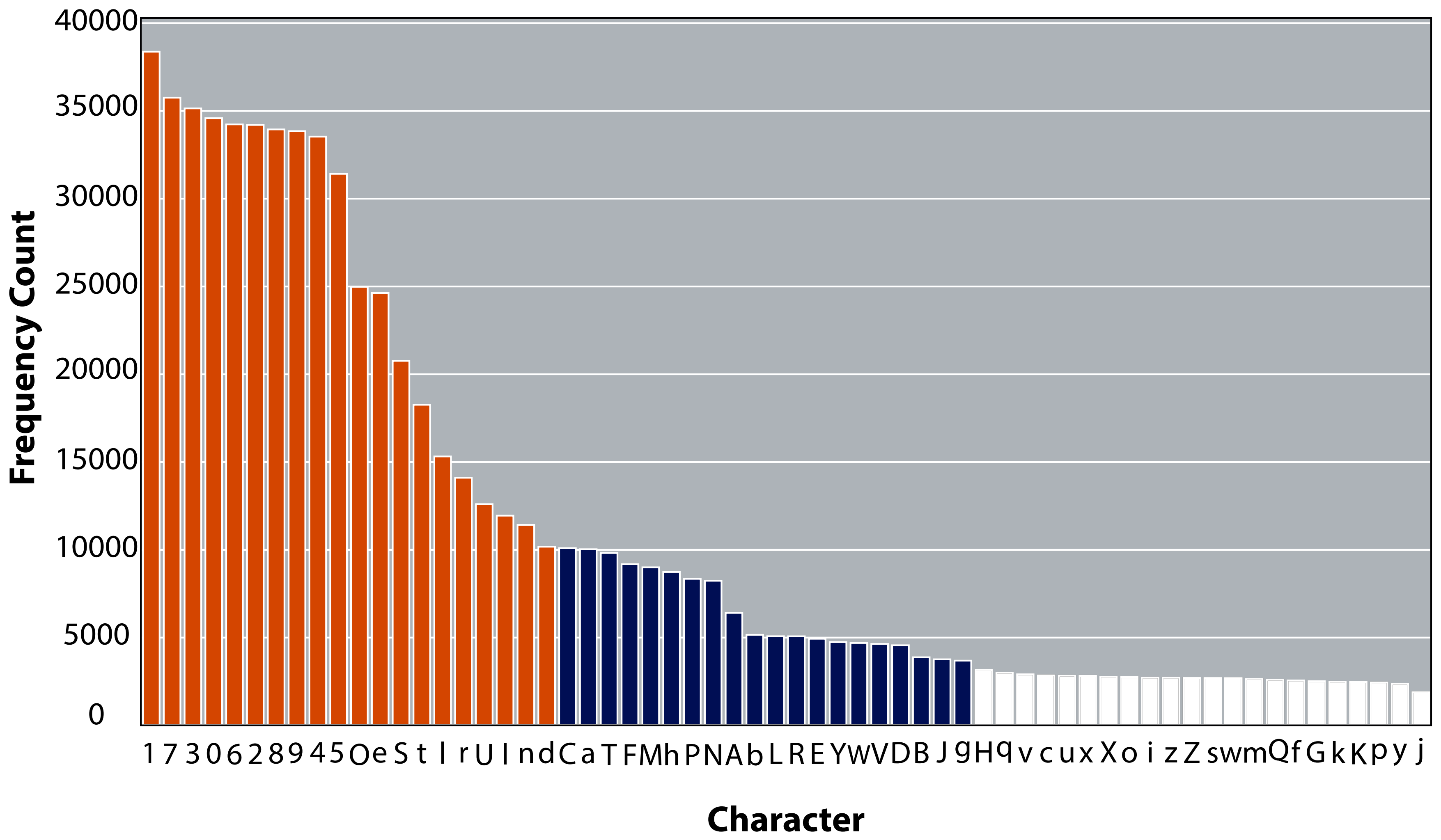
THE iSCHOOL
Syracuse University

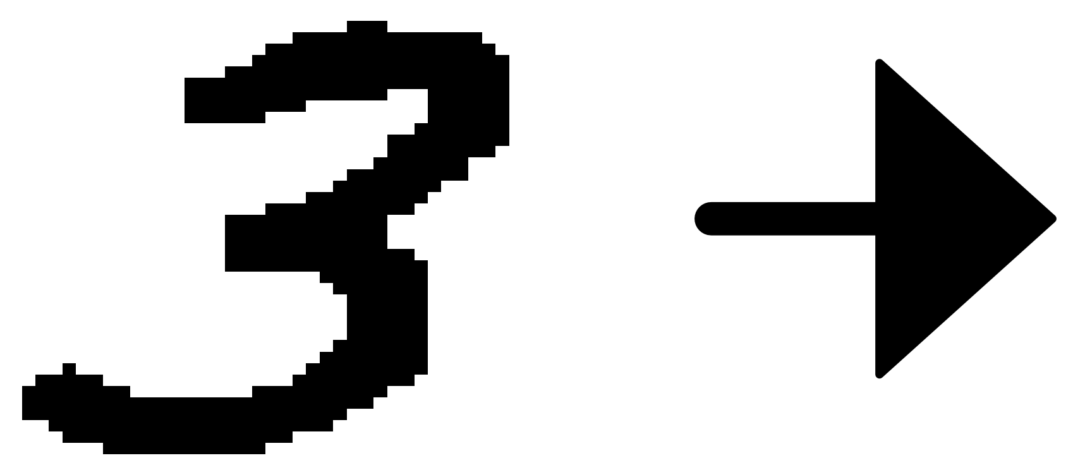
Character Recognition: Can Machine Learning Identify Human Written Characters?

