Digitization of Perioperative Surgical Flowsheets

A Technical Report for SYS 4054

Presented to the Faculty of the School of Engineering and Applied Sciences

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Systems Engineering

By

Nathan Ohene

Technical Project Team Members

Victoria Rho

Rex Focht

Angela Yi

Bhavana Channavajjala

Luke McPhillips

Sarah Winston Nathan

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Digitization of Perioperative Surgical Flowsheets

Victoria Rho
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
vr2by@virginia.edu

Angela Yi
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
ay2ug @virginia.edu

Bhavana Channavajjala
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
bc8ze@virginia.edu

Luke McPhillips
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
lrm3km@virginia.edu

Sarah Winston Nathan
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
swn2bf@virginia.edu

Rex Focht
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
rwf2cb@virginia.edu

Nathan Ohene
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
no9ga@virginia.edu

William Adorno
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
wa3mr@virginia.edu

Marcel Durieux
Anesthesiology Department, School of Medicine
University of Virginia
Charlottesville, Virginia
med2p@virginia.edu

Donald Brown
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, Virginia
deb@virginia.edu

***Abstract*— Perioperative mortality rate (POMR) is a metric widely used to describe the quality of treatment in hospitals. Perioperative data, or data collected during surgery, can be used to calculate POMR and determine factors that lead to adverse surgical outcomes. Access to such data is essential for decreasing POMR and improving medical treatment. In low- and middle-income countries (LMICs), perioperative data is often manually recorded on paper flowsheets. While these flowsheets capture essential information, their non-digital format leads to difficulty in analysis of perioperative data, as aggregating data and observing trends is a time-consuming and tedious task. The goal of this project is to design a system to digitize the information contained in surgical flowsheets that have been in use for six years at the University Teaching Hospital of Kigali in Rwanda. To accomplish this goal, the research team has done the following: 1) Designed a wooden scanning structure, SARA (Scanning Apparatus for Remote Access), to capture flowsheet images in a standard format, 2) Developed a web application to upload images and securely transfer them to UVA for processing, 3) Developed image processing programs to digitize medication, blood pressure, heart rate and logistical data, and 4) Created a PostgreSQL database system to store the digitized flowsheet data. Additional testing and validation of this system is needed to evaluate the accuracy of each processing technique in the fully integrated system.**

***Keywords* - Machine learning, Artificial Intelligence, Database, Image processing, perioperative mortality**

I. INTRODUCTION

Hospitals in Rwanda use handwritten flowsheets to collect patient data before, during, and after surgery. These flowsheets store essential information; however, their non-digital format makes data analysis difficult. Perioperative mortality rate (POMR), calculated from data stored in these hospital flowsheets, is a metric used to evaluate quality of surgical care [7]. Due to the handwritten format of the flowsheets, hospitals cannot easily calculate POMR or other useful metrics. Hospitals without digital records are at a disadvantage because they cannot easily aggregate surgical data and observe trends. Thus, a method to digitize handwritten hospital records will allow these hospitals to calculate POMR and other relevant metrics, which in turn, will lead to improvements in surgical outcomes.

The system designed by this research team will aid the digitization of perioperative data in low- and middle-income countries (LMICs), allowing hospitals to store data in a cohesive format and analyze surgical data more easily. This paper details the design of the system used to digitize information within the intraoperative flowsheets used in the University Teaching Hospital in Kigali, Rwanda. Hospitals in other LMICs can also adjust and apply the methods set forth in this paper to digitize their own hospital records. This paper details the digitization process in seven distinct parts: the scanning apparatus, the web application, image cropping, checkbox detection, graph reading, text interpretation, and the integration of these individual systems. We then report and discuss the results of the system.

II. PRIOR WORK

To adopt an electronic medical record (EMR) system, many hospitals around the world are actively converting from a paper-based work environment to paperless electronic records [11]. However, this is not completely possible in LMICs, where financial constraints compel hospitals to continue recording patient data on paper. This makes data difficult to access and aggregate for research purposes. POMR is a common metric used to assess the performance of a hospital and can also be reported to agencies such as the World Health Organization to receive funding for better hospital facilities. POMR and related metrics are most effectively calculated when data has been digitized, so that the data can then be analyzed and visualized.

