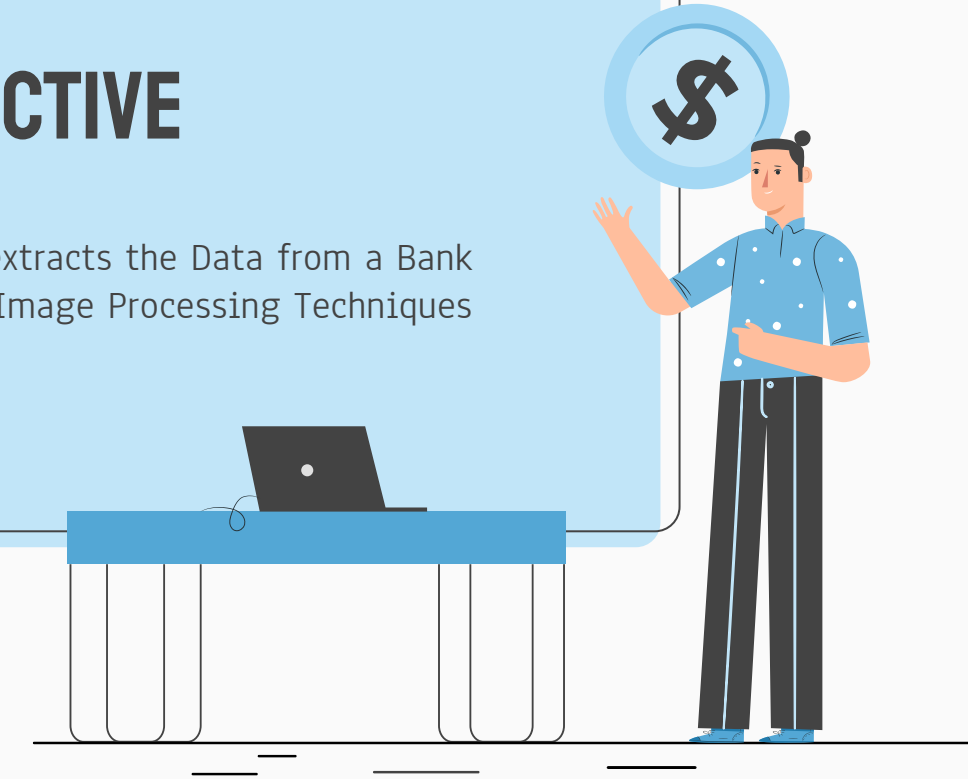
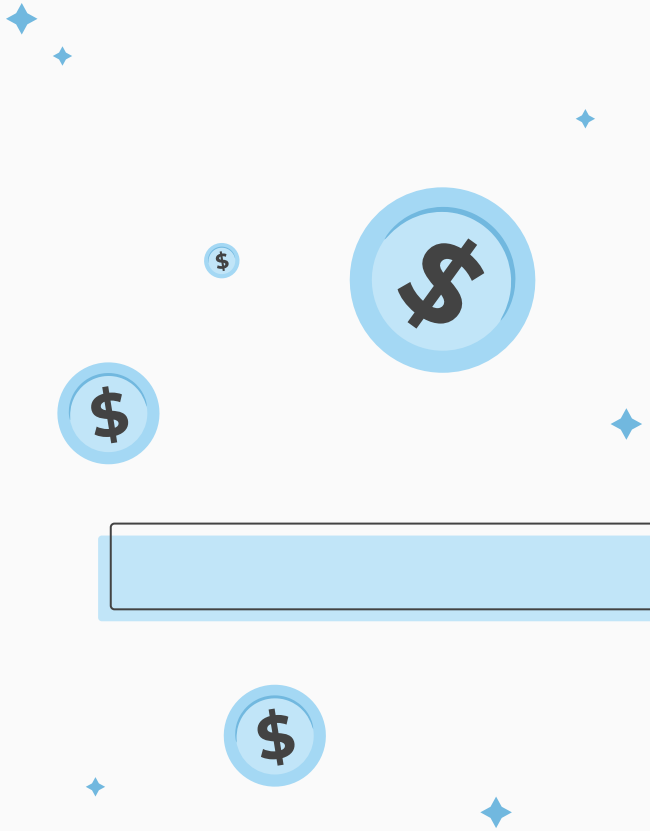
CHEQUE PROCESSING SYSTEM

Smit Rana
Sharatkumar Patangi

OBJECTIVE

To create a system which extracts the Data from a Bank Cheque using MATLAB and Image Processing Techniques





"I don't think necessity is the mother of invention. Invention, in my opinion, arises directly from idleness, possibly laziness, to save myself trouble."

