CREDIT CARD DIGIT SEGMENTATION

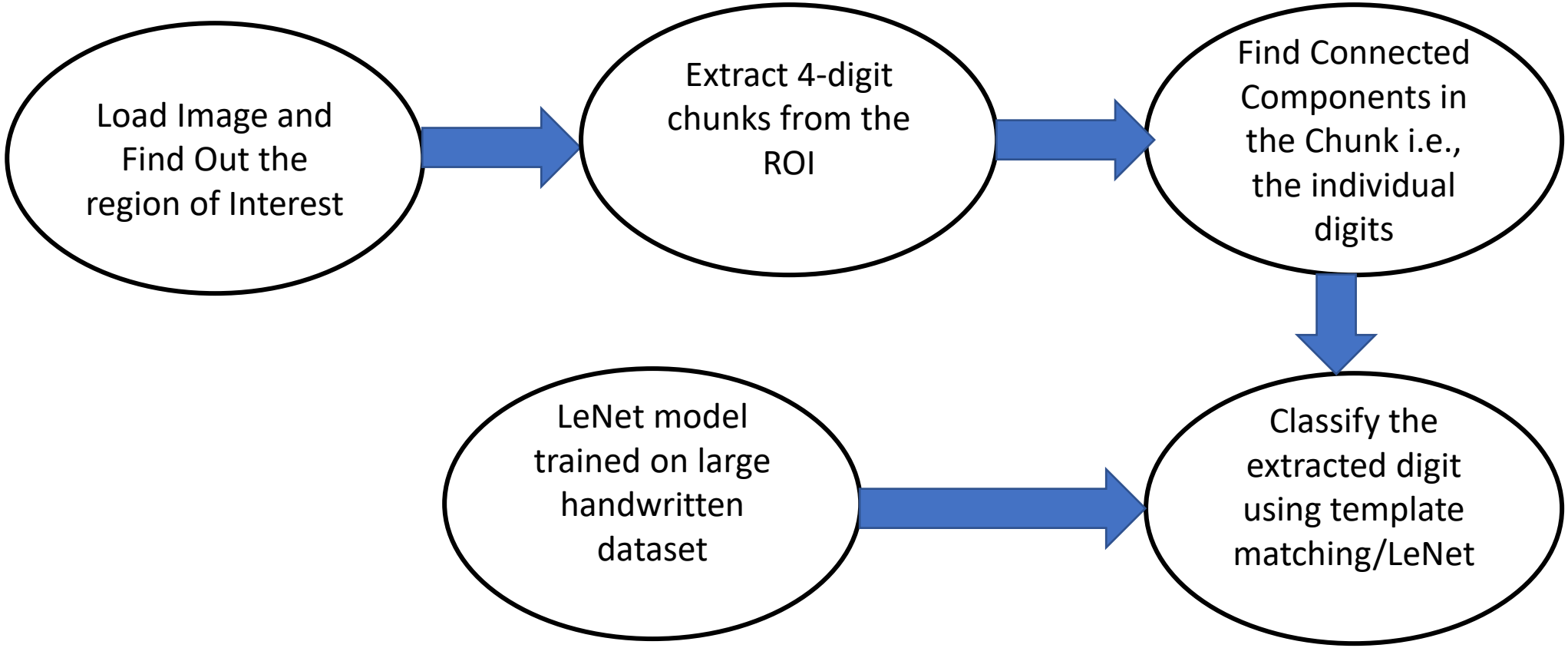
Presented By

- Saankhya Mondal
- Bhushan Deo

Introduction

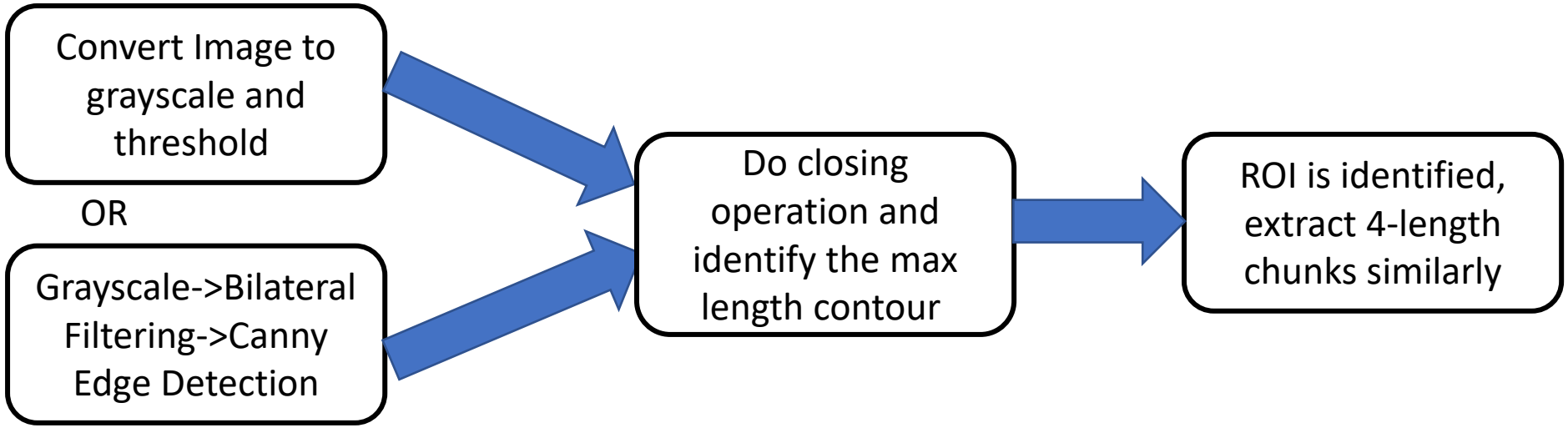
- The aim of our project is to segment and recognize each of 16 digits in credit/debit cards.
- From these detected digits, the card's company i.e.- Visa, Mastercard, or American Express and many other useful information can also be determined.
- In this project, we have created an image processing pipeline to locate the 16 digits in the card.
- Various algorithms like thresholding, morphological operations, and contour detection have been used for the purpose.
- Finally, the digits have been identified using a trained CNN classifier.

