# Data Quality Assessment

For the Data Quality Assessment I carried out the following three methods:

## Dip testing

By randomly extracting groups of labelled data by the label type I manually checked 100 images per character to ensure that they matched the labelled data. I did this to ensure that the data was correctly labelled, without having to check all the images within the dataset.

## Size Analysis

I checked the dimensions of all images to ensure they were a uniform 28px x 28px size. This is to ensure that the models would be able to learn each labelled class correctly and fairly to avoid bias.

## Peer documentation review

I peer reviewed several studies that used the MNIST and AZ datasets. This was to see if anyone else had found any issues with the dataset that I would have to rectify prior to usage.

# Feature Engineering

For my feature engineering stage I used de-noising techniques. This has been studied within the field of OCR and there are many options available, I chose a lightweight solution of median filtering but there are better solutions (by MSE). These better solutions using ML or CNN’s provided improved results but larger processing times.

De-noising the documents improved accuracy as it removed erroneous contours detected by OpenCV.

# Choice of Algorithm

For the Model I chose a KNN Classifier for the ML side and ResNet50 for the Deep Learning side. Both of these have high scores on the MNIST dataset site with extensive papers detailing their effectiveness at correctly predicting any given character.

While there are out of the box solutions such as Tesseract this would not provide a sufficient challenge or meet the criteria of the capstone project.

Using both these algorithms I was able to get effective F1 Scores of 0.91 and 0.96 respectively

# Framework

## Enterprise Data Warehouse

Within my organisation I have access to only one framework solution. The enterprise data warehouse is accessible for all members of my organisation and enables all uploaded documentation to be viewed within the organisation.

## Apache Spark

Currently the solution works on python within an Apache Spark environment, when deployed it will be changed to improve speed and efficiency with the tools within Apache Spark such as Resilient Distributed Dataframes.

# Model Performance Indicator

For model performance indication I used the F1 Score. This ranges from 0 if either the precision or recall of the model is zero to 1.0, which means it had a perfect precision and recall score.

I used this scoring method because it works very well with multiclass classification and gives an overall indication of the models performance. Using classification reports I was also able to see how well individual labels performed.