Analysis of OMR Sheet Using Machine Learning Model

[1] Harendra Kumar, [2] Himanshu Chauhan, [3] Sonam Mittal
[1, 2]Dept. of Computer Science, B K Birla Institute of Engineering & Technology, Pilani,
[3]Associate Professor, Dept. of Computer Science, B K Birla Institute of Engineering & Technology, Pilani
[1] dhairya.bhardwaj786@gmail.com, [2] hchauhan730@gmail.com, [3] sonam.mittal@bkbiet.ac.in

Abstract- 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. In this paper, we intend to implement a real-time automatically analyzing human marked OMR (Optical Mark Recognition) sheet and interpreting results using OpenCV. There are two ways by which we can analyze OMR is through Live Stream or Photos. In our Model we used both methods to interpret results. OpenCV is a machine learning software library which includes a library of programm ing functions used in real time computervision.

Keywords: OpenCV, OMR, Real Time Computer Vision, Machine Learning.

I. INTRODUCTION

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 Penas

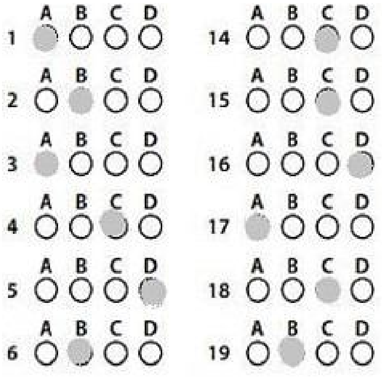


Fig 1: OMR sheet recorded answers using Pencil

Optical Mark Readers reads pencil or pen marks made in pre-defined positions on paper forms as responses to questions or tick list prompts. The OMR data entry system contains the information to convert the presence or absence of marks into a computer data file then compare it to the correct marked data which is already stored in the system then evaluates the result and show it to their respective candidate. Currently the organizations have to use dedicated scanner, special OMR software (like Remark Office OMR, Magenta Suite, Vision OMR, Scantron and many more)[6]. There are small organizations, institutes which are not able To pay for these resources. OMR scanner evaluates the sheet by evaluating MCQ one by one that will take more time.To overcome from these problems, we developed a software in which we just upload an image or show the OMR sheet on webcam as Livestream to our system then it willautomatically evaluate the result and show it in at that time in real time Machine Learning and ML module OpenCV. The main role of this software is that it scans the OMR sheets matches the bubbled answers with the correct answers provided and calculate the marks scored by that particular Candidate [8]. This method is most effective and best way to evaluate large answer sheets with no training required evaluating the paper. This method of OMR scanning can be used for multiple option questions. This software easily used on any OMR sheet after apply some little changes on code.

II. METHODOLOGY

In this paper we discuss about a real-time us of Image processing, form of signal processing for which the input is an image, such as a photograph or video frame, the output may be either an image or a set of characteristics or parameters related to the image as depends on the user need. Image processing is a computer-based technology used in almost every field like:

1. Medical Field Gamma ray imaging PETscan
   X-ray Imaging Medical CT
2. RobotVision

There are several robotic machines which work on the digital image processing. Through image processing technique robot finds their ways, for example, hurdle detection root and line follower robot [1, 2].

3. PatternRecognition

It involves the study of image processing; it is alsocombined with artificial intelligence such that computer-aided diagnosis, handwriting recognition and images recognition can be easily implemented. Nowadays, image processing is used for pattern recognition [1, 2], Biometrics Face Unlock Technology, Fingerprint Detection [1,2], Social Media Apps andwebsiteSnapchat,Instagramfiltersarealsorealtimeand many more [1,2].

