What does this program do?

This program extracts barcode and handwritten values from images.

Template of Image files:

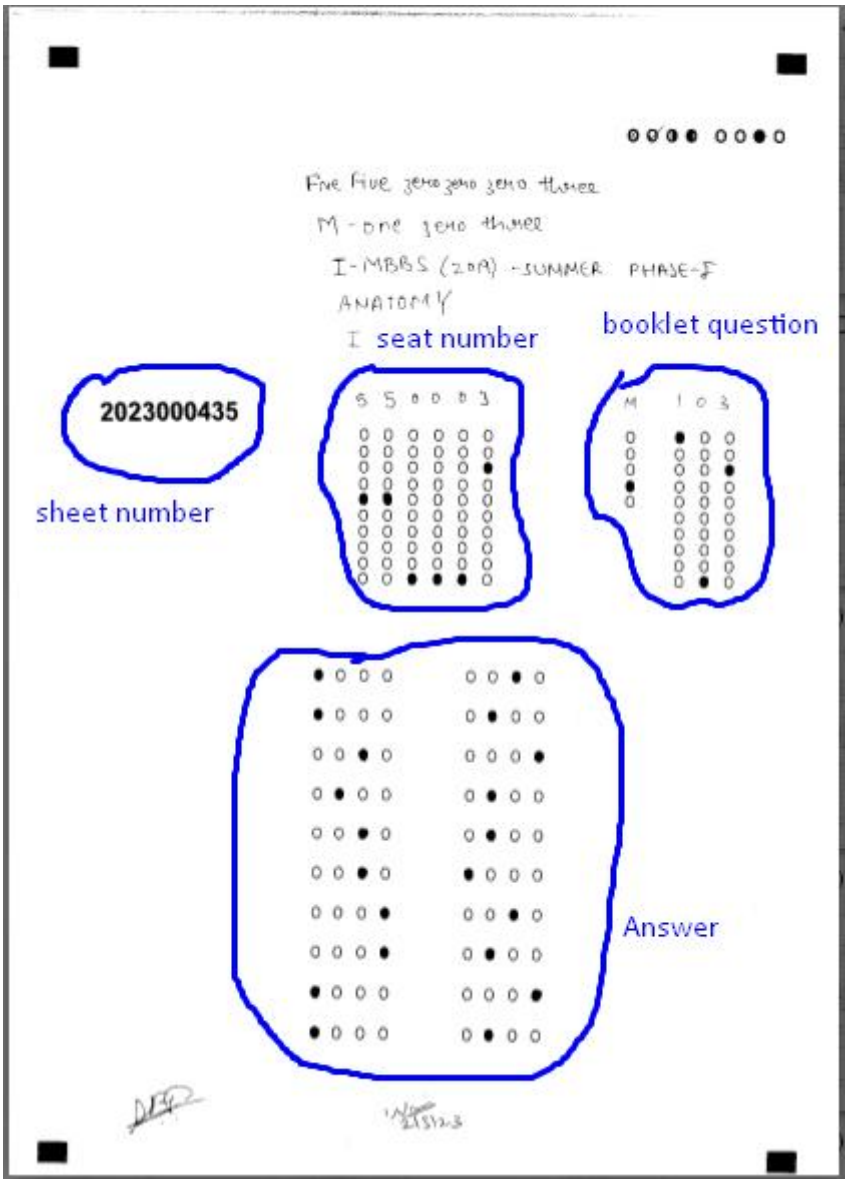


Fig.1 Template file for inputs of program

Quality of files

- Include skewed some images
- Lots of Noise in getting total value.

What is important in this program?

- Sheet number extraction
- Seat number extraction
- booklet question extraction
- Answer extraction.

What engine does this program use ?

Opencv, tesseract, CNN

How does this program work?

