



پارامتر بهینه K که در FKNNC و KNNC وجود داشت حل می‌گردد.

## ۲- مروری بر کارهای مرتبط

### ۱-۲- طبقه‌بند SRC

در SRC هر نمونه تست به صورت ترکیب خطی تنک از همه نمونه‌های یادگیری بیان می‌شود و انتظار می‌رود که عناصر غیر صفر بردار ضرایب، متعلق به نمونه‌های یک کلاس خاص باشند.

تئوری SR در واقع راه حلی برای یافتن بردار تنک ضرایب است. این موضوع به صورت زیر در نظر گرفته می‌شود:

$$\hat{\alpha}_0 = \arg \min_{\alpha} \|\alpha\|_0 \quad s.t. \quad y = D\alpha \quad (1)$$

که  $\|\cdot\|_0$  بیانگر  $\ell_0$ -نرم بردار  $\alpha$  است که تعداد عناصر غیر صفر  $\alpha$  را بدست می‌دهد. اما از آن‌جا که  $\|\cdot\|_p$  برای  $0 < p < 1$  محدب نبوده و از همه مهم‌تر، مشتق‌پذیر نیست، حل معادله (۱) به‌طور کلی NP-hard است [۱۳].

از آن‌جا که نرم یک ( $\|\cdot\|_1$ )، نزدیکترین نرم محدب به نرم صفر است، جایگزینی نرم صفر با نرم یک، منطقی‌ترین تقریب به نظر می‌رسد و معادله (۱) را می‌توان به فرم زیر بازنویسی کرد:

$$\hat{\alpha}_1 = \arg \min_{\alpha} \|\alpha\|_1 \quad s.t. \quad y = D\alpha \quad (2)$$

یا:

$$\hat{\alpha}_1 = \arg \min_{\alpha} \|D\alpha - y\|_2 + \lambda \|\alpha\|_1 \quad (3)$$

که  $\|\cdot\|_1$  بیانگر اپراتور نرم یک،  $\|\cdot\|_2$  بیانگر اپراتور نرم ۲،  $\lambda$  پارامتر تنظیم میزان تنکی بردار ضرایب  $\hat{\alpha}_1$  و ماتریس D بیانگر ماتریس دیکشنری شامل تمامی نمونه‌های یادگیری است. الگوریتم‌های زیادی برای حل معادله (۳) وجود دارد [۱۴-۱۶].

فرض کنید که  $\hat{\alpha}_1$  حل حاصل از مسئله بهینه‌سازی (۳) باشد، سپس هر نمونه تست توسط بردار ضرایب و نمونه‌های یادگیری مربوط به هر کلاس بازسازی می‌شود و نمونه تست متعلق به کلاسی خواهد بود که نمونه بازسازی شده‌ی توسط آن کلاس، کمترین باقی‌مانده را با خود نمونه تست داشته باشد.

این طبقه‌بند قادر به طبقه‌بندی ویژگی‌های غیرخطی و داده‌های با توزیع جهتی یکسان نیست. از این رو در مرجع [۱۷] نسخه کرنلی این طبقه‌بند با نام KSRC معرفی شده است. که نتایج بهتری را نسبت به SRC بدست می‌دهد.

### ۲-۲- طبقه‌بند FSNC

اخیراً طبقه‌بند جدیدی با نام FSNC [۱۲] معرفی شده است. در FSNC از الگوریتمی به نام NSR استفاده می‌شود. ضرایب بدست آمده از NSR میزان شباهت یک نمونه را به سایر نمونه‌ها نشان می‌دهد به‌طوری‌که ضرایب بزرگتر نشانگر شباهت بیشتر می‌باشند. الگوریتم FSNC را می‌توان در سه گام خلاصه کرد. در گام اول به کمک NSR، همسایه‌های مشابه هر کدام از نمونه‌های یادگیری تعیین می‌شوند. در گام دوم به‌منظور این‌که همه همسایه‌ها با وزن یکسانی در نظر گرفته نشوند و دوری یا نزدیکی همسایه‌ها لحاظ شود، از تئوری مجموعه فازی استفاده می‌شود. در گام سوم ابتدا با به‌کار بردن الگوریتم NSR بر روی نمونه تست  $y$ ، همسایه‌های مشابه نمونه تست تعیین می‌شوند. سپس با محاسبه اعضای تصمیم‌گیری فازی نمونه‌ی تست، طبقه‌بندی انجام می‌شود.

