



۱۳۰۷

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده مهندسی نقشه برداری

پردازش تصاویر ابر طیفی

پروژه نهایی

## Gabor-Filtering-Based Nearest Regularized Subspace for Hyperspectral Image Classification

استاد درس:

دکتر مختارزاده

دانشجو:

آرش عظیمی فرد - کیمیا محمدنژاد

تابستان ۹۹

## فهرست

۴.....	مقدمه
۸.....	الگوریتم ها
۸.....	Gabor filter
۹.....	الگوریتم NRS
۱۱.....	الگوریتم ترکیبی NRS-Gabor
۱۳.....	پیاده سازی
۱۳.....	داده های مورد استفاده
۱۳.....	داده های Indian pines
۱۴.....	داده های Pavia University
۱۵.....	پیاده سازی
۱۵.....	بررسی مقادیر $\delta$ های مختلف و اثر آن بر دقت کلی
۱۷.....	بررسی مقادیر $\lambda$ های مختلف و اثر آن بر دقت کلی
۲۱.....	تصاویر طبقه بندی شده
۲۴.....	نتایج
۲۵.....	کد برنامه :

## فهرست اشکال :

- شکل ۱ : کرنل های دو بعدی Gabor با جهت های مختلف از بالا به پایین و چپ به راست  
۸..... 0,  $\pi 8$ ,  $\pi 4$ ,  $3\pi 8$ ,  $\pi 2$ ,  $5\pi 8$ ,  $3\pi 4$ ,  $7\pi 8$  زوایای
- شکل ۲ : فلوچارت الگوریتم اول - طبقه بندی NRS ..... ۱۱
- شکل ۳: فلوچارت الگوریتم دوم - Gabor-filtering-based NRS ..... ۱۲
- شکل ۴ : تصویر منطقه Indian pines و کلاس های موجود ..... ۱۳
- شکل ۵ : تصویر منطقه Pavia University ..... ۱۴
- شکل ۶ : نمودار دقت کلی بر حسب تغییرات پارامتر  $\delta$  برای bw های ۱ و ۳ و ۵ داده های Indian Pines ..... ۱۵
- شکل ۷ : نمودار دقت کلی بر حسب تغییرات پارامتر  $\delta$  برای bw های ۱ و ۳ و ۵ داده های Pavia  
University ..... ۱۶
- شکل ۸: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر  
Indian pines با PC-NRS ..... ۱۷
- شکل ۹: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر  
Indian pines با PC-Gabor-NRS ..... ۱۸
- شکل ۱۰ : نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر  
Pavia university با PC-Gabor-NRS ..... ۱۹
- شکل ۱۱ : تصویر Pavia University طبقه بندی شده PC-NRS ..... ۲۱
- شکل ۱۲ : تصویر Pavia University طبقه بندی شده PC-Gabor-NRS ..... ۲۱
- شکل ۱۳ : تصویر Indian Pines طبقه بندی شده PC-NRS ..... ۲۳
- شکل ۱۴ : : تصویر Indian Pines طبقه بندی شده PC-Gabor-NRS ..... ۲۳

## مقدمه

به کمک تجمیع یک طبقه بندی به کمک نزدیک ترین زیرفضا ها (nearest subspace) با یک regularization Tikhonov که بر اساس فاصله وزن دار شده است به یک روش  $NRS^1$  برای طبقه بندی تصاویر ابرطیفی می‌رسیم .

$NRS$  در اصل برای طبقه بندی بر اساس پیکسل ها (pixel-wised classifier) طراحی شده بود که فقط اطلاعات (امضا طیفی) یک پیکسل مشخص را در نظر می‌گیرد و به اطلاعات پیکسل های همسایه (هرچند مفید) توجهی ندارد.

با استفاده از Gabor Filtering برای تصاویر ابرطیفی می‌توانیم به اطلاعات مکانی مفیدی در مورد داده ها دست پیدا کنیم .

هدف اصلی در این مقاله بررسی مزایای استفاده از Gabor Filtering برای یک طبقه بندی کننده  $NRS$  است .

طبقه بندی کننده ای که بر اساس Gabor Filtering ساخته شده است ، بر روی داده های ابرطیفی متعددی صحت سنجی شده است . نتایج بررسی های نشان داده است که دقت طبقه بندی کننده افزایش قابل توجهی داشته است .

معمولا تصاویر ابرطیفی به حالت یک مکعب ۳ بعدی نشان داده می‌شوند که چند صد کانال طیفی دارند . هر پیکسل ابرطیفی بازتاب طیفی را در این باند های طیفی باریک و به هم پیوسته ضبط می‌کند و به این ترتیب از این باند ها میتوان برای طبقه بندی استفاده کرد (تفاوت در میزان بازتاب در باند های مخالف منجر به تشخیص و تفکیک می‌شود) . از مهمترین کاربرد های طبقه بندی تصاویر میتوان بررسی گسترش شهری ، بررسی محیطی ، یافتن مواد معدنی و ... اشاره کرد .

یک طبقه بندی کننده بیز<sup>۲</sup> مانند بیشترین شباهت ( $MLE^3$ ) یا مدل تلفیقی گوسین ( $GMM^4$ ) از معمول ترین طبقه بندی کننده های تصاویر ابرطیفی هستند . به علت ابعاد بالای داده ها آنالیزگر های تفکیک خطی ( $LDA^5$ ) مثل معیار تفکیک فیشر ( $LFDA^6$ ) برای کم کردن ابعاد فضای داده ها مورد استفاده قرار

<sup>1</sup> Nearest Regularized Subspace

<sup>2</sup> Bayes

<sup>3</sup> Maximum likelihood Estimation

<sup>4</sup> Gaussian Mixture Model

<sup>5</sup> linear discriminant analysis

<sup>6</sup> local Fisher's discriminant analysis

می گیرد و معمولا با کم کردن ابعاد فضای ویژگی با توجه به هدف مساله ، نتایج طبقه بندی بهتر بهبود خواهد یافت .

البته باید توجه داشت در صورتی که تعداد نمونه ها کم باشد ( $SSE^1$ )، آنالیزگر های تفکیک کننده خطی معمولا به دلیل برآورد های آماری نامناسب مخصوصا هنگامی که فضای ویژگی ورودی بسیار بزرگ است ، دچار مشکل می شوند . یعنی در واقع هنگامی که تعداد نمونه های در دسترس از تعداد باند ها خیلی کمتر باشد ، دچار مشکل خواهیم شد .

یکی از روش هایی که به کمک آن می توان این مشکل را برطرف کرد استفاده از  $RLDA^2$  است. برای این کار یک ماتریس که مقادیر قطر اصلی آن اعداد کوچکی هستند به ماتریس پراکنش که شرایط نامناسب آماری دارد اضافه می شود که برآورد پروجکشن LDA به حالت پایدار برسد .

روش دیگری که برای این کار وجود دارد با عنوان "کاهش ابعاد بر اساس زیرفضا"<sup>3</sup> شناخته می شود که به عنوان یک پیش پردازش بر LDA انجام می شود . در این روش معمولا ابتدا آنالیز مولفه اصلی<sup>4</sup> انجام می شود تا ابعاد فضای داده ها کاهش پیدا کند به این صورت که فضای خالی در ماتریس پراکنش ک دارای کمبود رنگ است حذف می شود و به این ترتیب مشکل LDA حل می شود .

باید توجه داشت که هر دو راه جل بیان شده مشکلات مربوط به خودشان را دارند ، مثلا طبقه بندی کننده بیزی که عملیات کاهش ابعاد روی آن صورت گرفته باشد معمولا بر اساس توزیع داده های از پیش تعیین شده ای کار می کند که در عمل نامشخص است . از طرف دیگر به جز طبقه بندی کننده ، هم RLDA و هم LDA زیرفضا LDA به پارامتر های مساله وابسته هستند به طور مثال پارامتر های تنظیم کننده ( regularization parameters) برای RLDA و تعداد مولفه های کاهش یافته برای LDA .

بعضی از طبقه بندی کننده ها برعکس طبقه بندی کننده بیز، بدون توزیع داده از پیش تعریف شده و کاهش ابعاد نیز میتوانند روی داده های ابرطیفی ، طبقه بندی را به خوبی انجام دهند ، مثلا  $SVM^5$  یک طبقه بندی کننده بسیار خوب است که نتایج بسیار مناسبی را به طور ویژه در زمانی که شرایط SSS برقرار باشد ، در اختیار قرار می دهد .

<sup>1</sup> small-sample-size

<sup>2</sup> regularized LDA

<sup>3</sup> subspace-based dimensionality reduction

<sup>4</sup> principal component analysis

<sup>5</sup> support vector machine

در SVM هدف جدا کردن کلاس ها با استفاده از پیدا کردن ابرصفحه بهینه است که به بهترین حالت نمونه های آموزشی که از کلاس آن ها اطلاع داریم را در فضای ویژگی با ابعاد بالا را از هم جدا کند.

نگاشت غیر خطی کرنل در چهارچوب SVM معمولاً می تواند مسائل خطی غیرقابل تفکیک در فضای اصلی را به مسائل قابل تفکیک خطی در فضای ساخته شده بر اساس کرنل تبدیل می کند. برخی از توابع کرنل متداول برای SVM شامل کرنل چند جمله ای (polynomial) و کرنل عملکردی بر پایه شعاع (RBF) است.

