DEVELOPMENT OF AN AI SOLUTION FOR SURGICAL GAUZE MANAGEMENT

Lee Yun Fai

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

2021/2022

Project No. C006

Development of an AI Solution for Surgical Gauze Management

Submitted By: Lee Yun Fai

Supervisor: Associate Professor Cai Yiyu

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

A Final Year Project report presented to Nanyang Technological University in partial fulfilment of the requirements for the Degree of Bachelor of Engineering (Mechanical Engineering)

2021/2022

Abstract

Gossypiboma, the accidental retainment of gauze in a patient's body is a potentially fatal problem that persists on due to human error. This error not only affects the patient but also affect the hospital in the form of reputation loss and lawsuits that result in large monetary losses. Over the years, several solutions such as manual counting standard operating procedures, X-Ray detectable gauzes and RFID-tagged surgical gauzes were implemented with limited success. These solutions functioned as secondary safety-nets and in a limited extent assisted in further reducing the odds for Gossypiboma. However, these methods are still too expensive and inefficient for long term use.

As such, this project demonstrates a functional and practical smart system solution that can support medical staff in the Operation Theatre in the prevention of Gossypiboma in a more efficient and effective manner. Several new technologies such as Computer Vision, Transfer Learning and Edge Computing were investigated and integrated to develop a smart solution for Surgical Gauze Management.

With the advancements made in this new system till date, it can keep track of gauzes used during surgery with a high accuracy of 99.7% and process its video feed at 12 Frames Per Second, enabling real-time usage. It is also compatible with the current standard operating procedure in Singapore General Hospital. The main objective of developing a smart system solution that can support the medical staff in reducing human error in the aspect of Surgical Gauze Management in the Operation Theatre was achieved.

Acknowledgements

I would like to offer my heartfelt gratitude to my supervisor Associate Professor Cai Yiyu for his guidance and support throughout my whole Final Year Project. His extensive expertise and vast knowledge have aided me greatly for the consistent progress made throughout this project.

Additionally, I would like to thank SingHealth Group Singapore General Hospital Dr Luke Tay Hsein Ts'ung and Assistant Nurse Clinician Aaron Tham Xirong for providing insights into the medical guidelines and thought processes of medical staff. They were vital for the successful development of the Surgical Gauze Management solution.

Finally, I would like to express appreciation for Jishen in his contributions with regards to the physical design of the product and Soon Kim for his support even after he had passed down to me his work in the previous Android prototype.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AIoMT	Artificial Intelligent Internet of Medical Things
C3	Command, Control and Communication
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CSI	Camera Serial Interface
CVAT	Computer Vision Annotation Tool
FPS	Frames Per Second
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HDMI	High-Definition Multimedia Interface
IDE	Integrated Development Environment
IoMT	Internet of Medical Things
IoT	Internet of Things
IoU	Intersection Over Union
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
LED	Light Emitting Diode
mAP	mean Average Precision
Micro-SD	Micro Secure Digital
MicroSDHC	Micro Secure Digital High Capacity
MicroSDXC	Micro Secure Digital Extended Capacity
NMS	Non-Maximum Suppression
NVME	Non-Volatile Memory Express
PASCAL VOC	PASCAL Visual Object Classes

PC	Personal Computer
PCIE	Peripheral Component Interconnect Express
RFID	Radio Frequency Identification
RPM	Revolutions Per Minute
SD	Secure Digital
SDK	Software Development Kit
SGD	Singapore Dollar
SOP	Standard Operating Procedure
SSD	Solid State Drive
SSD MobileNet V1	Single-Shot Multibox Detection MobileNet V1
UI	User Interface
US	United States of America
USB	Universal Serial Bus
UX	User Experience
VS Code	Visual Studio Code
WHO	World Health Organization

Chapter 1
Introduction

1.1 Background

Gossypiboma, a postoperative complication from the accidental retainment of gauze in a patient's body, is a persisting problem worldwide. While it is infrequent and reported to occur in 1 out of 300 to 1000 of all surgical interventions, Gossypiboma is associated with significant morbidity to the patient and present severe medico legal consequences to the hospitals [1] [2]. Patients may remain asymptomatic for a long time but when symptoms present themselves, they often come in the form of pain, abscess formation and septic shock, highly likely leading to death [3]. As a result, the lawsuits for the emotional distress by the patient, subsequent hospitalisation for reoperation and compensation for perceived negligence of the facility and staff can run anywhere from USD $2 to $5 million, and even more [4] [5].

To tackle Gossypiboma, hospitals worldwide rely on care processes and standardised gauze counting protocols, such as the WHO Surgical Safety Checklist, to ensure the gauze count tally at specific time points throughout the surgery [6]. For example, in surgeries across SingHealth Group of hospitals in Singapore, gauzes in sets of 10 are placed on a side tray and dispensed by the Scrub Nurses only when requested. Afterwards, when they are removed from the patient, the gauzes are laid out neatly in a grid pattern on a white or green towel to systematically account for the numbers. Lastly, right before the patient is closed, a final tally is done to ensure no gauze is left in the patient. It is estimated that nurses commit an average of 35 minutes to the standardised counting protocols per operation [7].

Nonetheless, even with the standardised gauze counting protocols in place, Gossypiboma cases continue, especially in cases of major surgeries that were prolonged with multiple manpower shift handovers, or when large numbers of gauzes were used to deal with excessive blood loss [8]. In a study by Japan Council for Quality Health Care, 84% of reported Gossypiboma cases from 2016 to 2019 recorded a matching gauze count before they sewed up the patient [9]. In another study done in the US, count discrepancies were found in 1062 of 153,263 surgical procedures but the missing item was found in only 51 of the 1062 cases, with 17 found in the patient [10].

Even in the presence of clear and thorough gauze counting protocols, human error remains. Standardised gauze counting protocols are certainly necessary, but they alone are not enough to achieve zero cases of Gossypiboma.

In a bid for improved accuracy and efficiency, hospitals have tried to incorporate existing technologies to supplement the existing gauze counting protocols as secondary safety-nets. In Singapore, Tan Tock Seng Hospital has RFID-tagged surgical gauzes that can be used in operations [11]. However, RFID-labelled gauzes were only partially readable inside the body cavity, with limited accuracy. They are also expensive, costing up to SGD $0.50 per piece and may drive up the cost of surgery greatly [12]. In other hospitals, X-Ray detectable gauzes were also used, but only a short string attached to each gauze can be captured with X-Ray Machines. Therefore, they too suffer from limited accuracy [13]. It may also prove to be dangerous for people already weakened by surgery to be subjected to X-rays, so it can only be used in pressing situations.

These secondary safety-net solutions have to a limited extent assisted in further reducing the odds for Gossypiboma. Nonetheless, the problem is still not resolved.

With the recent advancements in Computer Vision and Artificial Intelligence, a smart system in place would be able to greatly reduce the chances for Gossypiboma. A machine that will not get tired and possess high accuracy in counting the gauzes is ideal to get this job done.

1.2 Objective

The main objective of this project is to develop a functional and practical smart system solution to support medical staff in the Operation Theatre in the prevention of Gossypiboma.

1.3 Scope of Work

This project was done in a partnership between Nanyang Technological University and SingHealth Group, Singapore General Hospital. It is a continuation of the previous study that made an Android App proof of concept with Computer Vision AI to track and count the number of gauzes within its video feed.

With the future possibility of mass-production in mind, this project was rebuilt on the NVIDIA Jetson platform, using the Jetson Xavier NX Developer Kit and its supported accessories. Several Deep Learning techniques for Computer Vision were applied for a quick and accurate object detection model to be trained then utilised. New features such as dual camera capabilities were integrated into the system and multiple UI/UX designs were explored to achieve an intuitive User Interface for quick adoption by the medical staff. The required interactions with the new smart system were also designed to be as similar as possible to the existing Standard Operating Procedure (SOP) for the manual counting of gauzes, to enable easier transition for our medical staff users.

Chapter 2
Literature Review
2.1 AIoMT

The Internet of Medical Things (IoMT) is the combination of medical devices and applications that can utilise networking technologies to connect to healthcare IT systems [14]. It is a broad market that can involve concepts such as remote patient monitoring, robotic surgery systems, and even development of smart devices ranging from fall-tracking wearables to ingestible sensors that can pinpoint sources of internal bleeding.

The combination of Artificial Intelligence (AI) and IoMT gives a new field of development called AIoMT. With AI being capable of imitating the manner and pattern in which humans complete a task, the deployment of compatible algorithms into IoMT devices can improve efficiency in clinical workflows and relieve some burden on the medical professionals to enable better patient care.

One of the most important effects of AIoMT is the new capability to keep track of large quantities of real-time data and allow medical interventions to happen immediately, or before they even happen [15]. An example of this is the Command, Control and Communication (C3) System in Tan Tock Seng Hospital that collect and process thousands of ongoing operations indicators for medical professionals to have real-time perceptibility of patient flow and resources within the hospital [16].

The global AI market for healthcare was worth USD $6.9 billion in 2021, and by 2027, it is projected to grow to USD $67.4 billion [17]. Similarly, the global IoMT market was worth USD $73.5 Billion in 2021, and by 2028, it is projected to reach USD $190 Billion [18]. This trend of growing wide adoption of AIoMT is clear that there will be great changes to the healthcare sector and the lines between medicine and technology will be continuously blurred.

2.2 Hardware

For this AIoMT project, multiple hardware devices were used. The most noteworthy ones being the Jetson Xavier NX development kit, Raspberry Pi Zero W and Raspberry Pi Camera Module v2.

2.2.1 Jetson Xavier NX Developer Kit

The Jetson Xavier NX is a small form factor system-on-module from the NVIDIA Jetson family. The developer kit at the size of just 103mm x 90.5mm x 34.66mm hold 384 NVIDIA CUDA Cores, 48 Tensor Cores, 6 Carmel ARM CPUs, and 2 NVIDIA Deep Learning Accelerators engines within its compute module. It also has over 59.7 GB/s of memory bandwidth, video encode and decode, making it efficient enough to run several modern AI networks in parallel while processing high quantities of sensory data. With multiple power modes, it can run with as little as 10W while supporting the whole NVIDIA software stack, enabling the deployment of frameworks with accelerated libraries for Computer Vision [19].

The Jetson Xavier NX was developed with critical embedded applications in mind, making it ideal to be deployed into high-performance AI systems in the medical setting. Figure 1, extracted from [20], shows the Jetson Xavier NX Developer Kit.

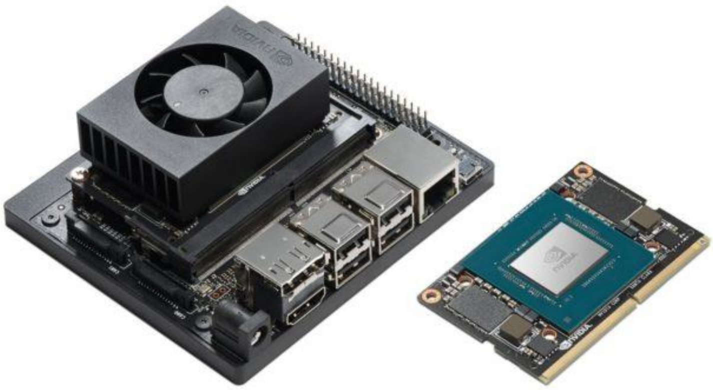


FIGURE 1: JETSON XAVIER NX DEVELOPER KIT AND ITS PRODUCTION COMPUTE MODULE

With multiple ports and interfaces available on the developer kit, various new features can be easily integrated into the product. For example, memory expansion via NVMe M.2 SSD PCIe Gen3x4, integration of cameras via MIPI CSI lanes and projection of the Graphical User Interface (GUI) via a HDMI 2.0 connected monitor can all be easily performed.

Table 1 below shows the breakdown of ports and interfaces within the developer kit [20].

Table 1: Jetson Xavier NX Developer Kit ports and interfaces

Purpose	I/O Port(s)
Compute	260-pin SO-DIMM connector to Jetson Xavier NX module
USB	(4x) USB 3.1 Type A | USB 2.0 Micro B (device mode + recovery)
Camera	(2x) MIPI CSI x2 (15-pin flex connectors)
Display	HDMI 2.0 | DisplayPort 1.4
Storage	microSD card slot | M.2 Key-M 2280 NVMe (PCIe x4)
Networking	Gigabit Ethernet (RJ45)
Wireless	M.2 Key-E (2×2 802.11 WLAN + BT 5.0 module provided)
40-Pin Header	(2x) I2C, (2x) SPI, UART, I2S, Audio Clock, GPIOs, PWMs
Button Header	Power, Reset, Recovery, Disable Auto Power-On
Power Supply	9-20V (19.5V supply provided) | 10/15W power modes

2.2.2 Raspberry Pi Zero W

The Raspberry Pi Zero W is a super-small ultra-low-cost computer. It measures at just 66mm x 30.5mm x 5mm and weighs only 9.3g. It had undergone extensive compliance testing and most noteworthy, has met the European standard for Restriction of Hazardous Substances (ROHS) Directive 2011/65/EU [21].

Capable of connecting to Wi-Fi without any additional components, the Raspberry Pi Zero W is appropriate for use as an extension of a smart system. With a single-core CPU with 512MB RAM, it can do some simple computations and relaying video feeds to the Jetson Xavier NX Developer Kit.

Figure 2 below shows the Raspberry Pi Zero W computer and Table 2 shows the breakdown of its ports and interfaces.

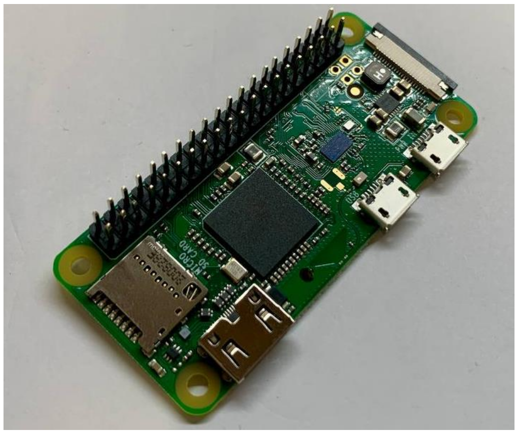


FIGURE 2: RASPBERRY PI ZERO W

TABLE 2: RASPBERRY PI ZERO W PORTS AND INTERFACES

Purpose	I/O Port(s)
Compute	1GHz, single-core CPU, 512MB RAM
USB	Micro-USB On-The-Go port
Camera	CSI camera connector
Display	Mini-HDMI port
Storage	microSD card slot
Wireless	802.11n wireless LAN | Bluetooth 4.0
40-Pin Header	HAT-compatible 40-pin header
Button Header	Composite video and reset headers
Power Supply	5V Supply

2.2.3 Raspberry Pi Camera Module v2

The Raspberry Pi Camera Module v2 has a Sony IMX219 8 Megapixel sensor that can capture pictures and videos in high definition. It has a good sensor resolution of 3280 x 2464 pixels and weighs in at just 3g with the dimensions of 3.68 x 2.76mm. Image quality, colour fidelity and low-light performances are not compromised in this small camera module, even when the connected 15 pin ribbon cable length vary between 15cm type and 2m type.

Most importantly, this camera is supported by the Jetson Camera Partners on the Jetson platform and can connect to the CSI port on the Jetson Xavier NX Developer Kit. It can also connect to the Raspberry Pi Zero W. Figure 3, extracted from [22], shows the Raspberry Pi Camera Module v2.

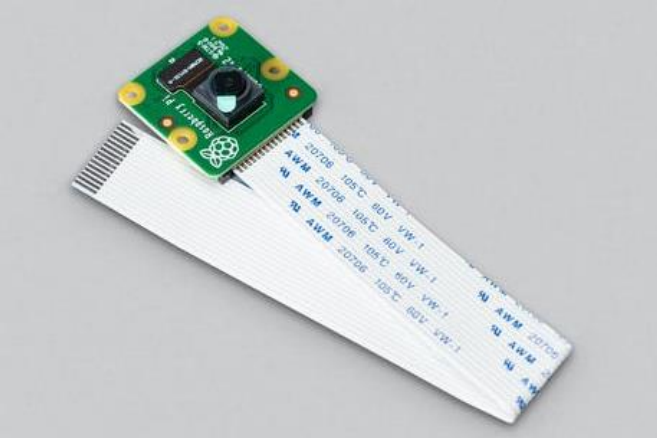


FIGURE 3: RASPBERRY PI CAMERA MODULE V2

2.3 NVIDIA JetPack Software Development Kit

NVIDIA JetPack SDK is a comprehensive solution towards the development of end-to-end accelerated AI applications. In the latest production release of JetPack 4.6, it included the Jetson Linux Driver Package with an Ubuntu-based Operating System, toolkits for GPU-accelerated applications, libraries for deep learning and more [23]. NVIDIA JetPack SDK support all Jetson modules and their developer kits.

The NVIDIA Jetson platform also support cloud-native technologies. Using NVIDIA Container Runtime with Docker integration, the Jetson platform can allow for GPU accelerated containerized applications. As a result, any application with all its dependencies held in one container image that works in one version of Jetson will work in another [23]. This is a strong advantage that can allow us to optimize within the different boards in the Jetson series and allow us to stay relevant even in the new Jetson releases.

With an OS based on Ubuntu 18.04, users of Jetson series developer kits not only enjoy the support of the Linux community, but also have the help of all the other Jetson users [24]. The NVIDIA development team and the open-source community together have also contributed to hundreds of thousands of samples, documentation, and developer tools online. Thus, enabling efficient progress to be made for all who use the Jetson series developer kits (Figure 4).

FIGURE 4: SCREENSHOT OF JETSON XAVIER NX DESKTOP

2.4 Computer Vision

Computer vision is an AI field that allow computers to derive information from the visual input of digital images and videos [25]. A Computer Vision system can be trained to work at an accuracy like that of a human, but at a speed greater than any human can achieve.

Computer Vision is accomplished with Deep Learning and Convolutional Neural Networks (CNN). Deep Learning is a subtype of machine learning which basically refers to a neural network with more than two layers. These neural networks allow them to learn from large amounts of data [26]. Among these neural networks, CNNs are especially useful for Computer Vision. With the neural network nodes arranged like those of the brain's frontal lobe – the region for processing visual stimuli in humans, it is especially effective to train for object detection purposes [27].

2.4.1 Transfer Learning

Generally, to train these deep learning models, large amounts of data and time must be used. The neural networks must run thousands of iterations over labelled image datasets that can consist of tens to hundreds of thousands of images, until they can distinguish objects and regions of interest from each other. This can take weeks or even months to train from scratch on very large datasets.

Without large datasets nor large amounts of time for training, there is still a way to achieve similar performances. This method leverages on already trained models to reduce a large percentage of the time spent and size of dataset required. It is called Transfer Learning.

Transfer Learning reuses model weights from pre-trained models as a starting point to retrain on your own dataset. It is a technique that use knowledge learned from a previous problem to improve prediction in a new but similar task. Looking into neural networks by and large, edges are detected in the first few layers, forms are detected in the middle layers then task-specific features are detected in the latter layers. In Transfer Learning, the early till middle layers are retained while the latter layers are retrained for a few hundred iterations, then the whole neural network's weights are toggled for further improvement in accuracy [28]. Figure 5 illustrates this process [29].

