5 Naïve Bayesian classifier

The Modified National Institute of Standards and Technology database (MNIST database) is a collection of 60,000 examples of handwritten digits and a test set of 10,000 examples. Information regarding the dataset are used to make a baseline model to extract features. The data set is composed of 10 classes which are restricted to the digits in the range of zero to nine. The purpose of having both a training set and dataset is to accurately gauge the systems performance once the training data has been extensively trained over many epochs. The test set will be used to validate the accuracy and performance of the baseline model.

It is important to have an overview of the type of data given in the MNIST dataset. The orientation of each example is unknown however, there are whats known as invariant in the data set where a single class can be transformed either by being, rotated, scaled, stretched, skewed, or even mirrored and can be seen by the digit one seen in Figure 6 where all three variations of the digit one are of a different orientation. For many of the possible features, it may be a difficult invariant to distinguish rotated classes such as a '9' and '6'. This can be easily miss classified.

Feature extraction is only possible due to the none degraded handwritten digits having distinct characteristics. Each image has an aspects ration of 1 to 1 with the size of each image being 28×28 pixels and are gray-scale. However the the unsigned integer values are ranged from black and white of values 255 and zero. Since a Gray-scale sub-images has been used for the data-set, the data-set can be normalized and re-scaled from 0 to 1 thus creating a new feature used to identify the example.

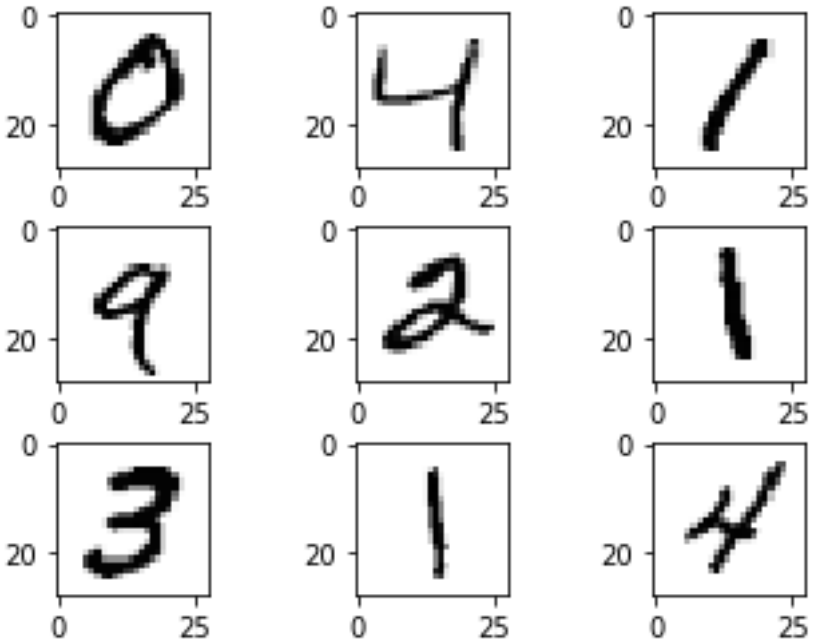


Figure 6. Example of MNIST data set

From the training class dataset that will be used to train the classifier, there are many variations of the same class (digit) which have a slight variation in its text width, height, ratio, or even if its slanted or rotated as seen by Figure 6 which shows 9 digits from the dataset.

