DENTAL AUGMENTED REALITY – AUGMENTED REALITY (AR) BASED ON DEEP LEARNING FOR DENTAL EDUCATION

NUR DINIE BINTI MOHD RAZALI

BACHELOR OF INFORMATION

TECHNOLOGY (HONS)

(INFORMATION TECHNOLOGY)

MAY 2021

CERTIFICATION OF APPROVAL

Dental Augmented Reality – Augmented Reality (AR) Based on Deep Learning for Dental Education

by

Nur Dinie Binti Mohd Razali

17004915

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS in partial
fulfilment of the requirements for the
Bachelor of Information Technology (Hons)
(Information Technology)
MAY 2021

Approved by,

____

(Dr. Mohd Nordin B Zakaria)

Universiti Teknologi PETRONAS
32610 Bandar Seri Iskandar
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

NUR DINIE BINTI MOHD RAZALI

ABSTRACT

A project has been carried out involving dental students, deep learning, augmented reality (AR), and markerless AR to identify the challenges that dental students face and provide a new type of learning based on the observation. COVID-19 has cause major disruptions on the educational institutions in forcing the student to adapt to online learning and cancellation of overall physical assessment. This project suggests a way for dental students to proceed with their education virtually or via online. This project focus on AR-based deep learning for dental education. The students could practice their knowledge by using the Dental AR prototype that trains them to identify several types of teeth and test their capabilities through the learn mode implemented inside the prototype. Hence, this project might be a great option for them to nurture their skills and strengthen their experiences without having face- to-face practical anywhere and anytime.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the Universiti Teknologi PETRONAS for letting me fulfill my role of being a student here. I would also like to thank the Information Technology Department for allowing me to write an honors thesis on my Final Year Project. Most importantly, I owe a great debt of gratitude to my supervisor, Dr. Mohd Nordin B Zakaria. I am appreciative for his advice and recommendations throughout my project, as well as for constantly listening and encouraging me as I built my project and formatted my paper. This project would not be possible without his help and guides. To all my friends, Yee Ben Kuek, Izzat Bin Mohamad Jehan Zaib, Muhammad Fakhrul Ariffin Bin Nazri, and Iylia Nur Ezaty Noor Azahar, and the others, who have always been kind and encouraging me throughout the year and for their greatest moral support in my life. Not to forget, my utmost source of strength and motivation, which is both of my parents, Salwana Bt. Aminuddin and Mohd Razali B. Abdul Hamid for supporting me through this year's turmoil and refusing to let me give up. I do not have any meaningful words to express my gratitude, but my heart is still overflowing with gratitude for the kindness shown to me by everyone.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Background

i. Dental Students

COVID was declared a pandemic by the World Health Organization (WHO) early last year, and it has since become a major public health concern globally. As a result, infection control and physical separation measures have been implemented to confront the virus and gain control of the pandemic scenario. Not to mention, the government has imposed a new compulsory policy of social separation, which has been applied in many countries, including Malaysia, resulting in the nationwide closure of institutional buildings such as schools and universities. According to the policy, even dentistry academic institutions are under pressure to comply with the required procedures and revisions to continue providing education and ensuring student academic advancement. As a result, all types of teaching and learning were quickly switched to a full E-learning environment. (Amir et al., 2020)

E-learning, often known as online learning, is a type of learning that takes advantage of technological aids such as information and communication technologies (ICTs). With technological resources and novel tactics, the teaching and learning process has been revolutionized. Previous research has found that various E-learning and its technologies have a positive impact on teaching and learning, particularly in the domains of medicine and dentistry. Furthermore, studies have demonstrated that the performance and knowledge gained by E-learning are comparable to face-to-face approaches. There are also new phrases such as mix or blend learning, which facilitates both classroom and online learning to help students reach their goals of becoming more involved, independent, and collaborative among themselves. In general, the mix learning strategy expands on traditional learning by combining physical and virtual learning experiences.

According to a prior survey, blended learning resulted in higher levels of satisfaction, performance, and student involvement. As a result, it was claimed that this technique promotes self-directed and active learning, which has gained popularity in dental education as the most highly advantageous strategy when compared to traditional learning. (Amir et al., 2020)

ii. Deep Learning

One of the subsets of machine learning called deep learning acts as an artificial intelligence (AI) function capable of imitating the human brain, particularly in data processing and pattern creation for use in decision making such as detecting objects, translating languages, and identifying speech, among other things. (How Deep Learning Can Help in the Prevent of Financial Fraud, n.d.) The impact of deep learning is that it can learn even without human intervention from unstructured and unlabeled data. We could make uses of Deep learning to distinguish any fraud or even money laundering.

iii. Object Detection with Deep Learning

The subgroup of computer vision techniques like object detection focuses on finding and classifying a variable number of items in an image or video. In comparison to classification issues, the results of object detection are variable in length forms because the number of items can be swapped from one image to another. (Tryolabs, 2017) Moreover, using this identification and localization, items in a scene can be counted and tracked to their specific locations, as well as precisely labelled. (Object Detection Guide | Fritz AI, n.d.)

According to Blog (2018), object detection not only recognizes and locates each object in an image but also solves the object detection problem. By putting a bounding box around the things that we want to differentiate from other items, we can attribute the relevant object category with each bounding box.

iv. Augmented Reality (AR) In Dental Education

Augmented reality (AR) technology extends virtual knowledge by incorporating real-world experiences, opening new possibilities in a variety of disciplines. Furthermore, unlike VR, AR does not require any large pieces of equipment, allowing consumers to feel comfortable with it in a sense of freedom. (Practice, 2018)

In medical education and training, AR technologies widely used as most of the medical applications of augmentation focused on surgery types, including neurosurgery, laparoscopic surgery, and plastic surgery. With the help of AR technology, the patient also can be exposed to how the process has been done in a virtual environment and able to enhance their experience within it. The key area of use in dentistry is oral and maxillofacialsurgery. Not to mention, the most frequent applications that were used is dental implant placement and orthognathic surgery. The dentist can even access themselves in the implementation of the latest applications of restorative dentistry, orthodontics, endodontics, etc. (Kwon et al., 2018)

v. Markerless Augmented Reality (AR)

As shown by estimations, the augmented reality business would be worth more than 198 billion dollars by 2025. (From 3.5 in 2017). Despite its immense potential, the sector is currently undergoing major technological disruption, both in hardware and software. Now, mobile devices may access augmented reality in three categories: marker-based, markerless, and location-based.

According to Schechter's (2020) research, a software application called markerless augmented reality (AR) does not require prior knowledge of a user's environment and it is also capable in overlay virtual 3D material into a scene and holding it to a fixed point in space. For example, to experience more on the marker-based augmented reality, a static image, sometimes known as a trigger photo, can be seen when a user browses an augmented reality app with their mobile device. Then, the smartphone scan will result on top of the previously created marking, the appearance of new content such as 3D animation, video, etc. (Zvejnieks, 2019)

1.2 Problem Statement

According to the latest research paper from Abdul Fatah et al., (2021), Coronavirus Disease 2019 (COVID-19) has caused global outbreaks as it is known for its infectious diseases. This outbreak has imposed numerous sectors to change and adapt to their new alternatives way in their daily practice, hence, dental education is no exception. This effect from this pandemic could be seen from the severe disrupt teaching, learning, and even clinical activities that more to the physical assessment.

Therefore, this project focused on exposing the new learning, distance learning methods in terms of teaching and learning activities. Dental students still could proceed with their education via the technologies nowadays from online learning such as videoconferencing and even applications that available for dental students.

1.3 Objectives And Scopes of Study

This project aims to create an application for dental students to strengthen their knowledge and practices their skills in real-time. This project will be able to detect diverse types of teeth through the function within the apps.

Therefore, the objectives applied for the project are:

1. To investigate and build the datasets for the training and testing phase of the object detection algorithms.
2. To design and develop a deep learning model for object detection of recognizing the several types of teeth.
3. To develop a Dental AR application prototype for dental student.

In terms of implementation, the development environment, tools, development platform, and levels of system users are also discussed.

i. Focus

The focus of this project is to study the impact of the teeth detection application on dental students so that they able to distinguish the distinct types of teeth. The boundaries and limitations of the project are to create prototype AR application that able to recognize the several types of teeth.

ii. Target user

The target users of my project are dental student who has been struggling to practices their skills and knowledge.

iii. Time limitation

The time limitation giving to complete this project is 8 months, which is 4 months for FYP1 (12 weeks) and 4 months for FYP2 (12 weeks) in total 24 weeks.

iv. The software and hardware used in the project are below:

- Python Programming Language
- SuperAnnonate
- labelImg
- Google Colab
- YOLO
- OpenCV
- Jupyter Notebook

v. Significance of study

The impact or importance of the project is to study the struggle of dental students that were forced to adapt to distance learning and online learning. As a result, providing a new learning way to continue their education without having a face- to-face physical assessment.

CHAPTER 2

LITERATURE REVIEW AND/OR THEORY

2. LITERATURE REVIEW AND/OR THEORY

2.1 Introduction

As the pandemic, COVID-19 be regarded as the worst public health crisis of the 21st century, daily life has been affected regardless of the significant efforts to alleviate the disease spread. In addition to that, students are no exception to this unsettling matter as the institutional has demanded all the campuses to closed and even reducing in-person to online learning. Not to mention, any physical meeting, clubs, and graduations were postponed, moved to online venues, or even cancelled. (Hung et al., 2020)

2.2 Dental Student

According to new research articles of Hung et al., (2020), an anonymous online survey comprises demographics of the students, the protocols of school reopening, student concern and insight of the institutional responses, and their psychological impacts has been distributed among dental students to get their opinion and experiences related to COVID-19.

The results from the survey show that most students believed social distancing in school could lessen any close contact that may relate to the development of COVID-19. Despite, students felt that clinical education has suffered even after the adaptation to online learning, they responded positively to the adjustments to other online curricular components. (Hung et al., 2020)

2.3 Deep Learning

Deep neural network or neural learning is the other term for deep learning which is capable to imitate the brain's functions like Artificial Intelligence (AI). It Is a type of machine learning that uses functions to make decisions in a nonlinear manner. Even without the supervision of unstructured data, neural learning can occur when decisions are made. Deep learning is used to execute tasks such as beholding, speech recognition, and language translation, to name a few.

How does deep learning work?

Deep learning can investigate data using hierarchical neural networks like the human brain where the neuron codes are linked together. Besides, to require a nonlinear method, the data structure of deep learning was used to refine input through a few layers, each of which will incorporate the following tiers of other information, unlike other standard linear programs in machines.

What is an example of deep learning?

Deep learning will use a variety of signals to detect fraud, including Internet Protocol (IP) address, sender, receiver, credit score, etc. It will assess the number sent in the first layer of its artificial neural network. This information will be used in a second layer, which will include the IP address, for example. in the third layer, the credit score is added to the current information, and so on until a judgment is reached.

Deep Learning vs. Machine Learning

One of the most widely used AI techniques for processing massive data is called machine learning, where are knowns as a self-adaptive algorithm that improves its analysis and patterns with experience or fresh data. For example, a digital payments corporation can detect any potential for fraud in its system via machine learning tools. All transactions will be measured by a computer model's computational algorithm that occurs on the digital platform. It also can locate patterns within the data set and include any anomalies discovered by it.

A hierarchical level of artificial neural networks within the deep neural network was utilized to support the machine learning development. Synthetic neural networks are constructed in the same way as the human brain, with neuron nodes connected in a web-like pattern. While traditional programs construct analyses using data in a relatively linear manner, deep learning systems' hierarchical function allows machines to analyze data in a nonlinear manner.

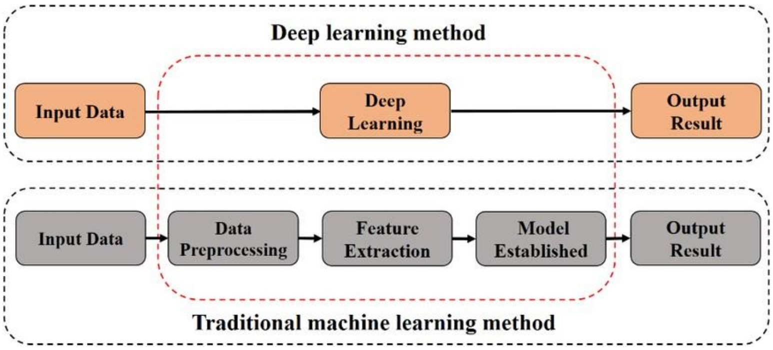


Figure 1: Deep Learning method and Traditional machine learning method.

2.4 Object Detection with Deep Learning

Object detection is a computer vision approach that recognizes and identifies any required objects in an image or video. It may be used to count the items in an image or video and establish their precise location, also known as ground truth while labeling them appropriately. (Fritz AI, Object Detection Guide, n.d.)
The image below, for example, displays one person and two pets. Using object detection, the objects in the image can be classified into different classes while also detecting instances of them inside the image.

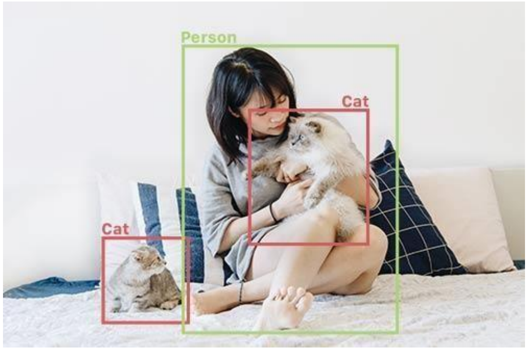


Figure 2. Object Detection Example

Object detection is more on labeling the objects using bounding boxes around the objects that have been distinguished which allows us to locate where the objects are in each scenario as above. (*Object Detection Guide | Fritz AI*, n.d.)

There are several computer vision techniques which is:

Table 1. Object Detection Example

Image Classification	Object Detection	Object Localization	Semantic Segmentation	Instance Segmentation
Determine the kind or class of an object in a picture.	Locate the existence of things in a picture using a bounding box and the types or classes of the objects found.	Detect the existence of items in an image and mark their location with a bounding box.	This technique generates a pixel-level understandin g of the elements of a scene.	Instead of a coarse bounding box, recognized objects are indicated by highlighting individual pixels within the item.

