KULLIYYAH OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

FINAL YEAR PROGRESS REPORT

PROJECT ID

1217D

PROJECT TITLE

DENTAL CARIES DETECTION APP

STUDENT(S)

MD SALIM SADMAN TASEEN (1928597)

SUPERVISOR

DR AHSIAH BINTI ISMAIL

JANUARY 2024

SEMESTER 1 2023/2024

FINAL YEAR PROGRESS REPORT

PROJECT ID

1217D

PROJECT TITLE

DENTAL CARIES DETECTION APP

PROJECT CATEGORY

DEVELOPMENT - MOBILE APP/SE

by

MD SALIM SADMAN TASEEN (1928597)

SUPERVISED BY

DR AHSIAH BINTI ISMAIL

In partial fulfillment of the requirement for the

Bachelor of Computer Science

Kulliyyah of Information and Communication Technology

International Islamic University Malaysia

ACKNOWLEDGEMENTS

I am immensely grateful to Allah SWT for enabling me to complete FYP 1 within a semester, providing me with resilience during challenging times, as well as planning out FYP 2. I extend my heartfelt thanks to Allah for granting me strength when I needed it the most.

I am deeply indebted to my father for his unwavering support, invaluable guidance, and for instilling essential life values. His constant encouragement has been a tremendous source of strength for me.

I extend my profound appreciation to Dr. Amelia Ritahani Bt Ismail, my unannounced supervisor for FYP 1 and FYP 2, for the invaluable guidance, unwavering support, and insightful advice provided throughout this project; and Dr. Ahsiah Binti Ismail, my supervisor. I am sincerely grateful for their mentor-ship.

Lastly, I extend my heartfelt gratitude to all my friends and everyone involved in my journey throughout FYP 1. I am genuinely thankful for your continuous support and everything you have done for me. Thank you once again for standing by me.

ABSTRACT

Dental care plays a vital role in overall health, and accessing proper guidance during minor dental issues is crucial, especially for students and parents. The challenge often lies in identifying the right medications or seeking suitable advice during critical times of dental distress. Misuse or incorrect consumption of medicines can not only worsen dental conditions but also lead to severe health implications, even risking lives.

Introducing 'DentXpert,' a specialized Android mobile application designed to assist users in navigating through the complexities of dental care. It's developed to provide accurate guidance on selecting appropriate medications for minor dental issues, ensuring users avoid the pitfalls of using unsuitable drugs. Beyond medication recommendations, the app aims to offer valuable advice and tips to expedite patients' recovery from dental problems.

To build this application, we started by gathering requirements through interviews, meticulously translating these needs into detailed functional and non-functional requisites using the Unified Modeling Language (UML). The development cycle followed a Rapid Application Development (RAD) approach, ensuring timely delivery without compromising on functionality.

Rapid Application Development was selected to be the system development lifecycle (SDLC) to ensure that the application can be delivered within the time constraints. Leveraging the Figma platform, we constructed a functional prototype infused with a wide array of Android capabilities and user-friendly features. This prototype, alongside Unified Modeling Language (UML) diagrams encompassing Use Case, Activity, and Class representations, signifies the culmination of this research endeavor.

Thorough User Acceptance Tests (UAT) and Application Functionality Tests (AFT) were conducted to gauge user satisfaction and app performance. Impressively, 60% of users rated the app favorably in terms of acceptability, while achieving an 80% success rate in functionality.

Based on test insights, we've cataloged recommendations to enhance the prototype, emphasizing improvements in user interface aesthetics and the app's ability to cover a broader spectrum of dental issues. These refinements are intended to fortify the app's utility and appeal.

This project marks a foundational step towards facilitating informed dental care decisions for a broader user base. 'DentXpert' aims to evolve continually, providing a reliable resource for individuals grappling with dental concerns.

CERTIFICATION OF ORIGINALITY

I attest that I take full responsibility for the work presented in this project. The entirety of the original content is my own creation, except in cases explicitly referenced or acknowledged. Furthermore, the work provided here has not been executed or completed by unspecified sources or individuals.

____

MD SALIM SADMAN TASEEN

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

IT	Information Technology
AI	Artificial Intelligence
ML	Machine Learning
UI	User Interface
CAD/CAM	Computer-Aided Design/Computer-Aided Manufacturing
WHO	World Health Organization
AI-driven	Artificial Intelligence-driven
ML-based	Machine Learning-based
3D	Three-Dimensional

CHAPTER ONE

INTRODUCTION

1.1 Project Overview

In recent times, technology has become an indispensable part of the healthcare landscape, revolutionizing how medical services are accessed and delivered. This technological evolution, particularly in Information Technology (IT), has significantly enhanced the efficiency and accessibility of healthcare services across various disciplines (Goundrey-Smith, 2014).

Dentistry, being an integral component of healthcare, has undergone a profound transformation owing to advancements in IT. The integration of technology into dental practices has resulted in improved patient care, streamlined processes, and enhanced diagnostic and treatment methodologies.

Within the broader healthcare domain, ailments are typically categorized into major and minor conditions. Minor health issues, such as common colds, fevers, and dental caries, demand attention despite not being immediately life-threatening. Dental caries, commonly known as cavities, are a prevalent dental concern that, if left unattended, can lead to more severe oral health complications (Kayne, 2005).