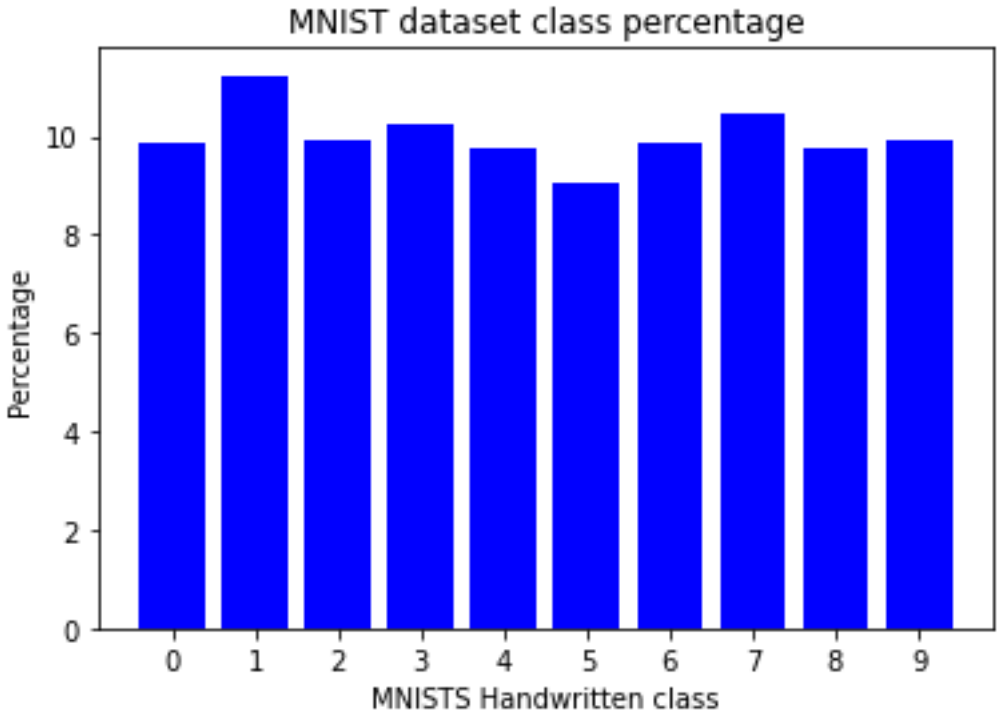


Figure 7. MNIST dataset class percentage

Figure 7 is the percentage of each class divided by the total number of data points in the training set to show that the occurrence of any number is equally distributed with a maximum standard deviation of 1.2% with the class one having the most occurrences and class five having the least.

5.1 Feature selection

Feature selection is understood to be the process of identifying feature subsets for a initial feature. Where the choice of a feature is dependents on the requirements of the system. A caution of selection of only the most relevant features as to avoid dimensionality issues when working with statistical classifiers [2]. It has been shown in Part A that by having to many variables used to determine the predicted class, over-fitting can occur and result in overly flexible assumptions [1].

In the past many methods have been used to extract features from a given dataset such as the MNIST dataset. Features can be to name a few, Pixel intensity [3] with each pixel graded according to its binary value. Projection of histograms that takes cumulative histograms of a small set of either horizontal or vertical pixels [4]. Large sets of data require **memory** and by

using less features, the system will be less memory intensive. Other constraints to consider are the **time** and **accuracy** which greatly impacts the **reliability** and **performance** of the model [5]. These performance measures will be used to determine the success of the each feature set.

The effects of training set-size on feature selection encountered greatly impacts the predictive performance of the system. With more data to train on, the maximum likelihood of predicting the correct answer or make a prediction closer to the actual answer is far greater. In the next sub sections the author will discuss the impacts of this with the use of a confusion matrix to show miss predicted results.

5.1.1 1) Histogram of Oriented Gradients (HOG)

The HOG method uses the gradient and orientation otherwise known as the magnitude and direction or localized portions since the HOG approach divides the imagine into workable portions. All images must be Preprocessed to the correct ratio[6].

The method of extracting the gradients and magnitude from the image is done knowing that the direction colour changes with a change in the colour scale - For this assignment, its a black background and white text thus gray-scale images. Knowing that the pixel is an images most independent form that values produced are discrete values and independent of their neighbours. The gradients is the change in pixel colour in both the x-axis and y-axis. With the gradient vector for both ¡x,y¿ calculated as follows :

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{38}$$

. Where the partial derivative, derived towards either x or y given by $\frac{\partial f}{\partial y}$ for y-axis and $\frac{\partial f}{\partial x}$ for x-axis. It is important to note that the gradient of the target pixel is done by taking the value of the pixel to either side of the target pixel along the chosen axis. The gradient vector can then be used to determine the magnitude of that specific pixel by taking the square root of the partial derivative, derived towards x and y square. With the equation

$$g = \sqrt{g_x^2 + g_y^2}, \tag{39}$$

giving the magnitude. By separating the magnitude into its x and y components, Theta can be expressed as

$$\theta = \arctan(\frac{g_y}{g_x}), \tag{40}$$

which is known as the direction of the gradient and shows stark contrast only when the pixel intensity changes drastically.

5.1.2 2) Projection histograms (First Principle)

In [4] the author states that the primary use of Projection histograms is used for segmenting characters, words and digits. However, the feature set struggles to adjust to the possibilities of slanted text and any kind of off centred rotation. Two equations are formed with the first being a Horizontal projection $y(x_i)$ and x_i the number of pixels in the horizontal axis. The second is a vertical rotation given by $x(y)$ which is independent of the disadvantage experienced by the horizontal projection of being slant variant [4].

$$Y\left(x_k\right) = \sum_{i=1}^{k} y\left(x_i\right) \tag{41}$$

By taking the cumulative histograms of the horizontal projection where the sum of the initial index bins is calculated by equation horizontal projection [4]. The dissimilarity measure of two projections is now calculated by,

$$D = \sum_{i=1}^{n} \left|Y_1\left(x_i\right) - Y_2\left(x_i\right)\right|, \tag{42}$$

where the cumulative absolute value of the cumulative histograms is compared between two projections.

5.1.3 3) Zernike moments

Zernike moments are used as a feature selection in pattern recognition and will aid in solving the classification problem encountered in this assignments.