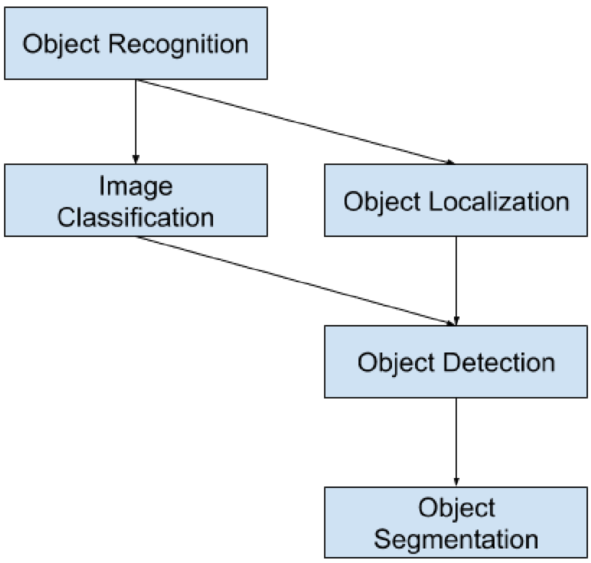


Figure 3. Overview on Object Recognition Tasks

Many people are confused about the difference between object detection and image recognition. Object detection is primarily about drawing a bounding box around the detected items, whereas image recognition is about providing a text label to image objects. For example, as illustrated in FIGURE 4,

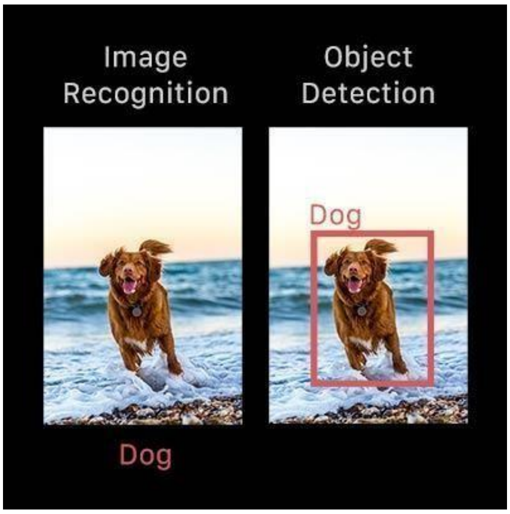


Figure 4. Comparison between Image Recognition and ObjectDetection

As we progressed, we discovered that object detection may be divided into two categories: machine learning-based approaches and deep learning-based approaches. In terms of our project, we concentrated on the second technique, which is based on deep learning. (Object Detection Guide | Fritz AI, n.d.)

2.5 Practical uses of object detection

Even though object detection is still of a new tool in the industry, there are already many useful and exciting applications using it.

Table 2. Face Detection, Counting, and Visual Search Engine

Face detection	**Counting**	**Visual Search Engine**
Certain point-and-shoot cameras have had the capacity to detect faces for speedier auto-focus since the mid-2000s. Even though it is a more specific type of object detection, the methodologies used are applicable to a broader range of items.	Counting is a simple but often disregarded use of object detection. Counting people, cars, flowers, and even microorganisms is a real-world requirement for a variety of image-based systems. With the recent advancement of video surveillance technology, there is now a greater than ever opportunity to use computer vision to convert raw data into structured data.	Finally, one of our favorite use cases is Pinterest's visual search engine. They index various regions of the image using object detection as part of the process. This way, if you're looking for a specific purse, you can locate examples of purses like the one you're looking for in a different context. This is far more powerful than using Google Image's reverse search to find similar images.

Table 3. Machine Learning-based approaches and Deep Leaning-basedapproaches

Machine learning-based approaches	Deep learning-based approaches
- Histogram of colors or edges - Recognize clusters of pixels that may be associated with an object	- Use convolutional neural networks (CNNs) to do end-to-end object detection - Unsupervised object detection - The state-of-the-art approaches to object detection

Now, a self-adaptive algorithm from machine learning is improving in terms of analysis and patterns with experience, even with newly added data. Within deep learning-based object detection models, there are two parts: encoder and decoder. An encoder uses an image as input to locate and name objects using a succession of blocks and layers that then learn to extract statical features. The encoder output is then sent to a decoder, which predicts bounding boxes and draws labels for each detected object. (Object Detection Guide| Fritz AI, n.d.)

Here are a few deep learning-based approaches to object detection:

i. **RCNN Model Family**

- **R-CNN**
- **Fast R-CNN**
- **Faster R-CNN**

ii. **YOLO Model Family**

Another prominent family of object recognition models is YOLO, or "You Only Look Once," which was developed by Joseph Redmon et al. in a customized framework called Darknet. Darknet is well-known for its versatile low-level language research platform, which has also created a succession of the greatest real-time object detectors in computer vision: YOLO, YOLOv2, YOLOv3, and now YOLOv4. (Nelson, 2021)

By integrating drawings of bounding boxes and identifying the class of each detected object in a one-to-end differentiable network, the actual YOLO model was the first object detection network. (Nelson, 2021) YOLO is a single convolutional network and a single-stage network that handles both object identification and classification in a single network pass. (Nelson, 2021) It then forecasts the bounding boxes as well as the class probabilities for these boxes. Although R-CNN models are more accurate, the YOLO family of models is faster, achieving object recognition in real-time.

YOLO

The YOLO model was first detailed in 2015 research titled "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon et al. It should be noted that Ross Girshick, the creator of R-CNN, was also an author and collaborator to this work.

The method employs a single end-to-end trained neural network that accepts a photograph as input and predicts bounding boxes and class labels for each bounding box directly. Although it operates at 45 frames per second and up to 155 frames per second for a speed-optimized version of the model, the technique provides lower prediction accuracy (e.g., more localization errors).

The model works by first dividing the input image into a grid of cells, with each cell predicting a bounding box if the centre of a bounding box falls within the cell. Each grid cell predicts a bounding box based on the x, y coordinates, width, height, and confidence. Each cell is also used to forecast a class.

For example, an image may be divided into a 7×7 grid and each cell in the grid may predict 2 bounding boxes, resulting in 94 proposed bounding box predictions. The class probabilities map and the bounding boxes with confidences are then combined into a final set of bounding boxes and class labels. The image taken from the paper below summarizes the two outputs of the model.

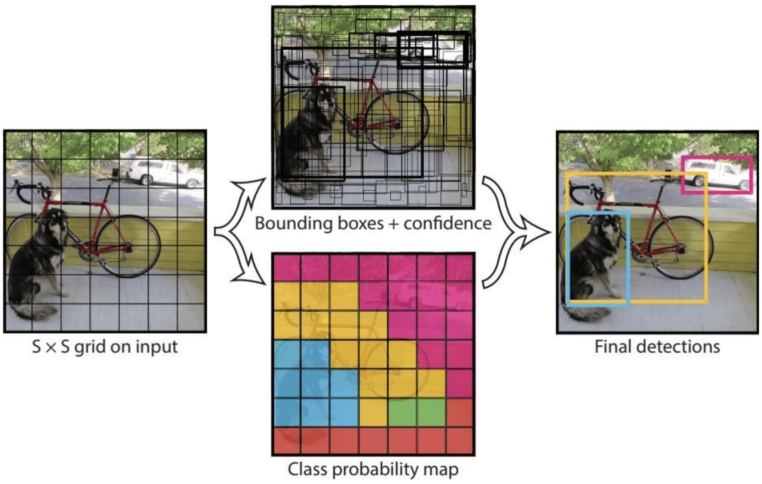


Figure 5. Summary of Predictions made by YOLO Model. Taken from: YouOnly Look Once: Unified, Real-Time Object Detection

YOLOv2 and YOLOv3

Both YOLOv2 and YOLOv3 were created by Redmon (n.d.). As for YOLO models released after YOLOv3 are stated to be the new authors – unlike strictly sequential releases to YOLOv3. (Nelson, 2021)

In 2017, YOLOv2 was released and earn an honorable paper mention at CVPR 2017 where the architecture made from several iterative improvements on top of YOLO including BatchNorm, anchor boxes, and higher resolution. Meanwhile, in 2018, YOLOv3 was released upon previous models by adding an objectness score to bounding box prediction, added connections to the backbone network layers, and improve the performance on smaller objects by making predictions at three separate levels of granularity. (Nelson, 2021)

YOLOv4, YOLOv5, and So on.

Joseph Redmon stepped aside from computer vision research after he released YOLOv3. Later new researchers like Alexey Bochkovskiy and innovators like Glenn Jocher adopted the YOLO models and began to open source their achievements in computer vision research. Not to mention,groups like Baidu also released their implementations of YOLO such as PP- YOLO through the demonstration of an improvement in mAP and decreased latency. (Nelson, 2021) In April 2020, YOLOv4 released by Alexey Bochkovskiy became the first paper in the "YOLO family" not authored by Joseph Redmon. Improved feature aggregation, mish activation, and many more were introduced through YOLOv4. (Nelson, 2021)

In June 2020, YOLOv5 was released by Glenn Jocher where it is the first model in the "YOLO family" that is not accompanied by any paper. YOLOv5 is like undergoing model repo development. It stated that at first Glenn maintained a version of YOLOv3 implemented in PyTorch, but as he continued in upgrading the improvement onto the architecture itself, then he decided to release a new repo branded as YOLOv5. (Nelson, 2021)

PP-YOLO, released in August 2020 by Baidu, surpasses YOLOv4's performance metrics on the COCO dataset. The "PP" stands for "PaddlePaddle," Baidu's neural network framework (akin to Google's TensorFlow). PP-YOLO notes improved performance through takingadvantage of a replaced improvements like a model backbone, DropBlock regularization, Matrix NMS, and more. (Nelson, 2021)

Scaled YOLOv4 came out in November 2020 by Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. The model takes advantage of Cross Stage Partial networks to scale up the size of the network whilemaintaining both the accuracy and speed of YOLOv4. Notably, Scaled YOLOv4 takes advantage of the training framework released in YOLOv5. (Nelson, 2021)

PP-YOLOv2, again authored by the Baidu team, was released in April2021. PP-YOLOv2 made minor tweaks to PP-YOLO to achieve improved performance, including adding the mish activation function and Path Aggregation Network. (Nelson, 2021)

2.6 Augmented Reality (AR) On Dental Education

The evolution of augmented reality (AR) may be seen in common gadgets like smartphones, tablets, and even smart eyewear. AR can be implemented with any device that has a camera and specific AR application software loaded in its environment. AR might use this to display any type of supplementary information that is partly real and partly digital on the display screen, such as texts, images in 2D or 3D, real-life films, and even animations.

The core idea of AR is the digital layer, in which the camera scans the object in the frame, the system recognizes it, and a new level of communication is begun, which must be external and seamlessly merge with reality. Unlike virtual reality (VR), augmented reality (AR) only incorporates some real-life characteristics into its surroundings rather than replicating the entire virtual universe. As a result of today's technological breakthroughs, more powerful computers with AR characteristics are now possible. New generations of mobile phones and devices appropriate for AR applications are being born in this era.

Augmented Reality (AR) in dental education can be different from virtual reality (VR) in sense of offering virtual information to the environment surrounding us. AR gives more freedom to the user where they do not require any physical equip with it. (Practice, 2018) For example, with VR, we can see how the dental assessment works side by side but with AR, we can watch the teeth on top of the table as shown in FIGURE 12. The comparison between VR and AR is that VR let us immerse ourselves in a scenario where dentist examines their patient teeth, while AR is that we can practice the procedures by ourselves everywhere with no pressure. (Practice, 2018)

Figure 6. Augmented Reality (AR) Apps

In dentistry education, augmented reality (AR) differs from virtual reality (VR) in that it provides virtual information to the environment around us. AR allows users more flexibility because they don't need any physical equipment to use it. (2018, Practice) For example, we can view how the dental examination works side by side in VR, but we can see the teeth on top of the table in AR. The difference between VR and AR is that VR allows us to immerse ourselves in a scenario where a dentist checks their patient's teeth, whereas AR allows us to practice operations without any pressure anywhere. (Practice, 2018)

The field of dentistry focused on the surgery realm. Even with a convenient pop-up screen at the push of a button, dentists could practice complex procedures or dissect their patient's vital signs. As it provides visual data in which extremely useful in terms of processing and retaining information efficiently, it acts as a beneficial tool for dentists especially with regards to education and further training. (Practice, 2018)

While dentistry is a highly concentrated technique that necessitates a wide range of exceptional abilities, the consequences of insufficient training and poor performance are extremely substantial. As a result, virtual medical simulation programs for dentistry have provided the best technique to build and develop a trainee's competence in a risk-free environment, particularly where we may test these concepts and treatments on real patients without any injury or pain. Furthermore, by providing a generally accepted level of dental treatment, an online program exposes a much larger pool of students to the possibilities of high-quality instruction. (Dental Training AR App by AugmentedReality.Health,n.d.)

AR's development may be seen with software programs that integrate digital content (visual, audio, and other forms) into the user's real-world surroundings. In terms of AR in medicine, a dentist may employ software programs to record a medical treatment, safely communicate clinical and other critical data, and conduct educational training. Take B Med, for example, as an example of an established software application system that caters to medical support.

Furthermore, as technology advanced, smart glasses were developed, and this concept was inspired by Frank L. Baum's book The Master Key, (L. Frank Baum’s “The Master Key” Imagines a Kind of Augmented Reality: History of Information, n.d.) in which the protagonist, a young boy, wears new advanced technology glasses that allow him to determine whether the person he sees is good, bad, stupid, or intelligent, among other things. Future smart glasses are wearable computers that can deliver audio and video streaming chats as well as acquire information instantaneously on the lenses of the glasses, all while staying completely hands-free.

Some AR gadgets incorporate a touchpad or remote control, while others rely solely on voice instructions. They have a wide variation and a functional design. Epson, Vuzix, and Realwear, among the most recent models on the market, appear to be basic attractive spectacles. However, they are equipped with augmented reality technology. Augmented reality is fast gaining popularity because it blends elements of the virtual world into the real world.

2.7 Markerless Augmented Reality (AR)

The augmented reality market is expected to grow to more than 198 billion dollars by 2025. Despite its great capabilities, whether in hardware or software, the sector is currently undergoing a significant technical upheaval. Augmented reality can be accessed via mobile devices and is classified into three types: marker-based, markerless, and location-based. All of them have limitations that engineers hope to solve in the coming years.

Figure 7. An example of Markerless Augmented Reality (AR)

Augmented Reality (AR) technology can provide additional digital information of a real-world scenario and has been utilized in a variety of applications including gaming, video browsing, virtual fitting rooms, and furniture arrangement simulation. Marker-based AR is a popular technique for creating AR applications since markers of specified patterns can be easily identified in the captured real scene. Camera posture estimation may be accomplished rapidly based on the geometry of these markers, allowing for the timely and appropriate insertion of vivid virtual content. (Chen et al., 2017)

However, the need for markers, which may limit the creation of AR applications/products, is an inherent disadvantage of marker-based AR. Markerless AR, on the other hand, estimates the camera posture based on visual or depth information from the collected scene and does not require a specific marker. (Chen et al., 2017) Unfortunately, the computational burden of markerless AR is substantially higher in most circumstances, restricting its use and development on mobile devices.

Marker recognition, on the other hand, can be local or cloud-based, which implies that marker databases can be maintained on the device and recognition can also take place on the device. The databases can also be stored in the cloud, and recognition occurs on a server; the phone just sends point clouds to the server. When utilizing device-based recognition, the content is immediately downloaded from the server; however, when using cloud recognition, the content is obtained from the server after a short delay. It normally takes a few seconds for the consumer to detect any augmented reality experience. (Zvejnieks, 2019)

When employing cloud recognition apps, the trigger image (marker) must be unique; at all costs, avoid using stock photos, as they may have already been used by other projects or apps. To avoid such issues, it is recommended to capture our one-of-a-kind material or create a custom-made design. Regardless of the software, the mobile device will use computer vision to bring this unique snapshot to life and display your augmented reality material. (Zvejnieks, 2019)

Marker vs. Markerless Augmented Reality

Marker-based and markerless AR are two of the most common ways to experience AR. Marker-based AR technologies were used in the early stages. Markerless AR is now the primary picture recognition method for AR applications. Marker-based AR apps use markers (target pictures) to indicate what is in each space. These markers define where the AR application inserts digital 3D content within the user's visual field or via a camera feed.