—AGATHA CHRISTIE

44,000

Number of Registered Banks in the World (2018 Survey)

5,153 MILLION

Cheques Deposited during 2017

8,438 $ BILLION

Money Deposited via Cheques

- According to the data provided by the Federal Reserve System

STEPS TOWARDS THE SOLUTION

01 CLICK IMAGE

The basic and the manual step - Click the image of your Cheque

02 PREPROCESS

First Step of our System - Pass the cheque through various Image Processing Algorithms

03 OCR

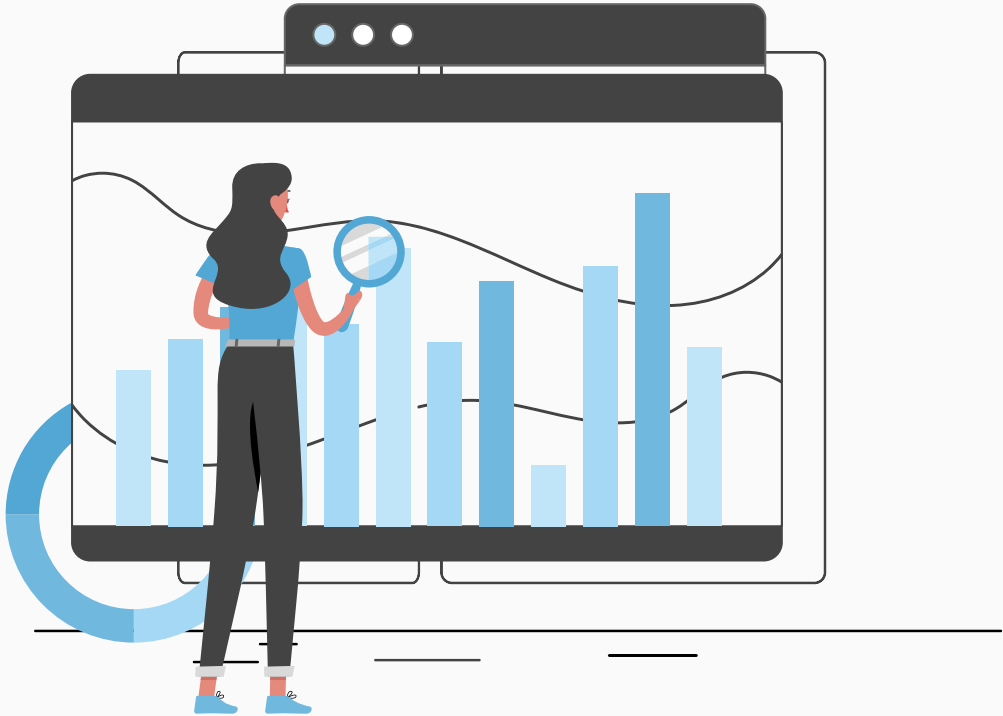
Pass the PreProcessed Cheques through **Optical Character Recognition**

04 DATABASE

Store the data and pass it for the final verification

PREPROCESSING

The Clicked Image is not ready for OCR, so it needs to be passed through various algorithms, selected after many experiments and trials and errors



