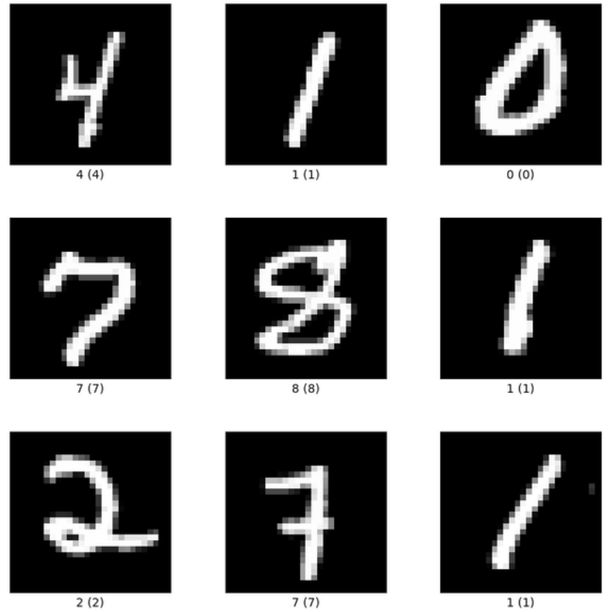
PRML Project

Handwritten Digit Classification Problem

Group ID 32

About the problem

- Handwritten digit classification involves recognizing handwritten numbers (0-9)
- We have used MNIST dataset which has 60000 training samples and 10000 test samples
- Each data point is in the form of a 1-D array having 784 pixel which can be converted into 28*28 2-D array.



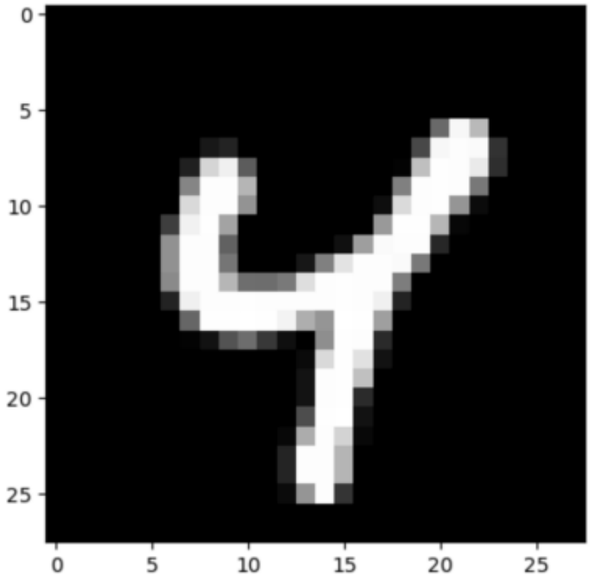
Our Approach

- Implement different traditional models like
  - K-NN
  - Decision Tree
  - SVM
  - Random forest
  - AdaBoost
  - Histogram Gradient Boosting
- Pre-processing techniques along with data augmentation
- Finally, choosing the model with best accuracy

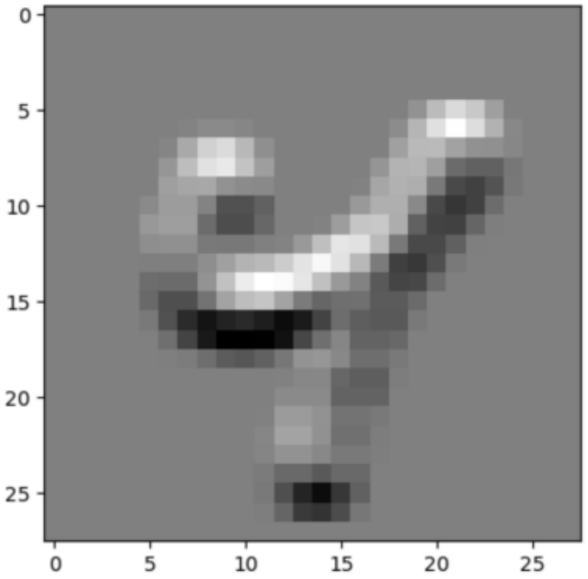
Datasets

- **Original dataset** (Normalised)
- **PCA**: 98% variance using 459 principal components
- **LDA**
- **Dataset using Edge Detector**: Prewitt Horizontal is applied for horizontal edge detection
- **Custom feature extractor**: This assigns more weight to those pixels having higher intensity.

