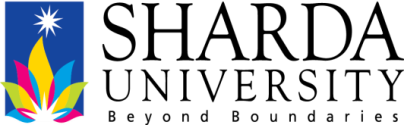
****

**PROJECT BASED LEARNING (PBL-4) LAB (CSP392)**

**Optical Mark Recognition using Computer Vision**

**B. TECH 3rdYEAR**

**SEMESTER: 6th**

**SESSION: 2022-2023**

**DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING**

**SHARDA UNIVERSITY, GREATER NOIDA**

**Submitted By:**

**Name:** Avishek Thakur Amitesh Kumar Sumeet Dhiman

**Roll No:** 200101079 200101051 200101323

**System ID:** 2020502389 2020568625 2020532155

**Section:**  E **Group:** G1

**Submitted To: Dr. Sandeep Kumar**

Table of Contents

[Project Title 3](#_Toc134468690)

[Technologies to be used 3](#_Toc134468691)

[Software Platform 3](#_Toc134468692)

[Hardware Platform 3](#_Toc134468693)

[Problem Statement 4](#_Toc134468694)

[Literature Survey 4](#_Toc134468695)

[Project Description 4](#_Toc134468696)

[Project Modules: Design/Algorithm 6](#_Toc134468697)

[1. Input Image 6](#_Toc134468698)

[Implementation Methodology 10](#_Toc134468699)

[Result & Conclusion 11](#_Toc134468700)

[Future Scope of the Project 11](#_Toc134468701)

[Advantages of this Project 11](#_Toc134468702)

[Outcome 12](#_Toc134468703)

[References 12](#_Toc134468704)

# Project Title

Optical Mark Recognition using Computer Vision.

This project focuses on providing a portable and convenient solution to the OMR sheet checking with the help of Computer Vision and proposing a web-based application made with python that allows you to upload OMR and get it evaluated accurately without the hassle of huge OMR scanners. We propose a software where you can post image of OMR sheet taken from different angles yet our application can detect the bubbles marked and evaluate the answers accordingly as per the answer scheme. It generates a *.csv* file as output with all the necessary fields such as roll no., question no., etc.

**Team / Group Formation:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Student Name** | **Roll Number** | **System ID** | **Role** |
| **1** | **Avishek Thakur** | **200101079** | **2020502389** | **Developer** |
| **2** | **Amitesh Kumar** | **200101051** | **2020568625** | **Tester** |
| **3** | **Sumeet Dhiman** | **200101323** | **2020532155** | **Developer** |

# Technologies to be used

## Software Platform

1. **Front-end**
   * Python 3
2. **OMR Checker**
   * Pyhton 3
   * Opencv
   * Rich – table generation
   * Numpy
   * Pandas
   * Matplotlib
   * Jsonschema
   * dotmerge

## 

## Hardware Platform

Camera for Capturing Images

# Problem Statement

In today’s world most of the competitive exams are based on MCQ (Multiple Choice Questions). The responses of these MCQ based exams are recorded in the Optical Mark Reader (OMR) sheet. Evaluation of the OMR sheet requires separate specialized machines for scanning and marking. The sheets used by these machines are special and costs more than a normal sheet.

OMR (Optical Mark Reader or Optical Mark Recognition) is the process of gathering information from human beings by recognizing marks on a document. OMR is accomplished by using a hardware device (scanner) that detects a reflection or limited light transmittance on or through a piece of paper. OMR analyzing is process of automatically analyzing human marked sheets used to record answers and interpreting their results. Candidates filling their OMR sheets using pencil or Ball-Point Pens. These hardware device lacks in portability and are expensive to use. Thus, limits the usage of OMRs widely which end up in lack of opportunities or say, we are unable to make the best out of this technology.

# Literature Survey

# Project Description

We propose a full-fledged OMR checking software that can read and evaluate OMR sheets scanned at any angle and having any color.