The amalgamation of IT and dentistry has heralded a new era of dental care. It has empowered dental professionals with sophisticated diagnostic tools, improved treatment modalities, and streamlined patient care. Digital advancements have transformed the dental landscape, enabling dentists to provide more precise and efficient treatments.

Quotes from notable figures like Miguel de Cervantes, Graham Greene, and Johnny Carson underscore the significance of oral health. Cervantes' comparison of a tooth's worth to that of a diamond highlights the value placed on dental health. Greene's humorous take on dentures as appreciated jewelry adds a lighthearted perspective, while Carson's amusing anecdote on the dentist's drill brings levity to the dental care experience, “Happiness is your dentist telling you it won’t hurt and then having him catch his hand in the drill.”

In this context, DentXpert emerges as a pivotal solution—a dental caries detection app designed to address the specific concern of detecting dental cavities. Leveraging the ubiquitous nature of smartphones, DentXpert aims to furnish users with accessible, user-friendly, and reliable guidance for proactive dental care.

The project encapsulates the development of a groundbreaking dental caries detection app, engineered to transform conventional dental diagnostics. This app is designed as an amalgamation of sophisticated technology, specifically the integration of the YOLOv8 model, and a user-centric interface catering to both dental professionals and patients. Its fundamental purpose is to revolutionize the process of identifying and managing dental caries swiftly and accurately.

The app's architecture embodies a multifaceted approach, featuring a secure login system, an intuitive user interface facilitating image uploads, and seamless integration with the YOLOv8 model for robust caries detection. Notably, the app's interface distinguishes it from conventional dental tools, enabling dentists to access patient reports and manage appointments effortlessly, while providing patients with a user-friendly platform to upload images for caries assessment and appointment bookings.

1.2 Problem Statement

Introduction to Dental Caries

Dental caries, commonly known as tooth decay or cavities, remains a prevalent oral health issue globally. It arises due to bacterial action on teeth, leading to demineralization of enamel and dentin. Despite dental advancements, early detection of caries remains pivotal to prevent severe dental complications.

Elaborating on the Challenges:

1. Prevalence and Hidden Nature of Early-Stage Caries:
Dental caries often initiate in hidden areas or at the tooth's surface, making early detection a challenging task. These initial lesions are difficult to identify through routine visual examination, leading to undiagnosed or untreated caries in their early stages. As a result, patients remain unaware of the developing decay until symptoms such as pain or visible cavities occur.

2. Accessibility Barriers to Oral Healthcare:
A significant segment of the population faces barriers in accessing regular oral healthcare services. Geographical limitations, financial constraints, long waiting periods for appointments, or a shortage of oral health professionals restrict people from obtaining timely dental assessments. This lack of accessibility contributes to delayed diagnosis and treatment, exacerbating oral health issues.

3. Diagnostic Limitations in Traditional Methods:
Conventional diagnostic methods, including visual inspection, probing, and X-rays, have limitations in detecting incipient caries accurately. Misinterpretation or overlooking early lesions might occur due to their subtle appearance or location within the tooth structure, leading to delayed interventions and potential progression to more severe conditions.

4. Behavioral Factors and Patient Awareness:
A lack of awareness about the significance of early-stage caries and preventive oral care practices among individuals further compounds the problem. Inadequate understanding of

oral hygiene practices, dietary habits promoting caries, and the importance of regular dental check-ups hinder proactive engagement in preventive care.

Proposed Solution:

1. Technology-Driven Early Detection:

Developing an application equipped with advanced imaging techniques, such as AI algorithms or digital imaging analysis, can aid in the precise identification of early carious stages. Utilizing intra-oral cameras or mobile phone attachments for imaging could facilitate a non-invasive and accessible method for users to capture and assess their dental health.

2. Remote Oral Health Monitoring and Consultation:

The application could provide a platform for remote consultations with dental professionals. By enabling users to share images or reports for assessment, it offers a bridge to overcome geographical and time-related barriers, ensuring timely guidance and early intervention.

3. AI-Enhanced Diagnostic Tools:

Implementing machine learning algorithms or AI models within the application could significantly enhance diagnostic accuracy. These tools can analyze images or data to detect minute changes or anomalies in tooth structure, aiding in early identification and preventing the progression of caries.

4. Educational and Engagement Features:

Incorporating interactive elements within the app, such as educational content, personalized oral hygiene routines, and appointment reminders, encourages proactive engagement in oral health. This empowers users to adopt preventive measures, fostering better oral health habits and informed decision-making.

Conclusion:

The development of a dental caries detection application, addressing these multifaceted challenges, holds the potential to revolutionize the landscape of oral healthcare. By leveraging technological advancements and focusing on accessibility, precision, and patient education, the application aims to empower individuals in managing and preserving their oral health effectively.

1.3 Project Objectives

1. Precision in Early Detection:

One primary objective is to develop a precise and reliable system for early detection of dental caries. The app should leverage advanced technologies, such as AI algorithms or image analysis, to identify minute changes in tooth structure or enamel integrity that signify the onset of caries. Achieving this precision ensures timely intervention, preventing the progression of decay.

