The Development of Mobile Application for Assisting COVID-19 Antigen Test Kit Results Reading

Rattapoom Waranusast* and Pattanawadee Pattanathaburt†
*Faculty of Engineering, Naresuan University, Phitsanulok, Thailand
E-mail: rattapoomw@nu.ac.th Tel: +66-055-964009
†Faculty of Public Health, Naresuan University, Phitsanulok, Thailand
E-mail: pattanawadeep@nu.ac.th Tel: +66-055-967398

Abstract— **The antigen test kits or ATKs have been widely used for screening COVID-19 infections because they can detect and give the results quickly and can be done easily by untrained patients. However, reading ATK test results could be difficult for some people and may lead to misinterpretations of the test results. This paper presents a preliminary study for developing a mobile application for helping in reading the results of the COVID-19 ATKs from an image using algorithms based on the YOLO object detection. The results are classified into 3 classes, negative, positive, and invalid. The negative and the invalid results are further refined by using the distances between the visible line and the letters on the test cassette. Experiments were conducted to test the efficiency and accuracy of the developed model with a mean of average precision or mAP of 0.986 and an F1 score of 0.970. The model was developed and put into a prototype mobile application using tools that support cross-platform technology.**

I. INTRODUCTION

Since the emergence of the omicron variant in the latter half of 2021, along with increasing rates of SARS-CoV-2 infections, have made the use of COVID-19 rapid antigen tests gained widespread acceptance as an alternative method for testing COVID-19 outside of hospitals. The COVID-19 rapid antigen test kits, also known as the ATKs in Thailand, have been used in conjunction with real-time polymerase chain reaction (RT-PCR) testing for screening and confirming COVID-19 cases. Patients who use test positive for the ATKs can isolate themselves from other people, even though they don't have symptoms [1]. The rapid antigen tests have certain advantages over the RT-PCR tests. An ATK testing can inexpensively detect and give the results quickly, within less than 30 minutes, while the RT-PCR method, could take 24-72 hours. The fast turnaround time allows for testing prior to entrance into public settings. The rapid antigen tests can be done by patients without having to be an expert or trained and can be done outside of hospital laboratories with expensive equipment. It makes it possible to distribute screening or testing for infected people in areas that are far from the laboratory [2][3]. However, the ATK assays are less accurate than RT-PCR tests, which may be caused by the patients do not follow the testing procedure correctly or the patients may be in the early stages of infection with low viral load causing incorrect test results [4]. Self-reading of ATK tests is also limited, for example, the test line could appear dimmed or unclear, and instruction manuals that explain the procedure usually have overwhelming information and have very small fonts which are not suitable for people with poor eyesight or the elderly. The limitations make the patient feel uncertain of what the test result is and may cause misinterpretation of the test result. Therefore, we came up with an idea to develop a mobile application to help read the results from the antigen test kit to reduce such problems. In addition, such an application can be extended to the governments or health-related agencies to collect patient data for follow-up or statistical analysis.

This paper presents a development of a mobile application to assist patients in reading the results of the ATK. In the next sections, we describe the related work, the proposed development method, experimental results, discussion, and conclusions, respectively.

II. RELATED WORK

The COVID-19 antigen test kit, or the ATK, is a lateral flow test that uses a sample of respiratory secretions from the patient's nasal cavity or throat and is mixed with a buffer solution. The mixed sample is then dripped onto a nitrocellulose test strip. On the nitrocellulose test strip are two stripes printed from reagents and antibodies of animals such as rats, chickens, and goats. When the reagent and antibodies attached to the nitrocellulose sheet capture the SARS-CoV-2 viral protein present in the sample drops, they change their colors, making it visible in about 15 minutes as shown in Fig. 1a [5]. To interpret the test results as shown in Fig. 1b, the control stripe (C) must change to visible red color to ensure the test result is valid. This control color bar is used to indicate that the liquid flow is proper and that the reagents of the test kit are active. If the test stripe (T) changes to a visible color, the test result is positive for the SARS-CoV-2 virus infection, even if the test line is very faint or not uniform. In contrast, if the test stripe position does not have a visible color, this means that the test result is negative for the SARS-CoV-2 virus infection. If no control line is visible, the test is invalid, and the result is not usable.