### ۳- الگوریتم پیشنهادی

با وجود برتری‌هایی که در FSNC وجود دارد، این روش در شناسایی همسایگی‌های غیرخطی و داده‌های با توزیع جهتی یکسان دچار مشکل می‌شود. برای حل این مشکل ما با تعمیم FSNC به حوزه کرنل، طبقه‌بند جدیدی با نام Kernel FSNC (KFSNC) ارائه می‌کنیم. اما در حل الگوریتم SR در حوزه کرنل، نیازمند معلوم بودن نگاهت غیرخطی می‌باشیم که معمولاً مجهول است در این قسمت از مقاله پس از ارائه یک روش حل برای SR در حوزه کرنل، الگوریتم پیشنهادی خود را ارائه می‌کنیم.

### ۳-۱- حل SR در حوزه کرنل

برای حل الگوریتم SR در حوزه‌ی کرنل، معادله‌ی (۱) به صورت زیر در نظر گرفته می‌شود:

$$\hat{\alpha}_1^{\Phi} = \arg \min_{\alpha^{\Phi}} \|\alpha^{\Phi}\|_1 \quad s.t. \quad \Phi(y) = D^{\Phi} \alpha^{\Phi} \quad (4)$$

به علت مجهول بودن نگاهت غیرخطی  $\Phi$ ، معادله (۴) قابل حل نمی‌باشد. برای حل این مشکل، قید معادله (۴) را به صورت زیر در نظر می‌گیریم:

$$D^{\Phi T} \Phi(y) = D^{\Phi T} D^{\Phi} \alpha^{\Phi} \quad (5)$$

به عبارت دیگر طرفین قید را در ماتریس  $D^{\Phi T}$  ضرب می‌کنیم. طرفین رابطه (۵) به صورت زیر قابل محاسبه هستند:

است. در نتیجه تعداد همسایه‌ها و درجه شباهت آن‌ها به طور  
 وقتی و به کمک الگوریتم NSR تعیین می‌شود.

**گام دوم (محاسبه ماتریس تعلق کلاس فازی):** به منظور اینکه همه همسایه‌ها با وزن یکسانی در نظر گرفته نشوند و دوری یا نزدیکی همسایه‌ها لحاظ شود، مشابه FSNC در گام دوم از تئوری مجموعه فازی استفاده می‌کنیم. ماتریس تعلق کلاس فازی  $U^\Phi = \{u_{ji}^\Phi\}_{n \times c}$  را که در شرایط زیر صدق می‌کند، تعریف می‌کنیم:

$$\sum_{i=1}^c u_{ji}^\Phi = 1 \text{ and } 0 < \sum_{j=1}^n u_{ji}^\Phi < n \quad (10)$$

عناصر ماتریس تعلق کلاس فازی  $U^\Phi$  را به صورت زیر تعریف می‌کنیم:

$$u_{ji}^\Phi = \begin{cases} 0.51 + 0.49 \frac{\xi_i(\hat{\beta}_j^\Phi)}{\xi(\hat{\beta}_j^\Phi)} & ; \text{ if } i = \text{lable of the } j\text{th sample} \\ 0.49 \frac{\xi_i(\hat{\beta}_j^\Phi)}{\xi(\hat{\beta}_j^\Phi)} & ; \text{ otherwise} \end{cases} \quad (11)$$

که  $\xi(\hat{\beta}_j^\Phi)$  جمع همه عناصر بردار  $\hat{\beta}_j^\Phi$  مربوط به  $j$ -امین نمونه است و  $\xi_i(\hat{\beta}_j^\Phi)$  جمع عناصری از بردار  $\hat{\beta}_j^\Phi$  است که مربوط به نمونه‌های یادگیری کلاس  $i$ -ام هستند.