Visualization of Feature Extractors



(a) A sample in the training dataset



(b) The same sample with edge detector applied

Original data

Edge Detector

Custom Transform

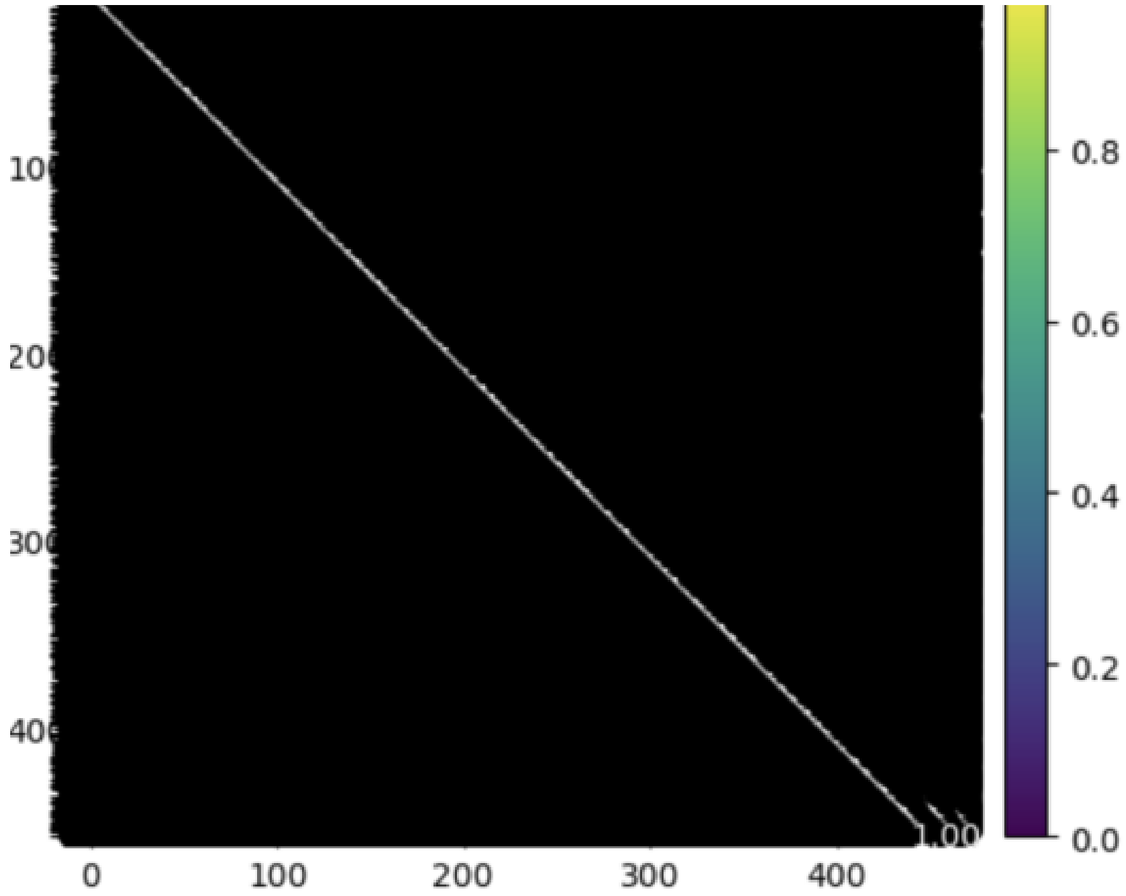
KNN classifier(k=3)