978-616-590-477-3 ©2022 APSIPA

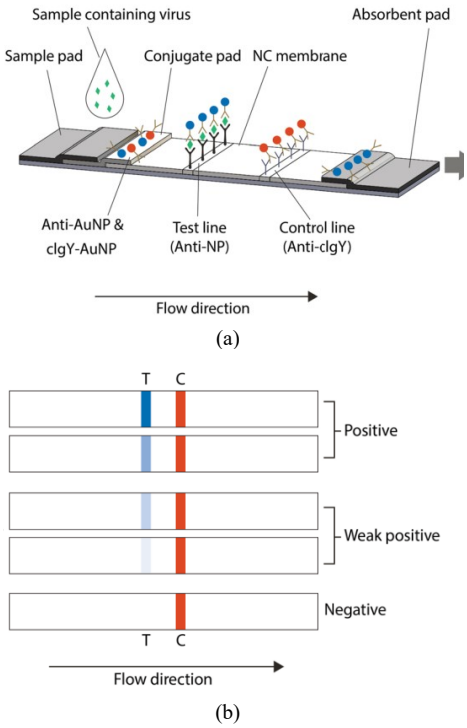


Fig. 1 The working principle (a) and interpretation (b) of a rapid antigen test kit from [5].

In the middle of 2022 as we are preparing this paper, the studies on automatic image-based COVID-19 ATK results reading on mobile phones have been limited, possibly due to the use of ATK test results in the diagnosis or screening of COVID-19 has just been officially accepted for general use in the second half of 2021. For example, in Thailand, there was an announcement "Guidelines for Screening with Antigen Test Kit (ATK)" on July 15, 2021 [1]. However, there have been some studies on automated systems that use image processing to read test results from other types of test kits such as for disease, pregnancy, and substance use. For example, the work of [6], developed a microfluidic paper-based system for the examination of urine samples to allow an analysis of multiple samples from images taken with a mobile phone. Zhu et al. developed a smartphone-based device as an alternative to the wide-field fluorescence microscope [7] and detecting Escherichia bacteria. coli in liquid samples in capillaries [8] for use outside the laboratory. Matthews et al. [9] developed a smartphone-based dengue test system by looking at colors on paper and comparing them with reference colors. Dell et al. [10] presented a mobile application for immunoassay testing using images that are aligned using image processing techniques and using a custom-made special lens to magnify the image. For test kits that are similar to the COVID-19 ATK, Mudanyali et al. [11] developed a mobile phone add-on accessory and a mobile app to read values from malaria, tuberculosis, and HIV test kits using image processing techniques. Carrio et al. [12] developed a smartphone-based device with an add-on accessory to read lateral flow drug abuse test kits using image processing in extracting color features from the visible lines and applying an artificial neural network to classify the results as positive or negative. Manasa et al. [13] and Abhishek et al. [14] developed image processing algorithms to detect the control and test lines on pregnancy test kits and predict the amount of human chorionic gonadotropin (hCG) from the intensity of the lines. However, both papers use a digital SLR camera and fixed settings to control the variations of the images.

III. The Proposed System

The system presented in this paper is a mobile application for assisting COVID-19 ATK results reading. This application works by utilizing cloud computing as an API (Application Program Interface), which the mobile application on the user's side sends an ATK image to be processed on the server side. The mobile application allows the user to take a new ATK photo or to choose an existing photo in the phone's gallery. The application then resizes the image to 640x640 pixels, sends the resized image to the server, and waits for the response. Once the server receives the data, the image is processed with an object detection model using YOLOv5 [15] to classify the result from the image. The result is refined using a comparison of distances between image landmarks, and the result is then sent back to the mobile application on the user's phone. Fig. 2 illustrates an overview of the proposed system for assisting COVID-19 ATK results reading. Each part will be described in detail in the following sub-sections.

A. Mobile Application

The mobile application is developed with Flutter, an open-source framework by Google based on the Dart programming language. Using flutter provides the advantage of being able to develop cross-platform applications which can be deployed on both Android and iOS systems. The Flutter framework provides many libraries for utilizing various hardware and software components inside the phone such as a camera, sensors, and file system. The user interface of the mobile application starts from the logo page as shown by the number (0) in Fig. 3, and then enters the instructions page (1). This page shows the user the instructions for using the application such as a guide on how to take a good ATK picture that works well with the application. The user can then choose (2) to use the image that is already in the phone's gallery (2.1) or choose to take a new photo (2.2). Then the user confirms the image to read the result (3) by pressing the "read result" button. Then, the application sends the image data to the server and waits for the response. When the server sends the result back to the application, The application displays the reading (4), with red letters for the positive/infected result, green letters for the negative/not-infected result, and blue letters for the invalid result. The user can choose to retake a new photo or select a new photo to read another result or press the "Suggestions" button to have the application display a suggestion page (5), which displays necessary instructions, such as how to contact the health care providers, quarantine, treatment, etc. From this page, the user can choose to read the result again or to quit the application.

