International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 8, Number 1 (2018), pp. 11-32
© International Research Publications House
http://www. irphouse.com

OMR Sheet Recognition

Lakshay Bansal, Aditya Sood, Harsh Wani

Under the guidance of
Prof. Swarnalatha P.

VIT University, Vellore,

Chennai, Tamil Nadu, India.

Abstract

The main objective of this project is to develop Image processing based Optical Mark Recognition sheet scanning system. In today's world a whole lot many entrance exams are held as competitive exams which comprise of MCQs. The students in such exams have to fill the right boxes or circles for correct answers. Usually a stencil is given to the examiner to evaluate the correct answers to the questions while the examining phase. But this manual process can lead to lot of errors such as counting mistake. To make sure that this is avoided On OMR system is used in which OMR answer sheet is to be scanned and this scanned image of answer sheet is taken in by the software system.

With the help of image processing, the answers marked to all the questions will be found by finding the region of interest .Summation of the marks & displaying of total marks will be also implemented. The implementation is done using MATLAB and MATLAB coding .We also have used GUI for storing the key for all the correct answers.

INTRODUCTION:

In recent years we have seen a massive change in the OMR technology. OMR technology is used everywhere even in schools, colleges and classes. Exams now-a-days use OMR answer sheet checking system as this makes the conduction of exams easy, powerful and cheap. Answer sheet layout is made using sheet design depending on our requirements. The scanner scans the filled sheet. Basically, the checking of an answer sheet and displaying of the results is done by a designed software.

OMR is a technology in which absence or presence of a mark, but not the shape of the mark is detected.OMR software comprehends the output from scanner, converts it into ASCII output. Forms that contain filled small circles referred as bubbles are scanned by scanner. The existence of written marks or filled bubbled can be judged by OMR by recognizing the darkness on the sheet.

Initially the OMR answer sheet is scanned and this image of answer sheet is input to the software system. Image processing helps us to find the answers to all the questions by finding the region of interest. The suggested system is made such that it is very easy to use, operating the system does not require any kind of special training and this system is very cost effective in-case the number of answer sheets to be evaluated is very large and the requirement is frequent.

Other many applications also can be realized by using this system by making small changes in software code and the form design. The proposed system can be realized by using camera instead of scanner by adjusting values of R-min and R-max.

The most perfect application area for OMR systems is University environment. Other than this the government and the medical industry also are major application areas for the OMR systems.

LITERATURE SURVEY:

- A paper regarding this project has proposed a new technique of OMR with ordinary scanner. It says that the given scanned image will be compared to its previously stored template and then on the foundation of the template and given criteria by the user, the answer sheet is marked. This makes the sheet evaluation technique very easy and competent.[1][2].Different modules required for this system are: Answer Feed Module, Criteria Defining Module, Assessment Module, Result Storage Module. Four types of feasibility studies should be taken into account for this system, which are : Technical Feasibility Operational Feasibility Economic Feasibility Schedule Feasibility[3]
- Another research[4] suggested a novel technique for cost effective optical mark reader. Initially in this technique a graphical user interface(GUI) is used to design a custom-made form. Scanning of filled bubbles is done by Regular grayscale scanner. The information of filled bubbles is involuntarily input by processing the scanned images. This system is made up of two independent parts:1)the interface by which the forms are designed and modified and 2)Reading of filled bubbles from the scanned form that is the recognition part.
- Another work done is based on computer vision OMR sheet evaluation with the help of OPENCV. The webcam and its program takes the image when the OMR sheet is kept in front of it. The main objective of this work was to completely eradicate the use of ordinary scanners. OPENCV libraries which are used around the world is used to make a program to further process that image for the extraction of the optical marks. Many steps of image processing are required for this extraction.
- OPENCV is one of the most powerful tool for image processing. Functions to

run the webcam are given by open source computer vision library. These functions are used to capture the image after saving and loading it for further processing.