Table 4. Strength and Weakness of Marker vs. Markerless AR

Strength	Weakness
- If the marker picture is properly created, marker-based AR content gives high-quality experiences, and tracking is very stable; the AR content does not waver - Simple to use; no comprehensive instructions are required for first-time users	- When moving the mobile camera away from the marker, the AR experience disappears, and the trigger photo must be scanned again. It is feasible to employ extended tracking, but in most circumstances, it makes matters worse - Scanning will fail if markers reflect light in certain circumstances - To improve tracking stability, the marker should have strong borders/contrast between black and white colours. Recognition will be impossible due to the smooth colour shift

Rather than recognising markers, markerless AR inserts virtual 3D objects in the physical area based on the environment's true properties. Object tracking systems are no longer required because of this distinction. Because of developments in cameras, sensors, CPUs, and algorithms capable of correctly detecting and mapping the real-world, markerless AR experiences are now conceivable.

Location-Based Markerless Augmented Reality

Location-based AR integrates virtual 3D items into the user's physical environment. The technology connects an object to a point of interest by using the device's position and sensors. Our immediate surroundings come to life depending on where we gaze. Niantic's mobile gaming sensation Pokémon GO which received over 1 billion downloads globally, employs markerless, location-based

AR technology.

How Does Markerless Augmented Reality Work?

Markerless AR combines digital data with real-time, where the real-world inputs recorded to a physical area. The technology integrates software, audio, and video graphics with the cameras, gyroscope, accelerometer, haptic sensors, and location services of a smartphone or headset to record 3D visuals in the actual environment.

Markerless AR detects objects or distinguishing features in a scene without prior knowledge of the environment, such as walls or intersection points. The technique is frequently connected with the visual effect that mixes computer images with real-world imagery. The first markerless systems interacted with available AR resources and defined their location and orientation in space by utilizing a device's location services and hardware.

The advancement of simultaneous localization and mapping (SLAM) technology enhanced the accuracy of markerless AR image processing. SLAM markerless image tracking analyses the surroundings and generates maps that show where virtual 3D objects should be placed. Even if the items are not in the user's line of sight, they do not move when the user moves, and the user does not have to scan new photos.

Markerless AR is Everywhere

The expansion of powerful camera systems, mobile operating systems (OS), and sensor technology in widely available mobile devices such as the iPhone aided in the accessibility of advanced AR applications. Anyone with a modern smartphone or tablet running the most recent version of the Android or iOS operating systems can enjoy markerless AR.

Markerless AR, sometimes known as "dead reckoning," use a combination of camera systems, specialised sensors, and algorithms to detect and map the real-world environment, such as the positions of walls and intersections. Without the requirement for a target image, an AR-enabled application can set virtual items in a real-world context and keep them immobile using a map of the area.

Different Types of Markerless AR Systems

In its most basic form, markerless AR superimposes virtual objects onto a static, pre-captured 2D image. Of course, it is not the most advanced method, and it blurs the line between augmented reality and photo manipulation. It is, however, simple and quick to deploy for apps that wish to provide offline AR experiences rather than live ones.

Markerless AR systems that use RGB-D SLAM and sensor fusion approaches are at the opposite extreme of the spectrum. Microsoft's HoloLens is the most notable example. These systems combine data from normal, red, green, and blue (RGB) cameras with cutting-edge infrared time-of-flight cameras to build a 3D map of the user's surroundings while the application is being used. This feature is critical to the SLAM tracking paradigm because it enables apps running on these devices to concretely place virtual content into the region.

Advantages of Markerless AR

Markerless AR, which eliminates the need for physical markers to activate virtual engagement, has the following benefits:

- Significantly increase the average range of motion while experiencing AR.
- Launch the programme from any location using a mobile device or a see-through headset such as glasses or goggles.
- Experience with others.
- Experience a wider field of view for AR content.
- With a map of the environment, an AR-enabled application can place virtual items in a real context and have them stay in position in the absence of a target image. For example, with Marxent's Relative Tracking, users can stroll around any open area (about 3-4 metres for most interior locations)—far beyond the range of any existing extended tracking.

Disadvantages of Markerless AR

Despite the benefits of markerless AR technology over earlier systems, challenges remain. The technique mainly relies on flat, textured surfaces to generate virtual visuals successfully. Apps that run on mobile devices use a lot of power as well. Adoption is slow due to a lack of acceptance for AR devices (glasses or headsets) and commercial investments. Nonetheless, as a gaming industry, AR headsets are increasing, with smaller markets forming in healthcare, academia, and industrial applications.

Table 5. Benefits and Challenges of Markerless AR

Benefits	**Challenges**
- Increase range of motion with AR - Use a headset to initialize an AR app - Share the experience - Wider field of view for AR content	- Depends on flat, textured surfaces - Apps running mobile consume a lot of power - Slow adoption

Markerless AR Software Applications

User-centric markerless tracking in VR headsets would allow for infinite virtual environments that could adapt to a variety of business demands. Markerless augmented reality software solutions that track motion inside a space have enormous promise to advance autonomous vehicle and robotics technologies.

Consider enhanced motorized wheelchairs that use markerless tracking technology. Wheelchair users may employ built-in obstacle avoidance and safe pathfinding technology to navigate the physical environment. They would activate by looking at a spot in front of them while wearing an eye-tracking headset. The same advancements could be employed for navigation in self-driving cars.

Industries Using Markerless AR

Retailers, educators, and game developers are examples of industries that are embracing markerless AR technology. As more industries adopt the technology, more data and input will enable designers and developers to create more consumer use cases. Expanding the number of recognised use cases for markerless AR necessitates affordable and accessible equipment as well as creative software applications. Here are a few industries that are embracing the experience:

Figure 8. Gaming Markerless AR

Gaming: The most major influence of markerless AR has been in gaming, but it is still in its early stages. A continuous trickle of AR games has been entering the market since the launching of Pokémon Go. With ARKit/ARCore, you can construct multi-user real-world games with fixed 3D material. Microsoft's HoloLens and Facebook's Oculus Rift are the market leaders in headgear AR gaming and communication. These tech titans are always creating new games and applications.

Figure 9. Microsoft's HoloLens and Facebook's Oculus Rift

Interior Design: Markerless AR allows you to use a mobile device to scan a real-world scene, such as a kitchen or living room, and virtually place an appliance or couch in the actual space. Shoppers can virtually arrange a whole space, swapping cabinet styles and flooring to see how they function before purchasing. Macy's, Amazon, IKEA, and Ashley Furniture are just a few companies that use augmented reality design and product visualisation to allow customers the freedom to shop and design from the comfort of their own homes. This trend is accelerating with each new generation of hardware and update to the AR SDKs.

Figure 10. Wayfair *implement* AR *technologies*

Retail: Amazon is exploring augmented reality to improve customer interaction and the online shopping experience for household items such as furniture. In the future, markerless AR may be used to show high-dollar products to buyers in 3D as an upgraded version of the try-before-you-buy experience. The opportunity to visually interact with and try on products such as jewelry, cosmetics, and apparel before purchasing them may help buyers who are apprehensive to buy these items online. Amazon is far from alone: Wayfair, Walmart, Made.com, and Alibaba are just a handful of the competitors vying to outperform Amazon in advanced AR experiences.

Figure 11. Advertising Markerless AR

Advertising: AR advertising began as tracker-based print advertisements: when the user held their device over the ad, a 3D character appeared. The technology liberated AR marketing from its print confinement, ushering it into the real world. Wayfair, the NBA, Sephora, and Gucci are among the companies that have already begun to use augmented reality advertising. The experience serves as a differentiating point for these large corporations. When wearables become more widely accepted, the potential for AR advertising will expand even further. Attendees and attendees at live events such as the Super Bowl might easily cycle between various virtual layers of advertising content and football-related experiences such as on-field statistics and instant replays.

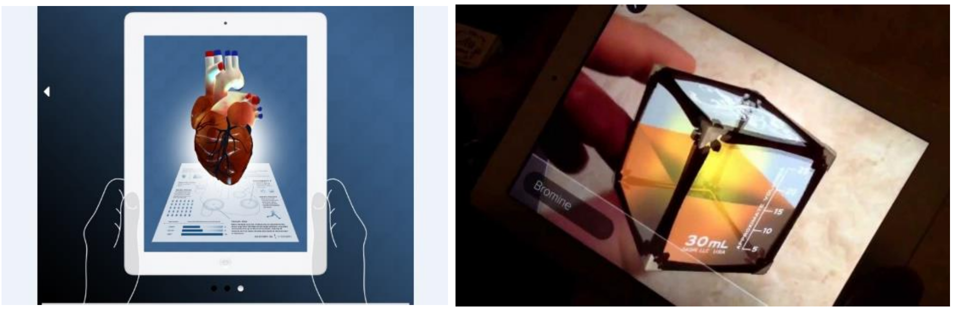


Figure 12. Anatomy 4D and Elements 4D

Education: Markerless AR's capacity to superimpose digital information on the actual environment is a boon to educators who use the tools to demonstrate complicated subjects. Apps like Anatomy 4D (for students learning about the human body) and Elements 4D (for those studying chemistry) are part of this early wave. Museums employ AR to supplement their physical exhibitions with useful digital content. Corporations and the United States military use Microsoft's HoloLens to develop training exercises and experiences ranging from repairing airplane engines to battlefield medicine and survival.

Figure 13. Healthcare Markerless AR

Healthcare: Consider a program that shows distinct color zones to identify the location of community flu and viral epidemics. Health officials send out real-time public health advisories to at-risk people via mobile applications that use the location of their smartphone. Surgeons can use sophisticated AI to perform virtual-assisted surgery on patients by superimposing 3D representations of human anatomy on them.

Figure 14. Tourism Markerless AR

Tourism: AR applications that are location-based assist vacationers in navigating foreign destinations. Visitors to Greece or Egypt may utilize augmented reality to view iconic sights such as the Parthenon and the Pyramids. An app can show these structures as they would have appeared to someone living thousands of years ago. Users who prefer the confines of an elite hotel can visit tourist hotspots and take in the sights without ever leaving their room. While sitting in front of your computer, online travel websites can take you on a virtual tour of your next vacation.

Entertainment: The availability of on-demand streaming content as well as high-definition audio and video has already changed how we watch movies and listen to music. By recreating live concerts and sporting events, markerless AR might usher in the next era of home entertainment. Consider sitting in on a live performance by your favorite singer or athlete.

The Future of Markerless AR

AR's future is projected to expand beyond consumer-focused applications and video games. Apple (ARKit and RealityKit) and Google (ARCore), the industry's major AR behemoths, continue to improve features and support for their AR developer platforms and devices.

With a market value of more than $18 billion expected in 2020, fusing immersive digital content with the actual environment has implications for all use cases. Improved imagery, commentary, and narration expand on these possibilities. Consider the significance of these aspects in distant teamwork. Consider the implications of markerless AR computing on the future of dispersed work and remote collaboration in a variety of industries, particularly healthcare and engineering.

AR computing platforms will be used by doctors, designers, engineers, marketers, and manufacturers to host virtual project rooms and communicate in real-time utilising virtual avatars. Working across any time zone utilising 3D information to review and scope projects without requiring travel has huge ramifications in a post-pandemic economy. Moving beyond simple video feeds to virtual avatars that move, collaborate, and communicate in the same virtual space opens the possibilities of complex workflows and synchronous collaboration that were previously reserved for in-person meetings.

CHAPTER 3

METHODOLOGY/PROJECT WORK

3. METHODOLOGY/PROJECT WORK

3.1 Development Life Cycle

This project will be created using the RAD, or Rapid Application Development, development approach. Rapid Application Development (RAD) is a fast-paced software development process that results in a finished product in a short period of time. Furthermore, the advantages of using this methodology include the fact that it does not waste time or money in generating the product, and it even shortens the time it takes to construct and implement information systems by involving many users.

The SDLC (System Development Life Cycle) is a set of processes that an organization must go through to assess, design, deploy, and maintain information systems. This methodology is typically used in huge projects that require lengthy development times of more than a year. The SDLC attempts to develop high-quality systems that meet or even exceed user expectations based on the user need by providing systems that go through each clearly defined phase within specified time constraints and cost estimates.

Rad Methodology:

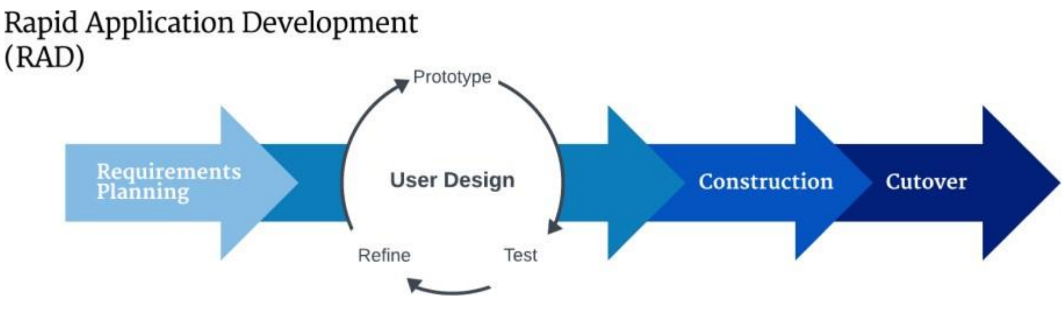


Figure 15. Rapid Application Development (RAD) methodology

The SDLC (Systems Development Life Cycle) consists of seven phases: planning, analysis, design, development, testing and integration, implementation, and maintenance. If any adjustments are required, the seven steps will not end but will continue to loop. The seven previously mentioned phases are listed below:

1. ***Planning***: The initial purpose is to do a preliminary analysis, provide potential solutions, identify costs and benefits, and submit a preliminary plan with recommendations after the issue statement has been identified.

2. ***Analysis***: The objectives, as well as the nature and extent of the problem under investigation, are discussed in this step. We need to figure out what the development system's goals are. Then we must consider how the problem under investigation fits into the overall project.

3. ***Design*****:** Gantt Chart is used to describe desired features and procedures in detail, including system diagrams, task workflow, UI (User Interface) drawings for eventual AR apps, and other documentation.

4. ***Development*****:** At this point, the real code is written in Python and executed using Google Colab and Jupyter Notebook.

5. ***Integration and testing****:* This phase involve putting all the elements together in a unique testing environment and checking for mistakes, defects, and interoperability.

6. ***Implementation*****:** The stage of initial development during which the project is put into development and runs in its final state.

7. ***Maintenance*****:** The system is evaluated at this stage of the system development life cycle (SDLC) to ensure it does not become obsolete. Changes to the initial project phase are also made here. It entails a constant assessment of the project's performance and functions.

Structure methodology refers to the framework used to structure, plan, manage, and sustain the development process of an information system. As a result, choosing the right technique is crucial for guaranteeing a smooth process development flow. Selecting the appropriate technique, for example, can minimise development time and expense.

