IERG4999U Final Year Project II

Design and Construction of Automatic Grading System with Handwriting Recognition for High School Mathematics Assessments

BY

Chan, Kai Yin

1155124983

A FINAL YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF INFORMATION ENGINEERING DEPARTMENT OF INFORMATION ENGINEERING THE CHINESE UNIVERSITY OF HONG KONG

May, 2022

1. Abstract

This project is an automatic grading system which aims to reduce the repetitive marking of mathematics teachers and help them have more time on preparing teaching material. The need of balancing workload of teachers is very paramount in any school. Workload of teachers has increased with extra administrative tasks and new teaching methods due to the pandemic. The whole system can be divided into three parts. First, a self-design assessment schema is generated by using computer vision tools, it allows teachers to generate different questions and answers with a given topic. Second, the answer sheet can be divided into parts according to the question and extract each answer row which consists of digits, variables or operators written from the student. Third, the handwriting recognition program will perform the recognition on the answer row.

The system is using neural network model for recognition and is trained by the CROHME dataset, a collection of handwritten math expression. It can perform answer recognition for designed topics. A graphical user interface is also provided for input answer sheet, monitoring and controlling the marking process.

2. Introduction

The project is an automatic grading system for mathematics assessment. The mathematics assessment can have a wide diversity on different topics. For the compulsory part of the university entrance mathematics examination in Hong Kong, the topics include rearranging variables, simplifying formulas, and so on. For the Math Extended Part Module 2 (M2), it extends more complex topics including binomial expansion, calculus calculation, and linear algebra. The first part of the compulsory exam is multiple choice and requires the candidate to answer it in a separate answer sheet and have restrictions on filling the answer circle such as using only a black or blue pen to fill in the circle. The multiple-choice answer sheet will be passed to the machine for marking by scanning the filled circle. But for the second part, it is marked by teachers. Most of the topics are requiring direct calculation for the answers. The final answers are mostly a real number or combination of variables. With a model that can recognize the handwriting answer, the system able to take input from the answer of the student and performs the marking. This system is mainly targeted at those topics and the coverage of topics can support the compulsory part and M2.

3. RELATED WORK

Finding the shape of the contour: Detecting the rectangle shape in image is needed in filtering the image outside the answer sheet and dividing the questions. A method to do it is using Hough transform with a sliding window to detect geometric shapes.[1] The Hough transform is mapping every possible parameter of line in pixel domain to the Hough space with ρ and θ and through the pooling get the most possible line. For rectangle, it has four peak to be detected and denote it as H1-H4.

$$\mathrm{H1} = (\rho 1, \theta 1), \mathrm{H2} = (\rho 2, \theta 2), \mathrm{H3} = (\rho 3, \theta 3) \text{ and } \mathrm{H4} = (\rho 4, \theta 4)$$

The algorithm will check the geometric condition of a rectangle to identify it. One condition is checking whether they are in pairs which cloud be H1 and H2 share the same $\theta 0$ while H3 and H4 share the same $\theta 1$. This can be done by calculating the angular difference $\triangle\theta$ between peaks and compare with the angular threshold T_θ. [1]

$$\triangle\theta = \left|\theta_i - \theta_j\right| < T_\theta$$

The generalized Hough algorithm can detect a more complex and arbitrary shape. The algorithm introduced a reference point inside the shape and able to handle the scale and rotation problem. But compared to the standard Hough transform, the GHF will involve more computation and memory consumption. [2] Using the normal Hough transform method already works find in detecting rectangle shape.

Projection of image: After finding the shape and vertices point from the image. This is for mapping the four vertices point to a rectangle shape for the image segmentation for separate questions and for later marking task. It is widely used in the implementation of augmented reality. It uses perspective transform to project the object on the targeted area. [3] It changes the size of the image based on the ratio calculated by the vertices of the target contour. The operation needs prior knowledge of the source vertices and destination four point. Then it calculates the transformation matrix and use it to every pixel in the source image to perform projection. x and y are corresponding to the vertex's location of source image and $t_i x_i$ and $t_i y_i$ is the value mapped to the new four vertices in the targeted size.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} * Trans\_Mat = \begin{bmatrix} t_i x_i \\ t_i y_i \\ t_i \end{bmatrix}$$

Extraction of answers: Extracting the answer information is needed for recognition. The answer format of extracted data could be in row or characters. Location of those answers can be detected by segmentation on the answer. A method to achieve that is using a neural network model to detect the answer area underline (AAU). [4] This method first requires the image standardized to a rectangular shape and it uses a segmentation network to detect the

AAU. The line can also be detected by Hough Transform method and Line Segment Detector. [5] The performances are similar among those method, but the segmentation network approach can detect the straight printed answer line on the answer sheet and filter the false answer line which could be a handwritten line. [4]

Mathematical Expression Recognition: There is a software named MathType created by Design Science that can convert mouse-written symbols into corresponding mathematical expressions. It aims to help people input the mathematical expressions to some document markup languages such as LaTeX. But it does not provide the function of identifying the handwritten on real paper and checking. A CROHME dataset has used in training recognition of handwritten mathematical expressions. In the competition on recognition of online handwritten mathematical expressions 2013, most tasks are using neural network to do the classification. [6]

Mathematics equation solver: EqsQuest Ltd. has developed a platform called Symbolab, which can obtain an equation from the latex or string input by the user, and then solve it step by step. Other platforms provide the same function such as Mathway and Wolfram Alpha. A paper by NGUYEN et al. proposed the Rela-Ops model which can solve the mathematic problem by studying the knowledge of relations and operators. [7] Their model can support complex problem such as linear equation solving. They design different algorithm for different domain of knowledge and able to combine domains. The domain of solving equation will convert to the domain of solving matrix. [7]

$$\{x + \mathrm{y} + 1 \;=\; 0,\; x \;+\; 2\mathrm{y} + 4 \;=\; 0\} \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$$

4. Formatted Question Paper

4.1 Question Design

The formatted question paper is helping the Scanner to detect and catch each question by creating an answer box. The scanner program will able to accurately detect the rectangle shape and perform the perspective transformation. This application is mainly based on Opencv2, a Python library for computer vision tool for making images from scratch. The purpose of having a formatted question paper is to allow the system to identify the type of questions and to use this as a basis for grading. A QR-code is attached to the question paper to pass the information including answer and question type to the automatic grading system (Fig. 1). In addition, the use of question classes in the program allows teachers to generate random variables and values to make various questions to prevent students from cheating. Large format conversion is required when images are transferred to different libraries for processing, such as Latex representation for rendering and OpenCV image format in order to merge together.