Researchers at Vanderbilt University explored an avenue for digitizing patient data without disrupting doctor workflow by introducing an electronic data collection system in a tertiary hospital in Kenya. The researchers created an offline electronic data collection system, using the Research Electronic Data Capture (REDCap) tool as a template, to allow staff to collect data at the point-of-care. This development allowed doctors and nurses to shift from manual data collection to electronic collection. POMR was then calculated from collected data and reported for each type of surgery conducted by the hospital [2]. While their approach was effective at the tertiary hospital in Kenya, low- and middle- income countries often have varying levels of resources, making it difficult to implement the same technology across countries.

While image scanners and devices have existed for at least 60 years [19], LMICs lack the support needed for technologies like these to function. In a personal conversation with Dr. Durieux, an anesthesiologist at UVA who works with the University Hospital in Kigali, noted the lack of resources and/or education when handling scanner technologies, making it difficult to use a technology-heavy solution [3][12]. Therefore, hospitals in these areas continue to use paper copies of flowsheets to keep track of perioperative data.

A 2017 study conducted in hospitals in Ghana revealed criteria for determining how ready hospitals were to digitize medical records. The criteria were derived from interviewing healthcare workers at Mampong-Ashanti Municipal Hospital. The workers were primarily concerned about: internet connectivity, inadequate power supply, lack of information/communication technology, lack of knowledge required to use technology, financial issues and resistance to new technology [22]. The same concerns were voiced by doctors at University Teaching Hospital in Kigali, Rwanda, informing the decision to design a digitization system capable of operating with existing technology in Rwanda without requiring substantial employee training and internet connectivity.

Since the 1980s, work has been done to teach computers to recognize handwriting. The technology really reached viability after 2009 when deep learning neural networks were developed that could utilize high computer processing power and large sets of examples to learn what human handwriting looked like [4]. Many effective open source handwriting recognition software such as Tesseract exist [24]. Researchers at the Rajiv Gandhi Institute of Technology were able to use Tesseract to recognize handwritten characters in images with up to 98% accuracy [14].

Overall, data digitization methods have been researched for a wide range of purposes. Techniques for recognizing handwriting, scanning sheets, and reading graphs are not easily applicable to this case because of the unique constraints LMICs have. Our work builds on these basic techniques, but focuses on how these constraints can be overcome by building an adaptable system. By testing the effectiveness of these techniques, the team has concluded to build upon methods to fit the needs as relayed in sections (A) through (F) in the next portion.

III. SYSTEM DESIGN

Most medical equipment in the University Teaching Hospital of Kigali in Rwanda lacks standardization between different operating rooms. The hospital also lacks the infrastructure and access to sufficient technical support to maintain an electronic data collection system. Additionally, the hospital has years of past handwritten data records that require digitization along with the new flowsheets that are created daily [12]. Considering these factors, the research team designed a system that digitizes handwritten flowsheets while accounting for the capabilities of the hospital.

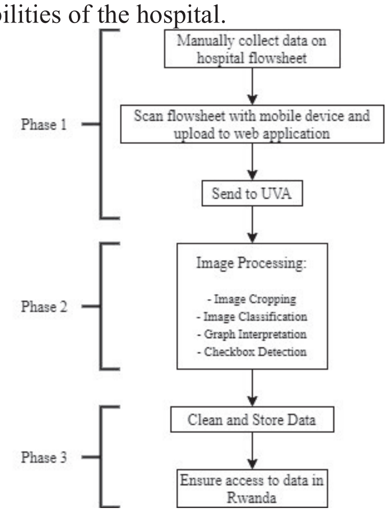

Fig. 1. Process flow diagram

The overall system works in three distinct phases: 1) uploading patient flowsheets in Rwanda and sending them to the University of Virginia, 2) processing flowsheet images at the University of Virginia, and 3) sending data back to Rwanda. These phases are detailed in Figure 1. In the first phase, the user operates SARA to scan handwritten flowsheets with a mobile device and then uploads the image to a web application, which sends the image to a UVA email address. Phase two integrates various image processing techniques. First, new flowsheet images are extracted from the UVA email inbox, along with a unique patient identifier and the email timestamp. Once an image is extracted, the program segments the flowsheet image into individual data pieces, which are then processed according to the type of information contained—either handwritten text, a graph, or a checkbox. The algorithm extracts data from the image and populates a backend PostgreSQL database with the data, using the patient ID and email timestamp as identifiers. The PostgreSQL database is hosted on a server at the University of Virginia. The system is currently in the final stages of integration. While the image processing techniques have been evaluated individually, future testing will determine how accurately they digitize the flowsheet as a whole.

