که در آن قید تساوی، رابطه (۱۴) باشد. این مساله را با توجه به تمام توضیحاتی که در روش قبل به طور مفصل ارائه شد، از نوع یک مساله بهینه سازی نرم ۱ انتخاب می نماییم. بنابراین، مساله بهینه سازی (۱۵) می تواند یک بازنمایی مطلوب به ما ارائه دهد.

$$\begin{aligned} & \text{minimize } \|\alpha\|_1 \\ & \text{subject to } \Phi(y) = \Phi \alpha \end{aligned} \quad (15)$$

از آن جا که ابعاد فضای ویژگی ها در مساله (۱۵) بسیار بزرگ تر از ابعاد فضای داده است و همین طور مدل ارائه شده در (۱۵) در فضای ویژگی ها تنک نیست، نمی توان به راحتی این مساله را حل نمود. برای حل این مشکلات و یافتن مقدار بهینه این مساله، از روش کاهش بعد با استفاده از توابع هسته استفاده می نماییم و این مساله را به یک مساله بهینه سازی فضای ممکن<sup>۲۴</sup> تبدیل می نماییم و اقدام به حل آن می کنیم.

### ۲.۱.۳ کاهش بعد با توابع هسته

برای اعمال کاهش بعد از فضای ویژگی ها، باید یک ماتریس انتقال تعریف نماییم که با ضرب آن در داده های موجود در فضای ویژگی ها، آن ها را به یک فضا با بعد کوچک تر نگاشت نماییم. فرض کنیم ماتریس  $P \in R^{D \times d}$  ماتریس انتقال

که در آن، تابع  $\Phi$  یک نگاشت غیرخطی و  $x$  و  $x'$  دو نمونه از داده های موجود در مجموعه داده هستند. توابع مختلفی می توانند با هر تابع هسته مورد استفاده قرار بگیرند. به عنوان مثال می توان از تابع گاوسی به عنوان غیرخطی کننده تابع هسته استفاده کرد. همین طور استفاده از شکل های مختلف توابع هسته امکان پذیر است.

استفاده از توابع هسته می تواند با غیرخطی سازی داده ها و انتقال آن ها به ابعاد بالاتر باعث افزایش تفکیک پذیری داده ها شده و مشکل روش دسته بندی کننده مبتنی بر بازنمایی تنک را از بین ببرد.

در این بخش ابتدا تئوری روشی را که توسط ژنگ و همکارانش در [۱] در سال ۲۰۱۲ ارائه شده است، مورد بررسی قرار داده و سپس به بیان الگوریتم روش ترکیبی و آزمایشات انجام شده آن می پردازیم.

## ۱.۳ تئوری

در این روش، سه مرحله وجود دارد که هر یک را به طور جداگانه بررسی می نماییم. تصویر ورودی ابتدا در مرحله اول توسط یک تابع هسته از فضای داده به فضای هسته که یک فضا با بعد بالاتر است، نگاشت می شود. سپس با به کارگیری یک روش مناسب کاهش بعد، فضای هسته به یک فضا با بعد کوچکتر کاهش داده می شود. در فضای جدید داده ها از تفکیک پذیری بالاتری نسبت به فضای اولیه برخوردار خواهند بود. سپس عمل دسته بندی در فضای جدید انجام می شود و برچسب مناسب به تصویر ورودی اختصاص می یابد.

### ۱.۱.۳ انتقال به فضای هسته

با فرض این که مساله دارای  $c$  کلاس مختلف باشد و تابع غیرخطی کننده هسته را با  $\Phi$  نمایش دهیم، می توانیم برای افزایش تفکیک پذیری داده ها از یک دیگر، تابع غیرخطی کننده هسته را به داده های اعمال کرده و داده ها را به فضای ویژگی ها  $F$  نگاشت کنیم. این نگاشت مطابق با رابطه (۱۳) انجام می شود.

