‘Antigen Rapid Test’ Image-Processing based Machine Learning Algorithm for ART Buddy
Christopher Hui-Kang Nah 1*, Weiling Wu 1, Samuel Ken-En Gan 1,2,#, Scott Wei-Gen Wong 3,#

1 Antibody & Product Development Lab, EDDC-BII, A*STAR, Singapore 138672
2 APD SKEG Pte Ltd, Singapore 439444
3 An’Rachelle Marketing Pte Ltd, Singapore 596746

*To whom correspondence should be addressed: samgan@apdskeg.com; scott.wong.wei.gen@gmail.com

DOI: 10.30943/2022/28032022 Handling Editor: Nalaka Edirisinghe
Submitted: 26 February 2022 ; Accepted:24 March 2022; Published: 28 March 2022.

ABSTRACT
2021 witnessed subsequent waves of COVID-19 sweeping across the world. As the number of daily cases rose in many countries, many adopted the utilization of antigen rapid test (ART) kits for faster detection and isolation of the infected. However, the accuracy of the ART can be impacted by incorrect usage and self-reporting biases. Despite self-administration, image processing of submitted images could be leveraged for validation. Given the ubiquitous use of the smartphone camera, mobile applications that included features such as user uploading of ART kit result images, facilitate validation by backend servers against incorrect self-reported ART results while improving compliance rates. For this purpose, we describe an algorithm that was incorporated into the ‘ART Buddy’ app for the classification of submitted positive and negative ART images. The algorithm was based on machine learning using the Convolutional Neural Network (CNN) to achieve an accuracy of 97.57%, precision of 79.31%, and recall of 88.46%.

KEYWORDS: COVID-19, image processing, scientific phone apps, ART, machine learning, automation, smartphones

INTRODUCTION
The repeated waves of SARS-CoV2 infections worldwide spurred on various containment measures and most recently, the use of antigen rapid test (ART) kits for rapid detection and monitoring. With the use of less invasive nasal swabbing and results within 15 minutes, it was incorporated as a routine screening tool in some countries. While inaccuracies of test results can result from incorrect usage, mistakes, and biases, these problems could be mitigated using smartphone apps. With smartphone apps commonly used for picture taking with numerous apps that aid in detection (Sim, et al., 2015; Wong, et al., 2016; Poh, et al., 2021; Ng, et al., 2017; Chew, et al., 2021) and the ubiquitous use of smartphone cameras, mobile apps could be the additional layer of verification of test results. With apps being essentially minicomputers with utility in aiding clinical usage (Gan, et al., 2016; Gan & Yeo, 2020), replacing various scientific equipment (Gan & Poon, 2016; Sim, et al., 2015) and with clear utility of smartphones as solutions during the pandemic (Gan, 2020), we utilized image processing leveraging on previous work (Nguyen, et al., 2016; Wong, et al., 2016) to create an algorithm for automated screening and integration into the ‘ART Buddy’ App.

Figure 1: ART Buddy App process

1) Step by Step
2) Take photo and crop
3a) User self reports
3b) AI Verifies backend
‘ART Buddy’ is a progressive web application functionalized into a mobile app for users to follow step-by-step instructions (Figure 1) for Antigen Rapid test (ART) self-testing. At the end of a 15-minute waiting period, users were to submit a photo of the ART kit, and self-report on the results (positive, negative, or invalid). After verification, the user would be directed to the prevailing infection control standard operating procedures. Healthcare policymakers would be able to get an aggregate report of the total number of ART tests submitted, with a breakdown of the various results. However, three main issues were prevalent: 1) mismatch in self-reported results to the ART image results, i.e. false positives and false negatives. This resulted in incorrect reporting of ART testing where users would be directed to the wrong instructions (Figure 2). The second problem was that of submitted photos lacking the ART test kits due to mistakes or other reasons, preventing the verification of the result. This led to a compliance issue with compulsory ART testing regimes, where users were required to declare an ART negative status before admission at events of workplaces. Thirdly, it was very laborious and time-consuming to manually validate the large number of submitted ART test kit images.

