The Application of Image Processing for Conversion of Handwritten Mathematical Expression

The Application of Image Processing for Conversion of Handwritten Mathematical Expression

Nasuha Iskandar[1], Wei Jen Chew[1,2] and Swee King Phang[1,2]

[1]School of Engineering, Faculty of Innovation and Technology, Taylor's University, Taylor's Lakeside Campus, No. 1, Jalan Taylor's 47500, Subang Jaya, Selangor DE, Malaysia.
[2]Digital Innovation & Smart Society Impact Lab, Taylor's University, Subang Jaya, Selangor DE, Malaysia.

WeiJen.Chew@taylors.edu.my

Abstract. Optical character recognition (OCR) is the conversion of printed or written text from a scanned document or image file into a machine-readable form to be used for data processing like editing. Handwriting has been a way of communication for centuries, but modern technology has made it easier with the introduction of modern computers. While people have adapted to typing out words using a keyboard, formulas and mathematical expressions requires additional add-ons installed in the word processor. The process can be time consuming and tedious. Therefore, an alternative method is proposed in this paper in which handwritten mathematical formulas are converted into computer readable text. Horizontal and vertical projection is used for segmentation while convolutional neural network for character recognition is used to increase the recognition accuracy. The proposed method was able to segment out handwritten mathematical equations from lined papers as well as extract out and identify each character written. The handwritten equation was then successfully converted to a digital format.

1. Introduction

Converting mathematical equation into text format is not new since handwritten mathematical expression recognition has been widely researched for more than 60 years [1]. It is widely used in the field of engineering and science since it requires mathematical calculation. Furthermore, there is also a demand to digitalize old mathematical equations that were written on paper. The challenge about this topic is the segmentation of the mathematical equation because handwritten character has significant number of different symbols, and the symbols can be of different sizes. There is a chance that the character could have been overlapped, or the image is unclear when using optical character recognition (OCR) [2].

LaTeX is a coding software that produces neat mathematical equations, but it requires prior knowledge on the syntax. Most researchers that are required to type out mathematical equations frequently would benefit from learning LaTeX, but for student or other non-researchers, they may not want to learn the coding language.

There are two classifications for handwritten mathematical equations (HME), online and offline. Online handwritten mathematical equations has a wider field of studies and character recognition

accuracy since when the equation is written on the tablet, the stroke is being recorded and passed to the recognition. The second classification is offline handwritten mathematical equation. For this method, the image is taken with a camera or scanned before it is sent to the recognition stage. Offline HME is known to be more difficult and challenging compared to online HME and fewer research had been done for this method since the image of handwritten character can be distorted character images, variations in the size and scale of symbols that makes offline HME harder for recognition [3]. Top 3 online HME recognition systems have obtained recognition accuracy of 80.73%, 79.82% and 79.15% whereas top 3 offline HME achieve recognition accuracy of 77.15%, 71.23% and 65.22% respectively [4]. While online HME is easier and have higher recognition accuracy, offline has more advantage since not everyone has access to a tablet to write out mathematical equations. Besides that, an offline HME can also be used to convert mathematical equations written in papers that have not been digitalized. Using this method, recognition of mathematical equations will not only require less effort in documents but can also be used to transfer existing documents into electronic formats, and between machines when needed [5].

Segmentation is a process of converting mathematical equation into separate character equation. The main challenge for offline recognition is the segmentation part since each person's handwriting is different as well as the problem of overlapping characters, variation in size and scale [2]. A higher rate of segmentation can be achieved when the handwriting of the mathematical equation is clear with proper space in between. Onyshchak [6] developed specific features to identify the character that have multiple contours and merge the two contours, such as the equal sign "=" instead of two minus signs "-". In this paper the feature is lacking on identifying subscript. To achieve efficient segmentation Bharambe [2] proposed a method where the equation is converted into isolated characters and each character is assigned to a number in a labelling process. The result shows that the output was able to differentiate the character of "i" and "j". Furthermore, Hossain et al. [7] use a method called compact horizontal projection where the projection is only on the y-axis to segment quadratic equation. Drsouza et al. [8] conducted research using recursive projection profiling. It has a similar method to the previous research but with an addition of vertical projection. This is useful to segment each character.

