OPTICAL MARK RECOGNITION (OMR) MCQ Automated Grading

Synopsis

Synopsis submitted in partial fulfilment of

Bachelor of Technology

in

Computer Science and Engineering Technology

at

Department of Computer Science and Engineering Technology
Hemvati Nandan Bahuguna Garhwal University

By Kartikeya, Priyanka Chandra

And

Shivam

Department of Computer Science and Engineering Technology
Hemvati Nandan Bahuguna Garhwal University

Under the guidance of

Mr. Rohan Verma
Computer science and Engineering
Hemvati Nandan Bahuguna
Garhwal University

Department of Computer Science and Engineering Technology
Hemvati Nandan Bahuguna Garhwal University

Certificate

I hereby certify that the work which is being presented in the synopsis entitled **OPTICAL MARK RECOGNITION (OMR) MCQ Automated Grading** By **Kartikeya, Shivam, Priyanka Chandra** in partial fulfilment of requirements for the award of degree of B.Tech. (Computer Science & Engineering)submitted in the Department of Computer Science & Engineering Hemvati Nandan Bahuguna Garhwal University under the Guidance of **Mr. Rohan Verma**. The matter presented in this synopsis has not been submitted by me in any other University / Institute for the award of B.Tech Degree.

Signature of the Student
Kartikeya, Shivam, Priyanka Chandra

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge

Signature of the Guide(s)

Department of Computer Science and Engineering Technology
Hemvati Nandan Bahuguna Garhwal University

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor **Mr. Rohan Verma** for enlightening me the first glance of research, and for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time ofresearch and writing of this synopsis. I could not have imagined having a better advisor andmentor for my project work.

Special thanks to Dean **Prof. Y.P. Raiwani** and technical support by **Mr. Rohan Verma,** is thankfully acknowledged.

Finally, I feel great reverence for all my family members and the Almighty, for their blessings and for being a constant source of encouragement.

Kartikeya, Shivam, Priyanka Chandra

Index

Synopsis

OPTICAL MARK RECOGNITION (OMR) MCQ Automated Grading

1. Abstract

The traditional OMR machine read OMR sheet . Optical mark recognition (also called optical mark reading and OMR) is the process of reading information that people mark on surveys, tests and other paper documents.OMR is used to read questionnaires, multiple choice examination papers in the form of shaded areas.

To overcome this problem, we design and create an model that allows every teacher or person to scan the OMR sheet & get all the information about marks,student name.

To reduce the risk of wrong marks tampering, we focused on shaded areas and used OPEN CV to verify true shaded marks.

In this synopsis, we design a scanning model, classifying and supplying the information and produce correct marks & also grades to teachers or person who use this model.We also evaluate the performance of this model by analysing the OMR sheets recognition process.

2. Introduction

- What is OMR

Optical Mark Recognition is used for recognizing certain “marks” on an image and using those marks as a point of reference to extract other regions of interest (ROI) on the page.

- What is OPTICAL MARK RECOGNITION (OMR) MCQ Automated Grading

Optical mark recognization (OMR) MCQ Automated Grading is a AI model which is Python & Open CV based which can run on any python compiler. By using this model we can scan omr sheet and grade them. It takes as input an image of an answered answer sheet process the sheet and give outputs which alternatives were marked and also grade the omr sheet

Checking OMR sheet without OMR machine is very difficult for anyone.Some person doesn’t fill dark shades,or they fill double shades in one column.& this is very time-consuming process. The following is the standard method for checking OMR sheet & gives grades:

- Check OMR sheet by OMR machine.
- Give manually grades too see the OMR sheet.

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation.Wrappers in several programming languages have been developed to encourage adoption by a wider audience. In version 3.4, JavaScript bindings for a selected subset of OpenCV functions was released as OpenCV.js, to be used for web platforms.

If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

A CUDA-based GPU interface has been in progress since September 2010.

An OpenCL-based GPU interface has been in progress since October 2012, documentation for version 2.4.13.3 can be found at **docs.opencv.org**.

OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo,BlackBerry 10. The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

3.Problem of Statement:

Many researchers attempt to develop and construct a technology supported solution forland registration system, and they also proposed different approaches for designing land registration. From the deep literature review the following facts have been identified that.