THE 4 STEPS OF PREPROCESSING

01
NOISE REMOVAL

- imgaussfilt
- bwareaopen

02
DIMENSION REDUCTION

- Rgb2gray
- im2bw

03
REDUCE PROCESS POWER

- Imresize
- regionprops

04
CROPPING

- cat & max
- imcrop

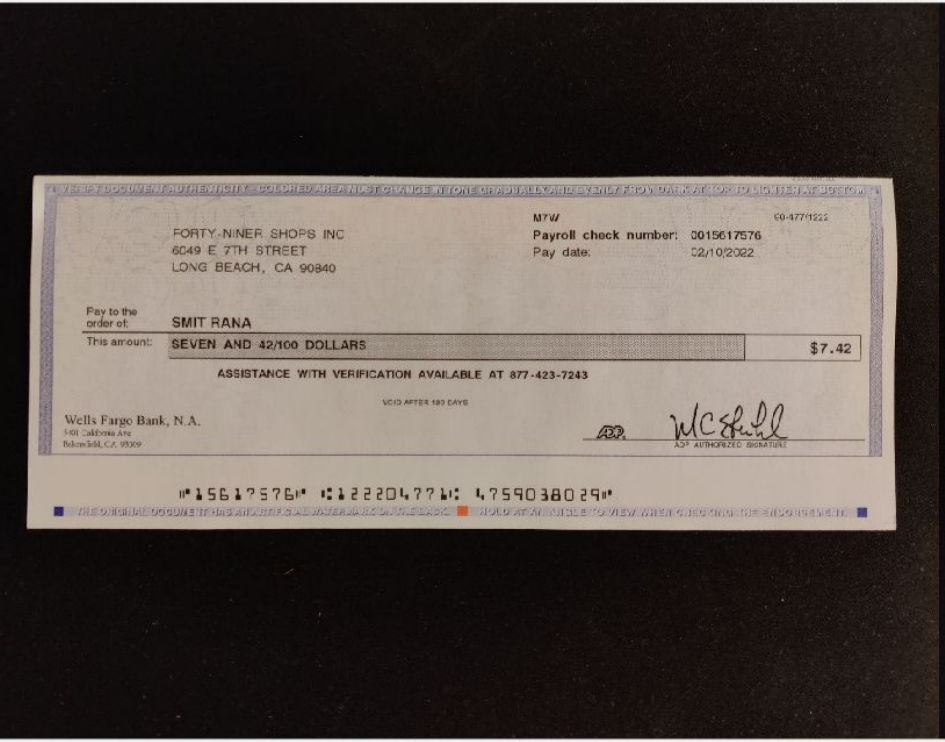
01. NOISE REMOVAL

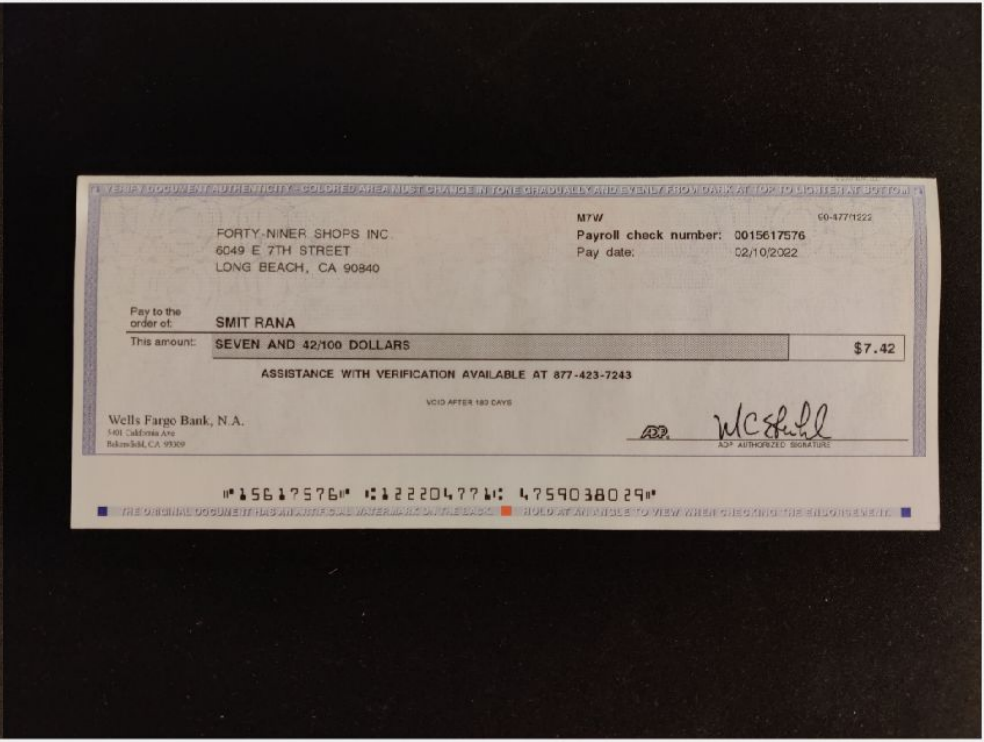
- No Image clicked by a camera can be without noise especially in the background which can interfere with our processing.