IST 718 | Nicholas Brown, Advait Iyer, Vijet Muley, Christopher Negiz

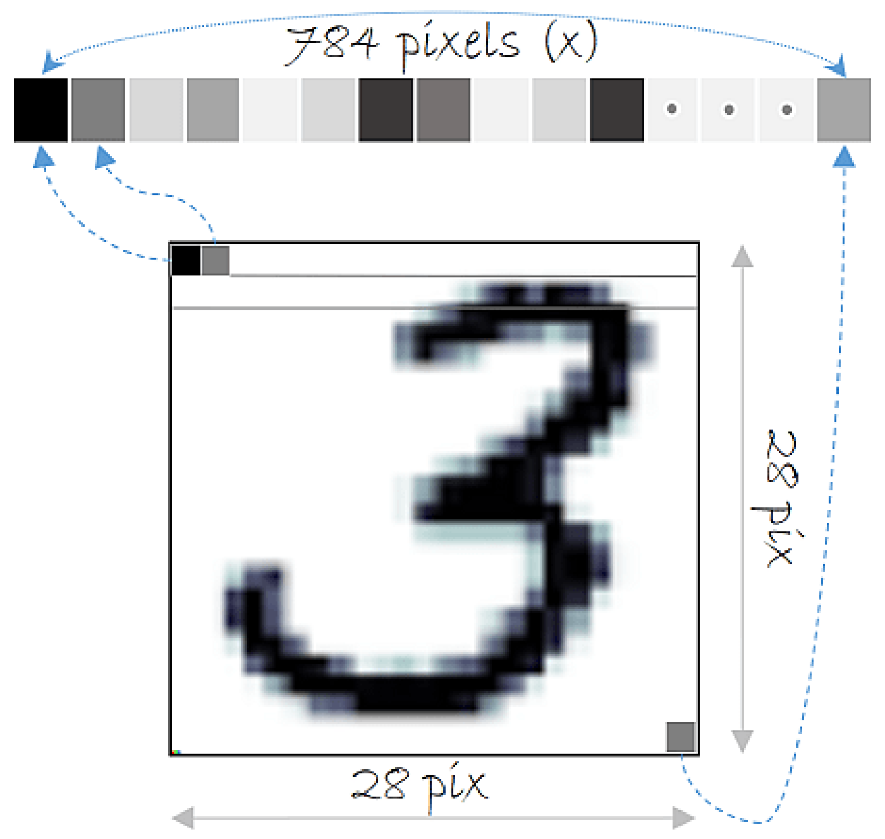
Data Description



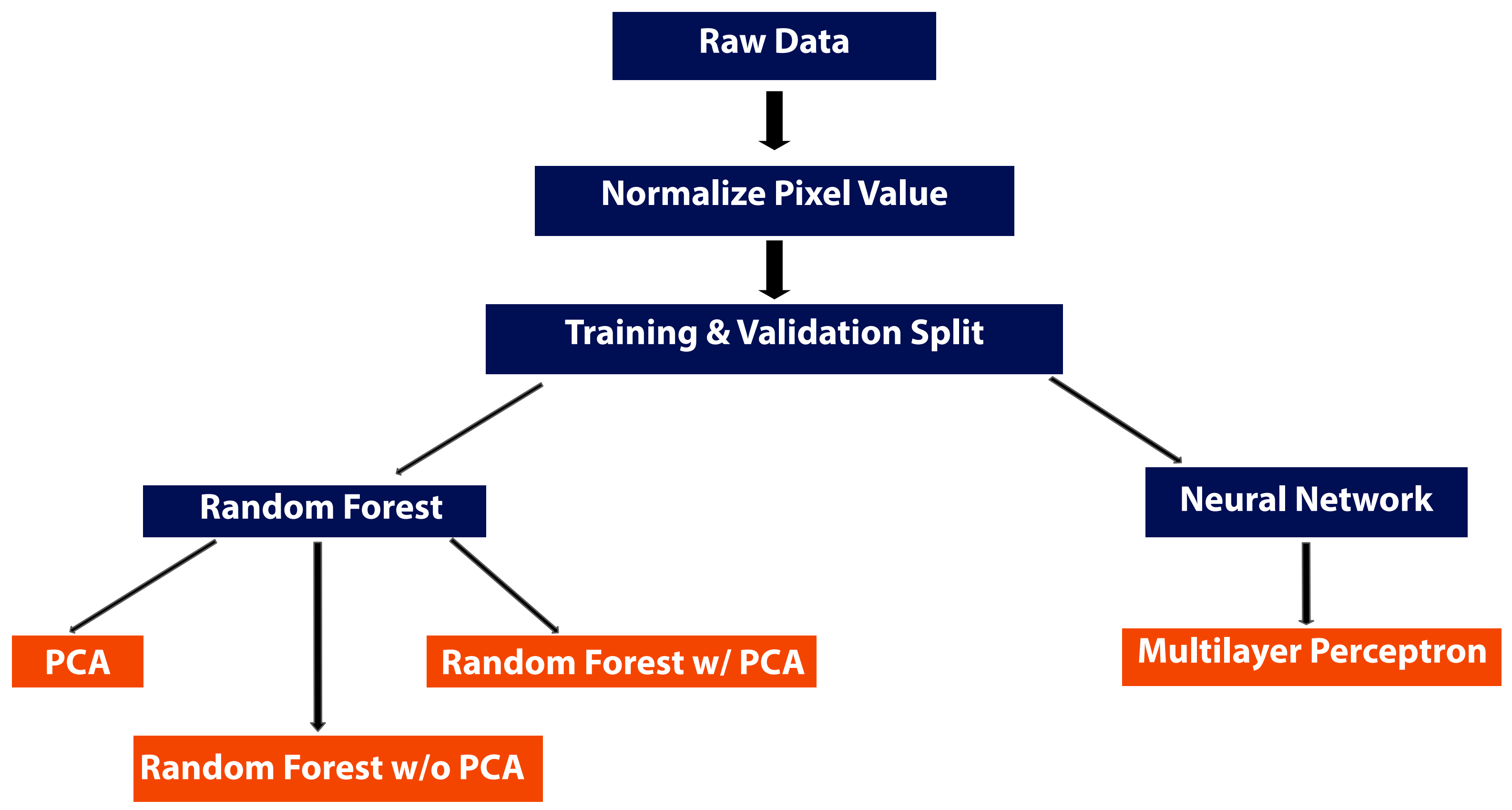
Original PNG Image:



Conversion to Flattened Form:



Data Flow



Problem and Objectives

Despite the rise of personal computers and smartphones, many people and businesses are dependant on hand written notes. For many people, written notes are faster and provide better information retention. Despite their benefits, handwritten notes fail to take advantage of modern technology. Drawbacks include inability for easy backup, sharing, and searching among other things. Currently the most popular method for sharing and backing up hand written documents involve photographing or scanning the document and converting it to an image. Now the document is shareable and easily duplicated, however, it still lacks the ability to be searched or understood by a machine. The ability for the document content to be understood by the machine provides numerous benefits including increased accessibility and searchability. We chose to tackle this problem using machine learning methods to attempt to identify individual characters from a 28x28 pixel digital image.

Data Description

The dataset contains 814,255 observations out of which 558,345 was used for training, 139,587 used for validation, and 116,323 for training. Each observation is a flattened 28 x 28 pixel grayscale image yielding 784 features. Each feature is a pixel grayscale value from 0 (white) to 255 (black). The distribution of characters was not even, with some values like '1' being very common with around 38,000 observations and other values like 'j' less common with around 1,800 observations. The dataset is numerically labeled from 0 - 61 with labels 0 - 9 corresponding to numbers 0 - 9, 10 - 35 contain the uppercase alphabet, and 36 - 61 contain the lowercase alphabet.