- OMR scanning model is very easy to use and handy as compare OMR scanning machine.
- In OMR machine we have to put our omr sheet then it scans it. It also needs a software which connect omr machine to the system to give the output (scanned image of omr sheet and grades) of the omr sheets on the other hand OMR scanning model scan the image and give you output.
- OMR scanning model is cheaper then omr machine.
- There are some cases where omr machine may skips bubbles in omr sheet but omr scanning model we need data set or live image of omr sheet the it will convert it in grey and then contour the fill bubbles as there are preset up of correct answer bubbles it compares contour bubbles with pre setup correct answers and

give the grades as there are less chances of mistakes in omr scanning model.

4.Objective of the study

- Identify the conventional methods and issue to acquire data about any omr sheet
- Checking correct dark shades of OMR sheet using open cv
- Test performance analysis of all dark shades in every single column factor for feasibility of model.
- Design fast and quick model for teachers or any person.

3. Proposed Methodology:

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation.Wrappers in several programming languages have been developed to encourage adoption by a wider audience. In version 3.4, JavaScript bindings for a selected subset of OpenCV functions was released as OpenCV.js, to be used for web platforms.OPEN CV uses webcam or camera for scanning for reading the OMR sheet & give grades automatically.

This application consists of two main modules, which are explained below.

OMR scan module: Open CV scan dark shades using previous trained data in this module

Give grades module: Whenever all dark shades read by the model then webcam show the how much grades in that OMR sheet.

OMR Scan Process:

In fig1 this process, user run this model. User train this model using dataset of images.When the model is trained , the model is ready to run the process the OMR sheets.In this process we use open cv to scan the dark shades of OMR sheets.

Give grades Process:

In fig 2 this process,User already trained the model, User put own OMR sheet to sacn the dark shades. Then this model scan all dark shades & gives proper Garde of that OMR sheet.