- So we pass our image through some noise removal techniques like

1. **Gaussian Filter**
2. **BWAreaOpen**

To remove the noisy pixels which are visible on right side of the image





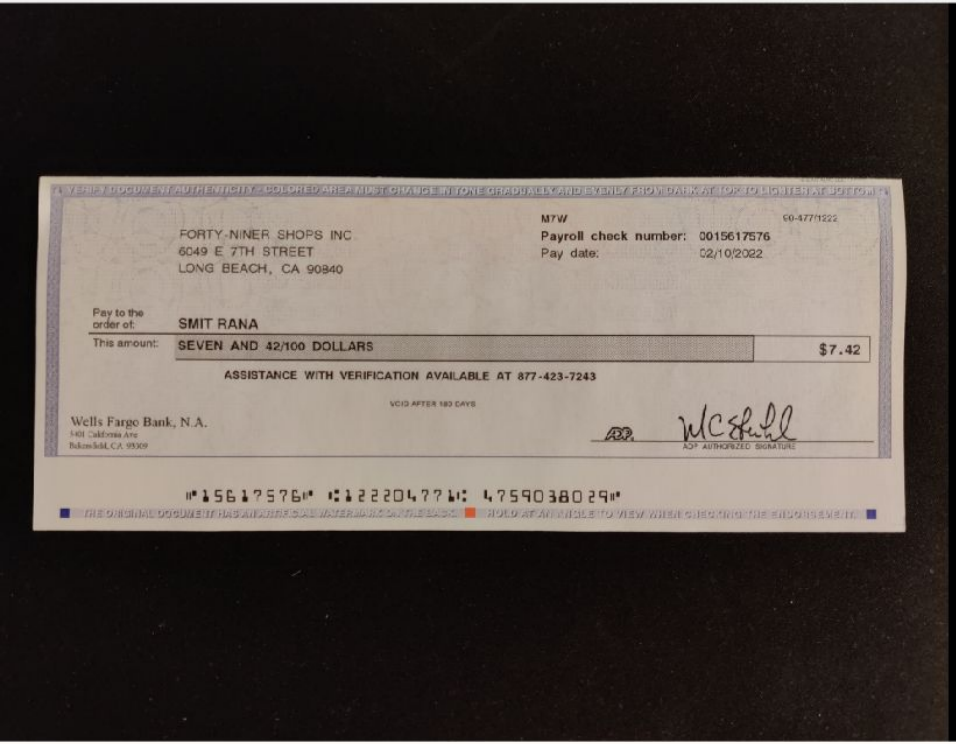
02. DIMENSION REDUCTION

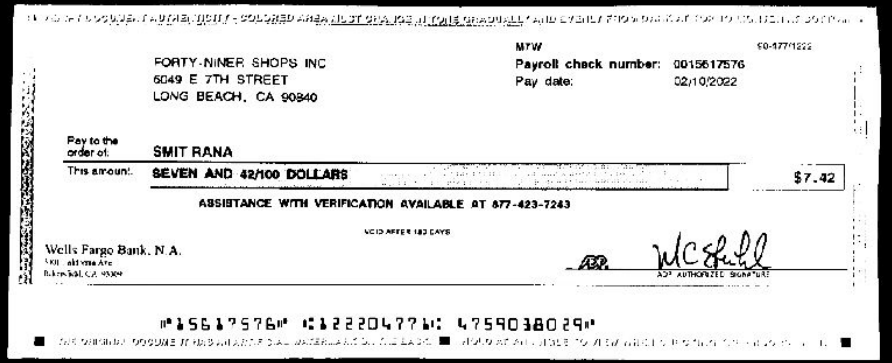
- There is no need for keeping the 3rd Dimension for an Image, It is just extra and unnecessary data which will be processed

- So we pass our image through algorithms which will remove the unneeded dimension

