Architectural Design of Medical Test Scanner Machine for Mass Medical Tests Scanning, Results Recording and Visualisation

Josue Uwimana

June 22, 2022

Architectural Design of Medical Test Scanner Machine for Mass Medical Tests Scanning, Results Recording and Visualisation

Anonymous Submission
Anonymous Affiliation

Abstract

In medical laboratories, many tests are taken in order to be analysed for providing better treatment to patients. During the times of rise of epidemics and pandemics, a mass testing approach for screening the spread of diseases in the population is used. Currently, it takes many hours for health workers to manually record tests' results in systems or logbooks. Based on object recognition with computer vision techniques, the OCR engine, and data analysis techniques, an efficient novel approach to design the Medical Test Scanner machine has been successfully achieved. The machine is designed to scan different RDT types for invalid, negative and positive results for a single or many patients at time. The results with the same patient identification number are clustered to be recorded on a single patient, and results with the same test type are clustered to be recorded on respective test types. As the RDT types are not predefined, all the existing and the future RDTs can be monitored due to the machine automatically recording the patient ID, the RDT types and their results dynamically. This machine can be used extensively in laboratories and in times of rise of epidemics and pandemics as it eases the work of health workers in recording and assessing the test results, laboratory tests management, mass testing, and screening the spread of diseases.

Keywords: medical test scanner, optical character recognition, rapid diagnostic test, computer vision, python tesseract

1. Introduction

Across the world, in hospitals, clinics, health centres, and in different medical laboratories, they conduct medical tests or examinations so that the results are based on healing the patients, providing medications to the patients and conducting further research to discover new drugs and other medical procedures. Currently, the rapid diagnostic tests (RDTs) are being used extensively due to their easy to use and easy interpretation of test results as they take a short time to get the results, and it does not require highly skilled staff to interpret the results. They can be used on various types of specimens, including whole blood. Recently, oral fluid tests have been developed that are non-invasive and do not require the use of sharps, making them the great choice to use in laboratories. Specifically, RDTs have a special advantage of being used for on-site or point-of-care testing in case of emergency care and for first-aid purposes.

In medical laboratories, the tests are labelled on the patient identification number with both handwritten or printed text so that the results are recorded on the corresponding patients, and the recording task is done manually in logbooks, diagnostics forms, and in systems or databases. There is no machine currently that processes a huge number of rapid tests using any kind of technology even though there is an increase rate of utilising technology in healthcare service delivery from consultation or diagnosis to surgery as the state-of-the-art of artificial intelligence makes it easier for automated intelligent machines and systems. In periods of epidemics and pandemics, the approach of mass testing the population is used but the time to get the results, recording them, and analysing them in order to screen the spread of the pandemic is critical and those who are tested have to wait for many hours in order to get their results as still the pandemic is spreading.

This paper is contributing to designing the machine that is going to help to do a mass testing, and solve the problem of recording manually the patients' test results in databases, digitising the medical laboratory environment in test results recording, processing, access, management, and visualisation. The results can be accessed with database mining methods in order to be visualised, and be integrated with the current medical systems so that the patients can access their results online or via SMS as usual, without the work of posting results manually by health workers as the test results are explicitly recorded on the respective patient identification numbers labelled on the RDTs.

2. State of the Art

Object detection techniques using deep learning models are potentially capable in handling complex tasks and they have achieved spectacular progress in computer vision with the state-of-the-art of artificial intelligence. Deep learning models for object detection are based on feature learning, contextual information learning, and occlusion handling. Deep learning object detection models[ ] are mainly divided into two categories: (i) two-shot detection models such as R-CNN, Fast R-CNN and Faster R-CNN and their different variants, and (ii) single-shot detection models such as YOLO and SSD. In two-shot detection models, the detection is performed in stages, in the first stage; region proposals are computed and classified in the second stage into object categories. In single-shot detection models, some methods, such as YOLO, SSD, consider detection as a regression issue and look at the image once for detection. The Single Shot Detector (SSD) uses a unified two-part network, the base network leveraging a pre-trained VGG16 network on ImageNet, truncated before the last classification layer to extract high level features, then converting FC6 and FC7 to convolutional layers. In the design of this machine, the SSD is used which seems to be a good choice for real-time object detection as it runs a convolutional network on input images once and computes a feature map, and its accuracy trade-off is also very extremely modest.

