COM2028 - Introduction to Artificial Intelligence

Filbert Miko Linardi
FL00296 - 6546907

Introduction

The use of Artificial Intelligence is very beneficial in this day of age as we can see its application being used by the medical field, businesses, surveillance system and many more. One of its applications is to build a computer vision system that can interpret and understand the visual world as it is able to classify objects

The main purpose of this research is to build a computer vision system that can be used as an OCR (Optical Character Recognition) by using different machine learning techniques and then evaluating them based on their performance. This will be done by developing a multi-class image classification computer vision system. The system must be able to successfully do image processing, boundary and line extraction, segmentation and feature extraction. These features will then be used as the training data for the model.

Objectives

- Develop a multi-class image classification computer vision system that can extract, classify and predict an object and its features for each task
- To understand the workflow of developing a useful computer vision system
- To develop machine learning techniques and evaluate their performance

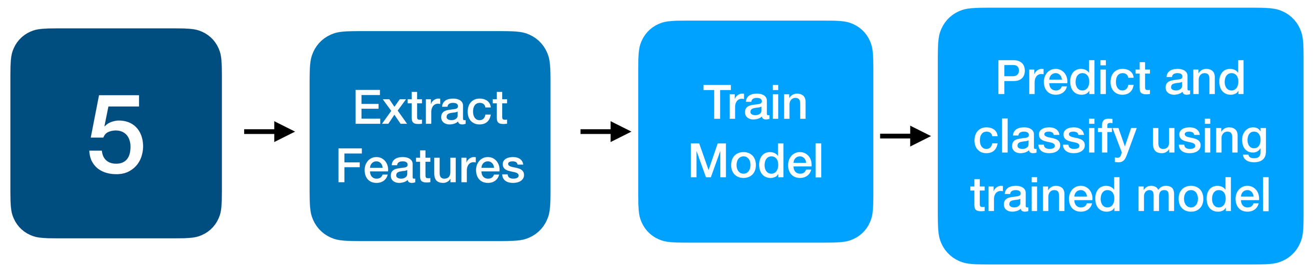


Figure 1

Tasks

This research focuses on how the selected machine learning techniques are be able to extract, classify and predict:

- Printed characters on an image in English and in a foreign language
- Handwritten characters on an image in English
- Signatures on an image

Methodology

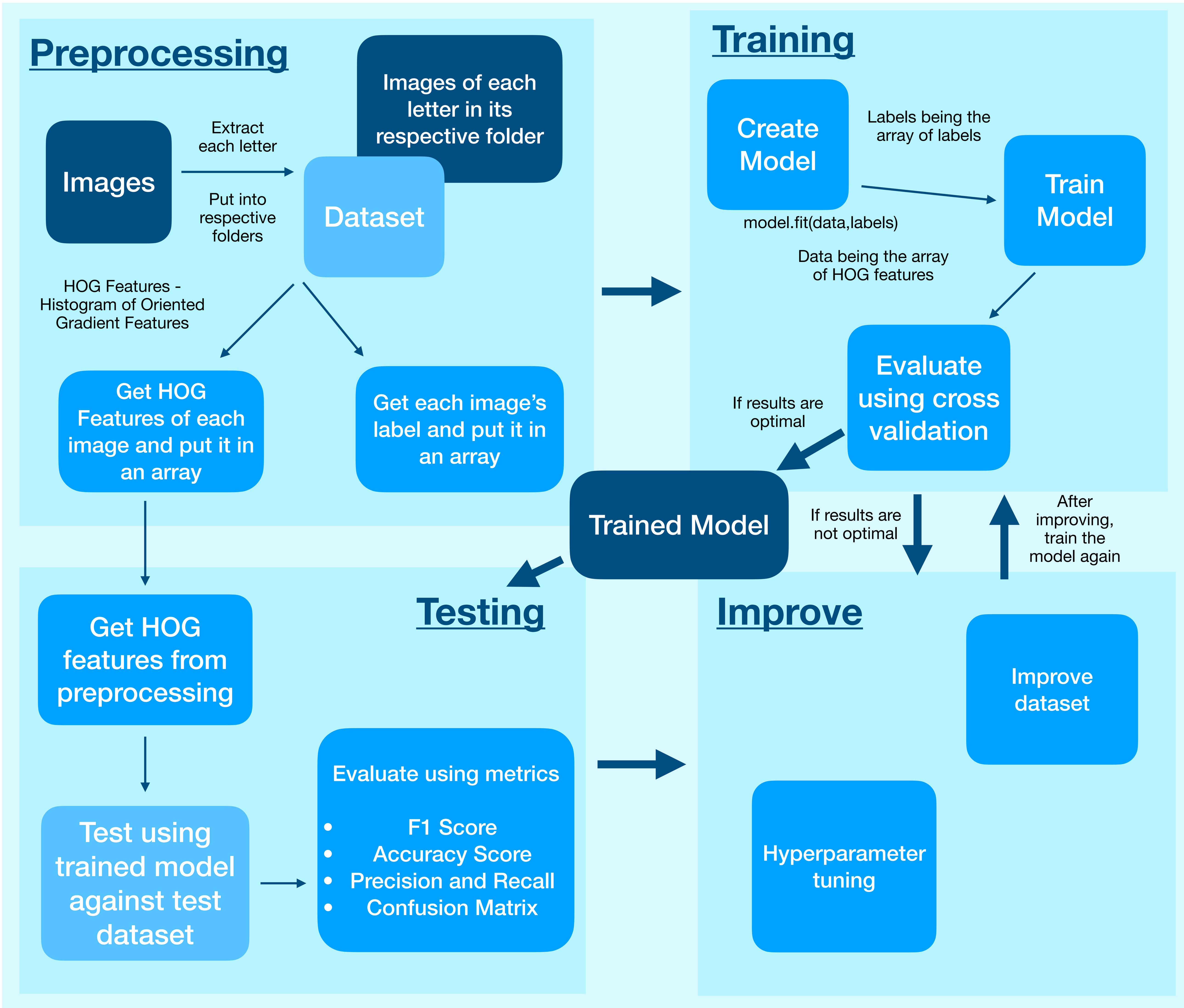


Figure 2

Challenges

- Extractor is not optimal. It can miss few of the characters as well as picking up unwanted characters like a parentheses or a colon
- Extractor couldn't properly pick up the tittles on i's and j's so they could both be mistaken for each other as well as an 'l'
- Training phase may not be optimal due to datasets issues such as lack thereof or having too much noise which can lead to faulty feature extraction process
- Parameter tuning can take several configurations for each machine learning technique to work optimally

