



گزارش پروژه دوم
(مبانی هوش محاسباتی)

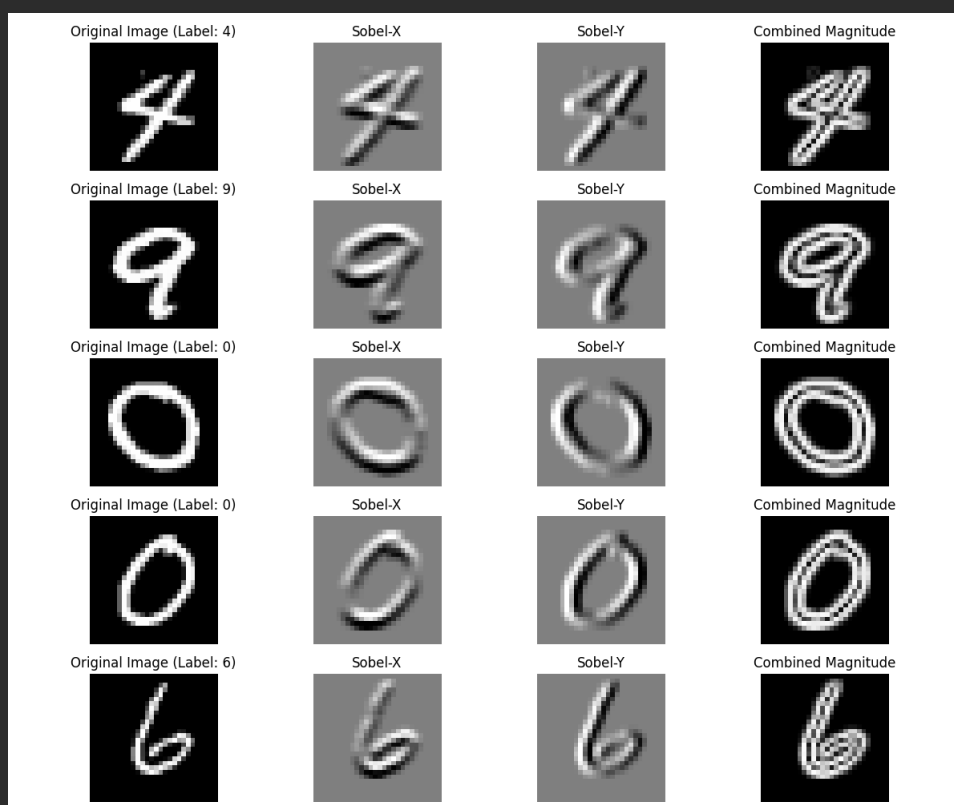
مهدی طاهری ۴۰۰۱۲۶۲۱۳۷

ایمان موقرمقدم ۴۰۰۱۲۶۲۱۱۵

آذر ۱۴۰۳

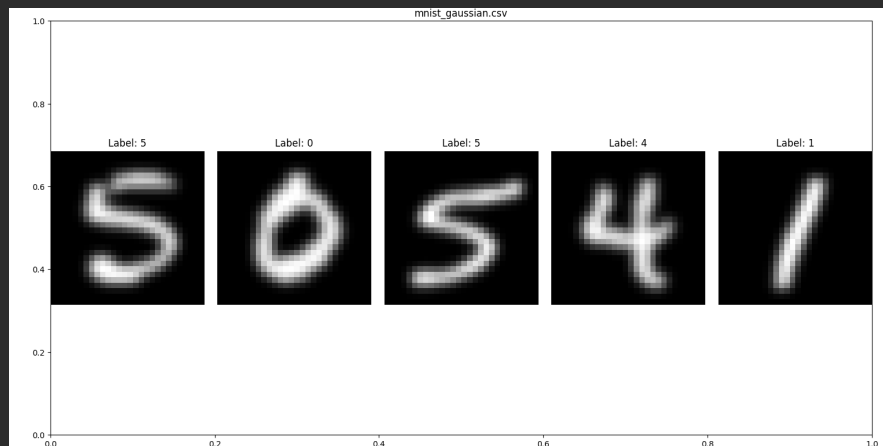
(۱) فاز اول

- **Sobel** : فیلتر سوبل اساساً برای تشخیص لبه استفاده می شود. این فیلتر با تمرکز بر لبه ها می تواند به الگوریتم های طبقه بندی کمک کند تا بر اشکال و مرزهای اشیاء درون یک تصویر تمرکز کنند. این می تواند عملکرد طبقه بندی کننده ها را بهبود بخشد، به ویژه در کارهایی که ساختار اشیاء حیاتی است (به عنوان مثال، تشخیص شی، تقسیم بندی).
- **HOG** : HOG یک توصیفگر ویژگی است که توزیع جهت گیری گرادینان را در بخش های محلی یک تصویر ثبت می کند و به ویژه برای تشخیص اشیا موثر است. ویژگی های HOG یک بازنمایی قوی از شکل و ظاهر اشیاء ارائه می دهد و آنها را برای طبقه بندی کننده هایی مانند SVM مفید می کند. HOG می تواند دقت طبقه بندی را با ارائه مجموعه ای از ویژگی های آموزنده تر که نسبت به تغییرات در روشنایی و تغییر شکل های کوچک تغییر نمی کند، بهبود بخشد.