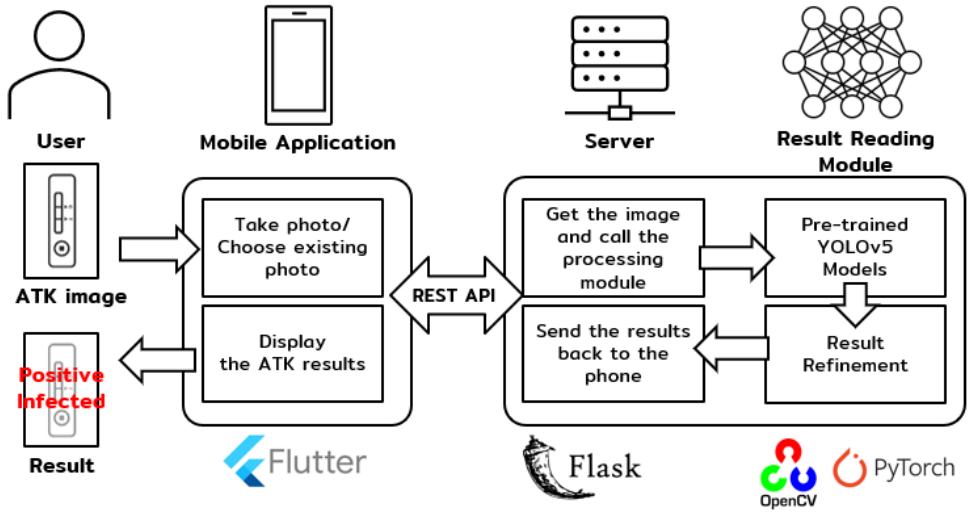


Fig. 2 The overview of the proposed ATK reading system

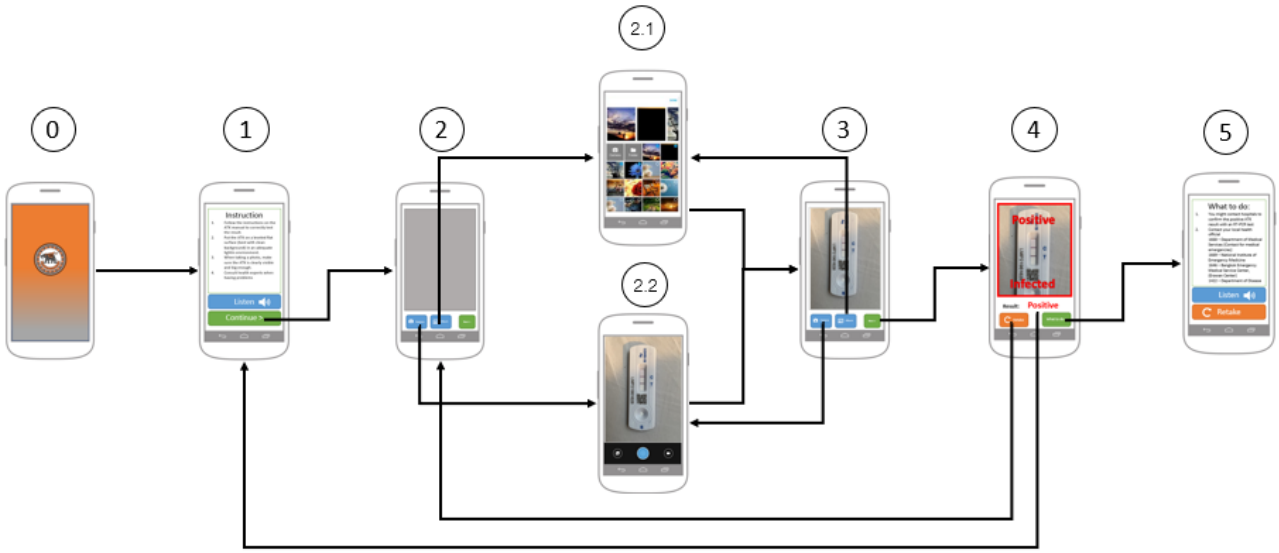


Fig. 3 A diagram of the user interface of the mobile application for assisting COVID-19 ATK results reading