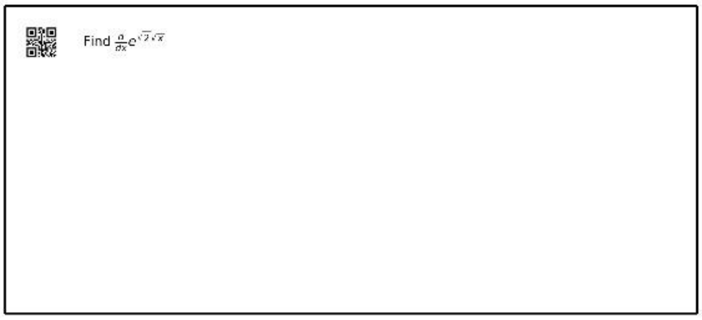

Figure 1 Question design

4.2 Question Paper Design

The layout of question paper is helping to make the question paper tidier. Teachers can choose the number of questions per page and the program will calculate the spacing and width of each question. (Fig. 2). Making this layout mainly need space calculation for each answer box and convert it into image size. The location of questions and QR-code needed to have spaces to avoid collusion. The location and size of the QR-code and font are needed to calculate based on the scale.

Figure 2 Question Paper design for different number of questions

5. Automatic Grading System

The automatic scoring system is based on formatted question papers, image processing, and handwriting recognition programs. (Fig. 3)

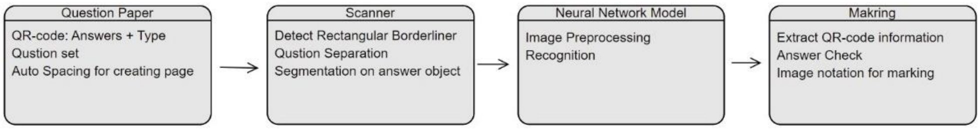

Figure 3 System flow

5.1 Image Scanner

This scanner application is used as an adapter from camera photos to meaningful cropped images. It passes through several image filters and then captures the contour of the image (Fig. 4). It uses the same perspective transformation technique that to obtain a contour size for clean and standard documents. After that, the application uses the image segmentation method to divide the problem into different parts. Suitable image segmentation is a crucial operation because it takes samples from the answer sheet and different segmentation methods will vary the prediction accuracy. Cooperating with the CNN prediction model requires a lot of tuning tasks such as row space of segmentation and overlapping detection. For the contour extraction, the performance of finding the right rectangle vary greatly to different image preprocessing. Before performing the contour detection, the image need convert to gray scale for edge detection, then smooth gaussian filter to reduce the noise of the image and finally canny edge detector. Those filter needed to tune with different parameters such as the kernel size and the iteration time. (Tab. 1)

Name of filter	Parameter
Convert to gray scale	No parameter
gaussian filter	Kernel size = (9, 9), sigma = 1
Canny edge detector	Lower threshold = 0 Upper threshold = 400

Table 1 Function Parameters

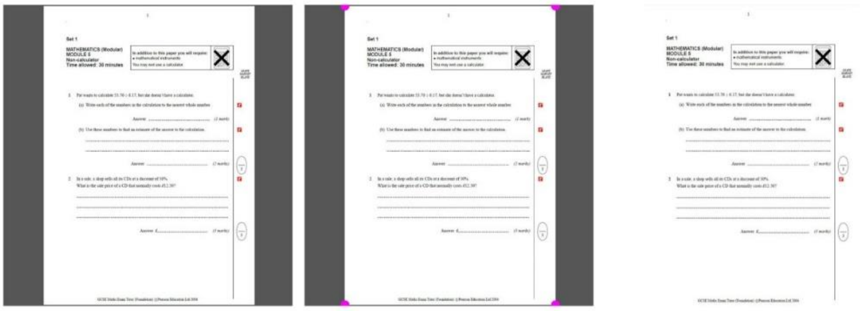

Figure 4 Contour Extraction and Perspective Transform

5.2 CROHME Dataset

The CROHME data set is the training data for the neural network model. It contains a large data set with different mathematical expression, symbols, and digits. The dataset is not suitable for use at first because the digit and symbols have certain properties such as the image are not centered and different sizes. It is needed for further processing in Sec 6.3. I use an image data augmentation function built-in Keras with some adjustments to the image data including width and high shift and rotation. (Fig. 5, 6). For making those adjusted data are suitable to train the neural network model.

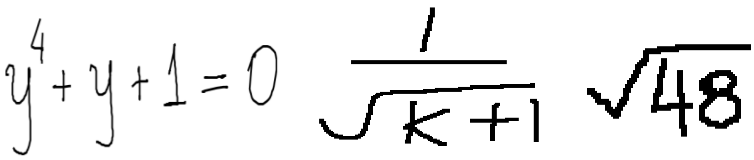

Figure 5 Adjusted CROHME Dataset on expression

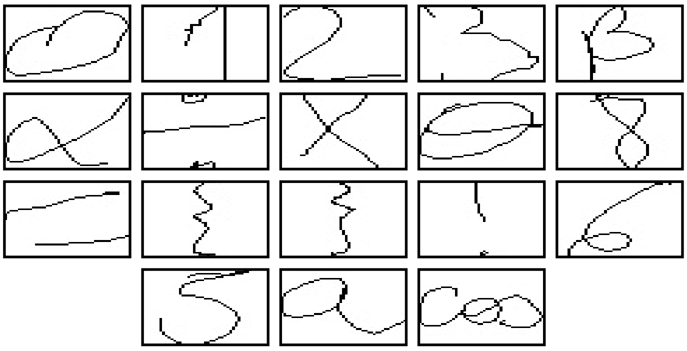

Figure 6 Adjusted CROHME Dataset on single character

5.3 First Model : Convolutional neural network - CNN

Based on the CROHME dataset, The program used TensorFlow to create a convolutional neural network. The model consists of a convolutional layer and a pooling layer, and the classification task is to use SoftMax to make predictions. (Fig. 8) The model starts with a single convolutional layer with 32 filters and size of (3, 3). After passing the maximum pooling layer with a filter size of (2, 2) to reduce the number of calculations, it will enter two same convolutional layers with a 64 filter and the filter size is (3, 3). It passes through max-pooling and flattens the image array to get the features map to the SoftMax classifier. After adjusting the different hyperparameters of the model such as learning rate, batch size, and the number of epochs, it has great accuracy on classification.