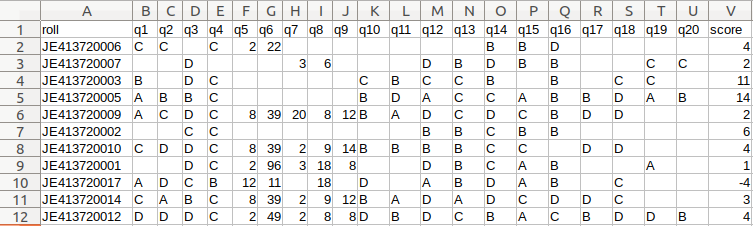
|  |  |
| --- | --- |
| **Specs** |  |
| **Accuracy** | Currently nearly 94% accurate on good quality document scans; and about 90% accurate on mobile images. |
| **Robustness** | Supports low resolution, xeroxed sheets. Minimum resolution **640x480** |
| **Fast** | Current processing speed without any optimization is 1**00 OMRs/minute.** |
| **Customizable** | Easily apply to custom OMR layouts, surveys, etc. |
| **Visually Rich** | Get insights to configure and debug easily. |
| **Lightweight** | Very minimal core code size. |
| **Large Scale** | Tested on a large scale. |

Once we capture and configure the OMR layout and image, we just need to throw images of the sheets at the software and you'll get back the marked responses in an excel sheet.

Images can be taken from various angles as shown below

*Figure 1: Input Images captured at various angles*

Output generated would be of the form of a .csv file which can be later on taken into various applications such as data analysis, surveys, etc.



*Figure 2: Screenshot of Output .csv file generated from OMR sheet*

## Project Modules: Design/Algorithm

## 1. Input Image

To read an image, we simply call the imread function of the cv2 module. This will return an image as a NumPy array. We can confirm this by calling the type function and passing as input the object returned by the imread. function.

“cv2.imshow('test.png',image)”.

**2. Detect the Image:**

We tested the code on the image of an OMR sheet. Multiple scanned images of the OMR sheet were taken for testing. The image on which the approach was tested. We also test our code on different inclinations. Current code works on OMR sheets with circles, it will not work with OMR sheets of other shapes. Though a variant of our code will work successfully on it, where the first step will be replaced with the respective transformation for detecting the new shapes.



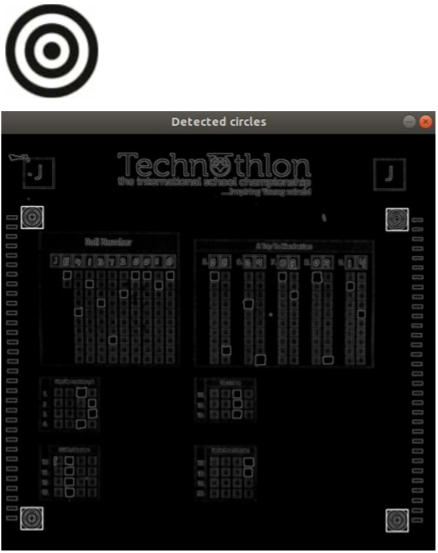
*Figure 3: Finding the Region of Interest (ROI) i.e., paper in the image*

**3. Extract the possible answer choices**

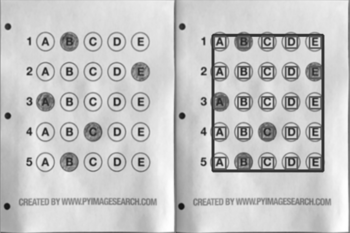
In order to detect the images of bubbles, we loop over each of the individual contours. Considering the aspect ratio of the contours we recognize a contour to be a bubble.Now, pre-process our input image as:

gray=cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY) blurred=cv2.GaussianBlur(gray,(5,5),0)

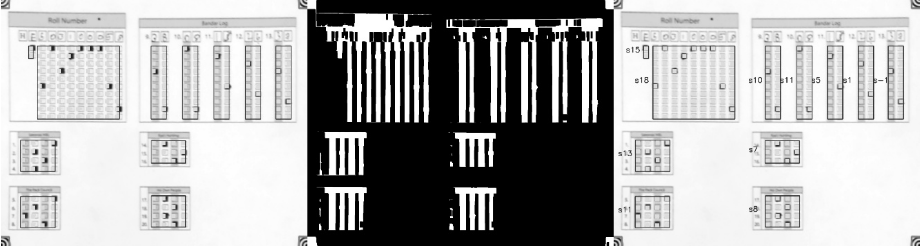
edged = cv2.Canny(blurred, 75, 200)