(۱۳)

$$\Phi : x \in \Psi \rightarrow \Phi(x) = [\Phi_1(x), \dots, \Phi_D(x)]^T$$

Feasible Optimization Problem<sup>۲۴</sup>

مناسبی باشد، آن گاه برای تبدیل داده‌ها مطابق با رابطه (۱۶) عمل می‌نماییم.

$$P^T \Phi(y) = P^T \Phi \alpha \quad (16)$$

برای یافتن ماتریس انتقال  $P$  با اقتباس از روش‌های  $KPCA$  و  $KFDA$  از یک ترکیب خطی از داده‌های موجود در مجموعه داده استفاده می‌نماییم. این ماتریس را می‌توانیم مطابق با رابطه (۱۷) محاسبه نماییم.

$$P_j = \sum_{i=1}^n \beta_{j,i} \Phi(x_i) = \Phi \beta_j \quad (17)$$

که در آن  $P_j$  ستون  $j$ ام از ماتریس انتقال  $P$  و  $\beta_j = [\beta_{j,1}, \dots, \beta_{j,n}]^T$  ضرایب ترکیب خطی هستند. بردار  $\beta_j$  را بردار شبه انتقال<sup>۲۵</sup> مربوط به بردار انتقال  $j$ ام از ماتریس  $P$  می‌نامند. اگر بخواهیم رابطه (۱۷) را در حالت ماتریسی بیان کنیم می‌توانیم از رابطه (۱۸) استفاده نماییم.

$$P = \Phi B \quad (18)$$

که در آن  $B = [\beta_1, \dots, \beta_d]^T$  است. اکنون می‌توانیم با جای‌گذاری رابطه (۱۸) در (۱۶) قید تساری را به شکل رابطه (۱۹) بازنویسی نماییم.

$$B^T k(., y) = B^T K \alpha \quad (19)$$

که در آن،  $k(., y) = [k(x_1, y), \dots, k(x_n, y)]^T$  و  $K = \Phi^T \Phi$  است. به عبارت دیگر فاصله تحت تابع هسته تمام داده‌های موجود در مجموعه آموزشی را با تصویر ورودی، در سمت چپ و فاصله تحت تابع هسته تمام داده‌های آموزشی با یک‌دیگر را در سمت راست قرار می‌دهیم.

با توجه به این نکته که با داشتن تابع غیرخطی‌کننده هسته،  $k(., y)$  و  $K$  هر دو قابل محاسبه هستند، مساله (۱۵) که قید تساوی آن با (۱۹) جای‌گزین شده باشد، قابل حل

می‌شود. در این‌جا ماتریس  $B$  نیز به مجموعه متغیرهای مساله اضافه می‌شود.

برای محاسبه  $B$  می‌توان از یکی از چهار روش زیر بهره گرفت:

۱. استفاده از بردارهای ویژه<sup>۲۶</sup> مربوط به بزرگ‌ترین مقادیر ویژه<sup>۲۷</sup> کاملاً شبیه آن‌چه در روش  $KPCA$  از آن استفاده می‌شود. در این روش با محاسبه مقادیر ویژه و انتخاب  $d$  مقدار ویژه بزرگ‌تر، بردارهای ویژه مربوط به هریک از مقادیر ویژه انتخاب شده را محاسبه کرده و از کنار هم قرار دادن آن‌ها در ماتریس  $B$  به عنوان ماتریس انتقال استفاده می‌نماییم. در این حالت در اصل با استفاده از روش  $KPCA$  بعد فضای ویژگی را کاهش می‌دهیم.

۲. با استفاده از روش  $KFDA$  می‌توانیم بعد فضای حالت را کاهش دهیم. در این حالت، ماتریس انتقال  $B$  پاسخ مساله (۲۰) است.

$$\max_B \frac{\text{tr}(B^T S_b^K B)}{\text{tr}(B^T S_w^K B)} \quad (20)$$

۳. مطابق با آن‌چه در روش دسته‌بندی‌کننده مبتنی بر بازنمایی تنک ارائه شد، می‌توانیم از یک ماتریس  $B$  کاملاً تصادفی برای کاهش بعد استفاده نماییم زیرا ماتریس تصادفی نیز اطلاعات کافی برای یافتن پاسخ بهینه مساله بهینه‌سازی نرم ۱ با تنک‌ترین پاسخ را دارد.