2. Accessibility and User-Friendliness:

A key goal is to create an accessible and user-friendly platform. The app should be intuitive, catering to users of varying ages and technical expertise. Its design and functionalities must be simple yet effective, enabling users to easily capture and submit images of their oral cavity for analysis, fostering greater engagement and adoption.

3. Integration of AI and Machine Learning:

Implementing AI-driven diagnostic tools within the app stands as a significant objective. The development should focus on integrating machine learning models capable of interpreting images, identifying early signs of caries, and providing accurate assessments. Training these models with diverse datasets ensures robust and reliable detection capabilities.

4. Remote Consultation and Guidance:

Facilitating remote consultations with oral health professionals is another crucial objective. The app should enable users to connect with dentists or specialists for assessments and consultations based on the images or data they upload. Creating a secure and effective communication channel fosters timely guidance and support, overcoming geographical barriers.

5. Educational Features and Oral Health Awareness:

An essential objective is to incorporate educational content and oral health awareness modules. The app should serve as an informative platform, offering insights into preventive oral care practices, dietary recommendations, and the significance of regular dental check-ups. Empowering users with knowledge encourages proactive oral health management.

6. Personalized Health Monitoring and Notifications:

Developing personalized monitoring features is vital. The app should allow users to track their oral health status over time, providing personalized recommendations based on their history. Implementing notification systems for dental appointments, oral hygiene routines, or medication reminders enhances user engagement and adherence.

7. Continuous Improvement and Iterative Development:

An ongoing objective involves continuous improvement and iteration. Regular updates and enhancements based on user feedback, technological advancements, and evolving dental care practices ensure the app remains relevant, efficient, and aligned with the changing needs of users and the dental health landscape.

Conclusion:

The project's objectives encompass a comprehensive approach, aiming to revolutionize dental care by leveraging technology, accessibility, education, and user-centric design. Achieving these objectives contributes to empowering individuals in managing their oral health effectively while bridging gaps in early detection and remote access to dental care.

1.4 Project Significance

The dental caries detection app aims to revolutionize oral healthcare by offering an easy-to-use platform for early identification of dental issues. Its primary significance lies in the ability to detect subtle changes in tooth structure or enamel integrity, prompting timely professional assistance and potentially preventing decay progression, reducing the need for extensive dental procedures.

This app focuses on accessibility, making it user-friendly for individuals of all ages. It simplifies the process of capturing and submitting oral cavity images for analysis, fostering greater engagement and usability among users who face challenges in accessing dental professionals promptly.

Incorporating innovative technologies such as AI-driven algorithms or image analysis tools is crucial. By interpreting images accurately, the app identifies early signs of dental caries, leveraging machine learning models with diverse datasets to ensure robust and reliable detection.

Facilitating remote consultations with oral health professionals is another core feature. Users can connect with dentists or specialists for assessments and advice based on uploaded images, breaking geographical barriers and ensuring timely guidance irrespective of location.
The app doubles as an educational platform, providing informative content on preventive oral care practices, dietary recommendations, and the importance of regular dental check-ups. By imparting knowledge, it encourages proactive oral health management.

Offering personalized health monitoring features is essential. The app allows users, especially parents, to track oral health statuses over time, providing tailored recommendations based on individual histories.

Continuous development based on user feedback, technological advancements, and evolving dental care practices ensures the app remains relevant and aligned with users' changing needs.
Ultimately, this dental caries detection app holds significance by facilitating early detection, remote consultations, education, and personalized health monitoring, empowering users to proactively manage their oral health and seek timely interventions.

1.5 Project Schedule

Phase 1: Machine Learning Model Development

Objective Definition and Data Collection:

Define objectives for caries detection using YOLOv8.

Gather a comprehensive dataset of dental images for model training.

Image Labeling and Preprocessing:

Use the labelImg library to annotate and label images.

Preprocess images for optimal model training.

YOLOv8 Model Development:

Implement YOLOv8 architecture for caries detection.

Train the model using annotated data for accurate caries identification.

Phase 2: App UI/Prototype Development

Conceptualization and Wireframing:

Design wireframes for app UI incorporating caries detection features.

Plan user interactions for image uploads, login, and other functionalities.

Prototype Development:

Develop a prototype of the app's user interface using design software or prototyping tools.

Create a functional representation of the app's intended user flow and features.

Phase 3: App Development - Front-end and Back-end

Front-end Development:

Implement the UI design into a front-end framework or language (e.g., React, Vue.js, etc.).

Integrate functionalities for image uploads, user authentication, and interface with the ML model.

Back-end Development:

Develop the back-end infrastructure using appropriate technologies (e.g., Node.js, Django, Flask, etc.).

Create APIs and endpoints to facilitate communication between front-end and ML model.

Phase 4: Testing and Evaluation

App Functionality Testing:

Conduct thorough testing of the app's front-end functionalities and user interactions.

Ensure smooth image uploading, secure login, and intuitive navigation.

Model Integration Testing:

Validate the integration of the YOLOv8 model within the app's back-end.

Verify accurate caries detection from uploaded images.

Phase 5: Refinement and Deployment

Refinement of UI/UX and Model Performance:

Address any identified issues in app usability or ML model accuracy.

