University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Using Semi-auto Annotation and Optical Character Recognition for Transcription of Patient Monitor using Smartphone Camera

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Jan Federico P. Coscolluela IV

June 2023

University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Using Semi-auto Annotation and Optical Character Recognition for Transcription of Patient Monitor using Smartphone Camera

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Jan Federico P. Coscolluela IV

June 2023

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled "Using Semi-auto Annotation and Optical Character Recognition for Transcription of Patient Monitor using Smartphone Camera" prepared and submitted by Jan Federico P. Coscolluela IV in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Marbert John C. Marasigan, M.Sc. (cand.)
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Avegail D. Carpio, M.Sc.	_____	_____
2. Richard Bryann L. Chua, Ph.D. (cand.)	_____	_____
3. Perlita E. Gasmen, M.Sc. (*cand.*)	_____	_____
4. Ma. Sheila A. Magboo, Ph.D. (*cand.*)	_____	_____
5. Vincent Peter C. Magboo, M.D.	_____	_____
6. Geoffrey A. Solano, Ph.D.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Vio Jianu C. Mojica, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Marie Josephine M. De Luna, Ph.D.
Chair
Department of Physical Sciences and Mathematics

Maria Constancia O. Carrillo, Ph.D.
Dean
College of Arts and Sciences

Abstract

Vital signs monitoring is a key function in healthcare delivery to ensure immediate and precise evaluation of a patient's well-being. It is done by attaching monitor devices to patients which collect, store, and display values on a screen. In many low-to-medium-income countries (LMICs), hospitals still rely on manual observation and handwritten documentation of vital signs, which is susceptible to human errors, data tampering, process inefficiency, and limited opportunities for comprehensive data analysis. More advanced hospitals utilize interface engines which transmit data to electronic medical records but tend to be model-specific and are very costly. Optical character recognition (OCR) offers a cost-effective and non-invasive alternative to digitizing manual transcription of vital signs data in healthcare settings with low financial resources. An image preprocessing pipeline is proposed to perform contour-based screen extraction of the patient monitor captured by a camera, thus providing a well-defined region more suitable for subsequent tasks of object detection and data extraction. The study offers a newly accrued dataset of over 4000 images of Mindray Beneview T8 patient monitor with multi-parameter annotations. Results showed that screen extraction prior to object detection significantly improved the mean Average Precision (mAP) of the model from 68.55% to 93.65% at an IoU threshold of 0.7.

Keywords: patient monitor, optical character recognition, object detection, image preprocessing, annotation

Contents

Acceptance Sheet i

Abstract ii

List of Figures vi

List of Tables vii

I. Introduction 1

A. Background of the Study . . . 1

B. Statement of the Problem . . . 1

C. Objectives of the Study . . . 2

D. Significance of the Project . . . 3

E. Scope and Limitations . . . 3

F. Assumptions . . . 4

II. Review of Related Literature 5

III. Theoretical Framework 11

A. Patient Monitor Screen . . . 11

B. Image Annotation . . . 12

C. Image Preprocessing . . . 12

1. OpenCV . . . 12

2. Grayscaling . . . 13

3. Gamma Correction . . . 14

4. Canny Edge Detection . . . 14

5. Skew Correction . . . 16

IV. Design and Implementation 18

A. Data Collection Setup . . . 18

B. Dataset Annotation . . . 19
1. Manual Annotation . . . 19
2. Object Detection Training . . . 19
3. Bounding Box Proposal . . . 20
4. Manual Fine-tuning . . . 20
5. Workload Estimation . . . 20
6. Labelled Dataset . . . 21
C. Image Preprocessing Flowchart . . . 22
1. Framing . . . 22
2. Image Preprocessing . . . 22
D. System Architecture . . . 23
E. Technical Architecture . . . 23

V. Results **24**
A. Dataset . . . 24
1. Manual Annotation . . . 25
2. Semi-Auto Annotation . . . 25
B. Image Preprocessing Pipeline . . . 27
1. Gamma Correction . . . 27
2. Edge Detection . . . 30
3. Accuracy . . . 31
4. Processing time . . . 32
C. Optical Character Recognition . . . 33
D. System . . . 35