اخیراً طبقه بندی به روش  $1^{\text{SRC}}$  برای طبقه بندی الگو ها توسعه داده شده است. در اصل این طبقه بندی کننده برای تشخیص چهره توسعه داده شده بود. در واقع این SRC ها بر این اساس کار می کنند که یک پیکسل را میتوان با  $l_1$ -norm minimization به عنوان یک ترکیب خطی پراکنده از داده های برچسب دار (داده آموزشی) در نظر گرفت

. برای محاسبه  $l_1$ -norm minimization در تشخیص تغییرات در تصاویر ابرطیفی به اطلاعات اولیه مثل داده های آموزشی نیاز داریم .

باید توجه داشت که مراحل آموزش و تست مانند حالت عادی در طبقه بندی های نظارت شده نیست بلکه یک پیکسل که طبقه بندی (تست) می شود با توجه به برچسب نمونه (یا داده آموزشی) و برچسب کلاس به آن کلاسی تعلق میگیرد که کمترین residual representation را داشته باشد .

در NRS برای هر نمونه تخمینی وجود دارد که به عنوان یک ترکیب خطی بر اساس تمام نمونه های برچسب دار موجود در هر کلاس به صورت مستقل نشان داده می شود و بر اساس آن طبقه بندی می شود.

پایه و اساس یک طبقه بندی کننده NRS ، پنالتی ترم  $l_2$  در قالب یک تنظیم کننده Tikhonov بر اساس وزن است .

از طرفی به جای اینکه یک فرض محکم در مورد هندسه مجموعه داده ها داشته باشیم (مثل حالت طبقه بندی کننده بیزین) ، ترم تنظیم کننده  $l_2$  تلاش میکند تا مشکلات ساختاری (ill-conditioning & ill-posedness) را در مساله معکوس، که نسبت به داده های با ابعاد بالا مقاوم است ، حل کند .

از طرف دیگر معیار وزن وابسته به فاصله و محاسبه ماتریس وزن برای ترکیب یک چهارچوب معنا دار را به ترم تنظیم کننده تحمیل می کند . با مقایسه روش هایی که بر مبنای نمایندگی پراکنده کار می کنند ، طبقه بندی کننده NRS فرم حل بسته ای دارد که باعث می شود هزینه محاسباتی بسیار کاهش پیدا کند .

<sup>1</sup> sparse-representation-based classification

هرچند یکی از معایب NRS این است که در اصل برای طبقه بندی بر اساس پیکسل توسعه یافته است که همانطور که قبل تر گفته شد باعث می شود تنها توجه به اطلاعات طیفیتک پیکسل باشد و پیکسل های همسایه بی تاثیر باشند در صورتی که در تصاویر ابرطیفی امکان اینکه دو پیکسل مجاور متعلق به یک کلاس باشند زیاد است .

یکی از عوامل تاثیرگذار در طبقه بندی تصاویر ابرطیفی ، اطلاعات مکانی است و برای این کار میتوان از ویژگی های Gabor برای دستیابی و نشان دادن اطلاعات مکانی مفید استفاده کرد .

در این مقاله مزایای استفاده از ویژگی های Gabor برای طبقه بندی کننده NRS در شرایط SSS مورد بررسی قرار می گیرد . پایگاه داده های متعددی جهت ارزیابی این روش مورد استفاده قرار میگیرند .

نتایج استفاده از ویژگی های Gabor برای طبقه بندی کننده SVM و SRC مورد بررسی قرار خواهند گرفت .

دو روش اصلی برای استفاده از ویژگی های مکانی وجود دارد ، روش اول استخراج انوعی از ویژگی هاست ( مثل بافت یا ویژگی های موج ) و روش دوم استفاده مستقیم از پیکسل ها در یک همسایگی کوچک برای طبقه بندی است با این فرض که پیکسل های این همسایگی متعلق به یک کلاس هستند .

اولین روش باعث بالا رفتن ابعاد فضای ویژگی می شود و همچنین ممکن است عملکرد آن به تصاویر استفاده شده برای استخراج ویژگی های مکانی حساس شود (به طور مثال ویژگی ها می توانند از باند های انتخابی یا PC ها استخراج شوند).

روش دوم معمولاً باعث افزایش ابعاد ویژگی ها نمی شود و عملکرد پایداری تری را در بعضی از موارد نشان می دهد . روش اول در صورتی نسبت به این روش ارجحیت پیدا می کند که استخراج و انتخاب ویژگی ها ، هر دو با دقت بسیار بالا طراحی شوند .

در این مقاله هدف استفاده از روش اول است . به طور دقیق تر هدف استخراج ویژگی های Gabor ای است که به راحتی عملیاتی می شوند . برای SRC ، شرط پیوستگی محلی باعث می شود همان نمونه برچسب دار برای نمایش پیکسل های همسایه انتخاب شود . با این حال ، به دلیل ماهیت مشترک (نه رقابتی) در بین نمونه های برچسب زده شده در NRS ، گسترش مستقیم چنین مفهومی ممکن است لزوماً باعث بهبود عملکرد این طبقه بندی کننده نگردد.

## الگوریتم ها

## Gabor filter

فیلتر Gabor یک تابع سینوسی مدل شده از حالت گاوسی است که در پردازش تصاویر کاربرد بسیاری دارد. در یک سیستم مختصات دو بعدی، این فیلتر شامل یک قسمت حقیقی و یک قسمت مجازی است که در روابط زیر نشان داده شده است.

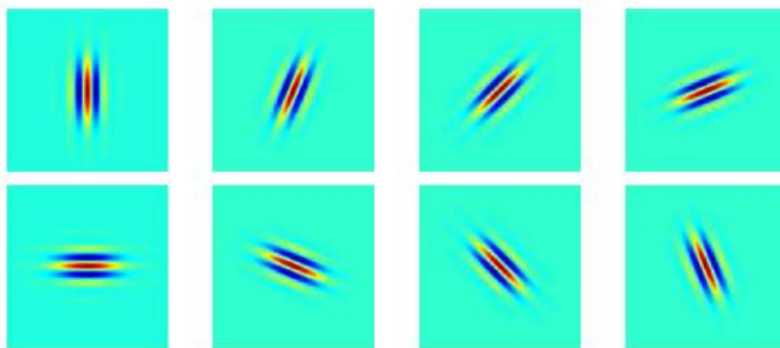
$$g(a, b, \delta, \vartheta, \psi, \sigma, \gamma) = \exp\left(-\frac{a'^2 + \gamma^2 b'^2}{2\sigma^2}\right) \times \exp\left(j\left(2\pi \frac{a'}{\delta} + \psi\right)\right) \quad \text{رابطه ۱}$$

که در این رابطه داریم:

$$a' = a \cos \theta + b \sin \theta \quad \text{رابطه ۲}$$

$$b' = -a \sin \theta + b \cos \theta \quad \text{رابطه ۳}$$

در این روابط  $\delta$  نشان دهنده طول موج فاکتور سینوسی و  $\vartheta$  نشان دهنده زاویه جهت جدایی فیلتر است که در شکل ۱ نشان داده شده است.  $\psi$  نشان دهنده فاز آفست و  $\sigma$  انحراف استاندارد مدل گاوسی و  $\gamma$  نسبت جنبه های مکانی<sup>۱</sup> است که نشان دهنده میزان بیضوی بودن support تابع Gabor است.



شکل ۱: کرنل های دو بعدی Gabor با جهت های مختلف از بالا به پایین و چپ به راست زوایای  $\left\{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}\right\}$

<sup>۱</sup> spatial aspect ratio



در صورتی که  $\psi = 0$  و یا  $\psi = \frac{\pi}{2}$  به ترتیب به قسمت حقیقی و مجازی فیلتر *Gabor* خواهیم رسید .  
پارامتر  $\sigma$  نیز با کمک  $\delta$  و فرکانس پهنای باند مکانی تعریف می شود .

$$\sigma = \frac{\delta}{\pi} \sqrt{\frac{\ln 2}{2} \frac{2^{bw} + 1}{2^{bw} - 1}} \quad \text{رابطه ۴}$$

### الگوریتم NRS

یک پایگاه داده با نمونه های برچسب گذاری شده  $X = \{x_i\}_{i=1}^n$  در یک فضای  $d$  بعدی و برچسب های کلاس های  $\omega_i \in \{1, 2, \dots, C\}$  که در آن  $C$  نشان دهنده تعداد کلاس ها است و  $n$  نیز تعداد کل نمونه ها است . اگر  $n_l$  را تعداد نمونه های موجود برای کلاس 1 ام در نظر بگیریم ، خواهیم داشت :

$$\sum_{l=1}^C n_l = n$$

برآوردی از نمونه هایی که به عنوان  $y$  طبقه بندی می شوند با ترکیب خطی از نمونه های برچسب دار هر کلاس بدست می آید . به این صورت که مقدار  $\tilde{y}_l$  با توجه به نمونه های متعلق به هر کلاس  $X_l$  (نمونه های متعلق به کلاس  $l$ ) محاسبه می شود .

$$\tilde{y}_l = X_l \alpha_l \quad \text{رابطه ۵}$$

در رابطه ۵ ،  $X_l$  ابعادی برابر  $d \times n_l$  خواهد داشت و  $\alpha_l$  نیز برداری  $1 \times n_l$  است که مولفه های آن بردار وزن را تشکیل می دهند .

با بدست آوردن  $\tilde{y}_l$  با رابطه ۵ برپسب کلاس  $y$  بر حسب اینکه باقیمانده برای کدام کلاس مقدار کمتری می شود ، تعیین خواهد شد .

$$\text{class}(y) = \arg \min_{l=1, \dots, C} r_l \quad \text{رابطه ۶}$$

$$r_l = \|\tilde{y}_l - y\|_2^2 \quad \text{رابطه ۷}$$

$r_l$  در این رابطه نشان دهنده مقدار باقیمانده بین پارامتر برآورد شده و نمونه نظیر در طبقه بندی است که از رابطه ۷ محاسبه می شود .