1. rgb2gray
2. im2bw

To make finding the border coordinates of the cheque easily



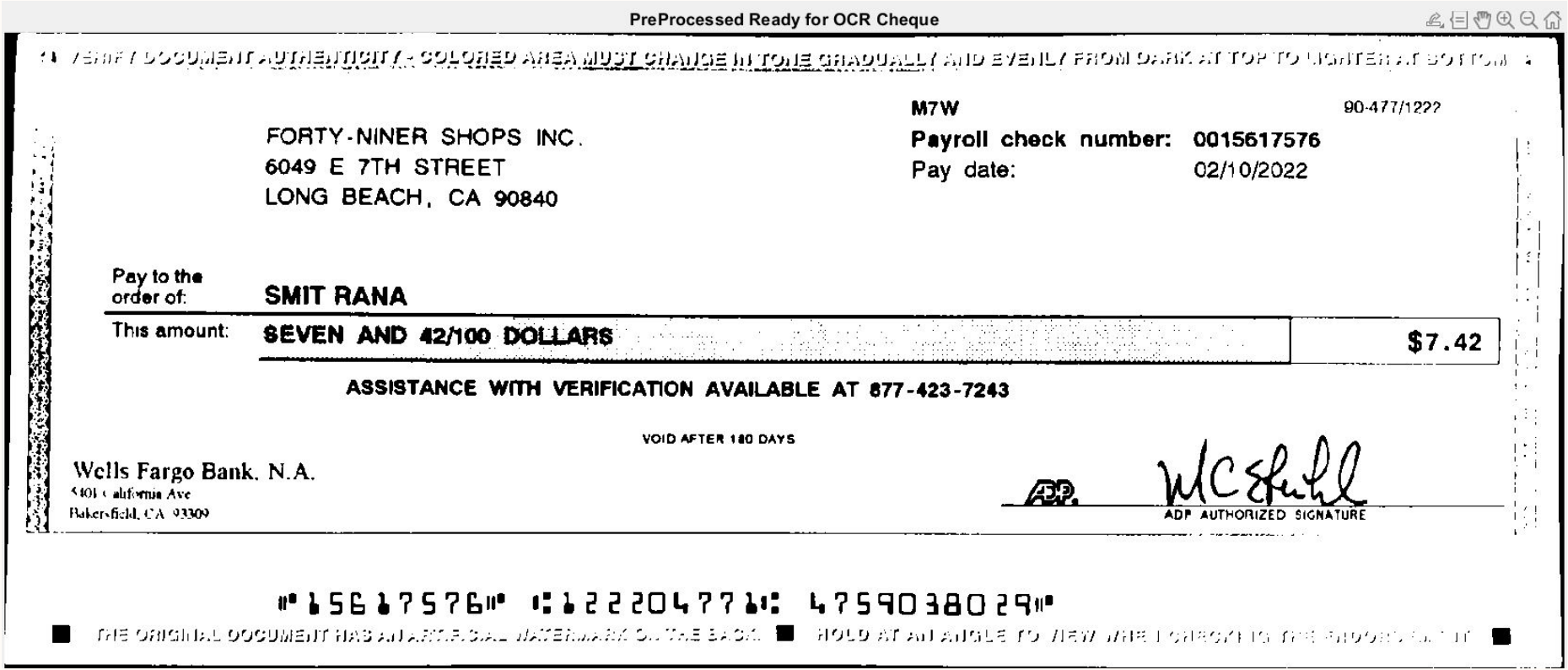


03. REDUCE PROCESS POWER & 04. CROPPING

- To further reduce processing power, We will reduce the size of the image which will make cropping the cheque as well as the sections consistent across more cheques

- We will use the following functions to achieve this

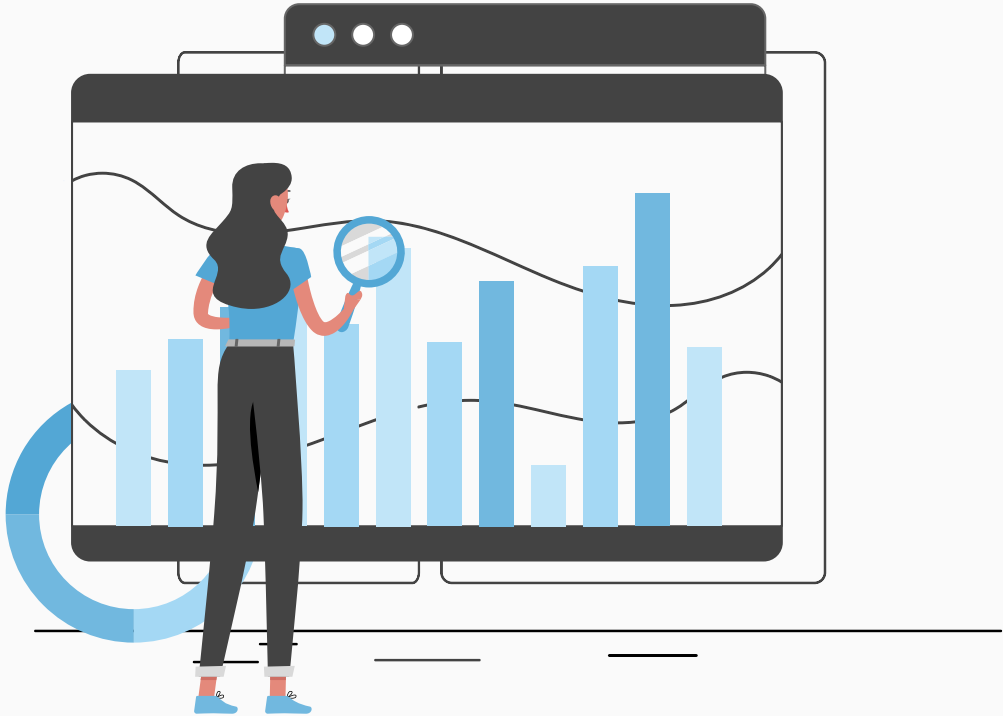
1. **imresize**
2. **regionProps**
3. **cat & max**
4. **imcrop**