Conclusion

Handwriting recognition is difficult for both humans and machines. Letters look similar depending on individual's writing style, and identifying the intricate relations between specific pixel-patterns, and predicting the label correctly is difficult due to very tightly bound decision-boundaries.

The results derived from implementation of the models assist us in the following conclusion:

Principal Component Analysis Results:

When the pixel values are combined to form Principal Components, it is observed that 90% of variance in the data can be explained by using 61 PCs; while using 184 PCs will explain 98% of the total variance. The dimensionality reduction helps with computation; reducing required power and time. And also makes it easier to interpret the procedure.

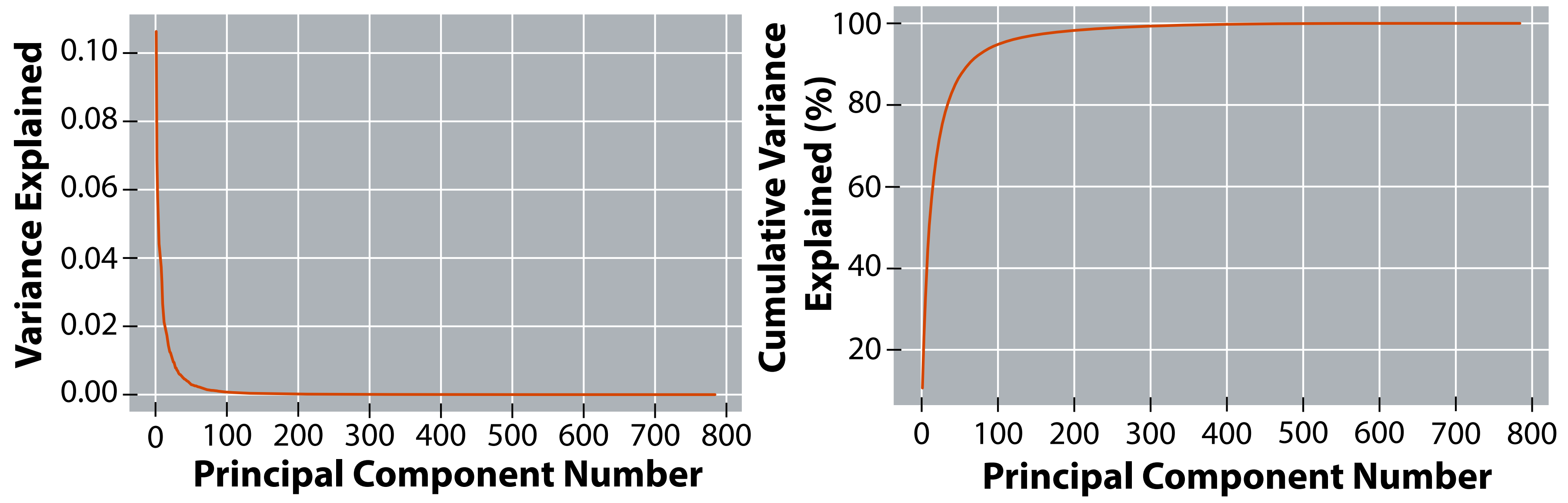
Random Forest Results:

Post PCA, we tried implementing random forest classifier. Multiple executions of instances with different parameters establish that our problem is best solved by a forest with 50 trees, where in every tree can have a maximum depth of 6. The outcome of PCA is used in this forest, in the form of 61 PCs.

Multilayer Perceptron Results:

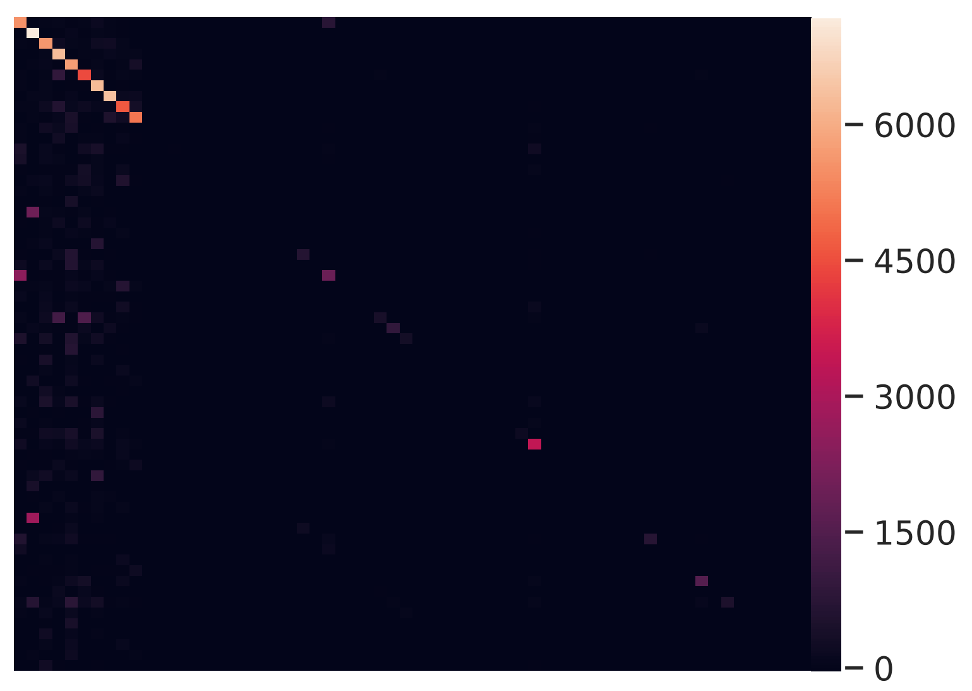
Multilayer perceptron gave an accuracy of 63%, which is the highest. We were anticipating this. MLP is the easiest form of neural networks. Achieving a decent accuracy of 63% with a relatively simpler form of neural network can help us say conclusively that usage of more complicated techniques, such as Convoluted Neural Network (CNN) will enhance the accuracy and the machine's capability to interpret human handwriting.

Principal Component Analysis



Random Forest Classifier

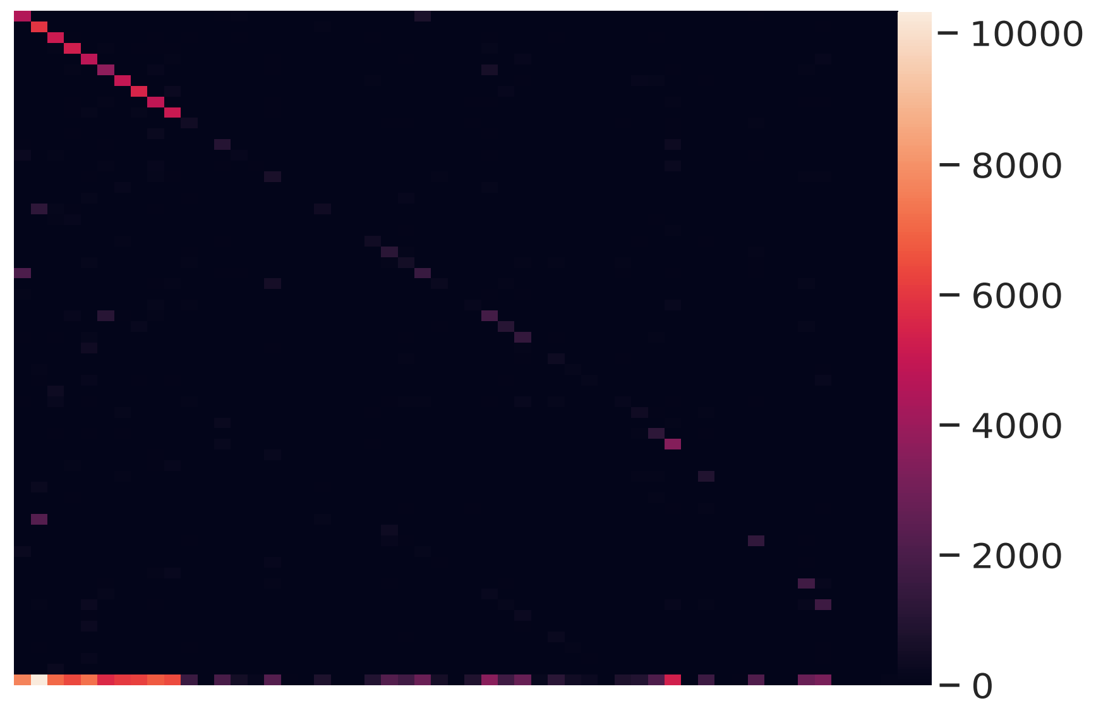
Confusion Matrix:

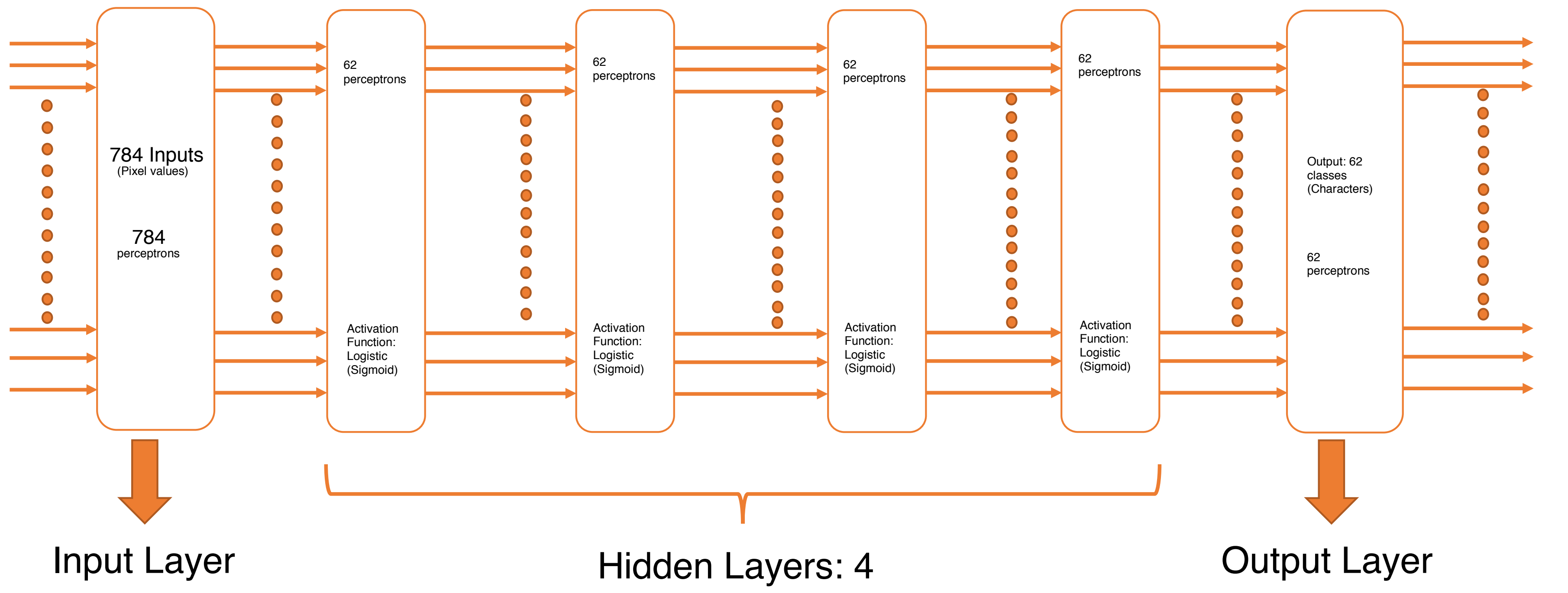


Model Summary:

Model	Parameters	Accuracy (%)
1.	Principal Component Analysis (61 PCs)+ Random Forest (30 trees, maximum depth 4)	29.94
2.	Principal Component Analysis (184 PCs)+ Random Forest (30 trees, maximum depth 4)	29.17
3.	Principal Component Analysis (61 PCs)+ Random Forest (40 trees, maximum depth 5)	33.96
4.	Principal Component Analysis (61 PCs)+ Random Forest (50 trees, maximum depth 6)	41.98
5.	Principal Component Analysis (184 PCs)+ Random Forest (40 trees, maximum depth 5)	32.45
6.	No Principal Component Analysis + Random Forest (all 784 pixel values)	39.48

Multilayer Perceptron


10000
8000
6000
4000
2000
0


784 Inputs
(Pixel values)
784
perceptrons
62
perceptrons
Activation
Function:
Logistic
(Sigmoid)
62
perceptrons
Activation
Function:
Logistic
(Sigmoid)
62
perceptrons
Activation
Function:
Logistic
(Sigmoid)
62
perceptrons
Activation
Function:
Logistic
(Sigmoid)
Output: 62
classes
(Characters)
62
perceptrons
Input Layer
Hidden Layers: 4
Output Layer