

## گزارش فاز دوم پروژه هوش محاسباتی

دانشجویان: غزاله بابایی – عطیه بنکدار

در این پروژه قصد داریم تا با استفاده از الگوریتم های طبقه بندی اعداد دست نویس دیتاست mnist را بهتر شناسایی کنیم.

برای این کار ابتدا فیلتر سوبل را روی دیتاست ران کرده که این کار توسط یک تابع `convolve` و دو کرنل `Gx`

و `Gy` پیاده سازی می شوند. در ادامه به تاثیرات و نقش این فیلتر و کرنل های آن برای طبقه بندی بهتر می پردازیم.

### کرنل `Gx`

- این کرنل برای محاسبه گرادیان تصویر در جهت افقی ( $x$ ) استفاده می شود.
- هدف آن شناسایی تغییرات شدت پیکسل ها در امتداد محور افقی است، به طوری که لبه هایی که به صورت عمودی قرار دارند، شناسایی شوند.

### کرنل `Gy`

- این کرنل برای محاسبه گرادیان تصویر در جهت عمودی ( $y$ ) استفاده می شود.
- هدف آن شناسایی تغییرات شدت پیکسل ها در امتداد محور عمودی است، به طوری که لبه هایی که به صورت افقی قرار دارند، شناسایی شوند.

هر دو کرنل مکمل یکدیگر هستند و با استفاده از ترکیب آن ها (مثل محاسبه مقدار گرادیان کل) می توان لبه ها را در هر جهتی شناسایی کرد.

### تاثیر فیلتر `Sobel`:

#### ۱. برجسته کردن ویژگی های لبه ای

- فیلتر سوبل لبه ها و گرادیان های تصویر را برجسته می کند و مرزهای ارقام دست نویس را واضح تر نشان می دهد.
- این ویژگی می تواند برای مدل های طبقه بندی که بر روی شکل های هندسی یا کانتور تمرکز دارند مفید باشد.

#### ۲. کاهش افزونگی اطلاعات

- با استخراج فقط اطلاعات مربوط به لبه ها، فیلتر سوبل میزان اطلاعات زائد (مانند نواحی یکنواخت در ارقام) را کاهش می دهد. این کار می تواند مجموعه داده را ساده تر کرده و توانایی مدل در تعمیم بهتر را افزایش دهد.

#### ۳. افزایش مقاومت نسبت به نویز

- ارقام موجود در مجموعه داده MNIST ممکن است نویز یا اشکالاتی داشته باشند. فیلترهای تشخیص لبه مانند سوبل نسبت به این نویزها حساسیت کمتری دارند، که می تواند عملکرد مدل را در مجموعه داده های پر نویز بهبود دهد.

نمونه ی عملکرد این فیلتر روی یک نمونه دیتا که نشان میدهد لبه ها از بقیه ی بخش های عدد متمایز شده اند.



در مرحله بعد فیلتر HOG را روی دیتای خام می‌کنیم. برخلاف فیلتر سوبل که تنها به محاسبه گرادیان در دو جهت خاص  $G_x$  و  $G_y$  می‌پردازد، HOG به طور کلی‌تر از گرادیان‌ها برای ایجاد هیستوگرام استفاده می‌کند.

گرادیان افقی تصویر، تغییرات شدت پیکسل‌ها در جهت چپ به راست را محاسبه می‌کند.  
گرادیان عمودی تصویر، تغییرات شدت پیکسل‌ها در جهت بالا به پایین را اندازه‌گیری می‌کند.  
برای هر پیکسل، از مقادیر  $G_x$  و  $G_y$  دو ویژگی مهم استخراج می‌شود:

- اندازه گرادیان: (Magnitude)

- زاویه گرادیان: (Orientation)

این زاویه مشخص می‌کند که جهت تغییرات شدت پیکسل در چه جهتی است.

تصویر به بخش‌های کوچک‌تر (سلول‌ها) تقسیم می‌شود. برای هر سلول، زاویه‌های گرادیان در یک هیستوگرام قرار می‌گیرند، به طوری که محور هیستوگرام نشان‌دهنده جهت گرادیان و ارتفاع ستون‌ها نشان‌دهنده اندازه گرادیان است. این هیستوگرام‌ها خلاصه‌ای از توزیع جهت لبه‌ها در هر سلول هستند.

یک نمونه از تصویری که روی آن فیلتر Hog اعمال شده است:

HOG Features of Sample Image



در این تصویر، نواحی روشن‌تر (پیکسل‌های با شدت بیشتر) نشان می‌دهند که در آن نواحی تغییرات شدت بیشتری در جهت خاصی وجود دارد.

خطوط و لبه‌های اصلی اعداد به صورت متمرکز در بخش‌هایی از تصویر مشخص هستند. این ویژگی‌ها معمولاً مناطقی را نشان می‌دهند که اطلاعات کلیدی برای تشخیص رقم در آن نهفته است.

برای فیلتر دلخواه ما فیلتر sharpen را انتخاب کردیم زیرا از این مزایا برخوردار است:

- تقویت جزئیات: به ویژه در دیتاست MNIST که تصویر ارقام گاهی تار یا محو هستند.
- کمک به الگوریتم‌های ساده‌تر: برای مدل‌هایی مانند SVM یا KNN که به وضوح بالای ویژگی‌ها متکی هستند.
- تثبیت لبه‌ها: که به تشخیص بهتر مرزهای عدد کمک می‌کند.