PS: This Cheque has 100% Accuracy of how much a computer engineer earns in 2022

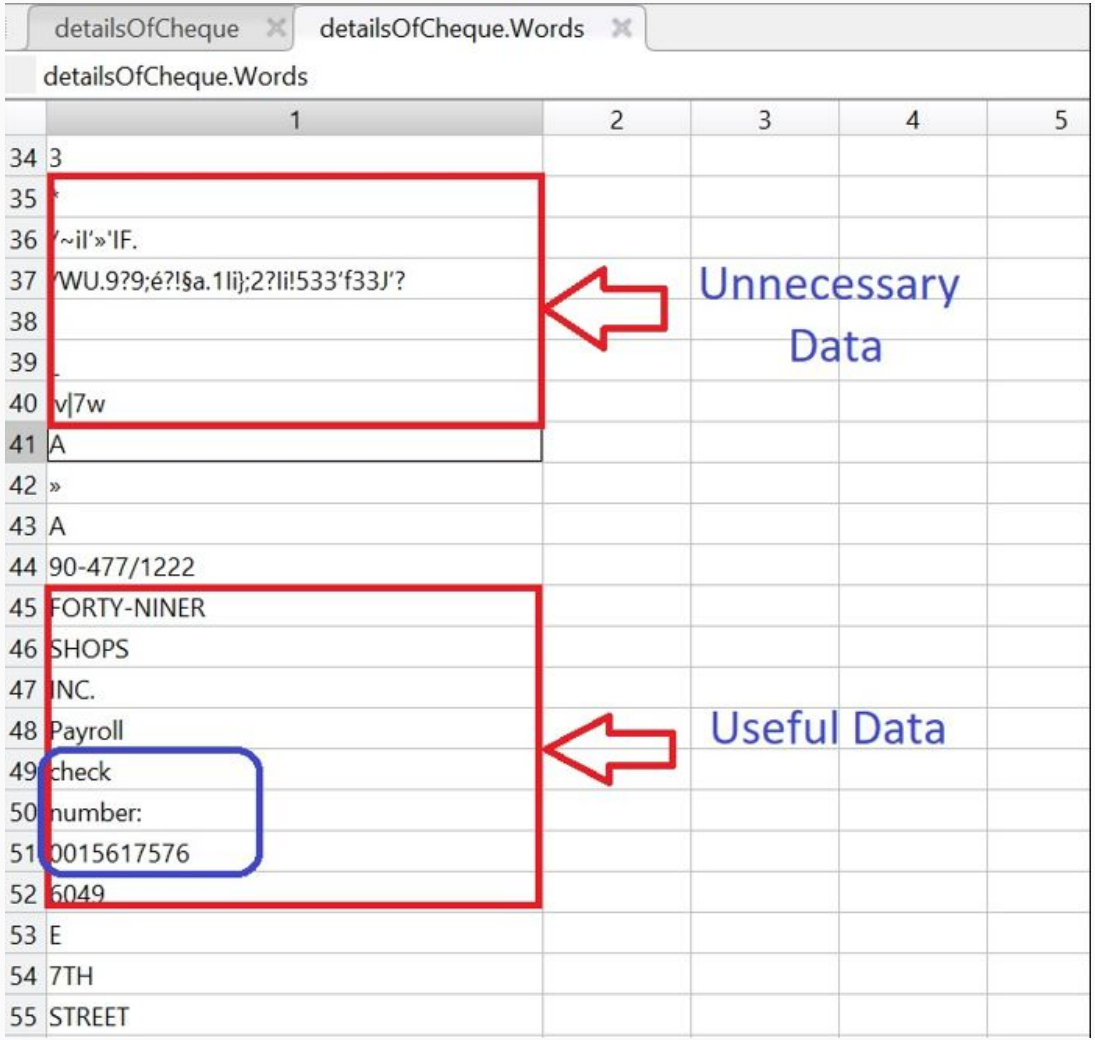
OCR

Now that our cheque image is all suited up and ready for passing through Optical Character Recognition which is the **Heart of our Project**