OneofthefieldsinwhichImageProcessingiswidelyusedis Education Field. One of the categories in Education field in Image Processing is used is used Analysis of OMR sheetand evaluation of marks according to that answered OMR. There are some steps that are followed in analysis ofOMR

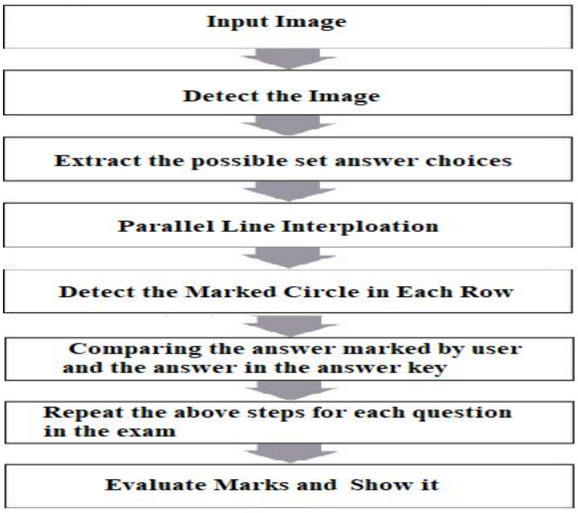

Input Image
Detect the Image
Extract the possible set answer choices
Parallel Line Interploation
Detect the Marked Circle in Each Row
Comparing the answer marked by user and the answer in the answer key
Repeat the above steps for each question in the exam
Evaluate Marks and Show it

Fig 2: Block Diagram of OMR sheet evaluation system

Common tasks in image processing include displaying images, basic manipulations like cropping, flipping, rotating etc, Image Segmentation, Classification and feature extractions, Image restoration and Image recognition. The commonly used Python libraries for Image manipulation tasks used in Aim of Evaluation of OMRsheet:

1. NUMPY: **NumPy** is one of the core libraries in Python programming and provides support for arrays. An image is essentially a standard Numpy array containing pixels of data points. Therefore, by using basic NumPy operations, such as slicing, masking and fancy indexing, we can modify the pixel values of an image. The image can be loaded using skimage and displayed using matplotlib.
2. OPENCV: OpenCV (**Open Source Computer Vision Library)** is one of the most widely used libraries for computer vision applications**. OpenCV-Python** is the python API for OpenCV. OpenCV-Python is not only fast since the background consists of code written in C/C++ but is also easy to code and deploy (due to the Python wrapper in foreground). This makes it a great choice to perform computationally intensive computer visionprogramsimutils: A series of convenience functions to make basic image processing functions such astranslation,rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much easier withOpenCV.
3. argparse: library to parse the command line arguments. It also calls vars on the object to turn the parsed command lineargumentsintoaPythondictionarywherethekeyto the dictionary is the name of the command line argument and the value is the *value* of the dictionary supplied for the command lineargument.

1. INPUTIMAGE

To read an image, we simply call the imread function of the cv2 module. This will return an image as a NumPy ndarray. We can confirm this by calling the type function and passing as input the object returned by the imread function [4]. “`cv2.imshow('test.png',image)`”

2. Detect theImage:

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.

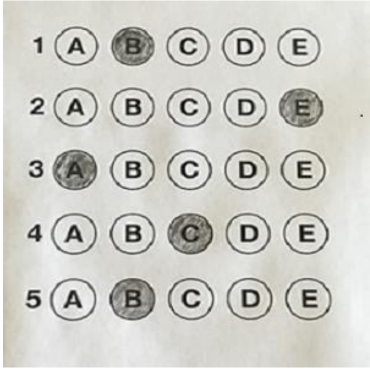


Fig 3: Original image (left) and grayscale image (right)

3. Extract the Possible set answerschoices:

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 [8].

Now, preprocess our input image as: `gray=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)`

```
blurred=cv2.GaussianBlur(gray,(5,5),0)
edged = cv2.Canny(blurred, 75, 200)
```

-converting it to grayscale
-blurring it to reduce high frequency noise
-applying edge detection

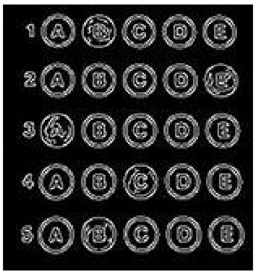


Fig 4: Foreground from background using thresholding

4. Parallel LineInterpolation:

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 centers, it will pass through missing circlestoo. Based on the idea above, we come up with an error minimization method to find the four parallel lines passing through detected circles [6]. Then we get the value which is the color of the pixel. Image pixels are pointswherethevalueofthefunction (pixelcolor)isknown. Interpolation contains intermediate values of the function (gives equation of lines), the image does not necessarily increase, the interpolation methods allow to arbitrarily changing the image size and aspectratio.