Dataset

- Printed English - **108** for each alphabet and **54** for each number from 1-9 making it **3294** total images
- Printed Greek - **31** for each alphabet and there are 24 alphabets which makes it **744** total images
- Handwritten English - **26** for each alphabet and **12** for each number from 1-9 making it **784** total images
- Signatures - 3 people with **29** signatures each making it **87** total images

Implementation

SVM

- Support Vector Machine is a supervised machine learning algorithm that can be used as a discriminative classifier that uses a separating hyperplane to classify the tags
- The larger the margins from both tags the more optimal the hyperplane
- Performs well when the margin of separation is clear and when the dataset is not large which makes it suitable for the tasks of this project but does not perform very well when there is too much noise

KNN

- K-Nearest Neighbour is a supervised machine learning algorithm that can be used as a classifier
- It classifies the data points by finding the distances between a query and all the examples in the data to determine the K-nearest neighbours and then choosing the most frequent label
- Suitable for the tasks of this project as it works better with smaller dataset but this can also be a drawback when a larger dataset is used

MLP

- Multilayer Perceptron Neural Network is a class of feedforward artificial neural network. It is composed of layers of perceptrons
- It uses an underlying Neural Network to perform the task of classification
- Can be used to learn a mapping from inputs to outputs

- SVM, KNN and MLP will all be used in each of the task.

Files needed for the implementation:

- An extractor class
- A class that will prepare the data (extract each letter and put them into its respective folders)
- A class to train the models classify the objects
- A class to evaluate and test the trained models.
- Each model will be tested with 2 images

COM2028 - Introduction to Artificial Intelligence

Filbert Miko Linardi
FL00296 - 6546907

Implementation

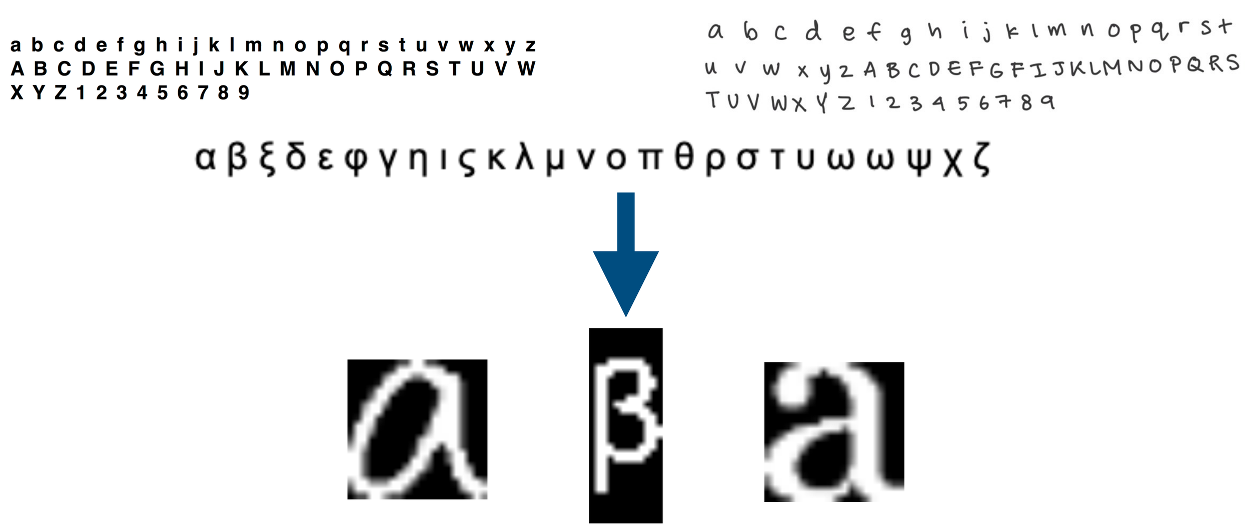
Building the Dataset

- Used a page full of alphabets and extracted it using the extractor class and putting each letter into its respective folder using the class which prepares the data
- Looping through each folder to get all of the file names that exist in training_type and then adding the added the path as well to the name of the folder and store it into an array. The code can be seen in Figure 3 in the Appendix
- Loop through the array and get each image's HOG features and label and store them into their respective arrays. The code can be seen in Figure 4 in the Appendix

Figure 3

Figure 4

Dataset Examples



Extracted letters

Cross Validation

- K-fold evaluation is used to evaluate each machine learning model. I will be using a 5-fold cross validation
- Cross Validation using K-Fold evaluation is used to find the mean accuracy of each machine learning model using the training dataset. It will loop through 5 times and each time it loops through the dataset, a block of the dataset will be locked and be used as the validation set so that each loop will have different validation set
- The result will tell us how each machine learning model work against our data
- It will also be able to tell us if a model is overfitting or underfitting