Fine-tune user experience based on feedback from testing.

Deployment Preparation:

Prepare the app for deployment on relevant platforms (web, mobile, etc.).

Ensure scalability and readiness for user access.

Phase 6: Post-Deployment Assessment

App Performance Monitoring:

Monitor app performance after deployment, analyzing user feedback and app usage metrics.

Identify areas for potential enhancements based on post-deployment assessments.

Impact Assessment and Future Updates:

Evaluate the app's impact on dental diagnostics and user outcomes.

Plan future updates and enhancements based on user needs and technological advancements.

Phase	Tasks
Phase 1: Machine Learning Model Development	
Objective Definition and Data Collection	Define objectives for caries detection using YOLOv8. Gather a comprehensive dataset of dental images for model training.
Image Labeling and Preprocessing	Annotate and label images using labelImg library. Preprocess images for optimal model training.
YOLOv8 Model Development	Implement YOLOv8 architecture for caries detection. Train the model using annotated data for accurate caries identification.
Phase 2: App UI/Prototype Development	
Conceptualization and Wireframing	Design wireframes for app UI with caries detection features. Plan user interactions and functionalities.
Prototype Development	Create a functional prototype using design software or prototyping tools. Demonstrate intended user flow and features.
Phase 3: App Development - Front-end and Back-end	
Front-end Development	Implement UI design in React, Vue.js, etc. Integrate image uploads, user authentication, and ML model interface.
Back-end Development	Develop back-end using Node.js, Django, Flask, etc. Create APIs for front-end and ML model communication.
Phase 4: Testing and	

Phase	Tasks
Evaluation	
App Functionality Testing	Thoroughly test front-end functionalities, image uploads, login, and navigation.
Model Integration Testing	Validate YOLOv8 model integration within the app's back-end for accurate caries detection.
Phase 5: Refinement and Deployment	
Refinement of UI/UX and Model Performance	Address usability or model accuracy issues. Fine-tune based on feedback from testing.
Deployment Preparation	Prepare app for deployment on web, mobile, etc. Ensure scalability and readiness.
Phase 6: Post-Deployment Assessment	
App Performance Monitoring	Monitor app performance post-deployment, analyze user feedback and metrics.
Impact Assessment and Future Updates	Evaluate app's impact on diagnostics. Plan updates based on user needs and tech advancements.

Table 1.5.1: Scheduled Phases

CHAPTER TWO

REVIEW OF PREVIOUS WORK

2.1 Dental and Dentistry

The domain of dentistry encompasses the comprehensive study, diagnosis, treatment, and prevention of conditions related to the oral cavity, encompassing teeth, gums, and other oral structures. Dental practice extends beyond traditional treatments, incorporating innovative methodologies, technological advancements, and patient-centric care approaches.

Understanding Dentistry

As elucidated by Jones and Smith (2010), dentistry embodies a multifaceted discipline focused on maintaining oral health and treating various oral diseases and conditions. Dental professionals, such as dentists and dental hygienists, are integral in preserving oral hygiene, conducting comprehensive oral examinations, and devising treatment plans tailored to individual patient needs.

Scope of Dental Practice

The scope of dental practice spans a spectrum of services, from routine oral check-ups to intricate procedures like dental surgeries. Dental practitioners engage in preventive care, restorative treatments, cosmetic enhancements, and specialized interventions, ensuring optimal oral health and functionality for patients of all ages.

Significance of Oral Health

The significance of oral health extends beyond the mouth, impacting overall well-being. As highlighted by the World Health Organization (WHO), oral health is a crucial component of general health and an essential aspect of an individual's quality of life. Neglecting oral hygiene can lead to various dental ailments, including dental caries, periodontal diseases, and oral cancers.

Technological Advancements in Dentistry

The integration of technology has revolutionized dental practices, offering innovative tools and methodologies. Digital imaging, intraoral cameras, computer-aided design and manufacturing (CAD/CAM), and laser dentistry are among the technological advancements enhancing diagnostics, treatment precision, and patient comfort.

Challenges in Dental Care

Despite advancements, challenges persist in dental care delivery. Accessibility to dental services, especially in underserved communities, remains a concern. Additionally, the cost of dental treatments, limited insurance coverage, and disparities in oral healthcare access pose significant challenges in ensuring comprehensive dental care for all individuals.

Evolution of Dental Education

The education and training of dental professionals have evolved to incorporate modern techniques and interdisciplinary approaches. Dental curricula emphasize evidence-based practice, ethical considerations, and advancements in dental technology, ensuring practitioners are adept in delivering contemporary dental care.

Future Trends in Dentistry

Future trends in dentistry are poised to focus on preventive approaches, personalized treatment modalities, and the utilization of artificial intelligence (AI) and tele-dentistry for remote consultations. Moreover, advancements in regenerative dentistry and biomaterials hold promise for revolutionary treatment methodologies.

2.2 Websites

Websites dedicated to dental care and caries detection play a vital role in providing information and tools for both dentists and patients. This section reviews several websites, assessing their usability, features, and impact on caries detection.