B. Server-side application

The software on the server is developed using Python programming language and the Python Flask library, which is a framework for web services to create APIs connecting to the client-side application. When the mobile application sends an image to the server, the server-side application receives the image and then calls the ATK recognition module developed in Python, with the OpenCV image processing library, and the PyTorch neural network library. Once the module detects and recognizes the ATK test result from the image, it returns the classified result with its position and the bounding box of the nitrocellulose plate (the slot with test and control lines). If the result is "Negative" or "Invalid", the image is passed to the refinement module to confirm the result. The reason that the refinement module is needed is because the invalid results in which the test line (T) is visible while the control line (C) is not visible as in Fig. 4, "Invalid 2" are visually similar to the negative results with the opposite direction of the order of the lines. Therefore, this module is developed to deal with the ambiguity between the negative and the invalid results. The refinement module computes the proximity of the letters C and T on the test cassette compared to the resulting line. In the negative case, the line is closer to the letter C than the letter T, while in the invalid case, the line is closer to the letter T. The server-side application then finally sends the result back to the mobile application as data in JSON format.

C. Deep learning object detection for result reading

This subsection describes how to detect objects and classify the ATK test result from an image.

Data preparation

We collected images of the ATK test kits from our tests, friends' and families' tests, and through internet searches of the keywords "Covid-19 ATK," "Covid-19 RAT," "Covid-19 LFT," and "Covid-. 19 Antigen Test Kit". Then we selected the images with written names or faces of persons to cut out or blur those parts to protect the privacy of the image owners. All images were then renamed to coded numbers. These images were taken in different environments, for example, using different brands of ATKs, different backgrounds, different lighting conditions, and different orientations, and some of them contained more than one ATK in the image. A total of 866 images were collected from this stage. The images were then labeled by defining bounding boxes around the objects of interest, which are the test strips of the ATKs. The objects are classified into 3 classes, which are "Positive", "Negative", and "Invalid" according to "Reading the results of the Self-test Antigen Test Kit (ATK) for the public" [4]. Out of 866 images, 538 objects were labeled as "Positive", 806 objects were labeled as "negative", and 62 objects were labeled as "Invalid". Fig. 4 shows an example of object labeling, negative, positive, and invalid.

Object detection

The labeled images were fetched to the object detection step which utilizes YOLO: You Only Look Once, as the main object detection algorithm. YOLO is a popular single-stage object detection algorithm based on the convolutional neural network used to predict the positions and probabilities of objects in an image [15]. This algorithm is popular because of its speed and its high accuracy. This algorithm divides the input image into regions and each region serves as an object detector if the center of the object is in that region. YOLO may give results as multiple bounding boxes from one object so that the non-maxima suppression algorithm is used to select a box that represents the object's position. When an object is detected by the YOLO algorithm, the algorithm results in the position of the object in the image as a bounding box. We use YOLO version 5 (YOLOv5) in the object detection module on the server, which has higher accuracy and faster processing speed than the previous versions by adding a focus layer and adjusting various parameters. This version uses the PyTorch library instead of the Darknet library that had been used in versions 1 to 4. These improvements make YOLOv5 faster and more memory-efficient to train than previous versions.

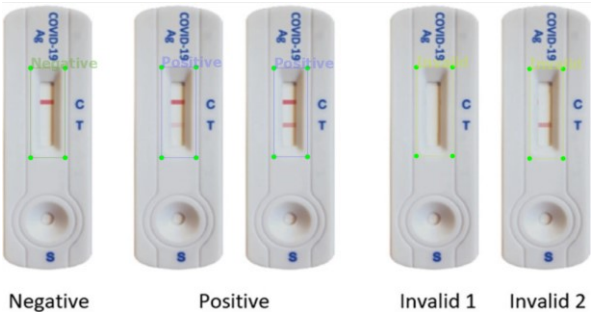


Fig. 4 Examples of object labeling to train the object detection model

A total of 866 ATK images obtained in the data preparation were randomly divided into 2 sets: a training set of 726 images and a test set of 110 images. The training set was used in training a YOLOv5 model for detecting the previously mentioned objects.

We increased the variety of images through the process of data augmentation. The data augmentation in YOLOv5 training is not directly increasing the number of images like a manual augmentation or using other third-party libraries for data augmentation. Instead, it randomly changes the images to increase the variety as defined in the hyperparameters, which occurs during the training process when the algorithm gets a batch of images to train in each epoch. This augmentation technique guarantees that every training image is unique. Fig. 5 illustrates the procedure for image data augmentation in training with YOLOv5, and Table I shows the image augmentation hyperparameters we used in the data augmentation process. To reduce training time, we also used transfer learning from the pre-trained YOLOv5L model, which was trained from the COCO image dataset.