Plan of Action



Preprocessing

Identifying the 4-length chunks in the image:



Finding Out the Region of Interest(ROI)

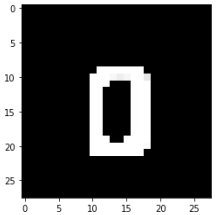
- We tried to first identify the 16 length ROI automatically by drawing a bounding box around the largest length contour as follows:

- But we encountered many problems in general such as spurious long contours, contours wrongly connected due to closing, etc.
- Rather than handling each case separately we decided to resize the image to a fix size ~(179x280) and assume the approximate location of the ROI (16-chunk)
- After the ROI is found, we tried to find the 4 length chunks as bilateral filtering->canny->contours

Extracting each digit and classification

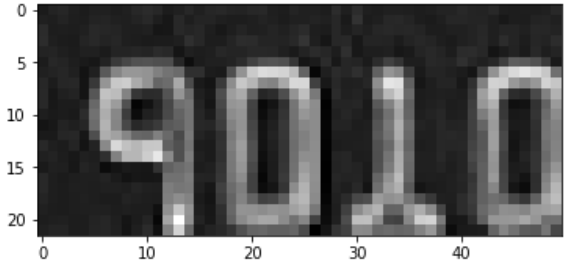
- So we extracted the ROI template:

- We centred each of the digits identified by connected components from the 4-length chunks, to provide some background info.



- We then tried template matching and LeNet to classify the digits accordingly.

A 4-digit chunk highlighting poor quality after extraction:



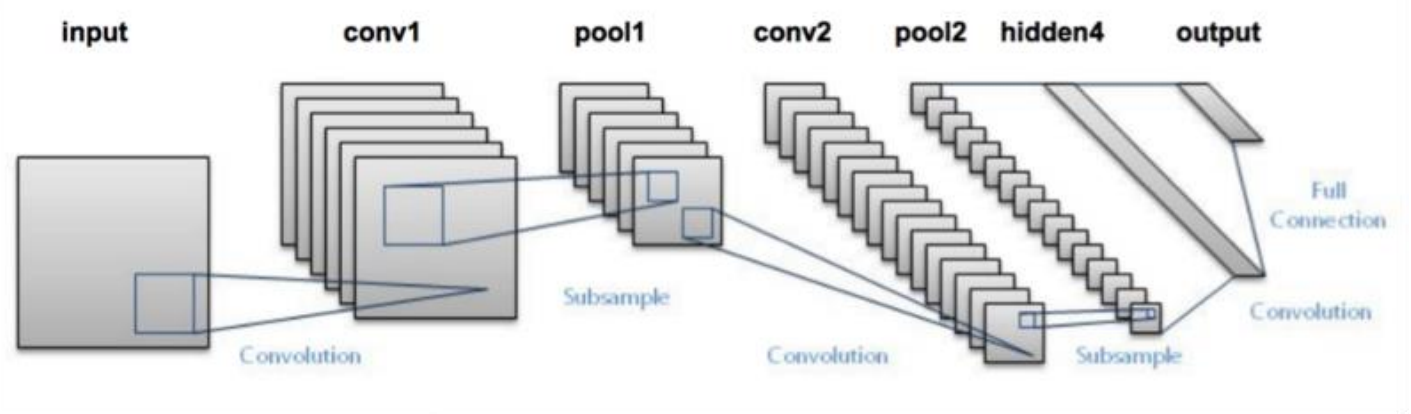
Classification using Template matching

- First, we used standard OCR-digit images templates to match with our extracted digits. But as the extracted digit had a lot of non-uniformity (because it is drawn from a small part of the large credit card image) we were not getting satisfactory results.
- We then extracted templates themselves from the credit cards but faced the following problems:
  - Each digit had different sizes across various images
  - Each extracted digit varied across images due to the closing/dilation operations which were used to coalesce broken parts of same digit

  So we moved to adopting LeNet based classification of the digits.