VI. Discussions **38**

VII. Conclusions **39**

VIII. Recommendations **40**

IX. Bibliography **41**

X. Appendix **46**

A. Ethics Board Approval 46

B. Philippine General Hospital Approval 49

C. Source Code 50

XI. Acknowledgment **65**

List of Figures

1 Standard Patient Monitor 11
2 Architecture of OpenCV 13
3 Gamma Correction 14
4 Canny Edge Detection 15
5 Skew Correction with OpenCV 17
6 Beneview T8 Patient Monitor 18
7 Semi-auto Annotation Methodoloy 19
8 Image Preprocessing Pipeline 22
9 Edge Detection Sub-Explosion 22
10 Manual Annotation using LabelImg 25
11 Automated Annotation using Object Detection 26
12 Semi-auto Annotation Workload Reduction. 26
13 Image Preprocessing Pipeline Implementation 31
14 Image Preprocessing Time 32
15 Precision-Recall Curves of Object Detection. a-c) Raw Dataset; d-f) Pre-processed Dataset 33
16 Detected Health Parameters at t_{min} $_{conf:score}$ = 0:5 34
17 Home Page 35
18 In-app Camera Access Page 36
19 File Upload Page 36
20 Dataset Page 37
21 Guide Page 37

List of Tables

1 Dataset File Naming 21

2 Dataset Image Classes Overview 24

3 Comparison of Dynamic Gamma Correction on an Image with Natural Lighting 28

4 Comparison of Dynamic Gamma Correction on an Image with Low Lighting 29

5 Error Deduction Summary for Image Quality Metrics (MSE, PSNR, SSIM) 29

6 Edge Detection Algorithms Comparison (min_{thresh} = 40) 30

7 Average Precision (AP) across Health Parameters (IoU 0.70) 34

8 OCR-extracted Vital Signs Data 34

I. Introduction

A. Background of the Study

Patient monitoring is a critical aspect of healthcare delivery, ensuring timely and accurate assessment of vital signs and overall patient well-being. Patient monitor devices serve as the primary tool to continuously measure and display vital signs such as heart rate, blood pressure, and oxygen saturation. This provides healthcare professionals with real-time information about a patient's physiological status for long-term observation and early medical interventions as needed [1]. Despite technological advancements, manual monitoring techniques, such as nurses manually observing or taking note of value, still persist in many hospitals. This approach poses several limitations, including the risk of human error, consumption of time, di culties in data storage and retrieval, and hindered opportunities for comprehensive analysis.

Digitizing manual patient monitoring through the utilization of Optical Character Recognition (OCR) technology o ers a promising solution to address these challenges as it grows to be a growing area of research [2]. This provides a cost-e ective solution with reduced hardware costs and connectivity expenses using camera without the need for expensive cords or third party software. To this end, the UP Manila Standards and Interoperability Lab (UPM SILAB) can incorporate data interoperability across health institutions involving patient monitor data, speci cally in the Philippine General Hospital (PGH).

B. Statement of the Problem

The inadequacy of publicly available patient monitor dataset poses challenge to the development of OCR system with speci c context and focus of application. Despite third-party software o ering simulated patient monitor videos, most of which are paid, these datasets are already high-quality and mainly designed for trend analysis studies.

Furthermore, one common challenge encountered when working with computer vision is the presence of noise in image datasets. The problem at hand is to develop e ective

techniques and methodologies to mitigate noise in image datasets, thereby enhancing the quality and reliability of the data for subsequent analysis and applications.