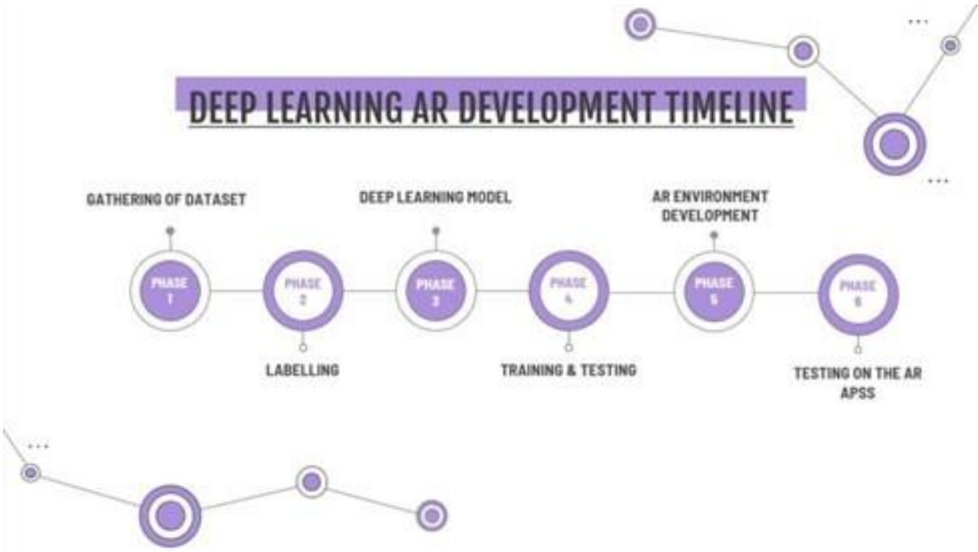


Figure 16. Deep Learning AR Development Timeline.

These are the six most important stages in the project's progression. The first phase is Gathering the dataset, which is the planning stage in which I will conduct research and collect all the essential data to be used in the project. In the second phase, I will label all the required objects with a labelling software called SuperAnnotate. The third process is model construction, which comprises the creation of the deep learning model. At the same time, in the fourth phase, I will train and test all my datasets, as well as compile the results. The fifth phase is the construction of the AR Environment, commonly known as the Graphic User Interface (GUI). I will be designing the graphical user interface (GUI) in this section. Finally, the graphical user interface (GUI) is being tested.

3.2 GATHERING OF DATASET (PHASE 1)

3.2.1 Introduction

First, we research and collect all the necessary data that will be required later. As for the dataset, we will be focusing on the various types of teeth that exist in our mouth, thus we will need numerous photographs or videos of people showcasing their teeth, whether from outside or inside their mouth. The dataset that I gathered is shown below:

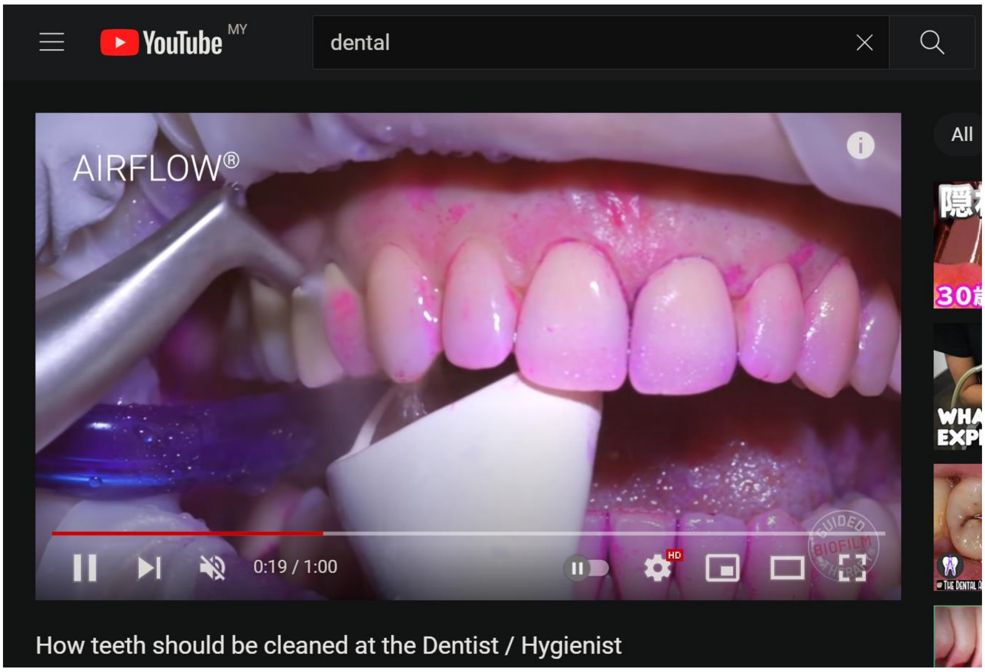
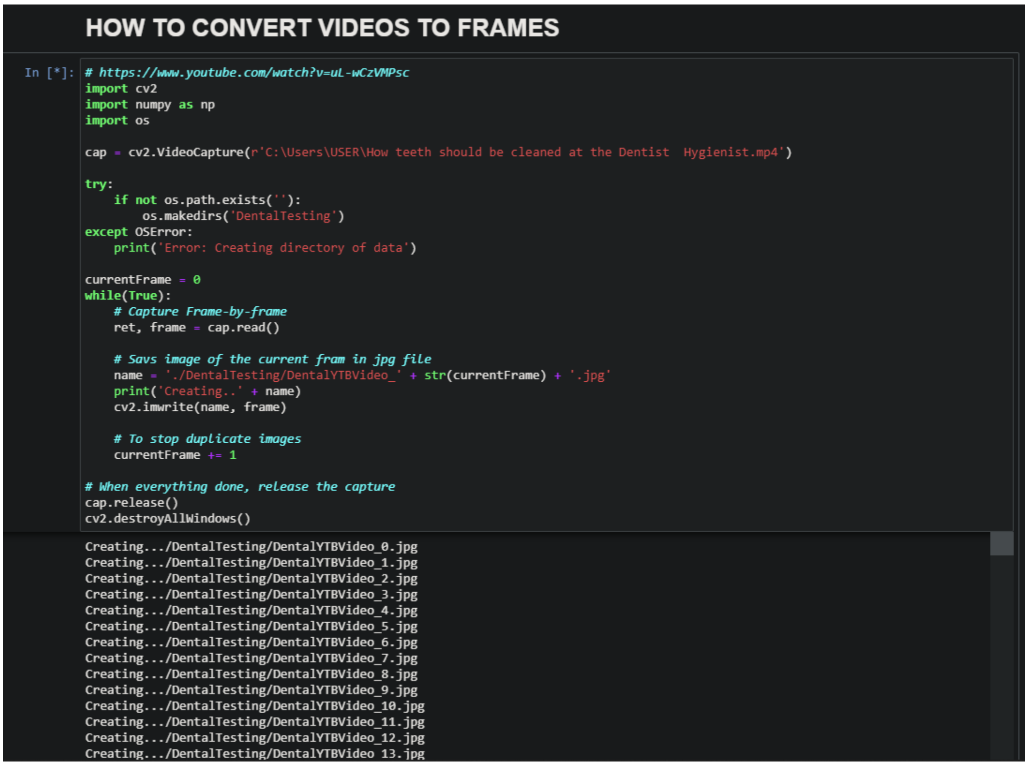


Figure 17. Dental YouTube video

From the YouTube video, I converted it into frames using python codes. I refer to this YouTube video link as for my dataset, https://www.youtube.com/watch?v=3wvVrkMKim8 .The python codes I used to transform the video above into frames are as described below.



HOW TO CONVERT VIDEOS TO FRAMES

```
In [*]: # https://www.youtube.com/watch?v=uL-wCzVMPsc
import cv2
import numpy as np
import os

cap = cv2.VideoCapture(r'C:\Users\USER\How teeth should be cleaned at the Dentist  Hygienist.mp4')

try:
    if not os.path.exists(''):
        os.makedirs('DentalTesting')
except OSError:
    print('Error: Creating directory of data')

currentFrame = 0
while(True):
    # Capture Frame-by-frame
    ret, frame = cap.read()

    # Savs image of the current fram in jpg file
    name = './DentalTesting/DentalYTBVideo_' + str(currentFrame) + '.jpg'
    print('Creating..' + name)
    cv2.imwrite(name, frame)

    # To stop duplicate images
    currentFrame += 1

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

Creating.../DentalTesting/DentalYTBVideo_0.jpg
Creating.../DentalTesting/DentalYTBVideo_1.jpg
Creating.../DentalTesting/DentalYTBVideo_2.jpg
Creating.../DentalTesting/DentalYTBVideo_3.jpg
Creating.../DentalTesting/DentalYTBVideo_4.jpg
Creating.../DentalTesting/DentalYTBVideo_5.jpg
Creating.../DentalTesting/DentalYTBVideo_6.jpg
Creating.../DentalTesting/DentalYTBVideo_7.jpg
Creating.../DentalTesting/DentalYTBVideo_8.jpg
Creating.../DentalTesting/DentalYTBVideo_9.jpg
Creating.../DentalTesting/DentalYTBVideo_10.jpg
Creating.../DentalTesting/DentalYTBVideo_11.jpg
Creating.../DentalTesting/DentalYTBVideo_12.jpg
Creating.../DentalTesting/DentalYTBVideo_13.jpg
```

Figure 18. Python code to convert video to frames

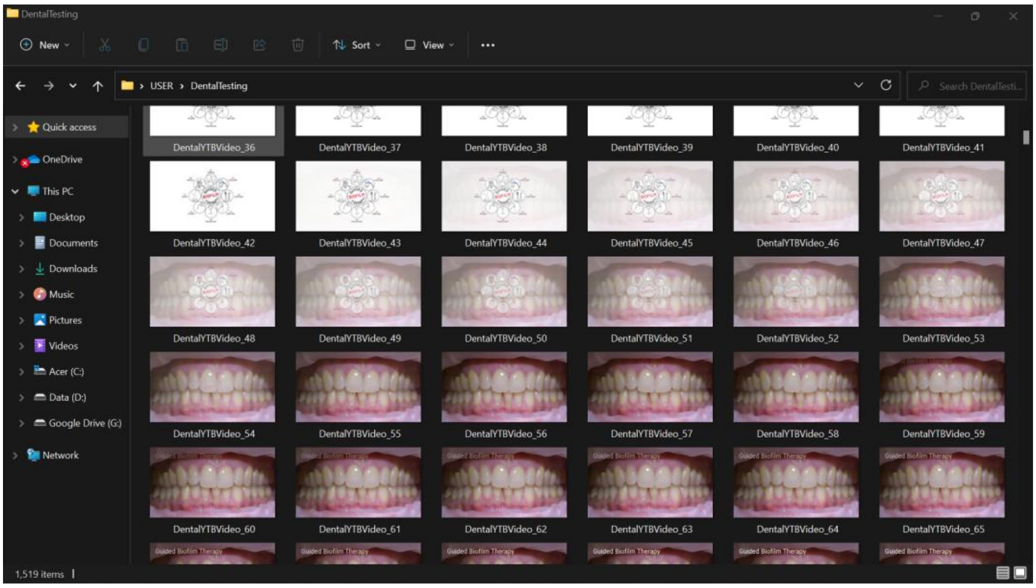


Figure 19. Frames from the conversion

3.3 Labelling (Phase 2)

By using a labelling software called SuperAnnotate, I able to create a bounding box around the objects that I want, and those x and y coordinates will be compiled in each txt files according to their frames.