در NRS نحوه محاسبه  $\alpha_l$  بر اساس رابطه ۸ انجام می شود .

$$\alpha_l = \arg \min_{\alpha^*} \|\mathbf{y} - \mathbf{X}_l \alpha^*\|_2^2 + \lambda \|\mathbf{\Gamma}_{l,y} \alpha^*\|_2^2 \quad \text{رابطه ۸}$$

در این رابطه  $\mathbf{\Gamma}_{l,y}$  ماتریس Tikhonov بایاس است که مختص به هر کلاس  $l$  و نمونه ای که قرار است طبقه بندی شود است.  $\lambda$  نیز یک پارامتر تنظیم کننده جهانی است که کمینه سازی بین باقی مانده و ترم های تنظیم کننده را متعادل می کند . باید توجه داشت  $\alpha^*$  فرمی از نمایش  $\alpha_l$  است . معمولاً از فرم  $\mathbf{\Gamma}_l$  قطری به شکل زیر استفاده می شود .

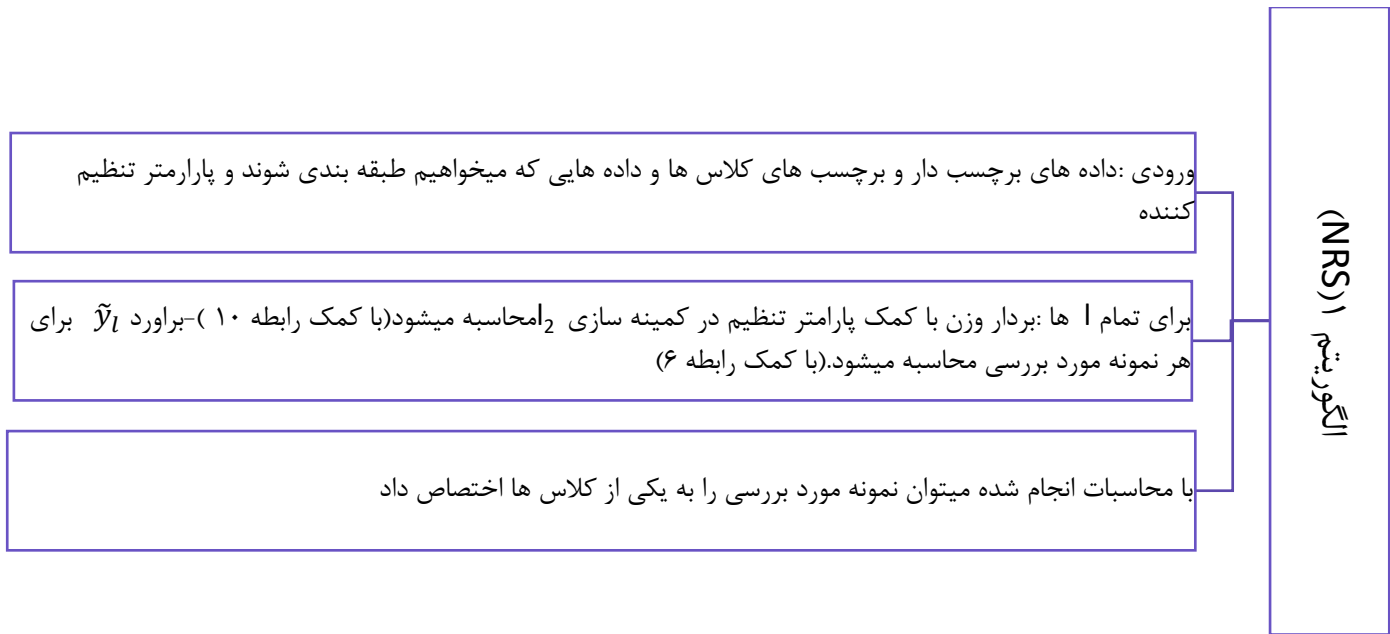
$$\mathbf{\Gamma}_{l,y} = \begin{bmatrix} \|\mathbf{y} - \mathbf{x}_{l,1}\|_2 & & 0 \\ & \ddots & \\ 0 & & \|\mathbf{y} - \mathbf{x}_{l,n_l}\|_2 \end{bmatrix} \quad \text{رابطه ۹}$$

در این ماتریس ،  $x_1, \dots, x_n$  ستون های ماتریس  $\mathbf{X}_l$  برای کلاس  $l$  ام هستند . با توجه به این روابط میتوان بردار وزن را در فرم بسته زیر تعریف کرد .

$$\alpha_l = (\mathbf{X}_l^T \mathbf{X}_l + \lambda^2 \mathbf{\Gamma}_{l,y}^T \mathbf{\Gamma}_{l,y})^{-1} \mathbf{X}_l^T \mathbf{y} \quad \text{رابطه ۱۰}$$

طبقه بندی NRS از لحاظ ریاضی می تواند به عنوان یک مساله کمینه سازی  $l_2$  فرموله شود که در آن تابع هزینه شامل دو ترم است . ترم اول میزان خطا بین نمونه ای که قرار است طبقه بندی شود و نمایندگی خطی بازسازی شده را اندازه گیری می کند و ترم دوم شباهت بین نمونه مورد بررسی و نمونه برچسب دار را بررسی می کند .

ساختار کلی این الگوریتم طبقه بندی در فلوچارت زیر نشان داده شده است .



شکل ۲: فلوچارت الگوریتم اول - طبقه بندی NRS

### الگوریتم ترکیبی NRS-Gabor

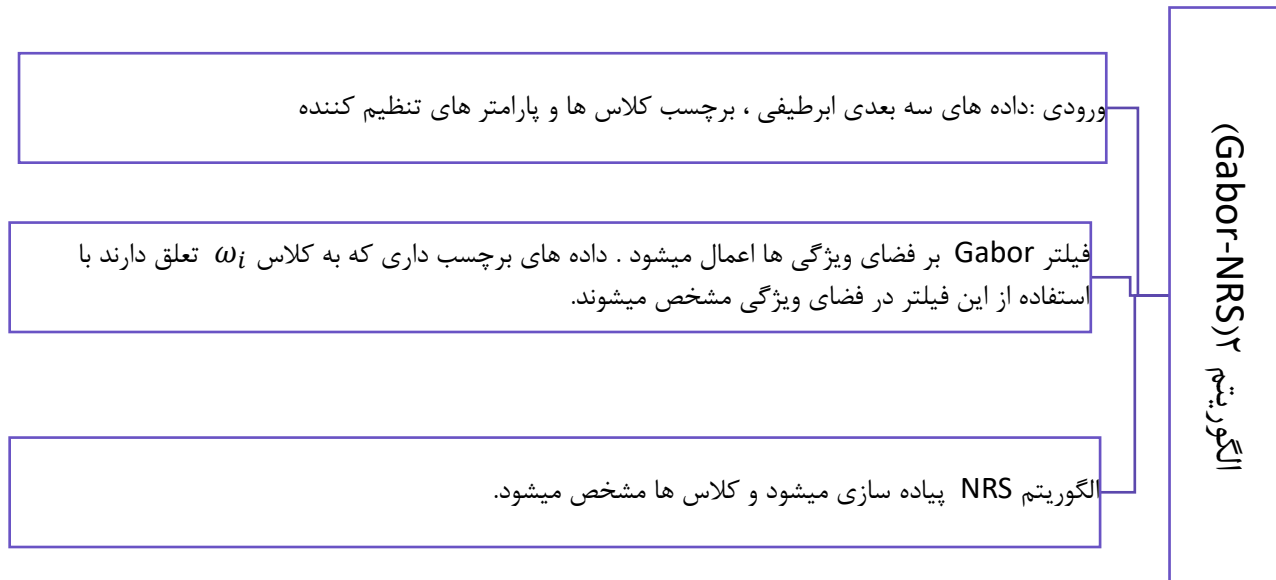
یک فیلتر Gabor می تواند تعدادی از ساختار های فیزیکی یک شی در تصویر را مثل اطلاعات مربوط به جهت گیری با کمک یک کرنل مکانی convolution اخذ کند .

ابتدا یک فیلتر دو بعدی در نظر گرفته می شود تا اطلاعات مکانی مفید را از یک زیر فضای PCA استخراج کند. سپس میتوان کارایی این فیلتر را در یک زیر مجموعه از باند های اصلی به کمک انتخاب باند ها ارزیابی کرد. تخمین خطا به صورت خطی برای انتخاب باند نظارت نشده معمولا بر اساس معیار های شباهت انجام می - شود.

در این پروژه ویژگی های Gabor به عنوان پیش پردازش برای طبقه بندی کننده NRS در نظر گرفته شده اند که باعث می شود یک طبقه بندی NRS بر اساس فیلتر Gabor<sup>1</sup> صورت بگیرد. هدف اصلی بررسی مزایای استفاده از فیلتر Gabor برای تصاویر ابرطیفی در حالت SSS است .

<sup>1</sup> Gabor-filtering-based NRS

روش کلی کار در فلوچارت شکل ۳ نشان داده شده است .