6. Identification of the Bubble:

We get the background of the image is black, while the foreground is white after threshing. This

ISSN: 0971-1260
Vol-22-Issue-10-November-2019

binarization will allow us to once again apply contour extraction techniques to find each of the bubbles on questions this means that there exists four parallel lines passing through all the real circles we find it .we find this from Parallel Line Interpolation This is able to find majority of them. This 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, the four parallel lines passingthroughdetectedcirclesand weknowtheequationof the first parallel lineas

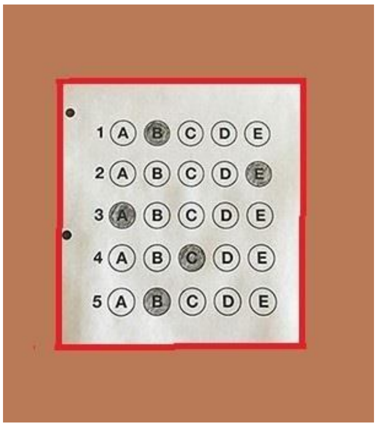


Fig 5: Drawing the contours associated with the exam on our original image.

5. Detect Marked Circle in each row:

After getting outline using contours of that paper We applied a perspective transform to obtain a 90 degree viewing angle of the document. But how do we go about actually grading the document. This step starts with binarization, or the process of thresholding/segmenting the foreground from the background of the image[4]

```
(image,   docCnt.reshape(4,   2))   warped=fourpointtransform(gray,
docCnt.reshape(4, 2))
```

Now we apply thresholding method to binarize the warped piece of paper

```
thresh          =          cv2.threshold(warped,         0,         255,
  cv2.THRESH_BINARY_INV|cv2.THRESH_OTSU)
[1]
```

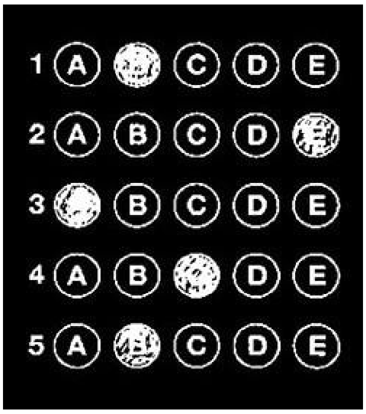


Fig 6: Foreground from background using thresholding after binarization.

$$my = x + c$$

where y is the Y coordinate, x is the X coordinate, m is the inverseoftheslopeoftheparallellines,cistheyinterceptfor line [6,7]. To find contours on our thresh binary image, followed by initializing questionCnts, a list of contours that correspond to the questions/circles [4]. To determine which regions of the image are bubbles, we first loop over each of the individualcontours

```
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
  cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts) questionCnts = []
for c in cnts:
  (x, y, w, h) = cv2.boundingRect(c) ar = w / float(h)
  ifw>=20andh>=20andar>=0.9and ar <=1.1:
    questionCnts.append(c)
```

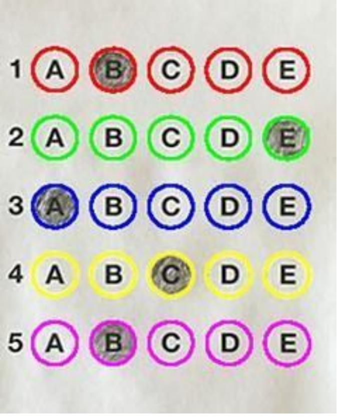


Fig 7: Identification of Bubbles

7. Compare the answer marked user and the right answer

In the starting we already initialize right answers in our code then we compare it by marked answer then evaluate it. Correct answers are which we store in our code as ANSWER_KEY = {0: 1, 1: 4, 2: 0, 3: 3, 4: 1} [4]

Then we get marked value by using our thresh image and counting the number of non-zero pixels (i.e., foreground pixels) in each bubble region then compare it with ANSWER_KEY then we get result on the basis of the testtakerwascorrectorincorrectyieldswhich colorisdrawn on the exam. If the test taker is correct, we'll highlight their answer in green. However, if the test taker made amistakeand marked an incorrect answer, we'll let them know by highlighting the correct answer in red [4] as