۴. همین‌طور می‌توان از یک ماتریس همانی به جای ماتریس  $B$  استفاده کرد که در این حالت، کاهش بعدی انجام نمی‌شود.

با استفاده از هر یک از چهار روش فوق می‌توان ابعاد فضای ویژگی را کاهش داد.

Eigenvectors<sup>۲۶</sup>  
Eigenvalues<sup>۲۷</sup>

Pseudo Transformation Vector<sup>۲۵</sup>

### ۳.۱.۳ دسته‌بندی

پس از کاهش بعد فضای ویژگی‌ها می‌توانیم اقدام به دسته‌بندی داده‌های ورودی و تست نماییم. مساله بهینه‌سازی که باید حل شود، مطابق با توضیحاتی که ارائه شد، به شکل (۲۱) قابل بیان است.

$$\begin{aligned} & \text{minimize } \|\alpha\|_1 \\ & \text{subject to } B^T k(\cdot, y) = B^T K \alpha \end{aligned} \quad (21)$$

**Data:** A set of training samples

$\{x_i, l_i\}_{i=1}^n$ , where

$l_i \in \{1, 2, \dots, c\}$ , a test sample

$y \in R^m$  and an optimal error tolerance  $\epsilon > 0$

با احتساب نویز برای داده‌ها مطابق با روش قبلی، می‌توانیم قید تساوی مساله فوق را تغییر داده و مساله (۲۲) را حل نماییم.

1. Select a kernel  $k(\cdot, \cdot)$  and its parameters.

$$\text{minimize } \|\alpha\|_1$$

2. Compute matrix  $K = \Phi^T \Phi$  and  $k(\cdot, y)$ .

$$\text{subject to } \|B^T k(\cdot, y) - B^T K \alpha\|_2 < \epsilon \quad (22)$$

3. Select a dimensionality reduction method and compute appropriate projection matrix  $B$ .

روند دسته‌بندی در این روش نیز کاملاً مطابق با روند دسته‌بندی در روش SRC است که قبلاً به طور مبسوط شرح داده شد.

4. Normalize the columns of  $B^T K$  and  $B^T k(\cdot, y)$  to have unit  $l^2 - norm$ .

### ۲.۳ الگوریتم

5. Solve the problem in (22).

6. Compute reconstruction errors  $r_i(y) = \|B^T k(\cdot, y) - B^T K \alpha_i\|_2$

الگوریتم دسته‌بندی‌کننده ارائه شده در [۱] در ۲ قابل مشاهده است. همان‌طور که مشاهده می‌شود تمام مراحل این الگوریتم کاملاً با مراحل الگوریتم دسته‌بندی‌کننده مبتنی بر بازنمایی تنک یکسان است به جز قسمت‌هایی که مربوط به نگاشت اولیه داده‌ها به فضای ویژگی و سپس کاهش بعد می‌شود. تمام مراحل مطابق با آنچه توضیح داده شده است انجام می‌پذیرد.

**Result:**  $\text{identity}(y) = \text{argmin}_i r_i(y)$

**Algorithm 2:** الگوریتم دسته‌بندی با استفاده از [۱] توابع هسته دسته‌بندی‌کننده مبتنی بر بازنمایی تنک ترکیب شده با

### ۳.۳ آزمایشات

#### ۴ یادگیری وزن‌های تابع هسته

#### ۵ استفاده از تطبیق هرم مکانی

#### ۶ جمع‌بندی و نتیجه‌گیری

## مراجع

- [1] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, and F.-Z. Li, "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol.60, no.4, pp.1684–1695, 2012.
- [2] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Transactions on Image Processing*, vol.22, no.2, pp.423–434, 2013.
- [3] A. Shrivastava, V. M. Patel, and R. Chellappa, "Multiple kernel learning for sparse representation-based classification," *IEEE Transactions on Image Processing*, vol.23, no.7, pp.3013–3024, 2014.
- [4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol.31, no.2, pp.210–227, 2009.