C. Objectives of the Study

This study intends to provide a newly accrued dataset of patient monitor images re ecting a realistic hospital environment. In particular, the goals of the paper are as follows:

ˆ Dataset Objectives :

1. Manually collect video recordings of patient monitor from the Post Anesthesia Care Unit of Philippine General Hospital
2. Utilize smartphone camera to perform data collection in both natural and low lighting conditions with di erent camera angles
   - { direct camera
   - { skewed to the left
   - { skewed to the right
   - { skewed upwards
   - { skewed downwards
3. Perform frame extraction to obtain patient monitor image dataset from captured videos
4. Implement a semi-auto annotation approach to expedite the manual dataset annotation procedure
5. Fully annotate raw and preprocessed images in PascalVOC XML format
6. Train a customized object detection model to locate vital signs from the newly accrued dataset
7. Apply optical character recognition to extract vital signs from model detections

ˆ System Objectives :

1. Allow the user to capture an image or video of a patient monitor via device camera in real-time
2. Allow the user to upload a pre-taken image or video of a patient monitor
3. Implement frame extraction to retrieve individual images from video input type at 2-second intervals
4. Implement a screen-extraction procedure using an image preprocessing pipeline
   (a) gamma correction
   (b) edge detection
   (c) skew correction
5. Allow user to download preprocessed image output in ZIP and CSV format

D. Signi cance of the Project

A new dataset of camera-captured patient monitor images with multi-parameter annotations is a contribution to computer vision which can aid benchmarking research, validation of computer vision algorithms, and remote patient monitoring.

Creating a web application that allows access to camera and integrates image enhancement procedures can also be scaled for future work of text and digit recognition or waveform interpretation. In particular, the proposed study can serve as an initial step for UPM SILAB's OCR project for PGH, focusing on image optimization before the OCR process. By optimizing images prior to OCR, a more suitable image data for subsequent tasks is obtained. The workload on healthcare sta can be minimized, and valuable patient information can be preserved for research purposes.

E. Scope and Limitations

This study operates under the following conditions:

1. The dataset only considers Mindray Beneview Series patient monitor.
2. Camera angle during data collection is adjusted manually without the use of any software.

F. Assumptions

This project operates under the following assumption/s.

1. The monitor is not obstructed by any object during capture.
2. The monitor is su ciently captured and not cropped.

II. Review of Related Literature

Computer vision is a field of artificial intelligence that has been extensively utilized across various domains such as real estate, businesses, and healthcare. It is used to simulate human visual abilities by enabling computers to analyze surroundings as humans do|or even more. A concept under computer vision known as optical character recognition (OCR) is a highly researched topic [2]. This is typically done by having a digital image of a document, performing image processing to remove unwanted information, training the computer to locate characters of interest, and finally segmenting the detected characters for identification [3]. For instance, self-driving cars incorporate OCR technology not only to facilitate detection of objects such as obstacles and nearby vehicles but also to perform corresponding actions to keep the car free from collision.

Majority of previous work utilizing OCR focused on number recognition, document analysis, and vehicular license plate recognition [4]. For instance, Zacharias et al [2] explored the extraction of Intermodal Loading Units (ILU) codes printed on the rear end of swap bodies (freight containers for road and rail transport) using a text recognition pipeline with the open-source Tesseract OCR engine. A small variation in illumination among the captured images was found to contribute to large errors in text recognition and thereby negatively affect model success metrics. Implementation of deep learning-based model can be promising to overcome the large fluctuation of model accuracy with scene text images [2].

Such recommendation for use of deep learning was explored by an optical character recognition post-correction study conducted by Karthikeyan et al. [5] which showed the feasibility of model accuracy improvement applied on medical reports. Correct transcription and recognition of documents is a key challenge identified in this paper due to presence of noise such as obscured, skewed, or illegible text. Specific medical terminologies deviating from general language lexicons were also found to compound the error rate of OCR process. Introducing medical terminologies to the vocabulary of employed OCR model is

a highlighted technique that could be employed in patient monitor dataset since health parameters and their symbols may not syntactically align with that of general language. This apparent dependence of OCR model accuracy on dataset quality suggests the critical role of data collection and proper image preprocessing as applied to a speci c context of data. In a medical study using lung MRI images as dataset, ltering techniques such as Wiener, median, and Gaussian reduced the time it takes to process the images [6]. Blur detection is a technique that can be explored in related works to assess the quality of captured medical image beforehand and retake the data collection phase prior to further processing.