- Another advancement is done by Ammar Awny Abbas[5], in which with the help of ROI image of base paper with correct answers and images of the test paper are read with the assist of the scanner, after which both images are changed into inverted and binary-tone. The test paper is then rotated to get in line with the base paper image along with the elimination of the small objects. Resulting in the elimination of the questions with multiple answers and multiplication of pre-processed two images leading to emergence of only correct answers in the resulted image.
- Garima Krishna, HemantRam Rana, Ishu Madan [6] came up with an another approach .Here they used an OMR sheet with a specific layout to find the corner points from the image they acquired with the help of scanner. After this it is checked if the image is straight by rotating it and bubbles in the sheet are identified to verify if they are filled ,This is verified by counting the number of black pixels within the bubbles.
- Azman Talib, Norazlina,Ahamad and Woldy Tahar [7] took an different kind of approach with involved matching. It had two parts known as recognition and training. The web-camera captures the OMR sheet image in the training phase, then smoothing filter techniques are used to process the image. Next, manually around one set of answer blocks with the template which is the question number ,a rectangular ROI(region of interest) is chosen. Now, in recognition phase the template image is placed on the OMR sheet for matching. Finally, using the intensity values the candidate image and the template are compared and then it is seen if the answer of the candidate matches with the template or not.
- Andrea Spadaccini described JECT-OMR, a technique that shows scans of multiple choice tests given by students by analyzing the digital images of these scans. A structural analysis of the document in performed by the system to get the chosen answer for every question ,and also has a bar-code decoder which is used to identify an additional information encoded in the document. Python programming language is used for the implementation of JECT-OMR. In order to complete its task it leverages the power of Gamera framework.[8]

METHODOLOGY:

We for this project have used MATLAB as our software system and the scanning of the OMR sheet is implemented with the help of the MATLAB code.

In the code first the image is input and this rgb images are converted into the grey-scale images.

The answer key for all the questions is then stored into the variable key using an GUI.

The first x and y co-ordinate of the first bubble and the spacing between the bubble next to it and the bubble below is input.

For loop is run for the first 10 questions and another for loop inside this for loop is run to get the options.

The grey-scale values of all the bubbles are taken and the mean grey-scale value is calculated(in this case 25).Using a formula based on comparing the mean grey-scale value with a certain calculated value for each bubble , we come to know if the bubble is filled or not. Same procedure is followed for the next 10 questions.

After this an entire for loop for 20 question is run to print the results in the form of a string.

We also have implemented GUI using MATLAB and integrated the GUI code with our MATLAB code in such a way that the answer key input given in the GUI is stored in a variable in our MATLAB code for OMR sheet scanning.

CODE:
Script File

```
omr_1;
uiwait(omr_1);
I=rgb2gray(imread('C:\Users\lakshay\Downloads\scan001.JPG'));
options='ABCD'; % Options

global answers;

ans=''; % This string stores the options entered by the student

key= char(answers); % This string contains correct answers

x=99; % x-coordinate of first bubble
y=200; % y-coordinate of first bubble


s=37; %spacing between each bubble
sr=32; %spacing between consecutive rows


for i=1:10
   sy=y+(i-1)*sr;


   for j=1:4
   sx=x+(j-1)*s;
```

```
 %   disp(sx);
 %   disp(sy);

  %  disp(I(sy,sx));
    if((I(sy,sx)<25))
        c(i,j)=0;
     else
        c(i,j)=1;
    end

    end
end




%Same process for the other half
x=289; % x-coordinate of first bubble in other half
y=200; % y-coordinate of first bubble in other half

for i=11:20
    sy=y+(i-11)*sr;


    for j=1:4
    sx=x+(j-1)*s;
%      disp(sx);
%      disp(sy);


      %disp(I(sy,sx));
     if((I(sy,sx)<25))
         c(i,j)=0;
    else
         c(i,j)=1;
     end


    end

end
    %printing the results in the string
```

```
for i=1:20
    m=0;
    for j=1:4
    if(c(i,j)==0)
        ans=[ans ' ' options(j)];
    else
        m=m+1;
    end
    end
    if(m==4)
        ans=[ans ' NONE'];
    end
end
score=0;
disp(ans);

for i=1:20
    [token,remain]=strtok(ans);
    [ktoken,kremain]=strtok(key);
    if(token==ktoken)
        score=score+1;
    end
    ans=remain;
    key=kremain;
end

 disp('You have scored : ')
 disp(score);
if(score==7)
    status='pass';
else
    status='fail';
end

h = msgbox(['You scored ' num2str(score) 'marks' '     ' ' RESULT : ' status], 'Test
Result');
```