A. Scanning Apparatus for Remote Access (SARA)

The need for a robust scanning apparatus was made clear following discussions with Dr. Durieux. Relying on electronic devices, even commercial scanners, is not feasible in Rwanda due to the lack of resources available to support technology-heavy solutions. Therefore, the scanning apparatus must be easily maintained and assembled, while still producing scanned flowsheets that are as consistent as possible [12]. The Scanning Apparatus for Remote Access (SARA) shown in Figure 2, allows the user to scan flowsheets using a mobile device that rests on top of SARA.

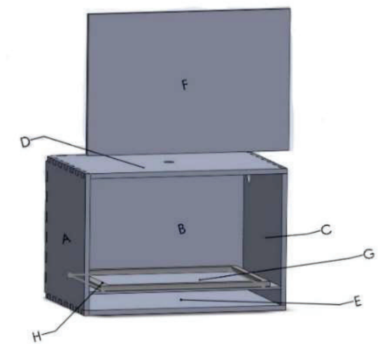

Fig. 2. 3D image of SARA and parts labeled

SARA is built using plywood (easily assembled and maintained) and the design ensures consistent lighting and allows for focal height adjustments. Five (A-E) of the seven pieces that constitute SARA are finger-locked, while the sixth piece F is hinged. Piece C has a hole to accommodate a consistent light source (e.g., lightbulb) while the top (D) has a hole in the center for mobile device camera placement. On the inside of the box, a few screws exist for the placement of a tray (G) for the flowsheet to sit on. These screws, not shown in Figure 2, are placed inside the box for focal height adjustment depending on the mobile device that is being used. Reference lines, located on the tray that sits inside the box, require the flowsheet to be placed in the same area each time a user scans an image. Finally, a hinged frame (H) holds the sheet flat in place via a magnet. To use SARA, the user would first place the mobile device on top so that the camera of the mobile device can focus on the flowsheet in the apparatus. They would then lift the hinged frame (H), place a flowsheet onto the tray (G) in the reference lines drawn, and lower the hinged frame (H) until placed onto the tray (G). The user would then close SARA using piece F, switch on the light source (not shown Figure 2), and adjust the mobile device resting on top (D) for a clear image of the flowsheet. The application Tiny Scanner is used to capture the scan and detect the edges of the paper [18]. The user will then need to ensure the paper has been captured to the edges of the flowsheet, and adjust accordingly as shown in Figure 3. Once this has been done, the scanned flowsheet can be saved to the device with a specific naming convention, and will be uploaded to the web application for image processing to begin.

B. Web Application

Transferring scans taken on the mobile device requires a secure, dedicated medium. A web application provides a customizable platform for future design iteration, and supports proper image encryption procedures and a simplified user experience. Built using the Flask framework in Python on an HTML base, the app is accessed from the user's device in Rwanda. To support data anonymization, users input patient medical record numbers (MRNs) into a provided app-independent Microsoft Excel sheet saved locally to generate randomized identification numbers (RIN). Patient RINs enable data to be routed through the appropriate following technical processes without compromising the rights of the patient. The user logs on to the app using system-approved identification and has the ability to upload an image file from their local device or transmit further commentary on a specific patient. Both actions require entering the corresponding RIN before information can be sent. All data is encrypted using an AES cipher, then sent via email to a designated UVA address. Transmitted emails include patient RIN, inputted data (file or commentary) from the user, and timestamp. Discussed further in sub-section G, received emails at UVA are automatically extracted on timed intervals from the inbox for processing.

C. Image Cropping

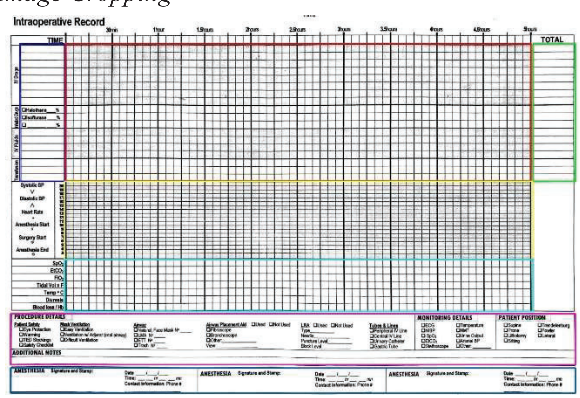

Fig. 3. Intraoperative Record Flowsheet Sections

The flowsheet contains various types of information, including handwritten text fields, graphs, and checkboxes. In order to digitize the flowsheet data, it is necessary to determine where individual data pieces are located on the flowsheet so that each data piece can be processed separately using the appropriate method for that type of data.