- **Gaussian** : فیلتر Gaussian برای صاف کردن تصاویر و کاهش نویز استفاده می شود. این یک تابع گاوسی را روی تصویر اعمال می کند که آن را تار می کند و به حذف نویز فرکانس بالا کمک می کند. با صاف کردن تصویر، فیلتر گاوسی می تواند به بهبود الگوریتم های طبقه بندی از طریق کاهش نویز و جزئیات نامربوط کمک کند.

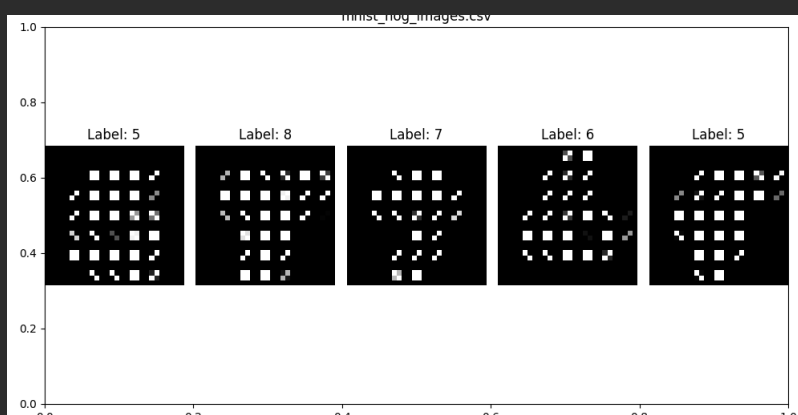


شکل ۱: از چپ به راست تصویر خام و فیلترهای Sobel

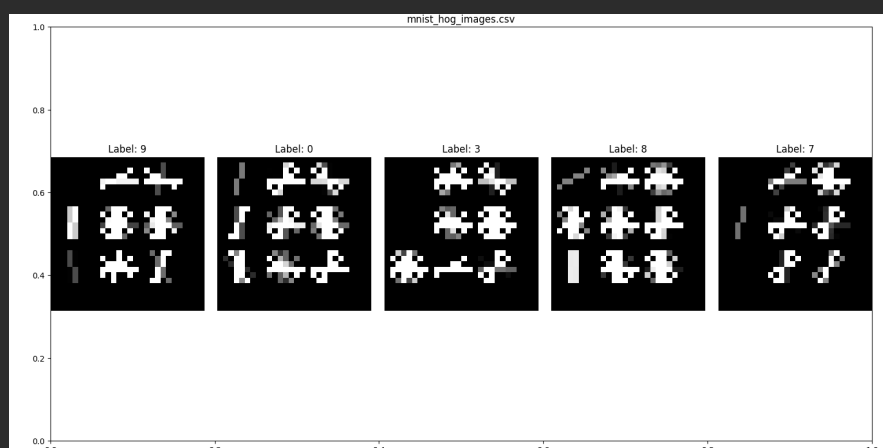
مطابق شکل ۱، Sobel x حاشیه های افقی تصویر و Sobel y حاشیه های عمودی و Sobel G (gradient magnitude) حاشیه ها را در حالت کلی بدست می آورد. فیلتر انتخابی Gaussian نیز مطابق شکل ۲ به تصاویر حالت smooth مانند می دهد و نویز ها را حذف می کند و لبه های تصاویر را نرم می کند.