هم چنین از آنجایی که باید فیچر وکتوری ترکیبی با HOG از این فیلتر تشکیل شود می‌تواند این نتیجه را در پی داشته باشد که فیلتر HOG ابتدا به استخراج جهت لبه‌ها و ویژگی‌های کلیدی تصویری می‌پردازد و فیلتر Sharpen وضوح تصویر و لبه‌ها را افزایش می‌دهد، بدون ایجاد هیستوگرام یا تحلیل جهت‌ها که ترکیب این دو می‌تواند در تشخیص بهتر اعداد به ما کمک کند.  
هم چنین قصد مقایسه ی آن با فیلتر Sobel را داشتیم چون هر دو این دو فیلتر برای استخراج لبه به کار می‌روند. به نتیجه ی مقایسه این دو فیلتر جلوتر پرداخته خواهد شد.

نمونه تصویری که فیلتر Sharpen روی آن اعمال شده است:

Original Image



Sharpen



### تغییرات ایجادشده توسط فیلتر Sharpen:

#### وضوح لبه‌ها:

- در تصویر Sharpen، لبه‌های عدد "۴" واضح‌تر شده‌اند و کنتراست بین نواحی سفید (خطوط عدد) و سیاه (زمینه) افزایش یافته است.
- این وضوح باعث می‌شود خطوط نازک یا محو، مانند بخش بالایی و گوشه پایین سمت راست عدد، برجسته‌تر شوند.

#### کنتراست زمینه:

- در تصویر Sharpen، زمینه خاکستری شده است (به جای سیاه مطلق)، اما این تغییر عمدی است و برای برجسته‌تر کردن خود عدد کمک می‌کند. چنین تغییری باعث می‌شود عدد به طور موثرتری از پس‌زمینه جدا شود.

#### افزایش شدت گرادیان:

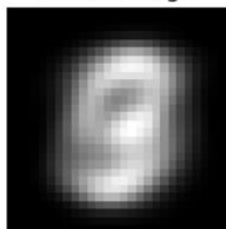
- شدت تغییرات بین پیکسل‌های تیره و روشن در تصویر Sharpen افزایش یافته است. این ویژگی به بهبود استخراج لبه‌ها و خطوط کمک می‌کند.

پس از اعمال این فیلترها فیچر وکتورهای مورد نظر برای فیچرهای حاصل از فیلتر Sobel، Sobel + HOG، Sharpen + Sobel و دیتای خام که مقادیر نرمالایز شده ی Gayscale پیکسل‌های تصاویر دیتاست را برای استفاده در فازهای بعد تشکیل می‌دهیم.

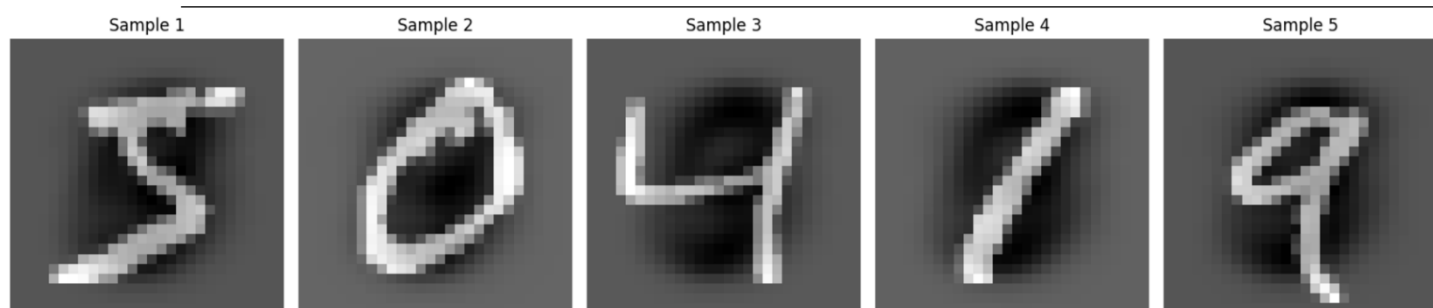
در بخش بعدی ابعاد دیتای خام و Sobel + HOG را با PCA کاهش داده و تعداد n\_components لازم برای PCA به گونه ای که بیشترین دقت را داشته باشیم با Scree Plot انتخاب می‌کنیم. برای این کار ابتدا تصویر میانگین را حساب کرده و مقادیر آن را از تصاویری که می‌خواهیم ابعادشان را کاهش دهیم کم میکنیم تا تصاویر Centered شوند.

