Optical Mark Recognition (OMR) MCQ Automated Grading Model

Submitted To

Mr.Rohan Verma

(Project Guide)

Submitted BY

Kartikeya(18134503008)

Shivam (18134501029)
Priyanka Chandra(18134501006)

B.Tech.CSE(8th sem)

Department of Computer Science and Engineering

School of Engineering and Technology

Contents

- Introduction
- Overview of Python Package to be used
- Software & Hardware requirement
- Model Workflow
- Features of OMR MCQ automated grading model
- Working of OMR MCQ automated grading model
- Advantages of OMR MCQ automated grading model
- Conclusion
- Future scope
- References

Introduction

What is OMR

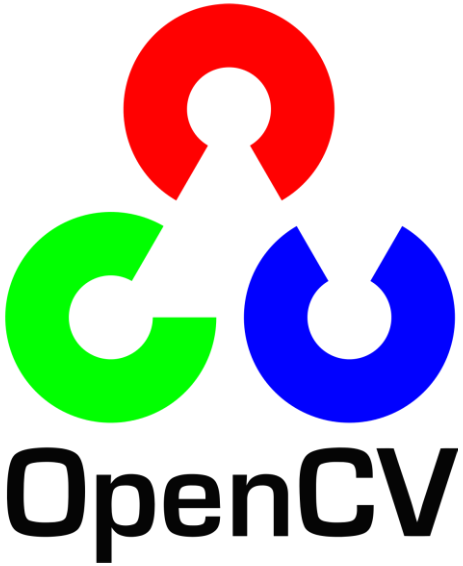
- The “OMR” stands for Optical Mark Recognition. Optical Mark Recognition (OMR) is a popular and highly accurate recognition technology for detecting specific "marks" on an image and utilizing those markings as a reference point to extract other regions of interest (ROI) on the page Optical Mark Recognition technology is used for collecting data from “fill-in-the-bubble” types of questions on student tests, surveys, ballots, assessments, evaluations, and many other types of forms.. Optical Mark Recognition enables the respondent to select an answer to a question by filling in a “bubble” or “mark” associated with an answer choice.
- The OMR response sheet is first scanned, and the image of the answer sheet is then entered into the software system. Image processing assists us in discovering solutions to all of our queries by locating the region of interest.
- The OMR technology has evolved dramatically in recent years. OMR technology is used everywhere, including in schools, universities, and classrooms. Exams currently employ an OMR answer sheet checking system because it makes exam administration simple, powerful, and inexpensive.

What is Optical Mark Recognition (OMR) MCQ Automated Grading Model

- Optical mark recognition (OMR) MCQ Automated Grading Model is a Python and Open CV-based Artificial Intelligence model that can operate on any Python. Optical mark recognition (OMR) MCQ Automated Grading Model will load the image from the dataset or captured image from webcam and use this answered omr sheet as input, processes the answered omr sheet and apply algorithms on the omr sheet. Model considered the dark shaded(filled bubbles) area as answer compare the list of shaded area(filled bubbles) with the referenced list of answers which are pre-defined in the model and give the outputs as well as grades the omr sheet.
- Some people do not fill in the dark colors, or they fill in two hues in one column this type of answered omr sheet will not scan properly by using manual omr machine and it will be a lengthy process but we have advantage in this model to scan the image again and again until we get the correct output because it consumes the less time as compare to OMR scanning machine. The model is easy to use it does not require any particular training to operate the model, and it will be highly cost-effective in future.