**گام سوم (محاسبه همسایه‌های مشابه نمونه تست در حوزه کرنل):** در این مرحله با به کار بردن الگوریتم NSR در حوزه کرنل، همسایه‌های مشابه نمونه تست  $y$  تعیین می‌شوند. برای این کار الگوریتم NSR به صورت زیر در نظر گرفته می‌شود:

$$\hat{\alpha}^\Phi = \arg \min_{\alpha^\Phi} \|D^{\Phi^T} D^\Phi \alpha^\Phi - D^{\Phi^T} \Phi(y)\|_2 + \lambda \|\alpha^\Phi\|_1 \quad (12)$$

s.t.  $\alpha_j^\Phi \geq 0, j=1, 2, \dots, n$

که  $\hat{\alpha}^\Phi = [\alpha_1, \dots, \alpha_n] \in \mathcal{R}^n$  است. مشابه حالت قبل می‌توان برای نمونه تست  $y$ ، مجموعه  $N\{\Phi(y)\} \in \mathcal{R}^{L \times k}$  را تعریف کرد که شامل تمامی همسایه‌های مشابه آن بوده و  $k$  تعداد این همسایه‌ها باشد.

**گام چهارم (محاسبه اعضای تصمیم‌گیری فازی نمونه تست):** اعضای تصمیم‌گیری فازی نمونه تست به صورت زیر تعریف می‌شوند:

$$D^{\Phi^T} \Phi(y) = [\Phi(x_{1,1}), \Phi(x_{2,1}), \dots, \Phi(x_{n_c,c})]^T \times \Phi(y) = \begin{bmatrix} K(x_{1,1}, y) \\ K(x_{2,1}, y) \\ \vdots \\ K(x_{n_c,c}, y) \end{bmatrix} \quad (6)$$

و همچنین:

$$D^{\Phi^T} D^\Phi \alpha^\Phi = [\Phi(x_{1,1}), \Phi(x_{2,1}), \dots, \Phi(x_{n_c,c})] \times [\Phi(x_{1,1}), \Phi(x_{2,1}), \dots, \Phi(x_{n_c,c})]^T \times \Phi(y) = \begin{bmatrix} K(x_{1,1}, x_{1,1}) & K(x_{1,1}, x_{2,1}) & \dots & K(x_{1,1}, x_{n_c,c}) \\ K(x_{2,1}, x_{1,1}) & K(x_{2,1}, x_{2,1}) & \dots & K(x_{2,1}, x_{n_c,c}) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_{n_c,c}, x_{1,1}) & K(x_{n_c,c}, x_{2,1}) & \dots & K(x_{n_c,c}, x_{n_c,c}) \end{bmatrix} \quad (7)$$

در نتیجه با معلوم بودن تابع کرنل  $K$  (که معمولاً معلوم است)، الگوریتم SR در حوزه کرنل قابل حل می‌باشد:

$$\hat{\alpha}^\Phi = \arg \min_{\alpha^\Phi} \|D^{\Phi^T} D^\Phi \alpha^\Phi - D^{\Phi^T} \Phi(y)\|_2 + \lambda \|\alpha^\Phi\|_1 \quad (8)$$

### ۳-۲- طبقه‌بند KFSNC

الگوریتم پیشنهادی ما در چهار گام اجرا می‌شود.