The Zernike polynomial is a complete orthogonal set which is constrained to the polar coordinates over a unit circle defined by: $x^2 + y^2 1$ [7]. The Zernike polynomials is defined according to $V_{nm}(x, y)$ can be calculated by,

$$V_{nl}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(\mathrm{j}m\theta). \tag{43}$$

The Zernike moment can be separated into two parts being a Real ($R_{nm}(\rho)$) and Imaginary ($\exp(\mathrm{j}m\theta)$) components. With n $\in \mathbb{R} = 0, 1, 2, 3..N$ and m defined according to the condition of: $n - |m| =$ even,$|m| \leq$ n. R_{nm} is the radial polynomial and can be calculated by,

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}. \tag{44}$$

The complex Zernike moment is defined according to,

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x, y) \left[V_{nm}(x, y)\right]^* dxdy \tag{45}$$

with f(x,y) the continuous image function. To perform the Zernike moment of a given class, the centre of the image is made to be the reference point and if the pixels are not within the circle, the pixels will not be used in the computation. Finally the Zernike moments for a digital image can be simplified by replacing the integrals with the double summation over the range and can be seen as follows,

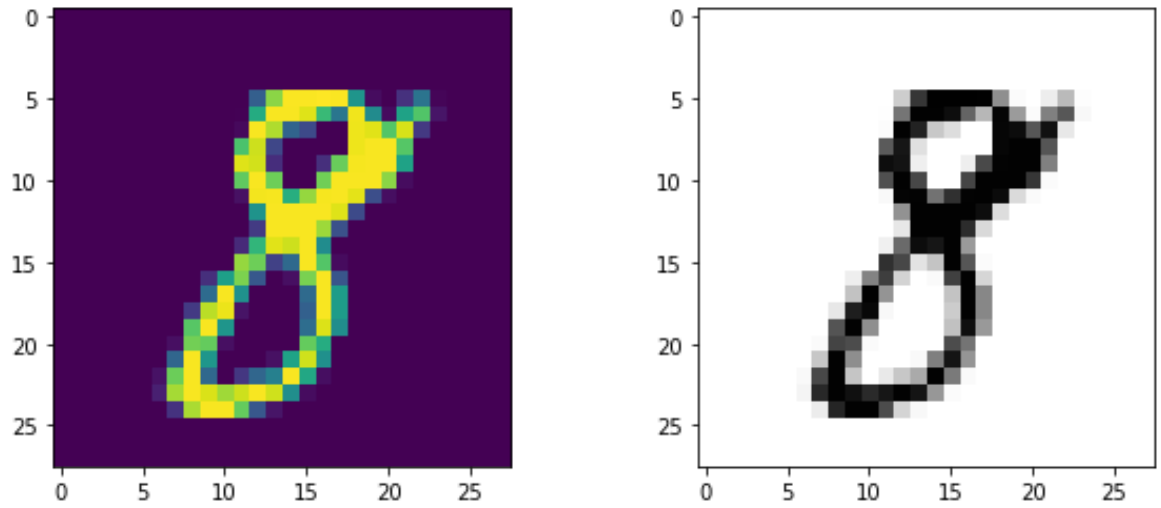
$$A_{nm} = \frac{n+1}{\pi} \sum_{x} \sum_{y} f_d(x,y) \left[V_{nm}(x,y)\right]^* . \tag{46}$$

With each image being classified according to their set of Zernike moments such that $A_{nm} = A_0, A_1, A_2 ... A_N$. There is a tolerance to the deviation that a single Zernike moment can have to be classified correctly and is defined according to $d_k = \left\{d_1^k, d_2^k, \ldots, d_{n_{\max}}^k\right\}$ where d_k defines the acceptable tolerance for a given class [7].

5.1.4 4) Pixel intensity

The simplest way to extract a feature from a dataset is by using its pixel intensities. This method is done by determining the change in the pixel intensity relative to its neighbours. A pixel can be represented in several different colours and broken down into one of four general colour schemes being gray-scale, red, green or blue. With those primary colours, all other colours can be formed by mixing them. The colour however in hardware is recognised as an integer value and scales from zero to two hundred and fifty five [8].

However for this assignment of using the MNISt dataset, the images used will only have a single channel being black and white which drastically simplifies the approach.It is clear why Pixel intensity values being one of the most popular feature sets used for classification due to the performance vs computational intensity [9].



(a) Digit eight without gray-scale

(b) Digit eight with gray-scale

Figure 8. Handwritten digit pixel preview

The handwritten digit seen in Figure 8 shows two images only different by the color scheme with the first showing the original dataset Figure 8a and the second being gray-scaled Figure 8b. However, the image shown is not the same set of information that the compiler will interpret but rather a graphical interpretation.