تصویر میانگین که با گرفتن میانگین پیکسل‌های تمام تصاویر دیتاست تشکیل شده است:

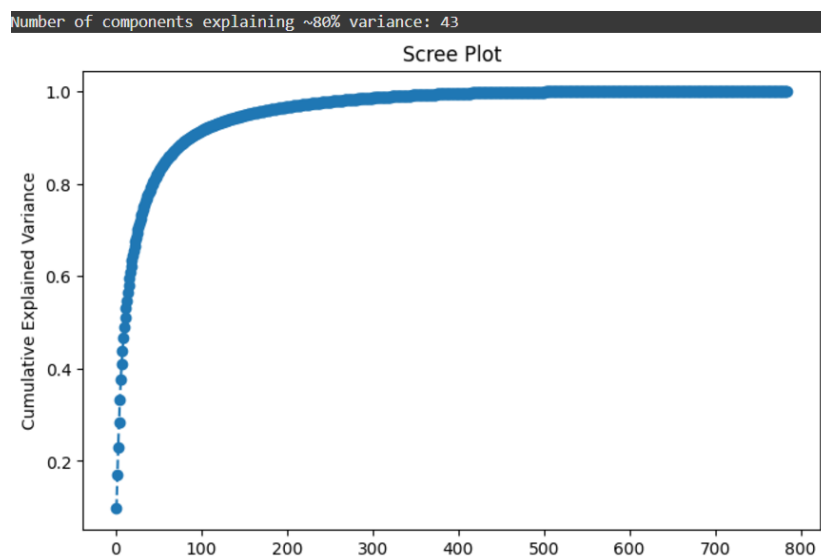
Mean Image



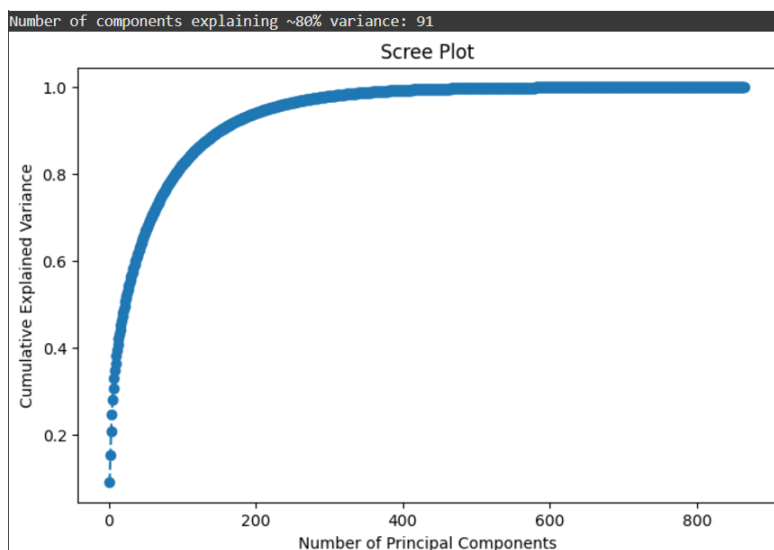
در ادامه تعدادی از تصاویر دیتاست را می بینیم که با کم کردن تصویر میانگین به دست آمده اند و این کار از این جهت مفید است که ویژگی هایی که در کلاس های مختلف مشترک است از هر کلاس کم شده و از اشتباه شدن کلاس ها با یک دیگر جلوگیری کرده و تشخیص هر کلاس با دقت بیشتری انجام خواهد شد.



Scree plot تشکیل شده برای دیتای خام و ابعاد مورد نظر برای PCA با ترشهولد ۸۰:



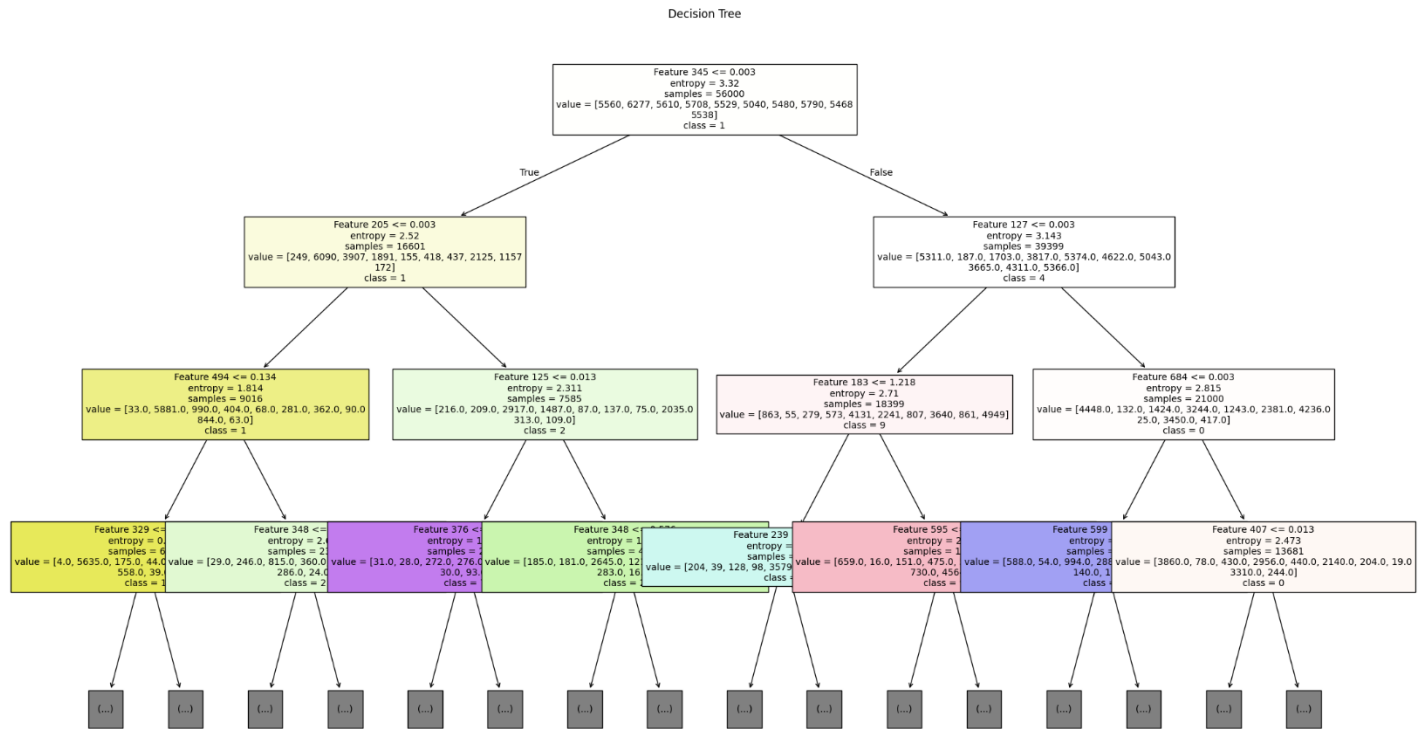
Scree plot تشکیل شده برای دیتای 'HOG+Sobel و ابعاد مورد نظر برای PCA با ترشهولد ۸۰:



دلیل بیشتر بودن تعداد component های HOG + sobel نیز بیشتر بودن تعداد ویژگی های این نوع دیتا در مقایسه با دیتای خام می باشد. در نتیجه تعداد کامپوننت بیشتری برای توصیف این ویژگی ها با در نظر گرفتن ترشهولد مناسبی از واریانس توضیح داده شده تجمعی لازم می باشد.

در فاز ۳ الگوریتم های طبقه بندی (SVM و Decision Tree) را انجام می دهیم.

Decision Tree for features extracted by Sobel :

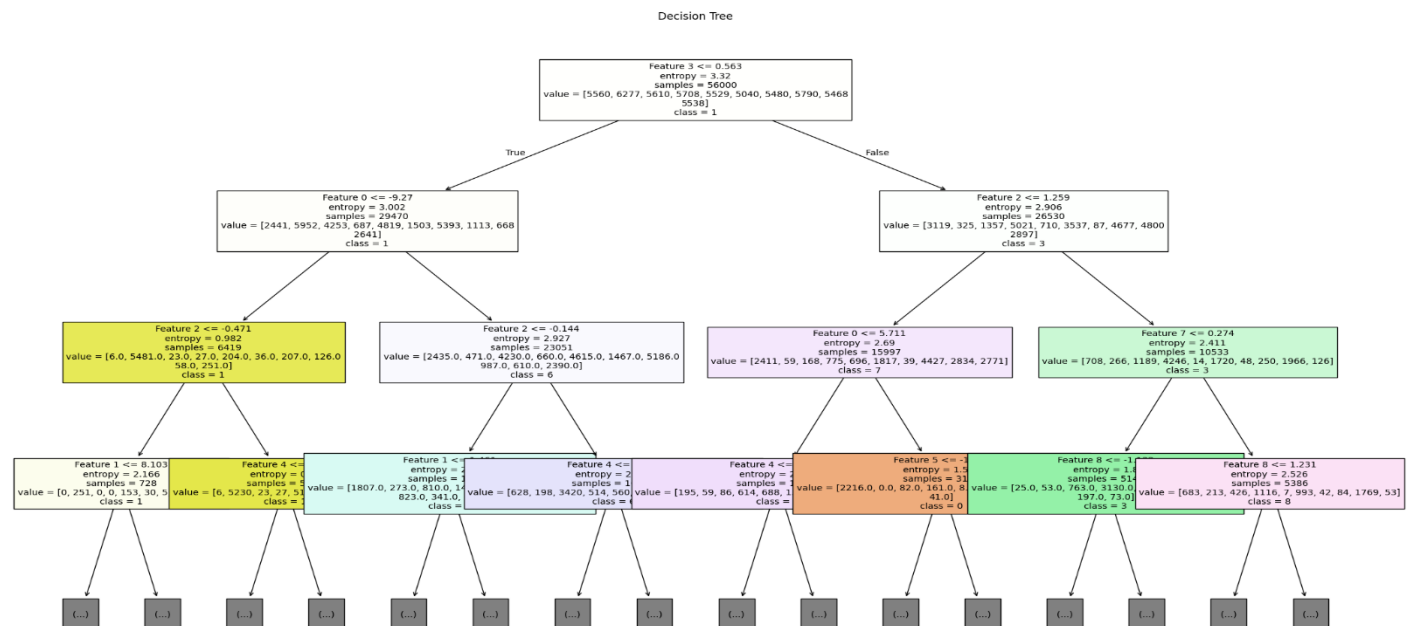


Best Hyperparameters: {'criterion': 'entropy', 'max\_depth': 15, 'min\_samples\_leaf': 7, 'min\_samples\_split': 3}