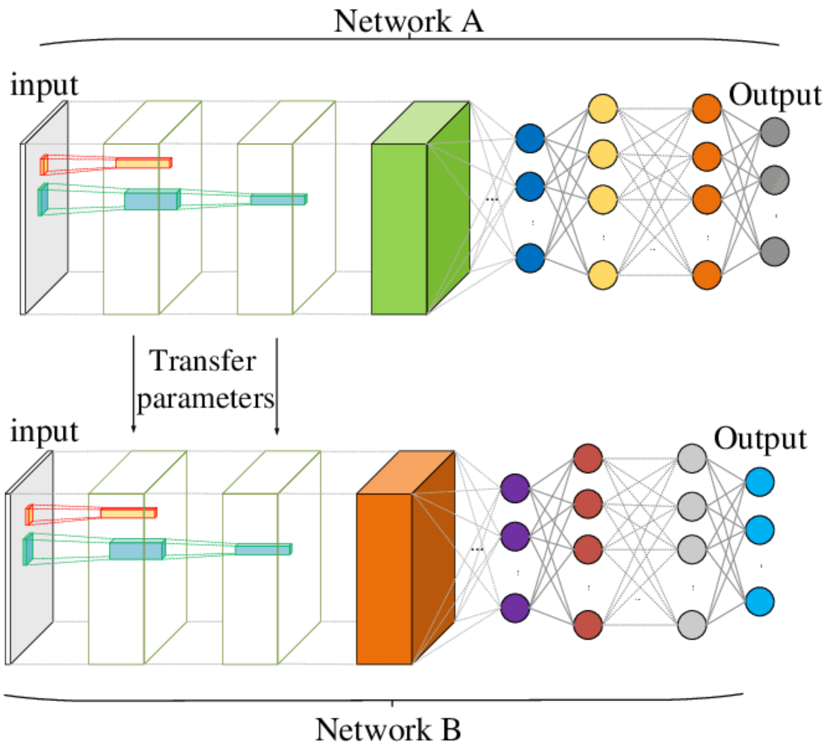


FIGURE 5: TRANSFER LEARNING PROCESS

2.4.2 Importance of Data Quantity and Quality

Even though Transfer Learning is a very useful technique, for the best object detection performances to be achieved, the new dataset must still be significantly large with minimal compromise to quality. Great Computer Vision datasets need both quantity and quality to succeed.

In a study by Datagen in 2021, they discovered some of the most prevalent data-related issues experienced by Computer Vision professionals were that there was poor annotation leading to quality issues, as well as insufficient and lack of variety of data for their use case, hence jeopardizing a project's progress. In the face of unsatisfactory training data, 99% of them stated that they had experienced project cancellations at one point and 80% of them had experienced delays lasting 3 months or more [30].

For good results to be achieved in the actual deployment of the AI product, the training dataset must conform to the data distribution in the actual application scenario in an unbiased manner as much as possible. Sufficient real image or video data from the actual application scenarios for the Computer Vision application must also be annotated accurately [31].

When this is done, data augmentation techniques such as geometric and colour space transformations can be applied to artificially increase the dataset size. One example of this is to decrease the brightness of several images in the dataset, while keeping the same labels in place, so that the Computer Vision system can adapt to low-light scenarios.

Even though this is a great technique that can improve the quantity of the dataset, it may not always improve the quality of the dataset. One risk is that for the case of colour transformations, it may cause loss in important colour information and so it is not always a label-preserving data augmentation [32]. Hence, data augmentation is a double-edged sword that should be selectively and carefully used.

2.4.3 Deployment of Neural Networks on Edge Devices

Edge computing is the practice of having compute power physically closer to the point of data generation, usually an IoT device. With edge computing used instead of cloud computing, there is lower latency and reduced cost. Data sovereignty is also ensured as all the data is processed at the location it is collected, making sure that no sensitive data can be hacked from the cloud servers [33].

Having said that, running a neural network requires significant amount of memory and computation resources. Edge devices are limited in that aspect and are unable to support many complex and high power consuming CNNs in real-time practical use. That is why certain architectures must be specifically developed for edge usage.

One such example is the SSD MobileNet V1, a hybrid neural network with MobileNet used as the base network in combination with the single shot multibox detection (SSD) algorithm. In the earlier MobileNet section, the depth wise separable convolution layers help to reduce computational cost and power required to use Transfer Learning. Meanwhile, the later SSD section only takes one shot to detect many objects in the visual input image. This algorithm is fast and works with a variety of different scales, enabling object detection to be performed on items of various sizes in the visual input. The final layer of Non-Maximum Suppression (NMS) is then used to ensure that each object has only one bounding box. Figure 6 shows an illustration of the SSD MobileNet V1 architecture [34].

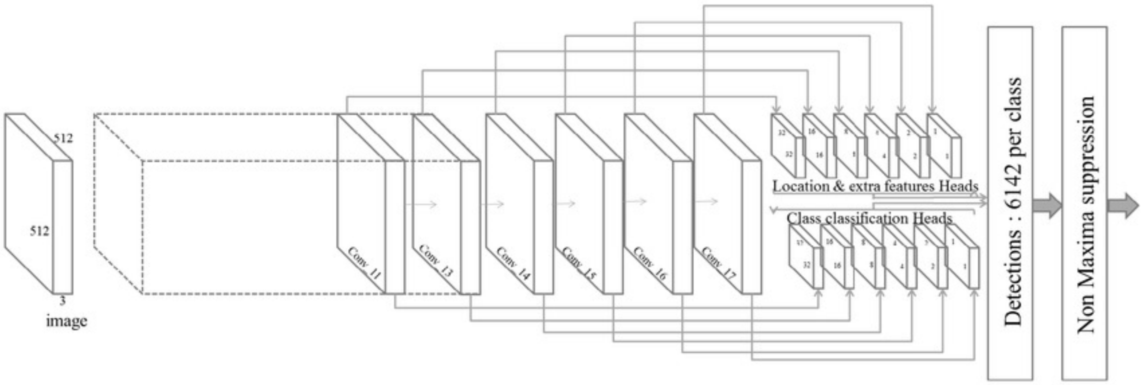

512
512
3
image
Conv_11
Conv_13
Conv_14
Conv_15
Conv_16
Conv_17
Location & extra features Heads
Class classification Heads
Detections : 6142 per class
Non Maxima suppression

FIGURE 6: SSD MOBILENET V1 ARCHITECTURE

Chapter 3
Methodology

3.1 Current Standard Operating Procedure

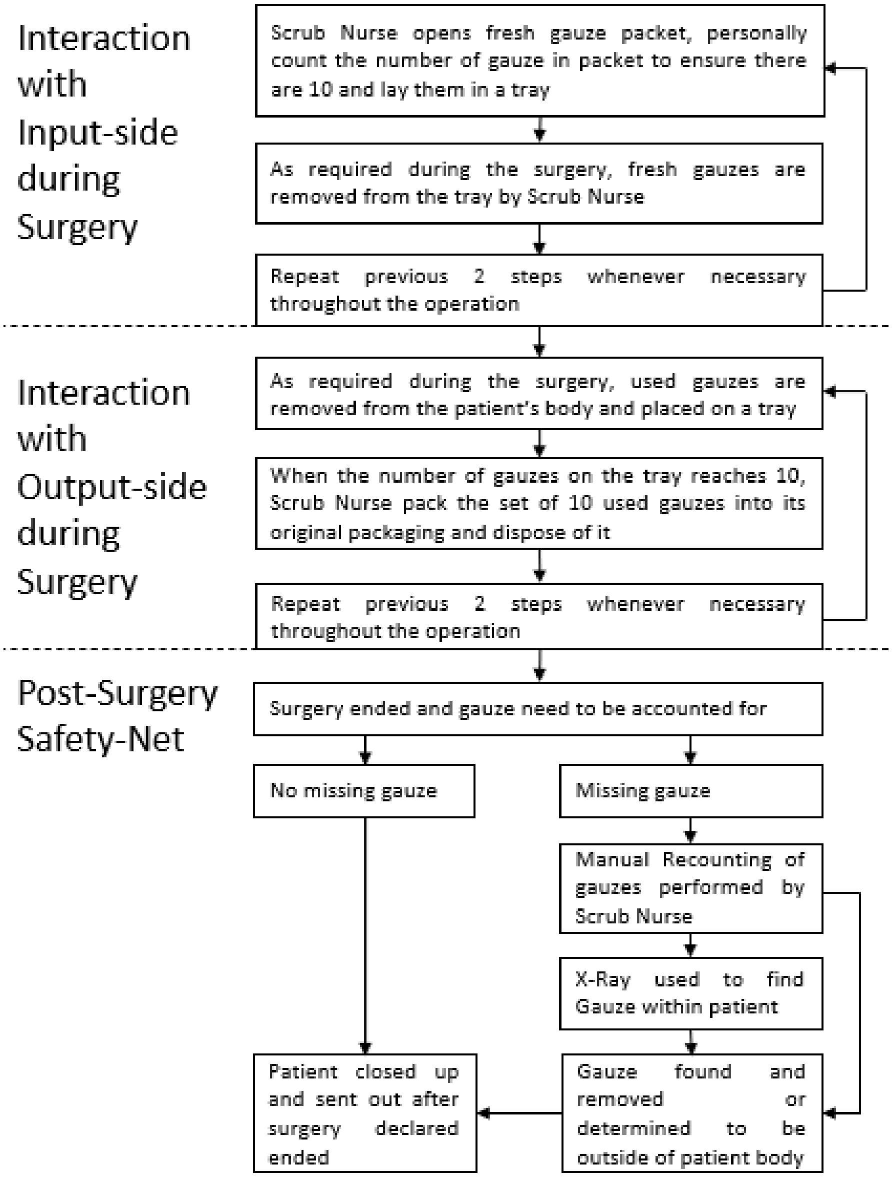

Interaction with Input-side during Surgery
Scrub Nurse opens fresh gauze packet, personally count the number of gauze in packet to ensure there are 10 and lay them in a tray
As required during the surgery, fresh gauzes are removed from the tray by Scrub Nurse
Repeat previous 2 steps whenever necessary throughout the operation
Interaction with Output-side during Surgery
As required during the surgery, used gauzes are removed from the patient's body and placed on a tray
When the number of gauzes on the tray reaches 10, Scrub Nurse pack the set of 10 used gauzes into its original packaging and dispose of it
Repeat previous 2 steps whenever necessary throughout the operation
Post-Surgery Safety-Net
Surgery ended and gauze need to be accounted for
No missing gauze
Missing gauze
Manual Recounting of gauzes performed by Scrub Nurse
X-Ray used to find Gauze within patient
Patient closed up and sent out after surgery declared ended
Gauze found and removed or determined to be outside of patient body

FIGURE 7: CURRENT GAUZE ACCOUNTING FLOWCHART

In SingHealth Group and other Singapore public hospitals, Standard Operating Procedures (SOPs) are in place to prevent Gossypiboma. The current procedure, illustrated in Figure 7, relies on the systematic accounting of gauzes in sets of 10’s as they are used throughout the operation. Nevertheless, human error persists even in the face of a precise and extensive SOP.

Tackling the problem of human error, an AI Computer Vision system that will not get tired and possess high accuracy in counting the gauzes is ideal to be used in accounting for the gauzes throughout surgeries.

While it is likely that this new system can completely replace humans in the counting of the surgical gauzes in due time, at the current moment, this system is to assist and not replace the Scrub Nurses in the accounting of gauzes. With that in mind, it is very important that the new SOP that involve the use of this new AI system must be as like that of the old SOP, for Scrub Nurses to experience minimal difficulty in the transitioning.

3.2 New Standard Operating Procedure

Thus, with Figure 8 illustrating the new SOP, the only major changes are that the Scrub Nurse must press a button before opening a fresh pack of gauze and another button after the collection of a set of 10 used gauzes.

The displayed items from the system onto the TV screen in the Operation Room are all designed to be very intuitive – ‘Gauze In’ accounts for all the gauzes that entered the operation table, ‘Gauze Out’ accounts for all the used gauzes that left the operation table and ‘Gauze In Play’ accounts for all the gauzes that are currently residing within the patient. A History Table is also at the bottom of the window to show the latest 2 actions performed on the ‘Gauze In’ and ‘Gauze Out’ counts with their relevant timestamps. This shows and remind the Scrub Nurses of the changes they have enacted to the counts, which may be very helpful when fatigue is high within the surgical staff during long surgeries.

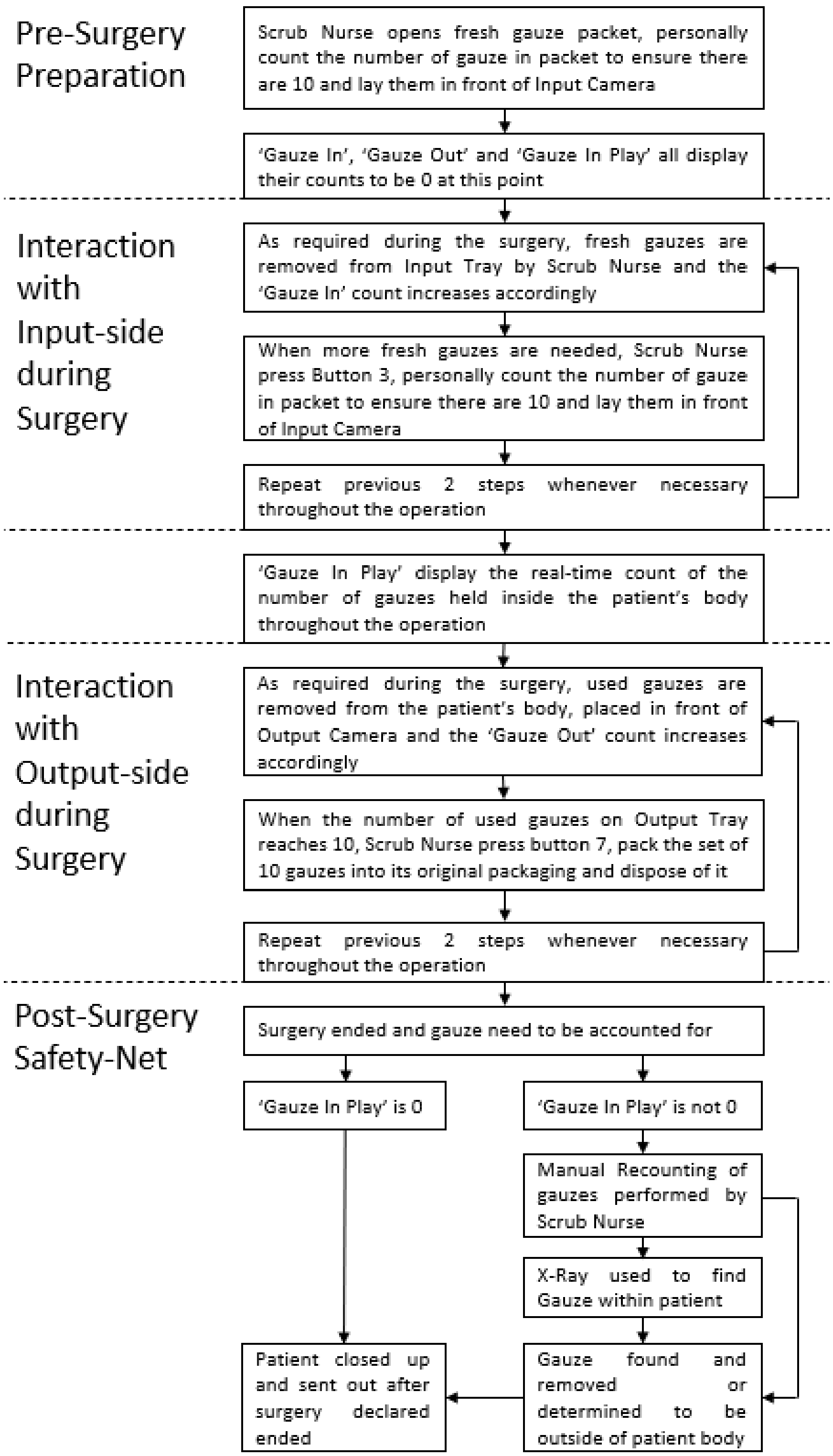

Pre-Surgery Preparation
Scrub Nurse opens fresh gauze packet, personally count the number of gauze in packet to ensure there are 10 and lay them in front of Input Camera
'Gauze In', 'Gauze Out' and 'Gauze In Play' all display their counts to be 0 at this point
Interaction with Input-side during Surgery
As required during the surgery, fresh gauzes are removed from Input Tray by Scrub Nurse and the 'Gauze In' count increases accordingly
When more fresh gauzes are needed, Scrub Nurse press Button 3, personally count the number of gauze in packet to ensure there are 10 and lay them in front of Input Camera
Repeat previous 2 steps whenever necessary throughout the operation
'Gauze In Play' display the real-time count of the number of gauzes held inside the patient's body throughout the operation
Interaction with Output-side during Surgery
As required during the surgery, used gauzes are removed from the patient's body, placed in front of Output Camera and the 'Gauze Out' count increases accordingly
When the number of used gauzes on Output Tray reaches 10, Scrub Nurse press button 7, pack the set of 10 gauzes into its original packaging and dispose of it
Repeat previous 2 steps whenever necessary throughout the operation
Post-Surgery Safety-Net
Surgery ended and gauze need to be accounted for
'Gauze In Play' is 0
'Gauze In Play' is not 0
Manual Recounting of gauzes performed by Scrub Nurse
X-Ray used to find Gauze within patient
Patient closed up and sent out after surgery declared ended
Gauze found and removed or determined to be outside of patient body

FIGURE 8: NEW GAUZE ACCOUNTING MAIN FLOWCHART

The new AI system must be versatile yet intuitive. In one likely scenario, as with all systems that deal with user interaction, there is a possibility that the Scrub Nurse will incorrectly press a button to cause an incorrect gauze count for either or both the ‘Gauze In’ and ‘Gauze Out’ count. When this happens, a procedure for him or her to rectify the wrong count is needed.

In another scenario, very rarely there are manufacturing defects in the number of fresh gauzes within one packet, such as getting 9 or 11 gauzes instead of 10. When this happens, the current hospital procedure is to isolate this faulty packet and send it back to the manufacturer. However, now with a smart system in place, it can count keep track of the number of gauzes even with irregular numbers in the packets.

Figures 9 and 10 illustrate that procedure.

Looking into the input stream’s data, the Scrub Nurse can use the first 4 buttons to edit it. From the left, Button 1 is used is to increase the input variable by a count of 1 from the default number of 10 with each push. The number 10 is chosen as default as that is the number of gauzes in a new packet. Button 2 decrease the input variable by a count of 1 from the default number of 10 with each push. Button 3 is pushed to increase the ‘Gauze In’ count by the input variable. Lastly, Button 4 is used is to decrease the ‘Gauze In’ count by the input variable. With each push of the Button 3 or Button 4, the input variable is reset to 10.

Similarly, to edit the output stream’s data of the ‘Gauze Out’ count, the same procedure stated to edit the input stream’s data using Button 1 to 4 can be used via Button 5 to 8 accordingly. The reason the default number 10 was also chosen is because the number of used gauzes that Scrub Nurses choose to re-packet into their original packaging after use and prior to disposal is also 10. It also helps that if both input and output stream use the same default number to edit, it will be less confusing.