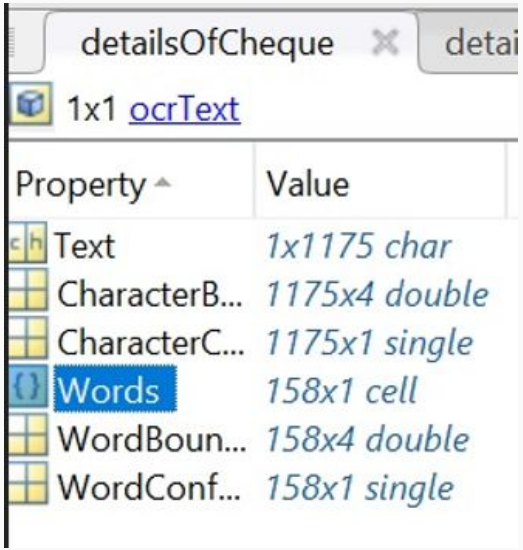


OCR : INTRODUCTION

- Widely used method in Computer Vision, Pattern recognition, and AI.
- Simple yet powerful method.
- Recognise and extract characters from a scanned document or an image.
- Converts written letters & symbols in a format that can be processed by computers.
- A newer version of OCR software uses a neural network that is capable of reading a whole sentence instead of a single character.



OCR OUTPUT ON THE WHOLE CHEQUE



OCR : CHALLENGES

- PreProcessing not enough for the whole cheque
- **SOLUTIONS**
    1. **Further cropping the image by roughly 5-10% from the edges**
    2. **MSER (maximally stable extremal regions)**
    3. **Extracting the Sections**
- To Remove unnecessary data we need a database with sections of the cheques

OCR : EXTRACTING THE SECTIONS

- We have a limited amount of data fields that needed to be extracted
- Extracting the Sections also makes the OCR much faster
- We are using imCrop to extract the coordinates and fill our database with the required data
- For our experiment we are extracting 3 fields : *Name*, *Amount*, *Date*

chequeBounds

4x3 table

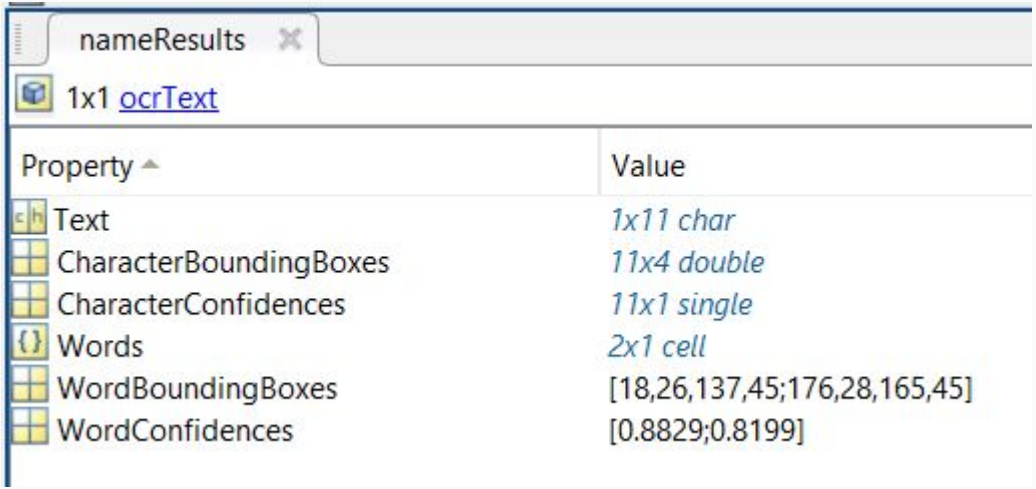
	1 Fields	2 boundariesWF	3 boundariesUS
1	"Bank Name"	"Wells Fargo"	"NULL"
2	"Name"	"578.5100, 549.5100, 1.076980000000000e+03, 77.980000"	"797.510000000000, 582.510000000000, 1118.98000000000, 122.980000000000"
3	"Amount"	"2948.51000000000, 660.510000000000, 446.980000000000, 86.9800000000000"	"2759.51000000000, 618.510000000000, 461.980000000000, 101.980000000000"
4	"Date"	"2692.51000000000, 278.510000000000, 305.980000000000, 76.9800000000000"	"2249.51000000000, 360.510000000000, 581.980000000000, 110.980000000000"

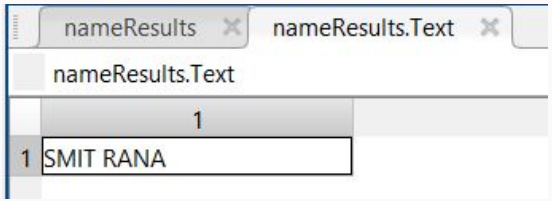
OCR : EXTRACTING FIRST SECTION

- Loading Bounding area for the Name field from the Database
- Running IMCROP on the cheque for the Bounding Area to get this result
- This cropped Image will be passed through OCR

OCR : RESULTS

- Running the OCR Algorithm on the cropped Section, OCR contents recognizes the text, its Location, and the Accuracy of the Words as well as the Characters and Provides us with the ocrText type output