In addition to object detection in images, images contain much insightful information that should be needed to be analysed or be used for making a decision using computer vision techniques by extracting information from the images and processing them into meaningful information by key information extraction. Extracting data like text from an image combines text detection and text recognition. The optical character recognition(OCR)[R. Smith.(2007)] engine is the commonly used tool to extract text from images and it has been extensively used in many applications[J. Ji and H. Lv. (2021)][Srivastava, S., Priyadarshini, J., Gopal, S., Gupta, S., Dayal, H.S. (2019)]. Python-tesseract tool extracts both printed and handwritten texts [J. Memon, M. Sami, R. A. Khan and M. Uddin. (2020)]. Apart from extracting features and results from camera footage, the test types can be more than one type and each test type has to have its own records for easing the work of assessing information by health workers. With Pandas, datasets are grouped based on test types and patient identification numbers for reasonable recording of the data and assessment of results purposes.

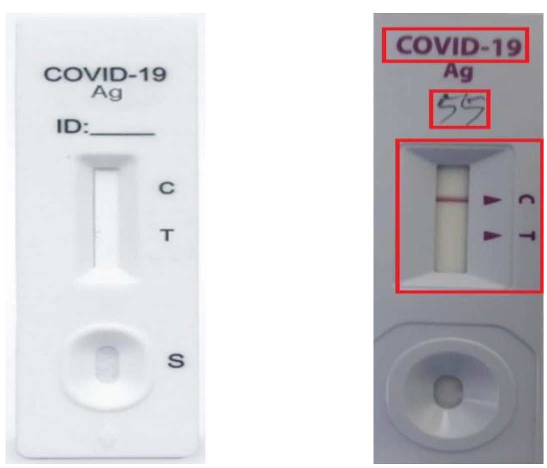

Figure 1: Features to be extracted from RDT (test type [e.g. COVID-19], ID [patient identification number] and test result [positive, negative or invalid])

[Park, Chunjong, et al.] and other researchers developed mobile applications and devices that scan a single RDT and share the information online to specific organisations. This is considered to be a personal use or test by test monitoring which cannot contribute much to the situation of the spread of pandemics where the mass teasing and screening are needed. During times of epidemics and pandemics, those systems and devices are considered to be unreliable. This novel research is going to contribute to solving the global problem of concern of mass testing and recording their results at time and real-time data analysis, scanning and recording dynamically test results of the future RDTs to come.

3. Method

The paper presents the best practices of using computer vision, the text extraction tools, and advanced data analysis techniques in order to design the Medical Test Scanner machine for mass medical tests scanning, results recording and visualisation of results. The work is divided in five key parts:
(1) Object recognition to check for positive, negative and invalid results;
(2) Text extraction and manipulation from image to recognise the RDT types and the patient identification (PID) numbers;
(3) Associate the extracted information with corresponding results with error correction, and creating datasets;
(4) Clustering the patient's test results based on the identification numbers of the patients (for multiple tests of a single patient), and based on test types (when test types);
(5) Recording in a database and visualising patients' test results based on clusters.

3.1. Machine Construction

The proposed machine has two main parts which are hardware and the set of algorithms and models operating the all machine functions. Figure 3 shows the detailed block diagram of the proposed machine including the subset of it, and the main parameters.