Despite the extensive studies using OCR, work centered particularly on scanned documents ranging from business forms, receipts, and bibliographic data. Commercially available OCR tools are also optimized for scanner-captured documents which results to drastic decrease of the transcription accuracy for camera-captured images due to apparent noise and distortion from environmental factors [7]. Further, lack of open studies around these commercial tools leads to low system repeatability and assessment. The challenge thus remains on expanding the application of OCR primarily in the context of healthcare where captured images or video from data sources are not readily suitable for modeling. For instance, medical data may come in the form of prescriptions and patient records which are typically stored on paper with the possibility of content smudges, handwritten corrections, and writing style di erences. Pronouncing this tendency of low-quality data is the fact that data collection in the medical context must be noninvasive and consensual which could translate to moving the capturing device at a distance or angle to avoid inconvenience.

Survey results showed that errors committed during data entry in clinical databases range from 2.3% to 26.9% which roots from data entry mistakes and misinterpretation of information [8]. Adriano et al. [3] aimed to reduce the high error rate of data entry using OCR applied on novel digital conversion model for hand- lled forms. Their dataset came from a selected special database that readily provided forms (containing handwritten

text) to facilitate character recognition and training of classi ers. Their best-performing pipeline used feature extraction via AlexNet, a convolutional neural network architecture. They recommended other CNNs for exploration namely ResNet and Squeezenet, as well as using other SVM kernels like Gaussian and RBF| points of work that presents great utility. Exploring scanned medical prescriptions, a camera inside an IoT-enabled smart medicine box embedded with OCR technology was explored by the work of Rumi et al.[9]. This targets elderly patients who cannot monitor their medication by notifying an individual about the medication information extracted from their respective prescriptions. The paper's focus on scanned prescriptions can be further extended to another clinical setup like patient information displayed on medical devices which not only requires text recognition but also correct mapping of numerical health values to their corresponding health parameters (e.g., heart rate, blood pressure).

In a work by Xue et al.[10], a text detection and recognition pipeline considered two real-life scenarios in the medical scene: (1) multilingual laboratory reports, and (2) documents with many textual objects each occupying a very small region. The authors proposed a deep learning approach that performs a patch-based training strategy applied to a detector that outputs a set of bounding boxes containing texts. A concatenation structure is then inserted into a recognizer that takes the areas of bounding boxes in the original image as input, thereby outputting the recognized texts. The patch-based strategy enforced by the authors in text detection module achieves 99.5% recall and 98.6% precision, a desirable result given the average quality of images. Likewise, the concatenation structure e ectively improved the recognition performance by being able to deal with images with di erent resolutions at 90% accuracy. Their patch-based strategy during text detection may be something worth looking into given its contribution to achieve desirable success outcomes in terms of recall and precision.

OCR works e ciently with printed text documents [11]. However, as mentioned above, medical data does not include textual forms alone but spans widely across di erent medical devices as such as blood pressure monitors and patient monitor systems among many

others. Few datasets exist such as the Queensland [12] and VitalDB [13] dataset but they are both high- delity vital signs database designed for anesthesia monitoring research and biosignal analysis, respectively. To the best knowledge of the researchers, there is a lack of camera-captured patient monitor images re ecting actual environment conditions (e.g., illumination variation, background noise, etc.) which are essential in optical character recognition.

In the work of Kulkarni et al. [14], OCR was used to digitize camera-captured blood pressure readings through a mobile application. The paper underscored medical data transcription errors as well as relatively inadequate technologies in low- to middle-income countries (LMICs). The use of ubiquitous phone camera to detect LCD frame location provides a cost-e ective solution to facilitate OCR without the need for expensive software or high-end capturing devices Their modular image enhancement algorithm including image binarization, LCD frame localization, and LCD frame normalization may also be used as reference when applied on a similar medical tool like patient monitor. Similar to previous works, low image quality was found to signi cantly degrade their model accuracy, hence post-OCR correction may be applied.