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  77 211 254 254 255 254 133   8   0  36  98   3   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   1 157 253 253 243 173 100 250 123   7 147 186  14   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   6 253 225  82  57   0   0 249 244 187 240  42   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 183 236  54   0   0   0   0 249 253 253 188   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 243 238  33   0   0  34 197 252 253 253 147   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 193 253 199   0   0 200 253 253 253 211   8   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  11 251 252 160 215 252 253 245  61   9   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 131 253 253 253 253 243  57   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  44 253 253 253 218  63   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  30 168 244 236 253 124   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 215 199  49  87 253 187   7   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  34 144 205  70   0  10 204 253  67   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  89 253 129   0   0   0  88 253 141   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  43 229 248  13   0   0   0  78 253 141   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 189 253 133   0   0   0   0  91 242 125   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  10 201 195  10   0   0   0   0 195 158   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  84 253 134   0   0   0  11 148 237 128   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0  12 198 253  86   0  41  84 111 253 153   9   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0  22 221 253 241 219 236 253 237 126   8   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  37 223 253 253 195  69  14   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
```

Figure 9. 28 x 28 pixel print of Digital class

Figure 9 shows the pixel values of the same digital class. Clearly from the figure, a white background is represent by zero and anything upwards becomes a darker shade of gray with 255 being pure black. In the figure is can be made out by inspection that the values of the pixels (non-zero) forms the shape of the digit eight. The Figure of the eight is similar to that found in Figures (8a, 8b).

5.2 Statistical and class discrimination analysis of features

This section of work will be used to interpret and analyse the results of the four different feature selection sets. By comparing results of the different extracted features, a decisive decision of the performance and suitability can be made accurately. This will be done for each extracted feature set by establishing a visual graphic representation of the predicted value in comparison to the expected value and performance metrics such as the standard deviation and variance.

5.2.1 Histogram of Oriented Gradients

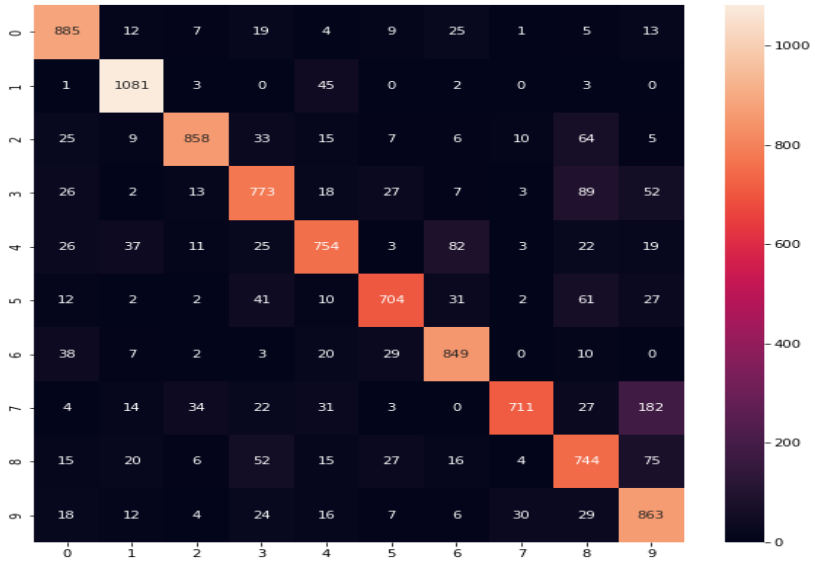


Figure 10. HOG confusion matrix

Once the ML HOG feature was implemented and trained, a test was conducted to see how the system would perform. To show the results, a confusion matrix was used to plot the comparison between the predicted values and known inputs. The results show that the general assumption across all class were well averaged except, for the digit seven which was often mistaken as the digit nine. Figure 10 shows the confusion matrix of the HOG feature set. It is clear from the plot that the HOG feature set worked well across all classes, given that the only outliers are the classes four and seven with that at least achieving an accuracy of 81.3%.