Overview of python package to be used

- OPEN CV
- Numpy



Source-opencv.org

Source-numpy.org

Software & Hardware requirements

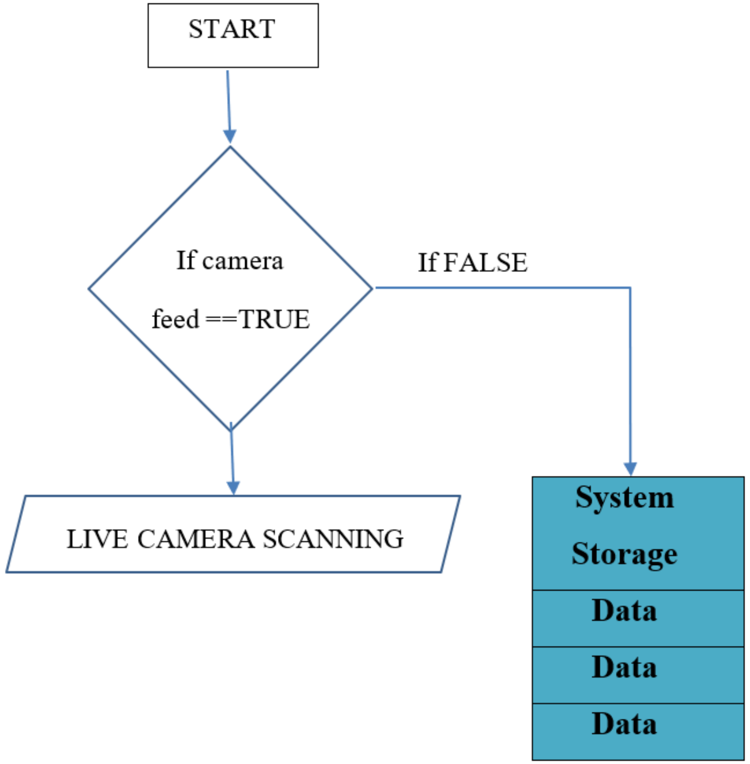
Software Requirements

- Anaconda
- Python 3.8
- Open CV
- Windows 10 or higher versions
- VS Code

Hardware Requirements

- Intel i5 or higher versions
- 8GB Ram
- Webcam
- 5GB storage HDD or SSD

Model Workflow



Flowchart Workflow

Features of OMR MCQ Automated Grading Model

Loading the data:

- Loading data means fetching required data from particular location and from input devices.In this project model will load the data that is an image which is from the given path of dataset and capture from webcam

Data Preprocessing

- Data preprocessing is the process of preparing raw data for use by a Artificial Intelligence model. It is the first and most important stage in developing any Artificial Intelligence model.
- Resize: Resize the input image of the omr sheet define the height and with as per the requirement.

Data Processing:

- In the data processing the data is transform into the useable information. In we enhance the color quality of the pixels of the image and apply some required operation on the data.

Grey Scaling:

- In this image will convert into the grey in color by applying grey scaling the BGR2grey color is used in this model

Edge Detection:

- After applying Gaussian blur to blur the image The technique is used in order to define the borders of the objects on the image named as Canny Edge Detector. This is used to find the corners, this method was developed by J. Canny in 1986. Canny Edge Detector is one of the most commonly used image processing devices that detects the edges very precisely. It is considered to be as the standard edge detecting method in industry. Canny accepted the edge detecting problem as a problem of signal processing optimization, thus developed it in order to optimize an objective function.

Contours:

- Contour is used to contoured the image the highlight the border of image what the imager is contain.

Corner Points:

- After applying the contour we find the corner point of the required region of the image. The rectangles.

Threshold:

- Threshold is used to darken the filled region the omr sheet we use value the 170, 255.

Warp Perspective

- We used the the Word IPU for the warp perspective for the better insights that we can get from the given information of the image.

Answer Shorting

- To sort the answer we model compare the input list of the referenced answer with the answers which we get from the processed image.