We trained this ATK image dataset, on the Google Colab service, which allocated a virtual computer with specifications equivalent to a computer with an Intel(R) Xeon(R) CPU@2.30GHz processor, 13 GB of RAM, and an Nvidia Tesla T4 graphics processing unit (GPU). The training ran for 500 epochs using approximately 24 hours.

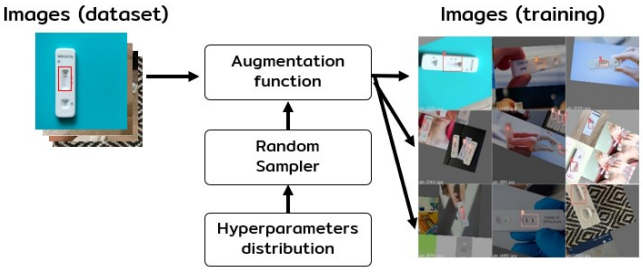


Fig. 5 Image data augmentation in YOLOv5 training

TABLE I. HYPERPARAMETERS IN IMAGE DATA AUGMENTATION

Image operation	Range
Adjust HSV-Hue	+/- 1.5%
Adjust HSV-Saturation	+/- 50%
Adjust HSV-Value	+/- 50%
Rotate	+/- 30 degrees
Translate	+/- 10%
Resize	+/- 50%
Shear	+/- 5 degrees
Flip horizontally	probability = 0.2
Flip vertically	probability = 0.2
Image mosaic	probability = 0.1
Image mix-up	probability = 0.2
Copy-paste parts of image	probability = 0.1

D. *Result refinement*

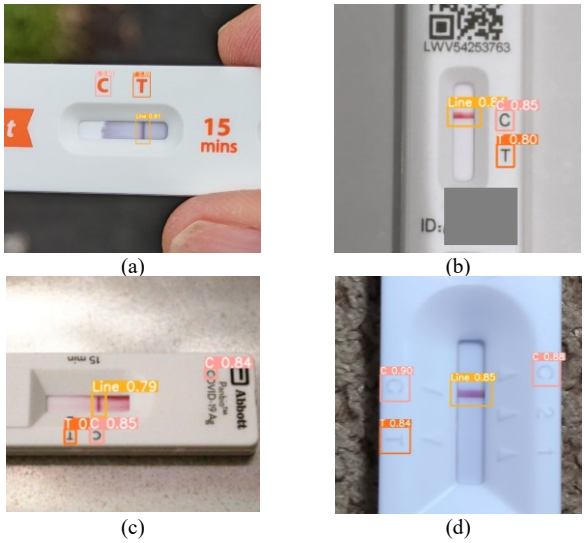
As mentioned earlier, the "Invalid 2" results as in Fig. 4 are the same as the "Negative" results when looking from the opposite direction (up-side-down). Even though this type of invalid result is rare, we developed an algorithm to deal with this ambiguity, based on the comparison of distances between the distance from the resulting line to the letter C on the cassette and the distance from the same line to the letter T. In the negative case, the distance between the line and the letter C is shorter than the distance between the line and the letter T, while it is the opposite in the invalid case.

Test kit landmarks detection

To compute distances between the landmarks, the line, the letter C, and the letter T, we have to begin with locating these landmarks. We utilize the YOLO object detector to detect these landmarks in the image of interest. We used the same image dataset as we used in the result reading process mentioned in the sub-section "*C. Deep learning object detection for result reading*". The landmarks in the images were labeled as "Line", "C", and "T" classes. Out of 772 training images, 1,699 objects were labeled as "Line", 2,315 objects were labeled "C", and 1,350 objects were labeled as "T". We also used the data augmentation process to increase the number of training data. Fig. 6 shows examples of detected landmarks in some images. Noted that some images have more than one C/T letter as in Fig. 6 (c) and (d) in the same image, and letters on some cassettes are embossed instead of printed as in Fig. 6 (c).

Distances comparison

After the landmarks are detected, the Euclidean distance from the center of the detected line to the center of the closest letter C and the Euclidean distance from the center of the detected line to the center of the closest letter T. If the center of the line is closer to the letter C than to the letter T, the result is "Negative", or else the result is "Invalid". Fig. 7 shows the pseudocode of the distance comparison algorithm.



(a) (b) (c) (d)

Fig. 6 Examples of detected landmarks on test kit cassettes with (a) an invalid result, (b) a negative result, (c) more than one C letter, and (d) an embossed cassette.