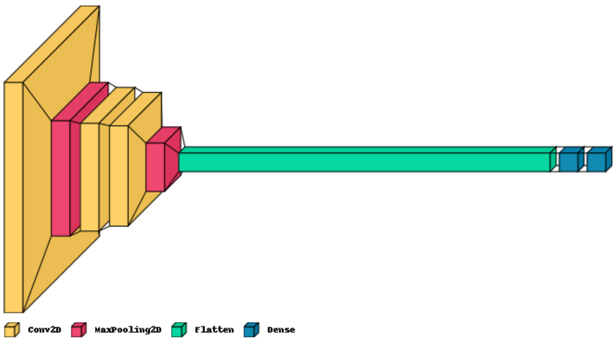

Figure 7 convolutional neural network

Layer	Output Shape	Param #
Conv2D	(62, 62, 32)	320
MaxPooling2D	(31, 31, 32)	0
Conv2D	(29, 29, 64)	18496
Conv2D	(27, 27, 64)	36928
MaxPooling2D	(13, 13, 64)	0
Flatten	10816	0
Dense	100	1081700
Dense	24	2424

Figure 8 Convolutional Neural Network Setting

The model is used with image segmentation techniques to separate each answer object from each line in the answer and pass the position information for building the expression. Then it takes a single answer object as input and outputs predictions. The image segmentation tasks also are handling different situation including separating the consecutive written characters and the position of power number. The model can utilize the information of location of the answer characters for recognition. For example, superscript characters have higher y-index values, while subscript characters have lower y-indexes to identify them. (Fig. 9)

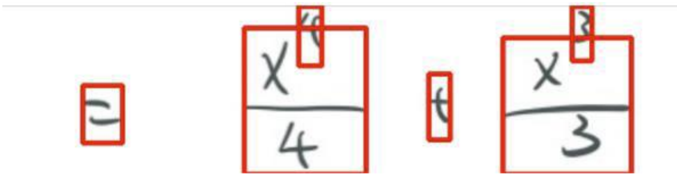

Figure 9 Superscript detection

Limitation:

Image segmentation is the main task of recognition. Due to the variety of mathematical expressions, it is difficult to detect accurately. The positions of different symbols can vary widely, such as line spacing between mathematical operators and multi-index problems: the square root of x to the power of 2, which makes the segmentation of mathematical symbols prone to mistakenly detect and cause loss of information. This model provides the basis and motivation for the Seq-Seq model.

5.4 Second Model : Seq-Seq Model : CNN + LSTM

The second attempt was an extension of the first model, which changed the CNN architecture to the INCEPTION_V3 architecture [8] as encoder and added an LSTM decoder to help with translation. (Fig. 10) The second model erases the problems and limitations of the first CNN model and is adopted by the system.

The model is able to take the entire answer line as input and output as a sequence. Through the CNN model, it outputs the features vector of the input image. These vector will then go through a linear layer to map the vector into the LSTM model. The LSTM model will first detect the \start character, which means it is the start of the sequence, and then pass the \start information to the next state to give contextual information to aid translation. The sequence ends with the character \end.

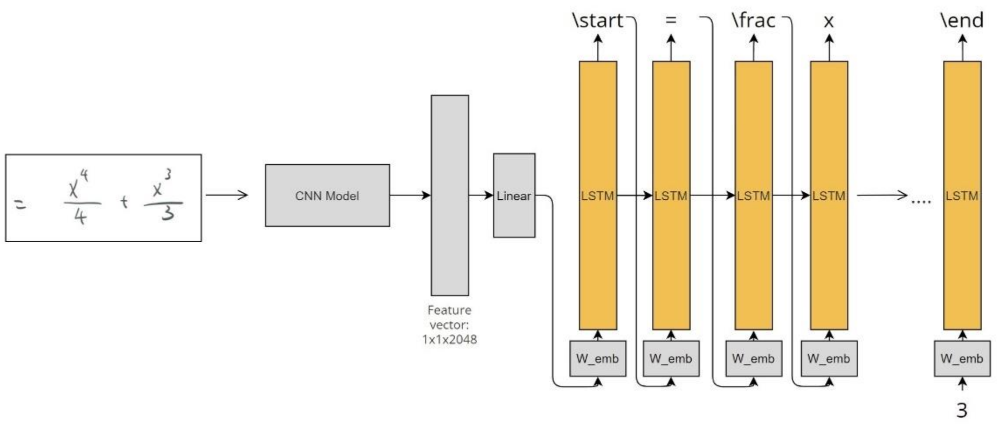

Figure 20 Seq-Seq Model Architecture

Encoder: Inception Architecture

The Inception architecture increases the complexity of the model and the number of parameters and performs well in feature detection. [8] Compared to the traditional CNN model, the depth of the inception network is deeper, and the inception module is introduced. (Fig. 11) Inside the network, each inception module performs multiple filters with different size and max pooling on the same level. It will then concatenate the outputs of all filters for the next layer. [8] The size of the filter will be learned by the model, which helps to catch characters when the font size varies greatly. For example, character 2 has a different size between superscript 2 and integer 2. This is a solution for solving the multi-index problem of first model. Using this architecture as an encoder for images would remove the SoftMax layers and link the linear layer to the LSTM decoder to match the embed size.