Grading:

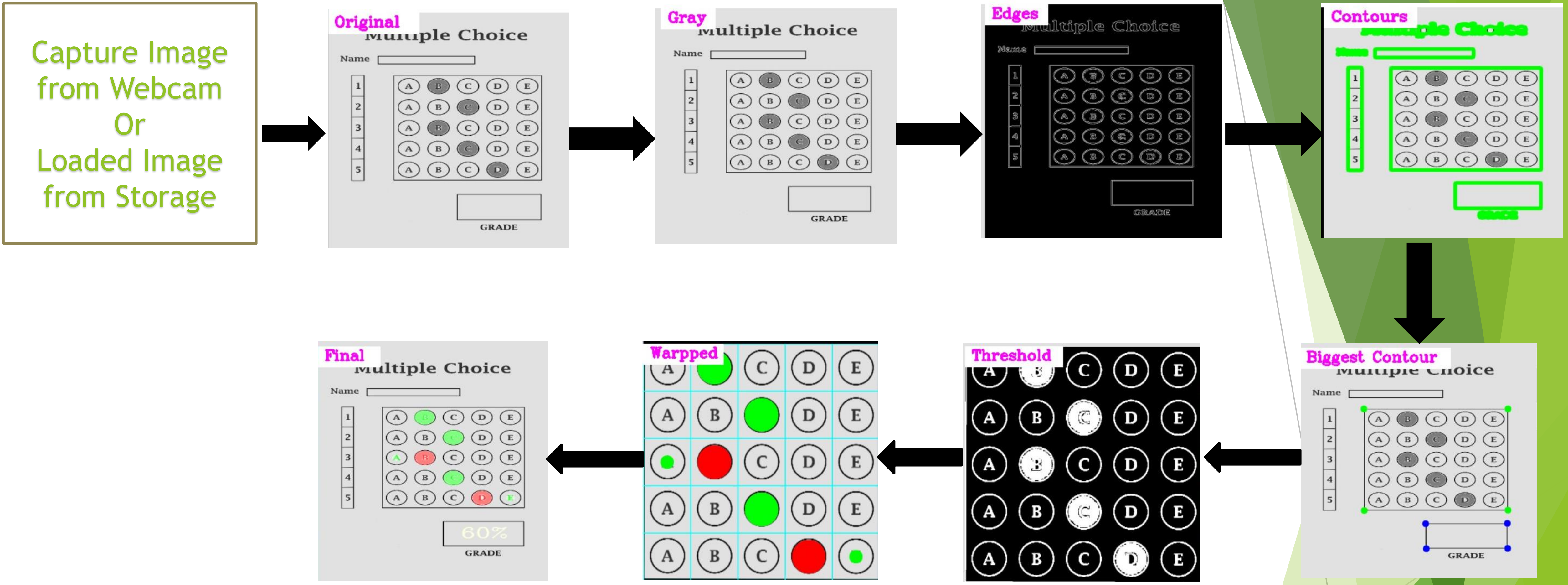
- After answer shorting we apply some which are mentioned in utlis file to get the final output that is graded answer omr sheet.

Output Saving:

- Save Image by entering the ‘s’ key on keyboard the will saved.