Figure 20. *SuperAnnotate*

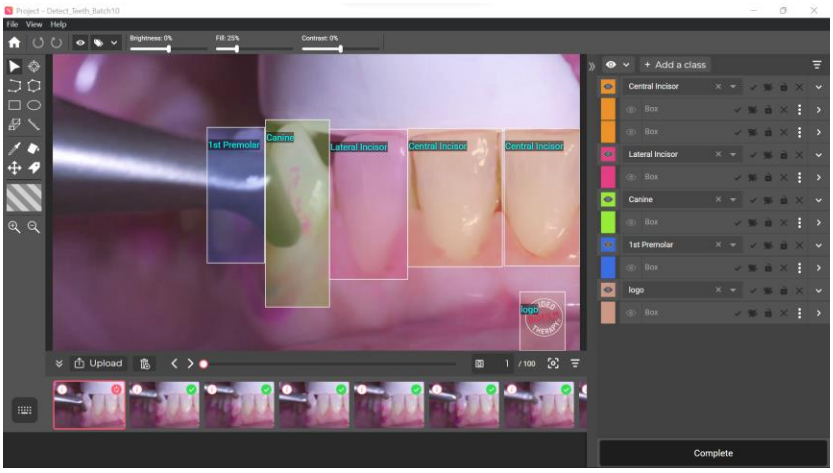


Figure 21. SuperAnnotate interface

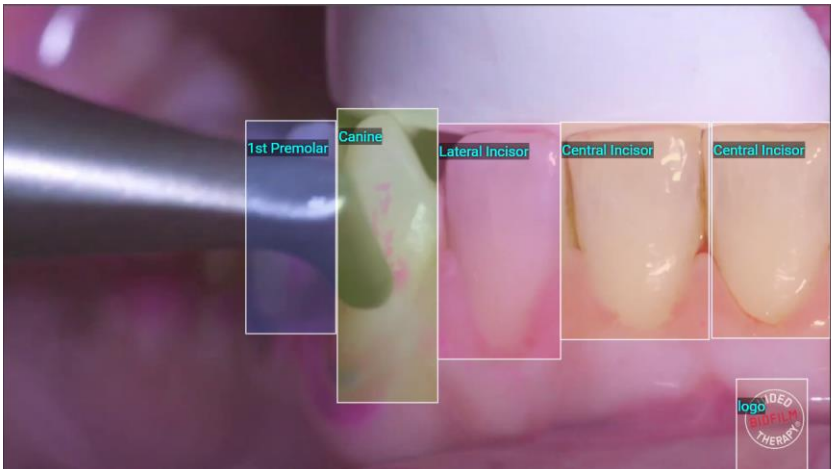


Figure 22. Frames that are labelled by SuperAnnotate

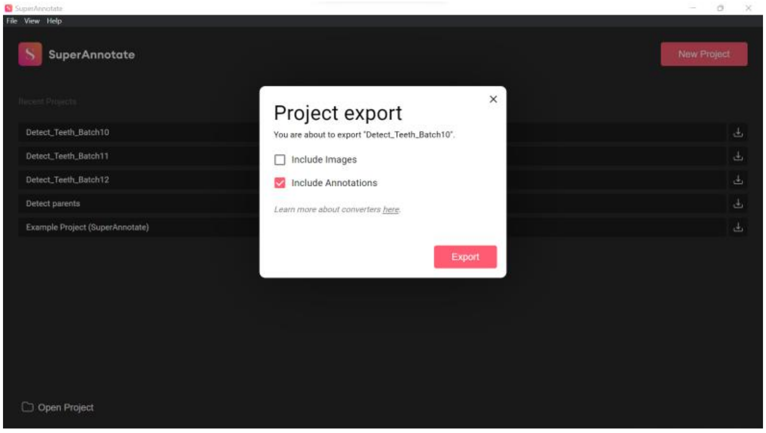


Figure 23. Export dataset

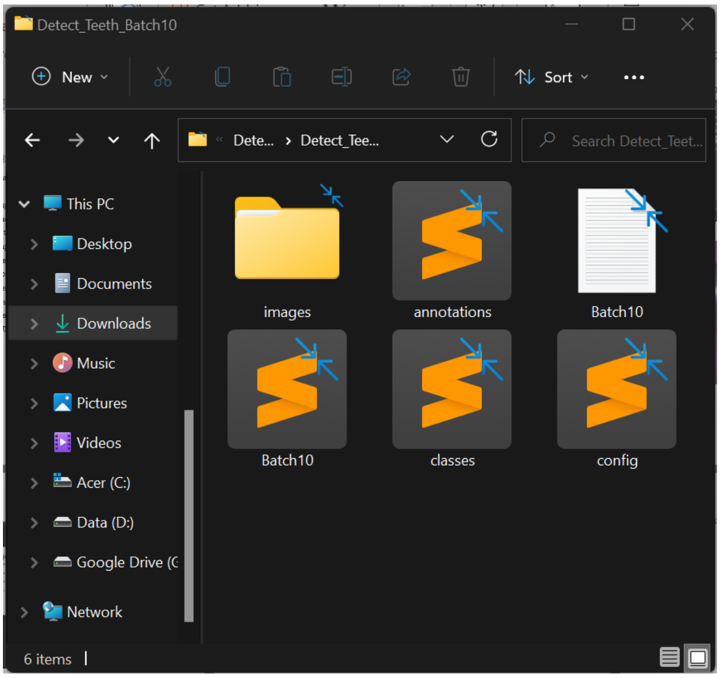


Figure 24. After export per batch

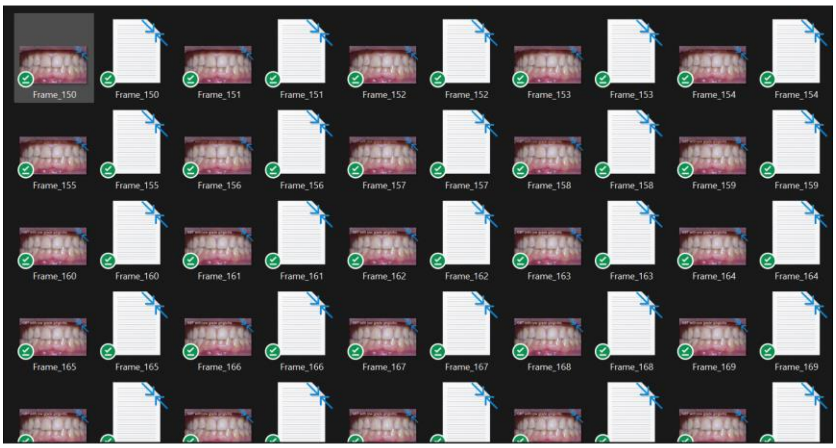


Figure 25. Inside the images folder

While working on the deep learning model, the dataset is divided into two folders: train and test. Then, inside each of the train and test folders, there will be images and labels folders, as mentioned further below.

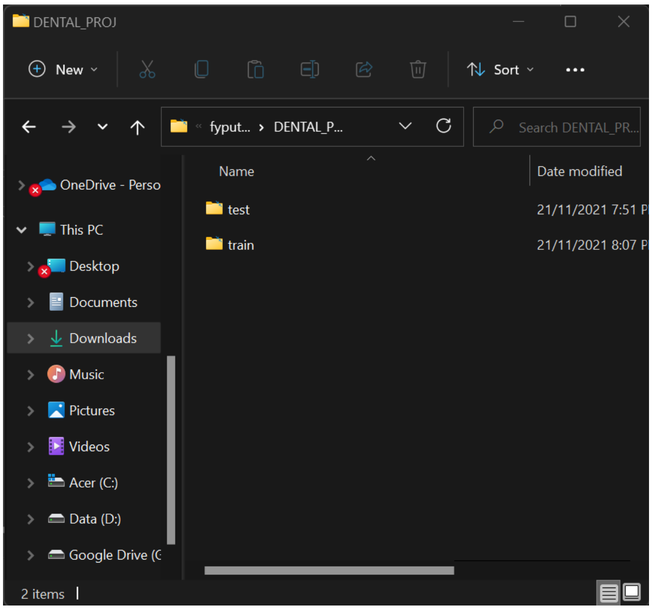


Figure 26. Test and Train folders

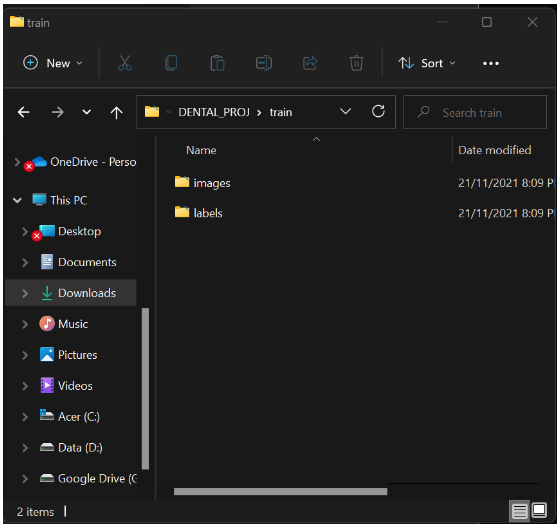


Figure 27. Images and labels folder inside Train parent folder

3.4 Deep Learning Model (Phase 3)

3.4.1 Yolov5

Below are the python codes of YOLOv5 model that I have done. First, we clone the YOLOV5 from the official repo to our computer.

Figure 28. Cloning the YOLOV5 from the official repository

Next, we unzip our dataset called main.zip from the GitHub using the below lines code.

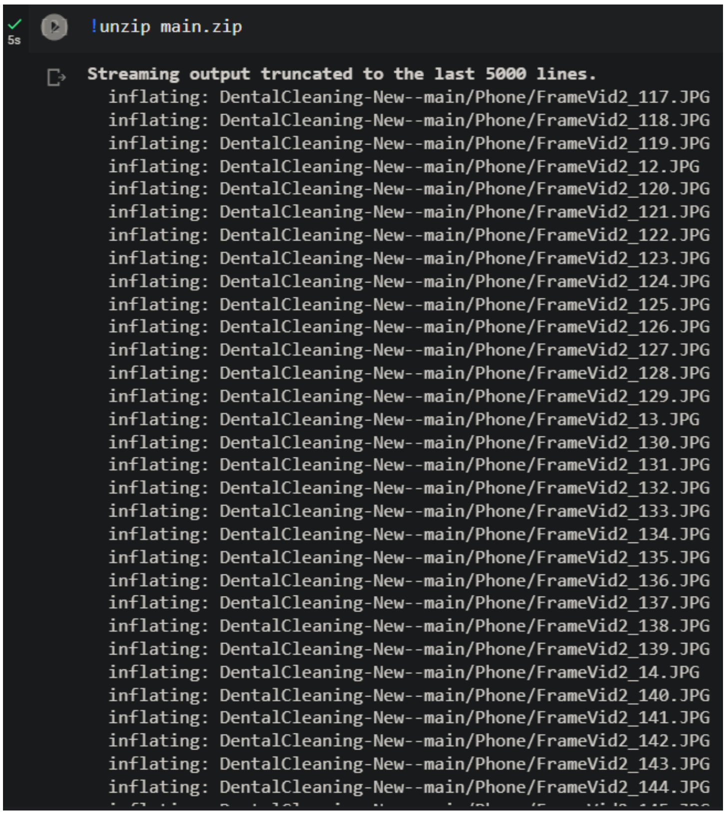


Figure 29. Unzip the zip file

After unzip the dataset, we clone the file to our own colab environment using the git clone codes.

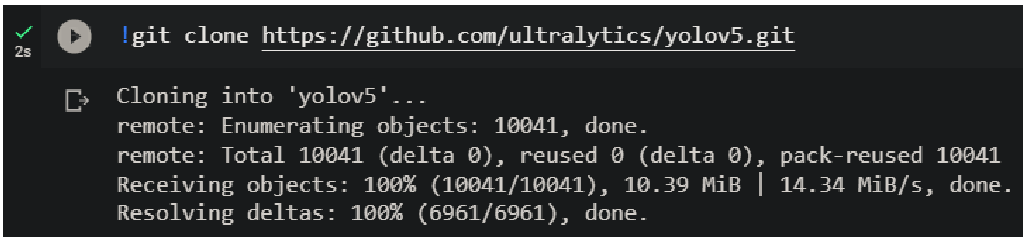


Figure 30. Git clone the file

Later, we change directory to YOLOv5 folder as shown below.

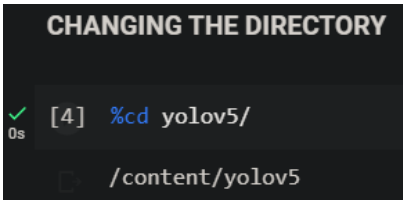


Figure 31. Change the directory

We need to install the required dependencies which necessary for the YOLOV5 to run properly.

Figure 32. Installing the dependencies

We then need to follow below instruction to test and run on our own custom dataset.

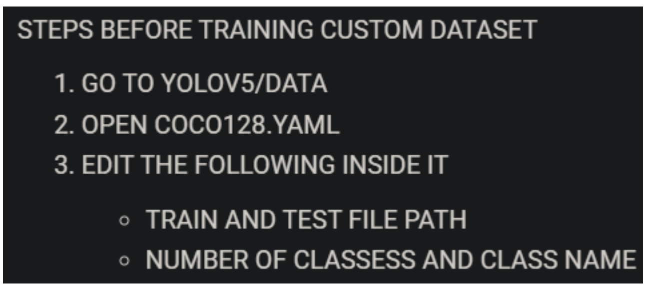


Figure 33. Instruction to test on our custom dataset

After all the configuration above, we will now start our training part. We implement the train.py from the YOLOV5 repo, we give python command and pass the argument of --img 700, --batch 8, --epochs 100, --data coco128.yaml, weights yolov5s.pt etc.

TRAINING THE YOLOV5 MODEL

```
!python train.py --img 700 --batch 8 --epochs 100 --data coco128.yaml --weights yolov5s.pt --nosave --cache
val: Scanning '/content/DentalCleaning-New--main/Test/labels' images and labels...852 found, 0 missing, 0 empty, 0 corrupted:
val: New cache created: /content/DentalCleaning-New--main/Test/labels.cache
val: Caching images (0.7GB ram): 100% 852/852 [00:24<00:00, 35.46it/s]
Plotting labels to runs/train/exp/labels.jpg...

AutoAnchor: 5.55 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Image sizes 704 train, 704 val
Using 2 dataloader workers
Logging results to runs/train/exp
Starting training for 100 epochs...

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      0/99     2.04G   0.09152    0.1246   0.05864        35       704: 100% 157/157 [02:10<00:00,  1.20it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 54/54 [00:17<00:00,  3.04it/s]
                 all        852       9318      0.358      0.163     0.0879     0.0303

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      1/99      2.2G   0.06739    0.0906   0.04847       106       704: 100% 157/157 [02:07<00:00,  1.23it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 54/54 [00:16<00:00,  3.21it/s]
                 all        852       9318       0.35      0.356      0.138     0.0487

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      2/99      2.2G   0.06366   0.08295   0.04514        47       704: 100% 157/157 [02:06<00:00,  1.24it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 54/54 [00:16<00:00,  3.30it/s]
                 all        852       9318      0.488      0.314      0.136      0.056

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      3/99      2.2G   0.05947   0.08004   0.03964       106       704: 100% 157/157 [02:06<00:00,  1.24it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 54/54 [00:16<00:00,  3.36it/s]
```

Executing (31m 45s) Cell > system() > _system_compat() > _run_command() > _monitor_process() > _poll_process()

Figure 34. Training the YOLOV5 Model

On the next lines, we will visualize the training metrics via Tensorboard as mentioned below.

VISUALIZING THE TRAINING METRICS

```
[ ] %load_ext tensorboard
    %tensorboard --logdir runs/train
```

Figure 35. Visualizing the Training Metrics

We now proceed on the testing part whereas we run the line code below, we can see the result after we train our custom dataset as mentioned above.

TESTING

```
[ ] !python val.py --weights /content/yolov5/runs/train/exp/weights/last.pt --data coco128.yaml --img 800
```

Figure 36. Testing on our custom dataset

So now, we can take any other pictures outside our dataset to test on the accuracy.

DETECTING NEW IMAGES

```
!python detect.py --source /content/DentalCleaning-New--main/Test --weights /content/yolov5/runs/train/exp/weights/last.pt --img 700 --save-txt --save-conf
```

Figure 37. Detect on new images outside our dataset

3.5 Training & Testing (Phase 4)

The dataset will then be divided into two parts: train and test folders. As shown here, the split ratio is roughly 70/30, with 70% being for the train and 30% being for the test.