شکل ۲: Gaussian



شکل ۳: HOG with cell per block: 2*2



شکل ۴: HOG with cell per block: 4*4 (chosen one)

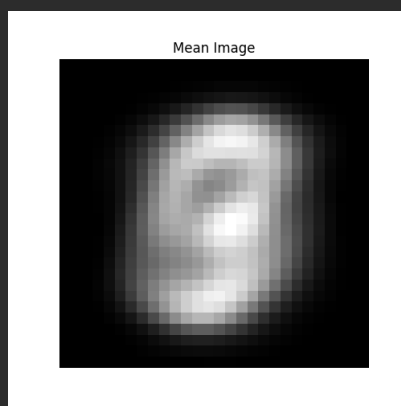
برای بهبود HOG از تکنیک نرمالسازی L₂-Hys استفاده شد. تا بردار ویژگی های HOG در برابر تغییرات در روشنایی و کنتراست مقاوم تر شود.

(۲) فاز دوم

کاهش تصویر میانگین از کل تصاویر یک عمل رایج در پیش پردازش داده های تصویر است. اگر این کار را انجام دهیم، به این نتایج خواهیم رسید:

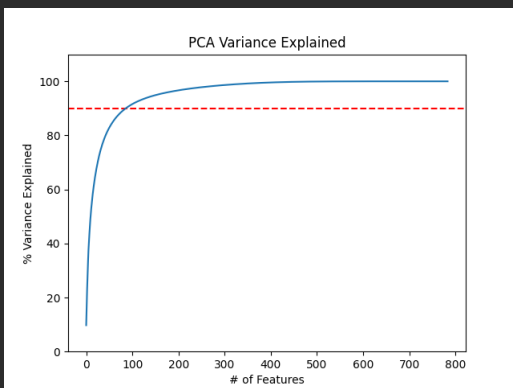
۱. مرکز کردن داده ها: کاهش تصویر میانگین باعث می شود که هر تصویر به گونه ای تغییر یابد که محور اعداد حول میانگین مجموعه داده ها بچرخد. این کار به کاهش واریانس در داده ها کمک می کند و آن ها را برای مدل های یادگیری ماشین قابل پیش بینی تر می سازد.

۲. کاهش نویز: کاهش میانگین تصویر می تواند برخی از نویزها و انحرافات عمومی را که در مجموعه داده ها وجود دارند، حذف کند و ویژگی های اصلی و متفاوت تر ارقام را برجسته سازد. تصویر زیر به عنوان میانگین بدست آمد.