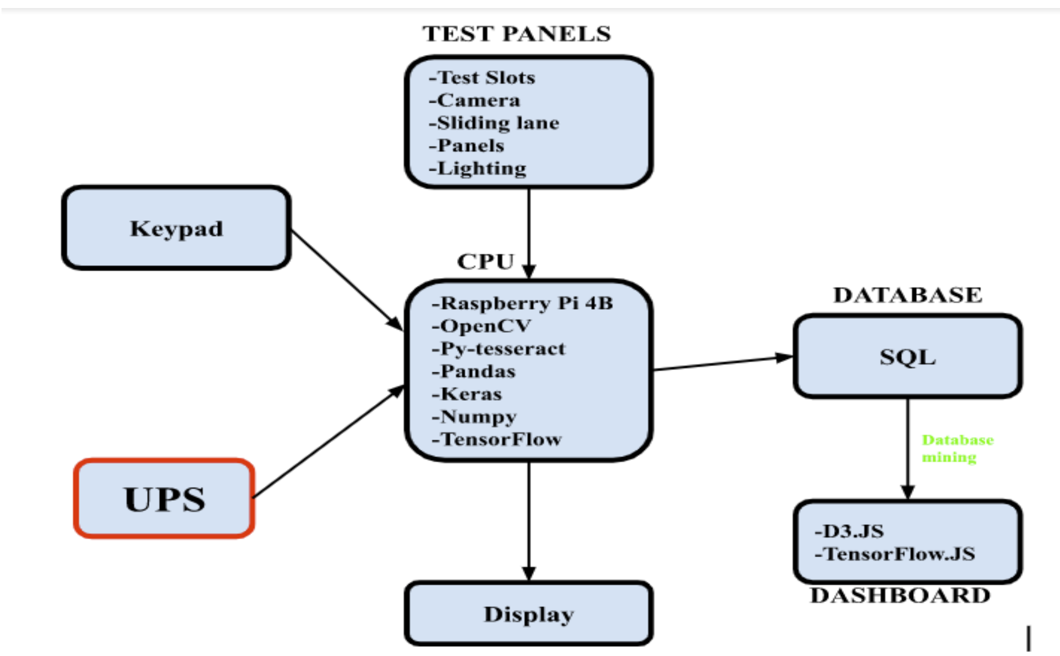

Figure 2: Machine construction block diagram

- **Test panels:** Machine has many panels with many slots to support mass testing. Each panel has a camera attached to a sliding lane for high speed automated scanning tasks.
- **CPU:** All machine functions (scanning, key information extraction & recording), models, and error correction methods are performed here.
- **Keypad:** The keypad is used in order to set up the focus of the camera, the speed of the sliding lane, the lighting system, the number of slots to scan from starting point, and the position of test panels.
- **Display:** It is used for displaying the scanning status, errors, and the overall working status of the machine.
- **Database:** Storing test results of patients, relational database is preferred to store many results on a single patient ID. Each test type has its own database table for results assessment purposes.
- **Dashboard:** We will visualise the test results on dashboard from a database with data mining approaches. TensorFlow.JS and D3.JS library will help to build an interactive real-time dashboard on a web browser.
- **UPS:** Uninterruptible source is needed because the scanning process must not be interrupted to avoid errors in generating final dataframes.

3.2. Overall Workflow Diagram of the Machine

The machine has two main parts including the hardware and the system. The hardware consists of test panels, CPU, and display. The system consists of all computer vision algorithms, models, and tools used for key information extraction and processing. Figure 4 illustrates the working principle of the medical test scanner machine.