شکل ۳: فلوچارت الگوریتم دوم - Gabor-filtering-based NRS

## پیاده سازی

## داده های مورد استفاده

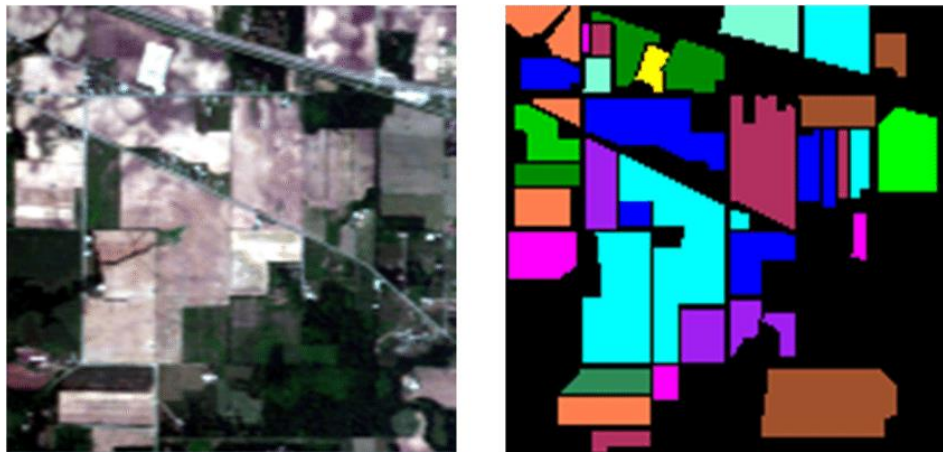
از دو سری داده اصلی برای پیاده سازی الگوریتم ها استفاده شده است :

- Indian pines
- City of Pavia

## داده های Indian pines

این داده ها با استفاده از سنسور طیف سنج مرئی و مادون قرمز (AVIRIS) هوابرد و اداره فضایی (NASA) اداره هوایی و فضایی آمریکا بدست آمده است و در ژوئن سال ۱۹۹۲ در سایت تست کاج هند واقع در شمال غربی ایندیانا جمع آوری شد.

این تصویر نشان دهنده یک سناریو طبقه بندی پوشش گیاهی است که ابعادی برابر  $145 \times 145$  پیکسل دارد و در  $220$  باند در محدوده  $0.4$  تا  $2.45$  میکرومتر (مرئی تا مادون قرمز) با قدرت تفکیک مکانی  $20$  متر برداشت شده است.  $16$  پوشش متفاوت زمینی در این صحنه تصویر وجود دارد پس در نقشه داده های واقعیت زمینی  $16$  کلاس داریم، اما از تمام کلاس ها برای طبقه بندی استفاده نمیشود،  $8$  کلاس با حدود  $20$  پیکسل برای هر کلاس (مجموعاً  $160$  پیکسل) استفاده می شود.

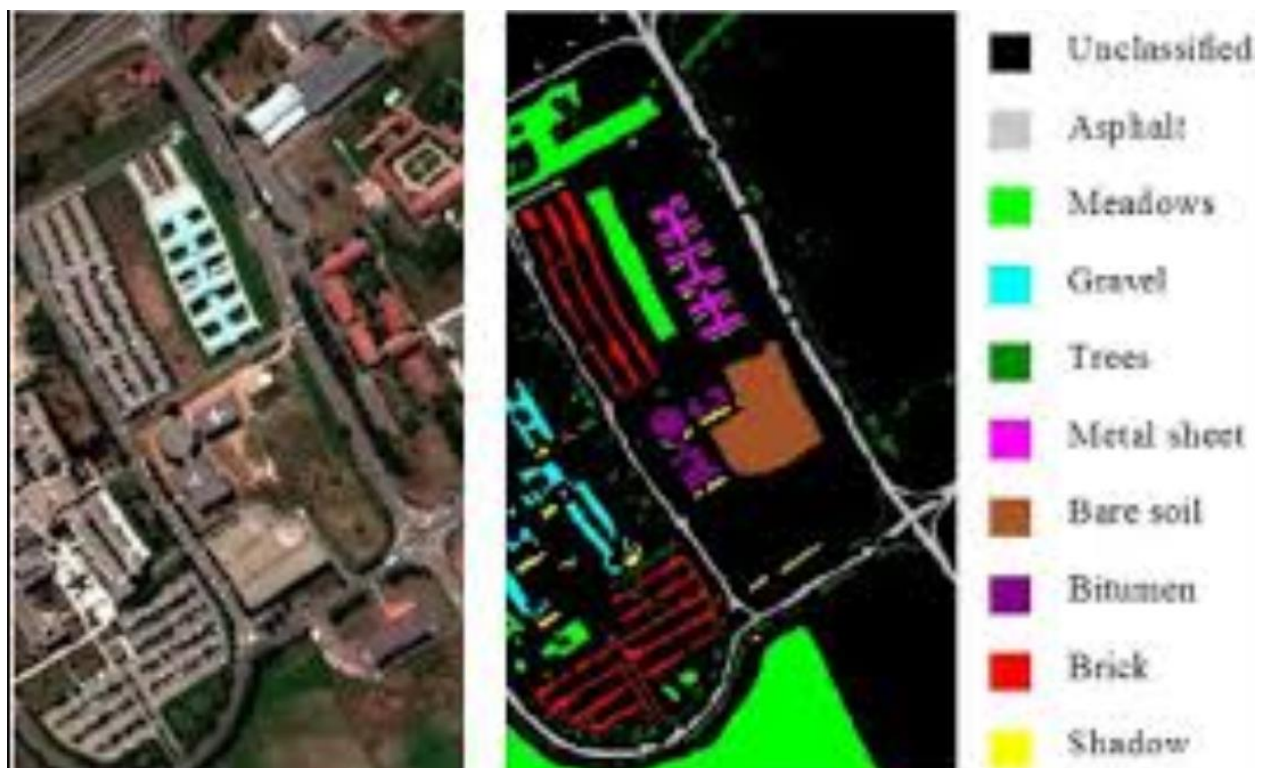


شکل ۴: تصویر منطقه Indian pines و کلاس های موجود

## داده های Pavia University

مجموعه داده های ابر طیفی دوم توسط سنسور طیف سنج تصویربرداری سیستم بازتابی (ROSIS) جمع آوری شده است. صحنه تصویر با پوشش مکانی  $۶۱۰ \times ۳۴۰$  پیکسل که شهر پاویا ایتالیا را پوشش می دهد، تحت پروژه HySens که توسط DLR (آژانس هوایی آلمان) اداره می شود، جمع آوری شده است.

این پایگاه داده ۱۰۳ باند طیفی دارد (بعد از حذف باند های آبی) که محدوده طیفی ۰.۴۳ تا ۰.۸۶ میکرومتر را پوشش می دهند. قدرت تفکیک مکانی آن نیز ۱.۳ متر است. حدوداً ۹ کلاس با مجموع ۴۲۷۷۶ پیکسل برای داده های واقعیت زمینی در نظر گرفت شده است (کلاس بدون طبقه بندی در نظر گرفته نشده است).



شکل 5 : تصویر منطقه Pavia University

### پیاده سازی

در ابتدا برای مقایسه با یک مرجع الگوریتم NRS را به تنهایی بر داده ها اعمال میکنیم ، نتایج بدست آمده در ادامه ارائه خواهد شد . سپس الگوریتم NRS بر مبنای فیلتر Gabor اعمال می شود که در ادامه بررسی می شود .

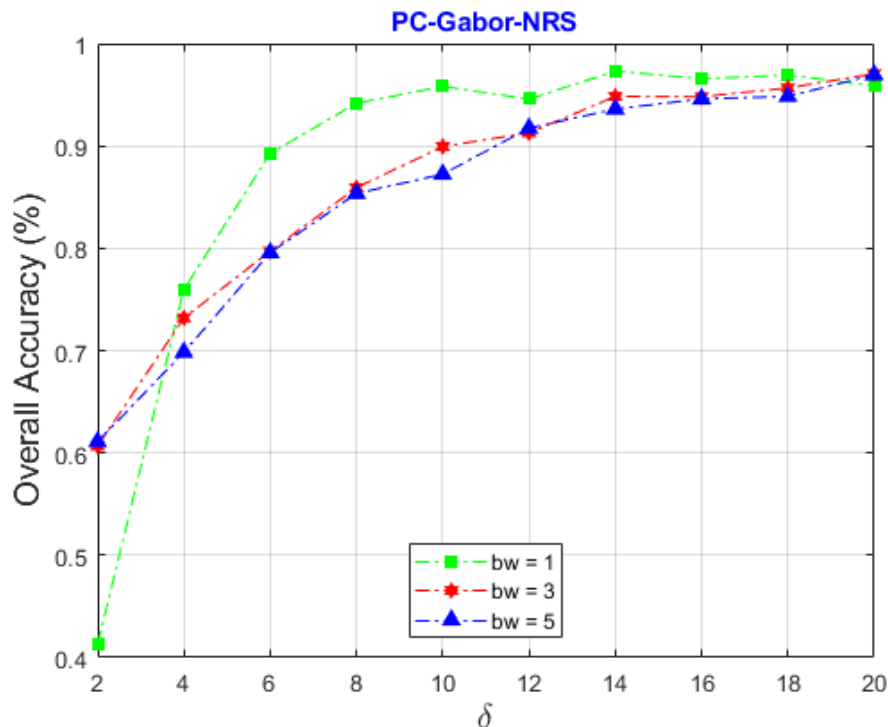
نتایج بدست آمده از الگوریتم NRS به تنهایی در تصاویر زیر نشان داده شده است .