**گام اول (محاسبه همسایه‌های مشابه هر نمونه یادگیری در حوزه کرنل):** در اولین گام به کمک حل الگوریتم NSR در حوزه کرنل، همسایه‌های مشابه هر کدام از نمونه‌های یادگیری تعیین می‌گردند. برای این کار معادله (۳) در حوزه کرنل به صورت زیر در نظر گرفته می‌شود:

$$\hat{\beta}_j^\Phi = \arg \min_{\beta_j^\Phi} \left\| \tilde{D}^{\Phi^T} \tilde{D}^\Phi \beta_j^\Phi - \tilde{D}^{\Phi^T} \Phi(x_j) \right\|_2 + \lambda \|\beta_j^\Phi\|_1 \quad (9)$$

s.t.  $\beta_{j,l}^\Phi \geq 0 \quad l=1, 2, \dots, n-1, j=1, 2, \dots, n$

که  $\hat{\beta}_j^\Phi = [\beta_{j,1}, \dots, \beta_{j,n-1}] \in \mathcal{R}^{n-1}$  و  $\Phi(x_j)$  به ترتیب نمونه یادگیری  $j$ -ام و بردار وزن مربوط به آن در حوزه کرنل است. ماتریس  $\tilde{D}^\Phi$  بیانگر ماتریس دیکشنری شامل تمامی نمونه‌های یادگیری، پس از نگاشت به فضای ویژگی با بعد بالا می‌باشد که نمونه یادگیری  $j$ -ام از آن حذف شده است.

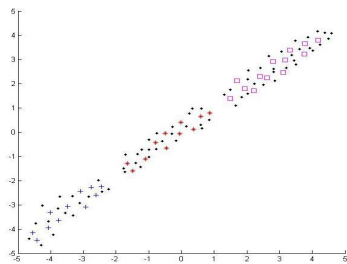
برای هر نمونه یادگیری  $\Phi(x_j)$  در حوزه کرنل، مجموعه  $N\{\Phi(x_j)\} \in \mathcal{R}^{L \times k'_j}$  را تعریف می‌کنیم که شامل تمامی همسایه‌های مشابه  $\Phi(x_j)$  بوده و  $k'_j$  تعداد این همسایه‌ها

که بزرگتر از  $10^{-3}$  بوده‌اند به عنوان عناصر غیر صفر در نظر گرفته‌ایم.

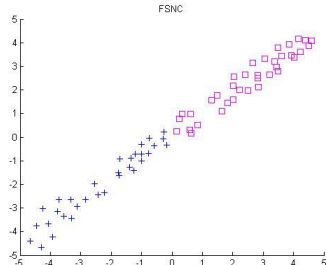
$$v_i \{ \Phi(y) \} = \frac{\sum_{j \in N\{\Phi(y)\}} u_{ji}^{\Phi} \alpha_j}{\sum_{j \in N\{\Phi(y)\}} \alpha_j} \quad i=1,2,\dots,c \quad (13)$$

#### ۴-۱- آزمایش بر روی یک نمونه از داده‌های با توزیع جهتی یکسان

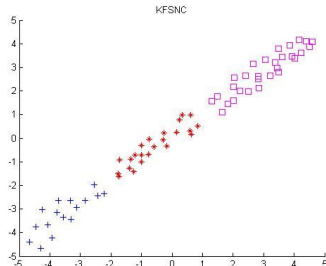
در شکل ۱-الف) نمونه‌ای از داده‌های با توزیع جهتی یکسان نشان داده شده‌است. شکل ۱-ب) و ۱-ج) به ترتیب نتایج حاصل از طبقه‌بندی نمونه‌ای تست را به کمک دو الگوریتم FSNC و KFSNC نشان می‌دهند.



(الف)



(ب)



(ج)

شکل ۶: طبقه‌بندی داده‌های با توزیع جهتی یکسان : در شکل(الف) نمونه‌های سه کلاس نمایش داده شده‌اند. نمونه‌های تست توسط " " مشخص شده‌اند. شکل(ب) و (ج) به ترتیب نتیجه طبقه‌بندی نمونه‌های تست به کمک الگوریتم‌های FSNC و KFSNC است.