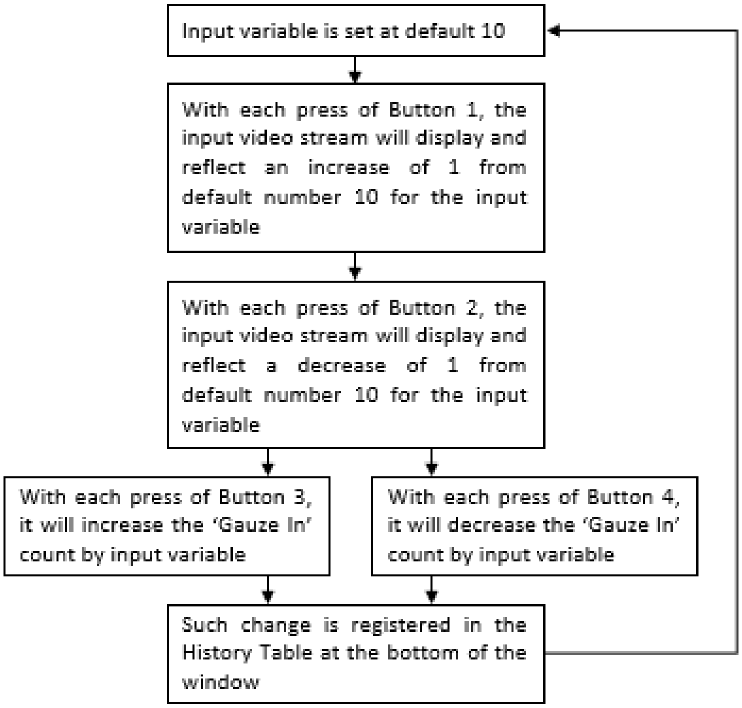

Input variable is set at default 10
With each press of Button 1, the input video stream will display and reflect an increase of 1 from default number 10 for the input variable
With each press of Button 2, the input video stream will display and reflect a decrease of 1 from default number 10 for the input variable
With each press of Button 3, it will increase the 'Gauze In' count by input variable
With each press of Button 4, it will decrease the 'Gauze In' count by input variable
Such change is registered in the History Table at the bottom of the window

Figure 9: Sub Flowchart for Input Stream Error Management

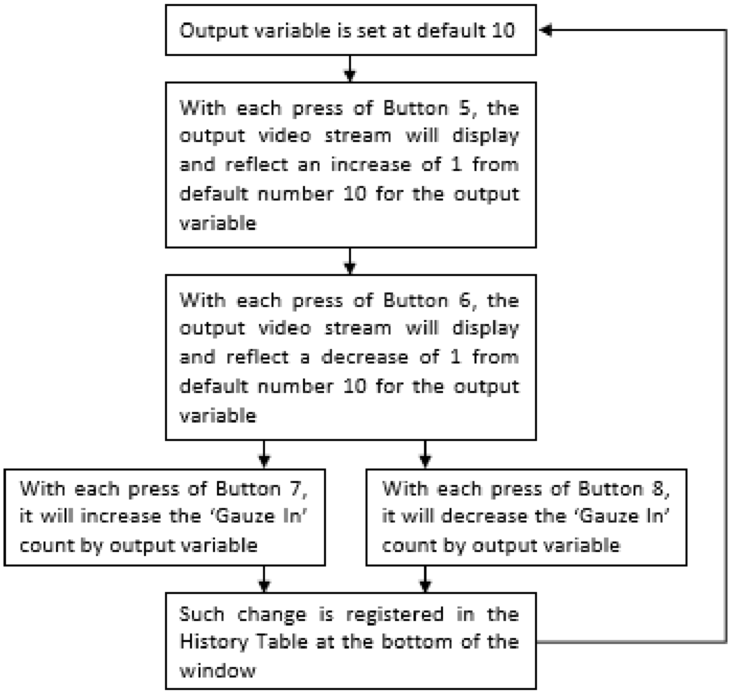

Output variable is set at default 10
With each press of Button 5, the output video stream will display and reflect an increase of 1 from default number 10 for the output variable
With each press of Button 6, the output video stream will display and reflect a decrease of 1 from default number 10 for the output variable
With each press of Button 7, it will increase the 'Gauze In' count by output variable
With each press of Button 8, it will decrease the 'Gauze In' count by output variable
Such change is registered in the History Table at the bottom of the window

Figure 10: Sub Flowchart for Output Stream Error Management

Chapter 4
Preliminary Work

4.1 Set-Up of Hardware and OS

4.1.1 Jetson Xavier NX

Before the Jetson Xavier NX can be used, the OS must be installed into it. To achieve this, the following set of procedure stated below must be executed [35] [36].

1. Using any PC or MacBook, an SD memory card formatter program, such as that from https://www.sdcard.org/downloads/formatter/, must be downloaded, executed, and installed.
2. Plug a micro-SD card into the computer then select it using the SD memory card formatter program to reformat it. It is important to ensure the size of the drive stated on the program user interface is exactly that of the size of the SD memory card so that the computer's hard drives are not accidentally reformatted. Figure 11 below shows the 64 GB SanDisk Extreme microSDXC Memory Card that proved to be successful throughout this procedure.

FIGURE 11: 64 GB SANDISK SDSQXA2-064G-GN6GN EXTREME MICROSDXC MEMORY CARD

3. An OS image to SD card flasher program, such as that from https://www.balena.io/etcher/, must be downloaded and installed into the computer.

4. Go to https://developer.nvidia.com/embedded/jetpack to download the NVIDIA JetPack SDK by selecting "Download SD Card Image" under the "JETSON XAVIER NX DEVELOPER KIT >" tab. It is not necessary to unzip the image file downloaded.
5. Using the OS image to SD card flasher program from step 3, flash the image file downloaded in step 4 into the freshly reformatted SD card. It is important that the SD card used has sufficient memory to fit the JetPack SDK for this step to be able to be executed successfully.
6. Insert the micro-SD card into its dedicated slot inside the Jetson Xavier NX Developer Kit. Figure 12 below shows the view after successful installation of the micro-SD card into the developer kit.

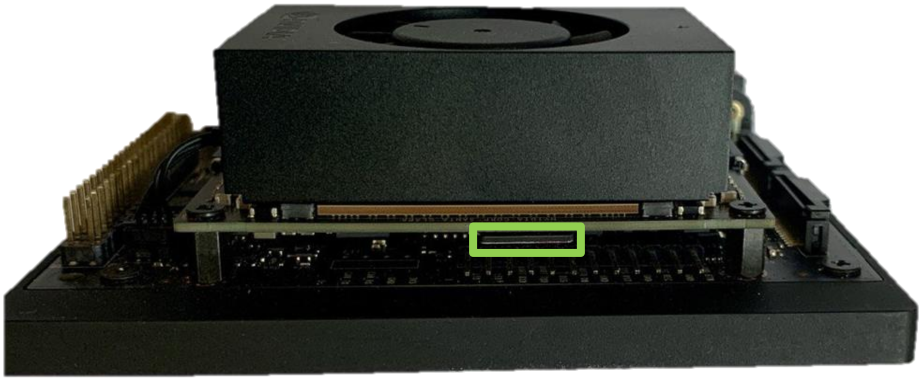


FIGURE 12: MICRO-SD IN JETSON XAVIER NX DEVELOPER KIT

7. At this stage, the developer kit can boot from the micro-SD card only. However, using the micro-SD card for both the OS and as general memory storage may cause reliability issues under long-term use. Resolving this, a NVMe PCIe M.2 SSD card must be inserted and secured underneath the Jetson Xavier NX Developer Kit. The developer kit will be configured to boot from the SSD card instead of the micro-SD card in the later steps. Figure 13 shows the view after successful installation of a Samsung 970 EVO Plus NVMe M.2 SSD 1 TB into the developer kit.

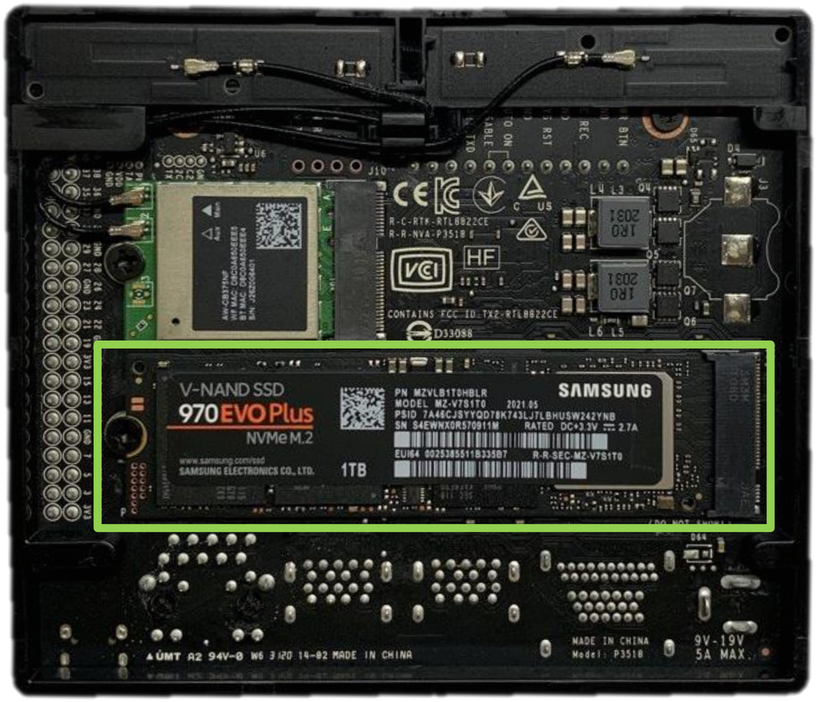


FIGURE 13: SAMSUNG 970 EVO PLUS NVME M.2 SSD 1TB IN JETSON XAVIER NX DEVELOPER KIT

8. To interact with the GUI, several devices must be connected to the developer kit beforehand. From left to right in Figure 14 below, the power plug must be connected, followed by a HDMI cable connected to a monitor or projector, then a USB dongle for the keyboard and mouse.

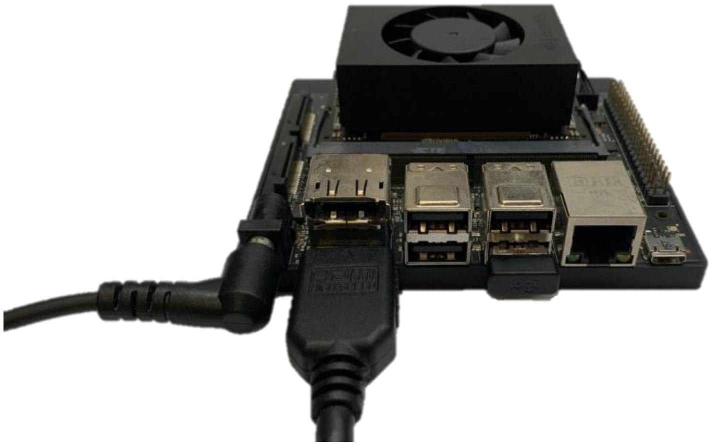


FIGURE 14: CONNECTED DEVICES TO JETSON XAVIER NX DEVELOPER KIT

9. By supplying power to the developer kit, it will automatically boot up. With the first boot, the Ubuntu-based system will prompt the user for the system configuration. These are simple tasks such as accepting the Terms and Conditions of NVIDIA End User License Agreements as well as setting up of Username and Password. The system will automatically restart after this step.
10. Key in the Username and Password that was registered in step 9 to log in.
11. Open the Disks application on the developer kit and select the SSD card. Then, click on the "≡" symbol to format it.
12. Still in the Disks application, select the SSD card and click on the "+" symbol to add a partition for 900 GB if a 1 TB SSD card is used. Some free space of 100 GB is kept in case other partitions need to be added afterwards. Similar proportions for new partition to free space should be selected if other SSD card sizes are used for this step. Ensure that "Ext4" type is selected in the last step during format volume, so the partition is configured to be compatible to Linux systems.
13. Open the Terminal application on the developer kit.
14. Run the following commands:
    a. $git clone https://github.com/jetsonhacks/rootOnNVME.git
    b. $cd rootOnNVMe
    c. $./copy-rootfs-ssd.sh
    d. $./setup-service.sh
15. Restart the developer kit and it will boot from SSD from this point onwards.

4.1.2 Raspberry Pi Zero W

Moving on to the Raspberry Pi Zero W, the following steps must be executed to install the Raspberry Pi OS into it:

1. Plug a micro-SD card into the computer then select it using the same SD memory card formatter program from earlier to reformat it. It is important to ensure the

size of the drive stated on the program user interface is exactly that of the size of the SD memory card so that the computer's hard drives are not accidentally reformatted. Figure 15 below shows the 16 GB SanDisk Edge microSDHC Memory Card that proved to be successful throughout this procedure.

FIGURE 15: 16 GB SANDISK EDGE MICROSDHC MEMORY CARD

2. Go to https://www.raspberrypi.com/software/ to download, install and execute the Raspberry Pi Imager.
3. Select 'Choose OS' to be 'Raspberry Pi OS (32-bit)'.
4. Select 'Choose Storage' to be the 16GB micro-SD card.
5. Select 'Write' to write the OS into the micro-SD Card.
6. Once done, remove the micro-SD card and insert it into the Raspberry Pi Zero W.
7. To interact with the GUI, several devices must be connected to the developer kit beforehand. From left to right in Figure 16, the mini-HDMI port must be connected to a mini-HDMI to HDMI adapter then to a HDMI cable that is connected to a monitor or projector. Following, the centre port is to be connected to a micro-USB to USB adapter then to a USB dongle for the keyboard and mouse. Finally, the power plug must be connected.

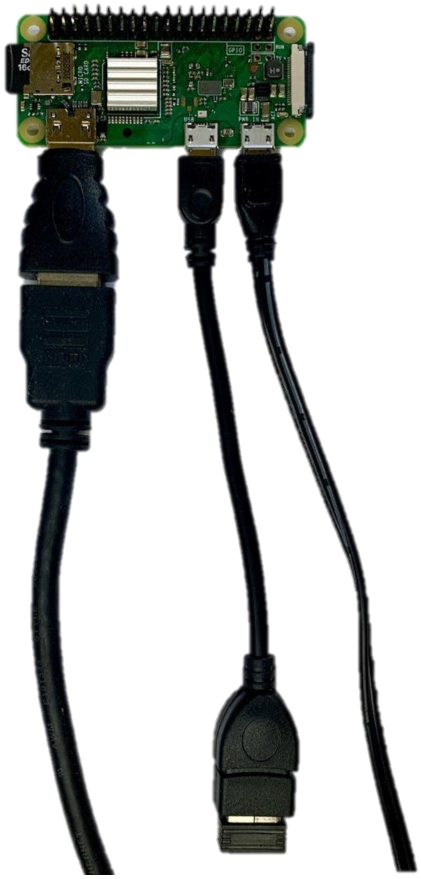


FIGURE 16: CONNECTED DEVICES TO RASPBERRY PI ZERO W

8. By supplying power to the Raspberry Pi Zero W, it will automatically boot up
9. Open Terminal
10. Enter the command:
    a. $ sudo rasp-config
11. Select '1 Change User Password' and change the password for the Pi accordingly

4.2 Set-Up of Software

4.2.1 Jetson Xavier NX

Several software and their support packages must be installed before development work can be performed. The following instructions are to be carried out on the developer kit:

1. Start the inbuilt fan at 100 revolutions per minute (RPM) to improve performance and prevent any possible heat damage during the later program installations, which can be intensive. Enter this command in the Terminal:
    a. $sudo sh -c 'echo 100 > /sys/devices/pwm-fan/target_pwm'
2. Install Visual Studio Code (VS Code) integrated development environment (IDE) for developers to create the software necessary for the system via the commands:
    a. $git clone https://github.com/JetsonHacksNano/installVSCode.git
    b. $cd installVSCode
    c. $./installVSCode.sh
    d. $code
3. With step 2d, VS Code will be open. Using its User Interface, select the extension tab, afterwards "Python", then install it. This will be the main programming language utilised for this project. Figure 17 below shows a screenshot of the selection.

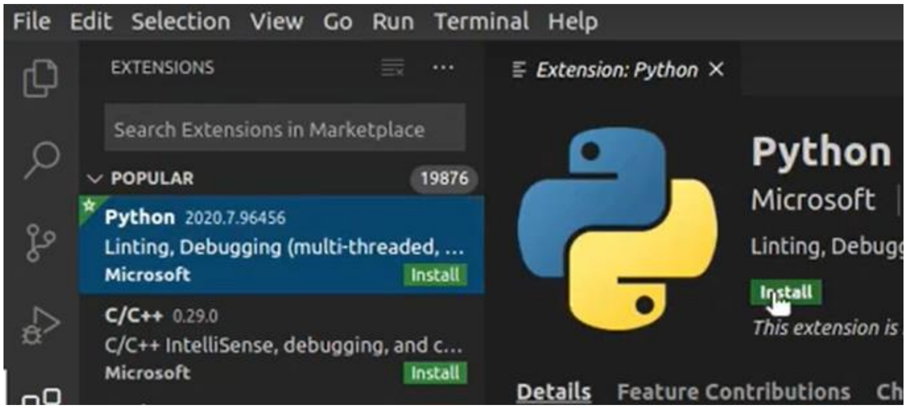


FIGURE 17: SCREENSHOT OF PYTHON INSTALLATION INTO VS CODE

4. Install pip3, the official package manager for Python 3 with the commands:
    a. $sudo apt install python3-pip
5. Install jetson-inference project, a deep learning training guide and base project on the NVIDIA platform provided by NVIDIA Jetson Developers. Take note to additionally select PyTorch to be downloaded in step g. [37]:

    a. $ sudo apt-get update
    b. $ sudo apt-get install git cmake libpython3-dev python3-numpy
    c. $ git clone --recursive https://github.com/dusty-nv/jetson-inference
    d. $ cd jetson-inference
    e. $ mkdir build
    f. $ cd build
    g. $ cmake ../
    h. $ make -j$(nproc)
    i. $ sudo make install
    j. $ sudo ldconfig

6. Install GStreamer framework, an open-source pipeline-based multimedia framework for media processing:
    a. $sudo apt-get install libgstreamer1.0-0 gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-doc gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-qt5 gstreamer1.0-pulseaudio

4.2.2 Raspberry Pi Zero W

Moving on to the Pi, the following steps must be executed to install the required GStreamer framework into it:

1. Go to Terminal
2. Run the following:
    a. $sudo apt-get install gstreamer1.0-tools

4.3 Set-Up of Electrical Components

A simple button interface was developed for User Interaction purposes with the system. The electrical schematic for the button interface and all connected devices to the developer kit are illustrated below (Figure 18).