در ابتدا باید ذکر شود پارمتر های این فیلتر در تصاویر ابر طیفی این پروژه ۸ جهت گیری در نظر گرفته شده است (در شکل ۱ نیز مشخص است).

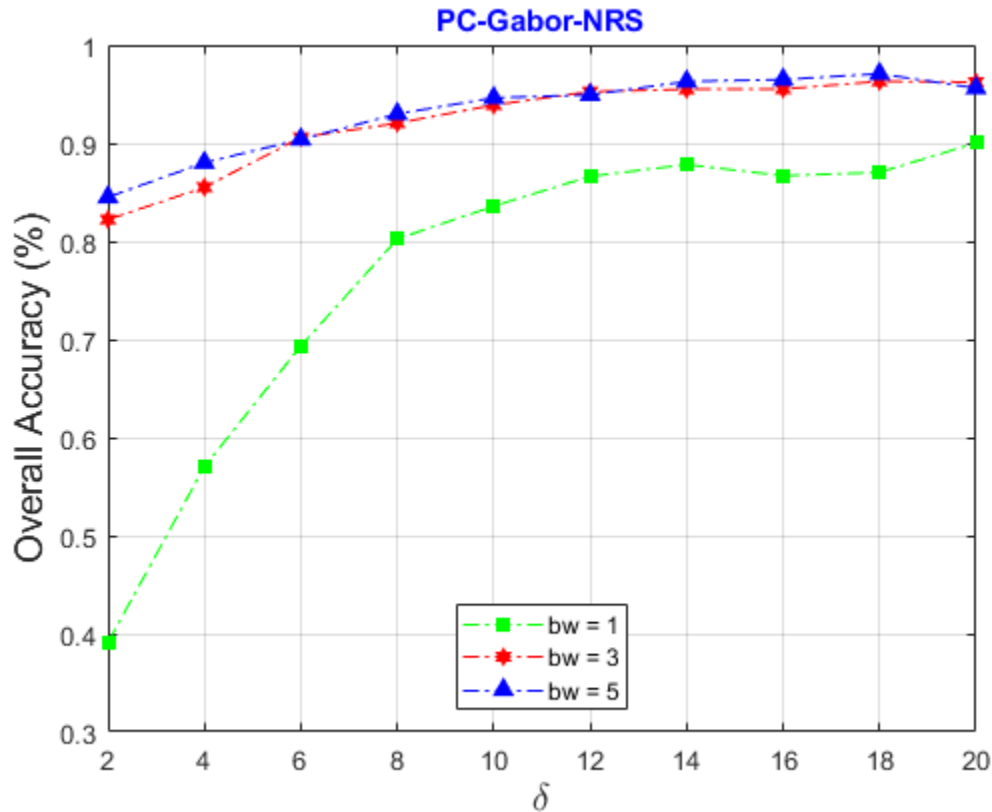
فیلتر Gabor در صورتی که با  $\delta$  های مختلف یا  $bw$  های مختلف اعمال شود، نتایج مختلفی خواهد داشت. همچنین به ازای  $\lambda$  های متفاوت نیز به دقت های متفاوت می رسیم که بررسی خواهد شد .

### بررسی مقادیر $\delta$ های مختلف و اثر آن بر دقت کلی

\*تعداد باند های استخراج شده مورد استفاده در این تصاویر برای اجرای الگوریتم ها با استفاده از الگوریتم PCA، ۱۰ باند است که پس از اعمال ۸ ویژگی Gabor بر هریک از آنها تعداد نهایی آنها به ۸۰ باند می رسد. بقیه پارامتر های موجود نیز بر اساس آزمون و خطا بهینه می شوند .



شکل ۶ : نمودار دقت کلی بر حسب تغییرات پارامتر  $\delta$  برای  $bw$  های ۱ و ۳ و ۵ داده های Indian Pines



شکل ۷: نمودار دقت کلی بر حسب تغییرات پارامتر  $\delta$  برای bw های ۱ و ۳ و ۵ داده های Pavia University

در نمودارهای نمایش داده شده در شکل های ۶ و ۷ تعداد داده های آموزشی ۷۰ داده بوده است و  $\lambda$  در نظر گرفته شده ۰.۱ است.

در این نمودار ها اثر مقادیر مختلف  $\delta$  بر دقت کلی برای bw های متفاوت نشان داده شده است. بر اساس نمودار ها مشخص است که مقدار بهینه  $\delta$  در هر دو تصویر ۱۸ است.

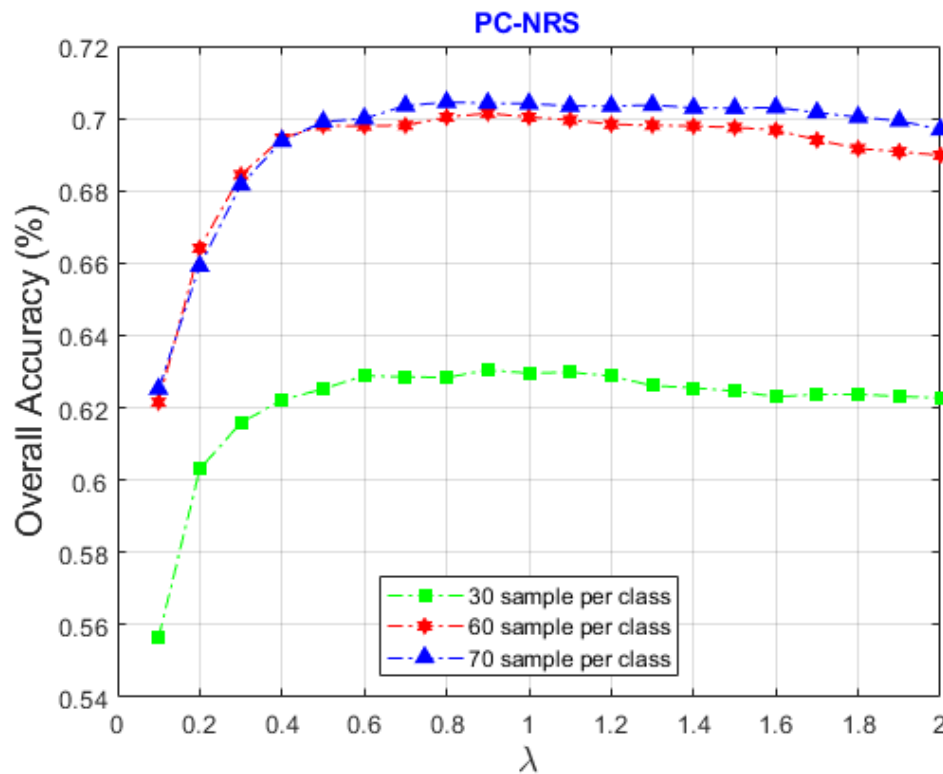
برای تصویر Indian pines بهترین مقدار  $bw=1$  است و برای تصویر Pavia university بهترین مقدار  $bw=5$  است. در شکل های ۶ و ۷ مشخص است که تفاوت زیادی بین bw های ۳ و ۵ وجود ندارد ولی اختلاف آن ها با  $bw=1$  زیاد است، این مساله در نمودار شکل ۶ برعکس است.

از این نمودار ها نتیجه می گیریم که همواره افزایش bw باعث بهبود دقت نمی شود و گاهی ممکن است به دلیل شرایط خاصی که بعضی از کلاس ها دارند و باتوجه به درهم تنیدگی آنها، مقدار پهنای باند های کمتر کارآمدتر باشند که در نمودار ۶ این موضوع را مشاهده می کنید.



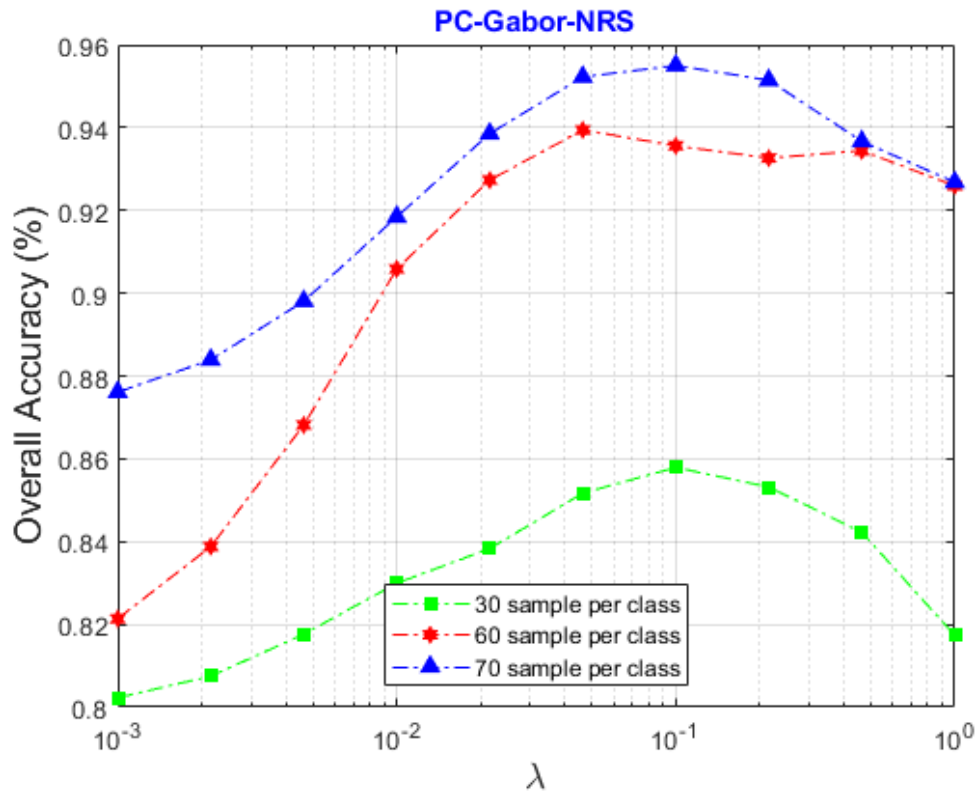
### بررسی مقادیر $\lambda$ های مختلف و اثر آن بر دقت کلی