To address the abovementioned problems, we created an Image-Processing based Machine Learning Algorithm that could: 1) check if a user had submitted an image of an ART test kit, and 2) verify the results of the ART test. The goal was to reduce the rate of non-ART images submitted and increase the compliance rate of ART testing for users (Figure 2), while also providing an accurate and automated surveillance of COVID en masse.

Utilizing Convolutional Neural Network (CNN), a class of deep learning algorithms that was used to process the input images (Figure 3), we assigned weights and biases to various aspects in the image to train the classifications (Tan, 2019). Integrated into the ART Buddy app, the algorithm reduced the manual verification of the ART images, allowing faster ring-fencing and better support containment measures.
MATERIALS AND METHODS

Positive and negative kit images class labelled and verified by a clinician were used. For a clean training data of negative samples, a pre-trained model was used to remove irrelevant images such as selfies uploaded by users.

To reduce the complexity and size of the input data to the downstream CNN model and to clean up “unclean” raw data (many users would write their initials or date of testing on the ART kit), pre-processing of the images was carried out. For the setting of the precise target for the CNN to learn the relevant features for generalization to real images during deployment, we imported a pre-existing object detection model (EfficientDet-Lite2) from Tensorflow Hub to crop out the result box section of the ART kit consistently and reliably, incorporating image enhancements to highlight the lines of the ART kit.

A cropping screening algorithm using bounding boxes identified by the object detection model was then used to verify, in order of descending scores, the object (result box). The ART kit dimensions were first standardized through resizing. Following this, the “detected object” which satisfied criteria such as the combined, maximum, and minimum height and width dimensions were selected to be prospective “result boxes”. The ratio of the height and width dimensions were then assessed to ensure that the object showed only the “result box” since the proportion of the result box would be fixed regardless of the image.

| Cropped ART kit “result box” | Labelling | Flagging for review (using the original image) |
|-------------------------------|-----------|-----------------------------------------------|
| Positive                      | Invalid result |
| Positive                      | Blank kit |
| Positive                      | Might be photographed from existing picture due to Moiré patterns (interference patterns) |
Positive

Blank kit, and likewise might be photographed from existing picture due to Moiré patterns (interference patterns)

Positive

Oversaturated image

Positive

Object detection model failed to crop

Using this methodology (Figure 4), a total of 3712 images were cropped. The dataset contained 86 positive result images and 67 ‘unclean’ images (e.g., blank, or ambiguous ART results). 15 images were not cropped correctly and another 3 failed to elicit any bounding boxes (from the object detection model) that satisfied the dimension requirements. These ART kits images and those flagged for manual review (using the original image) were labelled as ‘positive’ for the binary classification as having False Positives would be more effective for our usage in pandemic management than False Negatives (Table 1).

To address the lack of positive result boxes, we manually created 92 visually distinct “positive result boxes” by varying the cropping and perspectives of positive result images (Table 2). These source images were selected from those that first satisfied the object detector algorithm requirements to possess a “result box”

These cropped result boxes were then downsampled and standardized to image height and width of (200 x 80) pixels and passed into the CNN model. Additional layers of Keras image preprocessing augmentation layers were added due to the imbalanced input dataset of positives and negatives.

Table 2: The three “positive result boxes” from 1 positive sample

The complete dataset of labelled result boxes (263 positives and 3449 negatives) was then randomly split into training (20%) and validation (80%) datasets for usage during model training. Following this, various combinations of hidden layers, dropouts and kernel regularizes were trialed to obtain important validation metrics such as AUPRC, to prevent overfitting to the training set.

During this determination of model complexity, KerasTuner was also executed to obtain the best hyperparameters for each level of complexity of the defined model to produce the optimum HyperModel. With the aid of recall metrics monitored, it minimized the number of false negatives detected by the model.

When the improvements in validation recall became negligible and plateaued, metrics such as binary cross-entropy and accuracy and precision of the model were evaluated to determine the quality of the deployment model.

RESULTS AND DISCUSSION

The data preparation of using the object detection model coupled with the cropping screening algorithm is as shown in Table 3.
Table 3: Accuracy of image cropping

|                          |       |
|--------------------------|-------|
| Total number of images cropped | 3712  |
| Erroneous cropping       | 18 (3 due to object detection model) |
| Overall accuracy         | 99.51% |

We found image enhancements to improve the contrast of the images especially so for faint lines. Interestingly, some ART kits initially deemed to be single-line negative, were found to possess a positive faint band after image enhancement as shown in Figure 5.