ID	name	type	batch	ch_in	dim_in	ch_out	dim_out
1	data	Data		3	299x299	3	299x299
2	conv_conv2d	Convolution		3	299x299	32	149x149
3	conv_conv2d_bn	BatchNorm		32	149x149	32	149x149
4	conv_conv2d_relu	ReLU		32	149x149	32	149x149
5	conv_1_1	submodule(2)		32	149x149	32	147x147
8	conv_2_2	submodule(2)		32	147x147	64	147x147
11	pool	Pooling		64	147x147	64	73x73
12	conv_3_3	submodule(2)		64	73x73	80	73x73
15	conv_4_4	submodule(2)		80	73x73	192	71x71
18	pool1	Pooling		192	71x71	192	35x35
19	mixed_conv	submodule(2)		192	35x35	64	35x35
22	mixed_tower	submodule(19)		192	35x35	256	35x35
42	mixed_1	submodule(22)		256	35x35	288	35x35
65	mixed_2	submodule(22)		288	35x35	288	35x35
88	mixed_3	submodule(13)		288	35x35	768	17x17
102	mixed_4	submodule(31)		768	17x17	768	17x17
134	mixed_5	submodule(31)		768	17x17	768	17x17
166	mixed_6	submodule(31)		768	17x17	768	17x17
198	mixed_7	submodule(31)		768	17x17	768	17x17
230	mixed_8	submodule(19)		768	17x17	1280	8x8
250	mixed_9	submodule(28)		1280	8x8	2048	8x8
279	mixed_10	submodule(28)		2048	8x8	2048	8x8
308	global_pool	Pooling		2048	8x8	2048	1x1
309	flatten	Flatten		2048	1x1	2048	1x1
310	fc1	InnerProduct		2048	1x1	1000	1x1
311	loss	SoftmaxWithLoss		1000	1x1	1000	1x1

Figure 11 Inception Architecture [8]

Decoder: LSTM

The LSTM decoder model takes input from the last state of the inception model. Translations start with \start, which means the beginning of a line, and each translation is based on the previous context. The embedding layer convert the character encoded vector to lower dimension and help the model to learn the meaning. Additionally, LSTMs have gates that control the memory length, which determines whether contextual information will be passed to the next state. Input gate control whether to input this value and calculate the value. Output Gate controls whether to output the calculated value. Forget Gate controls whether to forget the previous memory. For instance, when we are passing a fraction into the model, the information of start of the power character ^ will be passed and help to get the opening bracket "{" and the closing bracket "}" to get the whole translation of superscript and forget it when finish. The training of the model adopted teacher forcing method which use the correct label to train the next state in LSTM to reduce the training computation cost. The model set with parameter embedded layer size and hidden size = 256 and vocabulary size = 112. (Table 2)

Parameter	Value
embed_size	256
hidden_size	256
vocab_size	112

Table 2 LSTM Parameters

5.5 Prediction

Using the splitted answer row that is output from the Scanner application, we pass it to the Seq-Seq model to do the prediction. (Fig. 12)

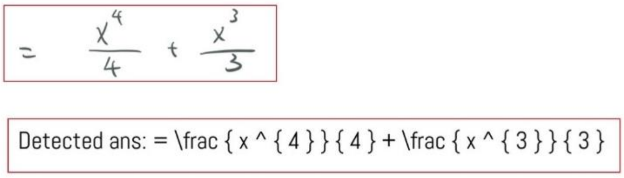

Figure 12 Prediction

5.6 Answer Parsing

The marking is based on the value calculated from the detected answer, with hidden information contained in the QR code. QR-code contains a string with key-value pair: [key, data]. The key refers to the given value of the variable in the question. For answers with multiple variables, returns the QR code of [key1, key2, data]. Students can scan the code with their phone. For prevent users to deceive the scoring system with direct input of the answer data value, the data will be encrypted with an XOR operation with the question number before making the code and decrypted when checking the answer. For instance, The question 2 with answer = 197. The answer is encrypted to 161. (Fig. 13)

Figure 13 Key-value pair : [5, 161]

5.7 Marking

After getting the detected answer from the answer sheet, it passes the math expression as a query to the Wolfram Alpha API with the key to get the calculated value. The marking is based on the comparison of the answer data in QR-code and the calculated value. The annotation with a tick or cross will attached after marking. (Fig. 14)

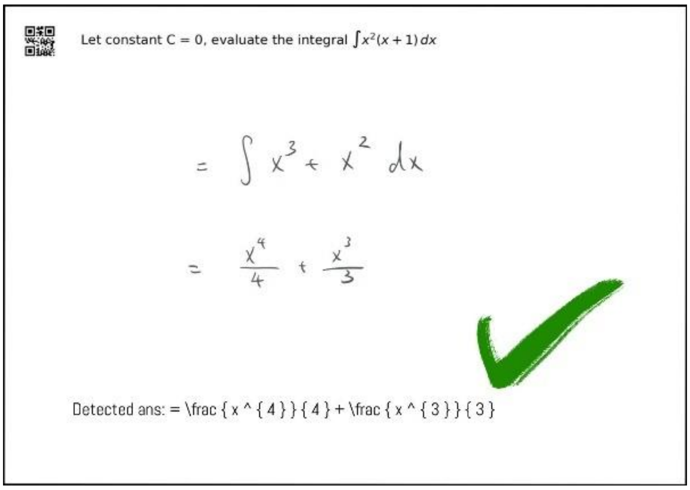

Figure 14 Marked Question

6. Graphical User Interface (GUI)

The GUI is dedicated to teachers. It provides an interface for teachers to view the input image, marking process and the marking result. (Fig. 15) The software packages the required modules, including the Scanner, neural network model and marking program. The program is written based on Tkinter, a Python GUI library for developing applications. It is compatible with other modules.

