TABLE DETECTION, INFORMATION EXTRACTION & STRUCTURING USING DEEP LEARNING

By:

HINA ABDULLAH – B16101053
IFRAH SOHAIL – B16101057
SYEDA MANAHIL MASOOD – B16101177
WAJEEHA JAVED – B16101184

Supervisor:

Dr Farhan Ahmed Siddiqui

Submitted in partial fulfillment
of the requirements for
BSCS 624

UBIT – Department of Computer Science
University of Karachi

1. INTRODUCTION:

Tables are widely used for presenting structural and functional information. They are present in forms including newspapers, research articles, handwritten marksheets, images and scientific documents, etc. If we want to transfer these tables into an excel sheet to have the possibility to edit them, we need to type manually one by one. This would take a lot of time and effort. So, to overcome this problem, we will design this project as a mobile application which will take a picture using our camera or take a screenshot of any tabular data, then it extracts all the digital information and stacks them into an excel sheet so anyone can manipulate the data easily. This saves an ample of time and is less erroneous.

PROJECT OBJECTIVE:

The goal of this project is to design and implement a mobile application, which will take a picture using mobile camera or capture a screenshot of any sort of tabular data in jpg or png format and then detect the table in that picture, extract its information and convert it into an editable excel sheet, where each cell can be edited and used for further analysis.

USE CASES:

- **Scanning Documents to Phone:** We often capture images of important tables on the phone and save them, but with the table extraction technique, we can capture the images of the tables and store them directly into excel. With this, we need not search for images or copy the table content to any new files, instead, we can directly use the imported tables and start working on the extracted information.
- **Invoice Automation:** There are many small-scale and large-scale industries whose invoices are still generated in tabular formats. These do not provide properly secured tax statements. To overcome such hurdles, we can use table extraction to convert all invoices into an editable format and thereby, upgrade them to a newer version.

2. METHODOLOGY:

The algorithm consists of three parts:

- The first is the table detection and cell recognition with *Open CV*.
- The Second part is the extraction of information in each allocated cell through Optical Character Recognition (OCR) with *pytesseract*.
- The last phase includes converting the extracted information from tables to compiling them in excel.

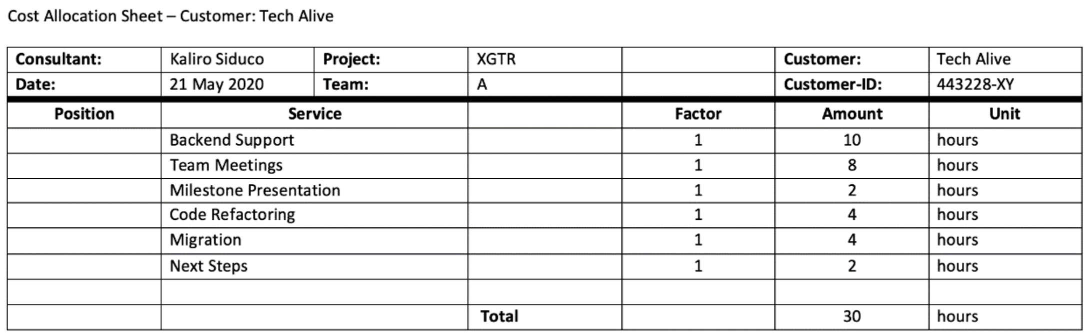
➢ TABLE DETECTION:

In this phase, we will identify where exactly the tables are present in the given input. We use different techniques and algorithms to detect the tables, either by lines or by coordinates. Clear and detectable lines are necessary for the proper identification of cells. In some cases, we might encounter tables with broken lines, gaps or no borders at all, where we need to opt for different methods. Then we will come to image transformation method.