Most research use machine learning for classification such as K-NN, Support Vector Machine (SVM) and Multi-Layer Perception (MLP), Feed Forward Neural Network (FFNN), Convolution Neural Network (CNN) and many more. Since CNN produce the highest success rate [9], this paper will be using CNN for the character classification with MATLAB's neural network toolbox. MATLAB provides a support package for transfer learning using a pre-defined layer such as AlexNet, ResNet, VGGNet and would achieve high accuracy [10]. Transfer learning are made to be versatile and it is not designed for mathematical equatons. The network consists of layers that are not needed for classifying mathematical equations and uses computing power. Debnath et al. [11] have design their own network with 7 layers that could achieve 87.72% recognition accuracy without sacrificing computing power.

The process of converting handwritten mathematical equations into a text-based format can be used in digitalizing old mathematics books into text-based format since physical books need to be maintained while digital books could be stored and shared all around the world. This process can be cheaper and faster to do compared to human typing. This project could also be used for student that have to convert their handwritten mathematical equation into Microsoft words for their assignment or uploading mathematical problems to the internet for research.

To conclude, typing out mathematical equations on a keyboard without the knowledge of LaTeX or MathML is a tedious task that requires time to input the individual characters of the equation as compared to writing on paper. Therefore, the objective of this paper is to convert mathematical equations into text base using Horizontal Projection Cutting (HPC) and Vertical Projection Cutting (VPC) on the segmentation and using CNN as the classification that is designed to be suitable for classifying HME. The goal is to have a better segmentation accuracy that will segment overlapping characters and high classification accuracy using minimal CNN layers and training data.

2. Methodology

Figure 1 shows the overall design for the research project. The aim of this project is to convert handwritten mathematical equation on paper, into a text-based format. Hence, the first step is to have handwritten picture of mathematical equation that consist of integral or derivatives. The picture will then be process by image processing. This step is important because the image need to be in black and white with no noise for efficient segmentation. After the pre-processing stage, the image will then undergo segmentation. This is a process that converts the image into individual character. The intention is to generate an algorithm to identify if the segmented image is a single character or needs to be further merged from two small character into one. For example, equals "=" character consist of two minus "-" character. The segmented algorithm needs to detect superscript such as x^2. Moreover, for classification of mathematical formulas, Convolution Neural Network (CNN) will identify the important feature of a mathematical symbol using convolution layer, pooling layer, fully connected layer, and Softmax. To accomplish high optical character recognition, adequate number of training and test datasets should be provided to train the network. Test datasets will be used to evaluate the accuracy of the CNN architecture. After the system identifies the character, the program will convert the symbol into text-based format. This project is a quantitative study since the accuracy of this study will be judged by how many of them is correct.

To conduct this project, MATLAB R2021b is chosen as the platform because it provides toolboxes for image processing and deep learning that would aid with the algorithm.

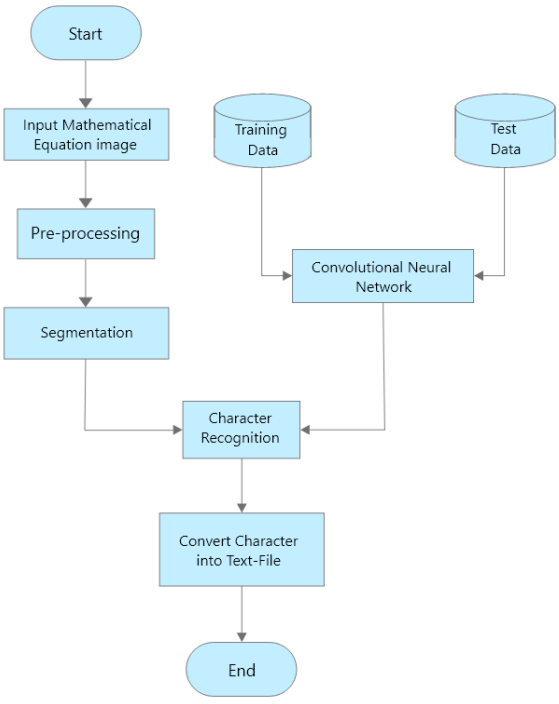


Figure 1. Flowchart of the overall project

2.1. Segmentation

To achieve good results, the segmentation technique needs to ensure that the mathematical equation is segmented correctly. This paper will be using Horizontal Projection Cutting (HPC) and Vertical Projection Cutting (VPC) method to do the segmentation as it requires little computation and produce promising results. Projection cutting method has its disadvantage when the mathematical equation has overlapping characters. To overcome the limitation, an algorithm that is able detect the overlapping character and identify if the equation should undergo further HPC and VPC has been developed.