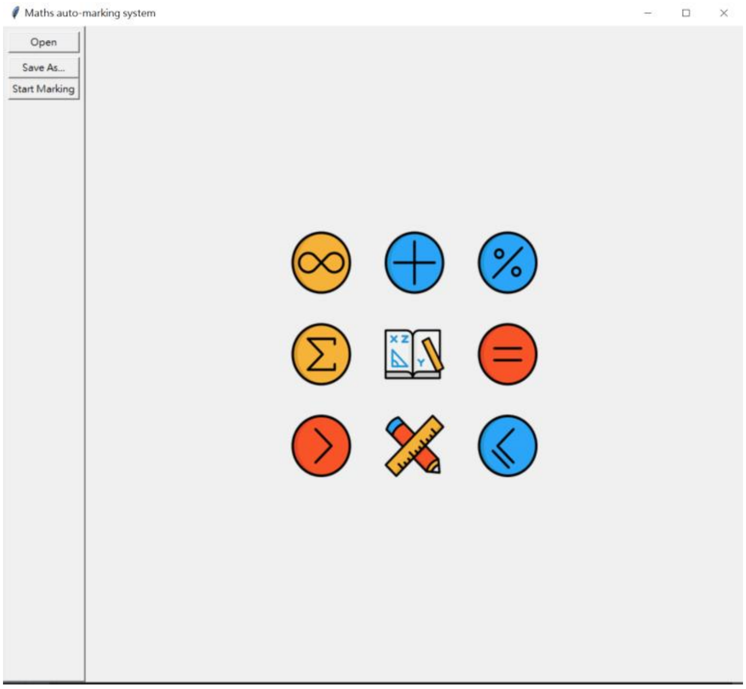

Figure 15 Start Page

6.1 Buttons

The user can import the image file using the "Open" button. After marking, teachers can also save the marked answer sheet to the local disk through the "Save As.." button. (Fig. 16) After import the answer sheet, press the “Start Marking” button, the program will detect the contour of answers and split the question. The button for questions will update based on the number of detected questions. (Fig. 17) The program will start marking after pressing the corresponding question button.

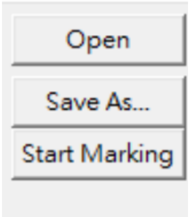

Figure 16 Initial Buttons

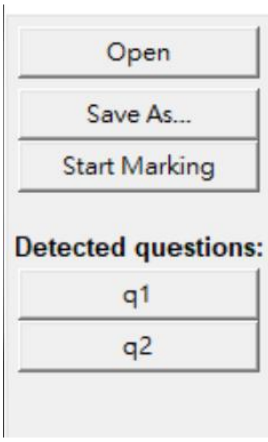

Figure 17 Detected Questions Buttons

6.2 Display Interface

The interface will first display the whole answer sheet in the right hand side of the area. (Fig. 18) Pressing the question button will start the marking. A progress bar will notify the user to wait for the result and the program will fezzes until getting the result. (Fig. 19) Finally, it displays the result with an external windows editing software for image. (Fig. 20) It allows teachers to do further editing or other image processing tasks.