Image Transformation:

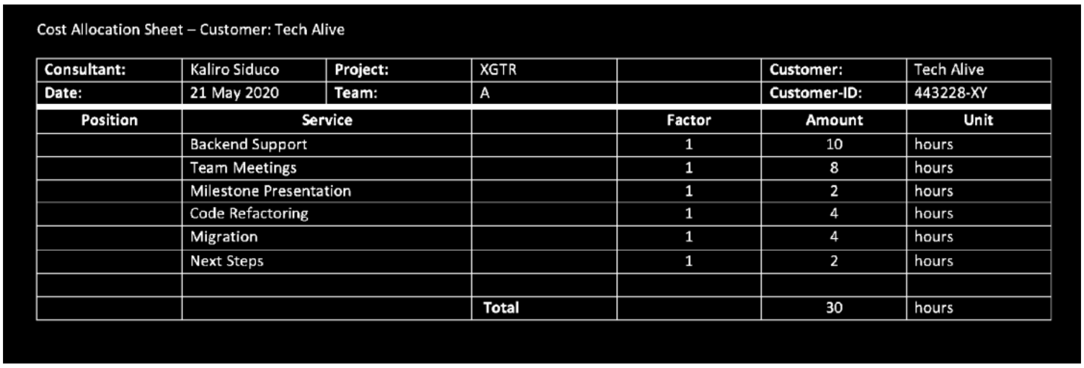
Image transformation is a primary pre-processing step in detecting tables. This includes enhancing the data and borders present in the table. The steps in image transformation are:

First, we apply thresholding to convert the input image to a binary image and inverting it to get a black background and white lines and fonts. This transformation step helps us to find the content more precisely.



Cost Allocation Sheet – Customer: Tech Alive

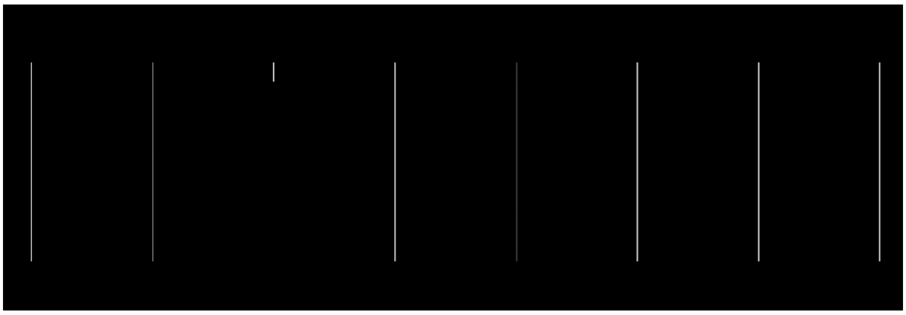
Consultant:	Kaliro Siduco	**Project:**	XGTR		**Customer:**	Tech Alive
Date:	21 May 2020	**Team:**	A		**Customer-ID:**	443228-XY
Position	**Service**			**Factor**	**Amount**	**Unit**
	Backend Support			1	10	hours
	Team Meetings			1	8	hours
	Milestone Presentation			1	2	hours
	Code Refactoring			1	4	hours
	Migration			1	4	hours
	Next Steps			1	2	hours
			Total		30	hours