To process individual pieces of data, the image cropping algorithm segments the original flowsheet into separate images. The program first determines which side of the flowsheet (Intraoperative Record vs. Anesthesia Record) is being processed via a simple black pixel ratio check of the top left corner of the flowsheet. The program then aligns the patient image to a standard image of either the blank Intraoperative Record or Anesthesia Record flowsheet. It determines feature points in the two images and matches features in one image to features in the other [23]. Then, the program aligns the patient image to the standard image based on the strongest matching features detected. Next, the program uses Hough Lines to detect the outside border of the chart area on the flowsheets [1]. The

program uses these edges as reference lines and determines the pixel locations of the different flowsheet sections, denoted by the colored lines in Figure 3.
The program locates the specific data pieces within these sections by their location on the standard flowsheets relative to these reference lines. The patient flowsheet image is then cropped into individual images at the pixel values of each data piece so that the appropriate image processing technique can process each image separately.

D. Checkbox Detection

On the Intraoperative Record, checkboxes are used to indicate procedure details, monitoring details and patient position during the surgery. The checkbox detection algorithm isolates the checkbox from its associated text using template matching with a set of 6 different templates [10]. In some cases, checkboxes were detected in samples even if the box was partially cropped. Checkboxes were also incorrectly detected in the text associated with the box. In order to eliminate these errors, the thresholds for matching templates to samples were adjusted to reject samples with partial checkboxes and to accommodate for samples where the full checkbox could not successfully be isolated. If a checkbox cannot be effectively isolated from the provided image even with adaptive thresholding, the checkbox is flagged for further review.

Successfully isolated checkboxes are converted into a matrix of pixels, where each element is a grayscale value between 0 and 255. If 34% of the pixels are above a blackness threshold of 115, the box is deemed as checked. The blackness threshold and associated proportion of pixels above the threshold were chosen using a grid search method.

E. Graph Reading

The graph on the intraoperative side of the flowsheet consists of symbols that represent measurements of heart rate, diastolic blood pressure, and systolic blood pressure taken at five-minute intervals. Handwritings can vary from person to person. The large variation in how different individuals draw the same symbol limits the effectiveness of image processing techniques such as template matching or shape recognition. Template matching is sensitive to rotation and scale changes [16], while it is difficult for shape recognition techniques to deal with overlapping objects [18]. Template matching is not an ideal approach because it is difficult to manually select templates that represent or generalize the infinite ways a symbol can be hand-drawn by different people. In the past few years, deep Convolutional Neural Networks (CNNs) have outperformed the state-of-the-art in many visual recognition tasks [15]. Graph reading is accomplished with a U-Net model architecture which is a CNN adaptation that was originally designed for biomedical image segmentation [17]. The U-Net model is trained with truth data to detect the location of each symbol on the graph. The truth data was obtained by manually annotating the symbol locations in a set of example graph images. In Figure 4, the top image shows the scanned patient graph which includes the symbols that were originally recorded by hand. The bottom image shows the predicted masks of the three symbols.

The predicted masks are post-processed into numerical time series data since the locations of symbols in the image are directly related to the values on the graph. The heart rate symbols are very small, so the location is obtained by finding the centroid of all interconnected objects in the heart rate mask. The centroid locations are then aligned with the 5-minute time series interval. The blood pressure symbols are larger objects, so values are obtained by iterating through the image at each time step. The collected measurements for heart rate and blood pressures are converted to the proper scale from the graph and then stored in the database.