A study by Shenoy et al. [15] developed a smartphone-based system that automatically reads and records biometric monitor results from a camera-captured monitor reading. This was, however, limited to seven-segment displays and does not involve recognition of alphanumeric content as observed in a patient monitor screen. Its target device is also limited to Apple's HealthKit in iOS, which leads to less generalizability but poses points for open work.

Storage itself of extracted information is as equally important as text recognition to facilitate research, drive business decisions, and assist in forecasting and policy making. However, medical devices and screens may have limited hardware capabilities to store and export data for further clinical research and diagnosis. This is particularly the case for LMICs, where technology may not be as advanced as other countries [14].

Document archiving and record management was explored with application of optical

character recognition in the paper of Jayoma et al. [16]. The authors of such document archival study focused on digitization of multiple forms of records in the Department of Social Worker and Development (DSWD) Caraga. Their general framework consolidating OCR and information storage used open-source technologies such as Django, MySQL, and Pytesseract which can be used as references to develop a system using similar technologies. This can further be extended in terms of a di erent dataset (i.e., images from medical devices).

In a work by Yadav et al.[17], a robust web application that uses OCR to extract information from handwritten and printed documents was developed. Their technical architecture comprised four sequential processes namely (1) adaptive thresholding, (2) connected component analysis, and (3) line and word detection, and (4) two-layer text recognition. Speci cally, the use of adaptive thresholding to account for variations in illumination in the image dataset may serve as reference in image preprocessing of di erent dataset. The study showed the feasibility of text recognition hosted online.

A system built by Froese et al. [18] extracted the desired information from real-time pump monitor images. Their methodology mainly used scripting to extract images from a medication pump which is then fed to an OCR model. Recognized text and values are then transferred to a real-time monitoring software. It was underlined that future work is required for more universal application of such system which can be explored by superimposing their model on a di erent medical dataset and assessing the accuracy. Their data collection setup through a USB camera capturing images from the medical pump at 60 frames/second can be employed in my paper. By observation, data capture used in such paper was relatively near the pump (i.e., the USB camera is immediately in front of the device). Their capturing conditions can be extended in this study by incorporating more realistic scenarios such as the camera slightly tilted or skewed with respect to the patient monitor. Hence, further image optimization encompassing variation in camera face angles can be explored [18].

The feasibility of using OCR to extract information from a patient monitor screen was

also shown in Bukhari's work [19]. Various image preprocessing such as binarization and bitwise masking were used on a high-quality dataset retrieved from SimCapture. The OCR pipeline used in such paper includes a script that extracts frames per second from input video and individually extracts health values eventually saved in a CSV le. This may serve as basis of the proposed system to implement a data export functionality in order to provide the user a downloadable le consisting of the extracted information in easily editable format. Future work was encouraged which can be summarized in three parts: (1) more image preprocessing to ensure that the model is dynamic, (2) automatic detection of all pixel color values of parameter for classi cation, since only 4 colors are considered in the paper, and (3) use of deep learning models in contrast to traditional image processing techniques. Given that high data quality is required to maintain the model accuracy [19], the study may be extended to be applied on patient monitor dataset taken from a real medical setting with environmental factors present such as brightness variation, blurring, distant capture, etc.

With all considered, data entry errors being committed in healthcare|let alone the tedious process of such task|slows down clinical procedures and leaves plenty of room for improvement. It was further pronounced that dataset quality is a key consideration in developing an accurate OCR model, upon which OCR post-correction methods and several image preprocessing techniques are possible workaround. To this end, the research aims to ll in the gap among previous studies through (1) use of smartphone camera to collect and curate realistic eld image dataset of patient monitor, (2) creation of an image preprocessing pipeline to improve image quality, and (3) development of a system to utilize the image preprocessing pipeline to enhance raw image of patient monitor. Contributing a new set of patient monitor images and developing an image preprocessing pipeline to enhance such images would provide a benchmark dataset and development of real-time OCR applications in a similar domain.