![Image enhancement revealed a line in this ART photo that was then flagged out as “+” by the MARS model/algorithm](image)

Figure 5: Image enhancement of a faint positive line for better detection sensitivity

Thus, the sensitivity of the ART kits were augmented through image enhancement incorporating different types of filters, order of layers, and combination of the values such as that of the color, edge enhancement, contrast, brightness, etc. Tables 4 and 5 show results from the optimal inputs that improved the final quality of the obtained prediction. The training-validation improvements across epochs are shown in the Supplementary Materials.

The images were resized to 200x80 pixels after cropping out the visible results box. Other technical validation metrics of our model are shown in Table 5.

Table 4: Results of True and False Positive and Negatives.

|                  |       |
|------------------|-------|
| True Positives   | 46    |
| False Positives  | 12    |
| True Negatives   | 677   |
| False Negatives  | 6     |

Table 5: Technical validation metrics of the model.

|                          |       |
|--------------------------|-------|
| Loss                     | 0.1519|
| Binary Cross Entropy     | 0.1463|
| Binary Accuracy          | 97.57%|
| Precision                | 79.31%|
| Recall                   | 88.46%|
| Area Under Receiver Operating Characteristic Curve (AUROC) | 0.9610 |
| Area Under Precision-Recall Curve (AUPRC) | 0.8280 |

For prediction, the small dataset of positive ART result images were tested, with results showing correct capturing (Table 6).

Table 6: The detection of positive result images.

| Input images (200x80 pixels) | Actual Result | Result by the model |
|------------------------------|---------------|---------------------|
| Positive                     | Positive      |                     |

With a validation set of 741 images, the model obtained results as shown in Figure 4.
‘True Negatives’ were successfully detected, expected from the dominant negative data available. ‘True Negatives’ (including those with poorer quality) were also successfully detected (Table 7).

Table 7: The detection of negative result images

| Input images (200x80 pixels) | Actual Result | Result by the model |
|-----------------------------|---------------|---------------------|
| Negative                    | Negative      | Negative            |
| Negative                    | Negative      | Negative            |
| Negative                    | Negative      | Negative            |
| Negative                    | Negative      | Negative            |
|        | Negative | Negative |        | Negative | Negative |
|--------|----------|----------|--------|----------|----------|
|        |          |          |        |          |          |
|        |          |          |        |          |          |
|        |          |          |        |          |          |
|        |          |          |        |          |          |

---

Scientific Phone Apps and Mobile Devices

www.scientificphoneapps.com
The ‘False Negative’ results were narrowed to be due to poor image quality or dimensions (Table 8), in which grainy images did not provide expected inputs to the model. For instance, the megapixels of certain phones or optical stabilization was expectedly lower compared to newer smartphones, causing the captured image to be blurred if not due to poor lighting. There is room for improvement for pictures with faint bands, although this can be circumvented with ongoing data monitoring and obtaining more images.

Unexpectedly, we did not detect ‘False Positives’ from the dataset that would be beneficial in not alarming negative individuals. On the other hand, ‘False Negatives’ would be a problem for infection containment, although this was mitigated by the comparison with the self-reports (Figure 2) in the ART Buddy workflow, reducing the chances of the positive cases going through undetected.

### Table 8: The false negative detection of positive result images

| Input images (200x80 pixels) | Actual Result | Result by the model |
|-----------------------------|---------------|---------------------|
| Positive                    | Negative      |
| Positive                    | Negative      |
| Positive                    | Negative      |

**CONCLUSION**

The integrated algorithm into the ART Buddy progressive web app functionalized on the smartphone, performed automated classification of ART images into negative and positive with the use of the inbuild machine learning model. The modal utilizes CNN with hyperparameters optimized and had been demonstrated to be reasonable accurate. This application can allow timely verifications on
ART images and improve the robustness and reliability of the verification systems for pandemic infection controls to verify self-reported results. The ART Buddy web app is available at artbuddy.healthpixel.co

FUNDING
This project was funded by GAP fund I21D1AG025 Mass Rapid Validation Software for ART (MARS) by the Agency for Science, Technology and Research A*STAR Enterprise, Singapore. ART Buddy is wholly owned by An’Rachelle Marketing Pte Ltd.