Best Accuracy (CV): 0.8459285714285713

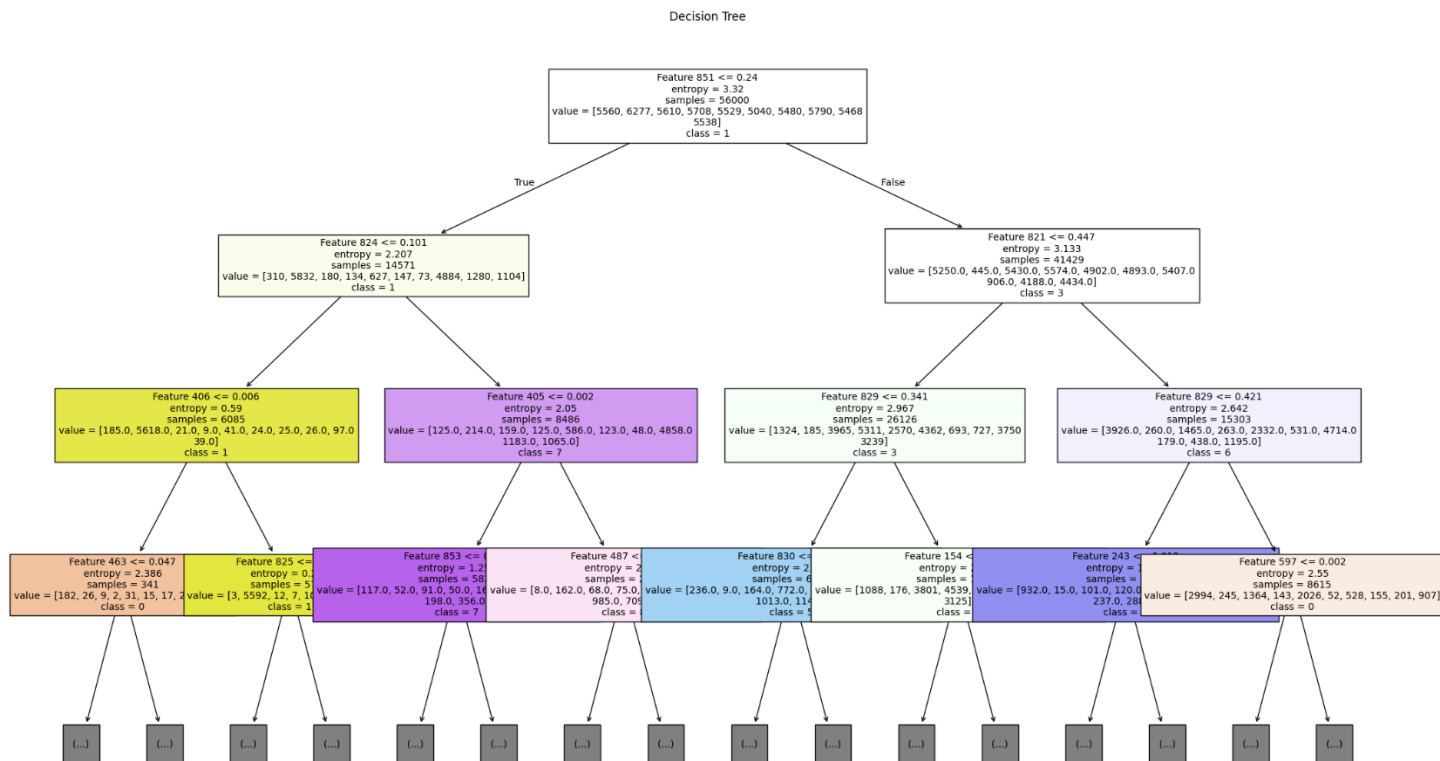
Test Accuracy: 0.8504285714285714

Decision Tree for features extracted by Sobel +HOG :



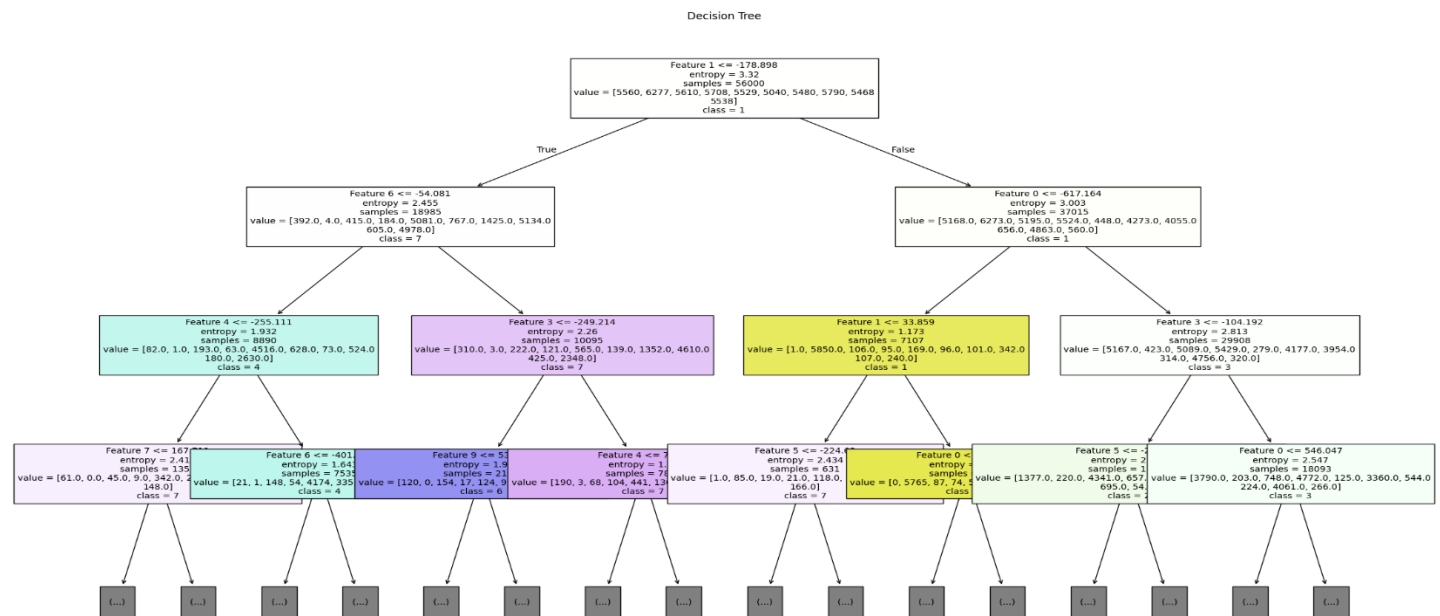
Best Hyperparameters: {'criterion': 'entropy', 'max\_depth': 15, 'min\_samples\_leaf': 7, 'min\_samples\_split': 3}  
Best Accuracy (CV): 0.8164821428571429  
Test Accuracy: 0.8283571428571429