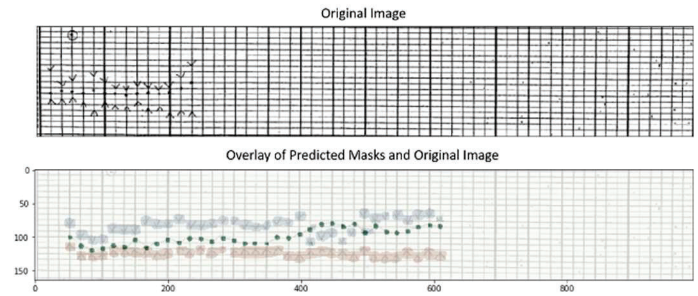

Fig. 4. Original Handwritten Graph and an Overlay of the Predicted Masks

F. Text Interpretation

Within the Intraoperative Record, a set of intravenously administered drugs and medications are listed in handwritten text providing information regarding the specific medication administered, the dosage of the medication, and the corresponding time series data regarding the surgery being conducted. Despite the accuracy that others have observed from using open-source software to recognize handwriting, we could not observe the same results due to the varied standardization of the flowsheet images and the characteristically rushed handwriting of doctors within them. Therefore, we built out an ensemble convolutional network from 3 different convolutional neural networks. When identifying and classifying the medications names, segmented images from the flowsheet containing the handwritten text are read into a set of three convolutional neural network (CNN) architectures (VGG-19 [8], ResNet50 [9], and SimpleHTR [5]), which jointly form an ensemble convolutional network that accounts for various edge cases within the handwritten text, thus increasing the threshold of accurately identified medications.

This Keras enabled ensemble addresses the variability within each handwriting style and precise location of each administered drug within the flowsheet in an attempt to minimize the rejection set of medications that cannot be identified [21]. As a subset of medication and drug classification, dosages for each medication were identified employing a digitization method modeled after the MNIST digit classification also utilizing a constructed convolutional neural network [6]. Time series markings were then identified on the corresponding graphical portion of the flowsheet using CNN model architecture. The outlined methodology shows the ability to digitize and provide information regarding specific drugs and fluids administered intravenously during surgeries.

G. *Integration*

Each image processing function is integrated into the cropping program. When the original flowsheet is segmented into separate images, the appropriate image processing function is called on its respective section of the flowsheet and returns the information contained within it. The diagram in Figure 5 demonstrates how the whole system works together.

First, the program extracts a flowsheet from a folder containing all of the unprocessed flowsheets. Then, each flowsheet is segmented into individual data pieces. Depending on the information contained, each data piece is processed through the checkbox detection program, the graph reading algorithm, or the text interpretation program. Each of these programs extracts data from the data piece, and uploads this data to the appropriate table in the database. Additional testing and validation are needed before this system can be implemented in Rwanda.

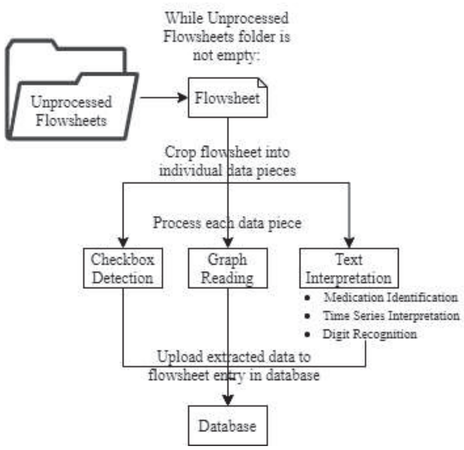

Fig. 5. Integration Diagram

IV. RESULTS AND DISCUSSION

Text identification processes utilized 276 images of handwritten text of medication names and yielded a testing accuracy of 90.21%. The ensemble convolutional network rejected 41 images, 14.9% of the sample, due to varied unreadability and low thresholds of accuracy in classification. Most rejections within the set were due to variance in the handwriting style of the medication images. The ensemble, which utilized the VGG-19 architecture, ResNet50 architecture, and flagged reviews within the SimpleHTR CNN, decreased the number of tested cases classified in the rejection set that were previously rejected by the individual CNN models during evaluation.

Given a testing set of 838 samples, the checkbox detection algorithm yielded an accuracy of 82.2%, while rejecting 9.9% of samples due to unreadability. 17.8% of checkbox samples were incorrectly classified. Among the incorrectly classified checkboxes, false negatives occurred when the proportion of pixels above the blackness threshold was under 34% for checked boxes. Similarly, false positives occurred when the proportion of pixels above the blackness threshold was over 34% for unchecked boxes. This problem was addressed by flagging the checkboxes with thresholds of 34% ± 5% for review in the database, allowing doctors to manually review such boxes. Similarly, checkboxes in the rejection set were also flagged for review by doctors, who can manually verify if the boxes are checked. Ultimately, the accuracy yielded by the checkbox detection program indicates that it is able to classify a large majority of readable checkboxes properly. While the output still requires oversight from doctors, the algorithm is functional in a research environment. Future work can explore implementing a neural network to classify checked and unchecked checkboxes at a higher accuracy.