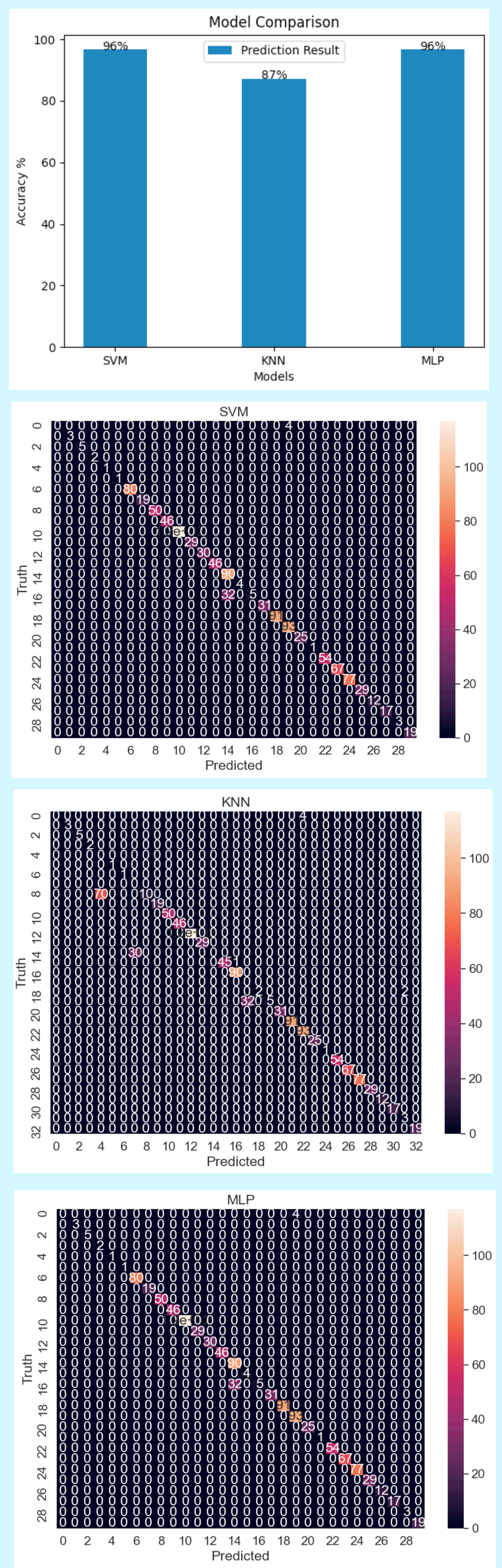
```python
#Getting Mean Accuracy from Cross Val Score
models = []
models.append(('SVM', clf))
models.append(('KNN', knnC))
models.append(('MLP', mlp)))

results = []
names = []
mean = []
for name, model in models:
    kfold = KFold(n_splits=5, random_state=12, shuffle=True)
    cv_results = cross_val_score(model, data, labels, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    output_message = "%s| Mean=%f STD=%f" % (name, cv_results.mean(), cv_results.std())
    mean.append(cv_results.mean()*100)
    print(output_message)
```

Figure 5 - Implementing Cross Validation using K-Fold

Results

Printed English

Image 1

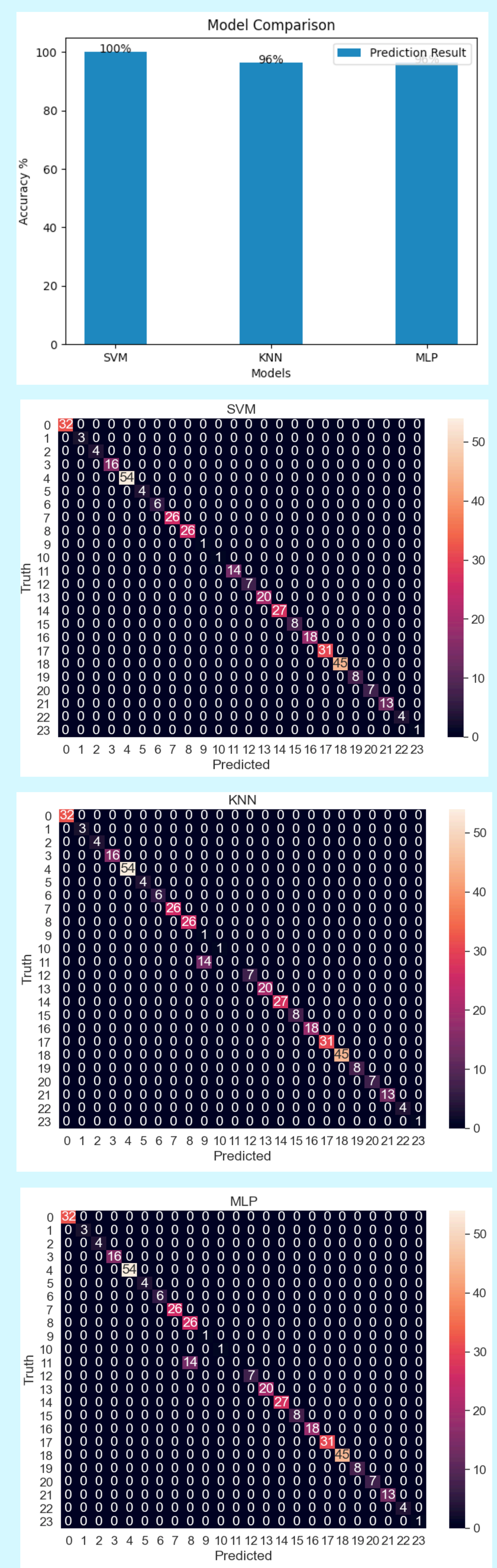


```
SVM Classification Report
Accuracy Score: 96.67590027700831
F1 Score: 96.67590027700831
Precision Score: 96.67590027700831
Recall Score: 96.67590027700831
```

```
KNN Classification Report
Accuracy Score: 87.16528162511543
F1 Score: 87.16528162511543
Precision Score: 87.16528162511543
Recall Score: 87.16528162511543
```

```
MLP Classification Report
Accuracy Score: 96.67590027700831
F1 Score: 96.67590027700831
Precision Score: 96.67590027700831
Recall Score: 96.67590027700831
```