Figure 2 shows the flowchart of the segmentation. The pre-processing stage will be used to remove noise and lines. After filtration, the image will be turned to grayscale and binarize. The result will only show the mathematical equation with a white background to ease the segmentation stage.

The binarized image will be projected horizontally to identify the gaps in the equation. The HPC will identify the parameters to crop the image and the individual equation is stored in a cell array. VPC is applied at the horizontal crop of the existing cell array images. In this stage, some of the characters are ready to be saved in the file and be classified in CNN. To identify that the crop image is ready to be saved, a connected component will check if the number of strokes on the cropped image is 1. This shows that the cropped image has only 1 stroke and could not be further segmented. Nevertheless, there are characters that have 2 strokes such as "=" that should not be further segmented to become '-'. If the cropped image has more than 1 connected components and when HPC applied to the image has 2 peaks which are within the threshold value, this indicates that the image is "=" and should be saved to a file and not be further segmented. For cropped images that are not within the parameters, the algorithm will identify if the cropped image are overlapping characters by checking that there are no 0 values on the HPC. When there are no 0 values on the projection, the character is isolated using the labelmatrix of the connected components. The isolated image is saved and the rest of the characters will undergo further segmentation. The code will continue to do HPC and VPC until there are no equations left that is unsaved.

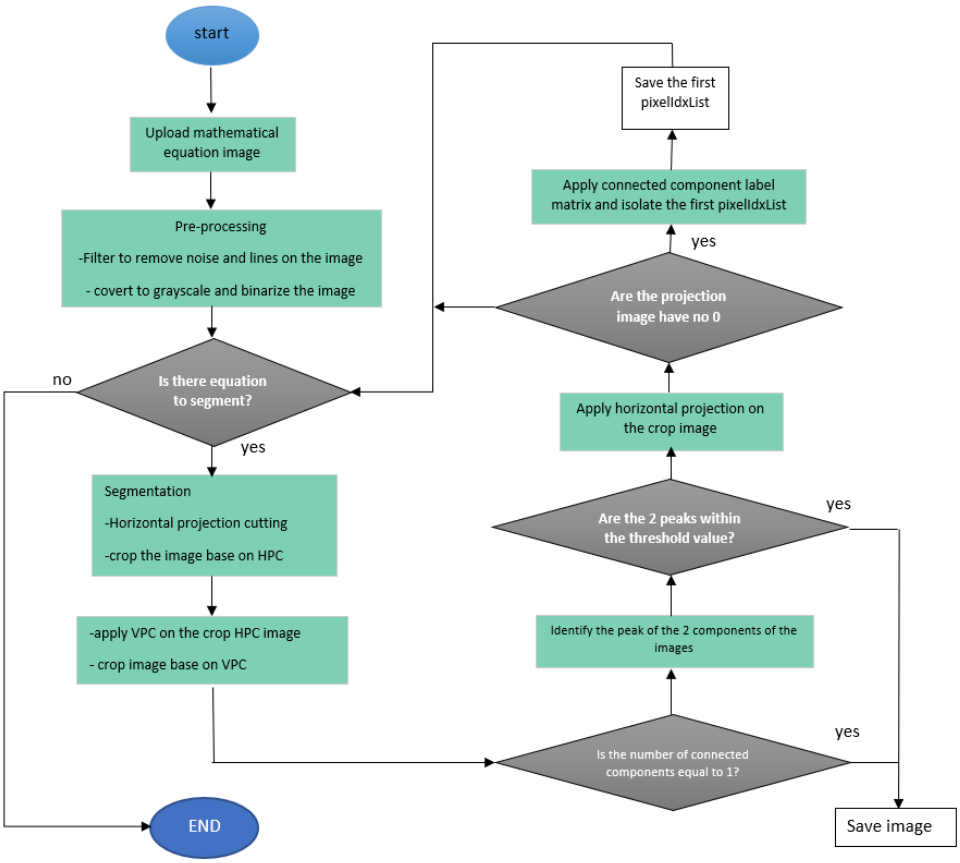


Figure 2. Flowchart of the segmentation algorithm

2.2. Classification

Building CNN architecture is a complex task. Therefore, this project would follow L. Drsouza et al. [8] architecture that have high accuracy while reducing the number of layers needed to save computational

resources. The classification layers are input – convolution – max pooling – convolutional – max pooling – convolutional – max pooling – fully connected and Softmax, as shown in Figure 3. The input image for this CNN network is a 45x45 jpeg image. The filter size of both convolution layer is 5x5 and the pooling layer is 2x2. The learning option used is stochastic gradient descent with momentum (SGDM) with the initial learning rate of 0.0001, max epochs of 30.