COMPLIANCE WITH ETHICAL STANDARDS
The images were provided via ART Buddy by Dr. SWGW under An’Rachelle Marketing Pte Ltd with permission.

CONFLICTING INTERESTS
The underlying algorithm is the intellectual property of A*STAR Enterprise and is licensed for ART Buddy as a commercial product. A technology disclosure for the app is filed under EDDC/Z/13227. The APC was waived for this article. The manuscript is handled by an independent member of the editorial board.

AUTHOR CONTRIBUTIONS
CHKN performed the machine learning for the image processing software and provided the draft. WWL assisted in the algorithm training and the writing of the draft. SWGW provided the images and provided the ART Buddy software package for the integration of the algorithm. SKEG oversaw the development of the algorithm and was involved in the process including securing the funding.

REFERENCES
Chew, Z.-Y., Wu, W.-L. & Gan, S. K.-E., 2021. APD LAMP Diagnostic App: Automated Colorimetric Analysis and Documentation. Scientific Phone Apps and Mobile Devices, 7(1), p. DOI: 10.30943/2021/17032021.

Gan, S. K.-E., 2020. “Set My Scientists Free” - Scientific Phone Apps and DIY equipment during lockdowns. Scientific Phone Apps and Mobile Devices, 6(4), p. DOI:10.30943/2020/14042020.

Gan, S. K.-E., Cornelius, K., Phi-Vu, N. & Haw, Y.-X., 2016. An overview of clinically and healthcare related apps in Google and Apple app stores: connecting patients, drugs, and clinicians. Scientific Phone Apps and Mobile Devices, 19 July, 2(8), pp. DOI:https://doi.org/10.1186/s41070-016-0012-7.

Gan, S. K.-E. K. C., Nguyen, P.-V. & Haw, Y.-X., 2016. An overview of clinically and healthcare related apps in Google and Apple app stores: connecting patients, drugs, and clinicians. Scientific Phone Apps and Mobile Devices, 2(8), pp. DOI:https://doi.org/10.1186/s41070-016-0012-7.

Gan, S. K.-E. & Poon, J.-K., 2016. The world of biomedical apps: their uses, limitations, and potential. Scientific Phone Apps and Mobile Devices, 2(6), pp. DOI:https://doi.org/10.1186/s41070-016-0009-2.

Gan, S. K.-E. & Yeo, J. Y., 2020. Editorial: The promises of Microcontroller kits and Smartphone apps for Psychological research. Scientific Phone Apps and Mobile Devices, 6(2), p. DOI:10.30943/2020/06022020.

Ng, K. M. et al., 2017. APD SpectBT: Arduino-based mobile vis-Spectrophotometer. Scientific Phone Apps and Mobile Devices, 5(6), p. DOI:10.1186/2019/s23122019.

Nguyen, P.-V. et al., 2016. Optimal processing for gel electrophoresis images: Applying Monte Carlo Tree Search in GelApp. PubMed, 15(16), p. DOI:10.1002/elps.201600197.

Poh, J.-J.et al., 2021. Spectrophotometer on-the-go: The development of a 2-in-1 UV–Vis portable Arduino-based spectrophotometer. Sensors and
Sim, J.-Z., Nguyen, P.-V. & Gan, S. K.-E., 2015. GelApp: Mobile gel electrophoresis analyser. *Scientific Phone Apps and Mobile Devices*, 5(4), p. DOI:10.1186/2019/23122019.

Tan, V., 2019. *towardsdatascience*. [Online] Available at: https://towardsdatascience.com/understanding-cnn-convolutional-neural-network-

Wong, C.-F., Yeo, J. Y. & Gan, S. K.-E., 2016. APD Colony Counter App: Using Watershed Algorithm for improved colony counting. *Scientific Phone Apps and Mobile Devices*, 5(5), p. DOI:10.1186/2019/c23122019.
SUPPLEMENTARY MATERIALS