The image segmentation model was trained with 23 images, a validation set of 4 images, and a test set of 2 images. Graph reading was then evaluated with 31 images. A miss is recorded when one of the following two criteria is met: 1) there is no symbol on the original scanned image, but the algorithm predicted that there is a symbol at the specific time; 2) there is a symbol on the original scanned image, but the algorithm did not predict that there is a symbol at the specific time. Using the symbols detected, the performance of the algorithm is evaluated with the mean squared error (MSE). MSE measures the average of the squares of errors, where errors are the difference between the actual measurement on the graph and the measurement predicted by the algorithm. A smaller MSE indicates that the algorithm has better performance. The U-Net model graph reading results are summarized in Table 1, with the results from a template matching technique [20] in parentheses. The U-Net approach has a lower missing rate for all three symbols compared to the template matching technique. For the points that were not missed, the accuracy of the U-Net approach is much higher than that of template matching for heart rate, and comparable for diastolic blood pressure and systolic blood pressure, as evidenced by the MSE values in Table 1. Deep learning performs better than image processing, because it utilizes over one million parameters to generalize the features of each symbol. A human can only provide a limited number of template alternatives for template matching. Additional training data and further improvements to the time series post-processing can decrease the missing rate and MSE for all symbols. Precision of the measurements are limited by the equipment in the hospital and the manual recordings of doctors. Thus, the results given by the graph reading algorithm are sufficient for preliminary digitization of heart rate and blood pressure measurements.

Symbol	Sample Size	Missing Rate	Mean Squared Error
Heart Rate	636	3.6% (15.9%)	3.28 (26.0)
Systolic Blood Pressure	610	6.2% (19.0%)	24.41 (27.5)
Diastolic Blood Pressure	597	8.5% (13.7%)	18.48 (20.0)

Table 1. Summarized results from graph reading algorithm as compared with a template-matching algorithm indicated in parenthesis

V. CONCLUSIONS AND FUTURE WORK

In this study, we were able to achieve our objective of digitizing perioperative flowsheets from University Teaching

Hospital in Kigali, Rwanda. We used SARA to capture images of flowsheets without light obstruction and uploaded the images to a web application to be encrypted and delivered to UVA. The processing software at UVA identified and cropped the various parts of the flowsheet, as needed for the individual image processing functions. The medication detection algorithm recognized administered medications at a 90.21% accuracy, while the checkbox detection program recognized checked and unchecked boxes with an accuracy of 82.2%.

Similarly, the graph reading algorithm recognized the symbols with a MSE of 3.28, 24.41, and 18.48 for heart rate, systolic blood pressure, and diastolic blood pressure respectively. The output of the various functions was written directly to a PostgreSQL database for future research use. The results of our study indicate that the system can be fully utilized for research purposes. Additional testing should be done before implementation in Rwanda to further validate the accuracy of the digitization. This system will then allow doctors in Rwanda to maintain their current workflow by providing a simple method for scanning and uploading flowsheets to be processed.

Future work can explore implementing this system in other hospitals in LMICs that use different flowsheets than the University Teaching Hospital of Kigali. We would like to explore building a more robust scanning system with durable material rather than plywood. By modifying the web application to be a mobile application, these processes can also be deployed locally in Rwanda. The applications and programs used in these processes could be improved to increase the accuracy and efficiency of phase two of the system design.

Acknowledgment

We would like to thank the following people who have helped the team complete this project - Dr. Benjamin Lobo for his expertise and guidance for the duration of this project, Dr. Christian Ndaribitse, for being our point of contact at University Teaching Hospital in Kigali and for his input on the design of our system, Mr. Sebring Smith, for cutting and construction of SARA, and Mr. Aman Shrivastava, for giving guidance on image processing techniques, and other advisors.

References

[1] A. Mordvintsev and A. K., "OpenCV-Python Tutorials," OpenCV, 2013. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html.