*Figure 4: Detecting Markers on Image in order to align template layout in next step*



*Figure 5.1: Template layout setting*



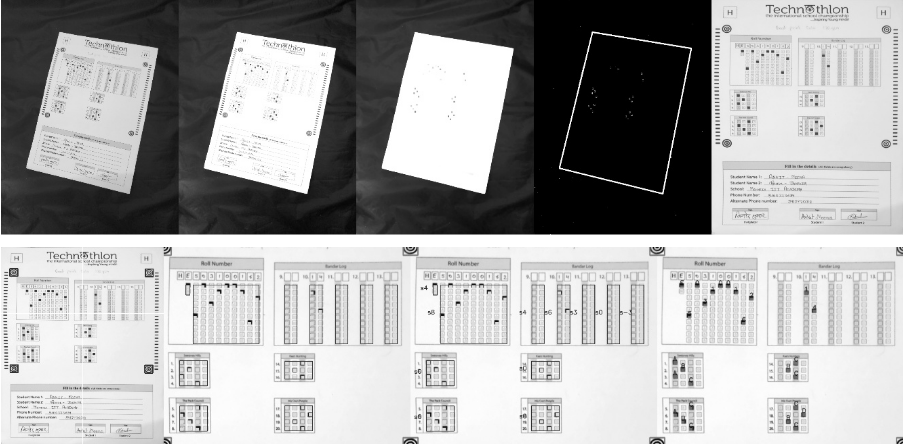
*Figure 5.2: Template Layout Alignment*

**4. Parallel Line Interpolation:**

When interpolating images, the image itself plays the role of an interpolated function, it means that if we interpolate four parallel lines passing through detected circles centres, it will pass through missing circles too. Based on the idea above, we come up with an error minimization method to find the four parallel lines passing through detected circles. Then we get the value which is the color of the pixel. Image pixels are Points where the value of the function (pixel colour) is known. Interpolation contains intermediate values of the function (Gives equation of lines), the image does not necessarily increase, the interpolation methods allow too arbitrarily