Feature Extractor	Test Accuracy
Original Dataset	0.9441
Edge Detector	0.9468
Custom Transform	0.9672

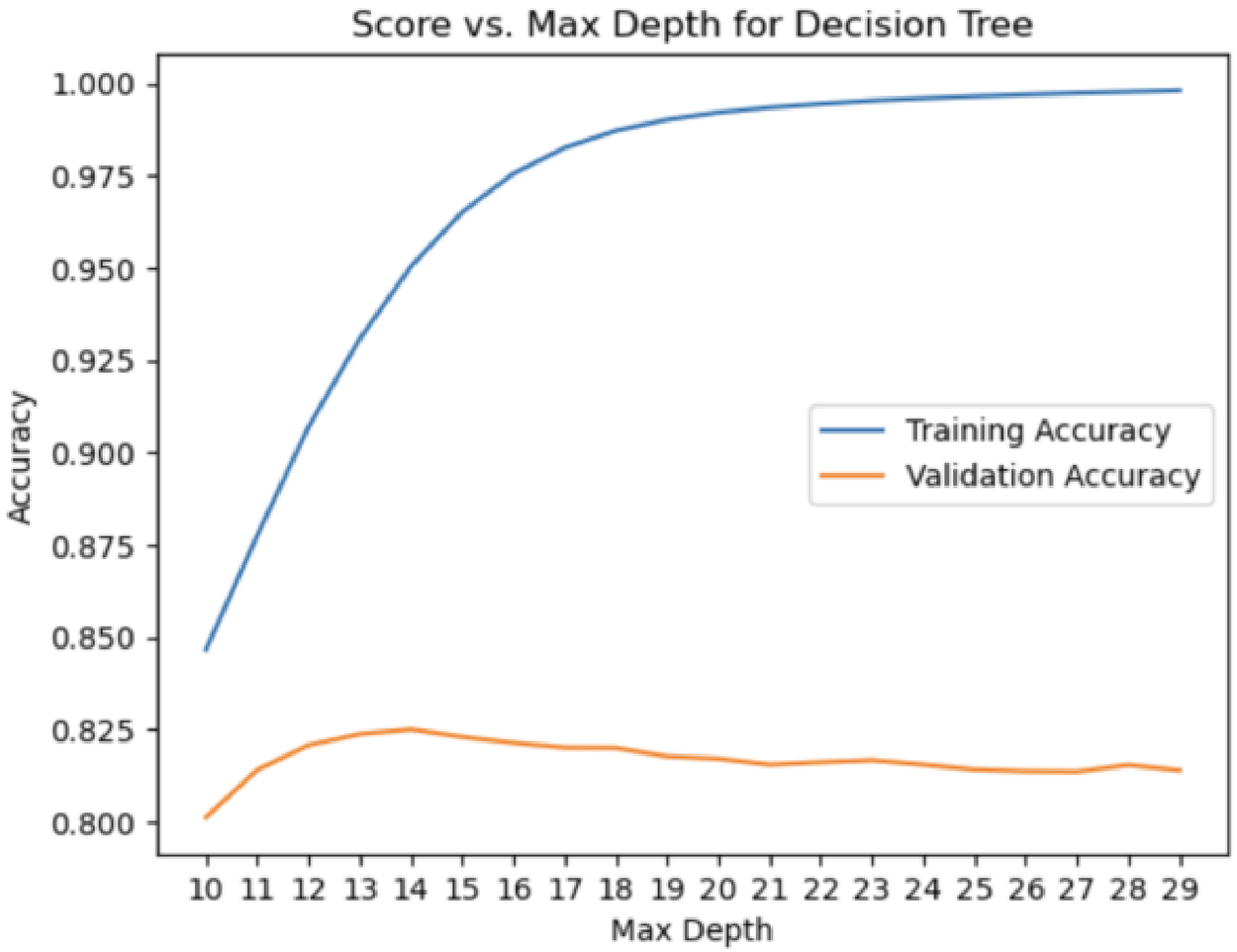
No feature extractor rejected as accuracy almost same for each