Image 2



```
SVM Classification Report
Accuracy Score: 100.0
F1 Score: 100.0
Precision Score: 100.0
Recall Score: 100.0
```

```
KNN Classification Report
Accuracy Score: 96.27659574468085
F1 Score: 96.27659574468085
Precision Score: 96.27659574468085
Recall Score: 96.27659574468085
```

```
MLP Classification Report
Accuracy Score: 96.27659574468085
F1 Score: 96.27659574468085
Precision Score: 96.27659574468085
Recall Score: 96.27659574468085
```

Cross Validation Scores

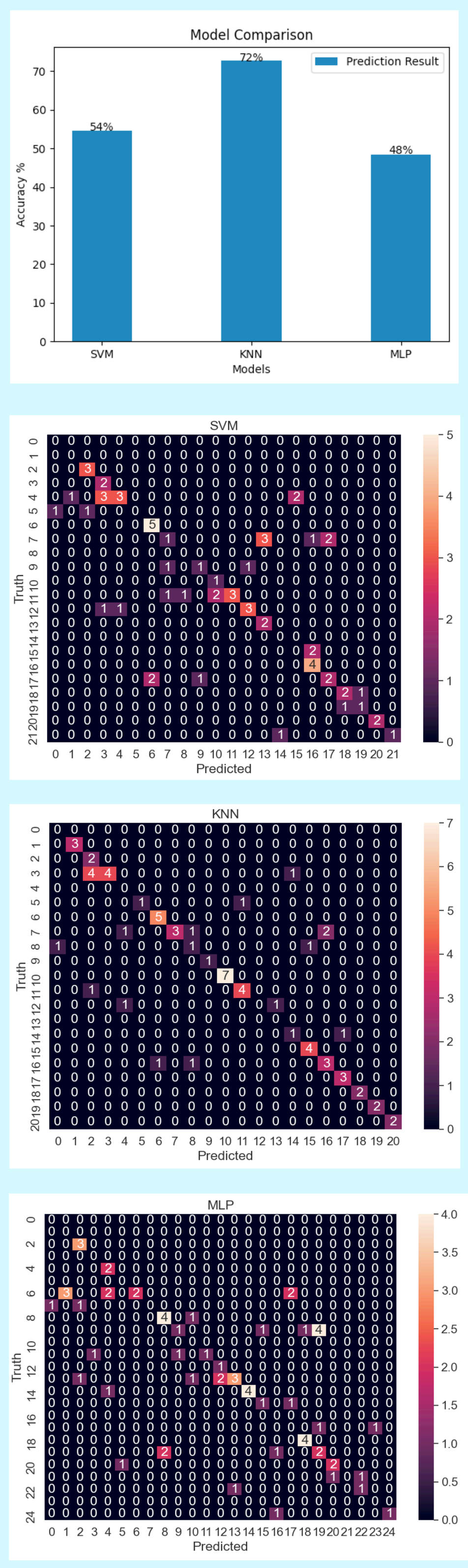
```
SVM| Mean=0.940806 STD=0.012572
KNN| Mean=0.900728 STD=0.003429
MLP| Mean=0.935339 STD=0.003996
```

Evaluation

- SVM performed the best overall
- KNN performed the worst in image 1 with 87% which makes it the worst performing machine learning model overall for this task
- SVM has the highest with a mean accuracy score of 94% with 1.25% standard deviation for the cross validation process

Handwritten English

Image 1

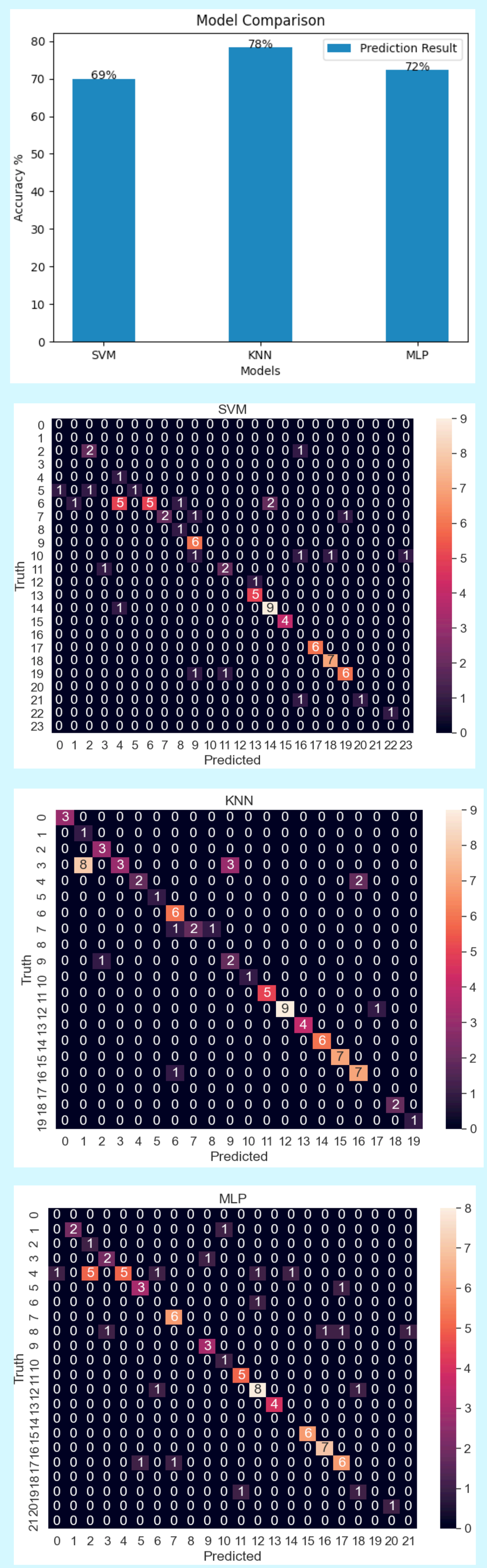


```
SVM Classification Report
Accuracy Score: 54.54545454545454
F1 Score: 54.54545454545454
Precision Score: 54.54545454545454
Recall Score: 54.54545454545454
```

```
KNN Classification Report
Accuracy Score: 72.72727272727273
F1 Score: 72.72727272727273
Precision Score: 72.72727272727273
Recall Score: 72.72727272727273
```

```
MLP Classification Report
Accuracy Score: 48.484848484848484
F1 Score: 48.484848484848484
Precision Score: 48.484848484848484
Recall Score: 48.484848484848484
```