نتیجه این آزمایش نشان می‌دهد که استفاده از رهیافت کرنل سبب می‌شود تا الگوریتم KFSNC در طبقه‌بندی داده‌های با توزیع جهتی یکسان، کارایی خوبی داشته باشد و این در حالی است که الگوریتم FSNC در طبقه‌بندی داده‌های با توزیع جهتی

در رابطه بالا،  $\alpha_j$  بیانگر میزان شباهت همسایه‌های نمونه تست و  $u_{ji}^{\Phi}$  میزان تعلق این همسایه‌ها را در حوزه کرنل به  $c$  کلاس مشخص می‌کنند. بعد از محاسبه  $v_i \{ \Phi(y) \}$  برای  $c$  کلاس، طبقه‌بندی به صورت زیر انجام می‌شود:

$$\max_i v_i \{ \Phi(y) \} = \text{کلاس نمونه تست} \quad (14)$$

#### ۳-۳- طبقه‌بند Fast-KFSNC

هنگامی که طبقه‌بندی داده‌های با تعداد ویژگی زیاد مانند تصاویر مد نظر است، الگوریتم‌های FSNC و KFSNC بسیار کند خواهند بود. این موضوع سبب می‌شود تا یافتن پارامترهای بهینه برای این الگوریتم‌ها مشکل باشد. برای حل این مشکل، در این قسمت از مقاله نسخه سریعی از KFSNC را با نام Fast-KFSNC ارائه می‌دهیم. بیشترین زمان اجرای الگوریتم‌های مبتنی بر بیان تنک در حل الگوریتم SR صرف می‌شود، در نتیجه استفاده از یک الگوریتم SR سریع، باعث افزایش سرعت طبقه‌بندی‌های مبتنی بر SR می‌گردد. الگوریتم OMP [۱۵] یکی از الگوریتم‌های سریع حل SR می‌باشد. ما برای افزایش سرعت طبقه‌بندی داده‌ها از این الگوریتم استفاده می‌کنیم. روند کلی الگوریتم Fast-KFSNC مشابه الگوریتم KFSNC می‌باشد با این تفاوت که پس از بدست آمدن ضرایب SR به کمک الگوریتم OMP، ضرایبی را که منفی هستند، صفر در نظر می‌گیریم. در نتیجه با استفاده از OMP، افزایش سرعت قابل ملاحظه‌ای در طبقه‌بندی داده‌ها ایجاد می‌شود.

#### ۴- آزمایشات

در این قسمت از مقاله، کارایی الگوریتم KFSNC را مورد ارزیابی قرار می‌دهیم. آزمایشات بر روی دیتاست‌های Wine و Iris از UCI [۱۸] و دیتابیس‌های تصویر ORL [۱۹]، UMIST [۲۰] و Extended Yale B (B+) [۲۱] انجام شده است. ما از PAC [۲۲] به منظور کاهش ویژگی و از توابع متلب "OMP" [۲۳] و "11-ls-nonneg" [۱۶] به ترتیب برای محاسبه ضرایب SR و NSR استفاده می‌کنیم. همچنین از کرنل معروف RBF به منظور نگاشت نمونه‌ها به فضای ویژگی با بعد بالا، استفاده شده است و ما ضرایب بردار بهینه بدست آمده از الگوریتم NSR را

یکسان دچار مشکل می‌شود. این موضوع بدین علت است که توزیع نمونه‌ها در الگوریتم‌های مبتنی بر حوزه کرنل، متفاوت با حوزه ورودی می‌باشد.