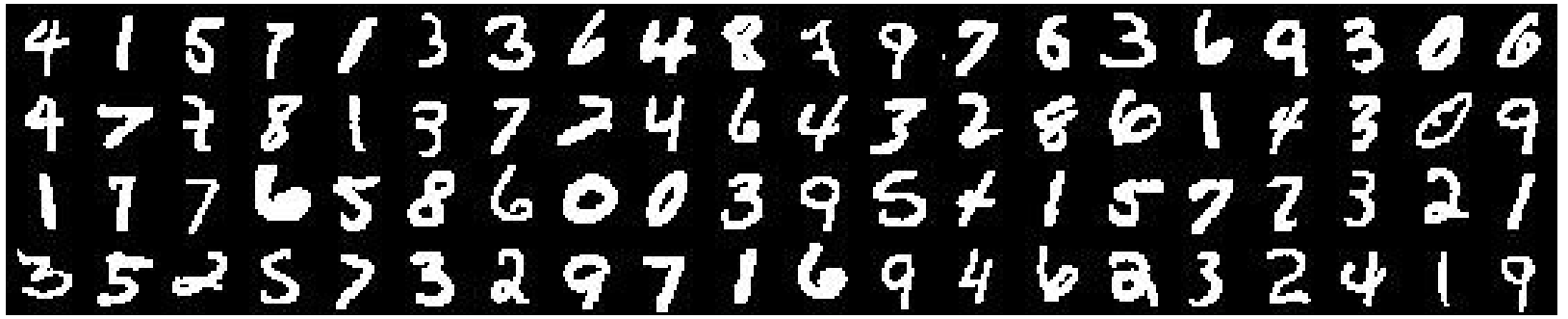
LeNet-5 Architecture

- LeNet-5 is a simple convolutional neural network (CNN) architecture.
- It consists of two sets of convolutional, activation, and pooling layers, followed by two fully-connected layers, and finally a soft max classifier.
- Size of the input is 28x28 and convolution kernels used are of size 5x5.
- Activation function used is ‘tanh’ activation function.
- Average pooling with kernel of size 2x2 is used.



MNIST dataset and Training

- MNIST dataset is a large dataset containing images of handwritten digits 0-9.
- There are a total of 70,000 images in the dataset. 60,000 images for training and 10,000 images for validation purpose.
- Each sample is a 28x28 grayscale image of one of 10 digits.
- This dataset has been used to train the LeNet model which will classify each of the 10 digits.

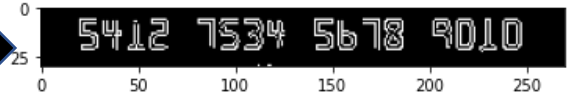


Classification using CNN

- We obtained better (but not 100% accurate results) after training with the LeNet model.
- The challenges we faced are:
  - OCR font is a bit different from handwritten digits in general
  - To do transfer learning/fine-tuning we should have large number of samples of 28x28 size (standard LeNet size)
  - Also after resizing images we found sort of aliasing effect (some blurred boundaries on the periphery of the digits ,etc.)

Result

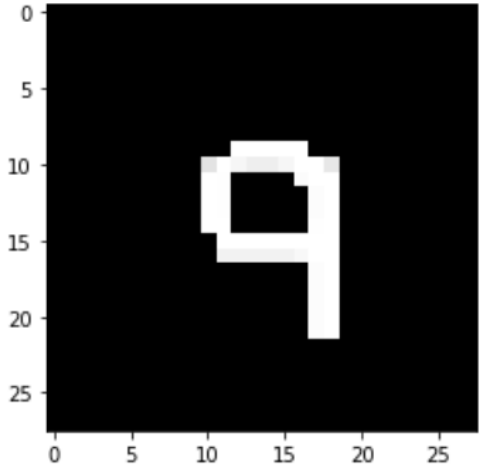
Extract general region of interest



Bilateral Filtering and Canny Edge Detection

Close

4 chunk digit extraction Using connected components



LeNet based classification

9

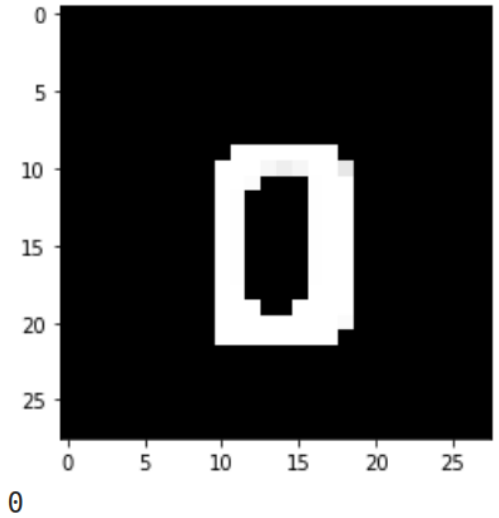
Result

Extract general region of interest

Bilateral Filtering and Canny Edge Detection

Close

4 chunk digit extraction Using connected components



LeNet based classification

Conclusion and Comments

- The methods work best when background is smooth and has no illumination problems.
- The LeNet-5 based classification is not accurate which was unexpected as it was trained on a large set of handwritten images and not OCR-A fonts.

THANK YOU