[2] B. Sileshi, M. W. Newton, J. Kiptanui, M. S. Shotwell, J. P. Wanderer, M. Mungai, J. Scherdin, P. A. Harris, S. H. Vermund, W. S. Sandberg and M. D. McEvoy, "Monitoring Anesthesia Care Delivery and Perioperative Mortality in Kenya Utilizing a Provider-driven Novel Data Collection Tool," Anesthesiology, vol. 127, no. 2, pp. 250-271, 2017.

[3] Chen, K., Kannan, A., Yano, Y., Hellerstein, J.M., Parikh, T.S. Shreddr: pipelined paper digitization for low-resource organizations. In Proc. of ACM DEV, 2012.

[4] D. Keysers, T. Deselaers, H. A. Rowley, L. Wang, and V. Carbune. "Multi-Language Online Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine intelligenc*e, Vol. 39, No. 6, June 2017.

[5] H. Scheidl, "Build a Handwritten Text Recognition System using TensorFlow," *Medium*, Jan. 05, 2019. https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5.

[6] J. Brownlee, "Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras," *Machine Learning Mastery*, 12-Sep-2019. [Online]. Available: https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/.

[7] J. L. Rickard, G. Ntakiyiruta and K. M. Chu, "Associations with Perioperative Mortality Rate at a Major Referral Hospital in Rwanda," *World Journal of Surgery,* vol. 40, no. April, pp. 784-790, 2016

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015. Available: http://arxiv.org/abs/1512.03385.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015. Available: http://arxiv.org/abs/1409.1556.

[10] L. S. V. Thomas and J. Gehrig, "Multi-template matching: a versatile tool for object-localization in microscopy images," *BMC Bioinformatics,* vol. 21, no. 44, 2020

[11] M. C. Freeman, A. P. Taylor, and J. U. Adelman, "Electronic Medical Record System in a Headache Specialty Practice: A Patient Satisfaction Survey," *Headache: The Journal of Head and Face Pain*, vol. 49, no. 2, pp. 212–215, 2009.

[12] M. Durieux, Interviewee, [Interview]. December 2019.

[13] M. Kulikova, "Shape Recognition for Image Scene Analysis," *Human-Computer Interaction [cs.HC].* 2009. https://tel.archives-ouvertes.fr/tel-00477661v2/document

[14] M. Shah, S. Mehta, P. Mody, A. S. Roy, S.P. Khachane "Handwriting Recognition of Diverse Languages" *International Journal of Computer Science and Mobile Computing,* vol. 7, no. 4, pp. 109-114, 2018.

[15] N. Hashemi, R. Aghdam, A. Ghiasi and P. Fatemi, "Template Matching Advances and Applications in Image Analysis," *American Scientific Research Journal for Engineering, Technology, and Sciences*

[16] O. Ronneberger, "Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation," *Informatik aktuell Bildverarbeitung für die Medizin 2017*, pp. 3–3, 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *SpringerLink*, 05-Oct-2015. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28.

[18] *Privacy Policy for Tiny Scanner for Android*. (2019, December 6). Retrieved from Appxy: http://www.appxy.com/scannerprivacy/

[19] R. A. Kirsch, "SEAC and the Start of Image Processing at the National Bureau of Standards," *IEEE Annals of the History of Computing,* vol. 20, no. 2, pp. 7-13, 1998.

[20] R. Dufour, E. Miller, and N. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1385–1396, 2002.

[21] S. Arora and M. P. S. Bhatia, "Handwriting recognition using Deep Learning in Keras," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida (UP), India, 2018, pp. 142-145.

[22] S. Asare, D. Otoo-Arthur and K. O. Frimpong, "Assessing the Readiness of the Digitization of Health Records: A Case of a Municipal Hospital in Ghana," *International Journal of Computer Science and Information Technology Research,* vol. 5, no. 4, pp. 76-90, 2017.

[23] S. Mallik, "Image Alignment (Feature Based) using OpenCV (C++/Python)," Big Vision LLC, 11 March 2018. https://www.learnopencv.com/image-alignment-feature-based-using-opencv-c-python/.

[24] "Tesseract, an Open Source Optical Character Recognition Engine," *SSA Data*. [Online]. Available: https://www.ssa-data.com/technologies/ocr-pdf/tesseract/. [Accessed: 18-Apr-2020].