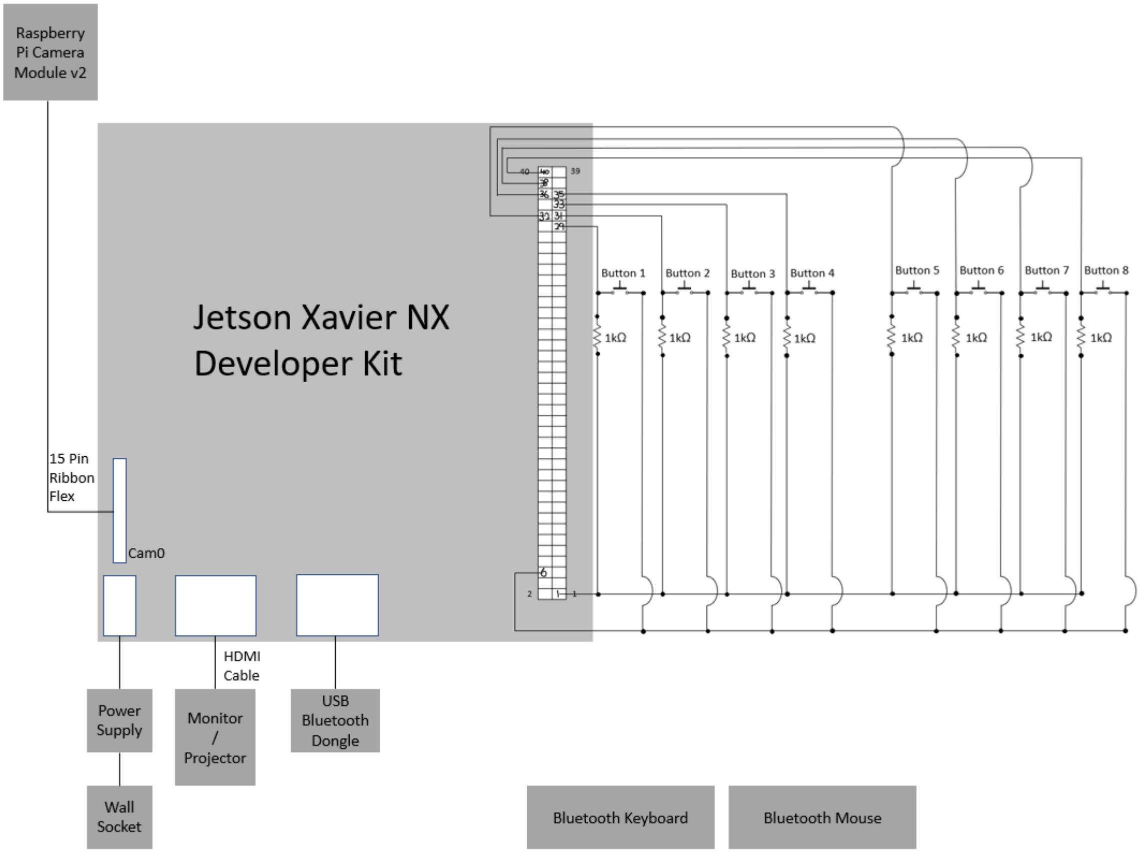


FIGURE 18: ELECTRICAL SCHEMATIC FOR DEVELOPER KIT

Figure 19 shows the real-life view of the electrical set-up for the developer kit.

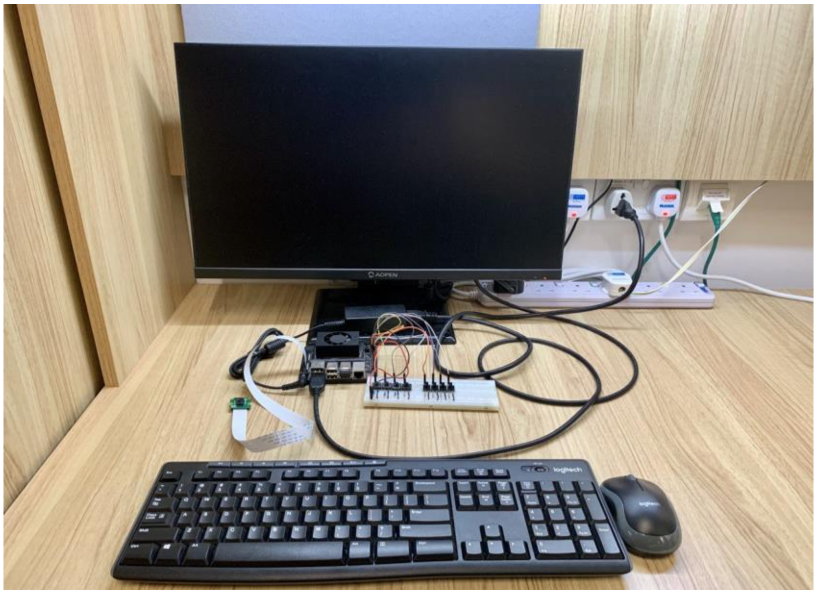


FIGURE 19: REAL-LIFE VIEW OF DEVELOPER KIT'S CONNECTED ELECTRICAL COMPONENTS

Similarly for the Raspberry Pi Zero W, there are also connected devices required for use during development work. Figure 20 illustrates the Pi's electrical schematic.

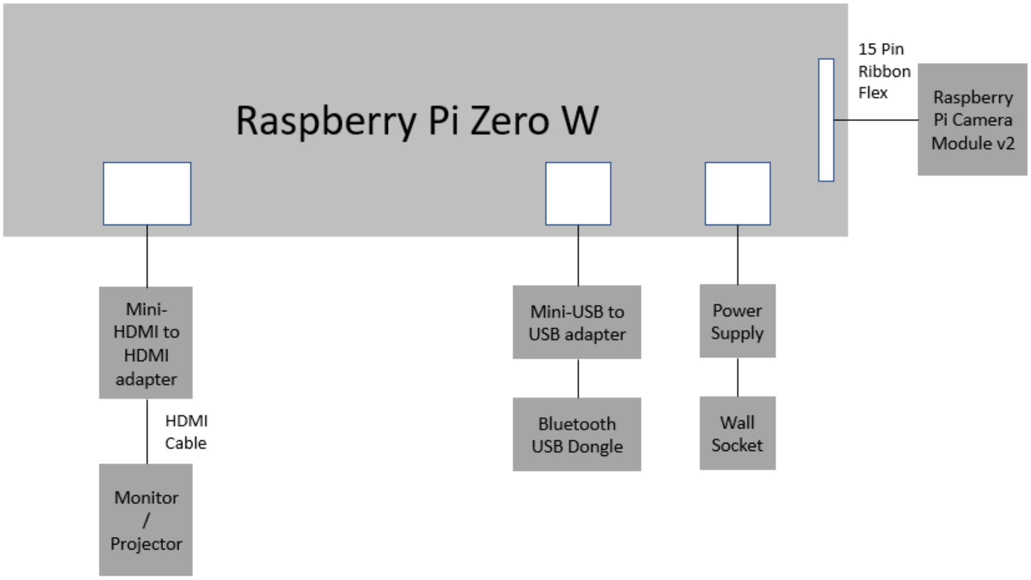


FIGURE 20: ELECTRICAL SCHEMATIC FOR PI

Figures 21 and 22 shows the real-life and zoom-in view of the Pi’s electrical set-up.

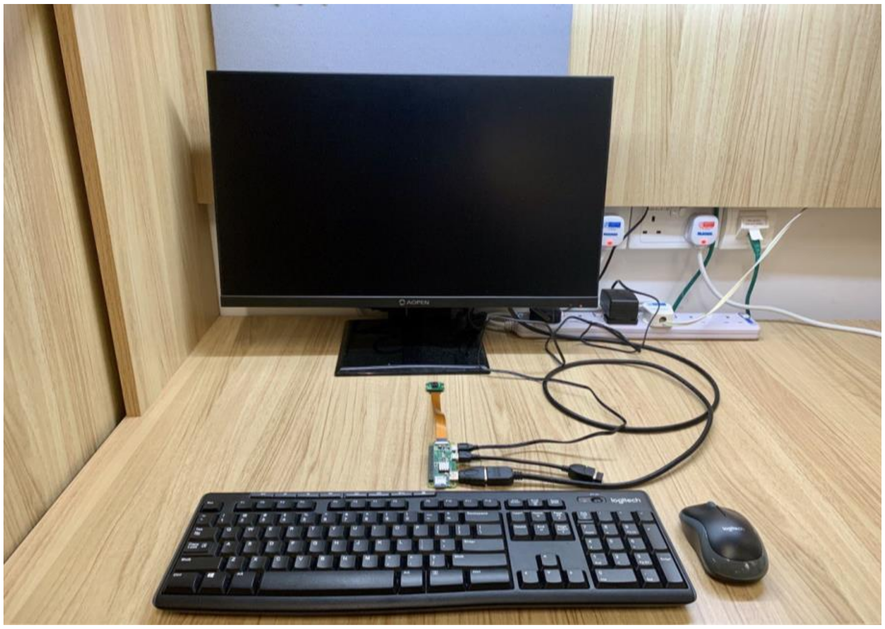


FIGURE 21: REAL-LIFE VIEW OF ELECTRICAL COMPONENTS FOR PI

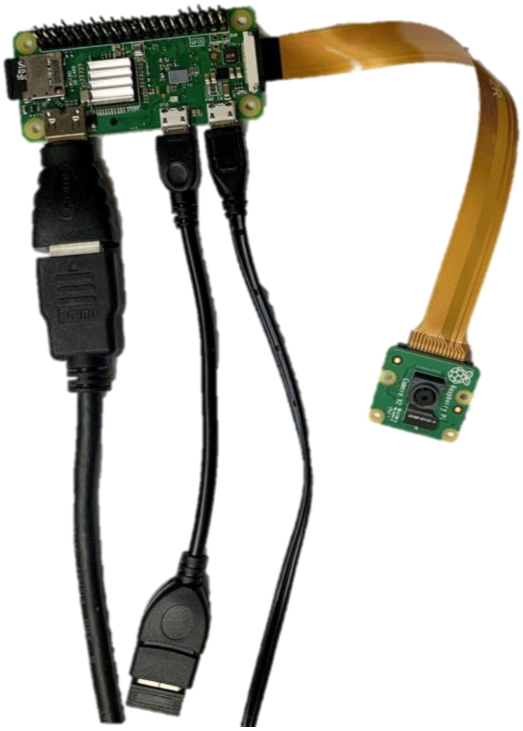


FIGURE 22: ZOOM-IN VIEW OF PI’S CONNECTED ELECTRICAL COMPONENTS

4.4 Set-Up of Transmission Protocols

For video feeds to be transferred from the Raspberry Pi Zero W to the developer kit, control of the Raspberry Pi Zero W must be given to the Jetson Xavier NX, and a common Wi-Fi network must be established between the two.

4.4.1 Jetson Xavier NX

Operation Theatres are not expected to have Wi-Fi within. Thus, an appropriate solution is to have the Jetson Xavier NX set-up a hotspot within the Operation Theatre for data transfer via the Wi-Fi protocol to be possible.

The following steps were executed on the Jetson Xavier NX Developer Kit [38] [39]:

1. Open System Settings
2. Go to the top bar and select the Network Connections icon then select 'Edit Connections…'
3. In the Network Connections window, select the '+' button at the bottom to add a connection
4. Select 'Wi-Fi' for the connection type in the new window, and select 'Create…'
5. Name the Connection name to be 'JetsonHotspot'
6. Under the Wi-Fi tab, for SSID, name the network to be 'JetsonHotspot' also
7. Select Mode to be 'Hotspot'
8. Select Device to be 'wlan0(D8:F2:CA:B5:AF:59)', the default for the wireless card on board the Jetson
9. Under IPv4 Settings tab, ensure the Method has 'Shared to other computers' selected
10. Similarly, under IPv6 Settings tab, ensure the Method has 'Shared to other computers' selected

11. Under General tab, ensure ‘Automatically connect to this network when it is available’ and ‘All users may connect to this network’ are selected, so that the hotspot is automatically set-up whenever the developer kit is booted on
12. Under Wi-Fi Security, select ‘WPA & WPA2 Personal’ and set the password for this hotspot network accordingly
13. Select ‘Save’ at the bottom of the Editing Hotspot window
14. Go to the top bar and select the Network Connections icon then select ‘Create New Wi-Fi Network’
15. Within the new window, under Connection select ‘JetsonHotspot’ and select ‘Create’
16. Connection is now established with the JetsonHotspot network
17. Complete the steps in section 4.4.2 Raspberry Pi Zero W before continuing the following steps below
18. Go to Terminal
19. Enter the command to check if the Pi and the developer kit can communicate with each other:
    a. $ping 10.42.0.232
20. Select ctrl-C on the keyboard to stop the ping
21. Go to Remmina Remote Desktop Client program
22. Select the ‘+’ icon
23. Within the new Remote Desktop Preference, select the Protocol to be ‘SSH-Secure Shell’
24. Under Server, enter the IP address. For the prototype, ‘10.42.0.232’ is keyed in here.
25. Under User name, key in ‘pi’
26. Under User password, key in the Pi’s password accordingly
27. Select ‘Save and Connect’ at the bottom

28. A new window with the IP address as the title will pop-up. This shows that the developer kit now has full remote control over the Pi.
29. Within this new window, key in the following code:
    a. $raspivid -t 0 -w 1280 -h 720 -fps 30 -b 2000000 -awb auto -n -o - | gst-launch-1.0 -v fdsrc ! h264parse ! rtph264pay config-interval=1 pt=96 ! gdppay ! tcpserversink host=0.0.0.0 port=8554
30. Video stream from the Pi is now sent to the developer kit via the Wi-Fi protocol

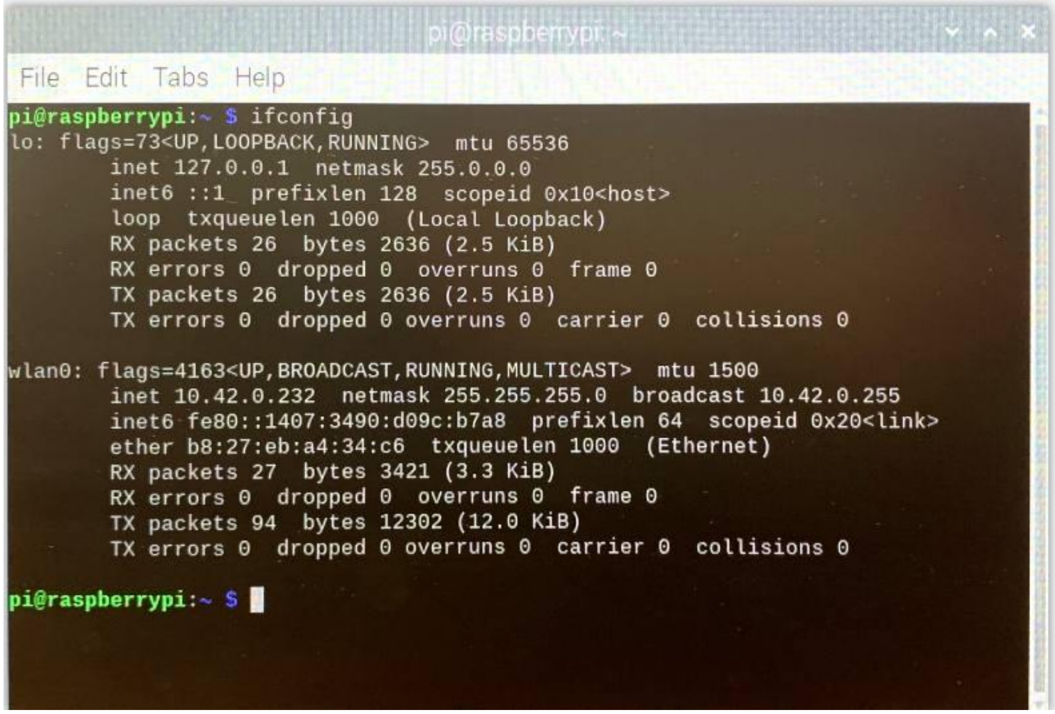
4.4.2 Raspberry Pi Zero W

The following steps were executed on the Raspberry Pi Zero W:

1. Open Terminal
2. Enter the command:
    a. $ sudo rasp-config
3. Select ‘2 Network Options’
4. Select ‘N2 Wireless LAN’, and key in ‘JetsonHotspot’ under ‘Please enter SSID’ line and select ‘<Ok>’
5. Under ‘Please enter passphrase. Leave it empty if none.’, enter the password to JetsonHotspot accordingly, and select ‘<Ok>’
6. Back to the raspi-config window, go to ‘5 Interfacing Options’, select ‘P1 Camera’ then select ‘<Yes>’ to enable the camera interface
7. Select ‘<Ok>’ to get back to the raspi-config window
8. Back to the raspi-config window, go to ‘5 Interfacing Options’ again, select ‘P2 SSH’ then select ‘<Yes>’ to enable the SSH server. This will take around 10 seconds to complete.
9. Select ‘<Ok>’ to get back to the rasp-config window
10. Select ‘<Finish>’ to exit the window
11. Enter the command in Terminal:

   a. $ifconfig

12. Record down the IP address of the Pi, stated within the paragraph for wlan0, in the second line, immediately after the word inet. For the prototype set, the IP address was 10.42.0.232. Figure 23 below shows a screenshot of this.



```
pi@raspberrypi:~ $ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 26  bytes 2636 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 2636 (2.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.232  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::1407:3490:d09c:b7a8  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:a4:34:c6  txqueuelen 1000  (Ethernet)
        RX packets 27  bytes 3421 (3.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 94  bytes 12302 (12.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $
```

FIGURE 23: SCREENSHOT OF IP ADDRESS IN TERMINAL

4.5 Set-Up of Structural Components

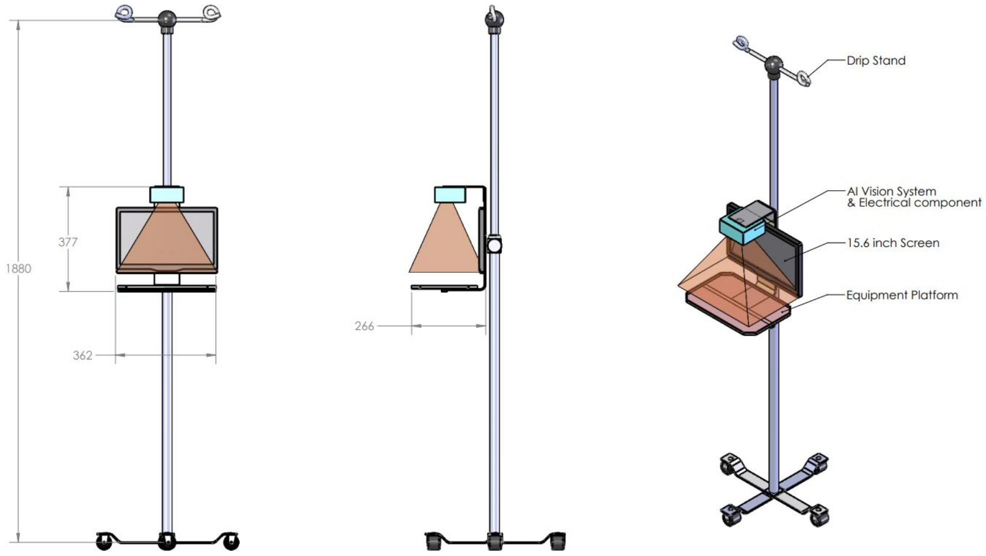


FIGURE 24: RETROFIT DESIGN ON DRIP-STAND

For the system to be authorized for use within Operation Theatre, it needs to comply to medical-grade sterilization standards. After several discussions with Scrub Nurses and Doctors with regards to workflow and sterilization requirements, a drip-stand-based retrofit design was deemed to be optimal and selected. The above Figure 24 shows the expected structure that the system is mounted onto.