Decision Tree

Idea: PCA pre-processed data would give better results as PCA decrease correlation and avoid overfitting



Correlation matrix



Feature Extractor	Test Accuracy
PCA dataset (without any optimal n)	0.8215
Using GridSearch +PCA (max depth=14)	0.8333
Original Dataset without PCA	0.882
Custom Transform+ PCA	0.8917

The use of PCA on the DT classifier resulted in overfitting

Logistic Regression

Feature Extractor	Test Accuracy
Original Data	0.925
Edge Detector	0.1135
PCA Dataset	0.9258
Custom Transform	0.9242
LDA Dataset	0.8857

Reasoning

- Edge detection feature extractor rejected because of reasonably less accuracy in logistic regression
- LDA was also rejected because LDA assumes normal distribution of classes and identical covariance matrix
- PCA was not consistent for all classifiers and its processing time was not significantly lower either

So only two datasets are performing consistently:

- **Original Dataset**
- **Custom Transform Dataset**

Gaussian Naive Bayes

Feature Extractor	Test Accuracy
Original Dataset	0.5558
Custom Transform	0.5484

Multinomial Naive Bayes

Feature Extractor	Test Accuracy
Original Dataset	0.8236
Custom Transform	0.8365

Reasoning

- Both models produced relatively low accuracy. Hence, both models are rejected.
- Possible reasons:
  - Features are not conditionally independent
  - Features within each class does not follow normal/multinomial distribution

Random Forest+AdaBoost+ Histogram Gradient Boosting

Classifier	Original dataset Test accuracy	Custom dataset Test accuracy
Random Forest	0.9613	0.9718
AdaBoost	0.6201	0.8542
Histogram Gradient Bossting	0.9452	0.9808

Till now, Histogram Gradient Boosting on Custom transform data has given highest accuracy

SVM using RBF kernel

Feature Extractor	Test Accuracy
Original Dataset	0.9612
Custom Transform	0.9791

Reasoning

Best Models till now:

- Custom transform+KNN: 0.9672
- Custom transform+ Random Forest:0.9718
- Custom transform + SVM(RBF kernel): 0.9791
- Custom transform + Histogram Gradient Boost: 0.9808

SVM rejected because of time complexity(868.2s) For the rest three, ensemble voting done but accuracies still similar:

Hard Voting	0.977
Soft Voting	0.9806

Data Augmentation

Every image rotated and translated in 9 directions:

- up, down, left, right
- left along main diagonal, right along main diagonal
- left along counter diagonal, right along counter diagonal
- no translation at all