#### ۴-۲- آزمایش بر روی دیتاست‌های Iris و Wine از UCI

برای دیتاست‌های Iris و Wine، به‌طور تصادفی ۱۰ درصد از داده‌های هر کلاس برای یادگیری، ۴۰ درصد از داده‌های هر کلاس برای تعیین پارامتر و مابقی برای تست انتخاب شده‌اند. برنامه ۱۰ بار اجرا شده است. جدول (۱) نتایج حاصل از این آزمایش را هنگامی که کرنل گوسی به کار رفته است، نشان می‌دهد. از جدول (۱) می‌توان دید که برای دیتاست Iris، الگوریتم SRC بهتر از FSNC و KSRC بهتر از SRC است و KFSNC بهتر از هر سه الگوریتم است. همچنین می‌توان دید که برای دیتاست Wine کارایی FSNC بهتر از SRC و KSRC بهتر از FSNC است و الگوریتم KFSNC دارای دقت طبقه‌بندی بالاتری نسبت به KSRC است.

جدول ۱: میانگین و انحراف معیار نرخ شناسایی (%) بر روی

دیتاست‌های Iris و Wine.

الگوریتم دیتاست	SRC	KSRC	FSNC	KFSNC
Iris	93.73±4.16	95.64±2.14	90.64±6.56	<b>96.58±2.69</b>
Wine	89.95±5.15	92.38±4.91	91.94±2.42	<b>93.14±2.83</b>

#### ۴-۳- آزمایش بر روی دیتابیس چهره ORL

به‌طور تصادفی ۳ تصویر از هر نفر برای یادگیری، ۳ تصویر از هر نفر برای تعیین پارامتر و ۴ تصویر از هر نفر برای تست انتخاب شده‌اند. آزمایشات بر روی تعداد ویژگی ۱۰، ۶۰، ۱۱۰ و ۱۶۰ انجام شده است. ما الگوریتم را ۱۰ بار اجرا کردیم. جدول (۲) میانگین نرخ شناسایی و انحراف معیار هر روش طبقه‌بندی را هنگامی که کرنل گوسی به کار رفته است، بیان می‌کند.

جدول ۲: میانگین و انحراف معیار نرخ شناسایی (%) بر حسب بعد، بر

روی دیتابیس چهره ORL.

الگوریتم بعد	SRC	KSRC	FSNC	Fast-KFSNC
10	80.63±4.92	82.53±5.39	78.73±5.26	<b>83.41±4.20</b>
60	88.20±3.84	88.84±3.41	89.31±2.81	<b>90.78±2.78</b>
110	89.01±4.01	89.77±3.70	90.26±2.09	<b>91.74±3.17</b>
160	90.07±2.93	90.78±2.24	91.42±2.85	<b>92.42±2.85</b>

از جدول (۲) نتایج زیر بدست می‌آید:

الف) صرف نظر از بعد نمونه‌ها و تحت شرایط یکسان، الگوریتم Fast-KFSNC بهترین کارکرد را دارد. ب) هنگامی که فضای ویژگی بعد بسیار کمی دارد، FSNC بدتر از سه روش دیگر است. ج) الگوریتم KSRC کارآمدتر از SRC است. د) از نکات اول و سوم این نکته استنتاج می‌شود که رهیافت کرنل قابلیت طبقه‌بندی را بهبود می‌دهد.

#### ۴-۴- آزمایش بر روی دیتابیس چهره UMIST

به‌طور تصادفی ۲۰ درصد تصاویر هر نفر برای یادگیری، ۳۰ درصد برای انتخاب پارامتر و مابقی تصاویر برای تست در نظر گرفته شده‌اند. آزمایشات بر روی زیر فضای ۱۰۰- بعدی PCA، ۱۰ بار تکرار شده است. به منظور ارزیابی سرعت الگوریتم‌ها، زمان تست مربوط به چهار الگوریتم SRC، KSRC، FSNC و Fast-KFSNC به همراه میانگین نرخ شناسایی و انحراف معیار حاصل از این آزمایش در جدول (۳) لیست شده است. با توجه به جدول (۳) نتایج زیر بدست می‌آید: الف) الگوریتم ما بهترین دقت طبقه‌بندی را در میان سه الگوریتم دیگر دارد. ب) الگوریتم SRC سریع‌ترین و الگوریتم KFSNC کندترین الگوریتم‌ها در میان چهار الگوریتم مطرح شده هستند. ج) الگوریتم Fast-KFSNC دارای نرخ دقت بالا و سرعت قابل قبول است.