Image 2



```
SVM Classification Report
Accuracy Score: 69.87951807228916
F1 Score: 69.87951807228916
Precision Score: 69.87951807228916
Recall Score: 69.87951807228916
```

```
KNN Classification Report
Accuracy Score: 78.3132530120482
F1 Score: 78.31325301204821
Precision Score: 78.3132530120482
Recall Score: 78.3132530120482
```

```
MLP Classification Report
Accuracy Score: 72.28915662650603
F1 Score: 72.28915662650604
Precision Score: 72.28915662650603
Recall Score: 72.28915662650603
```

Cross Validation Scores

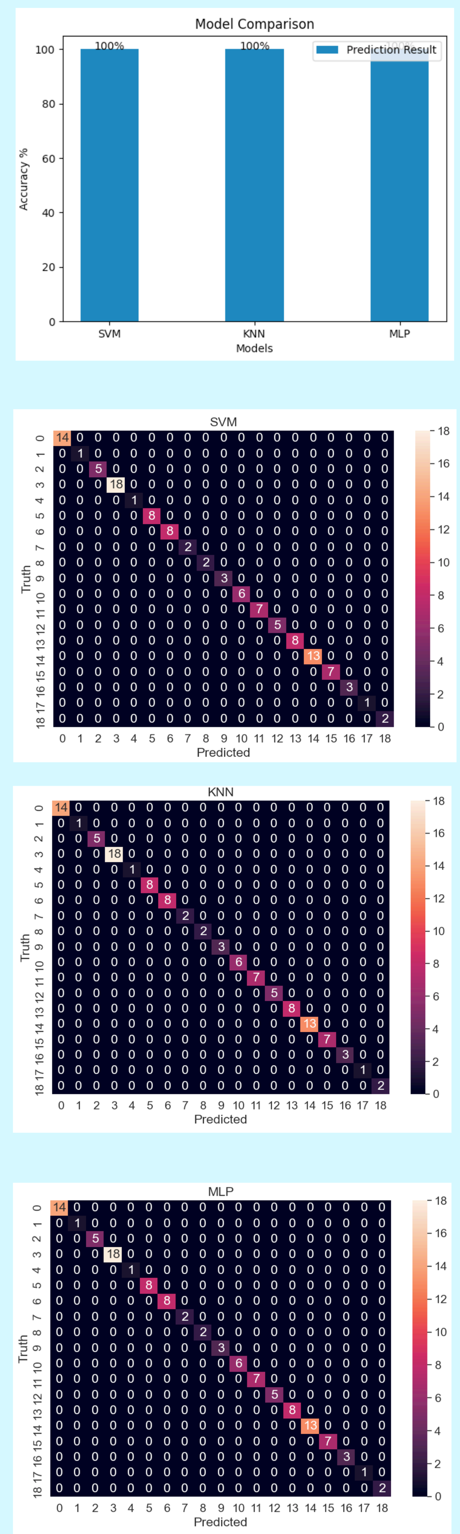
```
SVM| Mean=0.662415 STD=0.019654
KNN| Mean=0.641869 STD=0.035457
MLP| Mean=0.652200 STD=0.048741
```

Evaluation

- KNN performed the best overall
- MLP performed the worst overall with 48.5% for Image 1 although performing better than SVM for Image 2 by 3%
- SVM has the highest with a mean accuracy score of 96% with 1.96% standard deviation for the cross validation process