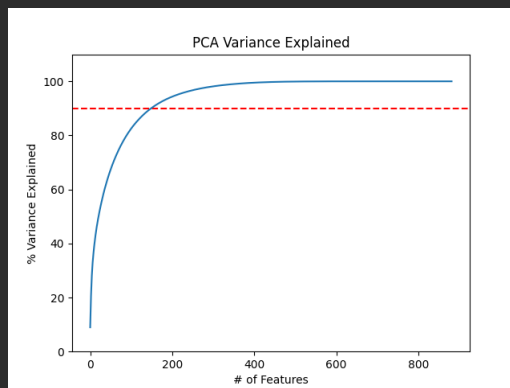


شکل ۵: تصویر میانگین

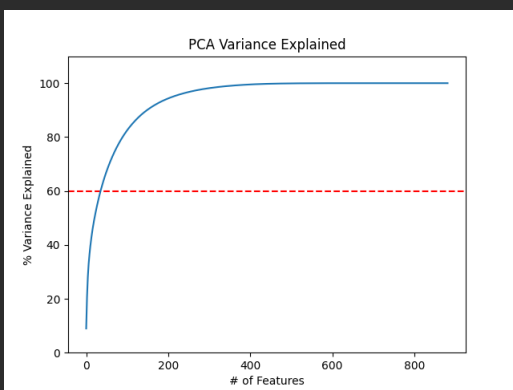
برای کاهش ابعاد از طریق PCA، با توجه به شکستگی نمودار در نزدیکی ۹۰ در ابتدا با ترشولد ۹۰ برای n_component مقدار ۱۵۰ بدست آمد (شکل ۶) و سپس ترشولد را کاهش دادیم و به حدود ۶۰ و n_component ۴۰ رسیدیم (شکل ۸). دلیل کاهش ترشولد به این دلیل بود که با ۱۵۰ فیچر مدت زمان اجرای svm در فاز ۳ بسیار بالا میبود. در صورتی که با کاهش تعداد فیچرها به ۴۰ بازهم به دقت بسیار مناسب ۹۸ درصدی برای دیتای خام و ۹۶ درصدی برای sobel+hog دست پیدا کردیم. در نتیجه کار کردن با ابعاد بالاتر منطقی نبود.



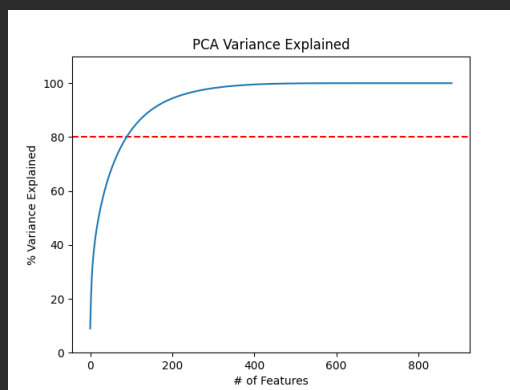
شکل ۷: خام



شکل ۶: HOG + Sobel



شکل ۹: HOG + Sobel



شکل ۸: HOG + Sobel

۳) فاز سوم

ابتدا با گزیدن سرچ پارامترهای مناسب درخت پیدا شد.

```
param_grid = {
    'max_depth': [None, 10, 15, 20],
    'min_samples_split': [5, 10, 20],
    'criterion': ['gini', 'entropy', "log_loss"],
    'min_samples_leaf': [1, 2, 5, 10]
}
```

شکل ۱۰: search grid

دقت درخت تصمیم با فیلترهای مختلف به صورت زیر است:

```
Best Hyperparameters: {'criterion': 'entropy', 'max_depth': 15, 'min_samples_leaf': 5, 'min_samples_split': 5}
Best Accuracy (CV): 0.8475535714285714
Test Accuracy: 0.8539285714285715
```

شکل ۱۱: DT با دیتای raw

```
Best Hyperparameters: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 5, 'min_samples_split': 5}
Best Accuracy (CV): 0.8224642857142858
Test Accuracy: 0.8335
```

شکل ۱۲: DT با دیتای Sobel

```
Best Hyperparameters: {'criterion': 'entropy', 'max_depth': 15, 'min_samples_leaf': 5, 'min_samples_split': 5}
Best Accuracy (CV): 0.8249285714285713
Test Accuracy: 0.8312857142857143
```

شکل ۱۳: DT با دیتای Sobel+HOG

```
Best Hyperparameters: {'criterion': 'entropy', 'max_depth': 15, 'min_samples_leaf': 5, 'min_samples_split': 5}
Best Accuracy (CV): 0.8538392857142856
Test Accuracy: 0.8601428571428571
```