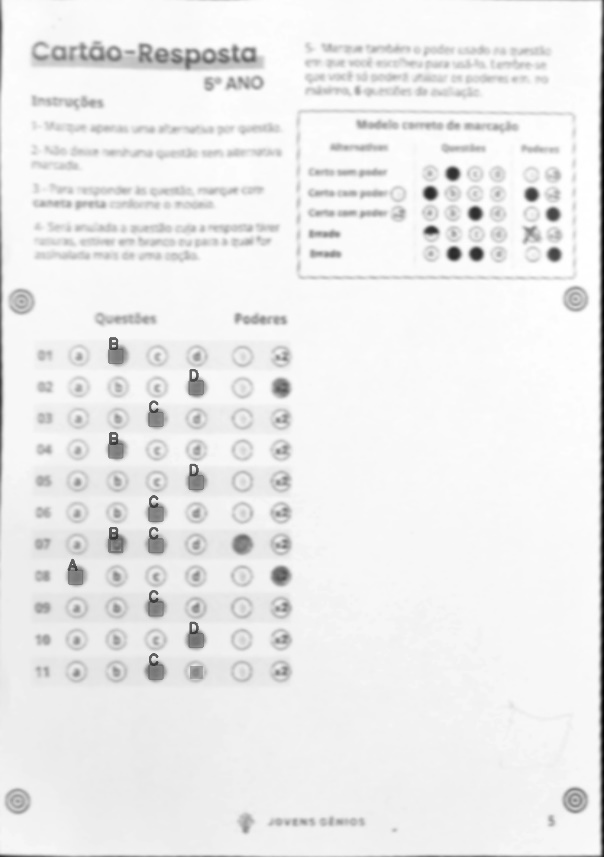
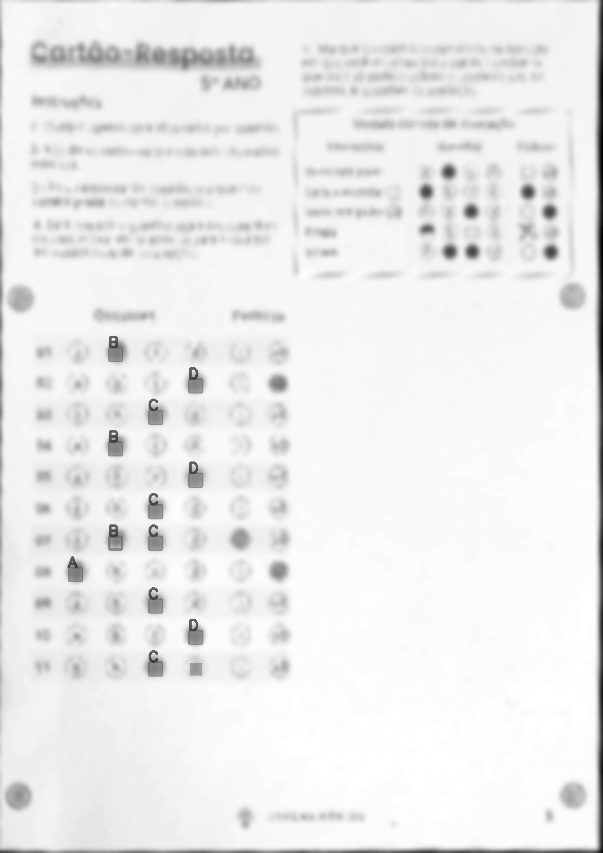
changing the image size and aspect ratio.

**Processing steps on various set of data items**

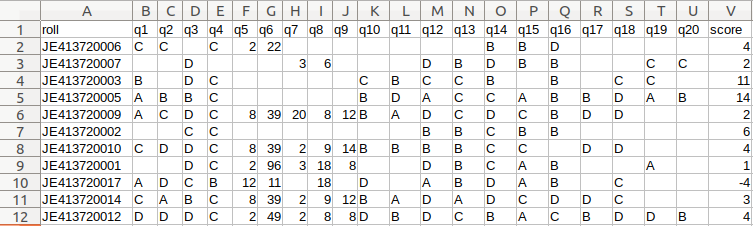


**Output module**

Output module has the main task to generate an image of OMR with the bubbles and their respective interpretation merged over image as shown in figure after the correction as well as generate a csv files.