As a result , we got 5,40,000 more training data

Data Augmentation

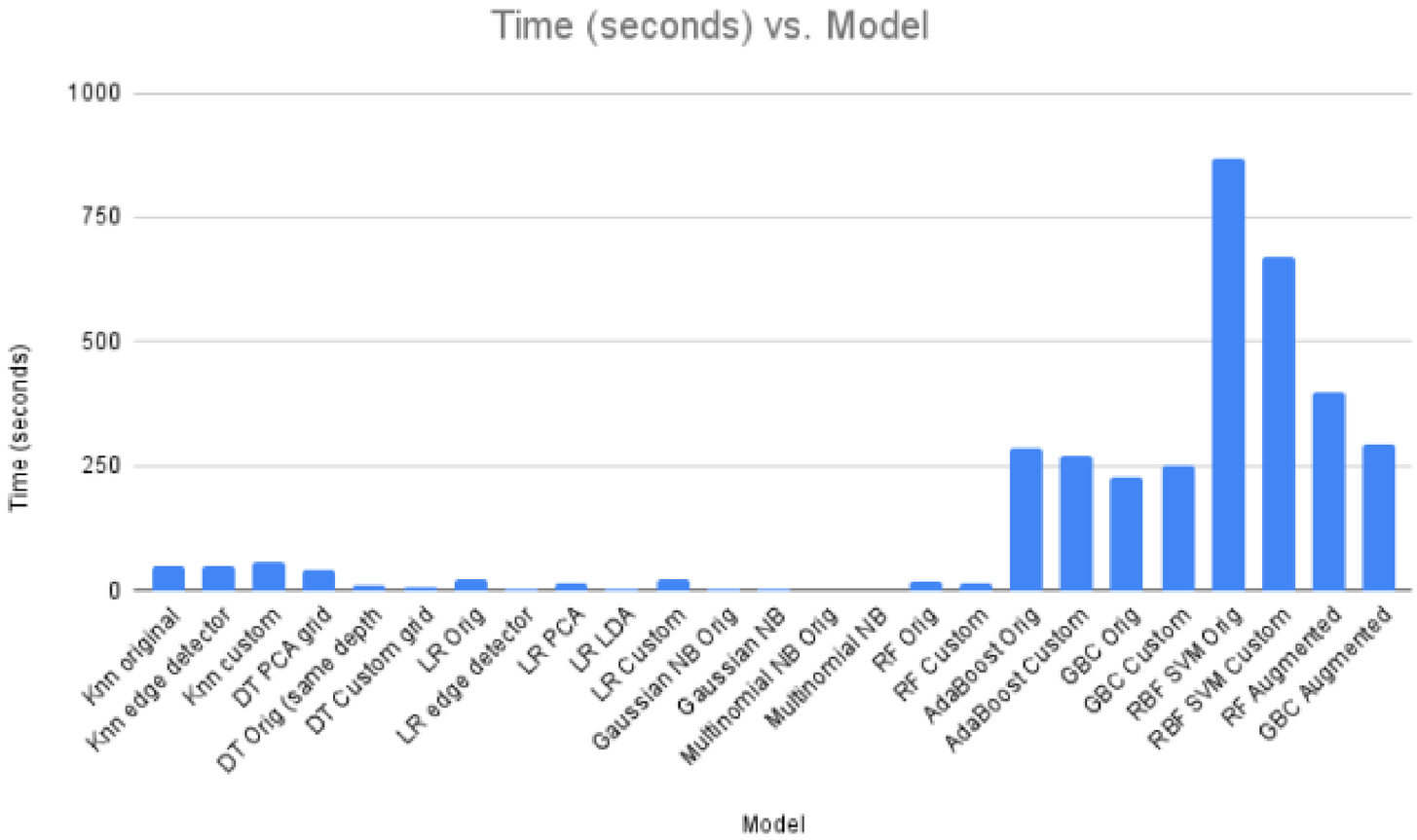
KNN was rejected due to time and memory constraints

Model	Image Augmentation	Testing Accuracy
Random Forest	Yes	0.9778
Random Forest	No	0.9718
Histogram Gradient Boost	Yes	0.9671
Histogram Gradient Boost	No	0.9808