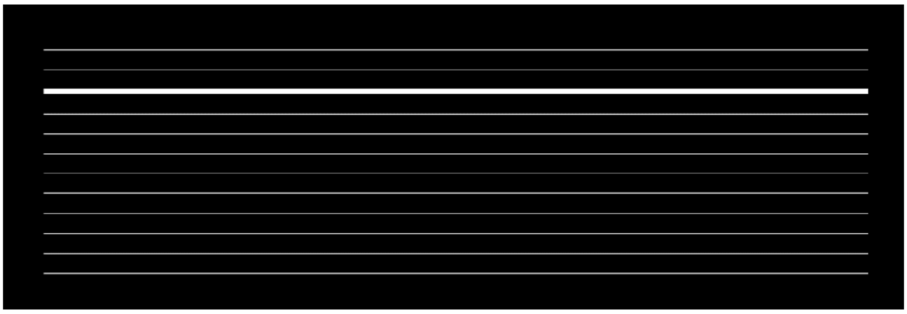


Cost Allocation Sheet – Customer: Tech Alive

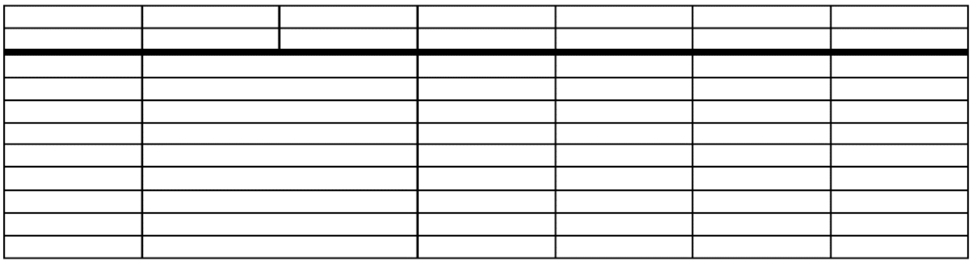
Consultant:	Kaliro Siduco	**Project:**	XGTR		**Customer:**	Tech Alive
Date:	21 May 2020	**Team:**	A		**Customer-ID:**	443228-XY
Position	**Service**			**Factor**	**Amount**	**Unit**
	Backend Support			1	10	hours
	Team Meetings			1	8	hours
	Milestone Presentation			1	2	hours
	Code Refactoring			1	4	hours
	Migration			1	4	hours
	Next Steps			1	2	hours
			Total		30	hours

Then noise is reduced using a median filter after thresholding process. Then we define a kernel to detect rectangular boxes, and followingly the tabular structure. First, we define the length of the kernel and following the vertical and horizontal kernels to detect later on all vertical lines and all horizontal lines.





We combine the horizontal and vertical lines to a third image, by weighting both with 0.5. The aim is to get a clear tabular structure to detect each cell.

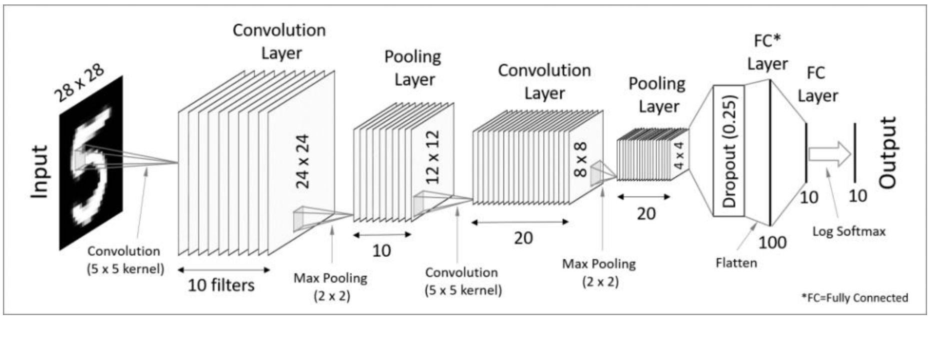


After having the tabular structure, we use a function to detect the contours. This helps us to retrieve the exact coordinates of each box.

➢ TABLE EXTRACTION:

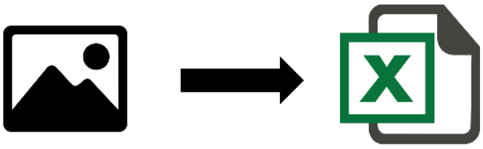
This is the phase where the information is extracted after the tables are identified. There are a lot of factors we need to focus regarding how the content is structured, recognition of handwriting style of an individual person, the content of the cells can either be numeric or textual. Handwritten digits recognition is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavours.
The neural networks are widely used for the recognition of characters. We will train and test a neural network classifier using MNIST database.

- MNIST dataset is probably one of the most popular datasets among machine learning and deep learning enthusiasts. it contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing. So, the MNIST dataset has 10 different classes. The handwritten digits images are represented as a 28×28 matrix where each cell contains grayscale pixel value.
- We need to localize each digit in the image and then use *digit recognizer* over the digits. In Image Processing, this task is known as localization. We will use OpenCV and neural network for recognizing each digit. A CNN model generally consists of convolutional and pooling layers.



➢ TABLE CONVERSION:

The last step is the conversion of the list to a data frame and storing it into an excel-file. We need to have a proper table layout to push the content in.



3. IMPLEMENTATION:

The Implementation phase will include the following aspects:

- **Software:** Python 3
- **Python Libraries:**
  - ✓ **For Building GUI:** Kivy (Python framework for mobile application development)
  - ✓ **For Table Detection & Information Extraction:** OpenCV-python, NumPy, sklearn, Keras, matplotlib, TensorFlow, pytesseract etc.