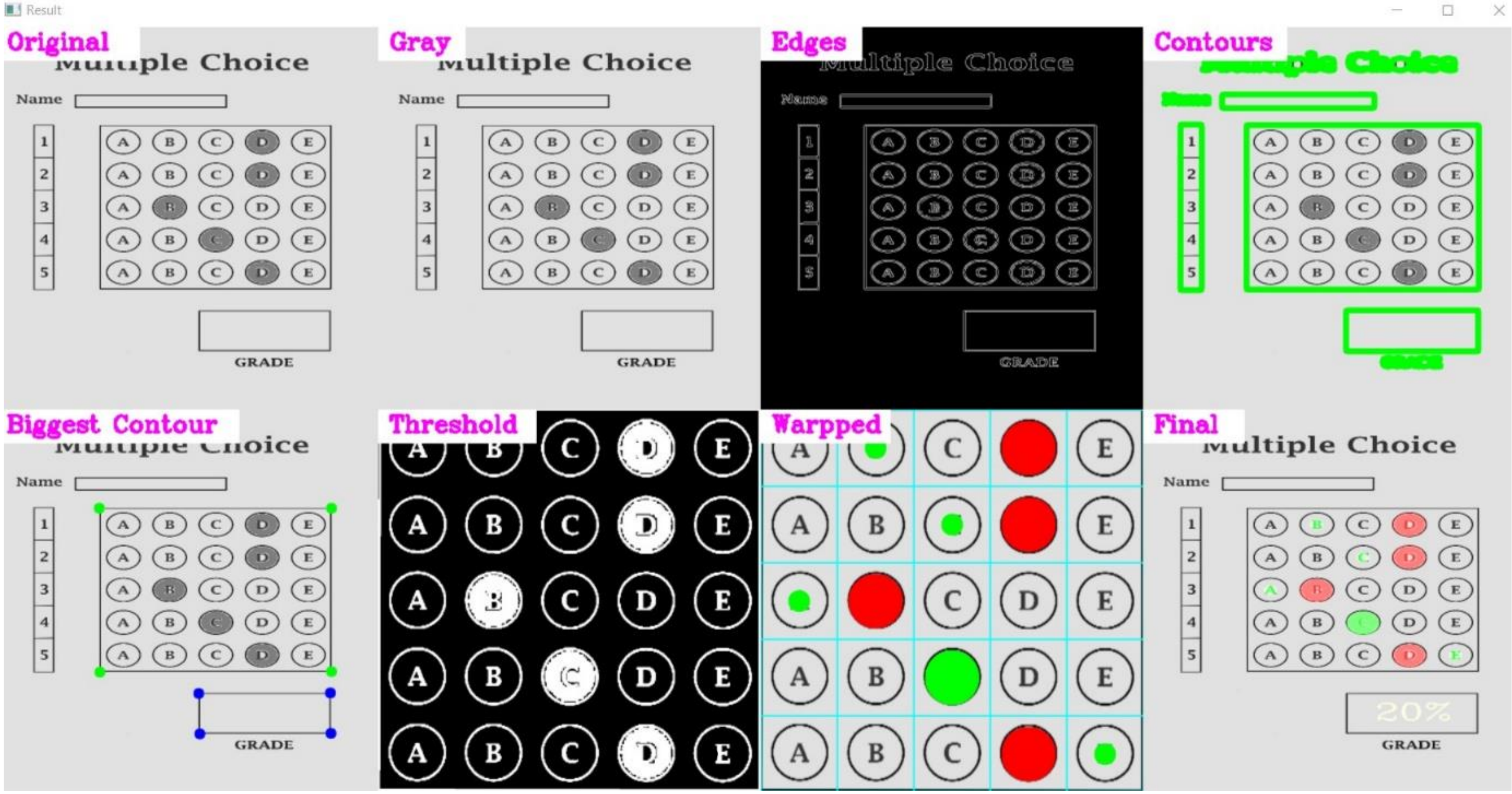
Working of OMR MCQ Automated Grading Model

- **Step 1:** Here it first capture the video/image from the webcam and or load image from the dataset from the given path.
- **Step 2:** Then Model will convert the captured or loaded image in grey color by using grey scale.
- **Step 3:** Model will find the edges from the grey scaled image using canny.
- **Step 4:** Model will apply the contours over the grey scaled image.
- **Step 5:** Model will find the biggest rectangles present in the image and their corner points
- **Step 6:** Then it will use wrap IPU which is the wrap perspective on the image from this we get the required portion of the image
- **Step 7:** Later model will apply some threshold on the image for the better visualization of the dark shaded bubbles.
- **Step 8:** Model will find the filled bubbles and compare the list of input with the reference list which we defined in the model. And generate the output.
- **Step 9:** lastly, we need to enter the “s” key on the keyboard to save the output.