در این قسمت مقدار بهینه برای پارامتر  $\lambda$  (پارامتر تنظیم کننده) در فضاهای ویژگی با ابعاد متفاوت بررسی می شود. مقدار بهینه برای پارامتر تنظیم کننده طبق نمودار بر اساس بهترین دقت بدست آمده برای هر تصویر تعیین می شود که در ادامه برای هر دو سری داده بررسی می شود. برای بررسی تاثیر پارامتر  $\lambda$  لازم است تا بقیه پارامتر هارا ثابت بگیریم و برای این کار با توجه به نمودارهای ۶ و ۷ پارامترهای  $\delta$  و  $bw$  را در بهترین حالت خود که از نمودار هم قابل استخراج است به طور ثابت قرار می دهیم و تاثیر پارامتر  $\lambda$  را در حالت تعداد داده های آموزشی مختلف بررسی می کنیم. همچنین این بررسی را برای حالتی که از فیلتر Gabor استفاده می کنیم و همچنین حالتی که از فیلتر Gabor استفاده نمی کنیم و NRS را بر باندهای PCA اعمال می کنیم نیز مقایسه می کنیم.



شکل ۸: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر Indian pines با PC-NRS

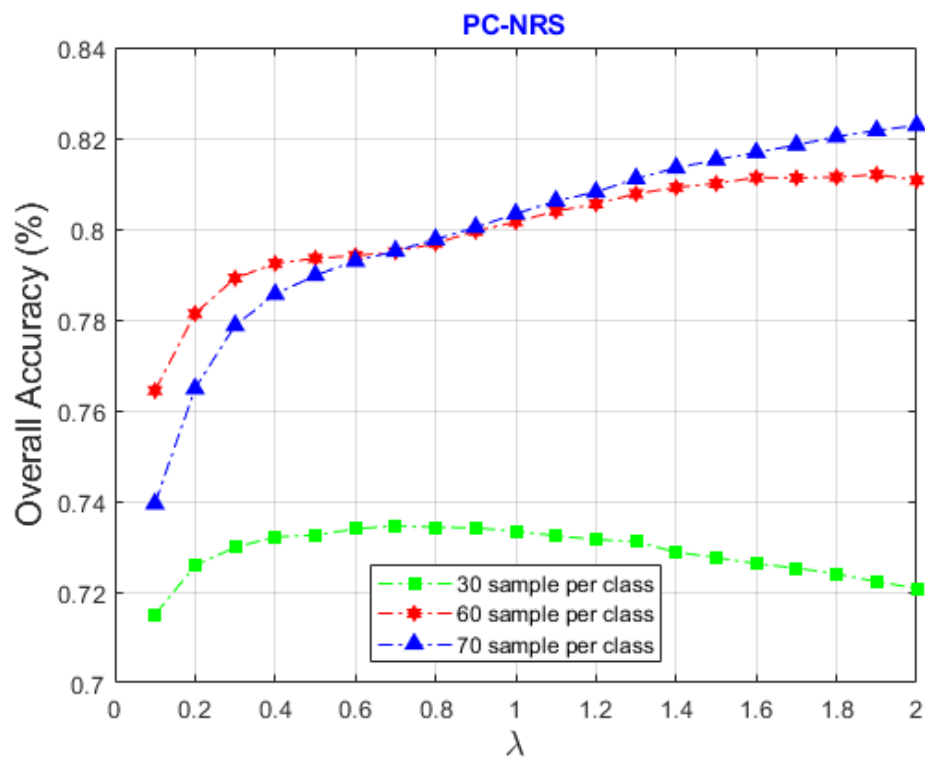
نمودار فوق نتیجه اعمال الگوریتم NRS بر داده خام Indian Pines بوده و هیچ فیلتر Gabor بر آن اعمال نشده است و فقط از PCA استفاده شده و ۱۰ مولفه اصلی استخراج شده اند. از نمودار مشخص است که بهترین حالت PC-NRS برای داده Indian pines ,  $\lambda=0.8$  و 70 داده آموزشی می باشد تصویر طبقه بندی شده با پارامتر های ذکر شده در ادامه گزارش آورده شده است. دقت طبقه بندی در این حالت 70.52 % است.



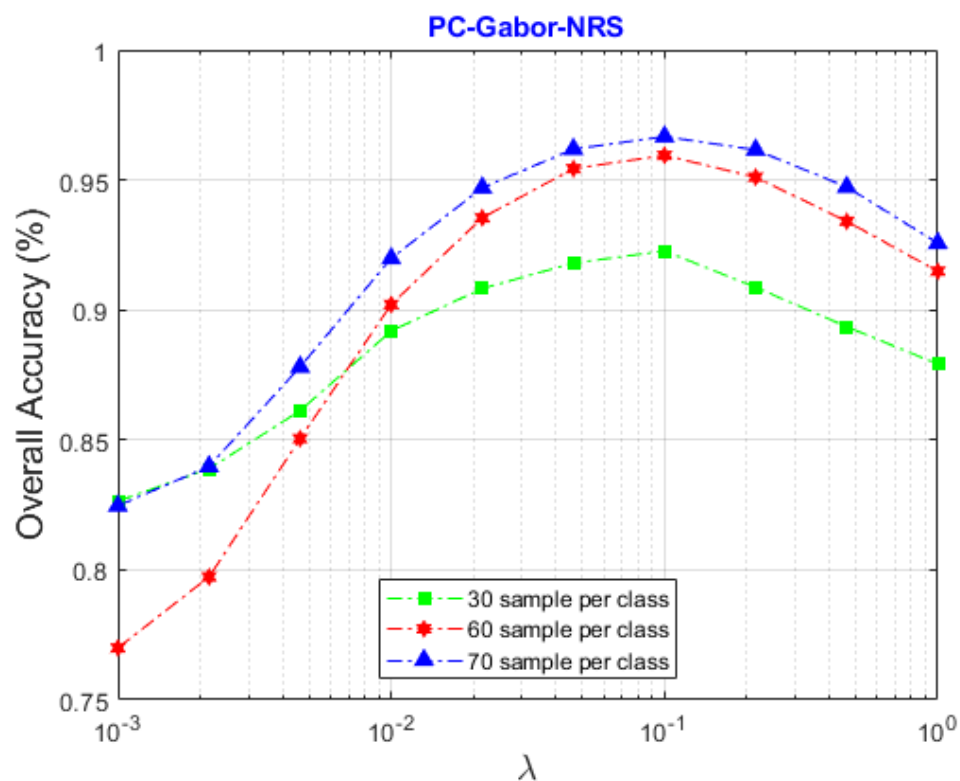
شکل ۹: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر Indian pines با PC-Gabor-NRS

نمودار فوق تاثیر پارامتر  $\lambda$  و همچنین تعداد داده های آموزشی را بر دقت کل در حالتی که از فیلتر Gabor استفاده شده است نشان می دهد. همانطور که مشاهده می کنید با افزایش تعداد داده های آموزشی دقت بهبود پیدا می کند که امری طبیعی است و بهترین مقدار پارامتر  $\lambda$  را در ۰.۱ مشاهده می کنید. اگر این نمودار را با نمودار تصویر ۸ مقایسه کنید می بینیم که دقت کل در حالتی که از فیلتر Gabor استفاده می کنیم بسیار بهتر از حالتی است که NRS را فقط بر ۱۰ مولفه اصلی (PCA) اعمال می کنیم.

\* در مورد داده Indian Pines به این دلیل که تعداد نمونه های سه کلاس ۱، ۷ و ۹ از ۳۰ نمونه کمتر بود، این کلاس ها حذف شدند و به جای ۱۶ کلاس، ۱۳ کلاس در نظر گرفته شده است.



شکل ۱۰: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر Pavia university با PC-NRS



شکل ۱۱: نمودار دقت کلی بر حسب تغییرات پارامتر  $\lambda$  با تعداد داده های آموزشی متفاوت برای تصویر Pavia university با PC-Gabor-NRS

نمودار شکل ۱۰ نتیجه اعمال الگوریتم NRS بر داده خام Pavia university بوده و هیچ فیلتر Gabor بر آن اعمال نشده است و فقط از PCA استفاده شده و ۱۰ مولفه اصلی استخراج شده اند. از نمودار مشخص است که بهترین حالت PC-NRS برای داده Pavia university ,  $\lambda=2$  و 70 داده آموزشی می باشد تصویر طبقه بندی شده با پارامتر های ذکر شده در ادامه گزارش آورده شده است. دقت طبقه بندی در این حالت 86.78٪ است.

نمودار شکل ۱۱ تاثیر پارامتر  $\lambda$  و همچنین تعداد داده های آموزشی را بر دقت کل در حالتی که از فیلتر Gabor استفاده شده است نشان می دهد. همانطور که مشاهده می کنید با افزایش تعداد داده های آموزشی دقت بهبود پیدا می کند که امری طبیعی است و بهترین مقدار پارامتر  $\lambda$  را در ۰.۱ مشاهده می کنید. اگر این نمودار را با نمودار تصویر ۱۰ مقایسه کنید می بینیم که دقت کل در حالتی که از فیلتر Gabor استفاده می کنیم بسیار بهتر از حالتی است که NRS را فقط بر ۱۰ مولفه اصلی (PCA) اعمال می کنیم.