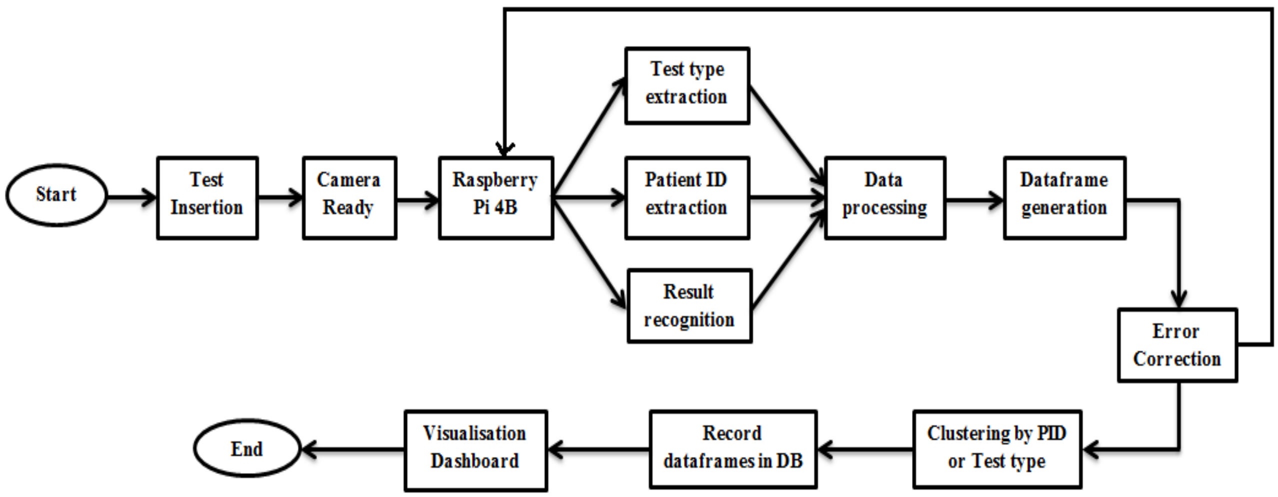

Figure 3: The overall workflow of the Medical Test Scanner machine construction

The workflow diagram shows the working principle of the entire machine and its structure. The raspberry pi is the minicomputer which does all computational, data analysis and decision making actions due to its capability of being able to accept the deployment of deep learning models into it. Python is the main language in developing deep learning models and data analysis. Features extraction is processed smoothly and data wrangling and analysis to understand and classify the extracted features so that the dataframes need are created. In the scanning process, the error may occur to jump a slot while scanning; this can cause inconsistency in creating dataframes. To correct the problem, re-scanning the test panel containing the missing rows is recommend and the empty slots considered to be invalid are supressed in order to create meaningfully dataframes.

Clustering or grouping dataframes based on some criteria is used to separate the dataframe of different test types and assign test results to respective patient IDs. The Pandas, a python tool has to be used for the advanced data analysis with Numpy to aggregate the data.

3.3. Machine Operational Algorithm

The machine operational algorithm describes the workflow process of the system operating on raspberry pi and how the hardware part is working with the overall system. The machine operates in four (4) states: (**i**) One patient, one test type; (**ii**) One patient, multiple test types; (**iii**) Many patients, one test type (mass testing for one disease/epidemic); (**iv**) Many patients, many test types (mass testing for multiple diseases). The operational algorithm of the Medical Test Scanner machine is shown in **ALGORITHM-1**.

```
1. Start machine
2. while machine ready do
3.   Insert RDTs
4.   Wait 15min to get results
5.   Start scanning
6.   while extracting features do --(patient id & test)
7.       Recognise results        --(+ve, -ve, invalid)
8.       Mark extracted features and results
9.       if no result or empty slot
10.          Mark result or feature as invalid
11.      end if
12.    end while
13. end while
14. Camera OFF
15. if test panels are more than one
16.   for each test panel
17.       Get lists of extracted features && results
18.       Merge lists to form a single list
19.       if test type and result are both invalid
20.           Suppress that row
21.       end if
22.       if any row or column is missing
23.           Start camera
24.           Repeat 5-12
25.           Camera OFF
26.       end if
27.       Generate dataframe
28.   end for each
29.   Append dataframes to generate one
30. end if
31. if test types are present more than once
32.   Cluster by test type
33.   Generate dataframe based on clusters
34. end if
35. if same patient IDs are present more than once
36.   Cluster by patient ID
37.   Generate dataframe based on clusters
38. end if
39. Prepare database connection
40. Record dataframes in database accordingly
41. if records successfully
42.   Reject RDTs
43. end if
44. Stop machine
```

Algorithm-1: Operational algorithm of the Medical Test Scanner machine

4. Experimental Results

4.1. Parameters setup

The experiments conducted in this paper are conducted on open source libraries such as Numpy [T.E. Oliphant. (2006)], Pandas [W. McKinney. (2011)], Tensorflow (Google) [M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard. (2016)], Keras A. Gulli, S. Pal. (2017)], and OCR Py-tesseract. Python [G. Van Rossum, F.L. Drake. (2011)] is a high-level general-purpose programming language that can interact with deep learning libraries and the popular wrapper language which powers the Py-tesseract OCR engine. The deep learning object detection model which is SSD trained on customised MobileNet datasets and data analysis techniques used in this paper are built with it.