Each system has two drip-stands. One holds the Raspberry Pi Zero W to track clean gauzes in the 'Gauze In' video stream, and another holds the Jetson Xavier NX Developer Kit to track the used gauzes in the 'Gauze Out' video stream.

With this final set-up, the preliminary work was completed, and the team proceeded to the development of an integrated solution.

Chapter 5
Development of an Integrated Solution

To achieve a successful solution to the unique problem of gauze management in a surgical setting, specialised software must be developed, and multiple design prototypes must be tried and tested.

First, a high-speed high-accuracy Computer Vision model, suitable for use on the Jetson Platform need to be trained using specially curated datasets and then integrated into the program. Second, post-processing computations for further improvement in object detection accuracy and user interactions must also be incorporated into the program pipeline. Third, the User Interface design need to be iterated through till an intuitive design for a good User Experience to be achieved.

5.1 Creation of Object Detection Capability

For the software to be able to locate objects in a scene and identify them, the following steps must be carried out.

5.1.1 Preparation of Dataset

Before any annotation can be done, the raw dataset of images must first be collected.

In the previous two years of this project, using smartphone cameras, 1000 images of gauzes were captured and collected. 133 of the 1000 were original images and the remaining 867 were artificially augmented images produced from the 133 original images as basis.

Both traditional augmentation methods of position augmentation and modern methods of colour augmentation were performed. Position augmentations include rotation and flipping of the images, while colour augmentation include change in brightness, contrast, saturation, and hue from the base image. These are effective methods to increase the size of the dataset significantly.

Figure 25 shows an example of one original image and Figure 26 show its augmented additions.

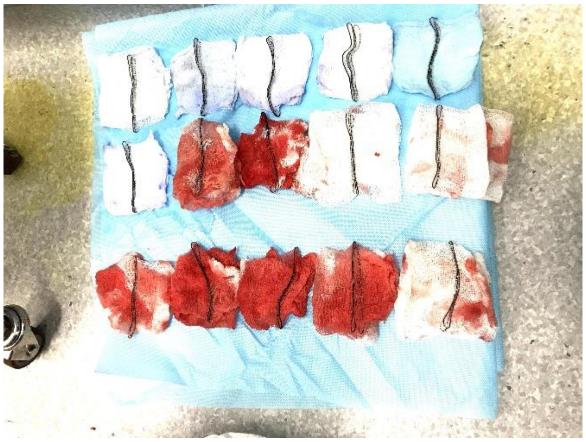


FIGURE 25: ORIGINAL IMAGE

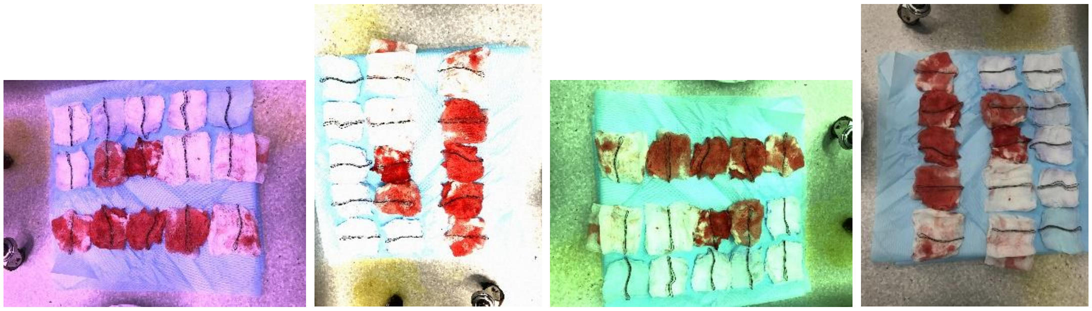


FIGURE 26: AUGMENTED IMAGES

These 1000 images were uploaded into Computer Vision Annotation Tool (CVAT) platform for image data labelling and annotation.

5.1.2 Computer Vision Annotation Tool

In the previous year's project, the annotation of the images was done in the COCO format using the COCO annotator. However, COCO format is not compatible for the Jetson

Platform to train with. The new training dataset is required to be in PASCAL VOC 1.1. format. Thus, re-labelling and re-annotation of the images must be done.

Computer Vision Annotation Tool (CVAT) is a free, open-source, online video, and image annotation tool. It was released by Intel in 2019 for the open-source community to speed up the annotation of video and image data for training Computer Vision algorithms [40]. It is an online tool that can export custom datasets in all the popular Computer Vision annotation formats. With this advantage in mind, CVAT was chosen for its high possibility of eliminating the future need for further re-annotation of the images for future researchers of this project.

Another change enacted with regards to annotation is the change from polygonal segmentation to bounding box use. The previous year's project used polygonal segmentation to annotate the gauzes in the image. This gives an edge for the software to achieve higher resolution to detect gauzes down to their precise outline. However, it is ultimately unnecessary as the software is only required to generate a drawn box around the gauze in the video feed. A bounding box used in replacement of polygonal segmentation will suffice.

It is also very helpful that using bounding boxes instead of polygonal segmentation will take less time in annotation and enable more time to be allocated for further refinement of the AI system.

Figures 27 and 28 show the screenshots and difference in polygonal segmentation using coco annotator and bounding box using the CVAT platform.

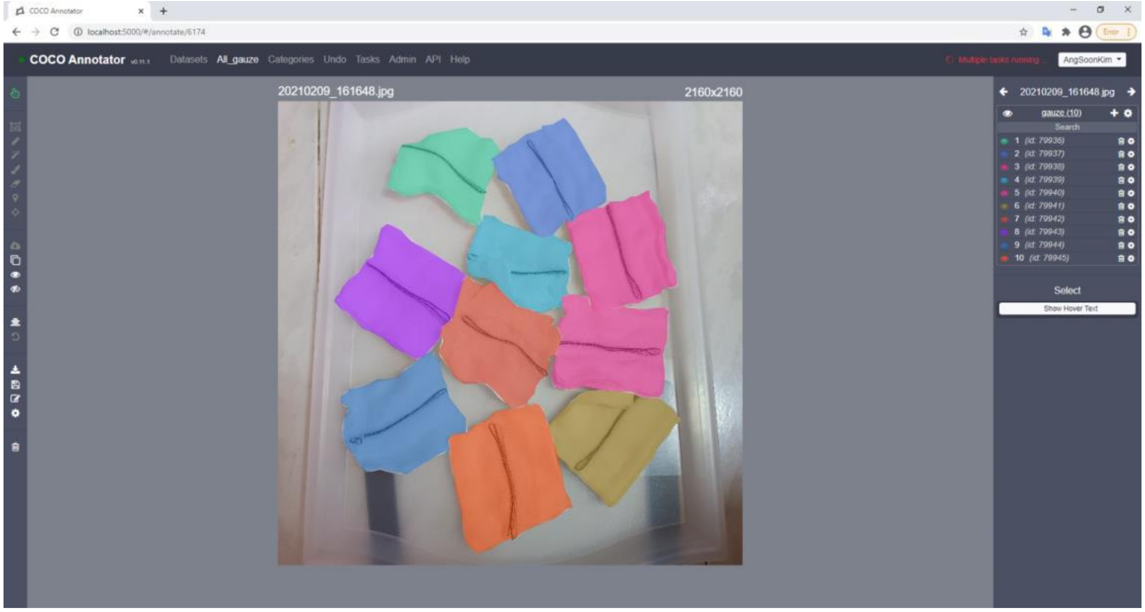


FIGURE 27: POLYGONAL SEGMENTATION USING COCO ANNOTATOR (OLD)

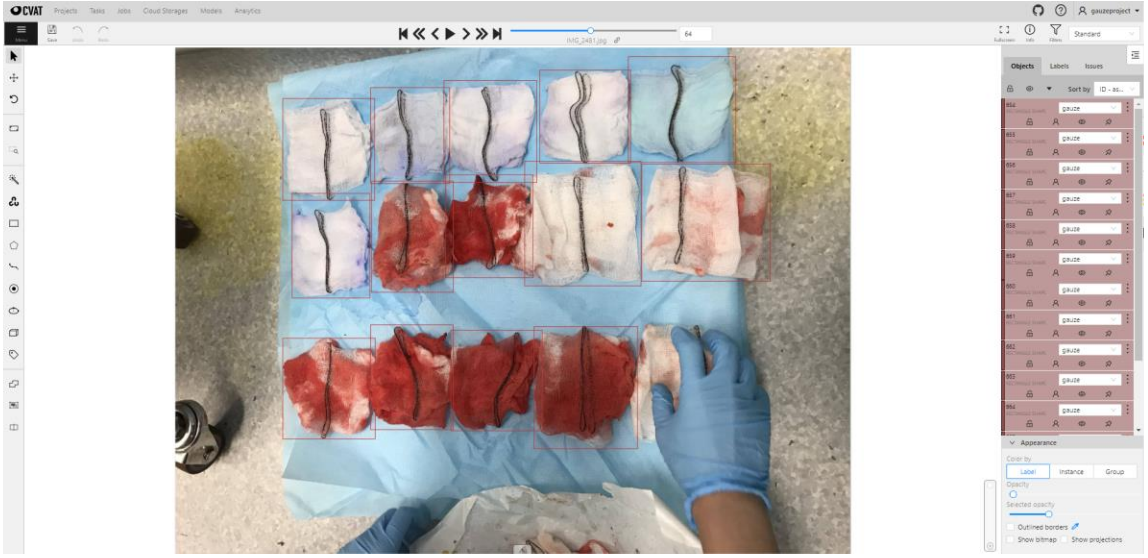


FIGURE 28: BOUNDING BOX USING CVAT PLATFORM (NEW)

The following instructions were carried out on cvat.org:

1. Create an account
2. Select 'Projects' from the top tab
3. Select '+ Create new project'

4. Enter ‘Project Gauze’ as the name
5. Select ‘Add label +’
6. Write ‘gauze’ in ‘Label name’ box
7. Select ‘Done’
8. Select ‘Submit’
9. Select ‘Projects’ from the top tab
10. Select ‘Project Gauze’
11. Select ‘+ Create new task’
12. Under Basic configuration, ‘Project Gauze Barebone Set’ was written as the name
13. 133 of the original smartphone-captured images were uploaded into this Task
14. Select ‘Submit’
15. Under ‘Project Gauze’, select ‘+ Create new task’ again
16. Under Basic configuration, ‘Project Gauze Augmented Set’ was written as the name
17. 867 of the augmented smartphone-captured images were uploaded into this Task
18. Select ‘Submit’
19. For both Tasks, click on the ‘Job’ blue hyperlink to get to the annotation screens
20. Using the keyboard, press ‘n’ and click once to locate the top left corner and click another time to locate the bottom left corner of the gauze
21. Repeat the previous step till all the gauzes in the image are annotated
22. Press ‘f’ to get to the next image and step 20 to 22 are repeated till all images are annotated in both Task sets
23. Select ‘Projects’ from the top tab
24. Under ‘Project Gauze’, hover over the ‘⁝’ symbol and select ‘Export dataset’
25. Select the export format to be ‘PASCAL VOC 1.1’ format
26. Tick the ‘Save images’ box
27. Give an appropriate name to this and select ‘OK’ to download the zip file

On this platform, using the 1000 smartphone-captured images, a total of 13310 labelled objected were created for use to train the new Computer Vision model.

5.1.3 Jetson-Inference Docker Annotation Tool

As the new AI system will not be using a smartphone camera but rather the Raspberry Pi Camera Module v2, the image dataset must contain annotated images using this new camera. To the human eye, the difference in identifying gauzes in captured images using different cameras may be insignificant but to a machine, it plays a big part in the accuracy of the trained object detection model. Image resolution, default background colour, size, and width of annotated gauzes within the image and the light intensity of the picture are all factors that play a big part in the deep learning process.

During this local system annotation procedure, the pictures captured were with both fresh and stained gauzes that were dyed with acrylic paint to simulate blood in operation setting. The background of these images was set to be that of a light blue, similar to the colour of the sanitary sheet used in operations. The lighting conditions were also varied using desk lamps adjusted to different intensities, to collect more organic images at different light levels. This is critical for the object detection model to maintain high accuracies with varied brightness in different Operation Theatres.

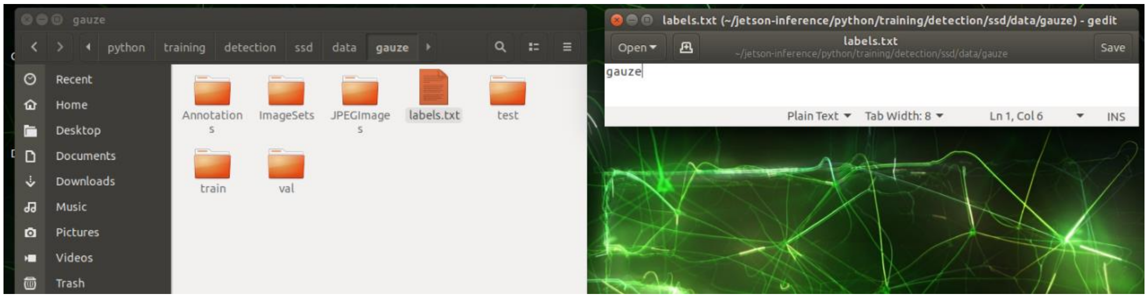
Hence, with the Jetson Xavier NX Developer Kit and using the Jetson-Inference base project's inbuilt annotation tool, images were captured using the Raspberry Pi Camera Module v2 and manually annotated within a docker environment. These annotations were then saved in PASCAL VOC 1.1 format and organized in a manner suitable for training of the neural network.

To achieve this, the following procedure was carried out:

1. To start the Jetson-Inference Docker Annotation Tool, the following commands were entered in the Terminal:

    a. $ cd jetson-inference/python/training/detection/ssd
    b. $ camera-capture csi://0

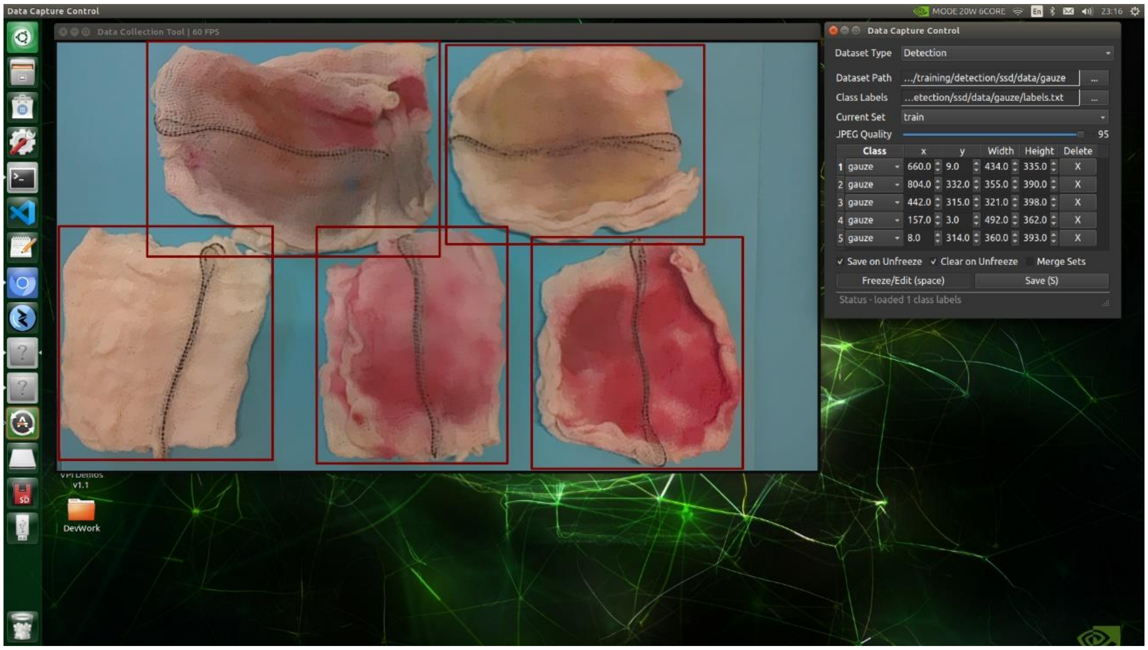
2. Within the Data Capture Control Window, under Dataset Type, select ‘Detection’
3. For Dataset Path, click the ‘...’ symbol, go to ‘jetson-inference/python/training/detection/ssd/data’ and create a new folder at this location. For this run, the folder is titled to be ‘gauze’. Thus, the Dataset Path is ‘jetson-inference/python/training/detection/ssd/data/gauze’
4. Within this new folder, create a text file titled ‘labels.txt’ and within it, it is only to have the word ‘gauze’. Figure 29 within next page illustrates this. This Figure displays multiple other folders within the ‘gauze’ folder which will be created automatically when the annotations are made.
5. Back to within the Data Capture Control Window, under Class Labels, select ‘jetson-inference/python/training/detection/ssd/data/gauze/labels.txt’
6. Select Current Set to be ‘train’, ‘val’ or ‘test’ accordingly. Maintain an 80% train to 10% validation to 10% test sub-datasets ratio while collecting this new dataset using the Jetson-Inference Docker Annotation Tool
7. Put the gauzes in front of the Raspberry Pi Camera Module v2 camera and the picture will show the gauzes in the Data Collection Tool window in real-time
8. Freeze the image by pressing the spacebar
9. Under the Data Collection Tool, use the mouse to draw bounding boxes around gauzes accordingly
10. Press ‘s’ in keyboard to save the annotations and image, and the Data Collection Tool will also unfreeze
11. Repeat Step 6 to 10 as required. A screenshot of the local system Jetson-Inference Docker Annotation Tool is displayed in Figure 30.



labels.txt (~/jetson-inference/python/training/detection/ssd/data/gauze) - gedit

```
gauze
```

FIGURE 29: SCREENSHOT OF LABELS.TXT AND ITS LOCATION



Data Capture Control

Dataset Type: Detection

Dataset Path: .../training/detection/ssd/data/gauze

Class Labels: ...etection/ssd/data/gauze/labels.txt

Current Set: train

JPEG Quality: 95

	Class	x	y	Width	Height	Delete
1	gauze	660.0	9.0	434.0	335.0	X
2	gauze	804.0	332.0	355.0	390.0	X
3	gauze	442.0	315.0	321.0	398.0	X
4	gauze	157.0	3.0	492.0	362.0	X
5	gauze	8.0	314.0	360.0	393.0	X

✓ Save on Unfreeze ✓ Clear on Unfreeze □ Merge Sets

Status - loaded 1 class labels

FIGURE 30: SCREENSHOT OF LOCAL SYSTEM JETSON-INFERENCE DOCKER ANNOTATION TOOL

5.1.4 Re-size

With smartphone-captured images and the Raspberry Pi Camera Module v2 captured images being very different in resolution, they must be resized to an appropriate resolution that the AI system is expected to deal with. Otherwise, the AI system may end up with an unnatural size bias for the gauzes, causing inaccuracies in the object detection algorithm.