Printed Greek

Image 1

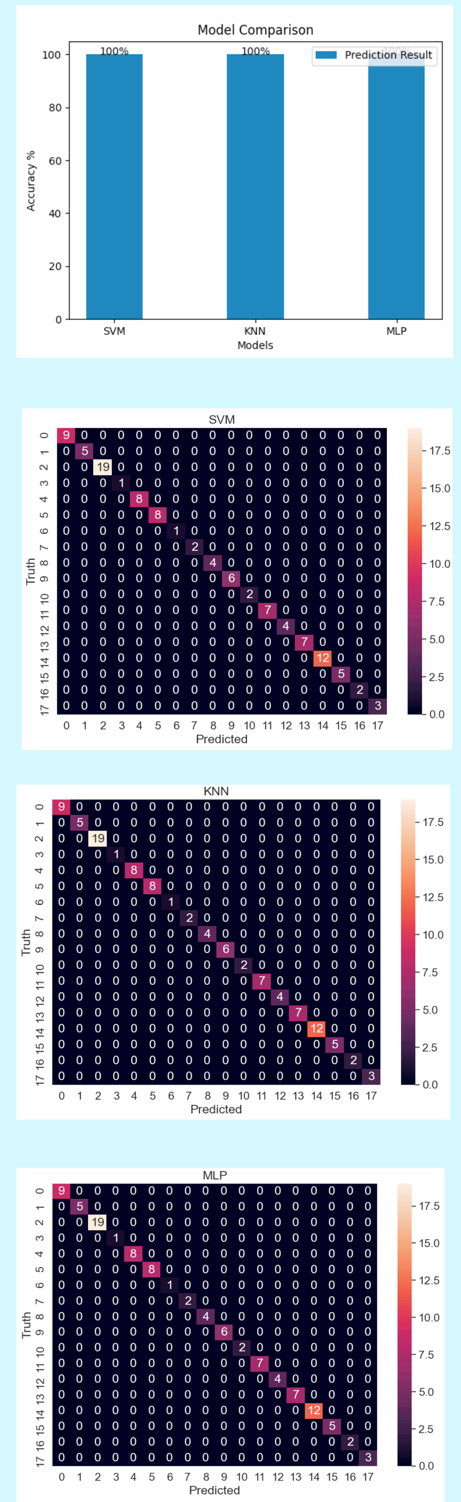


```
SVM Classification Report
SVM Precision: 100.0
SVM Recall: 100.0
SVM Accuracy: 100.0
SVM F1 Score: 100.0
```

```
KNN Classification Report
KNN Precision: 100.0
KNN Recall: 100.0
KNN Accuracy: 100.0
KNN F1 Score: 100.0
```

```
MLP Classification Report
Multilayer Perceptron Precision: 100.0
Multilayer Perceptron Recall: 100.0
Multilayer Perceptron Accuracy: 100.0
Multilayer Perceptron F1 Score: 100.0
```

Image 2



```
SVM Classification Report
SVM Precision: 100.0
SVM Recall: 100.0
SVM Accuracy: 100.0
SVM F1 Score: 100.0
```

```
KNN Classification Report
KNN Precision: 100.0
KNN Recall: 100.0
KNN Accuracy: 100.0
KNN F1 Score: 100.0
```

```
MLP Classification Report
Multilayer Perceptron Precision: 100.0
Multilayer Perceptron Recall: 100.0
Multilayer Perceptron Accuracy: 100.0
Multilayer Perceptron F1 Score: 100.0
```

Cross Validation Scores

```
SVM| Mean=0.988478 STD=0.006294
KNN| Mean=0.973175 STD=0.010146
MLP| Mean=0.991042 STD=0.003154
```

Evaluation

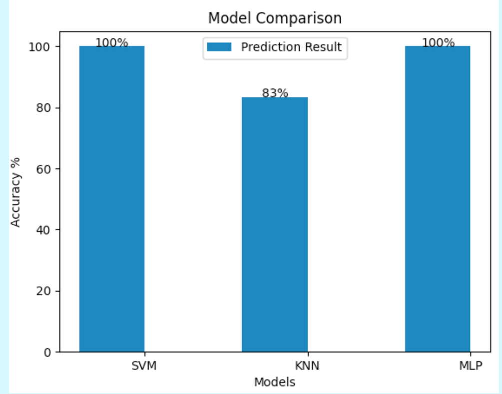
- SVM performed the best overall
- MLP performed the worst overall with 48.5% for Image 1 although performing better than SVM for Image 2 by 3%
- SVM has the highest with a mean accuracy score of 94% with 1.25% standard deviation for the cross validation process

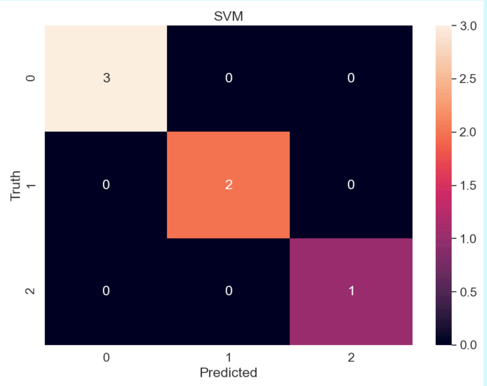
COM2028 - Introduction to Artificial Intelligence

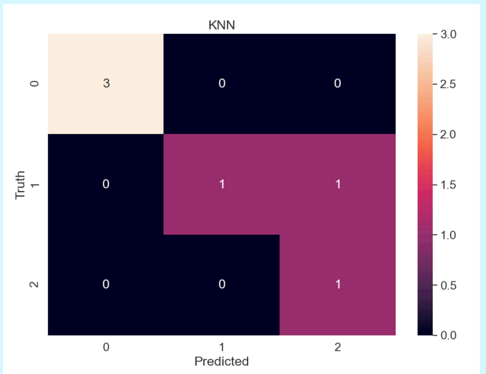
Filbert Miko Linardi
FL00296 - 6546907

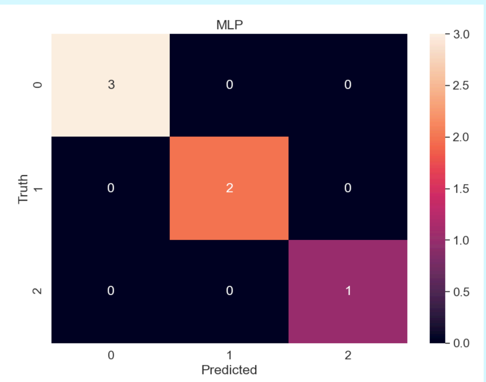
Results

Signature










```
SVM Classification Report
Accuracy Score: 100.0
F1 Score: 100.0
Precision Score: 100.0
Recall Score: 100.0
```

```
KNN Classification Report
Accuracy Score: 83.33333333333334
F1 Score: 83.33333333333334
Precision Score: 83.33333333333334
Recall Score: 83.33333333333334
```

```
MLP Classification Report
Accuracy Score: 100.0
F1 Score: 100.0
Precision Score: 100.0
Recall Score: 100.0
```