III. Theoretical Framework

A. Patient Monitor Screen

Patient monitoring system was introduced by Venetian Doctor Santorio in 1965 through his publication of methods to measure body temperature using spirit thermometer and pendulum for counting heart rate. With the advent of integrated circuits and advancement of technology, computer-based patient monitoring systems with better computing power have been developed. A widely used medical device is a patient monitor screen which continuously monitors patient parameters such as oxygen level, heart rate, blood pressure, etc. These data are observed via non-intrusive sensors on human body to check the condition of the patient over time which facilitates prompt assessment and decision-making relative to real-time patient status such as those coming straight from surgery in Intensive Care Units (ICUs). A standard patient monitor [20] based from is shown in Figure 1.

Figure 1: Standard Patient Monitor

A notable trend among patient monitors is that the numerical values are highly contrasted with a black background, with characters displayed in synthetic fonts. As of date, these medical devices are still widely used to monitor patients not just in the medical sector but also in social support such as retirement homes.

B. Image Annotation

Image annotation is the task of assigning labels to an image to create metadata for a training dataset in computer vision models. The model utilizes such annotations as its ground truth, and uses them to learn how to label or detect objects or images on its own. Image annotation is typically useful in object recognition, or object detection, which enables machines to identify a particular object in an image and apply the accurate label. An example is a self-driving car which labels its surroundings depending on whether vehicles and/or obstacles are nearby.

C. Image Preprocessing

The aim of image preprocessing is quality improvement by suppressing undesired distortions and enhancing some features to obtain more suitable data for further processing and analysis tasks.

1. OpenCV

Open-source computer vision (OpenCV) is an image preprocessing library that has gained popularity in computer vision given it is open-source. It was originally envisioned to support computer operations such as object identi cation, image recognition, and object movement tracking but has now expanded to over 2500 functions based on its documentation. This enables faster execution of tasks such as color conversion, image masking, and lter application. Furthermore, its interface exibility allows for multiple programming languages such as Python, Java, and C as well as di erent platforms such as Mac and Windows. Figure 2 presents the architecture design of OpenCV in a mobile imaging work [21].

Figure 2: Architecture of OpenCV

In comparison with other similar tools like Matlab, OpenCV provides a relatively detailed toolbox for image processing instead of generic solutions. The wide array of functions in OpenCV also e ciently integrates common noise removal and image quality manipulation techniques in one library.

2. Grayscaling

Most OCR engines normally perform better with grayscaled images which refers to a color space with only one channel. Pixels in typical images are represented in Red-Green-Blue (RGB) format which gives them the color that the human eye perceives. There are three ways on how to compute the new value of pixel from RGB: average, lightness, and luminosity. The average method takes the simple arithmetical mean across the color channels of certain pixel. Lightness is computed by averaging the maximum and minimum value of pixel color channel. Lastly, luminosity works with the average of all color channels, with every single channel weighted. Formulas for these conversions are shown in formulas (1), (2), (3).

$$\text{lightness} = (\max(R;G;B) + \min(R;G;B))/2 \quad (1)$$

$$\text{average} = (R + G + B)/3 \quad (2)$$

$$\text{luminosity} = 0:299(R) + 0:587(G) + 0:115(B) \quad (3)$$

3. Gamma Correction

This technique can be used to control the brightness of an image. Such method is typically used in image preprocessing to adjust the image brightness depending on how it was captured. Gamma values less than 1 will shift the image towards the darker end of the spectrum while gamma values greater than 1 will make the image appear lighter. A gamma value exactly equal to one will result in no change in image [22].

Figure 3: Gamma Correction

4. Canny Edge Detection

OCR generally performs better if the object of interest is narrowed down from the input image. For instance, a scanned receipt may be slightly skewed, with other non-essential objects included in the same image (e.g., pen, person, etc.). Edge detection is a technique that aims to extract the four corners of an object of interest such as documents or monitor display. One popular edge detection approach is Canny Edge Detection. The entire process of this detection [23] is summarized in Figure 4.