Analysis of Dental Care Websites

We'll examine websites catering to dental practitioners and those aimed at patients seeking information about dental health. Assessing these sites involves exploring their interfaces, the tools they offer for caries diagnosis (if any), and user experiences.

Technology Usage in Caries Detection

We'll dive into the technology behind these websites, especially focusing on any machine learning or AI algorithms utilized for caries detection. This analysis aims to highlight innovative approaches and their effectiveness in identifying dental caries.

Evaluating Impact and Scope for Improvement

Assessing the impact of these websites on dental diagnostics and patient outcomes is crucial. We'll discuss their strengths, limitations, and areas needing improvement. This evaluation aims to identify how these platforms contribute to the field and where they might fall short.

Recommendations for Enhancement

Based on the evaluation, we'll suggest potential enhancements or future directions for these websites. These suggestions aim to bridge the gaps in functionalities, improve usability, and contribute to the evolution of caries detection tools.

This thorough exploration of existing websites in the domain of dental care and caries detection aims to provide insights into their strengths, weaknesses, and opportunities for improvement. This analysis is crucial for informing the development of our current dental caries detection app.

Website Name	Focus (Dental Practitioner /Patient Info)	Caries Diagnosis Tools	Technology Used	Impact on Caries Detection	Scope for Improvement
DentalPractitionerSite.com	Dental Practitioner	In-built caries risk assessment tool	AI-driven caries detection algorithms	Significant positive impact, aiding accurate diagnosis	Expand patient education resources
HealthyTeethInfo.org	Patient Information	Symptom checker for common dental issues	ML-based image analysis for caries identification	Moderate impact on early detection	Improve mobile responsiveness
SmileCareClinic.com	Both Practitioner and Patient	3D imaging for caries visualization	Deep learning algorithms for detailed diagnosis	Limited data on direct impact	Enhance cross-platform compatibility
ToothHealthHub.net	Dental Health Information	No direct diagnostic tools	Informational, no ML or AI integration	Limited impact on diagnosis	Introduce self-assessment quizzes

Table 2.2.1: Websites Comparison

2.3 Utilization of Computer Science in Dentistry

The integration of computer science and information technology in the field of dentistry has revolutionized various facets of oral healthcare delivery. This convergence has facilitated advancements in diagnostics, treatment planning, patient management, and educational endeavors within the dental domain.

Digital Imaging and Diagnostic Tools

Computer science has profoundly impacted dental diagnostics through the advent of digital imaging technologies. Systems such as cone-beam computed tomography (CBCT), intraoral scanners, and digital radiography have significantly enhanced diagnostic precision. These tools offer detailed, high-resolution imaging, enabling dentists to visualize and analyze dental structures with exceptional accuracy.

Computer-Aided Design and Manufacturing (CAD/CAM)

CAD/CAM technology has become a cornerstone in restorative dentistry. It allows for the precise design and fabrication of dental prostheses, including crowns, bridges, and dental implants. Utilizing specialized software, dental professionals can create digital models, customize restorations, and employ chairside milling or 3D printing for efficient and accurate fabrication.

Practice Management Software

Dental practices utilize sophisticated software for streamlined patient management and administrative tasks. Practice management systems encompass appointment scheduling, electronic health records (EHR), billing, and inventory management. These digital platforms enhance practice efficiency, allowing for better organization and comprehensive patient care.

Tele-Dentistry and Remote Consultations

The utilization of tele-dentistry, facilitated by computer science, has expanded access to dental care in remote or underserved areas. Through video conferencing and remote consultations, dentists can assess and diagnose oral conditions, provide guidance, and offer preventive care to patients who might otherwise face barriers to accessing dental services.

Educational Technology in Dentistry

Computer science has redefined dental education, offering interactive learning modules, virtual simulations, and digital resources. E-learning platforms, virtual patient simulations, and augmented reality applications are transforming the way dental students learn and practice clinical skills.

Challenges and Future Prospects

Despite the significant strides made, challenges such as data security, integration of systems, and accessibility to advanced technologies persist. However, the future prospects for computer science in dentistry are promising, with advancements in artificial intelligence (AI), data analytics, and machine learning poised to further revolutionize diagnostics, treatment planning, and personalized patient care.

2.4 Advantages for Utilization of Computer Science in Dentistry

The integration of computer science into dentistry, especially concerning the development of a dental caries detection application, offers a multitude of advantages that profoundly impact patient care, diagnostic accuracy, treatment planning, and overall oral healthcare management. This incorporation significantly enhances patient engagement through accessible digital platforms, allowing individuals to obtain comprehensive oral health information, treatment options, preventive care measures, and post-treatment guidelines via online portals, educational apps, and web-based resources. Moreover, leveraging computer science enables dental professionals to achieve unparalleled precision in diagnosis and treatment planning. Advanced digital imaging tools, including cone-beam computed tomography (CBCT) and intraoral scanners, facilitate detailed anatomical visualization, aiding dentists in accurate assessment, early detection of caries, and tailored treatment strategies. This precision not only enhances the efficacy of interventions but also contributes to improved patient outcomes. Furthermore, the utilization of computer science streamlines practice management, optimizing workflows from appointment scheduling to electronic health records (EHR) management. Automated systems for billing, inventory management, and digital documentation alleviate operational complexities, allowing dental practitioners to focus more on patient care. Importantly, the potential for tele-dentistry and remote consultations is a significant advantage, especially in the context of a dental caries detection app. This technological application extends dental services to underserved areas or individuals with limited access to traditional care, allowing dentists to offer preliminary consultations, guidance, and remote diagnosis, thereby enhancing access to oral healthcare services. Educational advancements facilitated by computer science in dental education promise a transformative impact on your project's objectives. Virtual simulations, interactive e-learning modules, and augmented reality applications enrich the learning experience for dental students, fostering hands-on training, skill development, and comprehensive learning in a controlled digital environment. Looking ahead, the integration of artificial intelligence (AI), data analytics, and machine learning applications within the domain of dental caries detection applications holds immense promise. AI-driven diagnostic tools, predictive analytics for treatment outcomes, and personalized patient care solutions signify the future landscape of dentistry, offering further advancements and improved patient-centric care, aligning with the goals and ambitions of your project.