جدول ۳: میانگین و انحراف معیار نرخ شناسایی (%) و زمان تست (ثانیه) مربوط به SRC، KSRC، FSNC و Fast-KFSNC بر روی

دیتابیس چهره UMIST

الگوریتم	SRC	KSRC	FSNC	Fast-KFSNC
نرخ طبقه‌بندی	85.33±4.03	85.75±3.76	86.07±4.17	<b>87.26±3.37</b>
زمان تست (ثانیه)	<b>0.71±0.03</b>	1.31±0.27	53.88±4.92	3.87±0.24

#### ۴-۵- آزمایش بر روی دیتابیس چهره Extended Yale B(B+)

از این دیتابیس یک گروه از تصاویر شامل ۹ وضعیت متفاوت از ۲۸ شخص (مجموعاً ۲۵۲ تصویر)، انتخاب کرده‌ایم. برای این گروه از تصاویر به‌طور تصادفی ۵ تصویر از هر نفر برای یادگیری، ۲ تصویر برای انتخاب پارامتر و مابقی تصاویر برای تست در نظر گرفته شده‌اند. آزمایشات بر روی زیر فضای ۱۰۰- بعدی PCA، ۱۰ بار تکرار شده است. نتایج حاصل از این آزمایش شامل میانگین و انحراف معیار نرخ شناسایی (%) و زمان تست مربوط به SRC، KSRC، FSNC و Fast-KFSNC در جدول (۴) آمده است.

## مراجع

- [1] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, pp. 972-982, 2010.
- [2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse Recognition for Computer Vision and Pattern Recognition," in *Proceedings of the IEEE*, Vol. 98, No. 6, pp. 1031-1044, 2010.
- [3] H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: Theory and applications," *Signal Processing* 93(6), pp. 1408-1425, 2013.
- [4] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, 52(4), pp. 1289-1306, 2006.
- [5] E. Candes, "Compressive sampling," *Proceedings of the International Congress of Mathematicians*, pp. 1433-1452, Madrid, Spain, 2006.
- [6] S. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica Journal* 31, pp. 249-268, 2007.
- [7] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2), pp. 210-227, 2009.
- [8] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.
- [9] J. M. Keller, M. R. Gray, and J. A. Givens, Jr. "A fuzzy K-Nearest Neighbor Algorithm," *IEEE Trans. on SMC*, Vol. SMC-15, No. 4, pp. 580-585, 1985.
- [10] H. L. Chen, B. Yang, G. Wang, J. Liu, X. Xu, S. J. Wang & D. Y. Liu, "A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1348-1359, 2011.
- [11] H. Bian, L. Mazlack, "Fuzzy-rough nearest-neighbor classification approach," in: *Proceeding of the 22nd International Conference of the North American Fuzzy Information Processing Society*, pp. 500-505, 2003.
- [12] J. Xu, J. Yang, "A nonnegative sparse representation based fuzzy similar neighbor classifier," *Neurocomputing*, Vol. 99, pp. 76-86, 2013.
- [13] E. Amaldi and V. Kann, "On the Approximation of Minimizing Non Zero Variables or Unsatisfied Relations in Linear Systems," *Theoretical Computer Science*, vol. 209, pp. 237-260, 1998.
- [14] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129-159, 2001.
- [15] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. on Information theory*, vol. 50, no. 10, pp. 2242-2231, 2004.
- [16] SJ Kim, K Koh, M Lustig, S Boyd, D Gorinevsky, "A method for large-scale  $\ell_1$ -regularized least squares problems with applications in signal processing and statistics," *IEEE J. Sel. Top. Signal Process* 1(4), pp.606-617, 2007.
- [17] J. Yin, Z. Liu, Z. Jin, W. Yang, "Kernel sparse representation based classification," *Neurocomputing*, vol. 77, pp. 120-128, 2012.
- [18] C. Blake, C. Merz, UCI Repository of Machine Learning Databases, 1998.
- [19] F.S. Samaria, A.C. Harter, "Parameterisation of a stochastic model for human face identification," *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pp.138-142, 1994.
- [20] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition Face Recognition: From Theory to Appl.," *NATO ASI Series F, Comput. Syst. Sci.*, vol. 163, pp. 446-456, 1998.
- [21] A. S. Georghiadis, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643-660, Jun. 2001.
- [22] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.
- [23] E. Candès and J. Romberg. L1-magic: [Online]. Available: <http://www.l1-magic.org>, 2007.