Mean Accuracy	Variance	Maximum Accuracy	Recall
82.22 %	0.0%	85.5 %	N/A

Table 3. HOG summarized result

The results of the HOG feature set are summarized in Table 3 above with an accuracy of 85.5% and minimal standard deviation. The maximum accuracy recorder was 85.5%.

Epochs	Mean	Standard Deviation	Average Accuracy	Maximum Accuracy	Loss Function
150	91.7 %	0.0 %	90.58 %	95.66 %	0.4135

Table 4. Convolutional neural network classifier HOG summarized result

To show the performance of the chosen classifier an alternative classifier was designed based on the same feature set with the recorded results summarized in 4.

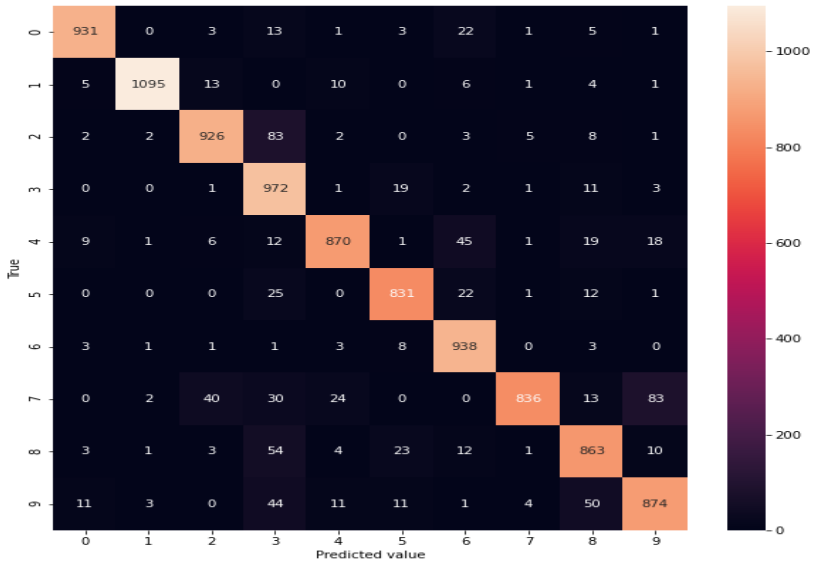


Figure 11. Convolutional neural network classifier with HOG confusion matrix

With the Convolution neural network classifier training over 150 epochs was bale to achieved an accuracy of 91.7%. The results of these classifiers will be extensively discussed in Section 5.5. Figure 11 is the confusion matrix of the CNN HOG feature set.

5.2.2 Pixel intensity

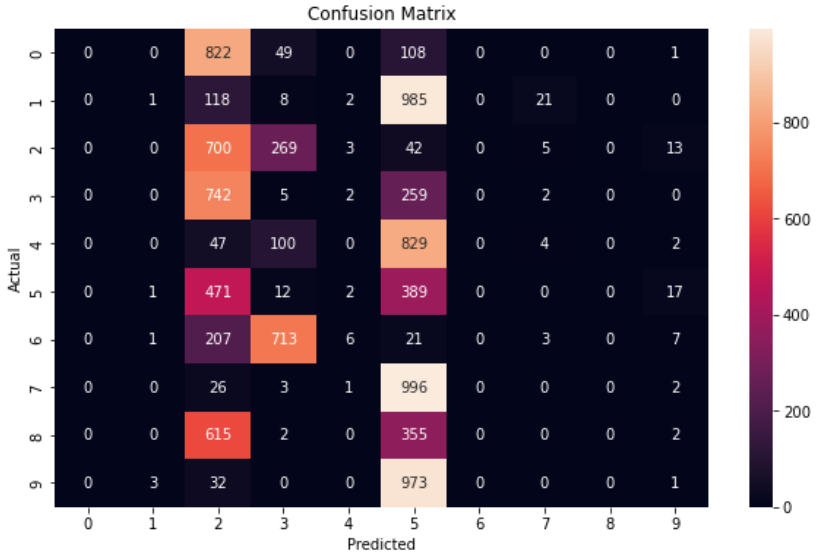


Figure 12. Pixel Intensity confusion matrix

The results obtained from the Pixel intensity feature set were highly inconsistent with a mean accuracy of 11.2%. The class analysis of this feature set is shown by the confusion matrix in Figure 12. It is clear from inspection that the mean accuracy is correct given that the diagonal generally formed across the confusion matrix is non existent. Most of the handwritten digits

were classed as either being the digit two, five three. However, this is not the case as the MNIST dataset is almost equally distributed per class.

Mean Accuracy	Variance	Maximum Accuracy	Recall
11.2 %	0.1%	12 %	N/A

Table 5. Pixel Intensity summarized result

The results from the Pixel intensity feature set were recorder in table 5 with a mean accuracy of 11.2% which is due to the classification of most classes.

5.3 Classifier theoretical analysis and derivation

The conditional probability of event A occurring given event B has taken place is denoted by $P(A|B)$ which is regarded as a conditional probability [5]. This can be further expanded to,

$$P(A \mid B) = \frac{P(A, B)}{P(B)}. \tag{47}$$

With the conditional probabilities of event A occurring given event B has occurred equating to the probability of event A and event B occurring divided by the probability of event B. The following stands to be true that, P(A, B) = P(B, A), which leads to Bayes theorem and can be derived as follows,

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{48}$$

However, for the implementation of Bayes theorem for a machine learning algorithm it is important to rewrite the equation baysiean as follows,

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)}. \tag{49}$$

Where y will be used to denote the response and x the feature (For this assignment x will be the MNIST data set). By using Bayes rules we are going from seeking the probability of P(x—y) to seeking the probability of P(y—x). This tells us how to predict the class of an unseen example given a train set [**?**]. By making the assumptions that the likelihood is of the Gaussian distribution form given by,

$$P\left(x_i \mid y\right) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}\right). \tag{50}$$

With μ and σ_y the variance and standard deviation.

Assumptions made for Naïve Bayes classifier

- It is assumed for this assignment that all input variables are independent of all other input variables. For all inputs denoted by X, $X = (x_1, x_2, x_3, \ldots\ldots, x_k)$ such that $x_1, x_2, x_3..x_k$ are independent. This implies that by changing the values of one feature, it will not influence the values of another.