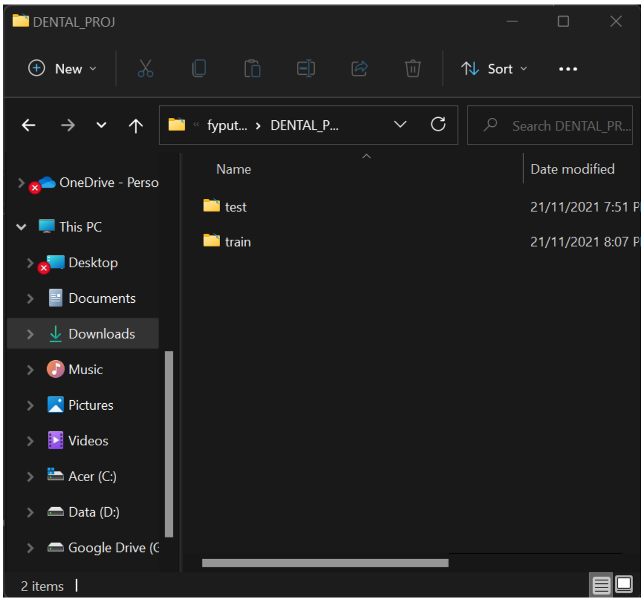


Figure 38. Test and Train folders

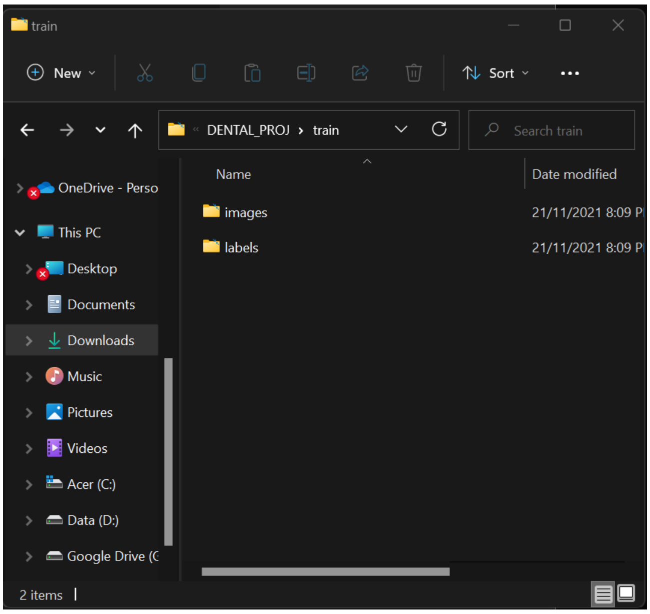


Figure 39. Images and labels folder inside Train parent folder

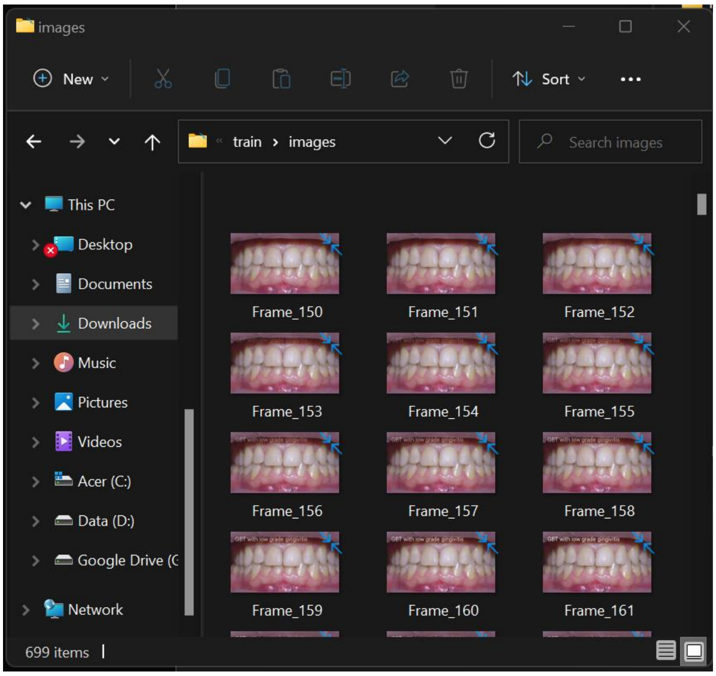


Figure 40. Images

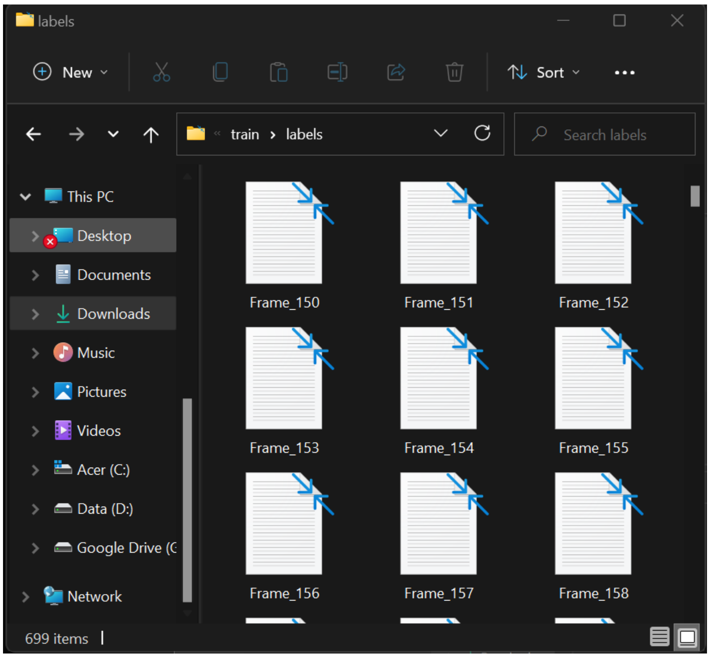


Figure 41. Labels

The sequence for test folders is then the same as for train folders above. In contrast to the train folders, the test folders have 300 photographs and labels in total, as we adhere to the 70:30 rule.

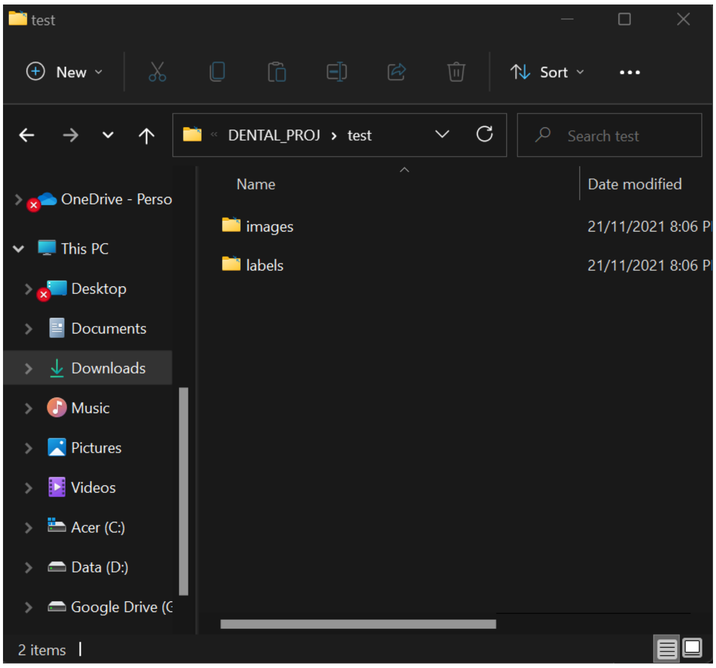


Figure 42. Images and labels folder inside Test parent folder

After splitting the dataset, we will train and test it with the YOLOv5 model, and the results will be as follows:

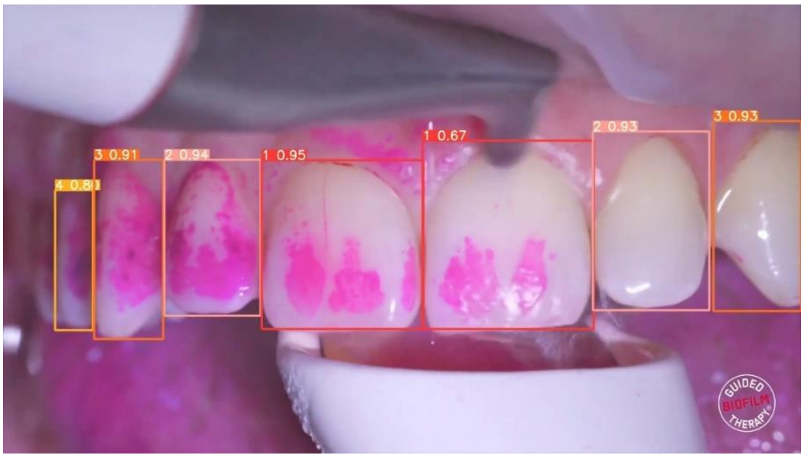


Figure 43. Result from testing the labelled frames

3.6 Graphic User Interface (Phase 5)

In terms of the graphic user interface, I created a simple GUI for the dental software, as shown below. There will be two modes: LEARN MODE and QUIZ MODE. The overall GUI I built is shown below:

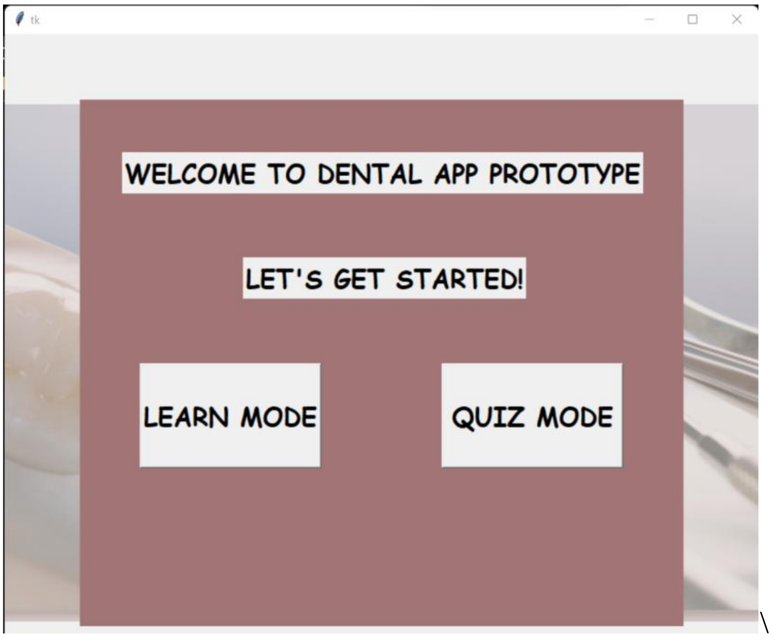
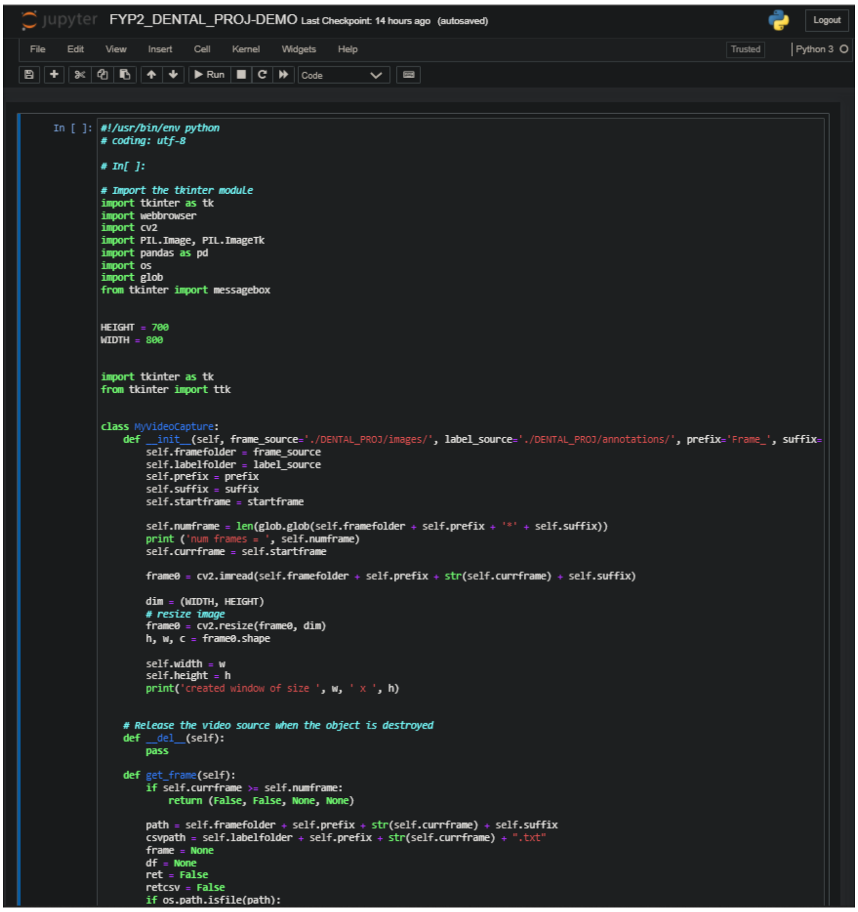


Figure 44. Graphic User Interface for prototype Dental App

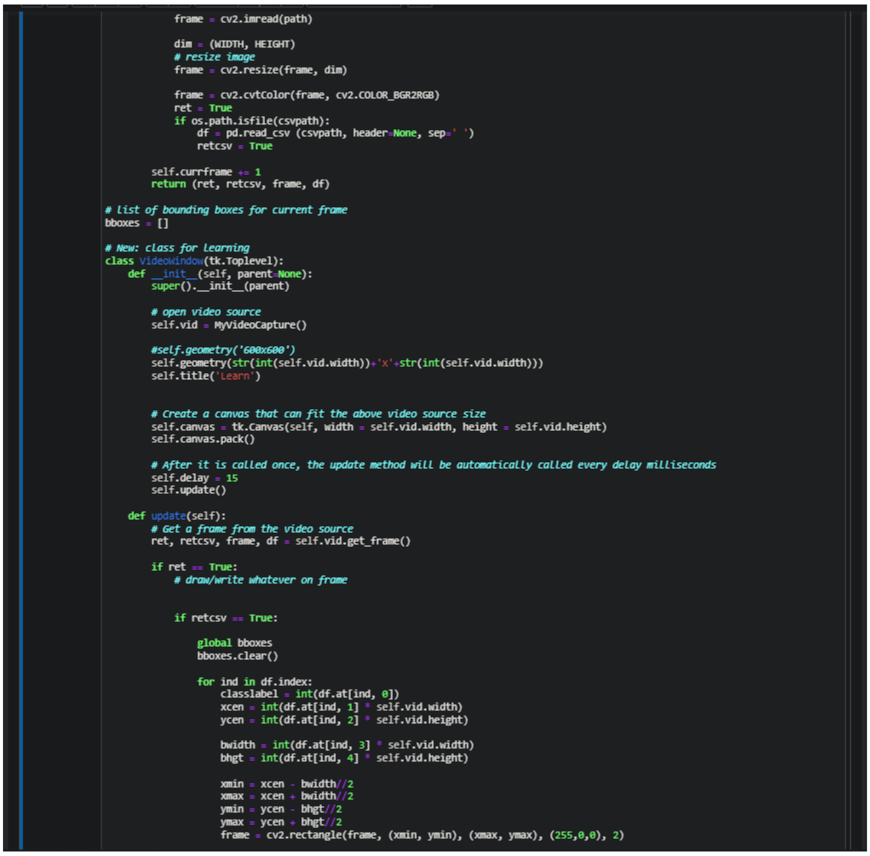
Firstly, with the implementation of the jupyter notebook, I will design the graphic user interface (GUI) for the Dental AR app prototype. The Python codes for the GUI is as follows:

3.6.1 Jupyter Notebook



```python
#!/usr/bin/env python
# coding: utf-8

# In[ ]:

# Import the tkinter module
import tkinter as tk
import webbrowser
import cv2
import PIL.Image, PIL.ImageTk
import pandas as pd
import os
import glob
from tkinter import messagebox


HEIGHT = 700
WIDTH = 800


import tkinter as tk
from tkinter import ttk


class MyVideoCapture:
    def __init__(self, frame_source='./DENTAL_PROJ/images/', label_source='./DENTAL_PROJ/annotations/', prefix='Frame_', suffix=
        self.framefolder = frame_source
        self.labelfolder = label_source
        self.prefix = prefix
        self.suffix = suffix
        self.startframe = startframe

        self.numframe = len(glob.glob(self.framefolder + self.prefix + '*' + self.suffix))
        print ('num frames = ', self.numframe)
        self.currframe = self.startframe

        frame0 = cv2.imread(self.framefolder + self.prefix + str(self.currframe) + self.suffix)

        dim = (WIDTH, HEIGHT)
        # resize image
        frame0 = cv2.resize(frame0, dim)
        h, w, c = frame0.shape

        self.width = w
        self.height = h
        print('created window of size ', w, ' x ', h)


    # Release the video source when the object is destroyed
    def __del__(self):
        pass

    def get_frame(self):
        if self.currframe >= self.numframe:
            return (False, False, None, None)

        path = self.framefolder + self.prefix + str(self.currframe) + self.suffix
        csvpath = self.labelfolder + self.prefix + str(self.currframe) + ".txt"
        frame = None
        df = None
        ret = False
        retcsv = False
        if os.path.isfile(path):
```