با توجه به نمودار های شکل های ۸ تا ۱۱ میتوان دید که در کل دقت برای هر دو تصویر هنگامی که ابتدا از فیلتر Gabor استفاده می شود (نمودار های شکل ۹ و ۱۱) بهتر از حالتی است که NRS به تنهایی اعمال می شود (نمودار شکل ۸ و ۱۰).

برای تصویر Pavia University و همچنین تصویر Indian Pines در حالت استفاده از فیلتر Gabor مقدار بهینه  $\lambda = 0.1$  است.

\*در این تصاویر برای مقادیر  $bw$  و  $\delta$  از مقادیر بهینه بدست آمده در مرحله قبل به شرح زیر است :

- تصویر *Indian Pines* :  $bw = 1$  و  $\delta = 18$
- تصویر *Pavia University* :  $bw = 5$  و  $\delta = 18$

باید به این نکته در مورد تعداد داده های آموزشی توجه کرد که در حالتی که از الگوریتم NRS به تنهایی استفاده می شود، در صورتی که تعداد داده ها به اندازه کافی باشد، تفاوت مشهودی در دقت کلی ایجاد نمی کند، این مساله با مقایسه بین منحنی های مربوط به ۶۰ و ۷۰ داده در کلاس ها در شکل ۸ مشخص است. اما این مساله در مورد استفاده از فیلتر Gabor به عنوان پیش پردازش صادق نیست که این موضوع از منحنی های مربوطه در شکل های ۹ و ۱۱ مشهود است. در حالتی که از فیلتر *Gabor* به عنوان پیش پردازش استفاده می کنیم تفاوت دقت بین ۴۰ داده آموزشی و ۷۰ داده آموزشی زیاد است.

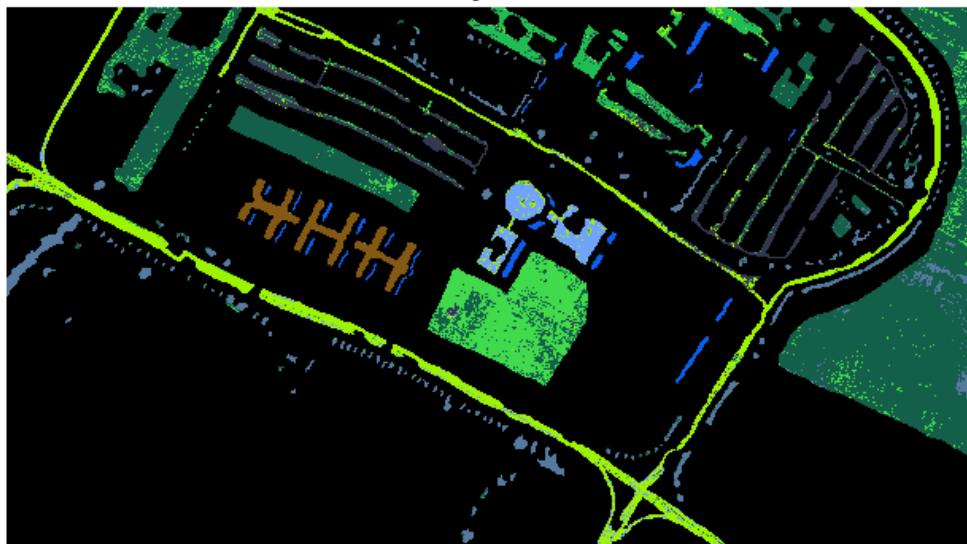
در مجموع اگر به نمودار ها توجه کنیم تفاوت دقت چندانی بین ۶۰ و ۷۰ داده آموزشی مشاهده نمی کنیم و این موضوع قدرت الگوریتم را نشان می دهد اما هرچه تعداد داده کمتر باشد سرعت اجرای الگوریتم بیشتر است.

## تصاویر طبقه بندی شده

ابتدا تصاویر طبقه بندی شده مربوط به داده های pavia university و سپس تصویر طبقه بندی شده مربوط به Indian pines در شکل های زیر آمده است .

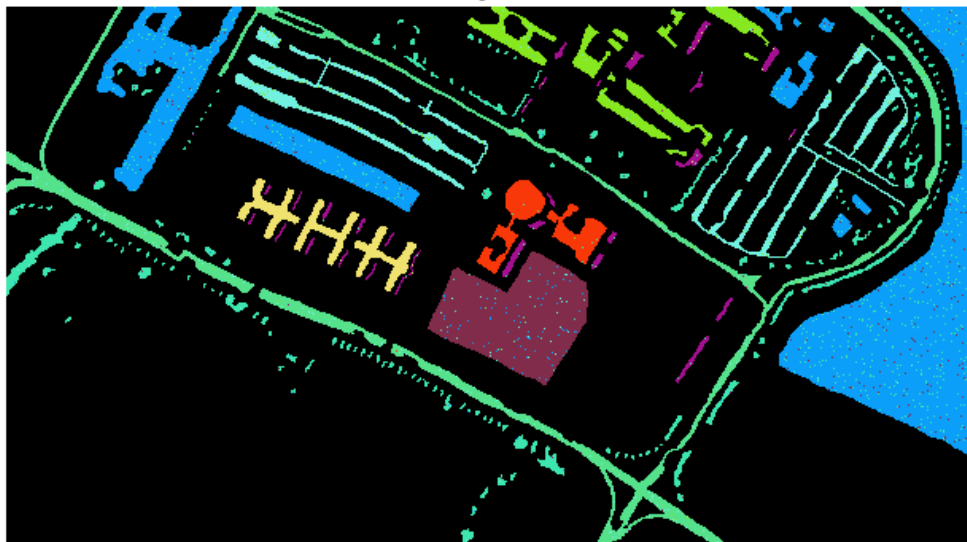
## تصاویر طبقه بندی شده Pavia University

accuracy = 86.7823



شکل ۱۲ : تصویر طبقه بندی شده Pavia University با PC-NRS

accuracy = 96.131



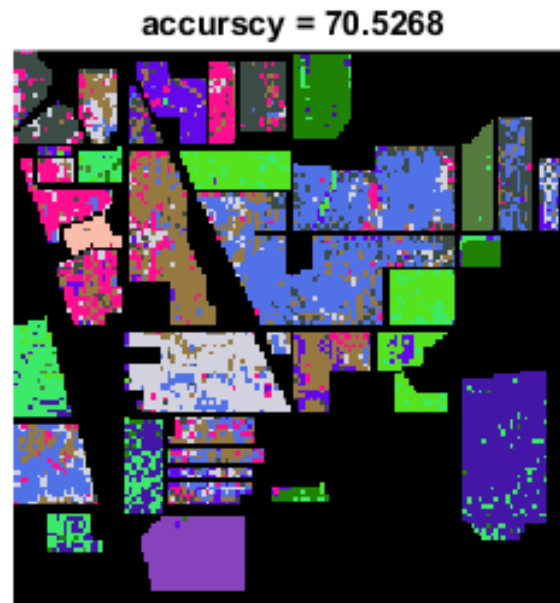
شکل ۱۳ : تصویر طبقه بندی شده Pavia University با PC-Gabor-NRS

شکل ۱۲ تصویر طبقه بندی شده pavia university را با الگوریتم PC-NRS، یعنی بدون اعمال فیلتر Gabor با مقدار  $\lambda=2$  و ۷۰ داده آموزشی نشان می‌دهد. همچنین شکل ۱۳ همین تصویر را اما این بار با روش PC-Gabor-NRS، یعنی اعمال فیلتر Gabor و با پارامتر های  $bw = 5$  و  $\delta = 18$  و  $\lambda = 0.1$  و همچنین تعداد ۷۰ داده آموزشی نمایش می‌دهد. این پارامترها با توجه به نمودار های ۷ و ۱۰ و ۱۱ بهینه ترین پارامترها می‌باشند.

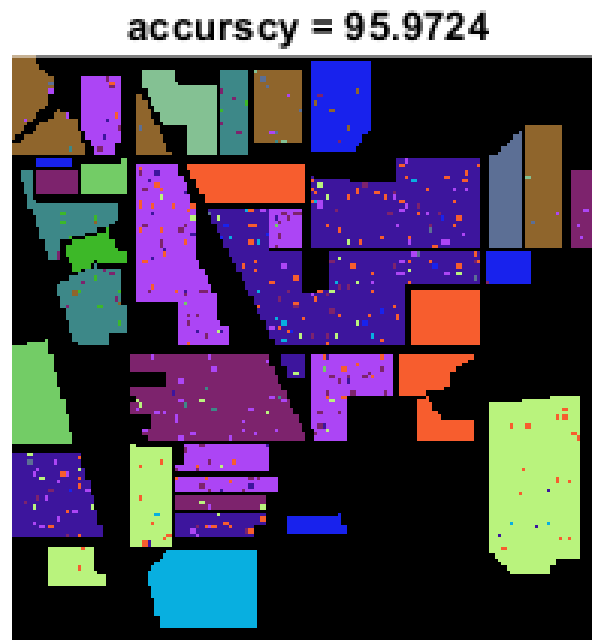
همانطور که از مقایسه دو تصویر مشخص است تصویری که با فیلتر Gabor طبقه بندی شده با اختلاف دقت بهتری دارد درحالی که تعداد داده های آموزشی آنها یکسان بوده است.