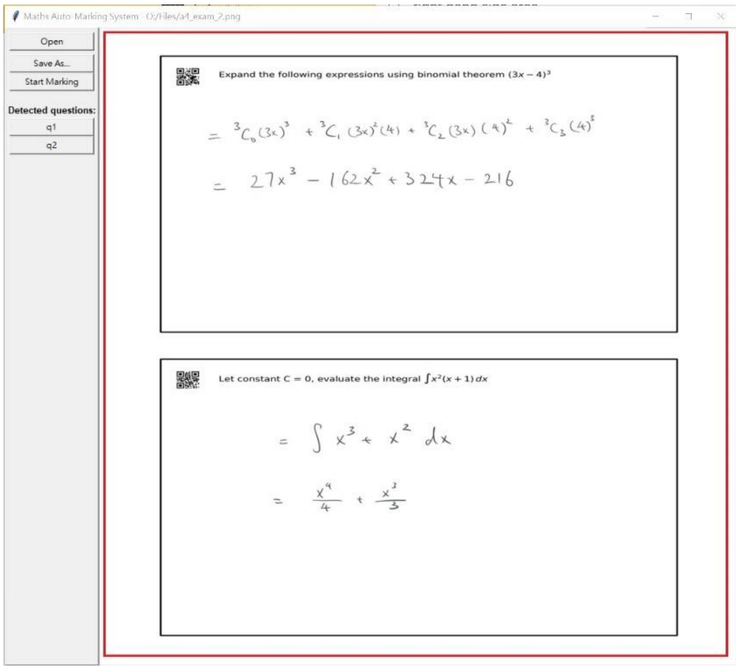

Figure 18 Question Paper Display

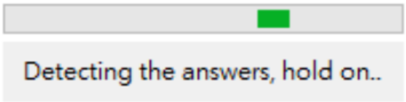

Figure 19 Progress Bar

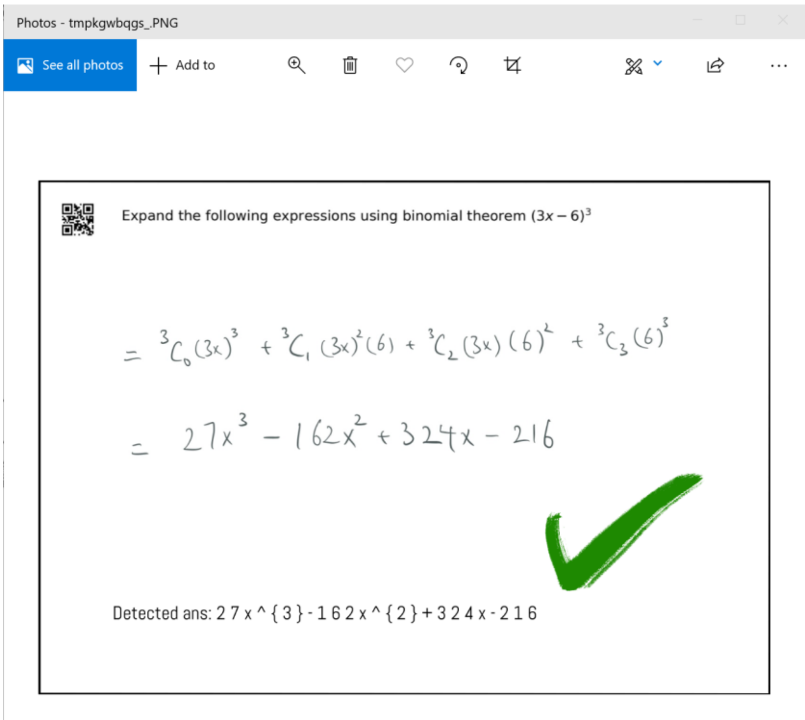

Figure 20 Correct Answer Display

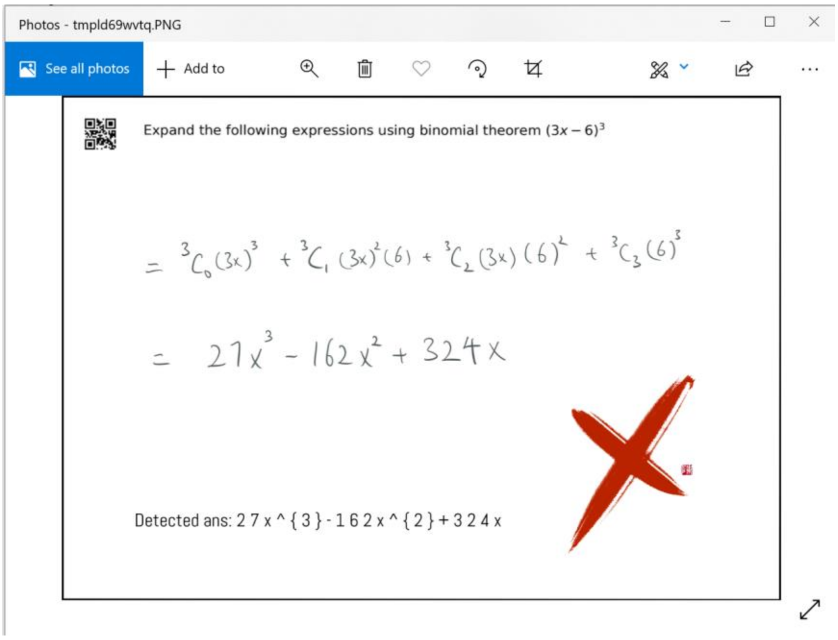

Figure 21 Incorrect Answer Display

7. METHODOLOGIES

The CNN model has a good performance on classification tasks and does not require much research and knowledge on image analysis. In this research, we are going to study the methods required to operate the automatic grading system.

7.1 Contour Extraction

This technique is usually done by locating the next point on the contour within 4 to 8 neighborhoods of the previous point [9]. This can be applied to track the contours of the answer box in the answer sheet. The widely used method is the Ramer-Douglas-Peucker algorithm, which recursively divides the line into smaller lines and finds the two points with the longest distance on the same line. An improved algorithm based on it, by adding a corner index [10]. After detection, the program adopts the RETR_EXTERNAL method which only returns the outer contour from the contour hierarchy (Fig. 22).

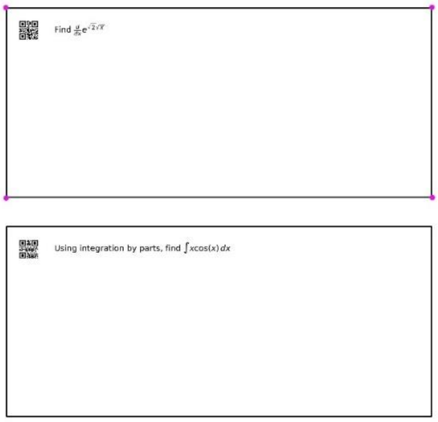

Figure 22 Outer Contour Detection

7.2 Image Segmentation

This is used to separate each object from the answer. (Fig. 25) There are different methods and different performances on segmentation tasks. The scanner application tried to use histogram projection to separate lines and each answer object. The image needs to be converted to grayscale in order to compare the value of black and white. It calculates the normalized histogram from vertical (Fig. 23) and horizontal (Fig. 24) and threshold = 0.001 of the whole count to make a cutline. It works fine when it is a less-noisy image and requires further preprocessing.[11] Another approach is using Markov random field to perform image segmentation and it is able to classify different textures with the training of the corresponding parameters. [12]

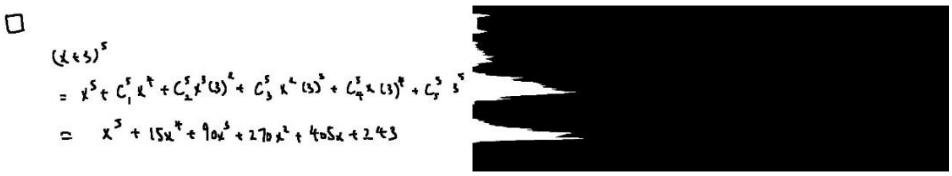

Figure 23 Vertical histogram on rows

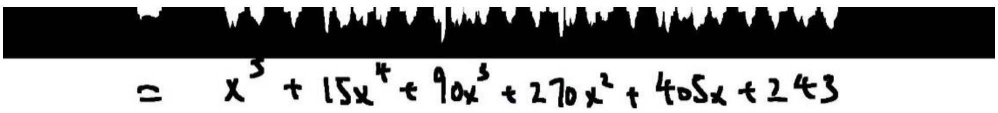

Figure 24 Horizontal histogram on answer object

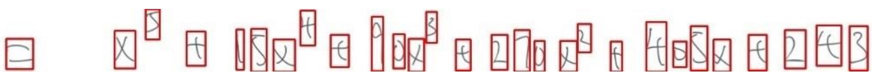

Figure 25 Answer Segmentation

7.3 Image Augmentation

This is a way to generate more data based on current data. This can be achieved by shifting, scaling, and flipping the image. The adjected CROHME dataset in Sec. 5.2, uses augmentation to generate extra data and simulate different handwriting styles, such as slanted words and differences in word size. Using it can avoid the overfitting problem of data and improve testing accuracy.[13] The impact of incorrect parameter settings may cause incorrect input for model training and reduce accuracy. The data used for training and testing uses a generator that is rescaled to the standard, rotated by 20 degrees, the width and height are moved by 10% of the size, sheer and zoom rate with 10% of the size. Example on digit 6, the original digit (Fig. 26) and the generated digits (Fig. 27).