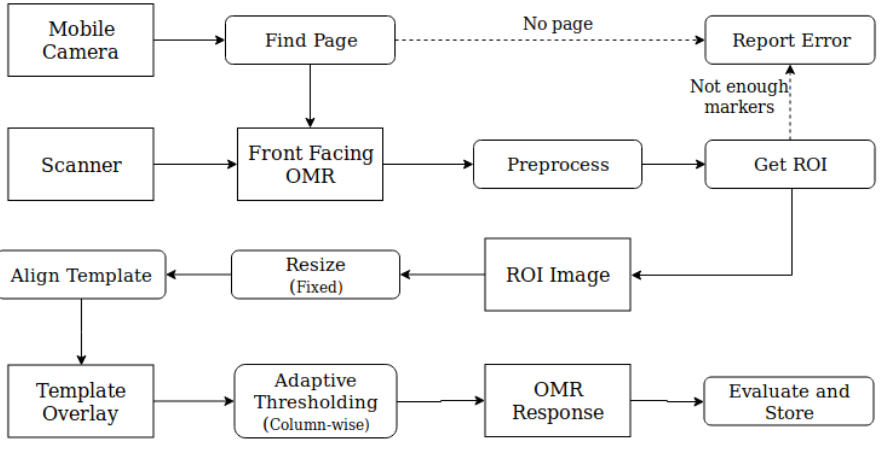


*Figure: Output OMR generated after checking*

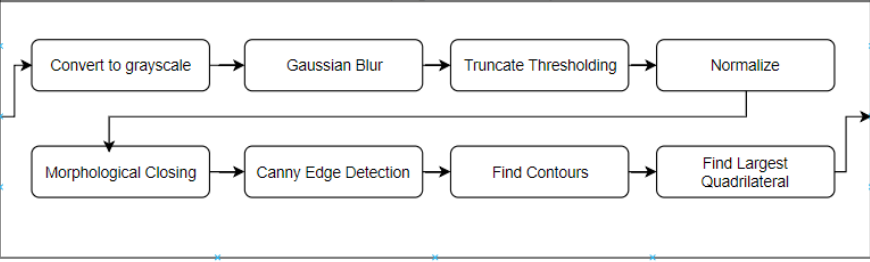


*Figure: .csv file generated as output*

# Implementation Methodology



*Figure: Block Diagram of the OMR Checker*



*Figure: Block Diagram of Find Page Algorithm*

# Result & Conclusion

Our Proposed model is Currently nearly 94% accurate on good quality document scans and about 90% accurate on mobile images.

OMR scanners are faster and more accurate than OCR scanners for processing OMR sheets, and that they can provide timely feedback to students. OMR scanning systems have been successfully implemented in various academic settings, with studies reporting high levels of accuracy and efficiency. These systems have the potential to save time and reduce the risk of errors compared to manual data entry or scanning. Additionally, OMR software that allows users to upload images directly without a scanner can offer convenience, flexibility, cost savings, improved accuracy, time savings, and data security. Overall, OMR checkers offer a reliable and efficient solution for grading multiple choice exams and evaluating student performance in large classes.

# Future Scope of the Project

* The integration of artificial intelligence (AI) with OMR software can significantly improve the accuracy and speed of processing OMR sheets. With AI, the software can learn from past data and optimize the processing of future sheets.
* Cloud-based OMR solutions can offer more flexibility, accessibility, and scalability. Users can access the software from anywhere and scale up or down as per their requirements.
* With the increasing use of smartphones and tablets, mobile-based OMR applications can be developed. Users can scan the OMR sheets using their mobile devices and process the data in real-time.
* With globalization, there is a growing need for OMR software that supports multiple languages. Multi-lingual support can help to process OMR sheets in different languages without any errors.
* OMR software can be integrated with other software such as data analytics tools, reporting tools, and data visualization tools. This integration can provide more insights into the data and help in making better decisions.
* OMR software can be used for advanced data analytics to gain insights into the data. With the help of machine learning algorithms, the software can detect patterns, trends, and anomalies in the data.

# Advantages of this Project

This project can offer several advantages, including convenience, flexibility, cost savings, improved accuracy, time savings, and data security.

* **Portable:** This project provides portability as it eliminates the need for huge hardware device such as scanners, one can use mobile application to get the OMR evaluated.
* **Convenience:** Users can easily upload images of OMR sheets without the need for a physical scanner. This can save time and effort, especially when working with large volumes of OMR sheets.
* **Flexibility:** Users can upload images from any location, as long as they have an internet connection. This can be particularly useful for remote work or when scanning OMR sheets from different locations.
* **Cost savings:** By eliminating the need for a physical scanner, users can save on the costs associated with purchasing, maintaining, and repairing scanners.
* **Improved accuracy:** OMR software that allows users to upload images directly can offer improved accuracy compared to manual data entry or scanning. This is because the software can automatically detect and interpret the marks on the OMR sheets with a high degree of accuracy.
* **Time savings:** By automating the scanning and processing of OMR sheets, users can save time and reduce the risk of errors. This can be particularly useful when processing large volumes of data.
* **Data security:** Uploading images directly to the OMR software can offer improved data security compared to physical scanners. This is because there is no risk of physical documents being lost, stolen, or damaged.

# Outcome

* Research Paper Publication
* Project to Product Development

# References