## Decision Tree for features extracted by Sharpen +HOG :



Best Hyperparameters: {'criterion': 'entropy', 'max\_depth': 15, 'min\_samples\_leaf': 7, 'min\_samples\_split': 3}  
Best Accuracy (CV): 0.8549464285714284  
Test Accuracy: 0.8632857142857143

## Decision Tree for features extracted by Raw Data



Best Hyperparameters: {'criterion': 'entropy', 'max\_depth': 15, 'min\_samples\_leaf': 3, 'min\_samples\_split': 7}  
Best Accuracy (CV): 0.8469821428571429  
Test Accuracy: 0.8575

همین طور که قابل مشاهده است بهترین نتایج از درختی حاصل شده که فیچرهای آن از دو فیلتر **Sharpen** و **HOG** استخراج شده که نشان می دهد که تاثیر فیلتر **Sharpen** در استخراج لبه ها بیشتر از فیلتر سوبل بوده و به تشخیص اعداد بیشتر کمک کرده. هم چنین میبینیم که نتایج سوبل و **Sobel + HOG** باعث کاهش دقت تشخیص اعداد شده اند که می تواند مربوط به این باشد که :

۱ - پیچیدگی مدل: درخت تصمیم به سادگی می تواند تحت تاثیر ویژگی های اضافی و غیرضروری قرار گیرد. اگر ویژگی های استخراج شده (مانند ویژگی های سوبل و **HOG**) شامل نویز یا اطلاعات غیرمفید باشند، درخت تصمیم ممکن است دچار **overfitting** شود و نمیتواند به خوبی بر روی داده های جدید عمل کند.

۲ - تعداد ویژگی ها: استفاده از ویژگی های زیاد (مثل **HOG**) ممکن است منجر به افزایش ابعاد داده ها شود که می تواند به "معضل ابعاد" منجر شود. در این حالت، درخت تصمیم ممکن است نتواند به خوبی یاد بگیرد و دقت آن کاهش یابد.

۳ - اگر **Sobel** و **HOG** با هم استفاده شوند، ممکن است این ترکیب ویژگی ها باعث افزونگی یا ناسازگاری در داده ها شود. به عنوان مثال: **Sobel** ممکن است ویژگی های لبه ای ساده ای فراهم کند که **HOG** نتواند آن ها را به طور موثری تکمیل کند. و مدل ممکن است به اشتباه وزن بیشتری به ویژگی های کم اهمیت بدهد.

نتایج حاصل از **SVM** روی دیتا **Sobel + HOG** :

```
SVM Results for Sobel + HOG Filter:
--- Mode 1: Parameters: {'kernel': 'linear', 'c': 1} ---
Accuracy: 0.9663571428571428
Classification Report:
              precision    recall  f1-score   support

    0           0.98         0.98         0.98        1343
    1           0.98         0.99         0.98       1600
    2           0.95         0.97         0.96       1380
    3           0.96         0.95         0.96       1433
    4           0.95         0.98         0.96       1295
    5           0.96         0.96         0.96       1273
    6           0.98         0.98         0.98       1396
    7           0.97         0.96         0.97       1503
    8           0.97         0.94         0.95       1357
    9           0.96         0.95         0.95       1420

 accuracy          0.97       14000
 macro avg         0.97         0.97       14000
 weighted avg      0.97         0.97         0.97       14000
```

توضیحات عملکرد **SVM**:

- میانگین دقت در طبقات برابر با **0.97** است، که نشان می دهد پیش بینی های مدل برای هر کلاس از اعداد تا حد زیادی صحیح بوده اند.
- برای اعداد ۰، ۱، و ۶، دقت به بالاترین حد خود رسیده است. (0.98)
- عدد ۲ کمی دقت پایین تری دارد (۰.۹۵)، که ممکن است به دلیل شباهت این عدد با ارقام دیگر (مانند ۳ یا ۸) در ویژگی های استخراج شده باشد.

### Recall

- بازخوانی به طور میانگین **0.97** است، که نشان می دهد مدل در شناسایی تمامی نمونه های صحیح عملکرد خوبی داشته است.
- برای عدد ۸، بازخوانی کمی پایین تر است (۰.۹۴). این ممکن است به دلیل ویژگی های مشترک بین ارقام ۸ و دیگر اعداد (مانند ۳ یا ۹) باشد.