The AI system is expected to have 900 by 675-pixel video streams so all images captured by the Raspberry Pi Camera Module v2 were resized to this dimension. For the dataset from CVAT, majority of the images were also resized to fit to this resolution. The remaining images were resized with the intention for the size of the gauzes within to be of similar size to that of the majority.

The following instructions were carried out on the Jetson developer kit [41]:

1. Download resize_dataset_pascalvoc ZIP package into the Downloads folder from https://github.com/italojs/resize_dataset_pascalvoc
2. Go to Downloads folder
3. Extract all contents from the ZIP package
4. Go to resize_dataset_pascalvoc-master folder
5. Put all xml and jpg files of the original dataset that is meant to be resized into 1 folder and place that folder into the resize_dataset_pascalvoc-master Folder. Rename that folder to be gauzejetsondataset in this case.
6. Go to Terminal
7. Run the command to get to the right directory:
    a. cd Downloads/resize_dataset_pascalvoc-master
8. Run the command to resize them into 900 by 675-pixel dimensions:
    a. $ python3 main.py -p gauzejetsondataset --output ./output --new_x 900 --new_y 675 --save_box_images 1
9. All the new xml and jpg files are found in the output folder inside resize_dataset_pascalvoc-master folder

5.1.5 Training Neural Network

After re-sizing, the files need to be transferred from the output folder into the jetson-inference project and arranged in a specific manner [42] [43].

In the directory of /Home/jetson-inference/python/training/detection/ssd/data, create a new folder. This folder is titled gauzejetsondataset_900by675 for this example. Within gauzejetsondataset_900by675 folder, create a new folder titled Annotations. Inside Annotations folder, place all the xml files inside. Then, within gauzejetsondataset_900by675 folder, create another new folder titled JPEGImages, where all the images for the dataset will reside. Going back to gauzejetsondataset_900by675, create another folder titled ImageSets. Within ImageSets, create another folder titled Main. Within Main folder, place 4 text files titled test.txt, train.txt, trainval.txt, and val.txt. These 4 files will contain the list of images to be used for test sub-dataset, train sub-dataset, train and validation combined sub-datasets, and validation sub-dataset respectively. Finally, within gauzejetsondataset_900by675, create a labels.txt file, which inside the file, only has the word 'gauze'. With this, the files are in the correct directories for training of the object detection model to be performed.

For this project, SSD MobileNet V1 has been selected and Transfer Learning was performed using the new custom dataset.

The following steps were taken:

1. Open the Terminal
2. Disable GUI on boot with the command. This is necessary as there will not be sufficient memory to perform training of the neural network while maintaining the graphics for the GUI.:
    a. $ sudo systemctl set-default multi-user.target
3. Restart using GUI
4. Sign-in with username and password set on the developer kit
5. Run fan at 200 RPM to prevent any possible heat damage to the developer kit during the training, with command:
    a. $ sudo sh -c 'echo 200 > /sys/devices/pwm-fan/target_pwm'

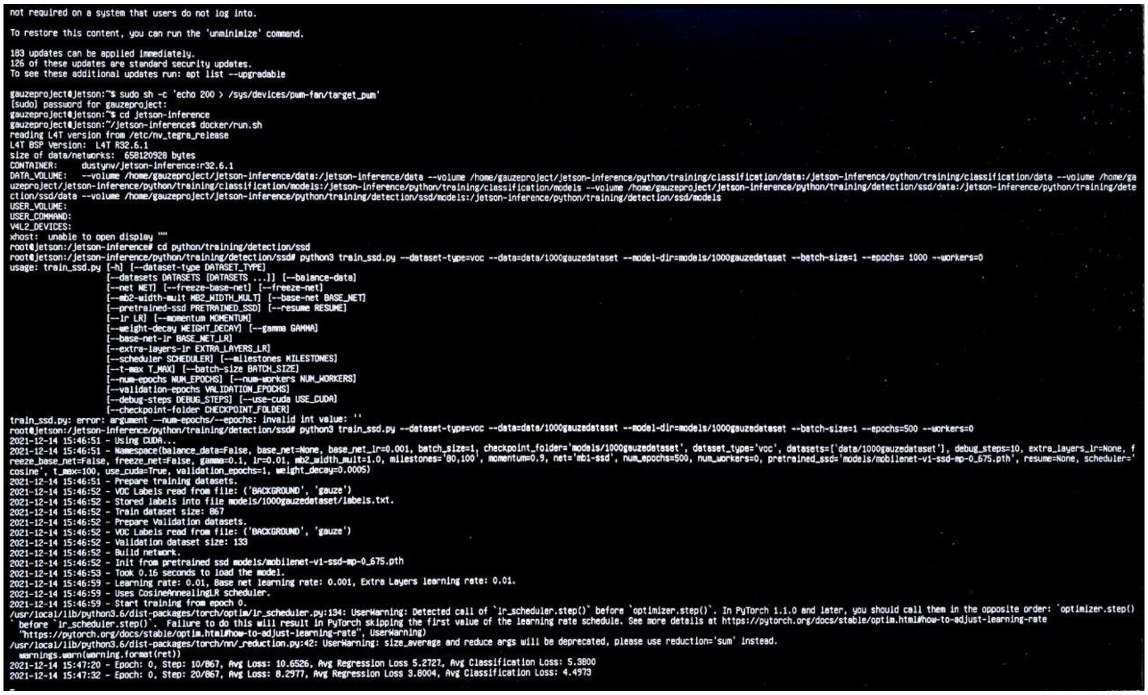
6. Start docker:
   a. $ cd jetson-inference
   b. $ docker/run.sh
7. Go to the directory containing the dataset:
   a. # cd python/training/detection/ssd
8. Run training command:
   a. # python3 train_ssd.py --dataset-type=voc --data=data/gauzejetsondataset_900by675 --model-dir=models/gauzejetsondataset_900by675 --batch-size=1 --epochs=100 --workers=0
9. Wait for training to complete. Be prepared that the training can take anywhere from a day to a few weeks depending on the dataset size (Figure 31).
10. Once training is done, press ctrl-D on keyboard. Observe the change in the symbol for the current line from '$' to '#'. This is a sign that the user has exited the docker container.
11. Enable GUI on boot with the command:
   a. $ sudo systemctl set-default graphical.target
12. Restart within 30 seconds with the command:
   a. $ shutdown -r
13. Open Terminal, start docker:
   a. $ cd jetson-inference
   b. $ docker/run.sh
14. Go to the directory containing the dataset:
   a. # cd python/training/detection/ssd
15. Convert the trained PyTorch-based model to ONNX format which is better for inference performance and has reduced inference time during deployment of the neural network:

a. # python3 onnx_export.py --model-dir=models/gauzejetsondataset_900by675

16. To test if the previous step was successful, run:

a. NET=models/gauzejetsondataset_900by675

```
detectnet --model=$NET/ssd-mobilenet.onnx --labels=$NET/labels.txt \
--input-blob=input_0 --output-cvg=scores --output-bbox=boxes \
csi://0
```



```
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

183 updates can be applied immediately.
126 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

gauzeproject@jetson:~$ sudo sh -c 'echo 200 > /sys/devices/pwm-fan/target_pwm'
[sudo] password for gauzeproject:
gauzeproject@jetson:~$ cd jetson-inference
gauzeproject@jetson:~/jetson-inference$ docker/run.sh
reading L4T version from /etc/nv_tegra_release
L4T BSP Version:  L4T R32.6.1
size of data/networks:  658120928 bytes
CONTAINER:     dustynv/jetson-inference:r32.6.1
DATA_VOLUME:   --volume /home/gauzeproject/jetson-inference/data:/jetson-inference/data --volume /home/gauzeproject/jetson-inference/python/training/classification/data:/jetson-inference/python/training/classification/data --volume /home/ga
uzeproject/jetson-inference/python/training/classification/models:/jetson-inference/python/training/classification/models --volume /home/gauzeproject/jetson-inference/python/training/detection/ssd/data:/jetson-inference/python/training/dete
ction/ssd/data --volume /home/gauzeproject/jetson-inference/python/training/detection/ssd/models:/jetson-inference/python/training/detection/ssd/models
USER_VOLUME:
USER_COMMAND:
V4L2_DEVICES:
xhost:  unable to open display ""
root@jetson:/jetson-inference# cd python/training/detection/ssd
root@jetson:/jetson-inference/python/training/detection/ssd# python3 train_ssd.py --dataset-type=voc --data=data/1000gauzedataset --model-dir=models/1000gauzedataset --batch-size=1 --epochs= 1000 --workers=0
usage: train_ssd.py [-h] [--dataset-type DATASET_TYPE]
                    [--datasets DATASETS [DATASETS ...]] [--balance-data]
                    [--net NET] [--freeze-base-net] [--freeze-net]
                    [--mb2-width-mult MB2_WIDTH_MULT] [--base-net BASE_NET]
                    [--pretrained-ssd PRETRAINED_SSD] [--resume RESUME]
                    [--lr LR] [--momentum MOMENTUM]
                    [--weight-decay WEIGHT_DECAY] [--gamma GAMMA]
                    [--base-net-lr BASE_NET_LR]
                    [--extra-layers-lr EXTRA_LAYERS_LR]
                    [--scheduler SCHEDULER] [--milestones MILESTONES]
                    [--t-max T_MAX] [--batch-size BATCH_SIZE]
                    [--num-epochs NUM_EPOCHS] [--num-workers NUM_WORKERS]
                    [--validation-epochs VALIDATION_EPOCHS]
                    [--debug-steps DEBUG_STEPS] [--use-cuda USE_CUDA]
                    [--checkpoint-folder CHECKPOINT_FOLDER]
train_ssd.py: error: argument --num-epochs/--epochs: invalid int value: ''
root@jetson:/jetson-inference/python/training/detection/ssd# python3 train_ssd.py --dataset-type=voc --data=data/1000gauzedataset --model-dir=models/1000gauzedataset --batch-size=1 --epochs=500 --workers=0
2021-12-14 15:46:51 - Using CUDA...
2021-12-14 15:46:51 - Namespace(balance_data=False, base_net=None, base_net_lr=0.001, batch_size=1, checkpoint_folder='models/1000gauzedataset', dataset_type='voc', datasets=['data/1000gauzedataset'], debug_steps=10, extra_layers_lr=None, f
reeze_base_net=False, freeze_net=False, gamma=0.1, lr=0.01, mb2_width_mult=1.0, milestones='80,100', momentum=0.9, net='mb1-ssd', num_epochs=500, num_workers=0, pretrained_ssd='models/mobilenet-v1-ssd-mp-0_675.pth', resume=None, scheduler='
cosine', t_max=100, use_cuda=True, validation_epochs=1, weight_decay=0.0005)
2021-12-14 15:46:51 - Prepare training datasets.
2021-12-14 15:46:52 - VOC Labels read from file: ('BACKGROUND', 'gauze')
2021-12-14 15:46:52 - Stored labels into file models/1000gauzedataset/labels.txt.
2021-12-14 15:46:52 - Train dataset size: 867
2021-12-14 15:46:52 - Prepare Validation datasets.
2021-12-14 15:46:52 - VOC Labels read from file: ('BACKGROUND', 'gauze')
2021-12-14 15:46:52 - Validation dataset size: 133
2021-12-14 15:46:52 - Build network.
2021-12-14 15:46:52 - Init from pretrained ssd models/mobilenet-v1-ssd-mp-0_675.pth
2021-12-14 15:46:53 - Took 0.16 seconds to load the model.
2021-12-14 15:46:59 - Learning rate: 0.01, Base net learning rate: 0.001, Extra Layers learning rate: 0.01.
2021-12-14 15:46:59 - Uses CosineAnnealingLR scheduler.
2021-12-14 15:46:59 - Start training from epoch 0.
/usr/local/lib/python3.6/dist-packages/torch/optim/lr_scheduler.py:134: UserWarning: Detected call of `lr_scheduler.step()` before `optimizer.step()`. In PyTorch 1.1.0 and later, you should call them in the opposite order: `optimizer.step()
` before `lr_scheduler.step()`.  Failure to do this will result in PyTorch skipping the first value of the learning rate schedule. See more details at https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate
  "https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate", UserWarning)
/usr/local/lib/python3.6/dist-packages/torch/nn/_reduction.py:42: UserWarning: size_average and reduce args will be deprecated, please use reduction='sum' instead.
  warnings.warn(warning.format(ret))
2021-12-14 15:47:20 - Epoch: 0, Step: 10/867, Avg Loss: 10.6526, Avg Regression Loss 5.2727, Avg Classification Loss: 5.3800
2021-12-14 15:47:32 - Epoch: 0, Step: 20/867, Avg Loss: 8.2977, Avg Regression Loss 3.8004, Avg Classification Loss: 4.4973
```

FIGURE 31: SCREENSHOT OF THE START OF TRANSFER LEARNING PROCESS

5.2 Post-processing Computations

With the object detection neural network trained, post-processing computations need to be implemented for a functional AI system. Functionalities such as integrating two video input streams into one window, button interactions and user interactions recording were implemented in this part of the program pipeline. The code for this was developed in python and to maintain access to the neural network, it was held at the specific directory of /Home/jetson-inference/python/training/detection/ssd folder.

Another modification to the jetson-inference base project was that a stronger NMS technique was applied. This technique removes the less likely bounding boxes and keep the bounding box with the best accuracy in an object detection area with multiple detections. Hence, enabling a more accurate overall count of the number of objects within the video stream. Figure 32 illustrates the problem when an inappropriate level of NMS is selected.

The following were the steps taken to modify the NMS:

1. Edited this line in /Home/jetson-inference/c/detectNet.h. With a lower float threshold, the lower the allowable overlap between boxes:
    a. int clusterDetections(Detection* detections, int n, float threshold=0.75f); to
    b. int clusterDetections(Detection* detections, int n, float threshold=0.60f);
2. Go to Terminal
3. Rebuild jetson-inference docker container to update the changes:
    a. $ cd jetson-inference
    b. $ cd build
    c. $ cmake ../
    d. In the new window popup, select all options presented to be default, but tick Pytorch option. Download Pytorch.
    e. $ make -j$(nproc)
    f. $ sudo make install

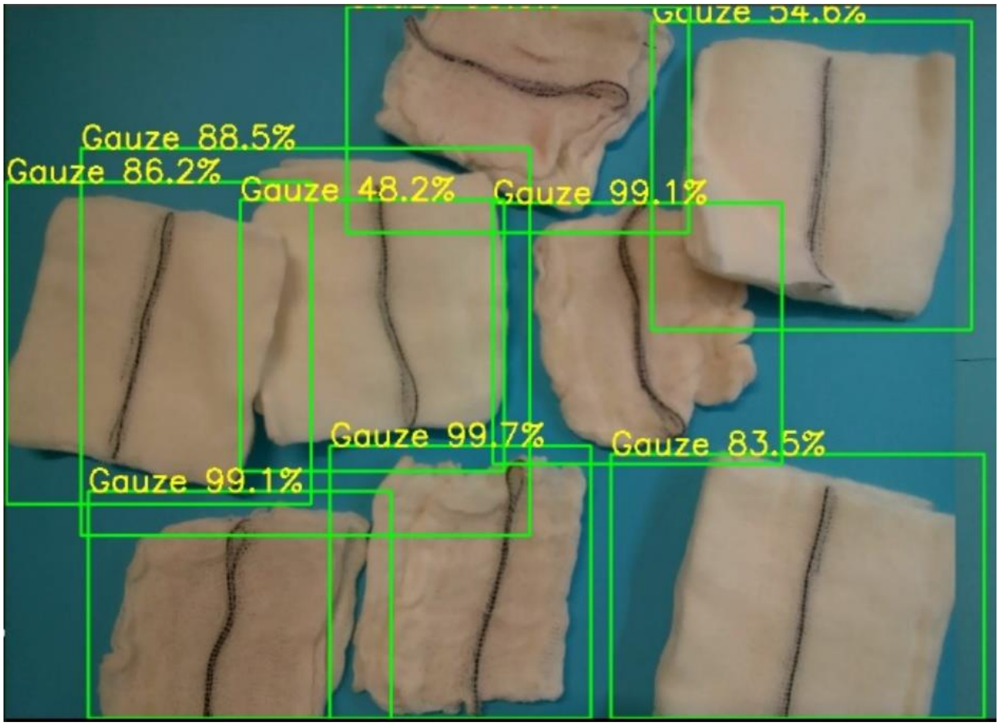


FIGURE 32: EXAMPLE OF INAPPROPRIATE LEVEL OF NMS

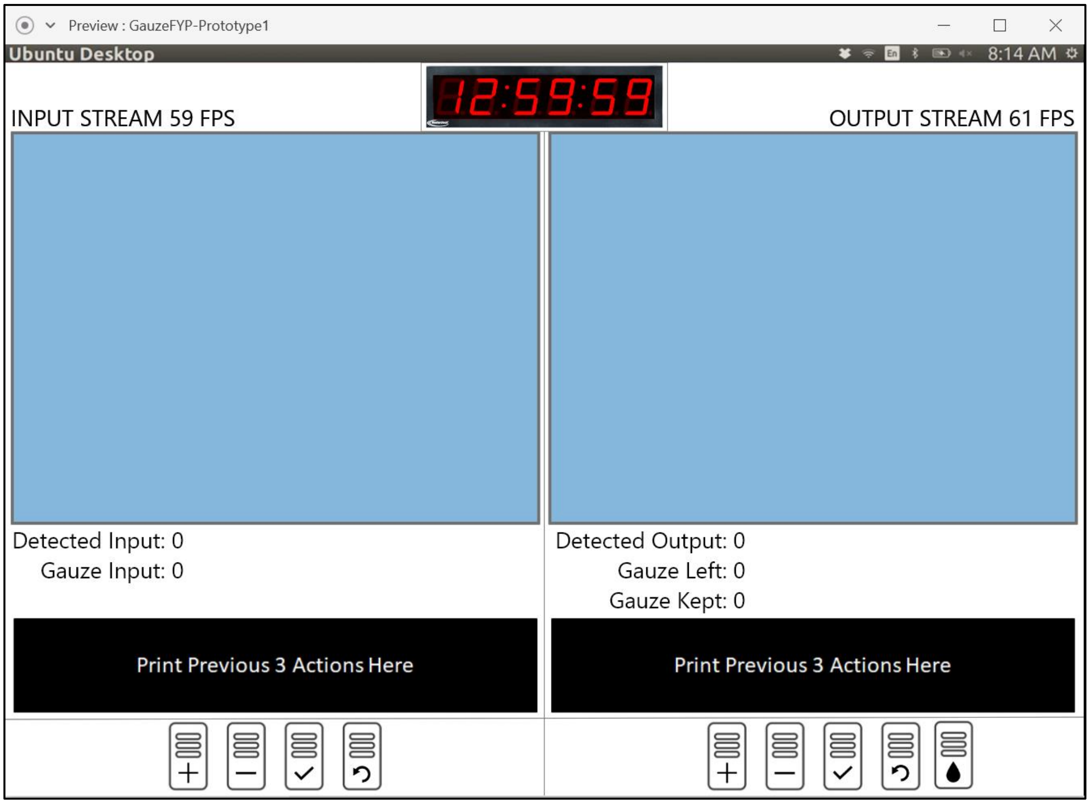
5.3 Intuitive Design and UI/UX

A product design is intuitive in the sense that the product works the way the user expects [44]. With a User Interface (UI) design that align with the Scrub Nurse's mental model, they will not be forced into a problem-solving mode to figure out what the next step is to operate this product. In an operation setting, the Scrub Nurses are already dealing with a multitude of tasks. Without an intuitive design, stress and inconvenience will be placed upon them, leading to a terrible User Experience (UX).