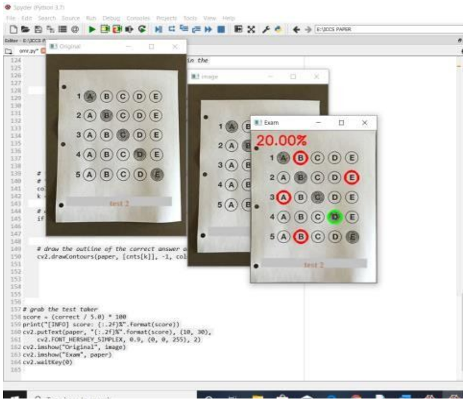
Fig 8: Identification of correct answer

8. Evaluate and showit:

At last we evaluate the result then show it on screen using approach

```
score = (correct / 5.0) * 100 print("[INFO] score:
{:.2f}%".format(score))   cv2.putText(paper,   "{:.2f}%".format(score),
(10, 30),
 cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0,
255), 2)
```

which shows result as:



```
    # draw the outline of the correct answer o
    cv2.drawContours(paper, [cnts[k]], -1, col

# grab the test taker
score = (correct / 5.0) * 100
print("[INFO] score: {:.2f}%".format(score))
cv2.putText(paper, "{:.2f}%".format(score), (10, 30),
    cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 2)
cv2.imshow("Original", image)
cv2.imshow("Exam", paper)
cv2.waitKey(0)
```

Fig 9: Test Case 1 of OMR Evaluation

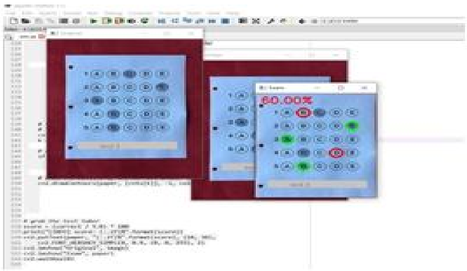
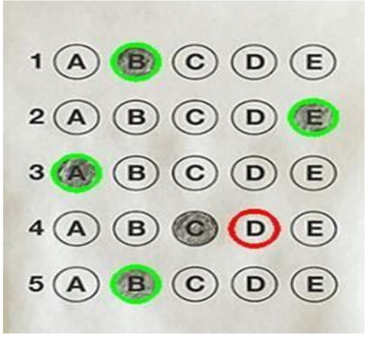


Fig 9: Test Case 2 of OMR Evaluation



III. RESULTS

We tested the code on straight images as well as on captured images. After full processing we were able to detect and evaluate these OMR sheets without leaving any circles and responses undetected and show results [8]. However there were images where we were unable to detect few circles.

IV.CONCLUSION

This software can be used to provide an efficient and easy way to evaluate answer sheet. It is also a less expensive method of the user who does not want to invest lot of money in heavy machinery so it will very useful for them.

Second use is on the basis of time. Time is very important part of every human life which plays a vital role and this method has made to save time. We are able to prove that this method is saves much time for the evaluate the paper and checks for the errors and gives the marks obtained by the candidate show it as real time captured image.

REFERENCES

[1] Sumit Tiwari, Sandeep Sahu “A Novel Approach for the Detection of OMR Sheet Tampering Using Encrypted QR Code”,2014 IEEE International Conference on Computational Intelligence and ComputingResearch.

[2] Tien Dzung Nguyen, Quyet Hoang Manh, Phuong Bui Minh Long Nguyen Thanh, Thang Manh Hoang, “Efficient and Reliable Camera based Multiple-Choice Test Grading System,” IEEE International Conference on Advanced Technologies for Communications (ATC),2011.

[3] Garima Krishna, Hemant Ram Rana, Ishu Madan, Kashif, Narendra Sahu, “Implementation of OMR Technology with the Help of Ordinary Scanner”, International Journal of Advanced Research in Computer Science and Software Engineering,April 2013.

[4] Adrian Rosebrock, professional Computer Vision/Deep Learning developer and researcher , online access.

[5] S, Rakesh and Atal, Kailash and Arora, Ashish, “ Cost Effective Optical Mark Reader,” International Journal of Computer Science and Artificial Intelligence, 2013.

[6] Amit Kumar, Himanshu Singal, Arnav Bhavsar, “Cost Effective Real-Time Image Processing Based Optical Mark Reader”, “International Scholarly and Scientific Research & Innovation 12(9) 2018”.