4.2. Experiment and results analysis

Initially when the machine turns on, it is required to configure some parameters like focus of the camera, speed of the sliding lane (scanning speed), lighting level and other related parameters. Once the RDTs are inserted in the machine, the machine turns in standby mode waiting for results. The camera feeds the footage to Raspberry Pi for results recognition based on transfer learning with an SSD model trained on MobileNet. With OCR python-tesseract tool, the test type and patient ID (PID) are extracted from the footage. List of results and extracted features (PID & test type) are merged to form one dataset. When there is data missing, the scanning process is restarted in order to fix the error.

Table 1. Dataframe generated after features extraction and dataframes merging (first 10 results)

	PatientID	TestType	Results
0	COV0028	COVID-19 Ag	Negative
1	HIV0809	HIV	Negative
2	PGN0028	Pregnancy	Positive
3	HCV0028	Hepatitis C	Invalid
4	COV9118	COVID-19 Ag	Positive
5	COV0809	COVID-19 Ag	Positive
6	COV11920	COVID-19 Ag	Positive
7	COV2822	COVID-19 Ag	Negative
8	COV9760	COVID-19 Ag	Negative
9	HIV9809	HIV	Negative

Now the dataframe is available, the clustering method based on grouping data in pandas can be done by test types and PIDs in order to facilitate recording results by test type or recording results per patient. Referring to Table 1, the two dataframes have been merged to form one main dataframe, the first dataframe contains the patient identification numbers and test type of the test taken and the second dataframe contains the test results. While we have one dataframe contains all information of patients, we can now manipulate data so that we have separate dataframes by clustering or grouping data, each dataframe for test type and the patient IDs are associated with their test information (test type and result). Now, the process of recording results in database is made easy as there are the separate entries.

In order to visualise the results as the all information are stored in database, the database mining methods are used to extract data from database which are processed in pandas as dataframes and are converted to TensorFlow datasets so that we have an interactive dashboard in a web browser with TensorFlow.JS and D3.JS library.

Extracted features, information details and detailed analysis performed are presented on Table 2.

Table 2. (a) Clustered based on patient ID

PatientID	TestType	Results	
COV0028	COVID-19 Ag	Negative	1
COV0809	COVID-19 Ag	Positive	1
	HepatitisB	Negative	1
	HepatitisC	Negative	1
	Pregnancy	Positive	1
COV11920	COVID-19 Ag	Positive	1
COV2822	COVID-19 Ag	Negative	1
COV9118	COVID-19 Ag	Positive	1
COV9760	COVID-19 Ag	Negative	1
HCV0028	Hepatitis C	Invalid	1
HCV2806	HepatitisC	Positive	1
HCV9999	HepatitisC	Invalid	1
HIV0011	HIV	Positive	1
HIV0012	HIV	Positive	1
HIV0013	HIV	Positive	1
HIV0809	HIV	Negative	1
HIV1111	HIV	Negative	1
HIV2009	HIV	Positive	1
HIV9809	HIV	Negative	1
Invalid	Invalid	Invalid	1
PGN0028	Pregnancy	Positive	1
PGN3311	Pregnancy	Negative	1
PGN3312	Pregnancy	Negative	1
PGN3321	Pregnancy	Negative	1
SYP0014	Syphilis	Negative	1

Name: TestType, dtype: int64

Table 2. (b) Clustered by test type

TestType	PatientID	
COVID-19 Ag	COV0028	1
	COV0809	1
	COV11920	1
	COV2822	1
	COV9118	1
	COV9760	1
HIV	HIV0011	1
	HIV0012	1
	HIV0013	1
	HIV0809	1
	HIV1111	1
	HIV2009	1
	HIV9809	1
Hepatitis C	HCV0028	1
HepatitisB	HBV2222	1
HepatitisC	HCV2806	1
	HCV9200	1
	HCV9999	1
Invalid	Invalid	1
Pregnancy	PGN0028	1
	PGN3099	1
	PGN3311	1
	PGN3312	1
	PGN3321	1
Syphilis	SYP0014	1