GUI file

```
function varargout = omr_1(varargin)
% OMR_1 MATLAB code for omr_1.fig
%      OMR_1, by itself, creates a new OMR_1 or raises the existing
%      singleton*.
%
%      H = OMR_1 returns the handle to a new OMR_1 or the handle to
%      the existing singleton*.
%
%      OMR_1('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in OMR_1.M with the given input arguments.
%
%      OMR_1('Property','Value',...) creates a new OMR_1 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before omr_1_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to omr_1_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help omr_1

% Last Modified by GUIDE v2.5 02-May-2017 01:04:39

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @omr_1_OpeningFcn, ...
                   'gui_OutputFcn',  @omr_1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT
% --- Executes just before omr_1 is made visible.
function omr_1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to omr_1 (see VARARGIN)

% Choose default command line output for omr_1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes omr_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = omr_1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

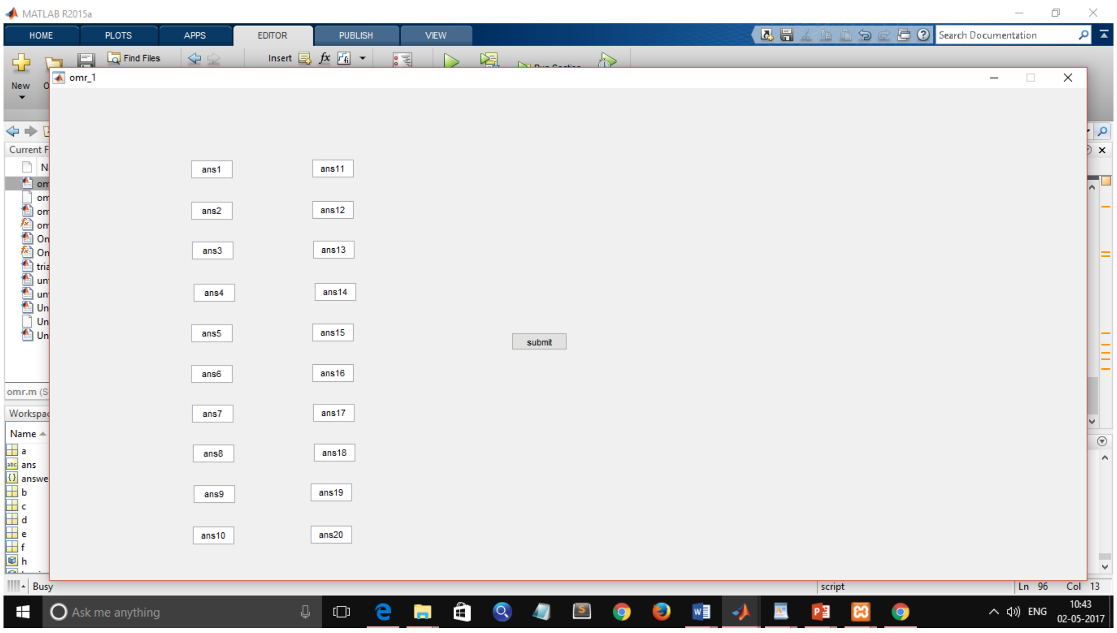
```
function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

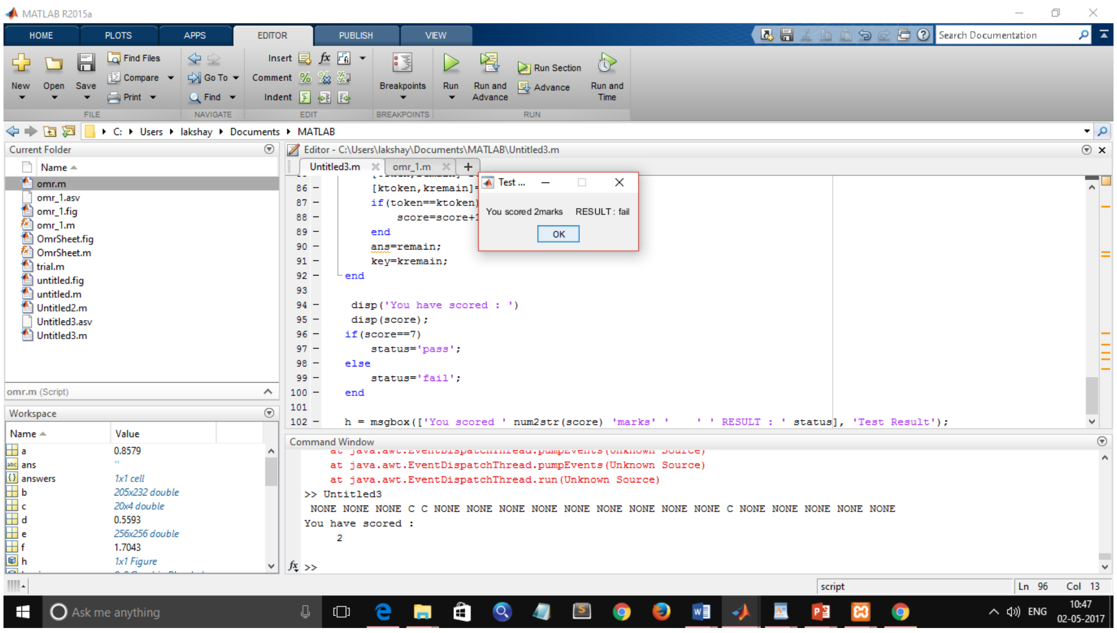
```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11 as a double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12 as a double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a double


% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15 as a double


% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a double


% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%        str2double(get(hObject,'String')) returns contents of edit19 as a double


% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
%        str2double(get(hObject,'String')) returns contents of edit20 as a double
% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global answers;
global S;
S=' ';
answers =
strcat({get(handles.edit1,'String')},{S},{get(handles.edit2,'String')},{S},{get(handles.
edit3,'String')},{S},{get(handles.edit4,'String')},{S},{get(handles.edit5,'String')},{S},
{get(handles.edit6,'String')},{S},{get(handles.edit7,'String')},{S},{get(handles.edit8,'
String')},{S},{get(handles.edit9,'String')},{S},{get(handles.edit10,'String')},{S},{get(
handles.edit11,'String')},{S},{get(handles.edit12,'String')},{S},{get(handles.edit13,'St
ring')},{S},{get(handles.edit14,'String')},{S},{get(handles.edit15,'String')},{S},{get(h
andles.edit16,'String')},{S},{get(handles.edit17,'String')},{S},{get(handles.edit18,'Str
ing')},{S},{get(handles.edit19,'String')},{S},{get(handles.edit20,'String')}});
```