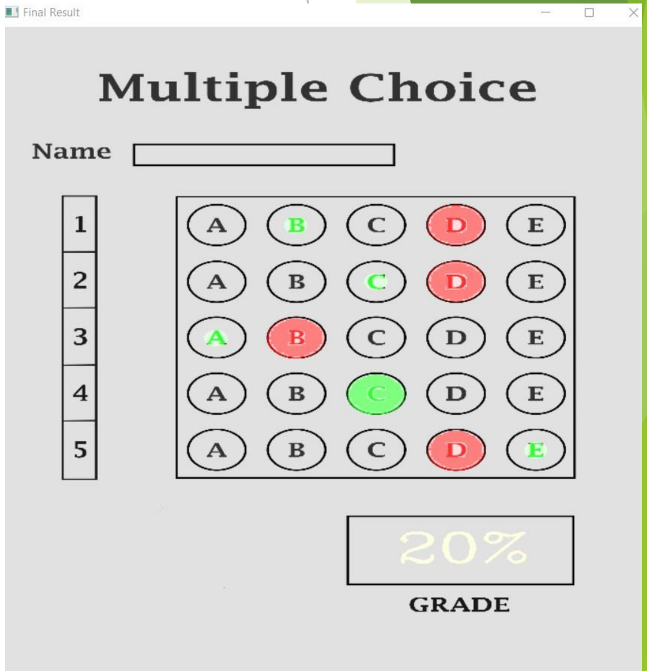


Capture Image from Webcam Or Loaded Image from Storage

Process OMR Sheet



Stacked Output



Final Output

Split Output

Advantages of OMR MCQ Automated Grading Model

- OMR MCQ automated grading model is very easy to use and handy as compared OMR scanning machine
- We must insert our omr sheet into the OMR machine, which then scans it. It also requires software to link the omr machine to the system in order to provide the output (scanned image of the OMR sheet and grades) of the OMR sheets. In this model we only need to load image from dataset and we can also use captured image from webcam rest of the process will done by the model like loading image, edge detection, contours etc
- OMR scanning models, on the other hand, scan the image and grade omr sheet as output.
- An OMR MCQ automated grading model is cheaper than omr machine.
- In some cases, the omr machine may skip bubbles in the OMR sheet, but in the omr mcq automated graded model, we need a data set or a live image of the OMR sheet, and it will convert it to grey and then contour the filled bubbles as there are present correct answer bubbles, and it compares contour bubbles with the pre-setup correct answers and gives the grades because there are fewer chances of mistakes in the omr mcq automated grading model.

Conclusion

This work presents a system for Optical Mark Recognition developed for multiple choice tests with the programming language Python. OMR scanners were traditionally hardware-focused, but we created one using an underlying model that simply required a collection of photos or live image from webcam. Scanned photos that have been rotated by some angle are also accepted, therefore the need for hardware for perfect alignment is not necessary here. A user interface is also supplied so that persons with less expertise may utilize it as well. The OMR sheet will be in front of the user, and if the picture is scanned incorrectly, it can be skipped and scanned again. Sometimes applicants may not fill out their information correctly or neglect to indicate certain important elements, which are then appropriately recorded in the database. As it is easy to use, the software can be easily used by teachers or school managers as well. Hence, fast and effective model like this will not only enable the personnel to save time but also it will be very cheap. Furthermore, students will be able to learn the results earlier.

Future Scope

- Currently we give the input image from the from webcam or from dataset model processed the input and apply algorithm on the image and we get the graded omr sheet as output.

- In the future, we may give user authentication to the system so that we have data on who has scanned the photographs and can reach out to the concerned person if something has transpired illegally. This allows us to restrict system access to users who have been educated in this program.

References

- Jingyi, T., Hooi, Y. K., & Bin, O. K. (2021, July). Image Processing for Enhanced OMR Answer Matching Precision. In *2021 International Conference on Computer & Information Sciences (ICCOINS)* (pp. 322-327). IEEE.
- Martinez, M. E., Ferris, J. J., Kraft, W., & Manning, W. H. (1992). Automated scoring of paper-and-pencil figural responses. *Journal of Educational Technology Systems*, *20*(4), 251-260.
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362.
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

Thank You