To train the network, CROHME's dataset is used. This dataset contains 10,000 different images. To train all 10,000 images will take hours to complete. Therefore, 1000 images from each numbers from 0-9 is extracted out and saved to a different file. The file name is its number for the network to learn. 60% of the images was used to train the network while 40% was used to test the accuracy.

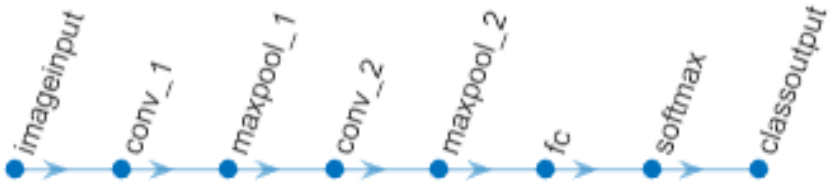
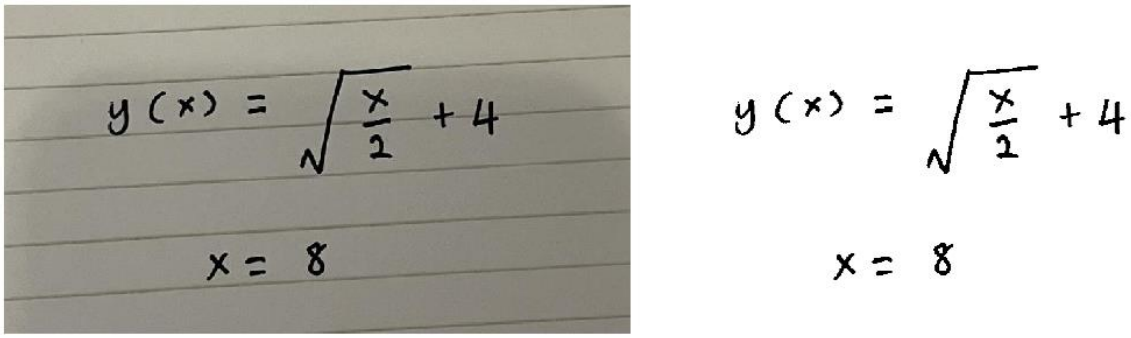


Figure 3. CNN network architecture

3. Results and discussion

3.1. Image processing result

Figure 4 shows original image and after the image is pre-processed. The image was filtered using median filter and adaptive filter, to remove noise and any unwanted lines on the image. After that, it is converted to grayscale and binarized. The binarize method in MATLAB by default is Otsu's method. It is a global thresholding method that converts gray scale images into bi-level images [12]. The equation is written with a 0.7 gel pen. Therefore, noise reduction is not needed. Skew correction is not implemented because the image is written in a straight line. The image was pre-processed successfully and only the wanted mathematical equation was binarized.

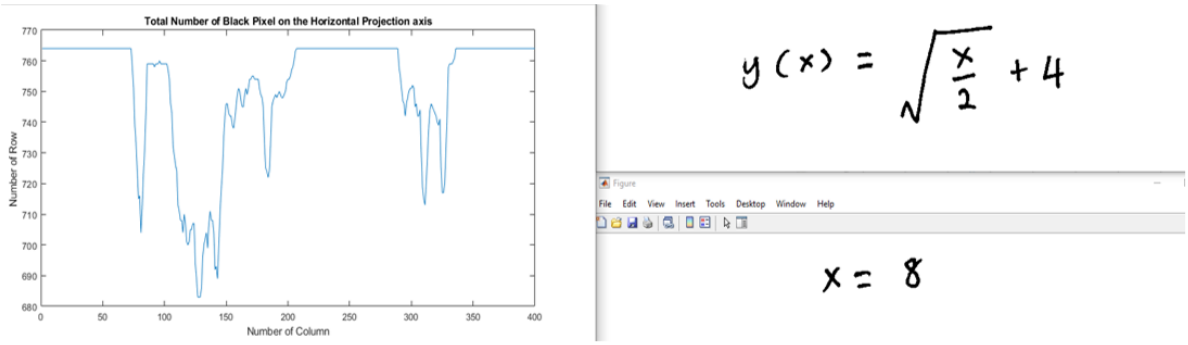


(a) (b)

Figure 4. Image binarization result (a) Original image (b) Pre-processed image

Figure 5(a) shows the total number of pixels on the horizontal projection. Black pixels have the value of "0" while white pixel have the value of "1". The equation is written on a black pixel. The graph that has all white pixel determine that there is a gap between the 2 equations. After determining all the gaps,

the image was then cropped horizontally. Figure 5(b) image shows the result of horizontal projection cutting.