Algorithm of this program

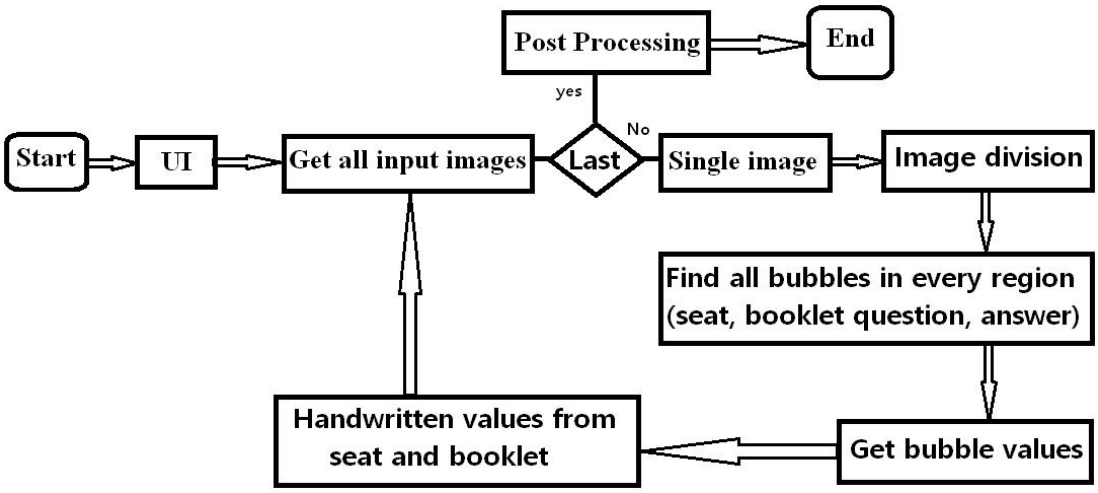


Fig. 2 Overall Algorithm

In code,

MCQ.py: take charge of parts of program starting, UI and several pre-processing such as getting input images.

main_mcq.py: reflect logics of overall algorithm.

In below fig, red takes charge of “Image division”, “find all bubbles from every region”, “get bubble value” and “handwritten values from seat and booklet”, blue for “Post Processing”.

```
    successImgs = []
    for im in img_list:
        t1 = datetime.datetime.now()
        im = im.replace("\\", "/")
        try:
            imBody = im.split('/')[-1]
            print(f"Parsing file_{cnt}/{main_self.total1}: {im}")
            val = next(extractMcq(im, template_path, temp_imgs, index))
            outputs.append(val)
            cnt = cnt + 1
            main_self.cnt1=cnt
            main_self.single_done1.emit()
            successImgs.append(im)
            if cnt % 20 == 0: K.clear_session()
        except Exception as e:
            fail_cnt += 1
            shutil.copyfile(im, os.path.join(err_dir, imBody))
            print(e)

        gc.collect()

    if len(outputs) > 0:
        post_processing(outputs, save_path, successImgs, temp_imgs, index)
        print(f"Success: {len(img_list)-fail_cnt}, Failed: {fail_cnt}")
        duration = datetime.datetime.now() - start
        print(f"Time taken: {duration}")
        return outputs, successImgs
```

Fig 3. Algorithm (explanation in code)

Image division in every image

Following style.json file is used for image division and bubble detection threshold.

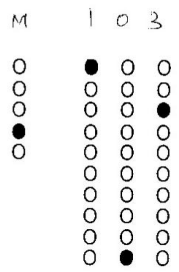
main_highlight.py | main_mcq.py | style.json

```
{
  "idQue_xWratio": 0.66,
  "id_yHRatio":0.26,
  "ans_yHRatio":0.51,
  "threshold" : 0.45
}
```