Algorithm 1 Distances comparison Algorithm

Input: A box of *line*, and lists of boxes *listC, listT*
Output: A class of the ATK result

procedure DISTANCESCOMPARISON
  $(xl, yl) \leftarrow \text{center}(line)$
  $distC \leftarrow \infty$
  $distT \leftarrow \infty$

  for each c in$(listC)$ **do**
    $(xc, yc) \leftarrow \text{center}(c)$
    $dist \leftarrow \text{sqrt}((xl - xc) * (xl - xc) + (yl - yc) * (yl - yc))$
    $distC \leftarrow \min(dist, distC)$
  end for
  for each t in$(listT)$ **do**
    $(xt, yt) \leftarrow \text{center}(t)$
    $dist \leftarrow \text{sqrt}((xl - xt) * (xl - xt) + (yl - yt) * (yl - yt))$
    $distT \leftarrow \min(dist, distT)$
  end for

  if $distC < distT$ **then**
    return "Negative"
  else
    return "Invalid"
  end if
end procedure

Fig. 7 The pseudocode of the distance comparison algorithm

IV. EXPERIMENTS AND RESULTS

To evaluate the effectiveness of the trained models, we used Precision, Recall, F1-score, and mAP@0.5 as criteria to measure the models' performances. Precision is calculated as the ratio of the number of objects correctly detected to the number of objects detected, while recall is the ratio of the number of objects correctly detected to the number of actual objects, as shown in (1) and (2), respectively. From these 2 equations, True Positive is the number of detected objects with the correct class and correct position (same as the label). False Positive is the number of objects that can be detected but are not actual objects (not the same as the label). False Negative is the number of undetected objects (there is a labeled object in the image, but the algorithm fails to detect). Because the precision is high or the recall is high does not guarantee that the model is highly accurate, therefore, we include the F1-score in the measures. The F1-score is the harmonic mean of the precision and recall as shown in (3). The highest possible value for F1 score is 1.0, which means it has the highest precision and recall (everything is correct), and the lowest value is 0.0 when the precision or recall is zero [16]. The mAP@0.5 or Mean of Average Precision at 0.5 IoU (Intersection of Union) threshold can be calculated by averaging the average precision (AP) of all object classes as in (4), where n is the total number of object classes and AP@0.5 can be calculated by averaging the precision at the IoU threshold of 0.5 [17].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1score = 2 * \frac{(Precision * Recall)}{Precision + Recall} \quad (3)$$

$$mAP@0.5 = \frac{1}{n} \sum_{i=1}^{n} AP@0.5_i \quad (4)$$

Table II presents the results of the evaluation of the trained result reading model tested on a test set of 110 images, comprising 80 “Negative” objects, 60 “Positive” objects, and 13 “Invalid” objects. The model gives a precision of 0.982, a recall of 0.958, an F1 score of 0.970, and an mAP@0.5 of 0.986. Fig. 8 shows the Precision-Recall curve of this experiment. The graph represents the trade-off between precision and recall when changing the confidence threshold in object detection. Therefore, if both precision and recall are close to 1, the model can detect objects very accurately.

Table III shows the results of the evaluation of the trained landmarks detection model tested on a test set of 89 images, with 207 “Line” objects, 250 “C” objects, and 152 “T” objects. The model gives a precision of 0.947, a recall of 0.905, an F1 score of 0.926, and an mAP@0.5 of 0.940. Fig. 9 shows the Precision-Recall curve of this experiment.

We tested the distances comparison algorithm with 50 negative results and 50 “Invalid 2” invalid results. The test data were small because we did not have many invalid results. This kind of invalid results does not occur frequently and when they do occur, the patients tend to discard those invalid results and have never taken any photos of them. Because of these reasons, we created invalid results for the experiment by using image manipulation software to remove a C-line from existing positive results. The algorithm gave a 100% accuracy for this experiment.

TABLE II. PERFORMACE OF THE RESULT READING MODEL

Class	Labels	Precision	Recall	F1 Score	mAP@0.5
Negative	80	0.964	0.988	0.976	0.986
Positive	60	0.983	0.964	0.973	0.984
Invalid	13	1.000	0.923	0.960	0.990
Total	153	0.982	0.958	0.970	0.986

TABLE III. PERFORMACE OF THE LANDMARKS DETECTION MODEL

Class	Labels	Precision	Recall	F1 Score	mAP@0.5
Line	207	0.931	0.875	0.927	0.949
C	250	0.980	0.916	0.947	0.954
T	152	0.930	0.875	0.902	0.914
Total	609	0.947	0.905	0.926	0.940