A good UX is necessary for this product to achieve widespread implementation across hospitals. Hence, it is paramount that the Scrub Nurses, our users for this product, need to be able to understand how to use the system without tutorial, onboarding, or other help.

To understand the Scrub Nurse and operating staff mental models, heavy user research was done. Several interviews and weekly meetings were conducted with Scrub Nurses and Doctors from SingHealth Group, Singapore General Hospital, to understand their perspectives towards using this product. Several design iterations were presented and critiqued till a final simple and most intuitive design was selected.

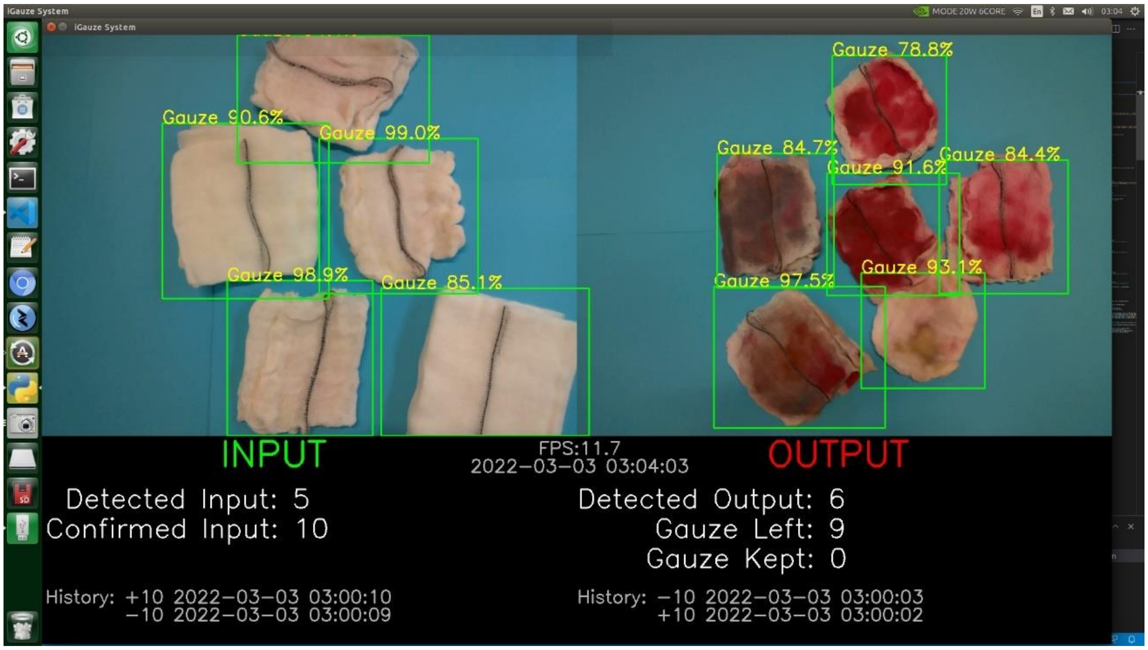
The following Figures 33, 34 and 35 shows the iterative process from a design wireframe prototype created in Adobe XD to this product's latest and optimized design.


Preview : GauzeFYP-Prototype1
Ubuntu Desktop
8:14 AM
12:59:59
INPUT STREAM 59 FPS
OUTPUT STREAM 61 FPS
Print Previous 3 Actions Here
Print Previous 3 Actions Here

Detected Input: 0
Gauze Input: 0

Detected Output: 0
Gauze Left: 0
Gauze Kept: 0

FIGURE 33: UI DESIGN IN ADOBE XD



INPUT

FPS:11.7
2022–03–03 03:04:03

OUTPUT

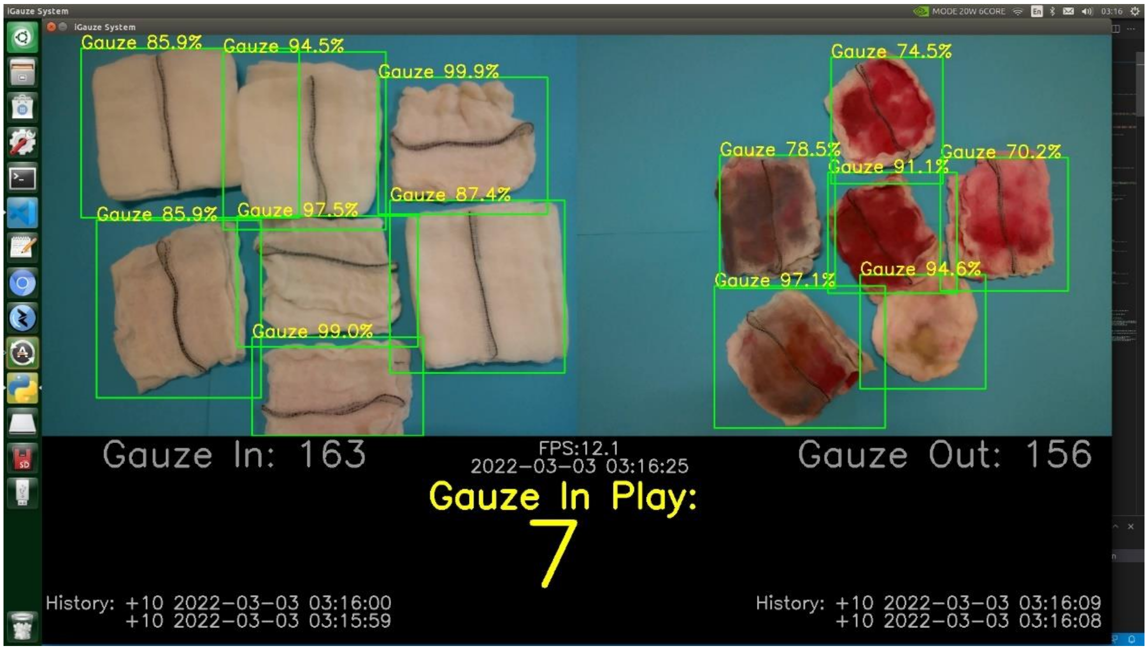
Detected Input: 5
Confirmed Input: 10

Detected Output: 6
Gauze Left: 9
Gauze Kept: 0

History: +10 2022–03–03 03:00:10
−10 2022–03–03 03:00:09

History: −10 2022–03–03 03:00:03
+10 2022–03–03 03:00:02

Figure 34: First Prototype with Functional UI



Gauze In: 163

FPS:12.1
2022–03–03 03:16:25

Gauze Out: 156

Gauze In Play:
7

History: +10 2022–03–03 03:16:00
+10 2022–03–03 03:15:59

History: +10 2022–03–03 03:16:09
+10 2022–03–03 03:16:08

Figure 35: Latest Product with Optimized UI

Chapter 6
Results and Discussion

6.1 Evaluation of AI Model

AI models are trained using the train dataset, with the validation dataset used to give an unbiased evaluation of the model fit on the overall dataset. With each epoch number which records the number of complete passes through the training dataset, the average loss for that iteration is recorded during the model training process. The lower the average loss value, the better the model performance.

Figure 36 below shows the average loss against the epoch number during training.

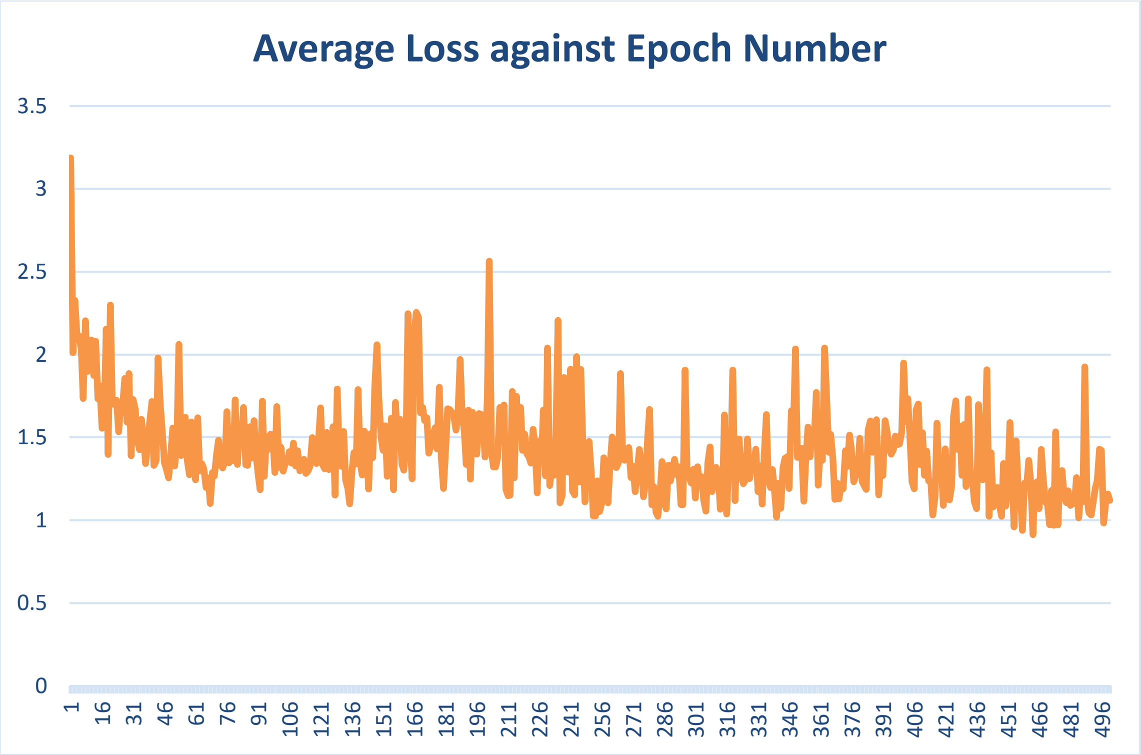


FIGURE 36: AVERAGE LOSS AGAINST EPOCH NUMBER

From the recorded values, it was observed that epoch 464 had the lowest average loss value of 0.9128984872138861. To prevent overfitting the model, from the training records of this specific epoch number, the parameters of the model were exported and utilised for real-time gauze object detection.

Following, to quantify AI development progress since the work done by the previous FYP student, the same evaluation metric was used. Mean average precision (mAP) at the Intersection over Union (IoU) of 0.5 was selected.

Looking into the terms of IoU and mAP. IoU is a value between 0 and 1 that defines the degree of minimal overlap between the expected and ground truth bounding boxes.

$$IoU = \frac{Area\ of\ Intersection\ between\ Expected\ and\ Ground\ Truth\ Boxes}{Area\ of\ Union\ of\ Expected\ and\ Ground\ Truth\ Boxes}$$

Figure 37 below illustrates this formula [45].

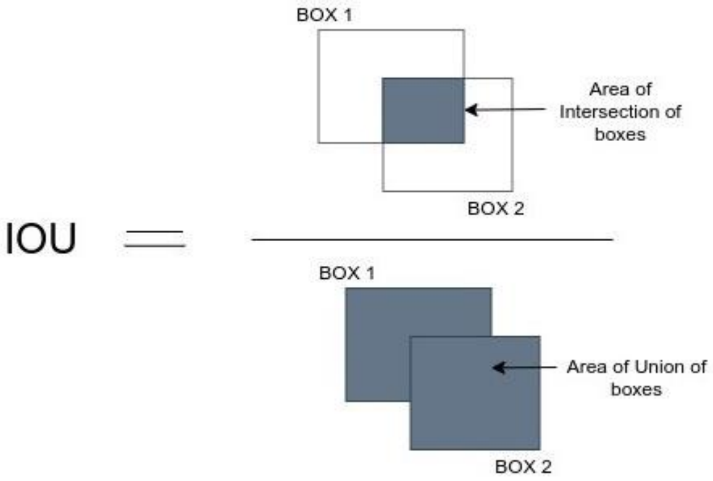


FIGURE 37: IOU FORMULA

At the value of IoU of 0.5, the minimal overlap between the expected and ground truth bounding boxes must be at least 50% for the detected object to be counted as a True Positive object. Conversely, with less than 50% of overlap, a False Positive object will be recorded. Additionally, if the algorithm fails to detect a ground truth object, it will be recorded as False Negative. With these values recorded, then calculation of mAP is made possible.

To calculate mAP, the following Precision and Recall values are first calculated.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Then, the mean area under the Precision-Recall curve is calculated to give mAP rating.

In terms of interpreting the mAP value, the higher the score, the more accurate the model's object detections are.

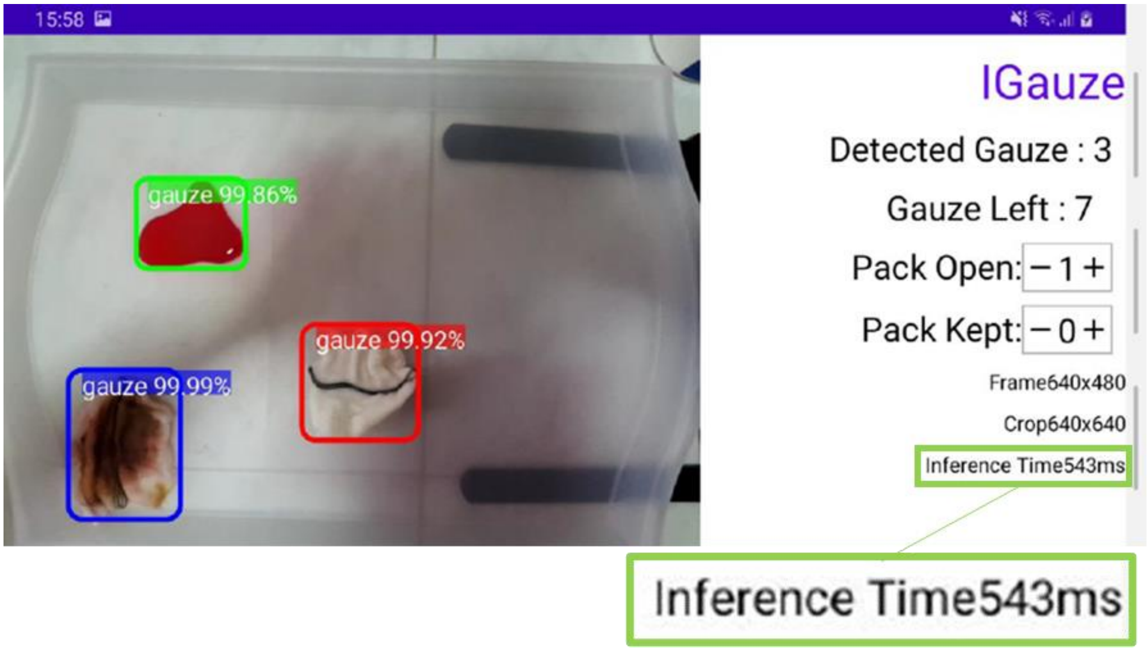
In comparison with the previous Android prototype’s mAP rating of 99.6%, the current product is superior by 0.1%, at 99.7%.

6.2 Evaluation of System Performance

To quantify the system’s performance, the metric Frames Per Seconds (FPS) was selected.

From Figure 38, the previous prototype processed each frame of its input video at the inference time of 543 milli-seconds. This roughly translates into 2 FPS. The current product functions at about 12 FPS, with Figure 39 supporting such performance.

With the upgrade in hardware and further optimizations performed on the software, a 6 times improvement in system performance based on video processing speed was achieved.



IGauze

Detected Gauze : 3

Gauze Left : 7

Pack Open: − 1 +

Pack Kept: − 0 +

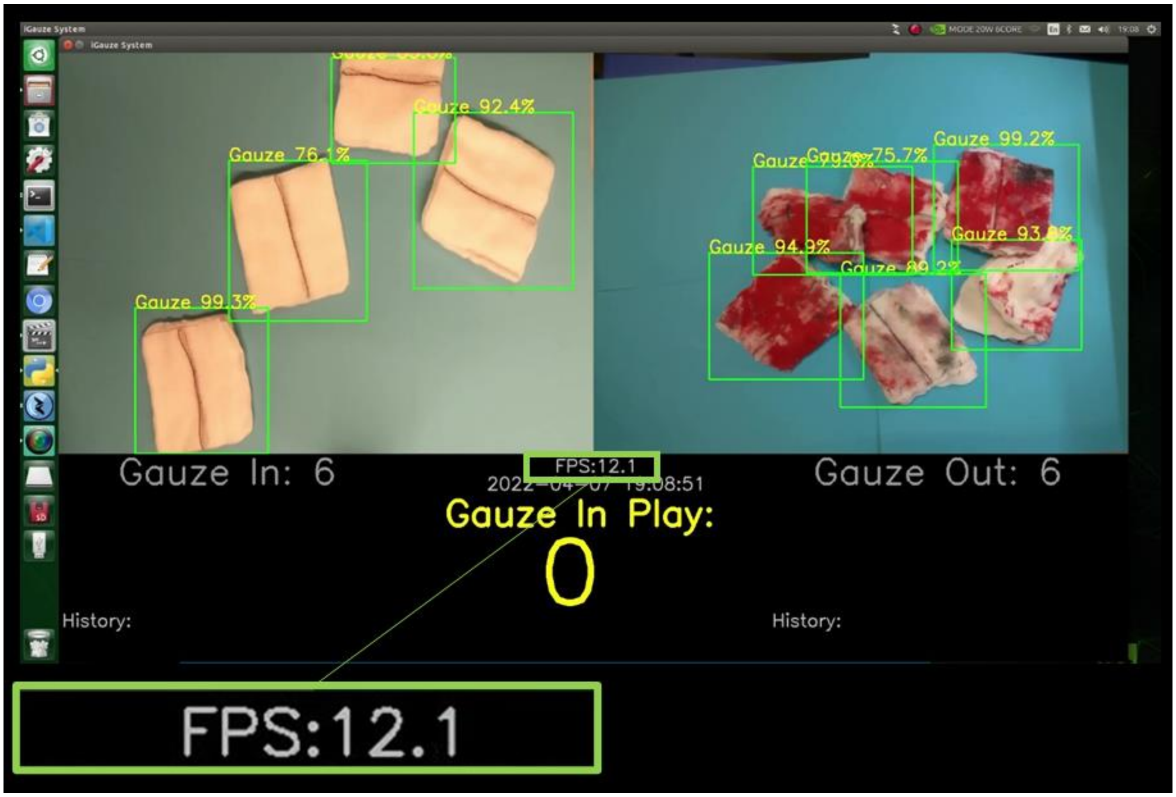
Frame640x480

Crop640x640

Inference Time543ms

Inference Time543ms

FIGURE 38: ANDROID PROTOTYPE'S INFERENCE TIME PER FRAME



Gauze In: 6

FPS:12.1

2022-04-07 19:08:51

Gauze Out: 6

Gauze In Play:

0

History:

History:

FPS:12.1

FIGURE 39: CURRENT PRODUCT'S FPS

The previous prototype also suffered from deteriorating performance as battery levels within the android phone drop.