Cross Validation Scores

```
SVM| Mean=0.987500 STD=0.037500
KNN| Mean=0.938889 STD=0.114564
MLP| Mean=0.962500 STD=0.112500
```

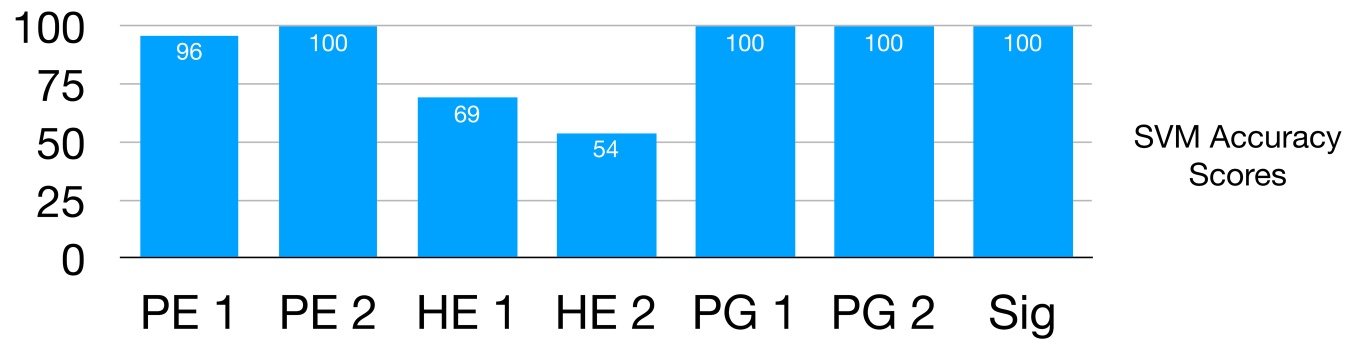
Evaluation

- SVM and MLP performed the best as they both have 100% accuracy for this task
- KNN seems to be to worst performer among the 3 machine learning models
- SVM has the highest percentage with 98.75% for the cross validation process

Discussion

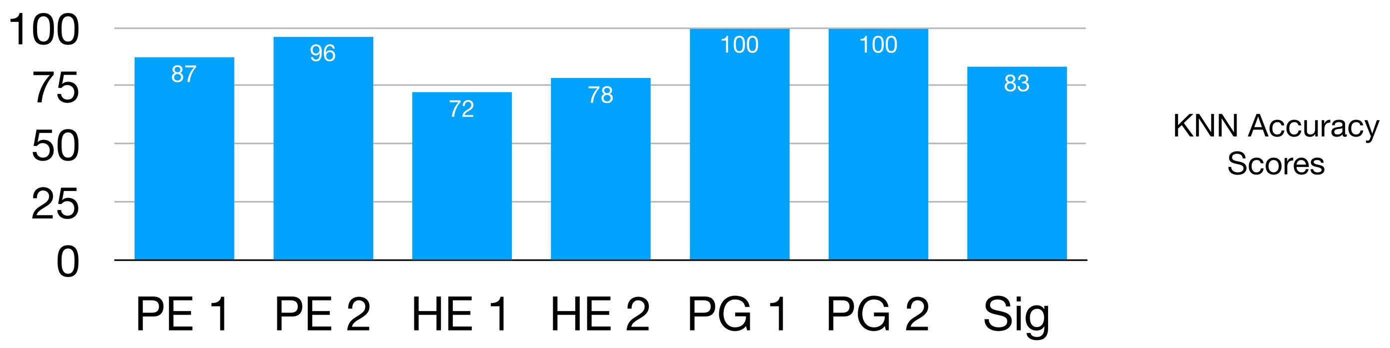
SVM Evaluation

SVM is the overall best machine learning technique for the tasks of this project. Even though it was the worst performer for Handwritten English, it was very accurate in classifying objects for Printed English, Printed Greek and Signature recognition as it had 96% and 100% for the Images in Printed Language as well as 100% in Signature Recognition and Printed Greek. Apart from this, there is a trend in this study where SVM has the highest mean accuracy during the cross validation process. SVM did not perform well in Handwritten English and this may be the result of poorly created dataset. There are images of other letters inside the folder for a designated letter which leads to increasing the amount of noise. From the results we can see the training is appropriate and sufficient for SVM except for Handwritten English as it is overfitting.