(a) (b)

Figure 5. Horizontal projection result (a) Horizontal projection cutting plot diagram (b) Horizontal cropped image

Figure 6 shows that the algorithm could segment most equation correctly except the ones that is having overlapping components. Those type of image will then go through further segmentation.

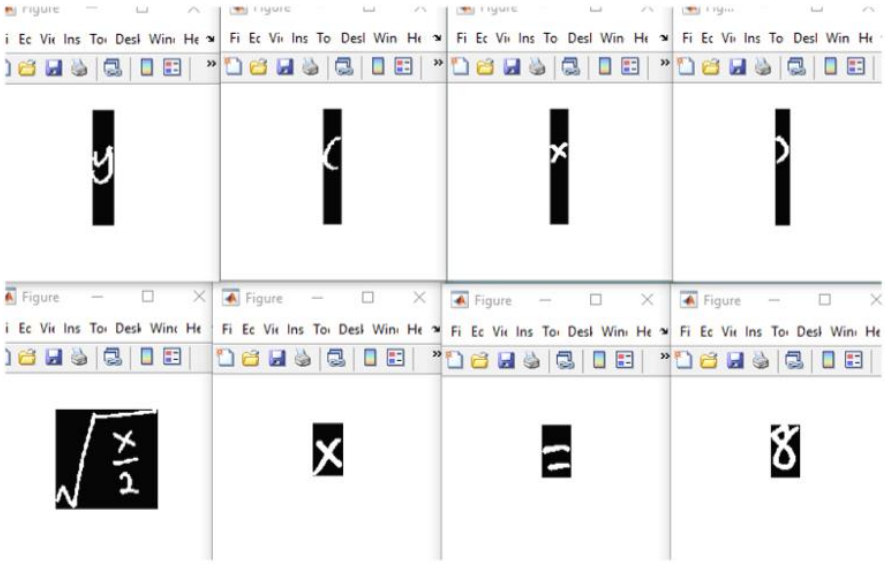


Figure 6. Vertical projection cutting

Figure 7(a) shows that the algorithm could detect overlapping characters and isolate into a different cell array. The remaining characters as shown in figure 7(b) can now do further segmentation. This process will continue until there are no more characters left in the equation to be segmented. The image will be resized and the characters will be converted to a single line to have it to look similar with CROHME's datasets and save into an image as shown in Figure 8.

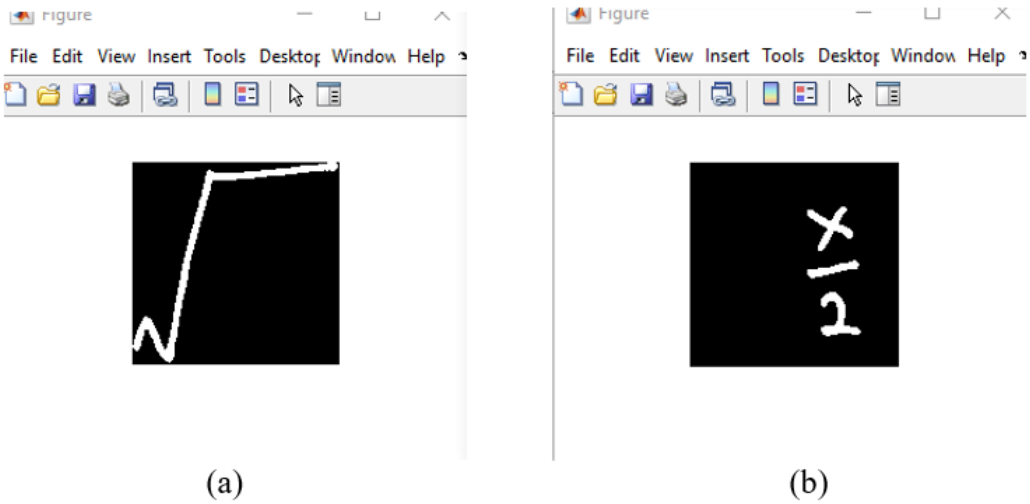


Figure 7. Segmented overlapping characters a) isolated character b) remaining characters