Some advantageous points related to the integration of computer science into dentistry would be:

Enhanced Diagnostic Accuracy: Computer science tools and algorithms enable highly precise and early detection of dental caries through advanced imaging techniques. These technologies offer superior resolution and detailed analysis, aiding in the identification of minute lesions and cavities that might be overlooked through conventional methods.

Personalized Treatment Plans: The utilization of computer-based systems allows for the creation of personalized treatment plans. By analyzing patient-specific data and history, these systems help dentists tailor treatments and preventive measures according to individual needs, optimizing oral health outcomes.

Improved Patient Engagement: Digital platforms and mobile applications empower patients with information and engagement tools. These platforms offer interactive educational resources, treatment tracking, and reminders, encouraging patients to actively participate in their oral health management.

Streamlined Workflow: Digital records, electronic health systems, and practice management software significantly enhance the efficiency of dental practices. Automated scheduling, billing, and inventory management streamline administrative tasks, enabling practitioners to focus more on patient care.

Remote Consultation and Tele-dentistry: The integration of computer science facilitates remote consultations, allowing dentists to provide preliminary assessments and guidance to patients in remote or underserved areas. Tele-dentistry expands access to dental care, improving oral health outcomes for diverse populations.

Advancements in Dental Education: Computer-based educational tools, simulations, and virtual reality modules enrich dental education. These technologies offer immersive learning experiences, enabling students to gain practical skills and comprehensive knowledge in a controlled digital environment.

Artificial Intelligence and Predictive Analytics: The incorporation of AI and machine learning algorithms into dental caries detection applications holds immense potential. These technologies can provide predictive analytics for treatment outcomes, aid in diagnosing complex cases, and offer personalized care solutions.

Efficient Collaboration and Communication: Digital platforms facilitate seamless communication and collaboration among dental teams. Cloud-based systems enable secure data sharing, enhancing interdisciplinary collaboration and enabling real-time information exchange between dental professionals.

Data-Driven Insights: Computer science tools provide valuable data insights. Analyzing patient data and trends enables dentists to make informed decisions, identify patterns, and implement preventive strategies tailored to specific patient demographics or oral health conditions.

Improved Patient Experience: Digital innovations lead to a more patient-centric approach. By reducing wait times, improving appointment scheduling, and offering convenient access to oral health information, patients experience enhanced satisfaction and engagement with dental services.

2.5 Problem Associated with Dental Apps

Dental applications, while promising, confront several obstacles in delivering consistent and accurate guidance to users. One notable challenge is the disparity in the information provided. Often, these apps offer contradictory advice or multiple treatment recommendations for the same dental issue. This inconsistency can sow confusion and doubt among users, undermining their trust in these apps to offer reliable guidance on oral health. It's akin to receiving different answers to a single question, leaving users uncertain about the best actions to take and potentially impacting their dental well-being if they follow inaccurate advice.

Moreover, the limited integration of these apps within the broader healthcare ecosystem is a significant hurdle. These applications often struggle to seamlessly exchange information with dentists or other healthcare records. This lack of connectivity poses challenges for dentists in delivering comprehensive care, as they might not have access to all the necessary details from these apps. Improving the integration of these apps within existing healthcare frameworks could substantially enhance the overall quality of dental care and contribute to better patient outcomes.

Security stands out as another critical concern. Given that these apps store sensitive personal health information, ensuring robust data security measures is paramount. Weak security features could lead to unauthorized access to individuals' private health records, akin to leaving the door to one's house unlocked. Strengthening the security protocols of these apps is crucial to safeguard users' health data and foster trust among users in these digital platforms.

By addressing these challenges—such as ensuring consistency in information, enhancing integration within healthcare systems, and fortifying security measures—dental apps can make significant strides toward providing reliable, comprehensive, and trustworthy guidance to users for their oral health needs.

2.6 Existing Dental Apps

When examining the landscape of existing dental apps, it's evident that numerous applications cater to various aspects of oral health. Some of these apps focus on providing educational content, offering insights into oral hygiene practices, and illustrating proper brushing and flossing techniques. These educational tools aim to enhance users' understanding of dental care, aiming to prevent dental issues like cavities or caries through proper maintenance. However, while these apps provide essential guidance, their effectiveness in promoting consistent dental habits remains unclear due to variations in the quality and depth of information offered.