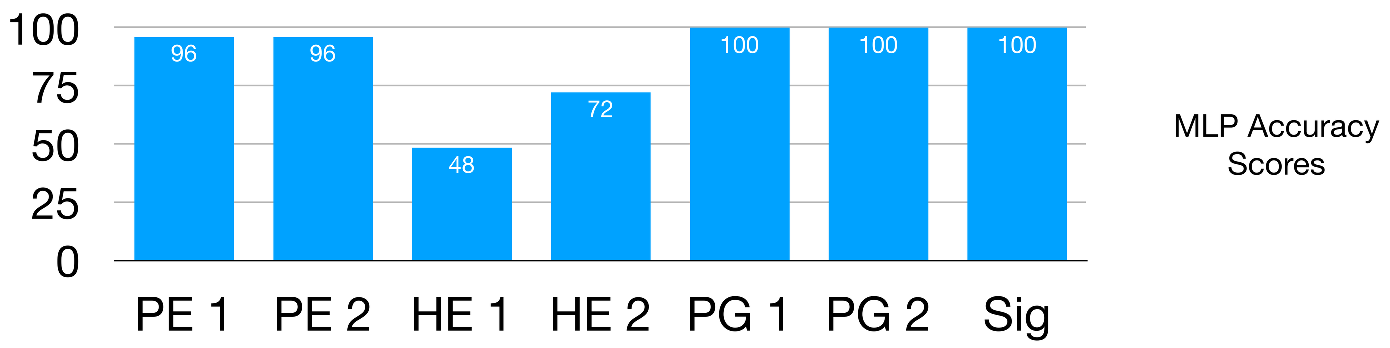
KNN Evaluation

KNN's accuracy score was the highest for both Handwritten English images with 72% for image 1 and 78% for image 2. This may be due to the fact that KNN is a lazy learner which means that it does all the work, finding the nearest neighbour, at prediction time. This means that unlike SVM, it does not output a set hyperplane or line that separates the tags. This might also be the reason for its poorer performances compared to SVM and MLP for the other tasks. Being said so, the performance of KNN throughout this whole project can be considered reliable as the lowest accuracy score it got was 72% and this is the highest score among SVM and MLP for that specific test which is Handwritten English image 1. However, the reliability of KNN cannot dismiss the fact that it may suffer overfitting and underfitting.



MLP Evaluation

MLP's accuracy is similar to SVM's but the difference in classifying and predicting the Handwritten English task is very noticeable. However, out of all the tests, MLP did produce the lowest accuracy score when it is tested against Handwritten English Image 1. As we can see on the chart below, it has an accuracy score of 48%. This machine learning model is also more likely to suffer from overfitting and underfitting compared to SVM.



Testing Environment

For the testing phase, it has already been made sure that the test should be as fair as possible in order to have a fair comparison.

Each machine learning model have been developed to its optimal level. This is done by both evaluating the cross validation as well as the metrics given from the testing phase. The process is repeated until each model have been developed to its optimal level.

The same dataset has been used for each machine learning model to make sure that it will be a fair comparison.

Findings of Study

From the charts on the left that describe the accuracy scores of SVM, KNN and MLP throughout this study, we can see that there is a clear correlation between accuracy score and quality of dataset. The 3 machine learning models failed to achieve the desired accuracies for classifying and predicting Handwritten English because of the lack of quality in the dataset. The data preparation phase for this task could be done better by improving the quality of dataset. One way we can improve the dataset is by controlling how much noise there is.

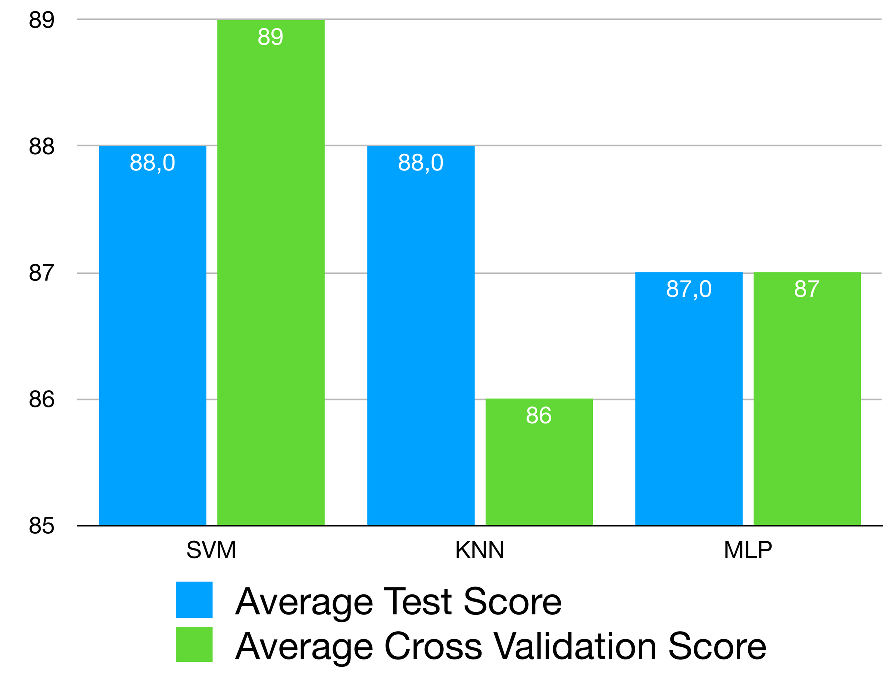
This study have made me realize how important the dataset is. As we can see from the trend that those tasks that have a proper and large enough dataset are able to have high accuracies. The variety of dataset for each class is also important. Initially I tried making a dataset for Printed Greek using only one font and it lead to each model having horrible accuracy scores.

Another factor that can make a huge difference is the extractor class. I kept having a problem trying to extract the test image because it kept on missing a few characters.

The extractor can make a huge difference because it is the tool that is being used to create the dataset as well. The fact that it can miss characters is a huge liability to the whole process of this Optical Character Recognition System.

Conclusion

From the chart below we can see that SVM's performance is the best compared to KNN and MLP in this study. With that being said, KNN can be considered to be the most reliable in this study as the lowest accuracy score that it got is 72%. Although KNN is more reliable, I would recommend SVM because SVM did very well when the dataset is good. So, from this study we can see that with a proper and quality dataset, SVM can perform very well.



Literature Review

To begin with this project, I went through all the labs carried out for this module and most of the code for this project comes from or inspired by those in the lab exercises. This is because this project is similar to the labs exercises. Other works based on the internet are also used to help me develop and evaluate the models used in this project. All the sources used are listed here in this section

- Lab 2 - Used as an inspiration for both developing a KNN model and metrics to evaluate the performance of a machine learning model.
- Lab 7 - Used as an inspiration for getting the HOG features of each image and how to use them to train a model and to develop the use of SVM model
- Advantages and disadvantages of SVM - https://medium.com/@dhiraj8899/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107
- How SVM work - https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72
- How KNN work - https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
- Cross Validation with K-Fold - https://pythondata.com/comparing-machine-learning-methods/
- How to Implement MLP - https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc
- When to use which Neural Network - https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/