Figure 45. Python codes via Jupyter Notebook for GUI part1



```python
            frame = cv2.imread(path)

            dim = (WIDTH, HEIGHT)
            # resize image
            frame = cv2.resize(frame, dim)

            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            ret = True
            if os.path.isfile(csvpath):
                df = pd.read_csv (csvpath, header=None, sep=' ')
                retcsv = True

        self.currframe += 1
        return (ret, retcsv, frame, df)

# list of bounding boxes for current frame
bboxes = []

# New: class for learning
class VideoWindow(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)

        # open video source
        self.vid = MyVideoCapture()

        #self.geometry('600x600')
        self.geometry(str(int(self.vid.width))+'x'+str(int(self.vid.width)))
        self.title('Learn')


        # Create a canvas that can fit the above video source size
        self.canvas = tk.Canvas(self, width = self.vid.width, height = self.vid.height)
        self.canvas.pack()

        # After it is called once, the update method will be automatically called every delay milliseconds
        self.delay = 15
        self.update()

    def update(self):
        # Get a frame from the video source
        ret, retcsv, frame, df = self.vid.get_frame()

        if ret == True:
            # draw/write whatever on frame


            if retcsv == True:

                global bboxes
                bboxes.clear()

                for ind in df.index:
                    classlabel = int(df.at[ind, 0])
                    xcen = int(df.at[ind, 1] * self.vid.width)
                    ycen = int(df.at[ind, 2] * self.vid.height)

                    bwidth = int(df.at[ind, 3] * self.vid.width)
                    bhgt = int(df.at[ind, 4] * self.vid.height)

                    xmin = xcen - bwidth//2
                    xmax = xcen + bwidth//2
                    ymin = ycen - bhgt//2
                    ymax = ycen + bhgt//2
                    frame = cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (255,0,0), 2)
```

Figure 46. Python codes via Jupyter Notebook for GUI part2

```python
                bboxes.append((classlabel, xmin, ymin, xmax, ymax))


            self.photo = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(frame))
            self.canvas.create_image(0, 0, image = self.photo, anchor = tk.NW)

        self.after(self.delay, self.update)

new = 1
url = "https://github.com/Ndinie/Dental_dataset"
LEARNINGMODE = 0
QUIZMODE = 1
currMode = LEARNINGMODE


def openbrowser():
    webbrowser.open(url,new=new)


def mouseclicked(event):
    print ("clicked at", event.x,',', event.y)
    global bboxes
    for box in bboxes:
        if event.x < box[1] or event.x > box[3]:
            continue
        if event.y < box[2] or event.y > box[4]:
            continue

        print ('Clicked on a box with label ', box[0])
        messagebox.showinfo("Information","You clicked on tooth "+str(box[0]))

        # New: create learning window
def open_lwindow():
        lwindow = VideoWindow()
        lwindow.bind("<Button-1>", mouseclicked)
        currMode = LEARNINGMODE
        lwindow.grab_set()

                # New: create learning window
def open_qwindow():
        lwindow = VideoWindow()
        lwindow.bind("<Button-1>", mouseclicked)
        currMode = QUIZMODE
        lwindow.grab_set()

root = tk.Tk()

canvas = tk.Canvas(root, height=HEIGHT, width=WIDTH)
canvas.pack()

background_image = tk.PhotoImage(file=r'D:\Acer Kiyoku\FINAL YEAR FINAL SEM (8)\FYP2\Dental.png')
background_label = tk.Label(root, image=background_image)
background_label.place(relwidth=1, relheight=1)

frame = tk.Frame(root, bg='#a17575')
frame.place(relx= 0.1, rely=0.1, relwidth=0.8, relheight=0.8)

button = tk.Button(frame, text = "LEARN MODE", font=20, command=open_lwindow)  #New: call open_window function
button.place(relx= 0.1, rely= 0.5, relwidth=0.3, relheight=0.2)

button1 = tk.Button(frame, text = "QUIZ MODE", font=20)
button1.place(relx= 0.6, rely= 0.5, relwidth=0.3, relheight=0.2)

label = tk.Label(frame, text="WELCOME TO DENTAL APP PROTOTYPE", font=100)
label.pack()
```

Figure 47. Python codes via Jupyter Notebook for GUI part3

```python
label1 = tk.Label(frame, text="LET'S GET STARTED!", font=50)
label1.place(rely= 0.3, relx= 0.27)

Font_tuple = ("Comic Sans MS", 20, "bold")

# Parsed the specifications to the
# Text widget using .configure( ) method.
label.configure(font = Font_tuple)
label1.configure(font = Font_tuple)
button.configure(font = Font_tuple)
button1.configure(font = Font_tuple)

root.mainloop()
```

Figure 48. Python codes via Jupyter Notebook for GUI part4

Later, when the user clicks on the LEARN MODE button, the user will see a dental video that shows mouth and teeth with a bounding box around them. As the user clicks on one of the teeth, a text will appear on top of the video explaining what kind of teeth the user just clicks on before.

Next, on the QUIZ MODE, the user will have a small quiz where the user will encounter the same video but with different text after they click on the teeth. The text will ask them whether the teeth are this kind of teeth and if they answer it correctly, a text will appear shows “You did a good job” and if they answer it wrong “Please try again”.

3.7 Testing (Phase 6)

3.7.1 Methods

As we run the codes, we will see the GUI below where there are two buttons of LEARN MODE and QUIZ MODE.

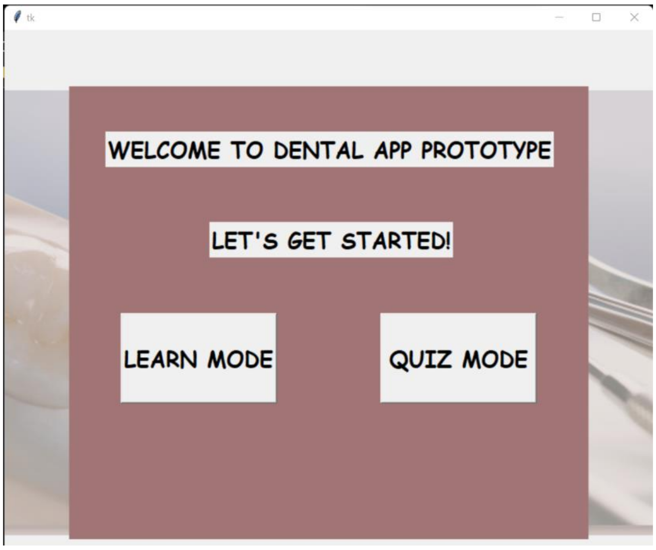


Figure 49. Dental App Prototype GUI

Inside the LEARN MODE, we could see a dental video with labelled teeth. In this LEARN MODE, user will click on one of the teeth to identify which types of teeth they clicked on.

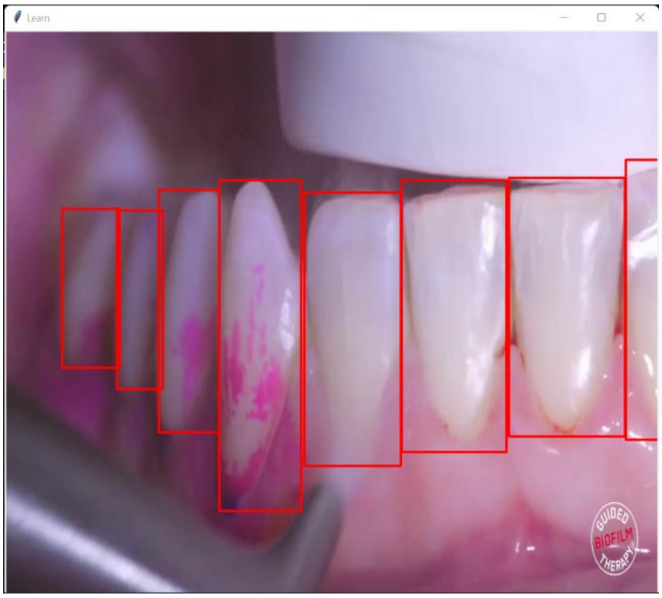


Figure 50. Dental Video with labelled teeth

As they clicked on the teeth, this text box will appear to show them which teeth they clicked on.

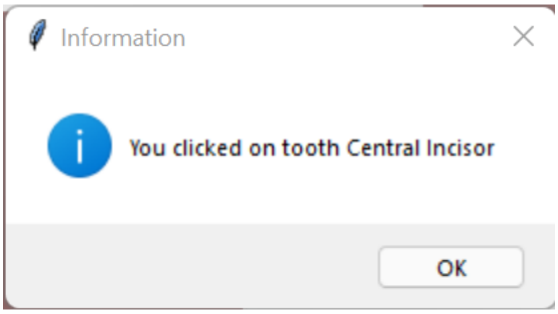


Figure 51. Pop-up Text Box

Later, the results show the summary of which teeth the user has selected.

```
num frames =  1100
created window of size  800  x  700
num frames =  1100
created window of size  800  x  700
clicked at 699 , 318
You have clicked on a box with label  Central Incisor
clicked at 621 , 367
You have clicked on a box with label  Central Incisor
clicked at 450 , 369
You have clicked on a box with label  Lateral Incisor
clicked at 358 , 393
You have clicked on a box with label  Canine
clicked at 250 , 404
You have clicked on a box with label  1st Premolar
clicked at 197 , 385
You have clicked on a box with label  2nd Premolar
clicked at 106 , 338
You have clicked on a box with label  1st Molar
```

Figure 52. Summary of what user had clicked

3.8 Project Gantt Chart

As previously indicated, I am conducting research on YOLOv5 model object identification on the FYP1 timeline, as seen in the first Gantt Chart below.

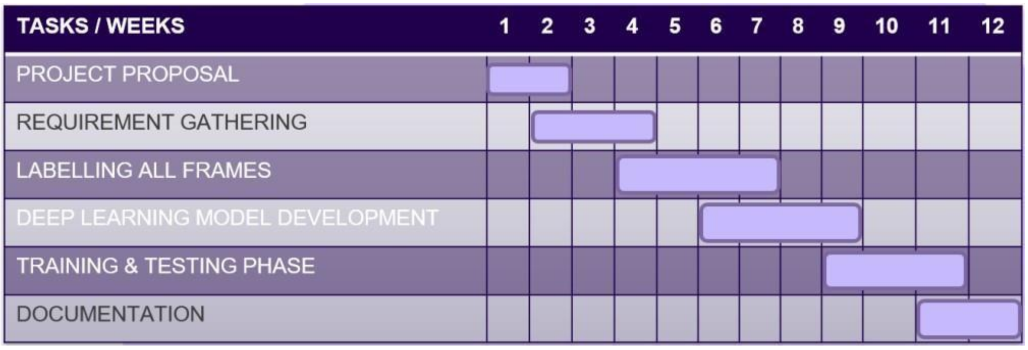


Figure 53. FYP1 Gantt Chart

We must submit our project concept to the supervisor of our choice and get it approved during the week between weeks 2 and 3. My supervisor, Dr. Nordin, suggested a project that was subsequently allowed by the criteria, as we discussed before the semester began. As a result, after submitting it, I spent week 2 to week 4 gathering criteria from my project's articles and journals and studying which tools are suited for the labelling phase, ultimately electing to use superannotate and later labelimg, labelling software.

I started preparing and identifying my data from week 4 to week 7. After I finished identifying them all, I decided to build my deep learning model using the YOLO method, which I did from week 6 to week 9. As a result, between weeks 9 and 10, I started training and testing all datasets, as well as managing any issues that appeared. Finally, during the documentation step, I will compile all the results into an interim report, exactly like I did throughout the training and testing phases.

3.9 Methods And Tools

These are some of the software and frameworks that I have studied and used to assist me accomplish the project throughout this Final Year Paper 1 (FYP1) and (FYP2), and most of the tools I use are familiar to me. These are the applications:

Table 6. Tools

SuperAnnotate	SuperAnnotate Desktop is the fastest image and video annotation application. It was built using SuperAnnotate's web platform, which was built with feedback from thousands of annotators who labelled for hundreds of thousands of hours. The tool allows computer vision engineers or small annotation teams to quickly annotate photographs and videos while also providing efficient features for communicating information and even exploring existing labelled datasets.
Labelimg	LabelImg, a Python-based image annotation tool with a Qt-based graphical interface, is a popular option. ImageNet employs the YOLO format, which may be saved as PASCAL VOC XML files.
Python Programming Language	Python is an immensely popular high-level, general-purpose programming language. The Python programming language (the most recent version is Python 3) is used in web development, machine learning applications, and all other cutting-edge software technology. Python is a great programming language for both novices and experienced programmers who have worked with other languages such as C++ and Java.

YOLO	YOLO is an expression for 'You Only Look Once.' This is an algorithm for detecting and identifying various objects in an image (in real- time). Object recognition in YOLO is done as a regression problem, and the class probabilities of the discovered photographs are reported.
Google Colab	Google Colaboratory is a free Jupyter notebook environment that runs on Google cloud servers and lets users access backend hardware such as GPUs and TPUs. This allows you to do anything you could do in a Jupyter notebook on your local system without having to install or set up a notebook.
OpenCV	OpenCV is a significant open-source library for computer vision, machine learning, and image processing that is currently used in real-time activities, which are crucial in today's systems. It can recognize objects, faces, and even human handwriting in photographs and videos.
Jupyter Notebook	Jupyter Notebooks are a derivative of the IPython project, which formerly had an IPython Notebook project. Jupyter gets its name from the programming languages it supports: Julia, Python, and R. Jupyter comes with the IPython kernel, which allows you to develop Python programs, but there are over 100 alternative kernels available. (Real Python, 2021)

In terms of the FYP1 timeline, I am using the above approach in my research on YOLO model object detection. The labelling programme, which I used to label all my photos under the Dental dataset, will subsequently give an XML file containing the ground truth, including x and y coordinates of the location of each tagged object inside each of the frames.

Following that, the frames and XML files will be used for deep learning development in Google Colab. Before doing so, all the frames required to be separated into train and test folders in a 70:30 ratio. In terms of deep learning development, I will use a supervised model, training and testing my dataset to determine performance accuracy, and I will also incorporate the YOLO model.

The system diagram for the deep learning development model employing YOLOv5 models is shown below.

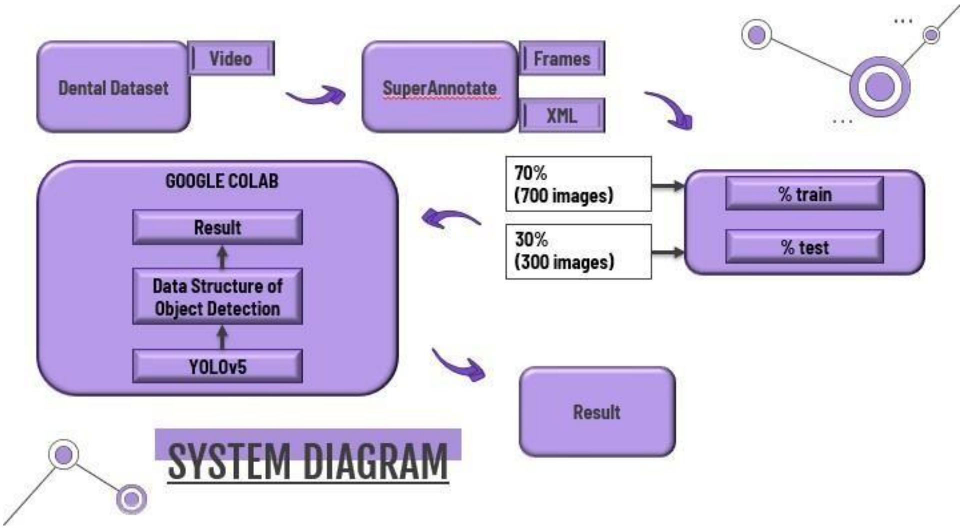


Figure 54. System Diagram

The deep learning AR development timeline for FYP1 and FYP2 is shown below.