Name: Results, dtype: int64

Table 2. (c) Split based on test type-Hepatitis C

	PatientID	TestType	Results
3	HCV0028	HepatitisC	Invalid
17	HCV9200	HepatitisC	Negative
18	HCV2806	HepatitisC	Positive
19	HCV9999	HepatitisC	Invalid

Table 2. (d) Split based on test type-Pregnancy

	PatientID	TestType	Results
2	PGN0028	Pregnancy	Positive
10	PGN3099	Pregnancy	Positive
11	PGN3311	Pregnancy	Negative
12	PGN3312	Pregnancy	Negative
13	PGN3321	Pregnancy	Negative

The extraction of features has been made separately, the test results, and the PID with test type which involves advanced data analysis techniques performed in Pandas. During the training of models and making data analysis, it is proven to have high performance but has limitations. Some RDTs have some irrelevant information printed on it, during the text extraction process as the test types are not predefined; it causes a problem to extract meaningless information so that they can not be classified as test type or PID. The machine works well in extracting features on the RDTs of the type presented in section 2, Figure 1. In order to visualise the data stored in the database, data mining technique is used and the dataframes are converted to datasets with TensorFlow.JS so that we have their access in a web browser. The D3.JS library is a JavaScript library used to build an interactive dashboard to display real-time data from a database.

5. Conclusion

The author proposed the design of the medical test scanner machine based on computer vision techniques. The training of models and data analytics have been performed in Google Colab. The use of this machine will ease the work of health workers in their daily work in laboratories as their task will be to take exams and insert RDTs in the machine only. In the rise of epidemics and pandemics, the machine can be used for mass testing and screening the spread of diseases. This novel research contributes to the level that the machine will keep being used for any RDT that will be produced for any disease to break out in the future due to its dynamic extraction of test type non-predefined. The author recommends the manufacturers of RDTs to consider the structure of RDTs used in training of the machine during their design in manufacturing the RDTs as the best choice for the better performance of the machine because they have only relevant information on the scanning side.

To improve the efficiency of the machine in future works, natural language processing algorithms will be used for natural language understanding purposes so that the machine will have the ability to understand the test type to extract and easily distinguish it from the patient identification number.

References

[R. Smith. (2007)] R. Smith, "An Overview of the Tesseract OCR Engine", *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.

[J. Ji and H. Lv. (2021)] J. Ji and H. Lv, "Design of Electronic Device Life Test Monitoring System Based on Optical Character Recognition Technology", *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2021, pp. 665-669, doi: 10.1109/ICAIBD51990.2021.9458980.

[Srivastava, S., Priyadarshini, J., Gopal, S., Gupta, S., Dayal, H.S. (2019)] Srivastava, S., Priyadarshini, J., Gopal, S., Gupta, S., Dayal, H.S. (2019). "Optical Character Recognition on Bank Cheques Using 2D Convolution Neural Network." In: Malik, H., Srivastava, S., Sood, Y., Ahmad, A. (eds) Applications of Artificial Intelligence Techniques in Engineering . Advances in Intelligent Systems and Computing, vol 697. Springer, Singapore. https://doi.org/10.1007/978-981-13-1822-1_55

[J. Memon, M. Sami, R. A. Khan and M. Uddin. (2020)] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)", in *IEEE Access*, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[T.E. Oliphant. (2006)] T.E. Oliphant, "A Guide to NumPy", *Trelgol Publishing USA2006*.

[W. McKinney. (2011)] W. McKinney, "pandas: a foundational Python library for data analysis and statistics", Python High Performance Sci. Comput., 14 (2011), pp. 1-9

[M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard. (2016)] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, "Tensorflow: A System for Large-Scale Machine Learning", 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) (2016), pp. 265-283

[A. Gulli, S. Pal. (2017)] A. Gulli, S. Pal, "Deep Learning with Keras", Packt Publishing Ltd (2017)

[G. Van Rossum, F.L. Drake. (2011)] G. Van Rossum, F.L. Drake, "The python Language Reference Manual", Network Theory Ltd (2011)