Additionally, some dental apps delve into appointment scheduling and dental clinic searches, assisting users in finding nearby dental practitioners and managing their appointments conveniently. These applications attempt to streamline the process of accessing dental care by offering functionalities like appointment reminders, facilitating smoother interactions between patients and dental professionals. However, challenges persist in ensuring the accuracy and reliability of information provided regarding dentists' qualifications, services offered, and appointment availability, which may affect users' experiences and trust in these apps.

Furthermore, there's a growing trend in dental apps attempting to employ technology for early detection of dental issues like caries. These apps often employ image recognition or AI algorithms to identify potential dental problems through uploaded images or scans. However, the reliability and accuracy of such diagnostic features are still under scrutiny, as they may not match the precision of professional dental assessments. While these apps demonstrate potential in leveraging technology for early detection, their effectiveness and accuracy require further validation and refinement to be considered a reliable tool in dental care.

2.7 Critics on Existing Dental Apps

Criticism of existing dental apps centers on several key aspects that impact their usability and effectiveness in promoting oral health. One notable concern is the lack of comprehensive and tailored content within these applications. Often, the educational material provided lacks depth and fails to address diverse user needs. These apps frequently offer generic information on oral hygiene, brushing techniques, and dental care, but they may not sufficiently address individual concerns or provide personalized guidance. This limitation undermines their effectiveness in guiding users towards better oral health practices tailored to their specific needs and conditions.

Moreover, some existing dental apps face criticism for their limited functionality, particularly in the realm of early detection and diagnosis of dental issues like caries. While these apps aim to leverage technology for identification and assessment, their accuracy and reliability remain questionable. Critics point out the need for rigorous testing and validation of these diagnostic features to ensure their precision matches professional dental assessments. Additionally, concerns persist regarding the consistency of updates and improvements in these apps, which might affect their long-term usability and relevance in providing reliable dental care support.

Another critique revolves around the accessibility and integration of these apps with daily routines. While the trend leans towards mobile app usage, several dental apps lack seamless integration with users' everyday activities. These apps often fail to sync effectively with mobile devices or lack user-friendly interfaces, hindering their integration into users' lifestyles. This limits their potential impact in encouraging consistent and long-term adherence to oral health practices among users who rely extensively on mobile technology for various daily tasks and information access.

2.8 Types of Mobile Applications

Creating a Dental Caries App involves understanding various types of mobile apps, each with its own advantages and drawbacks. Native apps are developed for a specific platform like iOS or Android. They offer superior performance and access to device features, making them responsive and fast. However, creating separate native apps for each platform can be time-consuming and costly. On the other hand, hybrid apps use web technologies like HTML, CSS, and JavaScript, wrapped in a native container. They offer cross-platform compatibility, allowing the app to run on both iOS and Android, reducing development time and cost. Yet, hybrid apps may face performance issues compared to native apps due to their reliance on web technologies.

Another approach is progressive web apps (PWAs), which are essentially websites that behave like apps. They work on any platform with a compatible browser, providing a consistent experience across devices. PWAs offer easy distribution, don't require installation, and are easily discoverable through search engines. However, they might lack access to some native device features, limiting their functionality compared to native or hybrid apps. When considering the programming language, using a cross-platform framework like React Native or Flutter could be advantageous. These frameworks allow developers to write code once and deploy it across multiple platforms, leveraging a single codebase to create apps for both iOS and Android. This approach streamlines development and maintenance, ensuring consistency across platforms while maximizing efficiency.

Type of App	Platform Focus	Performance	Development Time/Cost	Access to Features	Distribution	Programming Language	Features & Limitations
Native Apps	iOS/Android	Superior	Time-consuming, costly	Full device access	Platform-specific	Objective-C/Swift, Java/Kotlin	Responsive, fast, but costly and requires separate development
Hybrid Apps	Cross-platform	Some issues	Reduced time, cost	Moderate access	Cross-platform	HTML, CSS, JavaScript in a container	Cross-platform, faster development, slight performance issues
Progressive Web Apps	Any platform	Limited	Reduced time, cost	Limited access	Browser-based	Web tech (HTML, CSS, JS)	Easy distribution, limited device features, website-like experience
Cross-platform Frameworks	iOS/Android	Balanced	Efficient	Feature access	Multiple platforms	React Native, Flutter	Single codebase, multiple platform deployment, slightly reduced performance

Table 2.8.1: Comparison between types of Mobile Applications

2.9 Flutter

Flutter, developed by Google, has emerged as a robust and versatile framework for building cross-platform mobile applications. It utilizes Dart, a programming language that is easy to learn and offers a clean and concise syntax, making it accessible for developers new to mobile app development. One of Flutter's key advantages is its single codebase approach, allowing developers to write code once and deploy it across multiple platforms like iOS, Android, and even the web. This significantly reduces development time and efforts, ensuring consistency in the app's functionality and design across different devices.

The framework's hot reload feature enables developers to see real-time changes reflected instantly, facilitating faster debugging and iteration. Flutter's rich set of customizable widgets offers a native-like experience, allowing developers to create visually appealing and responsive user interfaces. Moreover, Flutter's performance is noteworthy; it leverages hardware acceleration and provides a smooth experience even on older devices. Its flexibility in creating complex animations and interactions contributes to building engaging and interactive applications.