از جدول (۴) می‌توان دید که الگوریتم FSNC کارآمدتر از KSRC و KSRC کارآمدتر از SRC است و Fast-KFSNC کارآیی طبقه‌بندی بهتری نسبت به هر سه الگوریتم دیگر دارد. همچنین نتایج این آزمایش نشان می‌دهد که الگوریتم‌های SRC و FSNC به ترتیب کندترین و سریعترین الگوریتم‌ها هستند و الگوریتم ما به نسبت سه الگوریتم دیگر دارای نرخ طبقه‌بندی بالا و سرعت قابل قبول است.

جدول ۴: میانگین و انحراف معیار نرخ شناسایی (%) و زمان تست مربوط به SRC, KSRC, FSNC و Fast-KFSNC بر روی دیتابیس چهره

Extended Yale B (B+)

الگوریتم	SRC	KSRC	FSNC	Fast-KFSNC
نرخ طبقه‌بندی	57.67±2.31	69.64±3.35	73.55±5.10	<b>78.55±2.63</b>
زمان تست (ثانیه)	<b>0.24</b>	0.39	13.93	1.55

## ۵- نتیجه‌گیری

در این مقاله طبقه‌بند KFSNC که به عنوان ورژن توسعه یافته‌ی غیر خطی از FSNC می‌باشد، ارائه شده است. برای استفاده از رهیافت کرنل در الگوریتم KFSNC و به علت مجهول نگاشت غیرخطی، تغییراتی را در حل الگوریتم NSR اعمال کردیم و توانستیم به طور موفقیت آمیزی الگوریتممان را با استفاده از تابع کرنل پیاده سازی کنیم.

استفاده از یک نگاشت مناسب سبب می‌شود تا توزیع نمونه‌ها در فضای جدید تغییر کرده و در نتیجه داده‌هایی که در فضای ویژگی اولیه، جداناپذیر خطی و یا دارای توزیع جهتی یکسان هستند، در فضای ویژگی کرنلی با بعد بالا جداپذیر خطی شده و دیگر دارای توزیع جهتی یکسان نباشند.

نتایج آزمایش نشان می‌دهد که کارآیی طبقه‌بندی KFSNC بالاتر از دیگر الگوریتم‌های طبقه‌بندی مبتنی بر SR است. اما KFSNC همانند FSNC الگوریتمی کند در طبقه‌بندی داده‌های با تعداد ویژگی زیاد مانند تصاویر ارائه شده است. در این مقاله الگوریتم دیگری با نام Fast-KFSNC برای طبقه‌بندی داده‌های با تعداد ویژگی زیاد مانند تصاویر ارائه شده است. نتایج آزمایش بر روی دیتابیس‌های تصویر نشان می‌دهد که الگوریتم Fast-KFSNC نه تنها دقت طبقه‌بندی بالاتری نسبت به FSNC و دیگر الگوریتم‌های طبقه‌بندی مبتنی بر SR دارد بلکه به مراتب سریعتر از FSNC است. در آینده ما بر روی یافتن وفقی پارامتر کرنل تمرکز خواهیم کرد.