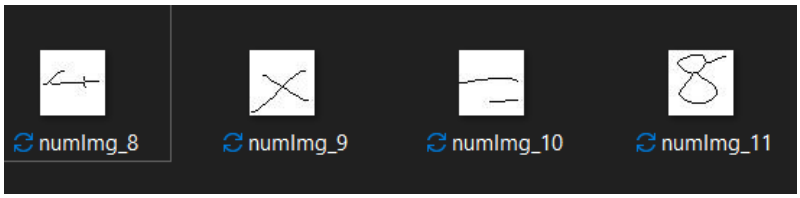


Figure 8. Image saved on file

3.2. Classification result

Table 1 shows is the training result from the CROHME's dataset. The number of epoch is set to be 30. There are 700 iterations and the mini-batch accuracy is from 4.69% to 100%. While the mini-batch accuracy has reached 100% at the forth iteration, it does not mean that the network is perfect and will not misclassify numbers. The test dataset result will determine the effectiveness of the CNN training.

Table 1. CNN training result

Epoch	Iteration	Time Elapsed (hh : mm : ss)	Mini – batch Accuracy	Mini - batch Loss	Base Learning Rate
1	1	00:00:04	4.69%	38.1729	0.0001
2	50	00:00:42	95.31%	0.3484	0.0001
3	100	00:01:08	98.44%	0.4015	0.0001
4	150	00:01:28	100.00%	0.0067	0.0001
5	200	00:01:48	100.00%	0.0007	0.0001
6	250	00:02:06	100.00%	0.0007	0.0001
7	300	00:02:25	100.00%	0.0011	0.0001
8	350	00:02:43	100.00%	0.0009	0.0001
9	400	00:03:01	100.00%	0.0007	0.0001
10	450	00:03:18	100.00%	0.0002	0.0001
11	500	00:03:38	100.00%	0.0004	0.0001
12	550	00:03:56	100.00%	0.0001	0.0001
13	600	00:04:15	100.00%	0.0002	0.0001
14	650	00:04:34	100.00%	0.0002	0.0001
15	700	00:04:52	100.00%	0.0001	0.0001

Figure 9 shows the confusion chart of the predicted test data. The network seems to misclassify between the number two and number six and are most confident in number one because it is easier to identify 1 stroke. The overall accuracy of the testing is 85%. Figure 10 shows the number that have been processed by the segmentation. The classification is very confident that the number given is the number eight and number four.