\*باید توجه داشت تغییر رنگ ها در شکل ها به معنای تفاوت در ماهیت کلاس ها نیست، صرفا در هر مرحله طبقه بندی رنگ دهی به صورت تصادفی انجام می‌شود.

## تصاویر طبقه بندی شده Indian Pines



شکل ۱۴: تصویر طبقه بندی شده Indian Pines با PC-NRS



شکل ۱۵: تصویر طبقه بندی شده Indian Pines با PC-Gabor-NRS

شکل ۱۴ تصویر طبقه بندی شده Indian pines را با الگوریتم PC-NRS، یعنی بدون اعمال فیلتر Gabor با مقدار  $\lambda=0.8$  و ۷۰ داده آموزشی نشان می دهد. همچنین شکل ۱۵ همین تصویر را اما این بار با روش PC-Gabor-NRS، یعنی اعمال فیلتر Gabor و با پارامترهای  $bw = 1$  و  $\lambda = 0.1$  و  $\delta = 18$  و همچنین

تعداد ۷۰ داده آموزشی نمایش می‌دهد. این پارامترها با توجه به نمودار های 6 و 8 و 9 بهینه ترین پارامترها می‌باشند.

همانطور که از مقایسه دو تصویر مشخص است در اینجا هم تصویری که با فیلتر Gabor طبقه بندی شده با اختلاف دقت بهتری دارد درحالی که تعداد داده های آموزشی آنها یکسان بوده است.

### نتایج

در الگوریتم اول برای تعیین کلاس هر پیکسل از میزان باقی مانده ها استفاده می‌شود و پیکسل به کلاسی که کمترین باقی مانده را دارد اختصاص داده می‌شود. وقتی از روش های مختلف استفاده می‌شود، باقی مانده هایی که بدست می‌آید متفاوت خواهد بود.

اطلاعاتی که از طریق روش PC-Gabor در مورد باقی مانده ها بدست می‌آید (برای تصمیم گیری در طبقه بندی) نسبت به NRS به واقعیت نزدیک تر است .

کارایی این روش ها را میتوان با تعریف درست بردار وزن به حالت بهینه رساند. در واقع می‌توان بردار وزن را به صورت انطباقی تعریف کرد چون ترم تنظیم کننده بر اساس فاصله بین نمونه مورد نظر و نمونه در یک کلاس مشخص تعیین می‌شود و نسبت به آن بایاس است .



## کد برنامه :

## بدنه اصلی برنامه

```

clear all; close all; clc

load PaviaU
paviaU = paviaU./max(paviaU(:));
Data = zeros(340,610,103);
for i = 1:size(paviaU,3)
    Data(:,:,i) = paviaU(:,:,i)';
end
[m n d] = size(Data);
load PaviaU_gt
paviaU_gt = paviaU_gt';

NTrain = 70;
no_class = 9;
num_train = ones(1, no_class)*NTrain;
num_test = [6631,18649,2099,3064,1345,5029,1330,3682,947] - NTrain;
mask = zeros(size(paviaU_gt));
for i = 1: no_class
    tmp = find(paviaU_gt==i);
    index_i = randperm(length(tmp));
    mask(tmp(index_i(1:num_train(i)+num_test(i)))) = i;
end

Feature_P = Gabor_feature_extraction_PC(Data, 5);

Data = []; Labels = [];
d = size(Feature_P, 3);
Data_tmp = reshape(Feature_P, m*n, d);
for i = 1: no_class
    pos = find(mask==i);
    Data = [Data; Data_tmp(pos, :)];
    Labels = [Labels, length(pos)];
end
clear Data_tmp Feature_P DataTest

DataTrain = []; paviaU = [];
CTrain = num_train; CTest = num_test;
a = 0;
for i = 1: no_class
    Data_tmp = Data((a+1):(Labels(i)+a), :);
    a = Labels(i) + a;
    index_i = randperm(Labels(i));
    DataTrain = [DataTrain; Data_tmp(index_i(1:num_train(i)), :)];
    paviaU = [paviaU; Data_tmp(index_i(1:end), :)];
end
it = 1;
for lambda = 0.1
    class_NRS = NRS_Classification(DataTrain, CTrain, paviaU, lambda);
    [confusion, accur_NRS(it)] = confusion_matrix_we(class_NRS, Labels);
    [lambda, accur_NRS(it)]

```

```

        it = it+1;
    end
    class = zeros(340,610);
    C = zeros(340,610,9);
    a = 0;
    T = [6631 , 18649 , 2099 , 3064 , 1345 , 5029 , 1330 , 3682 , 947];
    for i = 1:9
        fi = find(paviaU_gt == i);
        class(fi) = class_NRS(a+1:T(i) + a);
        a = T(i) + a;
    end
    for i = 1:9
        C(:, :, i) = class == i;
    end
    final = ToRGB(C);
    imshow(final)

```

### توابع محاسبه کرنل gabor :

```

function DataGabor = Gabor_feature_extraction_PC(Data, BW)

N_PC = 10;
[m n d] = size(Data);
DataTest = reshape(Data, m*n, d);
Psi = PCA_b(DataTest', N_PC);
DataTestN = DataTest*Psi;
DataN = reshape(DataTestN, m, n, N_PC);
[m n d] = size(DataN);

delta = 18;
psi = [0 pi/2];
gamma = 0.5;
N = 8;
DataGabor = zeros(m, n, N*N_PC);
for i = 1: N_PC
    img_in = DataN(:, :, i);
    bw = BW;
    theta = 0;
    for n=1:N
        gb = gabor_fn(bw,gamma,psi(1),delta,theta) + 1i *
        gabor_fn(bw,gamma,psi(2),delta,theta);
        DataGabor(:, :, (i-1)*N+n)=abs(imfilter(img_in, gb, 'symmetric'));
        theta = theta + 2*pi/N;
    end
end
end

function gb=gabor_fn(bw,gamma,psi,delta,theta)

sigma = delta/pi*sqrt(log(2)/2)*(2^bw+1)/(2^bw-1);
sigma_x = sigma;

```

```

sigma_y = sigma/gamma;

sz=fix(8*max(sigma_y,sigma_x));
if mod(sz,2)==0, sz=sz+1;end

[x y]=meshgrid(-fix(sz/2):fix(sz/2),fix(sz/2):-1:fix(-sz/2));

x_theta=x*cos(theta)+y*sin(theta);
y_theta=-x*sin(theta)+y*cos(theta);

gb=exp(-
0.5*(x_theta.^2/sigma_x^2+y_theta.^2/sigma_y^2)).*cos(2*pi/delta*x_theta+psi)
;
end

function Psi = PCA_b(input, pc)

[~, M] = size(input);

X_mean = mean(input, 2);
input = input - repmat(X_mean, 1, M);

Sigma = (input * input')/M;

[Psi, ~, ~] = svd(Sigma);

Psi = Psi(:, 1:pc);
end

```

### تابع طبقه بندی کننده NRS:

```

function class = NRS_Classification(DataTrain, CTrain, DataTest, lambda)

numClass = length(CTrain);
[m Nt]= size(DataTest);
for j = 1: m
    Y = DataTest(j, :);
    a = 0;
    for i = 1: numClass
        HX = DataTrain((a+1): (CTrain(i)+a), :);
        a = CTrain(i) + a;

        Y_hat = tikhanov(Y, HX, lambda);

        Y_snr(i) = SNR(Y, Y_hat);
    end
    [value class(j)] = max(Y_snr);
end
end

function predicted_vector = tikhanov(y, reference, lambda)

```

```

[m Nt]= size(reference);
lambda2 = lambda^2;

H = reference';

norms = sum((H - repmat(y', [1 m])).^2);

G = diag(lambda2.*norms);

weights = (H'*H + G)\(H'*y');

predicted_vector = H*weights(:);
predicted_vector = predicted_vector';
end

function S = SNR(x, x_hat)

if (~isvector(x) || ~isvector(x_hat))
    error('x and x_hat must be vectors');
end

x = x(:);
x_hat = x_hat(:);

diff = x - x_hat;
mse = diff' * diff / length(x);

variance = cov(x);
S = 10*log10(variance / mse);
end

```

تابع محاسبه ماتریس ابهام و دقت کل:

```

function [confusion, accuracy] = confusion_matrix_wei(class, c)
class = class.';
c = c.';

n = length(class);
c_len = length(c);

if n ~= sum(c)
    disp('WRANING: wrong inputting!');
    return;
end

confusion = zeros(c_len, c_len);
a = 0;
for i = 1: c_len
    for j = (a + 1): (a + c(i))

```

```

        confusion(i, class(j)) = confusion(i, class(j)) + 1;
    end
    a = a + c(i);
end
accuracy = sum(diag(confusion))/sum(c);

```

تابع تشکیل تصویر رنگی طبقه بندی شده:

```

function final = ToRGB(C)

final = zeros(size(C,1),size(C,2),3);
a = zeros(size(final(:,:,1)));
b = a; c = b;
for i =1:size(C,3)
    d = C(:,:,i);
    rgb = randi([0,255],256,1);
    idx = randi([1,256],256,1);
    a(d == 1) = rgb(idx(randi([1,256],1,1)));
    b(d == 1) = rgb(idx(randi([1,256],1,1)));
    c(d == 1) = rgb(idx(randi([1,256],1,1)));
end
a = uint8(a);
b = uint8(b);
c = uint8(c);
final = cat(3,b,a,c);
end

```