### F1-Score

- **F1-Score** نشان دهنده تعادل میان **Precision** و **Recall** است و مقدار میانگین آن برابر **0.97** است.

- **F1-Score** برای همه کلاس‌ها نزدیک به هم و متوازن است، که نشان می‌دهد مدل در تمام کلاس‌ها عملکرد مشابهی داشته است.

نتایج حاصل از SVM روی دیتای خام:

با توجه به اینکه SVM مانند درخت تصمیم بافیچر های حاصل از **sobel + hog** دچار **overfit** نمی شود انتظار می رود که بر خلاف درخت تصمیم دقت دیتای خام کمتر بین ۹۴ تا ۹۶ درصد باشد. که عدد حاصل ۹۶ به دست آمده است.

در مرحله بعدی به تحلیل عملکرد مدل **Sobel + HOG** برای SVM می پردازیم.

```
Loading the Decision Tree model from Drive...

Classification Report for SVM (Sobel + HOG):
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.94      | 0.95   | 0.95     | 1343    |
| 1            | 0.96      | 0.98   | 0.97     | 1600    |
| 2            | 0.89      | 0.91   | 0.90     | 1380    |
| 3            | 0.87      | 0.88   | 0.87     | 1433    |
| 4            | 0.88      | 0.90   | 0.89     | 1295    |
| 5            | 0.87      | 0.85   | 0.86     | 1273    |
| 6            | 0.95      | 0.95   | 0.95     | 1396    |
| 7            | 0.93      | 0.92   | 0.92     | 1503    |
| 8            | 0.89      | 0.86   | 0.88     | 1357    |
| 9            | 0.89      | 0.88   | 0.88     | 1420    |
| accuracy     |           |        | 0.91     | 14000   |
| macro avg    | 0.91      | 0.91   | 0.91     | 14000   |
| weighted avg | 0.91      | 0.91   | 0.91     | 14000   |

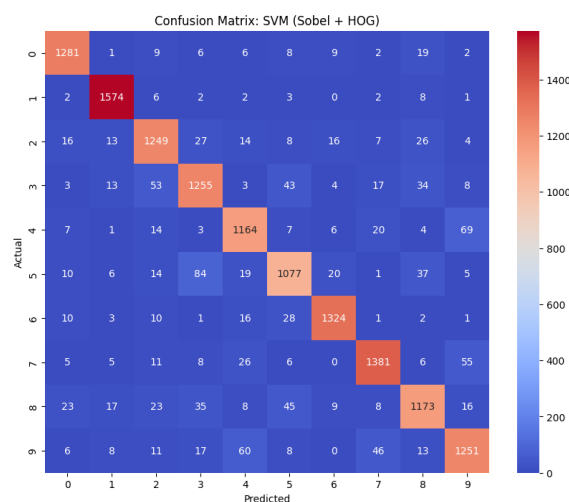
**Precision** میزان صحت پیش‌بینی‌های مدل. برای مثال، کلاس ۱ دقت بسیار بالایی (۹۶٪) دارد.

**Recall** نشان‌دهنده توانایی مدل در شناسایی تمام نمونه‌های یک کلاس خاص است. کلاس ۱ با مقدار ۹۸٪ بهترین عملکرد را دارد.

**F1-Score** میانگین هارمونیک **precision** و **recall**. این معیار برای کلاس‌هایی مثل ۱ و ۶ بالاترین مقدار (۰,۹۷ و ۰,۹۵) را دارد.

**Accuracy کلی:** مدل به دقت ۹۱٪ دست یافته است که نشان‌دهنده عملکرد بسیار خوب آن در کل مجموعه تست است.

**عملکرد کلاس‌ها:** کلاس‌های ۱ و ۶ بهترین عملکرد را دارند، در حالی که کلاس‌هایی مثل ۵ و ۸ کمی ضعیف‌تر عمل کرده‌اند.





## تحلیل نتایج confusion matrix

خطاها: در کلاس‌هایی مثل ۳ و ۵ خطاهای قابل‌توجهی در پیش‌بینی دیده می‌شود (اشتباهات متقابل بیشتری دارند).

تشخیص‌های درست (قطر اصلی ماتریس): اکثر پیش‌بینی‌ها درست بوده‌اند و مقادیر بالای قطر اصلی نشان‌دهنده عملکرد خوب مدل است.

اشتباهات خاص: مثلاً در کلاس ۹، اشتباهات بیشتر به کلاس‌های ۴ و ۸ نسبت داده شده است.

تعادل بین کلاس‌ها: مدل به خوبی میان اکثر کلاس‌ها تعادل برقرار کرده است، اما ممکن است برای بهبود کلاس‌های با دقت پایین‌تر (مثل ۵ و ۸) نیاز

به داده‌های بیشتری باشد.

## حلیل عملکرد مدل درخت تصمیم:

```
Loading the Decision Tree model from Drive...
Evaluating the loaded Decision Tree model...