The Android prototype relied on a battery power source and found itself to be draining at an unsustainable rate during operation, with its detection efficiency dropping proportionally to its battery life. A phone is not designed for heavy usage as an AI system. It is expected that a phone's performance will deteriorate quickly if used excessively in that manner. After consultation with the SingHealth Group Scrub Nurses, it was understood that there are no shortage of wall plugs in the Operation Theatres, with multiple wall plug towers within them. Figure 40 below shows one such wall plug tower within the Operation Theatres in SingHealth Group.

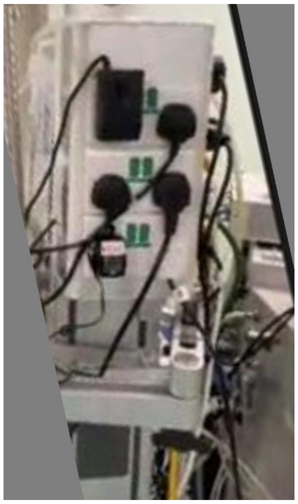


FIGURE 40: WALL PLUG TOWER IN OPERATION ROOM

Hence, the problem was solved by switching to the current product that is powered directly from the wall sockets. The long-term performance of the new product is also not compromised as it is using the Jetson Xavier NX which is designed to be able to act as a computational system of a medical product.

6.3 Current Limitations

Although major improvements were made, after testing, the current product was found to still have some limitations.

1. Good object detection performance is guaranteed only in good lighting conditions (Figure 41).
2. When two gauzes overlap excessively, the object detection algorithm will count these two as a single object, failing the product's main counting task (Figure 42).

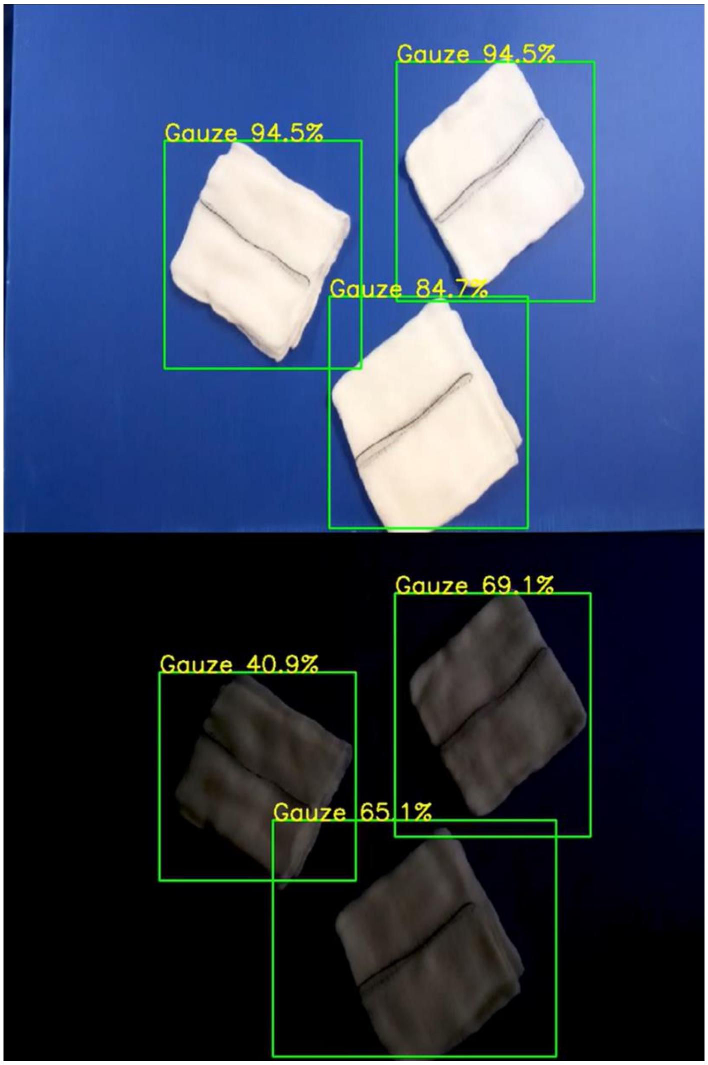


FIGURE 41: OBJECT DETECTION PERFORMANCE IN GOOD VERSUS BAD LIGHTING CONDITIONS

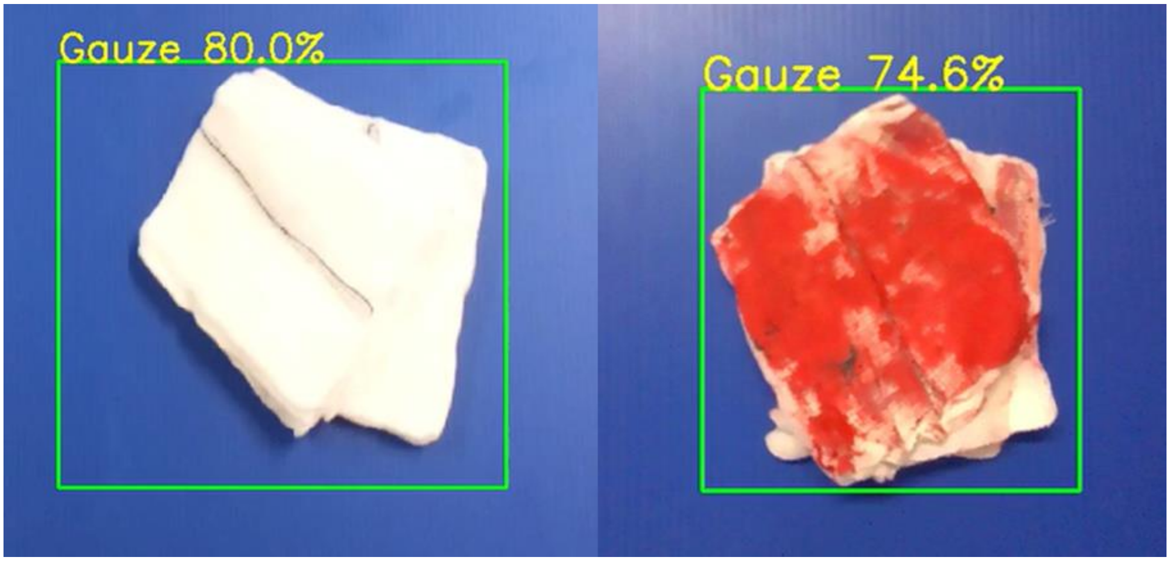


FIGURE 42: EXCESSIVE OVERLAP PROBLEM

To solve these limitations, the following actions can be taken:

1. LED lights are integrated with the product tray to ensure good lighting conditions are always present during counting task.
2. More image data is collected, and further retraining of the AI algorithm is done. With sufficient data, high accuracy can be achieved even when the gauzes overlap or are subjected to unusual light conditions.

Chapter 7
Conclusion

The main objective of this project is to develop a functional and practical smart system solution to support medical staff in the Operation Theatre in the prevention of Gossypiboma. This has been achieved with a solution built around the Jetson Xavier NX.

At every major checkpoint in the smart system's development, surgical staff of SingHealth Group were consulted to ensure that the changes made adhered to their medical regulations and guidelines.

Complementary functionalities such as dual camera capabilities and wireless connectivity were integrated and supports the product's use in the Operation Theatres. Designs with respect to the User Interface and user required interactions with the system were also made to be highly intuitive for quick and easy adoption by the medical staff.

Overall, this project was a success.

7.1 Contributions

In pursuit of the main objective while ensuring the smart system solution was well-suited for future mass-production, the following contributions were made:

- Main development work was done on the Jetson Xavier NX Developer Kit that can be remotely accessed via the Internet. Control over thousands of units from a single computer can be achieved if necessary.
- Majority of the annotation of gauze objects was done on CVAT, an open-source online platform, capable of exporting in most of the popular Computer Vision annotation formats. Hence, eliminating the future need for further re-annotation of images in this project.

- Re-training of the AI algorithm and software development work can be done on only one unit without sacrificing the productivity of other units.
- The Jetson Xavier NX Developer Kit's hotspot enable transmission from a maximum of 10 other devices within 28-meter radius at 54 megabits per second using the Wi-Fi protocol. This opens a lot of possibilities for the expansion of functionalities in the Operation Theatre.
- The User Interface and user necessary interactions with the system were heavily scrutinized to ensure maximal intuitiveness. It was vital for the product to be able to achieve quick adoption by medical professionals.

7.2 Recommendations

While the main objective of this project was achieved, the following recommendations can be executed to reach an even higher quality product:

- Real-world testing in Operation Theatres must be done to collect more samples to further improve the accuracy of the AI model and to confirm the system's reliability during actual use.
- Integrate a new function into the system to assist the recounting procedure to make the process more efficient and faster.

7.3 Future Work

In view of the ultimate goal of creating a successful product that can be used by hospitals throughout SingHealth Group and overseas, future work include:

- Ensure the system is ISO 13485 certified. The ISO 13485 specifies requirements for a quality management system where an organization needs to demonstrate its ability to provide medical devices and related services that consistently meet customer and applicable regulatory requirements. This achievement will give new users confidence in the system capabilities despite lack of experience in interacting with it.

- Expand the smart system’s detectable objects from only gauzes to include others such as needles, which also have the possibilities of causing Gossypiboma.
- Build upon the current system to achieve a Smart Operation Theatre through the development of new capabilities such as quick access to medical test results and the auto-generation of post-surgery documents.

References

[1] A. Mejri, K. Arfaoui, B. Aloui and J. Yaakoubi, “Gossypiboma: the failure of a successful intervention,” PAMJ, Jendouba, Tunisia, 2020.

[2] T. Ram, D. Dahiya and A. Naik, “Gossypiboma: case report and review of literature,” International Surgery Journal, New Delhi, India, 2019.

[3] P.-K. Shih, “Gossypiboma (retained surgical sponge) induces septic shock after previous breast surgery: a case report,” SAGE Open Medical Case reports, Taichung, China, 2019.

[4] H. E. Zemity, N. Hakami, M. A. A. Alfaki, M. Khurizi and A. Al-Zahrani Sr., “Intra-abdominal gossypiboma: a rare cause of palpable abdominal mass with a review of literature,” Cureus, Sudan, 2020.

[5] L. Bernstein, “When your surgeon accidentally leaves something inside you,” The Washington Post, 4 September 2014. [Online]. Available: https://www.washingtonpost.com/news/to-your-health/wp/2014/09/04/when-your-surgeon-accidentally-leaves-something-inside-you/. [Accessed 1 January 2022].

[6] World Health Organization, Implementation manual WHO surgical safety checklist 2009, International: World Health Organization, 2009.

[7] A. A. Gawande, “A randomised clinical trial of a bar-coded safety sponge system,” Brigham and Women's Hospital, Massachusetts, United States, 2006.

[8] S. Weprin, F. Crocerossa, D. Meyer, K. Maddra, D. Valancy, R. Osardu, H. S. Kang, R. H. Moore, U. Carbonara, F. J. Kim and R. Autorino, “Risk factors and preventive strategies for unintentionally retained surgical sharps: a systematic review,” *Patient Safety in Surgery,* no. PMID: 34253246, 2021.

[9] Japan Council for Quality Health Care, “Gauze remaining after surgery (1),” July 2019. [Online]. Available: https://www.med-safe.jp/pdf/No.152_MedicalSafetyInformation.pdf. [Accessed 20 January 2022].

[10] N. N. Egorova, A. Moskowitz, A. Gelijns, A. Weinberg, J. Curty, B. Rabin-Fastman, H. Kaplan, M. Cooper, D. Fowler, J. C. Emond and G. Greco, “Managing the prevention of retained surgical instruments: what is the value of counting?,” *Ann Surg,* no. PMID: 18156916, pp. 247(1): 13-8, 2008.

[11] National RFID Centre, “National RFID Centre creates diverse RFID eco-system in Singapore,” National Archives of Singapore, Singapore, 2009.

[12] A. Roger, E. Jones and D. Oleynikov, “Radio frequency identification (RFID) applied to surgical sponges,” *Surgical Endoscopy,* vol. 21, no. 7, pp. 1235-7, 2007.

[13] The Ministry Of Health, Brunei Darussalam, “Retained fragment of surgical gauze in hip arthroplasty,” *Brunei International Medical Journal,* vol. 13, no. 6, pp. 207-210, 2017.

[14] Alliance of Advanced Biomedical Engineering (AABME), “Internet of medical things revolutionizing healthcare,” ASME, 2017. [Online]. Available: https://aabme.asme.org/posts/internet-of-medical-things-revolutionizing-healthcare#:~:text=IoMT%20examples%20include%20sensors%20in,offer%20faster%20last%2Dmile%20delivery.. [Accessed 27 Jan 2022].

[15] S. Sharma, “How AI and IoT play an important role in healthcare today,” Ai4 - Artificial Intelligence Conferences, New York, 2021.

[16] H. L. Goh, Interviewee, *AI futures: how AI is augmenting Singapore’s healthcare.* [Interview]. 29 July 2021.

[17] ReportLinker, “The artificial intelligence in healthcare market is projected to grow from USD 6.9 billion in 2021 to USD 67.4 billion by 2027; it is expected to grow at a CAGR of 46.2% from 2021 to 2027,” Yahoo!, New York, 2021.

[18] E. Kunz, “Internet of things (IOT) in healthcare market to reach over $190 Billion by 2028 – powered by increasing implementation of cloud computing,” Vantage Market Research, Washington DC, 2022.

[19] Nvidia, “Jetson Xavier NX series,” Nvidia, 14 May 2020. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/. [Accessed 28 Jan 2022].

[20] D. Franklin, S. H. Sheshadri and S. Todd, “Bringing Cloud-Native Agility to Edge AI Devices with the NVIDIA Jetson Xavier NX Developer Kit,” Nvidia, Pittsuburgh, 2020.

[21] Raspberry Pi, “Raspberry Pi Zero W,” Raspberry Pi, 28 Feb 2017. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-zero-w/. [Accessed 17 Mar 2022].

[22] Raspberry Pi, “Raspberry Pi Camera Module 2,” Raspberry Pi, April 2016. [Online]. Available: https://www.raspberrypi.com/products/camera-module-v2/. [Accessed 28 Jan 2022].

[23] Nvidia, “JetPack documentation,” 2021. [Online]. Available: https://docs.nvidia.com/jetson/jetpack/introduction/index.html. [Accessed 28 Jan 2022].

[24] M. Elanotta, “Linux vs Windows: which is better for programming,” net2, 17 Feb 2021. [Online]. Available: https://net2.com/linux-vs-windows-which-suits-best-programmers/. [Accessed 28 Jan 2022].

[25] IBM, “What is computer vision?,” IBM, 2021. [Online]. Available: https://www.ibm.com/sg-en/topics/computer-vision. [Accessed 4 Feb 2022].

[26] IBM Cloud Education, “Deep Learning,” IBM, 1 May 2020. [Online]. Available: https://www.ibm.com/cloud/learn/deep-learning. [Accessed 4 Feb 2022].

[27] TechTarget Contributor, “Definition convolutional neural network,” TechTarget | SearchEneterpriseAI, April 2018. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network. [Accessed 4 April 2022].

[28] P. Sharma, “Understanding transfer learning for deep learning,” Data Science Blogathon, New Delhi, 2021.

[29] J. Lemley, S. Bazrafkan and P. Corcoran, “Transfer learning of temporal information for driver action classification,” in *28th Modern Artificial Intelligence and Cognitive Science Conference*, Fort Wayne Indiana, 2017.

[30] Datagen, “New study finds over 96% of computer vision (cv) teams already using synthetic data for training and testing of visual machine learning models,” GlobeNewswire by notified, Israel, 2021.

[31] V. Chawla, “How do data scientists create high-quality training datasets for computer vision,” *Analytics India Magazine,* 5 Aug 2020.

[32] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Jounal of Big Data,* p. 60, 2019.

[33] T. Yeung, “What is edge computing?,” Nvidia, 22 Oct 2019. [Online]. Available: https://blogs.nvidia.com/blog/2019/10/22/what-is-edge-computing/. [Accessed 4 Feb 2022].

[34] D. Poddar, S. Nagori, M. Mathew, D. Maji and H. Garud, “Deep learning based parking spot detection and classification in fish-eye images,” in *2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, 2019.

[35] P. McWhorter, “Jetson Xavier NX lesson 1: introduction and first boot for beginners,” YouTube, 28 May 2020. [Online]. Available: https://www.youtube.com/watch?v=LuWYXKrQcJ8&list=PLGs0VKk2DiYw_Xx7c3M_9HMEcn9hPhFJV&index=3. [Accessed 7 Feb 2022].

[36] JetsonHacks, “Jetson Xavier NX - run from ssd,” YouTube, 25 May 2020. [Online]. Available: https://www.youtube.com/watch?v=ZK5FYhoJqIg. [Accessed 7 Feb 2022].

[37] D. Franklin, “Hello AI World,” NVIDIA Jetson, 19 Jun 2019. [Online]. Available: https://github.com/dusty-nv/jetson-inference. [Accessed 7 Feb 2022].

[38] D. Franklin, “Wi-Fi Hotspot Setup - NVIDIA Jetson Developer Kits,” NVIDIA Jetson, 5 Oct 2021. [Online]. Available: https://www.youtube.com/watch?v=GAOvGAdwiHk. [Accessed 18 Mar 2022].

[39] P. McWhorter, “AI on the Jetson Nano LESSON 62: Make a Streaming IP Camera from a Raspberry Pi Zero W,” YouTube, 3 Oct 2020. [Online]. Available: https://www.youtube.com/watch?v=7Bz0QzlK6ps. [Accessed 18 Mar 2022].

[40] K. Wiggers, “Intel open-sources CVAT, a toolkit for data labelling,” *The Machine, Making sense of AI,* p. 1, 5 March 2019.

[41] I. Jose, “Resize-pascal-voc,” Open-source, 11 Jan 2021. [Online]. Available: https://github.com/italojs/resize_dataset_pascalvoc. [Accessed 2 March 2022].

[42] D. Franklin, “Collecting your own Detection Datasets,” NVIDIA Jetson, 28 Oct 2020. [Online]. Available: https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-collect-detection.md. [Accessed 2 Mar 2022].

[43] D. Franklin, “Jetson AI Fundamentals - S3E5 - Training Object Detection Models,” YouTube, 3 Nov 2020. [Online]. Available: https://www.youtube.com/watch?v=2XMkPW_sIGg&t=978s. [Accessed 2 Mar 2022].

[44] C. Kreitzberg, “The intuitive interface,” Princeton University User Experience Office, 3 Feb 2017. [Online]. Available: https://ux.princeton.edu/learn-ux/blog/intuitive-interface. [Accessed 3 Mar 2022].

[45] V. S. Subramanyam, “IOU (Intersection over Union),” 17 Jan 2021. [Online]. Available: https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef. [Accessed 27 Apr 2022].