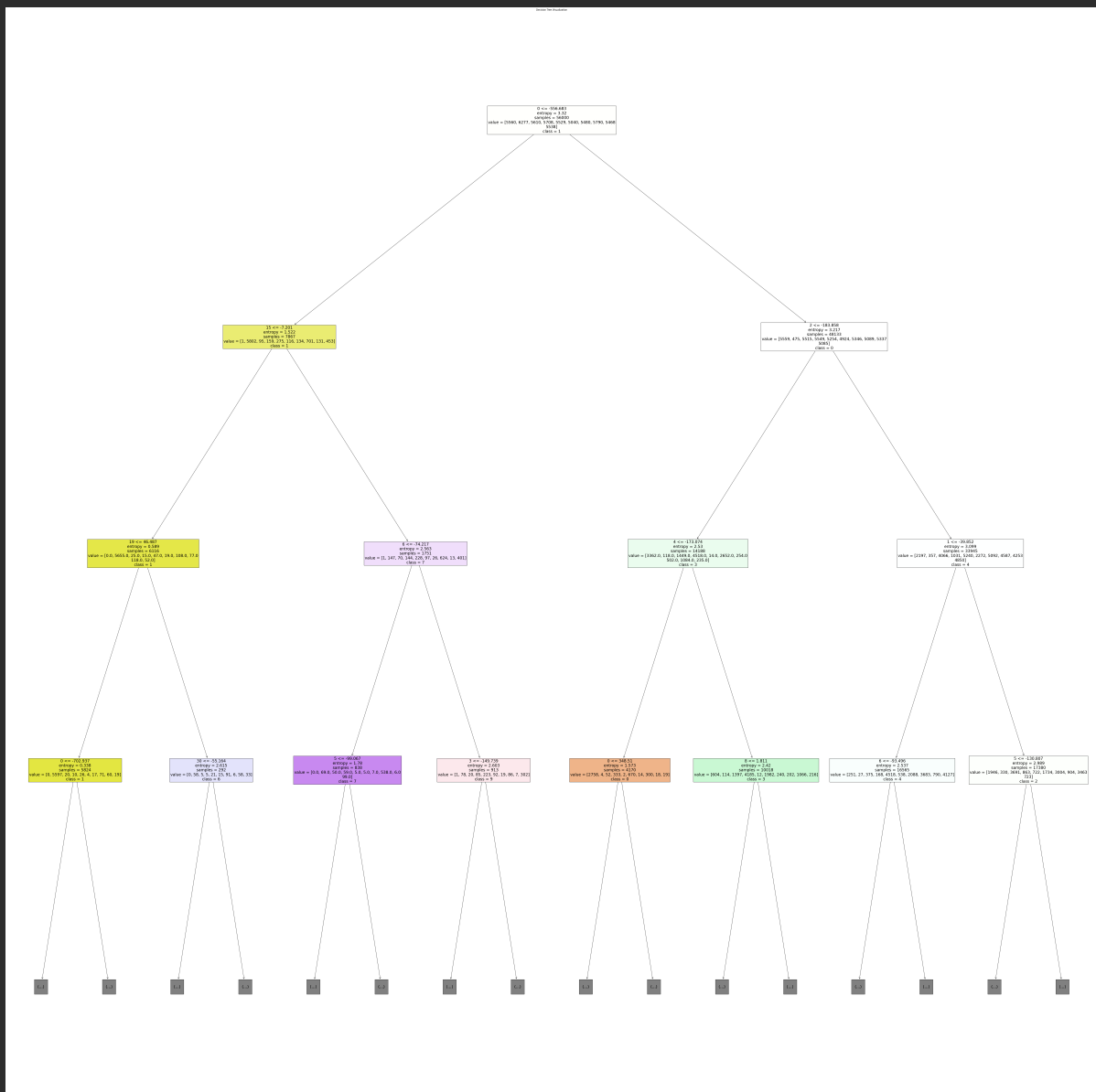
شکل ۱۴: DT با دیتای Gaussian+HOG

و همچنین دقت SVM به صورت زیر:

```
Accuracy of svm for sobel_hog_reduced: 0.96
Accuracy of svm for raw_reduced: 0.98
```

شکل ۱۵: SVM با دیتای raw و Sobel+HOG

همانطور که دیده میشود دقت SVM از DT به مراتب بیشتر است.



شکل ۱۶: DT

با توجه به نمودار شکل ۱۶ مشخص است که ابتدا همه داده ها به عنوان کلاس ۱ در نظر گرفته شده اند و پس از آن کلاس شماره ۰ در تفکیک اثرگذاری بیشتری داشته است سپس کلاس ۴ ... همچنین در مورد تاثیر فیچرها در تفکیک میتوان گفت که فیچر ۰ بیشترین تاثیر و پس از آن ۲ و ۱۵ بیشترین تاثیر را داشته اند.