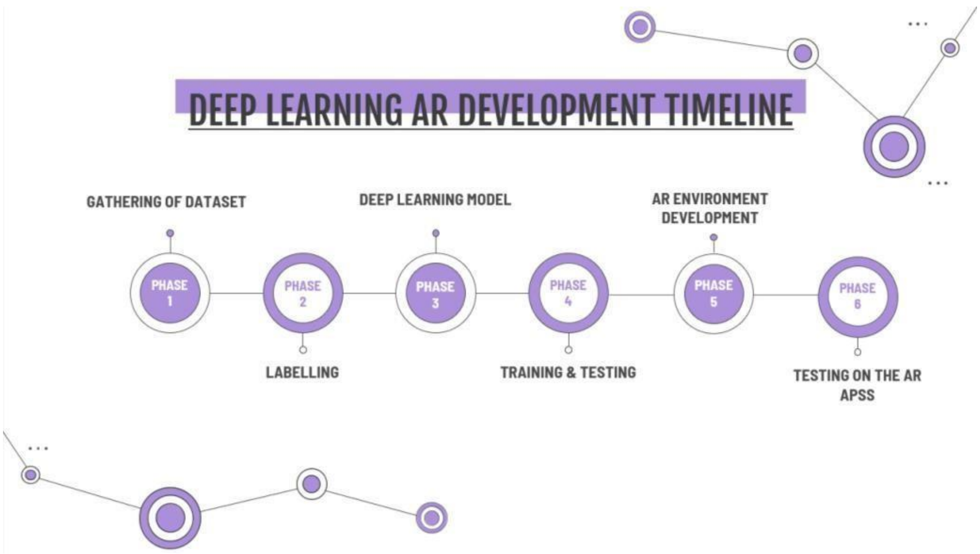


Figure 55. Deep Learning AR Development Timeline

CHAPTER 4

RESULT AND DISCUSSION

4. RESULT AND DISCUSSION

4.1 Introduction

Augmented reality (AR) technology extends virtual knowledge by incorporating real-world experiences, opening new prospects in a variety of fields. Furthermore, unlike VR, AR does not require any large pieces of equipment, allowing people to feel at ease with it in a sense of freedom. (2018 Practice)

Because augmented reality technology is widely employed in medical education and training, most augmentation medical applications are centered on surgical kinds such as neurosurgery, laparoscopic surgery, and plastic surgery. The patient can also be shown how the process is carried out in a virtual environment, allowing them to improve their experience. Oral and maxillofacial surgery is the most common use, with dental implant implantation and orthognathic surgery being the most common. Furthermore, the dentist can gain access to the most recent applications of restorative dentistry, orthodontics, endodontics, and other dental specialties. (Kwon et al., 2018)

4.2 Software Markerless Augmented Reality (Ar)

According to Schechter's (2020) research, markerless augmented reality (AR) is a software application that does not require prior knowledge of the user's environment to overlay virtual 3D material into a scene and hold it to a fixed point in space. This type of AR, also known as recognition-based AR or image recognition, is dependent on the identification of markers or user-defined images to function. To activate an augmentation in marker-based AR, a marker must be used.

4.3 System Overview

The programming languages that I used mainly is python programming language where I have implemented it to convert the video into frames, build the YOLOV5 model, create a basics GUI for the dental app etc.

4.4 System Architecture

The system architecture is shown below where there are two functions inside the system which called LEARN MODE and QUIZ MODE. We could see from the LEARN MODE, that as user click on it, it will load video with labelled teeth and as the user click on one of the labelled teeth, it will then displayed text box explained on which types of teeth that they have clicked on.

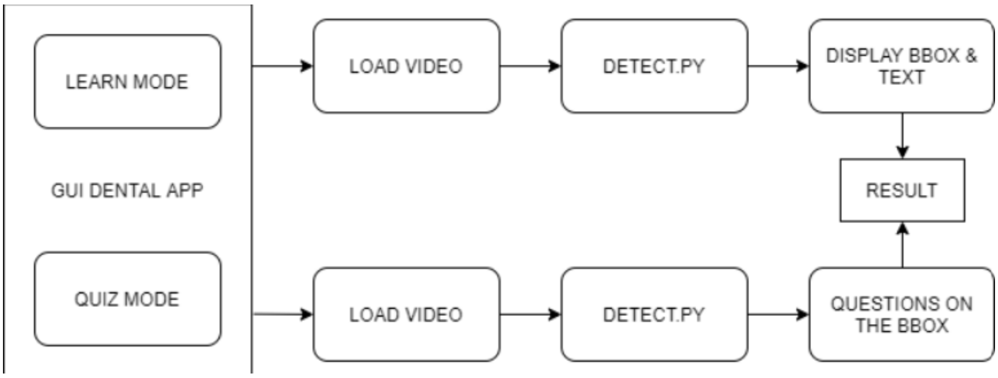


Figure 56. System Architecture

As for the QUIZ MODE, it will also load the same video with labelled teeth but as the user click on one of the teeth, there will be a pop-up text box of question that ask them whether the teeth that they click is the correct one or not.

4.5 System Interfaces

Here is the system interface I designed:

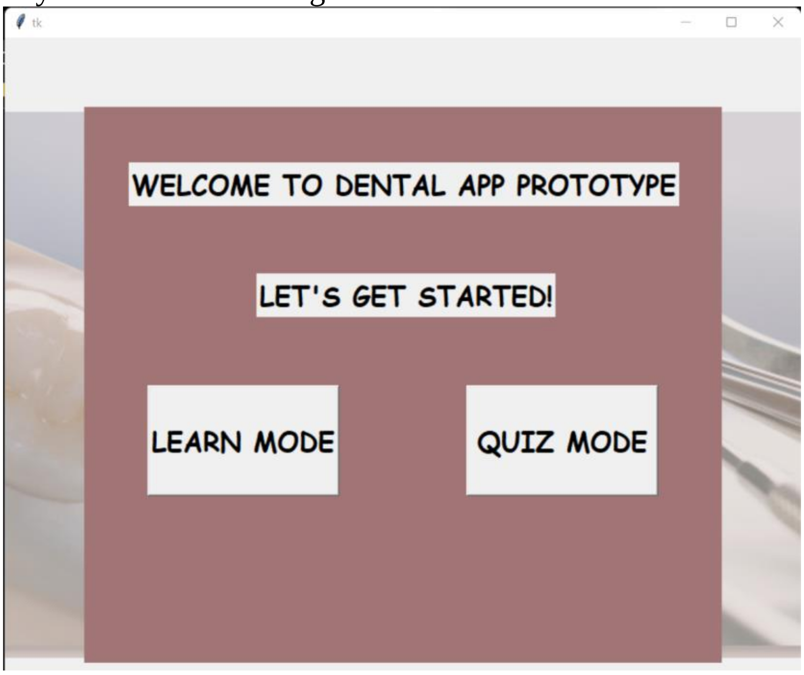


Figure 57. Graphic User Interface (GUI) for Dental App

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Summary

The project's goal is to create a prototype of Deep Learning-based Augmented Reality Application for Dental Students that provides a user-friendly and accessible environment in which students may apply their knowledge while being unable to continue their education face-to-face. As illustrated in FIGURE 56, I had done the research to create a prototype Augmented Reality (AR) application that will allow dentistry students to experience and practice their education.

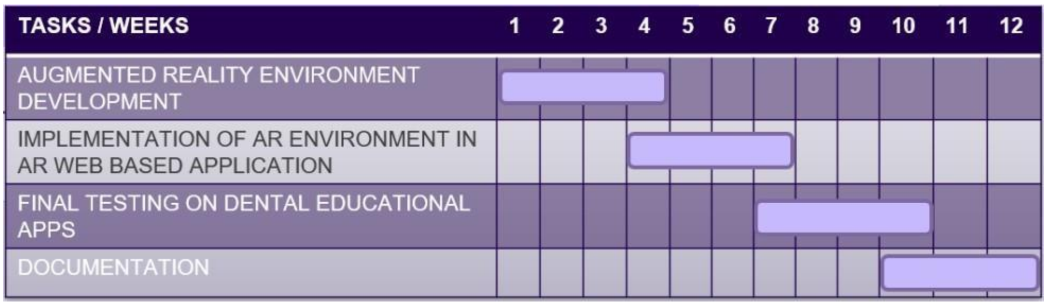


Figure 58. FYP2 Gantt Chart

5.2 Future Works

Because our limitation extends just to the most fundamental markerless AR, we can only achieve what has been described previously. Furthermore, we anticipate that this project will be expanded to include more advanced augmented reality by using ARToolkit or even smart augmented reality (AR) glasses to achieve exceptional performance and a richer experience within its environment. AR smart glasses are computer-capable spectacles that add more information, ideally 3D graphics and information such as animations and films, to the user's real-world sceneries by superimposing computer-generated or digital material on the user's real environment. We hope to accomplish the finest experience possible for dentistry students to use technology to continue their studies and knowledge wherever they are soon.

REFERENCES

Abdul Fatah, F., Ahmad Tarmidzi, N. A., Mat-Baharin, N. H., Md Zahid, D., Mohamad Hanafiah, R., Mohamed, N., Wan Ali, W. N. S., & Ibrahim, Z. (2021). Impact of COVID-19 Pandemic on Dental Education: Experience Sharing from a Malaysian Public University. *Ulum Islamiyyah*,166–178. https://doi.org/10.33102/uij.vol33no1.297

Amir, L. R., Tanti, I., Maharani, D. A., Wimardhani, Y. S., Julia, V., Sulijaya, B., & Puspitawati, R. (2020). Student perspective of classroom and distance learning during COVID-19 pandemic in the undergraduate dental study program Universitas Indonesia. *BMC Medical Education*, *20*(1). https://doi.org/10.1186/s12909-020-02312-0

Computació, I. S. D. D. C. I. S. D. D. V. D. P. U. Y.-I. (2020, June 25). *A Mobile Augmented Reality System for the Learning of Dental Morphology*. Riunet. https://riunet.upv.es/handle/10251/133556

Chen, C. W., Chen, W. Z., Peng, J. W., Cheng, B. X., Pan, T. Y., Kuo, H. C., & Hu, M. C. (2017). A Real-Time Markerless Augmented Reality Framework Based on SLAM Technique. *2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*. Published. https://doi.org/10.1109/ispan-fcst-iscc.2017.87

Dental Training AR App by AugmentedReality.health. (n.d.). Augmented Reality. Retrieved June 23, 2021, from https://augmentedreality.health/dental-training.cfm

Gandhi, R. (2018, December 3). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. Medium. https://towardsdatascience.com/r-cnn-fast-r-cnn- faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Practice, M. (2018, October 3). *The Future of Dentistry with Augmented and Virtual Reality*. Net32 Inc. https://blog.net32.com/the-future-of-dentistry-with-augmented- and-virtual-reality/

Real Python. (2021, February 27). *Jupyter Notebook: An Introduction*. Realpython.Com. Retrieved November 23, 2021, from https://realpython.com/jupyter-notebook-introduction/

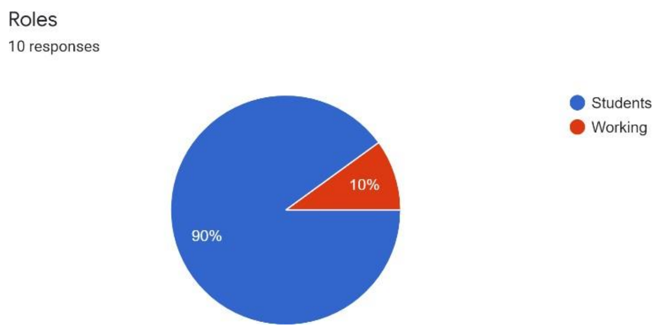
Salvatori, C. (2019, October 1). *What is Augmented Reality?*

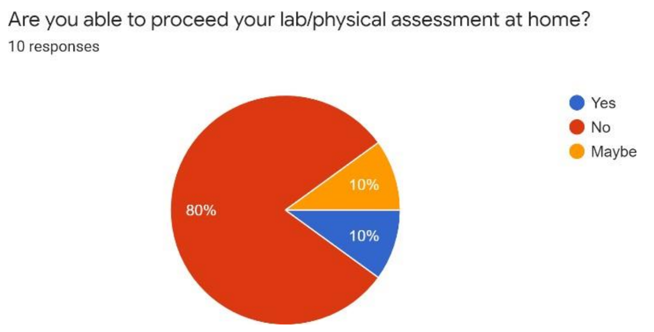
Brochesia. https://www.brochesia.com/what-is-augmented-reality/

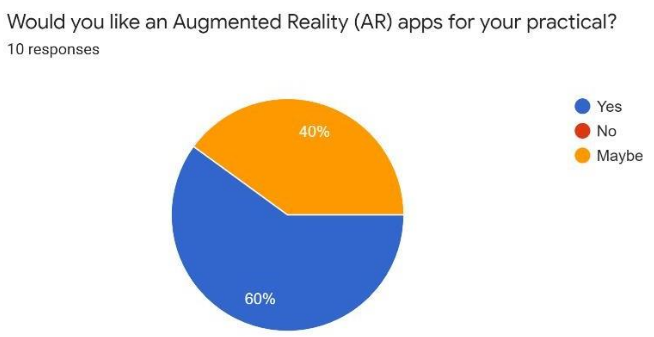
T. (2019b, October 15). *AUGMENTED REALITY BECOMES MOBILE AUGMENTED REALITY*. Brochesia. https://www.brochesia.com/augmented-reality- becomes-mobile-augmented-reality/

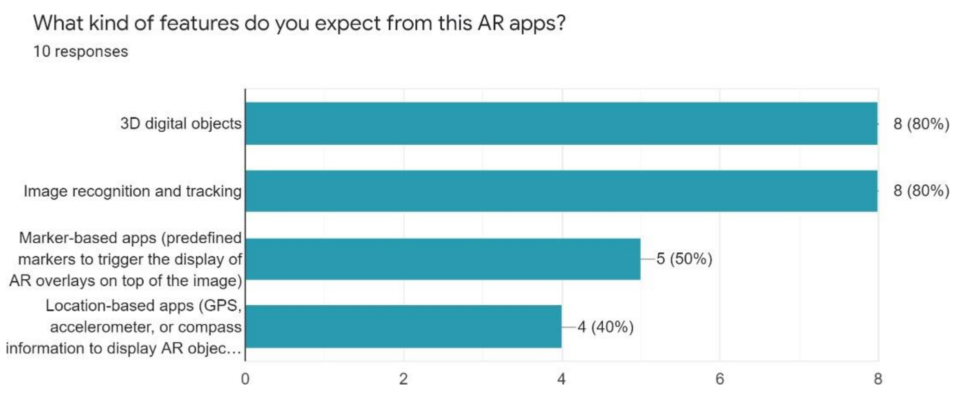
APPENDICES

APPENDIX A


Roles
10 responses
Students
Working
10%
90%


Are you able to proceed your lab/physical assessment at home?
10 responses
Yes
No
Maybe
10%
80%
10%


Would you like an Augmented Reality (AR) apps for your practical?
10 responses
Yes
No
Maybe
40%
60%


What kind of features do you expect from this AR apps?
10 responses
3D digital objects
8 (80%)
Image recognition and tracking
8 (80%)
Marker-based apps (predefined markers to trigger the display of AR overlays on top of the image)
5 (50%)
Location-based apps (GPS, accelerometer, or compass information to display AR objec…
4 (40%)
0
2
4
6
8