Figure 4: Canny Edge Detection

- Noise Reduction via Blurring

  Edge detection results are particularly sensitive to image noise and one way to address this is through the application of Gaussian blur to smooth an input image. To do so, image convolution technique is applied on an input image with a Gaussian Kernel which may have varying kernel size such as 3x3, 5x5, etc. The kernel size in uences the intensity of blur, where higher value leads to more visible blur e ect.

- Gradient Calculation

  This step detects the intensity of edges as well as direction via calculation of the gradient in the image using edge detection operators. A change of pixels' intensity represents an edge. Filters can be applied in order to highlight such intensity change in horizontal and vertical directions and easily detect the edges.

- Non-maximum Suppression

  Thin edges are ideal in the output images. Hence, presence of thick edges can be addressed through non-maximum suppression to thin them out. The algorithm essentially iterates through every point on the gradient intensity matrix and locates the pixels whose value in the edge directions is maximum.

- Double Threshold

  The goal of this step is to identify three kinds of pixels namely strong, weak, and non-relevant:

- Strong pixels are those with relatively high intensity that assures as about their contribution to the final edge.
- Weak pixels are those with intensity that is neither high or low enough to be considered strong or non-relevant, hence are still potentially contributors in the edge.
- Any other pixel not classified under the two aforementioned types belong to this class.

With this considered, high threshold is used to identify the strong pixels while low threshold is used to identify the non-relevant ones. On the other hand, the rest of the pixels having intensity between both thresholds are identified as weak which are then further filtered out by the next step to delineate whether it ultimately belongs to strong or non-relevant.

- Edge Tracking by Hysteresis

  Based on the threshold results, the hysteresis consists of transforming weak pixels into strong ones, if and only if at least one of the pixels around the one being processed is a strong one [23].

5. Skew Correction

Raw image content, especially text, sometimes tend to be skewed or tilted at a certain angle. This is contributed by the point of capture where the camera is not leveled with that of the object. For computer vision tasks, skew correction is essential to improve model accuracy by ensuring as much visually normal input as possible. Python has libraries to implement correction of perspective like OpenCV [24].

Figure 5: Skew Correction with OpenCV

In machine learning, especially computer vision, the quality of the data is just as important (if not more) as the model itself. Hence, performing necessary image preprocessing procedures on raw images have signi cant contribution toward noise reduction and overall positive e ect on model training.

IV. Design and Implementation

Ethical approval from UP Manila Research Ethics Board (UPM REB) and Philippine General Hospital Expanded Hospital Research O ce (PGH-EHRO) is obtained to proceed with the manual collection of dataset.

A. Data Collection Setup

A smartphone camera (iPhone 11) is mounted on a tripod to capture data from a Mindray Beneview T8 patient monitor attached to ve (5) healthy volunteers at the Post-Anesthesia Care Unit (PACU) of PGH. A sample image of the monitor is illustrated in Figure 6.

Figure 6: Beneview T8 Patient Monitor

The inclusion criteria for data acquisition were as follows:

- Aged 18 - 65 years old
- Student, or faculty from College of Arts and Sciences and/or College of Medicine

Vital signs data are recorded at a resolution of 1920 x 1080 at 60 frames/second. Every volunteer session lasts for 30 minutes and camera placement is adjusted every 3 minutes

to account for di erent capture conditions. The tilt and angle to which the camera was skewed are manually adjusted up to a maximum of 45 degrees.

B. Dataset Annotation

In this paper, a practical heuristic for bounding box annotation on the proposed image dataset is presented through a trained object detection model to automate the manual approach. This intends to reduce workload by shifting the majority of human involvement to the correction stage only.

Figure 7: Semi-auto Annotation Methodoloy

1. Manual Annotation

The process begins with domain experts manually annotating a randomly selected batch of images (η_1) from the unlabelled dataset. The annotation involves full human involvement to draw bounding boxes around health parameters and provide their corresponding class labels. The open-source annotation software LabelImg is used with no speed-up procedures.