- It is also assumed that the likelihood of any particular class occurring y, given any feature x is equally likely.

The probabilistic machine learning classifier based on Bayes theorem with the assumptions made can be expanded upon to whats known as Naive baysiean classifier. To do this, it is also assumed that there are multiple classes and by reviewing equation 49 to produce a model based Bayes theorem. Which can be expressed as,

$$P(y = k \mid x) = \frac{P(x \mid y = k) \cdot P(y = k)}{P(x)} \tag{51}$$

where k denotes the class of a given response. By using the MNIST data set, there will be 10 classes as described. The equation can be expanded upon by considering all possible classes of x,

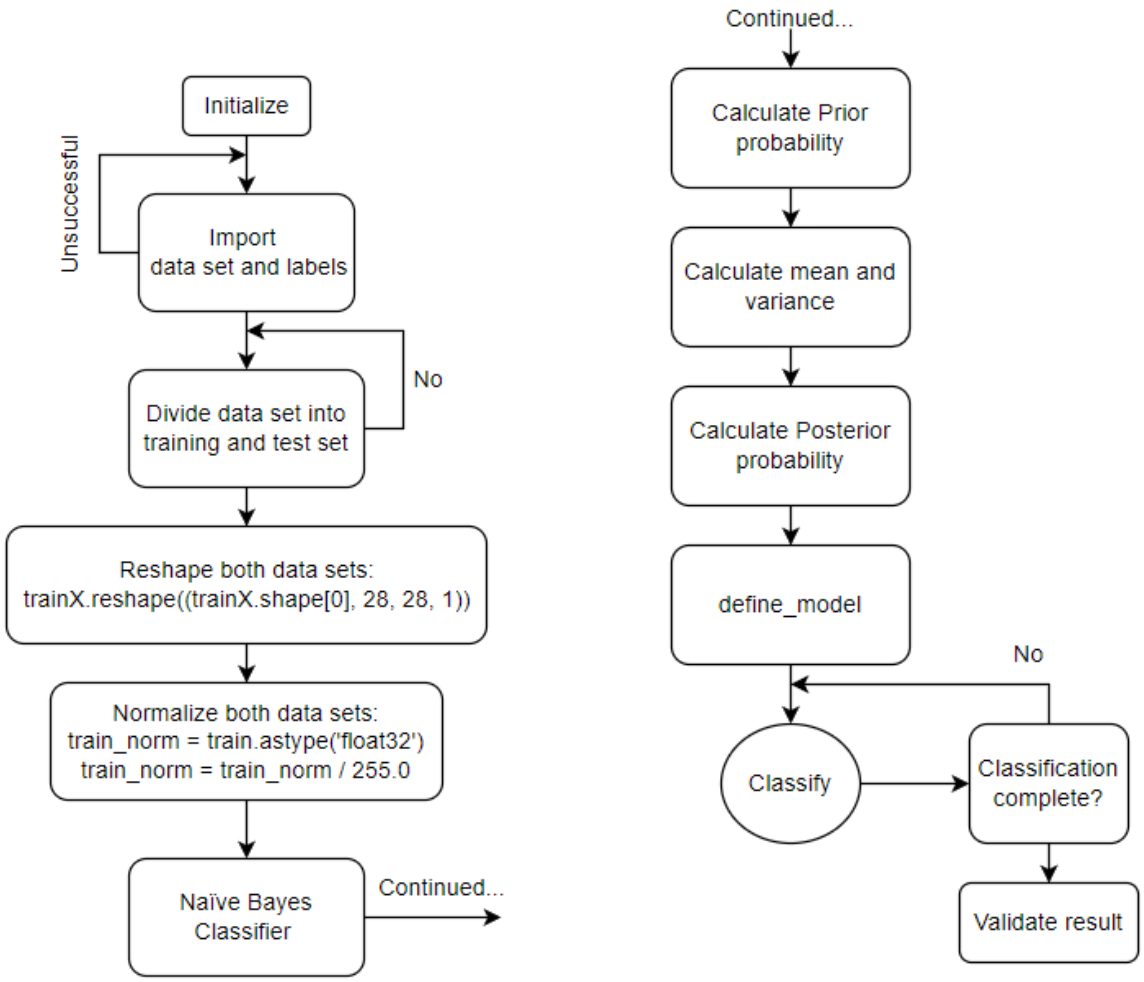
$$P(y = k \mid x_1..x_n) = \frac{P(x_1 \mid y = k) \cdot P(x_2 \mid y = k) \ldots \cdot P(x_n \mid y = k) \cdot P(y = k)}{P(x_1) \cdot P(x_2) \ldots \cdot P(x_n)}. \tag{52}$$

The implications of equation 52 implies that the probability of likelihood of the class which is regarded as the conditional probability of a certain class occurring given a particular response multiplied by the probability of a response forms the numerator. The denominator is divisible by the summation of the probability of x=k and can be further simplified to,

$$P\left(C_i \mid E\right) = \frac{P\left(C_i\right)}{P(E)} \prod_{j=1}^{a} P\left(v_j \mid C_i\right), \tag{53}$$

which is will be implemented to determine the maximum likelihood of the predicted value.

5.4 Naïve Bayesian classifier algorithmic development



(a) Importing and Preprocessing data **(b)** Classifier algorithmic development

Figure 13. Classifier algorithmic development

The process of implementing a Naïve Bayesian classifier starts with importing the dataset and splitting them into test and training set as discussed in Section 5 - Overview. The data is then further split to have the data points (handwritten digit) and label (actual value) of both test sets. The handwritten digits then need to be processed by reshaping the N-dimensional array with a single channel for the gray-scale given that no RGB colour scheme is used. The image pixels are kept in integer values ranging from [0, 255] and will be normalized by first converting every pixel into a floating point and dividing each 28 x 28 pixel by 255.0 to have the pixel intensity range form [0, 1] which can be seen by Figure 13a.

The second part is the algorithm development for the Naïve Bayes classifier is done according to Figure 13b, where the prior probability is first calculated. The process follows to retrieve the mean and variance used to calculate the posterior probability.

5.4.1 Implementation of the Naïve Bayesian classifier

The Implementation of the Naïve Bayesian classifier in python is done by separating the process into smaller subsystems being pre-processing, classification and analysis. This section will cover a high-level interpretation of how to setup the Naïve Bayesian classifier along with the importing and restructuring the datasets.

Algorithm 1 Import and Pre-processing

Require: $fromtensorflow.keras.datasetsimportmnist$

1: $(trainX, trainY), (testX, testY) = mnist.load\_data()$ ▷ Import data
2: $dataset = reshape((28, 28, 1))$ ▷ Reshape matrix with single channel
3: $dataset = astype('float32')$
4: $Normalized = dataset/255.0$ ▷ Normalized data