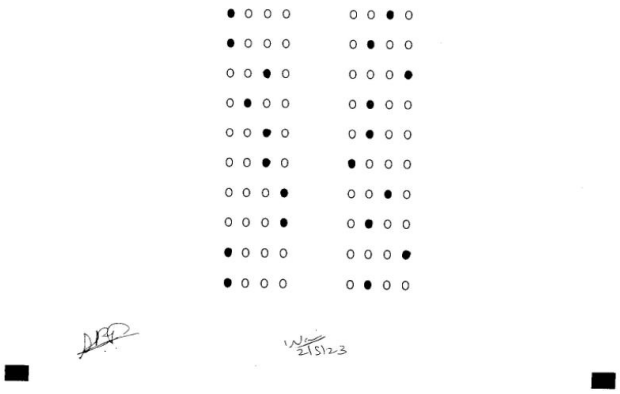
Splitted images:

Seat_img:

Booklet image:



Answer image:



Find all bubbles in every region

```
376     ### Finding all bubble in every region ###
377     id_loc, id_img, idW, idH = Match(template_path/f"O_2.jpg", id_img, threshold, 50)
378     q_loc, que_img, qW, qH = Match(template_path/f"O_2.jpg", que_img, threshold, 50)
379     a_loc, ans_img, aW, aH = Match(template_path/f"O_2.jpg", ans_img, threshold, 50)
380     ### seat number consideration ###
381     seat_num, seat_num_hand, que_num, que_num_hand = 'xxxxxx', 'xxxxxx', 'xxxx', 'xxxx'
382     try:
383         id_coors = getCoors(id_img, id_loc, idW, idH)
384         id_rows, id_cols = getRow_Col(id_coors, 1)
385         id_rows, id_cols = RowsColsCheck(id_rows, id_cols, 10, 6)
386         seat_num = SfindMark(id_rows, id_cols, id_img, idW, idH)
387         seat_num = ''.join(seat_num)
388         idCheckImg = id_img[id_rows[0]-120:id_rows[0]-40, id_cols[0]-30:id_cols[-1]+30]
389         sheetNumImg = id_img[id_rows[0]-120:id_rows[0]+40, 0:id_cols[0]-50]
390         seat_num_hand = next(Recognize_Digit(idCheckImg, "seat"))
391     except Exception as e:
392         print(e)
```

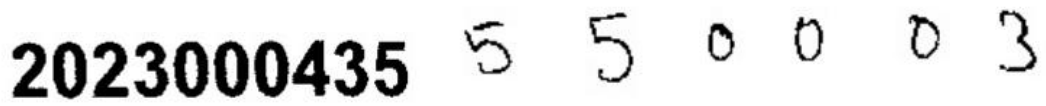
Function **Match(), getCoors(), getRow_Col(), RowsColsCheck()** are used for this.

Match(): return all bubbles similar to “O_2.jpg”.

getCoors(), getRowCol(), RowsColsCheck() get exact bubble coordinates.

To find all bubbles in every region (seat, booklet, answer images), **Match()-> getCoors()->getRow_Col()-> RowsColsCheck()** procedures is applied.

And, cropped sheetnumber image and cropped seat image is gotten by seat number bubble coordinate.



Also, cropped booklet image is gotten by booklet bubble coordinates.

Here, cropped sheetnumber image is not handwritten. This can been extracted by using pytesseract. Its accuracy will be high. But cropped seat image and booklet image can not been by using pytesseract because they are handwritten.

Therefore, sheet number value will be extracted by using tesseract.

Get bubble values

Function SfindMark() is used.

Principle: Based on bubble coordinate (row and column coordinates), when average values of binary image by coordinate and calculated width and height is greater than certain threshold, it will be filled bubble. To get bubble value means finding filled bubbles.

Handwritten values from seat and booklet.

Input images for this have been obtained in the part of “Find all bubbles in every region”.
To get a handwritten value of seat number, model.h5 trained in CNN is used as well as barcode handwritten digit detection.
And, to get the handwritten value of the Booklet image, two steps are needed.
- Handwritten alphabet recognition
- Handwritten digit recognition.

We can use model.h5 for handwritten digits.
And, for alphabet recognition, we use the pretrained model “Alphabet_Recognition”.

Post-processing

- This part makes xlsx file and json of output.