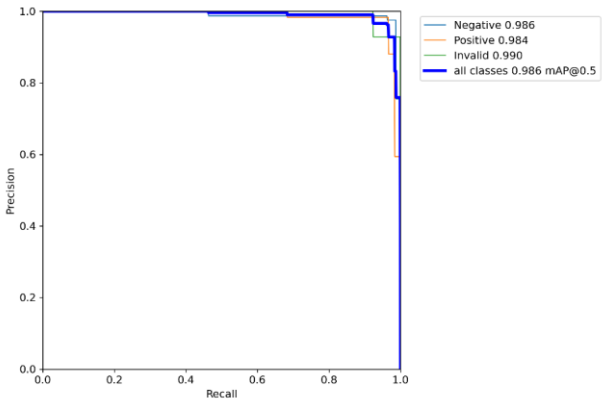


Fig. 8 Precision-Recall Curve

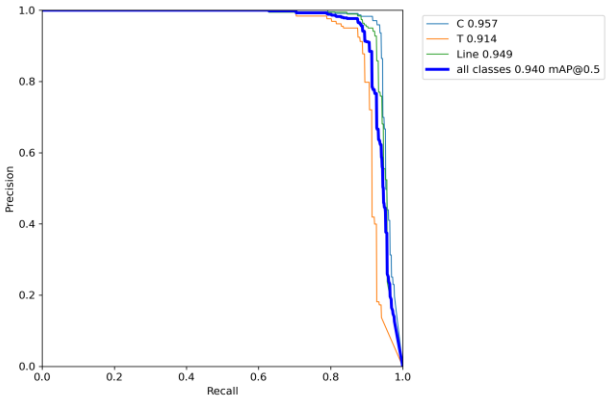
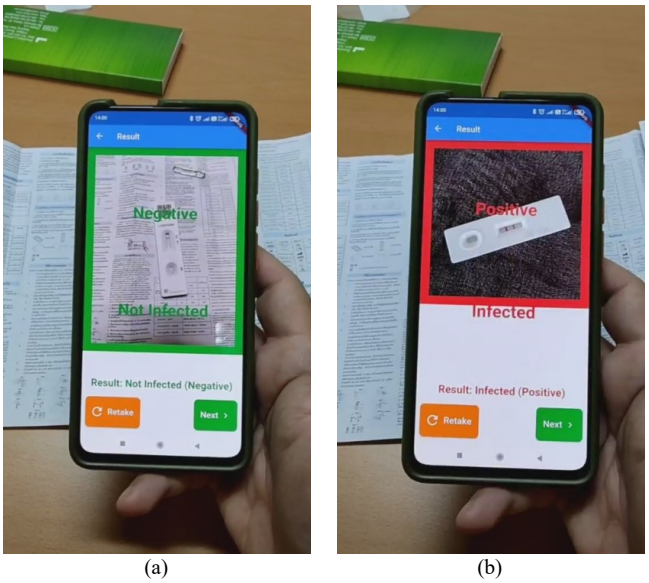


Fig. 9 Precision-Recall Curve

V. DISCUSSION AND CONCLUSION

We have developed a mobile application to read the results of a COVID-19 antigen test kit. The user can choose to take a new ATK image from the phone’s camera or use an existing image. Then the application sends the image to the server. On the server, the object detection module is run to recognize the ATK result from the image and sends the result back to the user’s phone. The object detection model used in this study was trained with YOLOv5. Fig. 10 shows examples of a user using the application with the negative (a) and the positive (b) results.

For the performance of the trained model that we used to recognize the results of the ATKs, it was found that the developed model has high accuracy, with a precision of 0.982, a recall of 0.958, a F1-score of 0.970, and a mAP@0.5 of 0.986. These are higher than other similar works such as [13] and [14], which proposed image processing methods to recognize color stripes on pregnancy test kits with accuracy higher than 80%, and higher than 95%, respectively.



(a) (b)

Fig. 10 Examples of a user was using the application with (a) the negative result, and (b) the positive result.

However, the images used in this study consisted of fewer images of invalid results compared to other classes, i.e., 62 invalid objects out of a total of 1,406 objects in the image dataset. This is because invalid results occur rarely and when they occur, the patients are most likely to throw those invalid kits away and not take photos of them. In particular, the invalid results in which the test line (T) is visible while the control line (C) is not visible as in Fig. 4, "Invalid 2" are very rare and are visually similar to the negative results with the difference only in the direction of the order of the lines. We developed a result refinement algorithm based on the comparison of distances between the distance from the line to the letter C and the letter T on the cassette. However, the test data were limited. For the future development of this study, adding the ability for applications to read multiple test kits at the same time would be useful for organizations, meetings, schools, offices, factories, or events that require testing for COVID-19 before entering the settings.

