

Pre Processing :

By : Team 405 Found

- 1. Aligning Images & Cropping to relevant parts only
  - a. Using align layers feature in Photoshop, we can align thee layers of two or more images and we can also perform this over a batch of images using recorded actions on sample images
- 2. Resizing
  - a. Resizing the image to 485 px X 300px (or any similar lower resolution to reduced the computation time)
- 3. Enhancing the image
  - a. Images can be enhanced using Adobe photoshop lightroom which improves clarity on images collectively at one time syncing with the metadata of all the images and converting it to black and white image

Build A CLASS

Notes

Age:

Class:

Section:

School:

Phone

Index:

Team Name:

SL NO	OPTION A	OPTION B	OPTION C	OPTION D
1	9	1	20	54
2	16	64	2	0
3	32	8	9	240
4	8	26	153	179
5	8	0	16	1
6	20	96	6	4
7	16	6	84	8
8	19	9	65	8
9	152	38	16	6
10	160	3	12	2
11	68	18	28	4325

SL NO	OPTION A	OPTION B	OPTION C	OPTION D
1	9	1	20	54
2	16	64	2	0
3	32	8	9	240
4	8	26	153	179
5	8	0	16	1
6	20	96	6	4
7	16	6	84	8
8	19	9	65	8
9	152	38	16	6
10	160	3	12	2
11	68	18	28	4325

Image 1 (basic unticked image)

Image 1(processed) (resized, aligned and Enhanced Image)

Build A CLASS

Notes

Age:

Class:

Section:

School:

Phone

Index:

Team Name: Sultan Durrani

SL NO	OPTION A	OPTION B	OPTION C	OPTION D
1	9 ✓	1	20	54
2	16	64	2 ✓	0
3	32	8 ✓	9	240
4	8	26	153	179 ✓
5	8 ✓	0	16	1
6	20	96	6 ✓	4
7	16	6 ✓	84	8
8	19	9	65	8 ✓
9	152	38 ✓	16	6
10	160 ✓	3	12	2
11	68	18	28	4325 ✓

Image 2 (scanned image)

SL NO	OPTION A	OPTION B	OPTION C	OPTION D
1	9 ✓	1	20	54
2	16	64	2 ✓	0
3	32	8 ✓	9	240
4	8	26	153	179 ✓
5	8 ✓	0	16	1
6	20	96	6 ✓	4
7	16	6 ✓	84	8
8	19	9	65	8 ✓
9	152	38 ✓	16	6
10	160 ✓	3	12	2
11	68	18	28	4325 ✓

Image 2 (processed)

This image is resized and aligned as per the alignment of Image 1 As mentioned in step 1.a

## 2) Reading the image data

1. Reading the image data of Image 1 (processed ) in octave using imread.m function which reads the intensity values of each pixel of the image in a matrix of 485 x 300 (Suppose I)
2. Similarly reading the image data for Image 2 (processed) in a corresponding matrix (Suppose J)

## 3) Processing the image

1. Based on the intensity values we have, we calculate the difference of the two matrices giving the approximated areas where there might be tick mark.
2. By measuring the approximate pixel values where each box(area provided for marking a tick for corresponding option of a question) starts and ends, we can calculate the sum of the intensity at every pixel in the box.
3. For every question the tick mark will be present in a box with maximum sum of intensities.



The resultant image formed by subtracting the intensity values of scanned image and sample basic Image. As we can see that only tick marks are left and there are some lines which are being shown because of not perfect alignment of the two images, which is also not possible practically because the images scanned are scanned by mobile [hone camera which incur the lens deviation factors

Algorithm : imageProcess(Z)

```
{
//here Z is the difference of intensity matrix of scanned image and referenced image
//this algorithm performs the processing over the first part of the image
q[1]=172;q[2]=278;q[3]=368;q[4]=458;
                                                                    //pixel values calculated at every box along x-axis

i_now:=32; i_next:=i_now+19;
j_now:=46; j_next:=q[1];
sum:=0,max:=-1;
response={0,0,0,0,0,0,0,0,0,0};
que:=1;
while(que<=11){ //question number for the page upto 11
  p:=1;
  while (p<=4){ //for each question iterating through each part a,b,c,d;
    sum:=0 //finding the sum of intensity in each block and declaring tick at
            //maximum intensity
    for i=i_now to i_next do{ //for iterating through rows

      for j=j_now to j_next do{ //for iterating through columns
        sum=sum+Z(i,j);
      }
    }
    if(sum>max){
      max:=sum; //finding maximum intensity for a question
      response[que]:=p; //storing corresponding tick mark;
    }
    if(p!=4){//not equal to
      j_now:=q[p]
      j_next:=q[p+1]
      p:=p+1
    }
    else{ //if the last part of the current question was checked recently then we are
          //moving to the next question so updating the columns to initial values
      j_now:=46
      j_next=q[1]; //updating columns
    }
  }
  i_now=i_now+23; //updating the rows
  i_next=i_next+20;
  q++;
}
}
```

Similarly, using the same idea we can implement this algorithm for back side of the paper too.

Based on the responses recorded in the above algorithm for every question we can look for corresponding error codes and export them in desired file format.

Output for one paper, it can be calculated similarly

	A	B	C
1	Error Code	Frequency	
2	ME3081	4	
3	ME2081	2	
4	ME306	3	
5	ME3062	0	
6	ME3115	5	
7	ME3116	1	
8	ME3110	2	
9	ME2063	1	
10	ME311A	0	
11	ME311	0	
12	ME3111	0	
13	ME3112	2	
14	Null	8	

**Improvement that can be made:** Using boundary detection better alignment of images can be made possible and thus more accurate results can be achieved.

**Other approaches:** We can define other approaches like pattern recognition for which different patterns of tick marks will be pre-fed to a Machine Learning algorithm and then it can detect the position at which these tick marks are present on the scanned images.