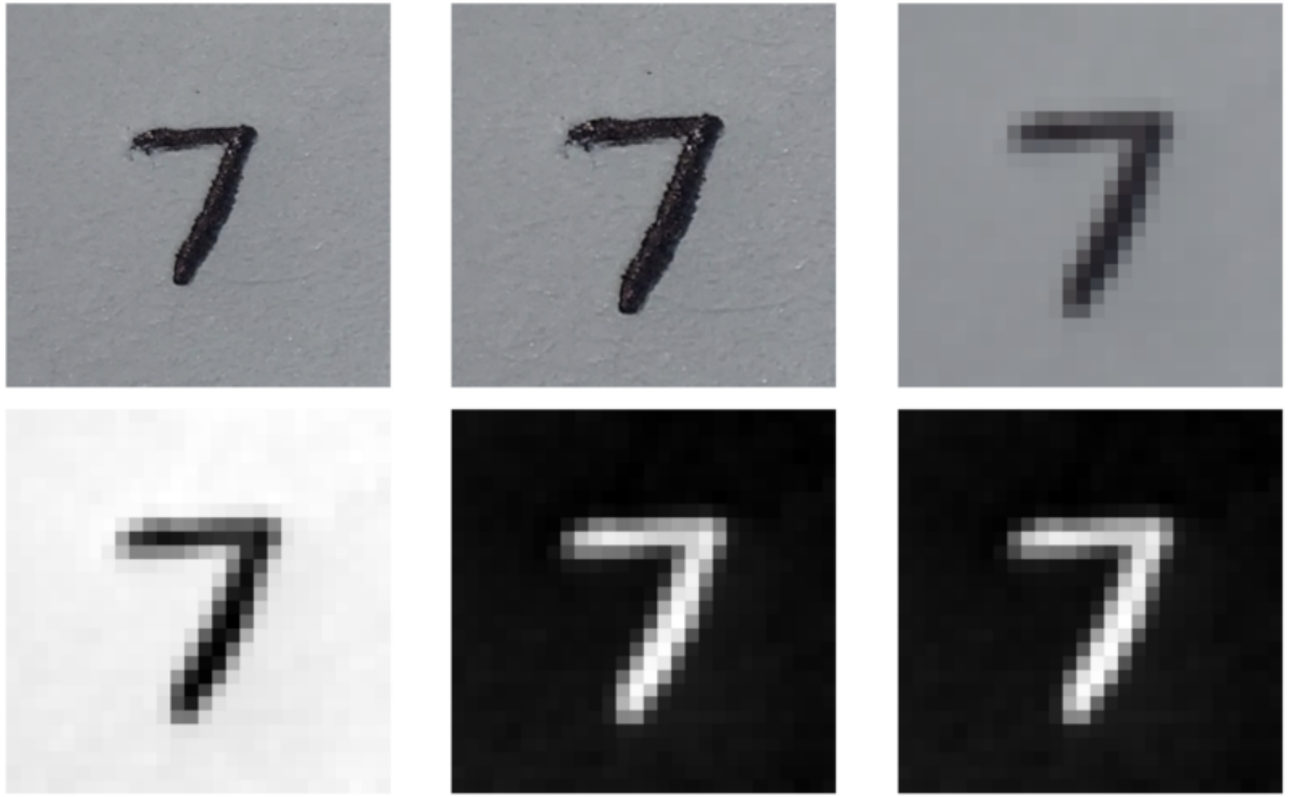
Reasoning

The model chosen: **Random Forest with custom transform** because its accuracy of 0.9778 is after performing data augmentation, which would make it more robust and generalized.

Time taken by each model:



Prediction on real image



Applying Custom Transform

Preprocessing Steps

- RGB image to grayscale
- Negative of entire image
- Min-Max scaling

Prediction on real image

Digit	Probability
0	0.0
1	0.013333
2	0.01
3	0.02666667
4	0.0066667
5	0.0
6	0.0
7	0.93333
8	0.0
9	0.01