.

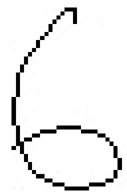

Figure 26 Digit 6

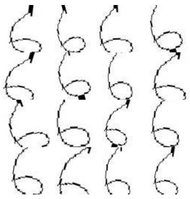

Figure 27 Generated digits from digit 6

7.4 Neural Network Model Training

The model takes input from CROHME DATASET in Sec. 5.2. It required tuning with different hyperparameters including learning rate, momentum, number of epochs and batch size. The setting of parameters can significantly affect the accuracy of the prediction, the model started tuning with common settings. The learning rate controls the speed and convergence of the learning process. Momentum uses the knowledge of previous steps to find the direction of the next step. The number of iterations provides more training for models with serval iterations. The batch size is the number of samples provided to the model in one iteration.

For the second model, with the parameter set above. The hyperparameter including the learning rate of the trained model is 0.001, the momentum is 0.9, the number of epochs = 25, and the batch size = 6.

7.5 Embedding layer

The embedding layer gives the meaning for the character and helps translation to get the whole information. Each character is represented as an integer and represents in One-Hot-Encoding form. The embedding layer learns the featurized representation of the character. For instance, the integer should have a similar vector in the embedding layer while operators have a vector that have large difference with integers. This features help the LSTM decoder to learn the parameters of different gates. [14]

7.6 Answer Parsing Protection

Encryption

Encryption with key according to question number. The encryption key set to set as the question number mod 3, which further use this number as index to extract a key in [98, 99, 100]. The data got encrypted by XOR operation. For instance, the question number 2 will map to index 2 of the array and the data will got encrypted with 100. Question number 2 with key5 and answer 197 will XOR by 100. The key-value pair stored inside the QR-code would be [5, 161]. The list could be change frequently to prevent secret list leak problem.

Decryption:

After reading the data from the QR code. The value is decrypted with the same key that was encrypted and get the actual answer.

8. Result

The project combines different programs to complete the auto-marking task. It first takes an image of an answer sheet as input. (Fig. 28) The Scanner then filters the non-answer information and uses the detected contours to segment the question. The Scanner splits the answer line and outputs the last answer line for marking. (Fig. 29) The Seq-Seq model takes the last answer line and performs predictions for the latex representation. Finally, the marking program check predicted answers, make annotation, merge questions with scoring section. (Fig. 30) The marked answers can be concatenated as a whole page with the score calculated. Using the “Save as..” button can output it in local disk.