Algorithm 1 is the process of importing the data set using Tensorflow, Keras and splitting the data into two sets of two. The sets are comprised of data points and labels, and split into a training and test set. Both the train and test sets are reshaped to have the form of a 28 by 28 pixel image that is grey scaled with the data for the pixels being converted from an integer value to float. The dataset are then normalized to be in the range of [0,1] by diving it by the highest possible pixel value of 255.0. By normalizing the datasets before being classified, will help reduce computational power and improve model's efficiency.

Algorithm 2 Naïve Bayesian classifier

Require: $fromtensorflow.keras.datasetsimportmnist$
Require: $importnumpyasnp$
Require: $fromsklearn.naive_bayesimportGaussianNB$

1: $(trainX, trainY), (testX, testY) = mnist.load\_data()$
2: $count = np.unique(trainY)$ ▷ Count number of classes
3: **for** i in range(N): **do** ▷ Calculate prior
4: count.append()
5: prior.append(trainY['label']/count)
6: $mean.append(np.mean(prior[i]))$ ▷ Calculate variance
7: $S\_deviation.append(np.std(prior[i]))$ ▷ Calculate mean
8: $prediction = 1/(np.sqrt(2 * np.pi * in) * np.exp(-np.square(post - in)/(2 * in)))$

The maximum likelihood probability given by, $P(x|y)$ is calculated by obtaining both the variance and mean in lines 6 and 7 of Algorithm 2. These values are then used to calculate the maximum likelihood estimation given by the Gaussian distribution function as seen in line 9. Parameter estimation is regarded as the crux of the Naïve Bayesian classifier. Several values need to be calculated to achieve a predicted value. Maximum Likelihood estimation is done by estimating the maximum value of along the given axis.

5.4.2 Validation of the Naïve Bayesian classifier

The Performance evaluation metrics for the each feature set will be the same to justify if one feature set does perform better than another. Each model will be verified by a sufficiently trained feature set, where the results are not that of a single epoch run but over the course of at least 100 epochs. This will ensure that the average of each metric is taken and not that of a single iteration. The models are verified by using a k-fold cross-validation method. This implies that there exists both a training and test set to ensure a new source of data is available and unforeseen to the trained algorithm to ensure better, more accurate results.

The parameters used to gauge the performance of the model are, **accuracy**, **mean**, **standard deviation** and **loss function**. However the main validation method used for this assignment will be a confusion matrix which is a 2 dimensional axis with a heat map displaying a higher contrast in colour over regions with more occurrences. The confusion matrix has the y-axis as the true value and x-axis marked as the predicted value being the two input into the systems over the entire dataset.

By using a confusion matrix - can be used to see where the system on average made the wrong decision. For example if the classifier tended to identify the number six as the number nine many of the iterations will be made visible in the results and can be used by the author to make changes to the system. It is possible to identify issues of the classifier by interpreting the results of the confusion matrix.

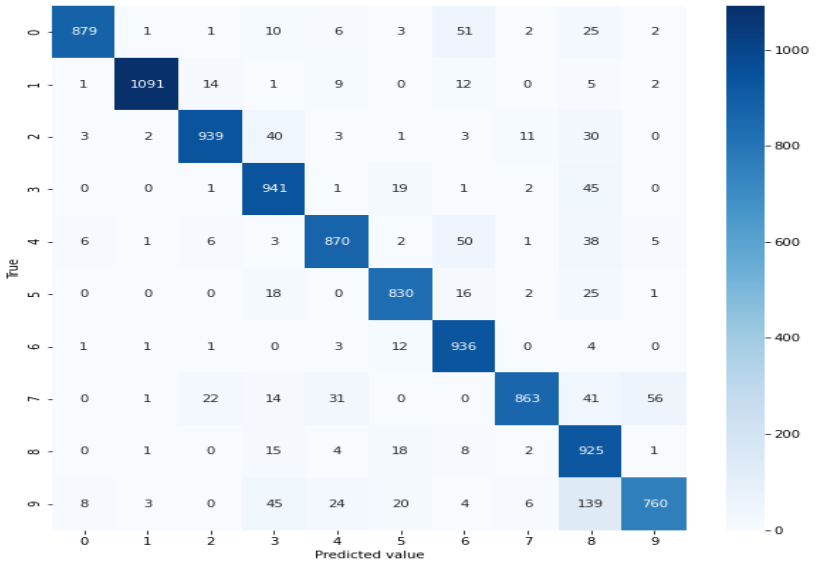


Figure 14. Example confusion matrix

An example of a confusion matrix is seen in Figure 14 above. Where the last row and column are interpreted as the value of nine was predicted correctly the majority of the time however, across many of the iterations, it was misinterpreted as an eight. This in turn helps to identify class classifications errors in the system.

5.5 Performance evaluation and critical analysis of classifier