REFERENCES

[1] Guidelines for screening with Antigen Test Kit (ATK) and testing for COVID-19, (in Thai), Ministry of Health. 2021 [Online]. Available: https://ddc.moph.go.th/viralpneumonia/file/g_health_care/antigen_test_kit_130764.pdf

[2] Roche, "SARS-CoV-2 Rapid Antigen Test. Diagnostics," 2022. [Online]. Available: https://diagnostics.roche.com/global/en/products/params/sars-cov-2-rapid-antigen-test.html

[3] Infectious Disease Society of America, "Rapid Testing," April. 2022. [Online]. Available: https://www.idsociety.org/covid-19-real-time-learning-network/diagnostics/rapid-testing/

[4] Department of Medical Sciences, "Self-test Antigen Test Kit (ATK) for the public.", (in Thai), 2021 [Online]. Available: https://www3.dmsc.moph.go.th/post-view/1243

[5] E. Frew et al., "A SARS-CoV-2 antigen rapid diagnostic test for resource limited settings," *Sci Rep*, vol. 11, no. 1, p. 23009, Nov. 2021, doi: 10.1038/s41598-021-02128-y.

[6] A. W. Martinez, S. T. Phillips, E. Carrilho, S. W. Thomas, H. Sindi, and G. M. Whitesides, "Simple telemedicine for developing regions: camera phones and paper-based microfluidic devices for real-time, off-site diagnosis," *Anal Chem*, vol. 80, no. 10, pp. 3699–3707, May 2008, doi: 10.1021/ac800112r.

[7] H. Zhu, O. Yaglidere, T.-W. Su, D. Tseng, and A. Ozcan, "Cost-effective and compact wide-field fluorescent imaging on a cell-phone," *Lab Chip*, vol. 11, no. 2, pp. 315–322, Jan. 2011, doi: 10.1039/c0lc00358a.

[8] H. Zhu, U. Sikora, and A. Ozcan, "Quantum dot enabled detection of Escherichia coli using a cell-phone," *Analyst*, vol. 137, no. 11, pp. 2541–2544, Jun. 2012, doi: 10.1039/c2an35071h.

[9] J. Matthews, R. Kulkarni, M. Gerla, and T. Massey, "Rapid dengue and outbreak detection with mobile systems and social networks," *Mobile Netw Appl*, vol. 17, no. 2, pp. 178–191, Apr. 2012, doi: 10.1007/s11036-011-0295-5.

[10] N. L. Dell, S. Venkatachalam, D. Stevens, P. Yager, and G. Borriello, "Towards a point-of-care diagnostic system: automated analysis of immunoassay test data on a cell phone," in *Proceedings of the 5th ACM workshop on Networked systems for developing regions*, New York, NY, USA, Jun. 2011, pp. 3–8. doi: 10.1145/1999927.1999931.

[11] O. Mudanyali, S. Padmanabhan, S. Dimitrov, I. Navruz, U. Sikora, and A. Ozcan, "Smart rapid diagnostics test reader running on a cell-phone for real-time mapping of epidemics," in *Proceedings of the Second ACM Workshop on Mobile Systems, Applications, and Services for HealthCare*, New York, NY, USA, Nov. 2012, pp. 1–6. doi: 10.1145/2396276.2396278.

[12] A. Carrio, C. Sampedro, J. L. Sanchez-Lopez, M. Pimienta, and P. Campoy, "Automated low-cost smartphone-based lateral flow saliva test reader for drugs-of-abuse detection," *Sensors (Basel)*, vol. 15, no. 11, pp. 29569–29593, Nov. 2015, doi: 10.3390/s151129569.

[13] K. Manasa *et al.*, "An automated algorithm for the quantification of hCG level in novel fabric-based home pregnancy test kits," in *2013 Asilomar Conference on Signals, Systems and Computers*, Nov. 2013, pp. 245–247. doi: 10.1109/ACSSC.2013.6810269.

[14] K. Abhishek *et al.*, "An enhanced algorithm for the quantification of human chorionic gonadotropin (hCG) level in commercially available home pregnancy test kits," in *2014 Twentieth National Conference on Communications (NCC)*, Feb. 2014, pp. 1–5. doi: 10.1109/NCC.2014.6811289.

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

[16] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania, 2006, pp. 233–240. doi: 10.1145/1143844.1143874.

[17] "Coco - common objects in context." https://cocodataset.org/#detection-eval (accessed May 08, 2022).