۴) فاز چهارم

ریپورت ارزیابی مربوط به درخت تصمیم:

Classification Report for Decision Tree:

Class 0 -> Precision: 0.8664 , Recall: 0.8935 , F1-score: 0.8798

Class 1 -> Precision: 0.9412 , Recall: 0.9600 , F1-score: 0.9505
 Class 2 -> Precision: 0.8342 , Recall: 0.8239 , F1-score: 0.8290
 Class 3 -> Precision: 0.7354 , Recall: 0.7390 , F1-score: 0.7372
 Class 4 -> Precision: 0.7792 , Recall: 0.8448 , F1-score: 0.8107
 Class 5 -> Precision: 0.7797 , Recall: 0.7313 , F1-score: 0.7548
 Class 6 -> Precision: 0.9144 , Recall: 0.8954 , F1-score: 0.9048
 Class 7 -> Precision: 0.8631 , Recall: 0.8430 , F1-score: 0.8529
 Class 8 -> Precision: 0.7636 , Recall: 0.7568 , F1-score: 0.7602
 Class 9 -> Precision: 0.7953 , Recall: 0.7852 , F1-score: 0.7902

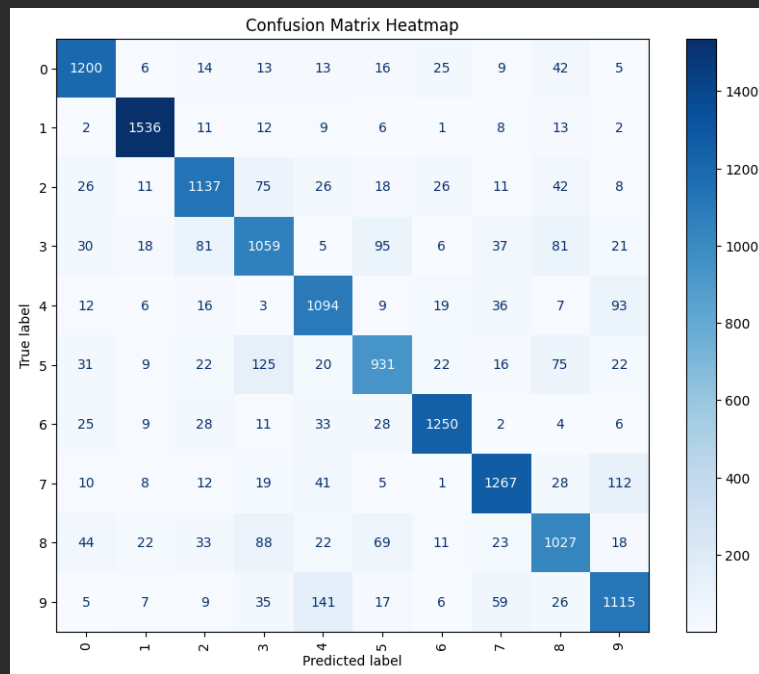
ریپورت ارزیابی مربوط به SVM :

Classification Report for SVM:

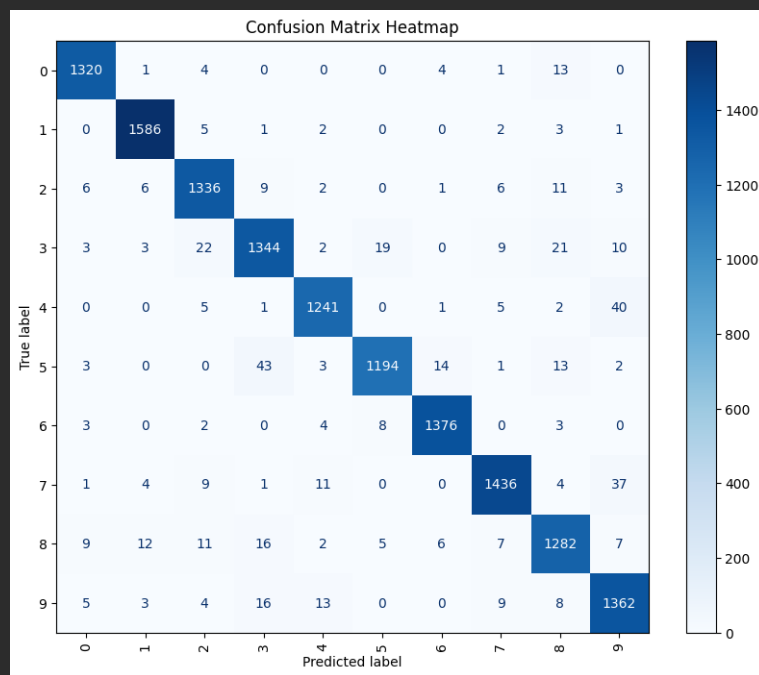
Class 0 -> Precision: 0.9778 , Recall: 0.9829 , F1-score: 0.9803
 Class 1 -> Precision: 0.9820 , Recall: 0.9912 , F1-score: 0.9866
 Class 2 -> Precision: 0.9557 , Recall: 0.9681 , F1-score: 0.9618
 Class 3 -> Precision: 0.9392 , Recall: 0.9379 , F1-score: 0.9385
 Class 4 -> Precision: 0.9695 , Recall: 0.9583 , F1-score: 0.9639
 Class 5 -> Precision: 0.9739 , Recall: 0.9379 , F1-score: 0.9556
 Class 6 -> Precision: 0.9815 , Recall: 0.9857 , F1-score: 0.9836
 Class 7 -> Precision: 0.9729 , Recall: 0.9554 , F1-score: 0.9641
 Class 8 -> Precision: 0.9426 , Recall: 0.9447 , F1-score: 0.9437
 Class 9 -> Precision: 0.9316 , Recall: 0.9592 , F1-score: 0.9452

مطابق confusion matrix مربوط به درخت تصمیم (شکل ۱۷) بیشترین مقدار خطا مربوط به ۹ و ۴ است به این صورت که ۱۴۱ تصویر با لیبل ۹ به اشتباه ۴ تشخیص داده شده اند که این به نوعی میتواند قابل توجه باشد زیرا که این دو عدد شباهت بسیار زیادی دارند (هر دو متشکل از یک دایره مانندی چسبیده به سمت چپ بالای یک خط عمودی هستند) به همین ترتیب ۱۲۵ تصویر با لیبل ۵ به اشتباه ۳ تشخیص داده شده که احتمالا این تصاویر به دلیل شباهت در نصفه پایین و کمی شباهت در نصفه بالا مثل هم تشخیص داده شده اند . بقیه اعداد هم به همین صورت البته باید توجه داشت که اگر اعداد در مرکز تصاویر قرار نداشته باشند ، ممکن است مدل یک پیش بینی تصادفی بدون هیچ منطقی داشته باشد.

با توجه به confusion matrix مربوط به SVM (شکل ۱۸) خطاهای درخت تصمیم مشاهده نمیشود و دقت کار به طور قابل توجهی بالاتر بوده است ولی بازهم آن مشکل عدم قرارگیری عدد در مرکز تصویر یا حتی ناخوانا بودن عدد میتواند باعث این بوده باشد که اعداد به طور اشتباه پیش بینی شوند .



شکل ۱۷: DT



شکل ۱۸: SVM

(۵) فاز پنجم

پارامترهای دو مدل overfit شده و pruned به صورت زیر است:

```
overfit_clf = DecisionTreeClassifier(max_depth=None, min_samples_leaf=1,  
min_samples_split=2, random_state=42)  
  
pruned_clf = DecisionTreeClassifier(criterion='entropy', max_depth= 15,  
min_samples_leaf= 5, min_samples_split= 5, random_state=42)
```

شکل ۱۹

همانطور که مشاهده میشود در حالت اورفیت هیچ محدودیتی روی عمق درخت نگذاشتیم و مینیمم تعداد سمپل ها در برگها نیز نسبت به حالت pruned کمتر است. مقایسه دقت train و test این دو حالت نیز در زیر آورده شده:

```
=== Overfitted Model Results ===  
Training Accuracy: 1.00  
Testing Accuracy: 0.82  
  
=== Pruned Model Results ===  
Training Accuracy: 0.85  
Testing Accuracy: 0.81
```

شکل ۲۰

مطابق شکل ۲۰ در حالت اورفیت اختلاف زیادی بین دقت train و test وجود دارد که در حالت pruned اینگونه نیست و نشان میدهد که مدل در این حالت به خوبی generalization را انجام میدهد.