RESULTS AND DISCUSSIONS:





```
        [ktoken,kremain]=
        if(token==ktoken)
            score=score+1
        end
        ans=remain;
        key=kremain;
    end

     disp('You have scored : ')
     disp(score);
    if(score==7)
        status='pass';
    else
        status='fail';
    end

    h = msgbox(['You scored ' num2str(score) 'marks' '      ' ' RESULT : ' status], 'Test Result');
```

You scored 2marks RESULT : fail

```
    at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
    at java.awt.EventDispatchThread.run(Unknown Source)
>> Untitled3
 NONE NONE NONE C C NONE NONE NONE NONE NONE NONE NONE NONE NONE C NONE NONE NONE NONE NONE
You have scored :
     2
```

Name	Value
a	0.8579
ans	''
answers	1x1 cell
b	205x232 double
c	20x4 double
d	0.5593
e	256x256 double
f	1.7043
h	1x1 Figure

CONCLUSIONS

In the end, any new technology that is being introduced should always serve the purpose of well being of common community. OMR (Optical mark recognition) is a data capture technology which is used to entry automated data into a computer system. Today, rapidly it is gaining wide acceptance in educational institutes for various computer related assessments.

The proposed OMR system as we can see in the Results and Discussions is very efficient and accurate. This work focuses development of the OMR for MCQs through using a new technique which paves the way to the future works for development of more efficient OMR in speed and accuracy.

Finally, the system is developed to meet the following goals:

- The system is used for computer aided assessment of class tests.
- The system is designed and implemented with minimum cost.
- The system is designed with easy user interface

REFERENCES

[1] Fairley R. (2002) Software Engineering Concepts (For project size) New York: Tata Mac Graw Hill.

[2] Pressman Roger S. (2001) Software Engineering- A Practitioner's Approach- Fifth Edition, New York: Tata Mac Graw Hill.

[3] Software Project Management – A Unified Framework by Walker Royce

[4] Stephen Hussmann and Peter Weiping Deng, "A High Speed Optical Mark Reader Hardware Implementation at Low Cost using Programmable Logic", Science Direct, Real-Time Imaging, 11(1), 2005

[5] A. A. Abbas, "An Automatic System to Grade Multiple Choise Question Paper Based Exams," Journal of Al-Anbar University for Pure Science, vol. 3, no. 1, 2009.

[6] G. Krishna, Hemant Ram Rana et al, "Implementation ofOMR Technology with the Help of Ordinary Scanner," International Journal of Advanced Research in Computer Science and Software Engineering , vol. 3, no. 4, pp. 714-719, 2013

[7] A. Rajasekaran and Senthilkumar. P, "Image Denoising Using Median Filter with Edge Detection Using Canny Operator," International Journal of Science and Research,vol. 3, no. 2, 2014

[8] Andrea Spadaccini, "A Multiple-Choice Test Recognition System based on the Gamera Framework", 2011, arXiv: 1105.3834v1 [cs.CV] 19 May 2011.