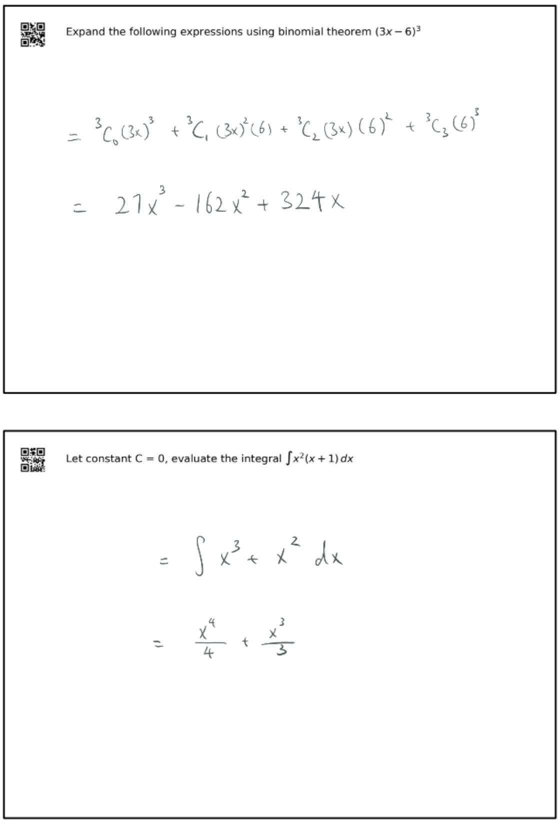

Figure 28 Image Input to the Scanner

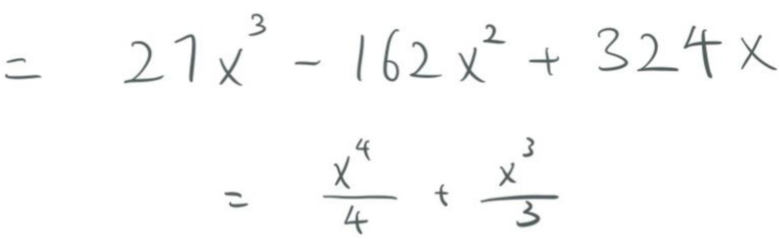

Figure 29 Scanner output and input to the Seq-Seq model

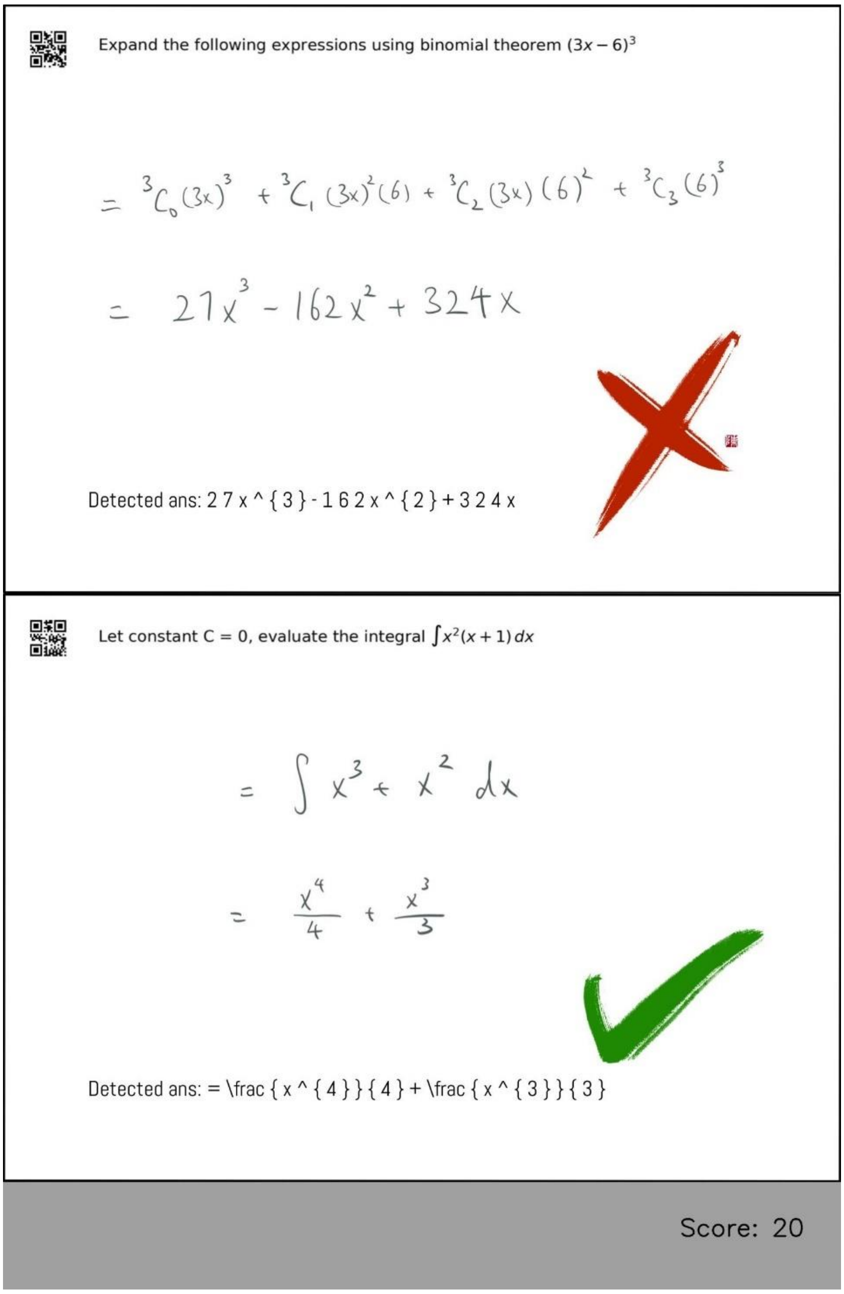

Figure 30 Marked answer sheet

9. Conclusions

I have completed a scanner application, image segmentation program, a Seq-Seq model and a graphical user interface. The whole system involves large data conversion between different programs. Compatibility is required to work with different applications. The Seq-Seq model provides excellent performance and it is contributed by the technique used in image processing and data generator. GUI provides interaction and control of the program for teacher to use.

10. Future Directions

The system can be added a report showing question statistics, such as the most incorrectly answered questions, the distribution of score and the least attempted question. The system may have a reporting scheme which lets user spot the error and give feedbacks to the system. The model now supports questions that have a calculated value for answer. For other types of questions, including proof problems and graph problems, they need a new language processing model in order to get the learn the meaning of those answer. Finally, the recognition model can have more accuracy by having samples from the user and keep training the model.

11. References

[1] Jung, C. R., & Schramm, R. (2004, October). Rectangle detection based on a windowed Hough transform. In Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing (pp. 113-120). IEEE.

[2] Di Ruberto, C. (2012). Generalized hough transform for shape matching. International Journal of Computer Applications, 975, 8887.

[3] Rusiñol, M., Chazalon, J., & Diaz-Chito, K. (2018). Augmented songbook: an augmented reality educational application for raising music awareness. Multimedia Tools and Applications, 77(11), 13773-13798.

[4] Li, X., Yue, T., Huang, X., Yang, Z., & Xu, G. (2019). BAGS: an automatic homework grading system using the pictures taken by smart phones. arXiv preprint arXiv:1906.03767.

[5] Von Gioi, R. G., Jakubowicz, J., Morel, J. M., & Randall, G. (2012). LSD: a line segment detector. Image Processing On Line, 2, 35-55.

[6] Mouchere, H., Viard-Gaudin, C., Zanibbi, R., Garain, U., Kim, D. H., & Kim, J. H. (2013, August). Icdar 2013 crohme: Third international competition on recognition of online handwritten mathematical expressions. In 2013 12th International Conference on Document Analysis and Recognition (pp. 1428-1432). IEEE.

[7] Nguyen, H. D., Do, N. V., Pham, V. T., Selamat, A., & Herrera-Viedma, E. (2020). A method for knowledge representation to design Intelligent Problems Solver in mathematics based on Rela-Ops model. IEEE Access, 8, 76991-77012.

[8] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[9] Papari, G., & Petkov, N. (2011). Edge and line oriented contour detection: State of the art. Image and Vision Computing, 29(2-3), 79-103.

[10] Li, L., & Jiang, W. (2010, October). An improved Douglas-Peucker algorithm for fast curve approximation. In 2010 3rd International Congress on Image and Signal Processing (Vol. 4, pp. 1797-1802). IEEE.

[11] Zhang, J., & Hu, J. (2008, December). Image segmentation based on 2D Otsu method with histogram analysis. In 2008 international conference on computer science and software engineering (Vol. 6, pp. 105-108). IEEE.

[12] Kato, Z., & Pong, T. C. (2006). A Markov random field image segmentation model for color textured images. Image and Vision Computing, 24(10), 1103-1114.

[13] Bloice, M. D., Stocker, C., & Holzinger, A. (2017). Augmentor: an image augmentation library for machine learning. arXiv preprint arXiv:1708.04680.

[14] Li, S., & Gong, B. (2021). Word embedding and text classification based on deep learning methods. In MATEC Web of Conferences (Vol. 336, p. 06022). EDP Sciences.