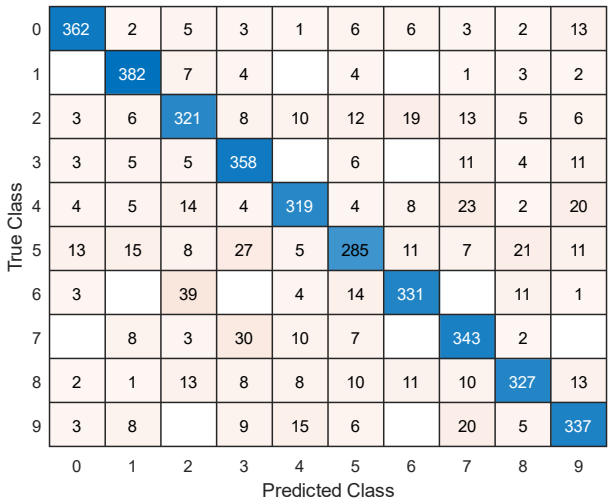


Figure 9. Confusion chart of the predicted test data

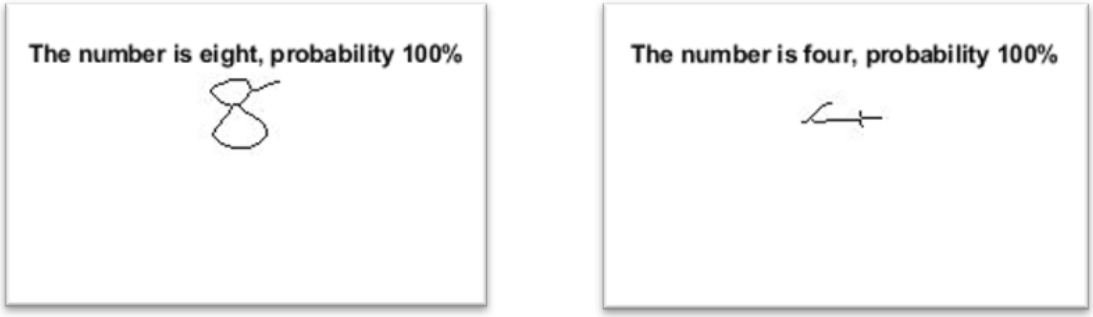


Figure 10. Classification result

4. Conclusion

In conclusion, the project is able to preprocess handwritten mathematical equations on a lined paper as the input image and convert it to a binary image consisting of only the equations. The execution of segmentation using Horizontal Projection Cutting (HPC) and Vertical Projection Cutting (VPC) shows good result as it is able to segment the equation with the overlapping square root character, as shown in Figure 7, and identified equals characters without further segmenting it into two minus characters, as shown in Figure 8. The segmented equation is resized into 45x45 pixel and saved to be classified by CNN. The CNN could identify numbers from 0 – 9 with the test accuracy of 85% by using 600 images to train while 400 for testing. Further testing should be done to have higher accuracy of the CNN. The significance of the project is to be able to produce mathematical formulas on the computer using OCR without typing the equation as it is time consuming. This system could also be used to digitalize old mathematics books that could be shared around the world. In the future, the project will tackle the issue of segmenting imaginary numbers and having higher test accuracy.

References

[1] Anderson, R.H., 1967, August. Syntax-directed recognition of hand-printed two-dimensional mathematics. In *Symposium on interactive systems for experimental applied mathematics: Proceedings of the Association for Computing Machinery Inc. Symposium* (pp. 436-459).

[2] D'souza, L. and Mascarenhas, M., 2018, August. Offline handwritten mathematical expression recognition using convolutional neural network. In *2018 International Conference on Information, Communication, Engineering and Technology (ICICET)* (pp. 1-3). IEEE.

[3] Veres, O., Rishnyak, I. and Rishniak, H., 2019. Application of Methods of Machine Learning for the Recognition of Mathematical Expressions. In *COLINS* (pp. 378-389).

[4] Chan, C., 2020. Stroke extraction for offline handwritten mathematical expression recognition. *IEEE Access*, **8**, pp.61565-61575.

[5] Dai, J., Sun, Y., Su, G., Ye, S. and Sun, Y., 2019, January. Recognizing offline handwritten mathematical expressions efficiently. In *Proceedings of the 10th international conference on E-Education, E-Business, E-Management and E-Learning* (pp. 198-204).

[6] Onyshchak, O., 2019. "A Tutorial on Optical Character Recognition in Mathematical Domain Image Recommendation for Wikipedia Articles View project A Tutorial on Optical Character Recognition in Mathematical Domain." doi: 10.13140.

[7] Hossain, M.B., Naznin, F., Joarder, Y.A., Islam, M.Z. and Uddin, M.J., 2018, June. Recognition and solution for handwritten equation using convolutional neural network. In *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 250-255). IEEE.

[8] D'souza, L. and Mascarenhas, M., 2018, August. Offline handwritten mathematical expression recognition using convolutional neural network. In *2018 International Conference on Information, Communication, Engineering and Technology (ICICET)* (pp. 1-3). IEEE.

[9] Le, A.D., 2020. Recognizing handwritten mathematical expressions via paired dual loss attention network and printed mathematical expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 566-567).

[10] Ramadhan, I., Purnama, B. and Al Faraby, S., 2016, May. Convolutional neural networks applied to handwritten mathematical symbols classification. In *2016 4th international conference on information and communication technology (ICoICT)* (pp. 1-4). IEEE.

[11] Debnath, B., Anika, A., Abrar, M.A., Chowdhury, T., Chakraborty, R., Khan, A.I., Fattah, S.A. and Shahnaz, C., 2018, October. Automatic Handwritten words on Touchscreen to Text file converter. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 0219-0223). IEEE.

[12] Chauhan, S., Sharma, E. and Doegar, A., 2016, September. Binarization techniques for degraded document images—A review. In *2016 5th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO)* (pp. 163-166). IEEE.