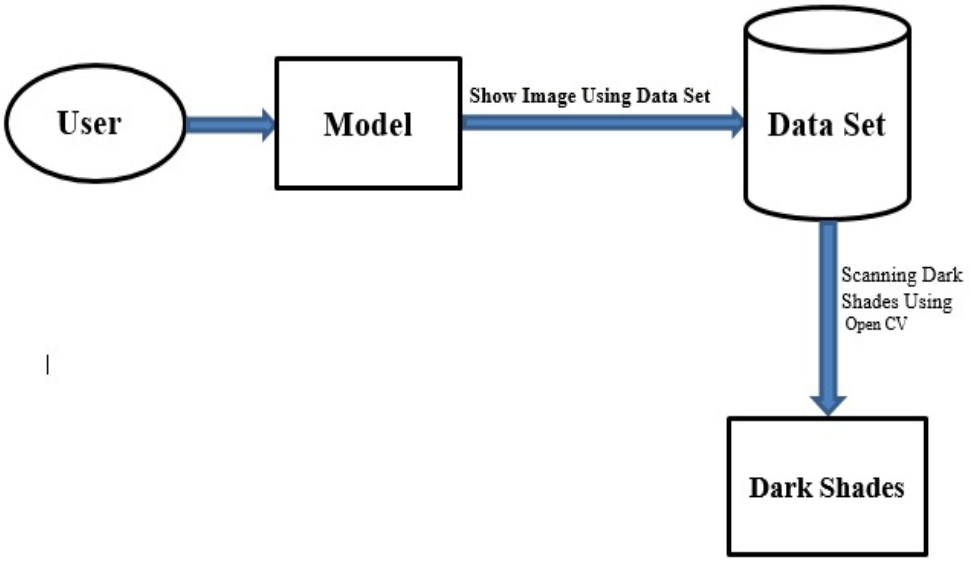


Fig. 1. OMR scan Architecture

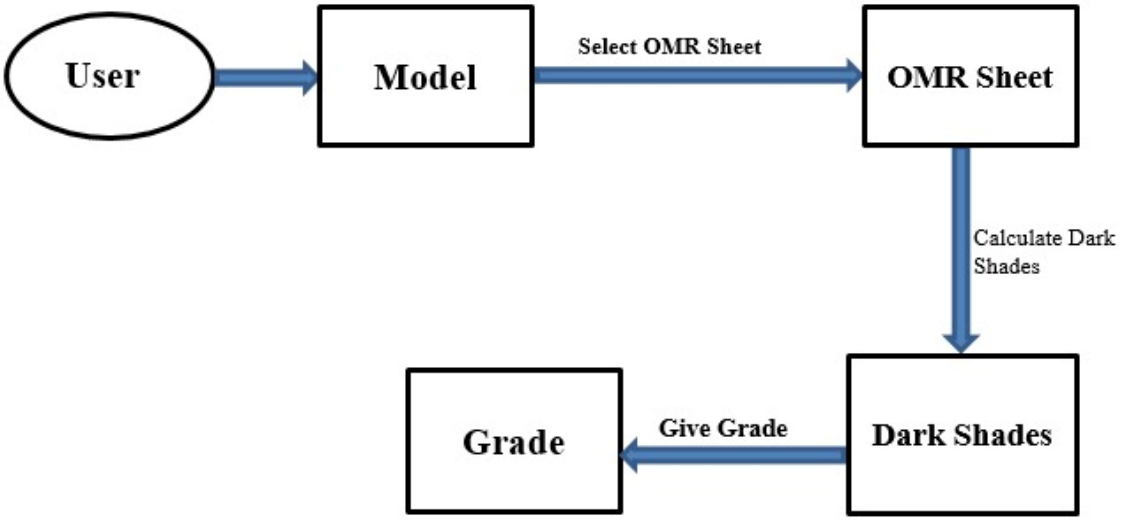
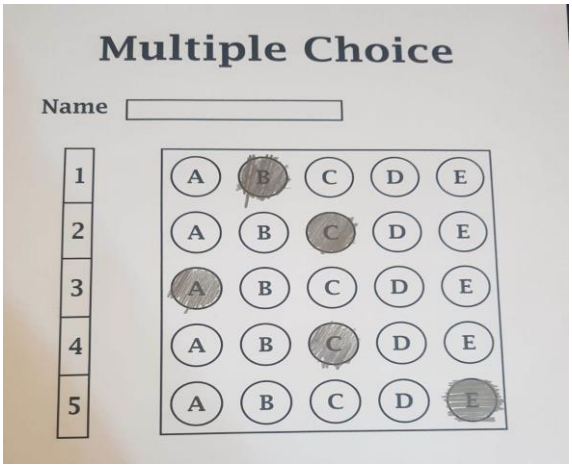


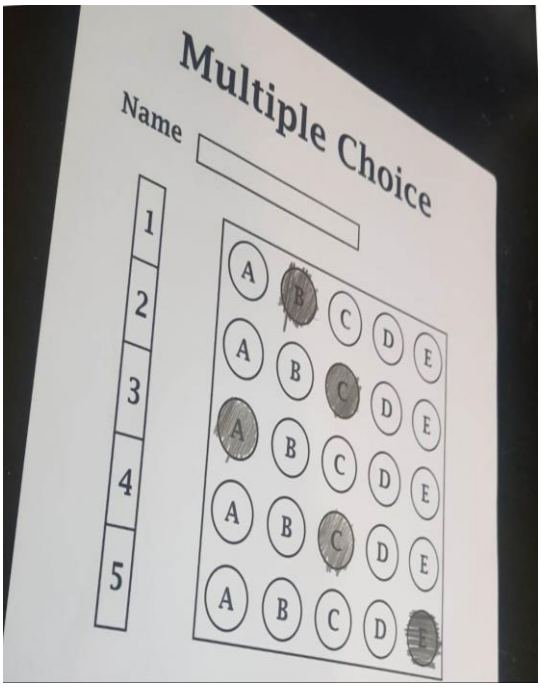
Fig. 2. OMR scan Architecture

Methods:

- **Step 1:** First we take an original image or image from dataset.
- **Step 2:** convert the taken image in grey scale.
- **Step 3:** Find the edges in grey scaled image.
- **Step 4:** Find the contour present in grey scaled image.
- **Step 5:** find the biggest rectangle present in the image and their corner points
- **Step 6:** Then we take wrap IPU which is wrap perspective and later apply some threshold on the image.
- **Step7:** We find the marks where each correct mark are present.
- **Step 8:** lastly, we save the output

Result & Analysis-





Conclusions:

OMR scanner used to be hardware centered but here we implemented it using al model and it requires only a set of images. Scanned images rotated by some angle are also accepted, so hardware used for proper alignment is not required here The user interface is also provided so that people having less knowledge can also use this.

OMR sheet will be in front of the user and so if sometimes the image is scanned too badly by mistake it can be skipped and can be scanned again. Sometimes candidates don't fill in their details properly or forget to mark some necessary details so that's also will be marked properly in the database

References:

- **https://www.techopedia.com/definition/8670/optical-mark-recognition-omr**
- **https://opencv.org**
- **https://github.com/posquit0/Awesome-CV**
- **https://en.wikipedia.org/wiki/NumPy**
- **https://numpy.org/**