Classification Report for Decision Tree (Sobel + HOG):
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.87      | 0.89   | 0.88     | 1343    |
| 1            | 0.94      | 0.95   | 0.95     | 1600    |
| 2            | 0.82      | 0.82   | 0.82     | 1380    |
| 3            | 0.74      | 0.74   | 0.74     | 1433    |
| 4            | 0.81      | 0.82   | 0.81     | 1295    |
| 5            | 0.78      | 0.75   | 0.76     | 1273    |
| 6            | 0.92      | 0.92   | 0.92     | 1396    |
| 7            | 0.87      | 0.84   | 0.85     | 1503    |
| 8            | 0.74      | 0.73   | 0.74     | 1357    |
| 9            | 0.77      | 0.80   | 0.78     | 1420    |
| accuracy     |           |        | 0.83     | 14000   |
| macro avg    | 0.83      | 0.83   | 0.83     | 14000   |
| weighted avg | 0.83      | 0.83   | 0.83     | 14000   |

## تحلیل گزارش طبقه‌بندی Decision Tree (Sobel + HOG):

دقت کلی (Accuracy): Decision Tree (Accuracy) دقت کلی ۸۳٪ داشته، که کمتر از دقت SVM (۹۱٪) است.

عملکرد کلاس‌ها:

کلاس‌هایی مثل ۱ و ۶ بهترین عملکرد را دارند (با F1-Score به ترتیب ۰/۹۵ و ۰/۹۲).

کلاس‌هایی مثل ۳، ۵ و ۸ عملکرد ضعیف‌تری دارند (F1-Score حدود ۰/۷۴ تا ۰/۷۶).

Precision و Recall برای کلاس‌هایی مثل ۳ و ۸ پایین‌تر هستند که نشان‌دهنده ضعف مدل در شناسایی و پیش‌بینی صحیح این کلاس‌هاست.

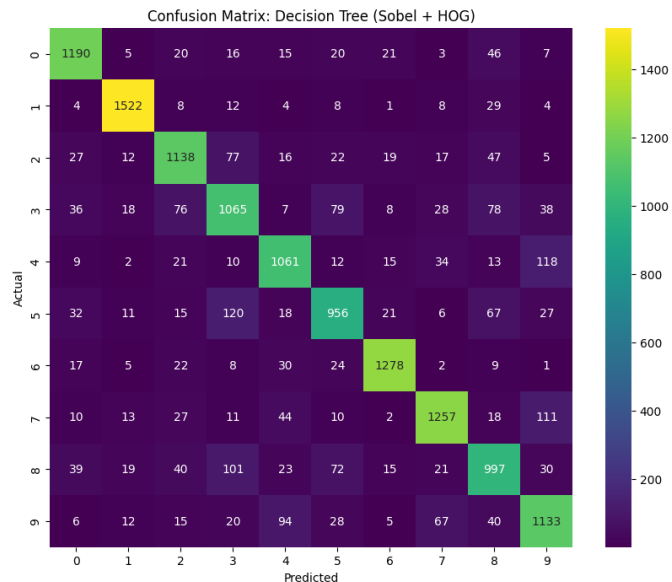
## تحلیل نتایج confusion matrix

اشتباهات متقابل: در کلاس‌هایی مثل ۳ و ۵، مدل اشتباهات متقابل بیشتری دارد (پیش‌بینی اشتباه نمونه‌های کلاس‌های دیگر به‌عنوان این کلاس‌ها).

تشخیص‌های درست (قطر اصلی ماتریس): مقدار درستی در قطر اصلی ماتریس برای Decision Tree کمتر از SVM است.

خطاهای خاص: برخی کلاس‌ها (مثل ۸ و ۹) به کلاس‌های مجاور خود اشتباه پیش‌بینی شده‌اند، که نشان‌دهنده همپوشانی ویژگی‌ها در این

کلاس‌هاست.



در مرحله ی آخر نیز به دنبال جلوگیری از اورفیت روی درخت تصمیم هستیم. که این کار را با تکنیک **pre pruning** انجام داده ایم.

**تحلیل نتایج حاصل:**

۱. **Best Model:**

- دقت کلی آزمون **82.84%** که نشان دهنده عملکرد قابل قبول مدل است.
- تفاوت بین دقت آموزش و آزمون کم است که به معنای عدم بیش برآزش (**Overfitting**) می باشد.
- بهترین دقت و تعادل بین داده های آموزش و آزمون را دارد.

۲. **Overfitted Decision Tree:**

- دقت آموزش **100%** که نشان دهنده بیش برآزش (**Overfitting**) مدل روی داده های آموزشی است.
- دقت آزمون **81.46%** است که کمتر از سایر مدل هاست، زیرا مدل روی داده های جدید تعمیم پذیری کمتری دارد.
- استفاده از تنظیمات پیش فرض (مانند عمق نامحدود) باعث شده مدل به جزئیات غیر ضروری داده حساس شود.

۳. **Pruned Decision Tree:**

- دقت آزمون **81.99%** که کمی بهتر از مدل **Overfitted** است.
- به دلیل اعمال محدودیت هایی مثل عمق حداکثر و حداقل تعداد نمونه، عملکردی پایدارتر نسبت به مدل **Overfitted** دارد.
- دقت آموزش **91.35%** به معنای کاهش اندکی در عملکرد آموزش برای جلوگیری از **Overfitting** است.

تحلیل جزئی تر بر اساس گزارش های طبقه بندی:

- دسته هایی که مدل ها در آن ها بهترین عملکرد را داشته اند:
  - کلاس 1 و 6 با **Precision** و **Recall** بالا در همه مدل ها.
- دسته هایی با عملکرد ضعیف تر:
  - کلاس های 3، 5، و 8 با **F1-Score** کمتر به دلیل هم پوشانی ویژگی ها.