5.5.1 Performance evaluation of classifier

The result obtained in this experiment shows that by assuming that all input variables are independent of each other. For all inputs denoted by X, $X = (x_1, x_2, x_3, \ldots\ldots, x_k)$ such that $x_1, x_2, x_3..x_k$ are independent. This implies that by changing the values of one feature, will not influence the values of another. Is valid for the Naïve Bayesian classifier to an extent of its performance reaching a peak accuracy of 85.5% across all feature sets. The feature set used for the classification was important and helped enhance the overall performance of the system such that the worst accuracy for the Pixel intensity feature set was 11.2% and highest for the HOG feature set was 85.5%. When the NB classifier was trained without any of the feature set only pre-processing to fit the input variables, the results were as low as 5.5% at the worst of times. By cleaning up the input data by removing redundancies, the overall performance of a NB classifier is greatly improved.

Even with the enhanced performance due to pre-processing and feature selection, I strongly believe that the assumptions made about the classifier were not correct and that by changing the values of one feature, it will influence the values of another. The Naïve Bayesian classifier did not have nearly perfect results which was to be expected if the assumptions were false. The classifier with the best feature was able to achieve a mean of 82.22%.

The Naïve Bayesian classifier worked fairly well, although it is challenging to have results without a comparison thus an alternative convolutional neural network classifier (CNN) was developed and tested on the best feature set (HOG). By doing this, I was able to assess the general performance of the NB classifier and found the results to help defend my new notion of the assumptions being false. Give the CNN classifier had a better mean accuracy of 91.7% which is an improvement of 9.5% and peak accuracy during its 150 epochs was 93.3%. The overall performance of the CNN classifier over NB had a great improvement and can be seen in Unit 5.2.1.

5.5.2 Critical analysis of classifier

Several limitations on the type of feature selections were made based on the data being analyzed and trained on. This is highly important to the type of system that is being designed for. Since the images in this example set are gray-scaled the consideration of a multiple channel being RGB is not relevant and thus minimizes the amount of computations required. The NB classifier performed well when aided with a feature such as the ones found in [4]. When compared to alternative classifiers such as a convolutional neural network there is a great difference in its performance, although it is still capable of detection digit values the majority of the time.

6 Conclusion

Part A encouraged the author to experiment with different curve fitting algorithms such as the least square and maximum likelihood. It was discovered that the performance of the maximum likelihood was similar to that of the least square. Part B discovered that Histogram of gradients is a highly accurate method of extracting features from the MNIST dataset for handwritten digits however, due to the number of computations required, the efficiency of the model is in consideration. By using feature extraction on the MNIST dataset, the general performance of the Naïve Bayesian classifier had shown to improve tremendously with the use of the Histogram of gradients although the results for pixel intensity feature set were surprisingly low, as much as 5 times less than HOG.