2. Object Detection Training

The next step is to train an object detection model. Transfer learning is applied by using a pre-trained SSD network and ne-tuning on the proposed dataset [25]. The single shot detector (SSD) network [26] proposed by Liu et al. is used for the detection architecture

given its lightweight nature. It is pre-trained with MS COCO dataset and is typically the model of choice for resource-limited inference scenarios given that the detections are produced directly in a single forward pass of the network [27]. Furthermore, the Mobilenet V2 [28] is applied for the backbone.

3. Bounding Box Proposal

The trained model is used to predict bounding boxes for the unlabelled images with an associated con dence level for each detection. A con dence threshold value between 0 and 1 is used to de ne a true positive. In other words, the model will only draw bounding boxes around a detected parameter if and only if its associated con dence level is equal to or higher than the speci ed threshold.

4. Manual Fine-tuning

The resulting annotations proposed by the model are inspected and manually corrected by the domain experts through several corrective measures as follows:

- Addition : Missing bounding box is manually drawn around a parameter, if needed.
- Removal : Incorrectly predicted box is deleted from the annotation.
- Label Correction : Mislabeled class is corrected.
- Box Adjustment : If the predicted box is too wide or insu ciently encloses a parameter, the box is recalibrated accordingly.

5. Workload Estimation

We estimate human workload by comparative analysis of how much time is spent between the manual and semi-auto annotation strategies. For the manual approach, the total time (T) to complete the annotation as described in Section B.1 is measured with a timer. The average ( ) is then calculated using the formula $\frac{T}{n_1}$ which corresponds to the estimated

time to annotate a single image. This value is then multiplied to the total number of images in the dataset to estimate the overall duration to label the dataset exclusively through a manual approach. On the other hand, the semi-auto annotation strategy is measured by adding the time consumed both in Section B.1 and Section B.3.

6. Labelled Dataset

After the correction stage, the fully labeled image dataset is saved as a ZIP le containing the images in JPG format and their corresponding annotations in Pascal VOC XML format. This is done for both raw dataset and its preprocessed counterpart (screen-extracted).

To enable easier navigation of the dataset, les are named as follows: volunteer number le code _ frame count. For instance, the image with le name 01_01_1.jpg corresponds to the rst extracted frame from the rst volunteer data with a direct camera and natural lighting condition.

File Code	Capture Orientation	Lighting Condition
01	Direct Camera	Natural
02	Direct Camera	Low
03	Skewed to Left	Natural
04	Skewed to Left	Low
05	Skewed to Right	Natural
06	Skewed to Right	Low
07	Skewed Upward	Natural
08	Skewed Upward	Low
09	Skewed Downward	Natural
10	Skewed Downward	Low

Table 1: Dataset File Naming

C. Image Preprocessing Flowchart

Figure 8 summarizes the proposed image preprocessing pipeline.

Figure 8: Image Preprocessing Pipeline

1. Framing

The OCR cannot process an input le in video format, hence frames are extracted. The video dataset is fragmented into individual images at 2-second intervals.

2. Image Preprocessing

Once framing is done, the brightness of an image is automatically adjusted using the concept of dynamic inverse gamma correction. Afterward, the brightness of an image is automatically ne-tuned using dynamic inverse gamma correction followed by image smoothing to blur the image. It is followed by edge detection to identify the edges of the patient monitor to be extracted. The proposed ow for this detection is shown below.

Figure 9: Edge Detection Sub-Explosion

Lastly, skew correction is applied using OpenCV python library to address any degree of skewness in the image information.

D. System Architecture

Monixor is a web application that uses PostgreSQL as the database server. It is developed using the Python-based framework Django to enable easier integration with machine learning, image preprocessing, and optical character recognition implementations.

E. Technical Architecture

The minimum requirements for the server machine include:

- Apache 2.4.23
- 1GB RAM
- PostgreSQL 14

The client-side must satisfy these minimum requirements:

- Google Chrome 57.0.2897
- Mozilla Firefox 43.0.1
- Windows 7 / Android 7.0+ / iOS 12.4+
- Intel Core i5-4200U
- 4GB RAM