OCR : RESULTS

nameResults.CharacterConfidences

	1	2	3
1	0.9015		
2	0.9048		
3	0.9228		
4	0.9099		
5	NaN		
6	0.8276		
7	0.9014		
8	0.9114		
9	0.9115		
10	NaN		
11	NaN		
12			
13			

nameResults.CharacterBoundingBoxes

	1	2	3	4	5	6
1	19	27	34	45		
2	57	28	43	45		
3	105	28	12	45		
4	120	29	37	44		
5	157	27	20	46		
6	177	28	36	45		
7	217	29	41	45		
8	261	30	37	44		
9	301	30	42	45		
10	343	27	0	0		
11	343	27	0	0		
12						
13						
14						

OCR : STORING IN DATABASE

- Output from the OCR is saved in the database
- This can be passed for final verification of the cheque (Once all field's data is extracted)

ExtractedData

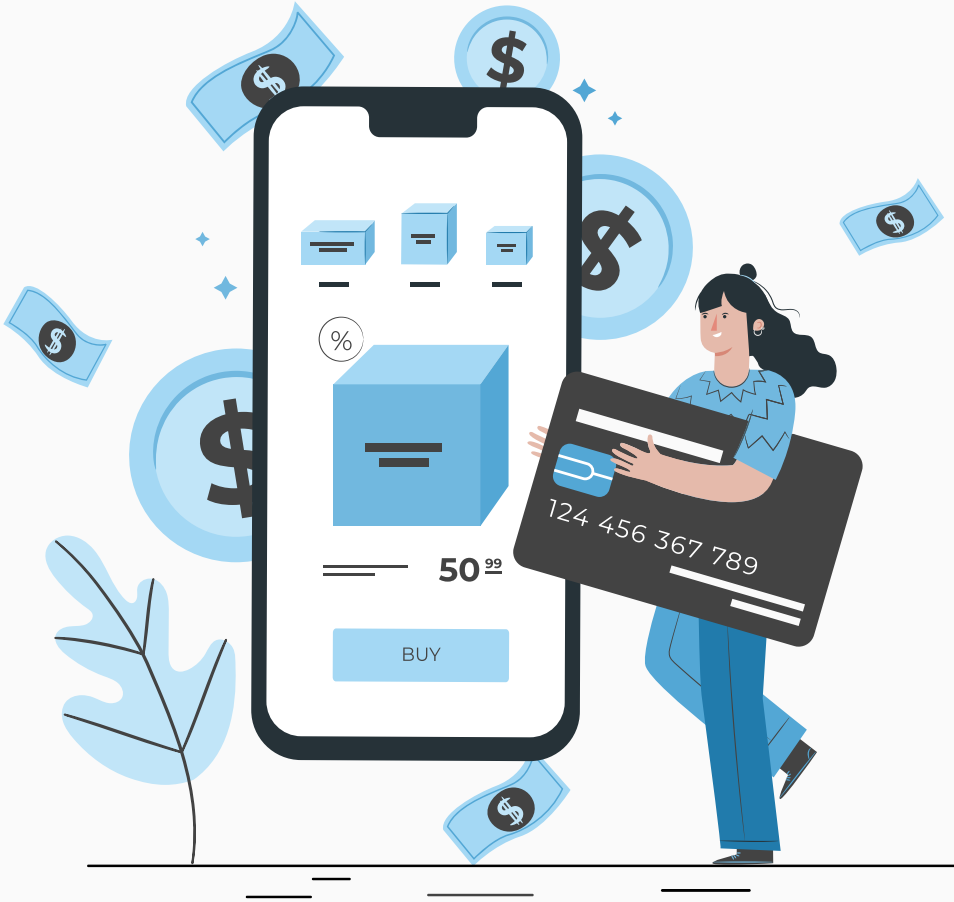
4x2 table

	1 Title	2 Data	3	4
1	"Bank Name"	"Wells Fargo"		
2	"Name on the Cheque"	"SMIT RANA"		
3	"Amount"	"$7.42"		
4	"Date"	"02/1 0/2022"		

CHALLENGES

- Finding a Sweet Spot for the Preprocessing (Less time consuming and Efficient)
- Finding Database for Testing and Training
- Better Noise Removal Tools

TIME FOR A DEMO



FUTURE PROSPECTS

- Dynamically Finding the Sections
- Making a Standalone App or Module for an App
- Improving Preprocessing for any background
- Validation of Cheque and the Data (For which more and Proper Dataset Needed)
- Signature Verification : Requires many images of the same signature

THANKS

Does anyone have
any questions?