Predicted value=7

Example of failed test cases

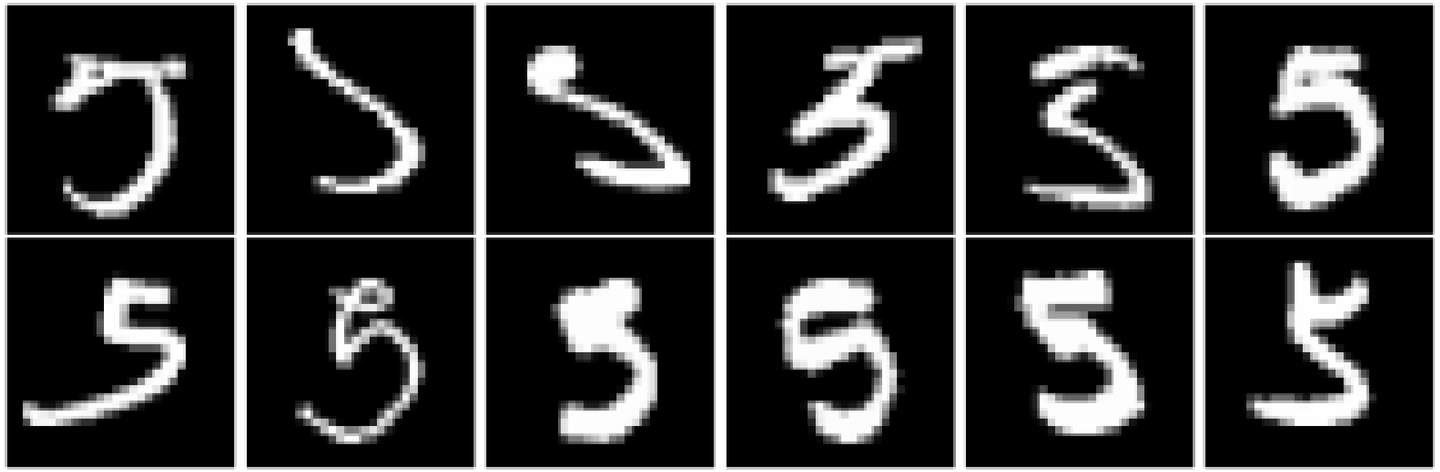


Figure 19: Samples in which 5 was misclassified as 3

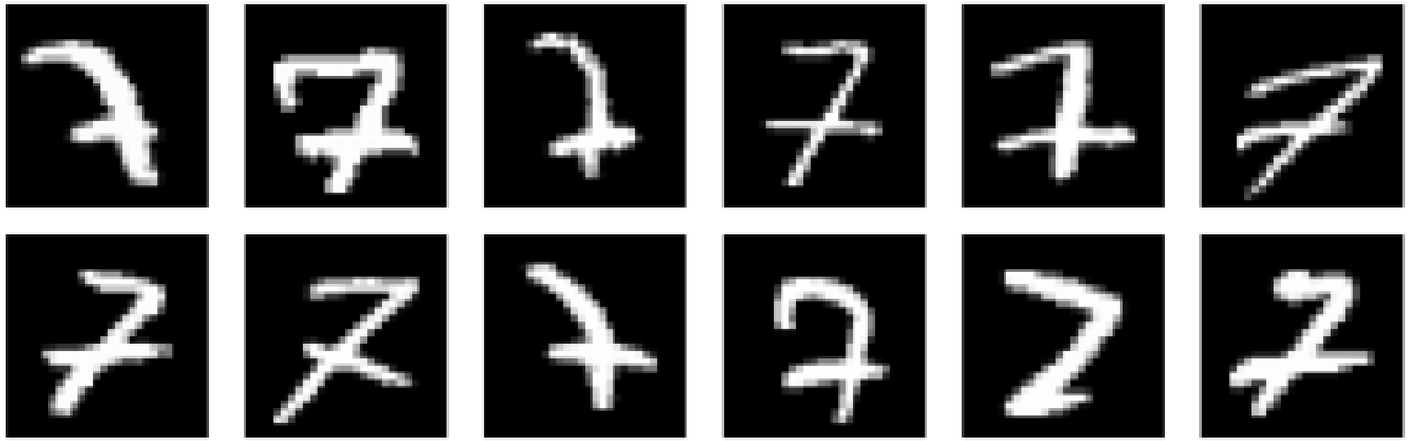


Figure 20: Samples in which 7 was misclassified as 2

Group members and contributions:

- **Rishabh Acharya**: Ideation and implementation of custom feature extractor; Data Augmentation; Training best models; Designing project page
- **Ayush Pekamwar**: Training with Decision Trees and Naive Bayes models; Designed the website which works as an interface for showing working; Video presenter
- **Raj Nandan Singh:** Training with Histogram Gradient Boosting and RBF SVM; Report documentation; Making the presentation
- **Ankit Kumar**: Brief ideation; Training with Random Forest and AdaBoost; Making the presentation
- **Pujit Jha:** Making new datasets with different feature extractors; Training with KNN and Logistic Regression classifiers; Preparing the report