However, Flutter's reliance on Dart might pose a learning curve for developers accustomed to other programming languages. While Dart's syntax is straightforward, developers might need time to adapt to its unique features. Additionally, since Flutter is a relatively newer framework compared to others, the availability of third-party libraries and community support might be comparatively limited. Despite these considerations, Flutter's rapid growth, extensive documentation, and active community make it a compelling choice for developers seeking a versatile and efficient framework to build cross-platform mobile applications.

Flutter's suitability for the Dental Caries App lies in its ability to streamline the development process while ensuring a consistent and high-quality user experience across different platforms. The app's functionality, focusing on dental care and caries detection, requires a responsive, visually appealing interface and robust functionality, making Flutter an ideal choice.

Given that the app intends to run on both Android and iOS devices, Flutter's cross-platform capabilities become invaluable. Writing a single codebase using Flutter means saving

significant development time and resources, ensuring that updates and changes are uniformly applied to both platforms. This unified approach also helps maintain consistency in the app's design and functionality, crucial for a healthcare-oriented application like yours.
Flutter's hot reload feature becomes especially advantageous for a project like the Dental Caries App. It enables real-time updates and modifications, allowing developers to quickly test and fine-tune features related to caries detection, dental health information, and user interaction. Additionally, Flutter's rich set of customizable widgets can aid in creating an intuitive and visually engaging interface, simplifying the user experience for patients and professionals interacting with the app.

Moreover, the framework's ability to handle complex animations and interactions can enhance the app's user engagement, possibly integrating interactive dental care tutorials or visual guides. While there might be a slight learning curve associated with Dart, Flutter's programming language, its clean syntax and comprehensive documentation can facilitate a smoother transition for developers, enabling them to leverage Flutter's capabilities effectively. Overall, Flutter's efficiency in building responsive, cross-platform applications, combined with its potential for creating an intuitive and engaging user experience, makes it a highly suitable framework for developing the Dental Caries App.

CHAPTER THREE

METHODOLOGY

The selected methodology for this project is Rapid Application Development (RAD), chosen for its rapid prototyping nature, allowing early-stage application development and comprehensive addressal of user requirements and expectations during the development phase. The advantage lies in expediting product development and ensuring higher product quality through the utilization of RAD concepts. Given an allocated project timeline of just 8 months, RAD methodology's ability to enhance application development speed becomes crucial, enabling quick results and accelerated development. This approach facilitates swift delivery of parts or functionalities to users, boosting development pace and ensuring user involvement at various development stages, thereby elevating the final product's quality.

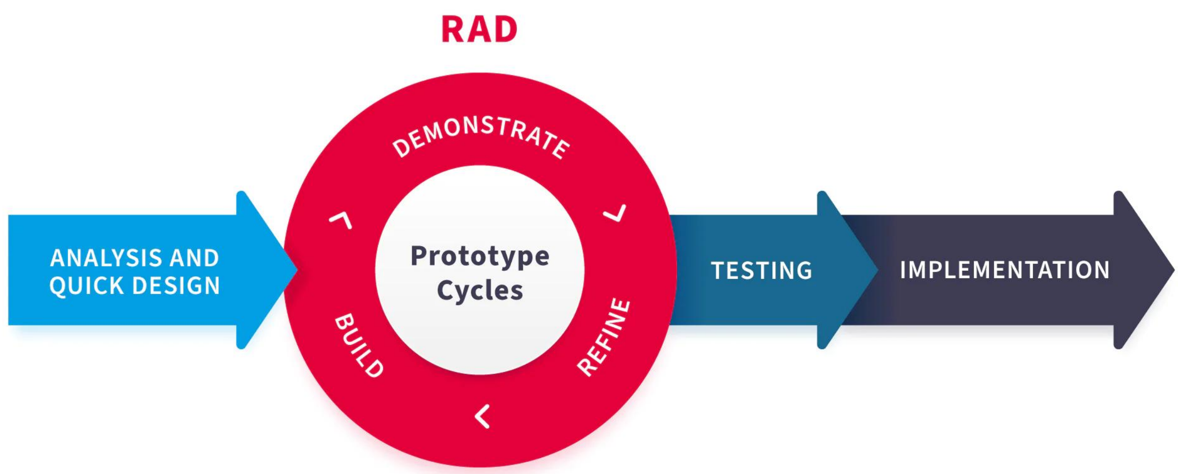


Figure 3.0.1: Rapid Application Development Cycle

RAD consists of four primary phases:

1. Analysis and Quick Design
2. Prototype Cycle
3. Testing
4. Deployment/Implementation

These phases operate concurrently, initiating with analysis and research on mobile application development to gather end-user requirements and domain knowledge from experts. The data collected is recorded as functional and nonfunctional requirements, translated into Unified Modeling Language (UML) for prototyping. Functional requirements include features like user account creation, mode selection, symptom selection, and medication consumption reminders, while nonfunctional requirements cover user authentication, application availability, user interface simplicity, Android version compatibility, and database updates.

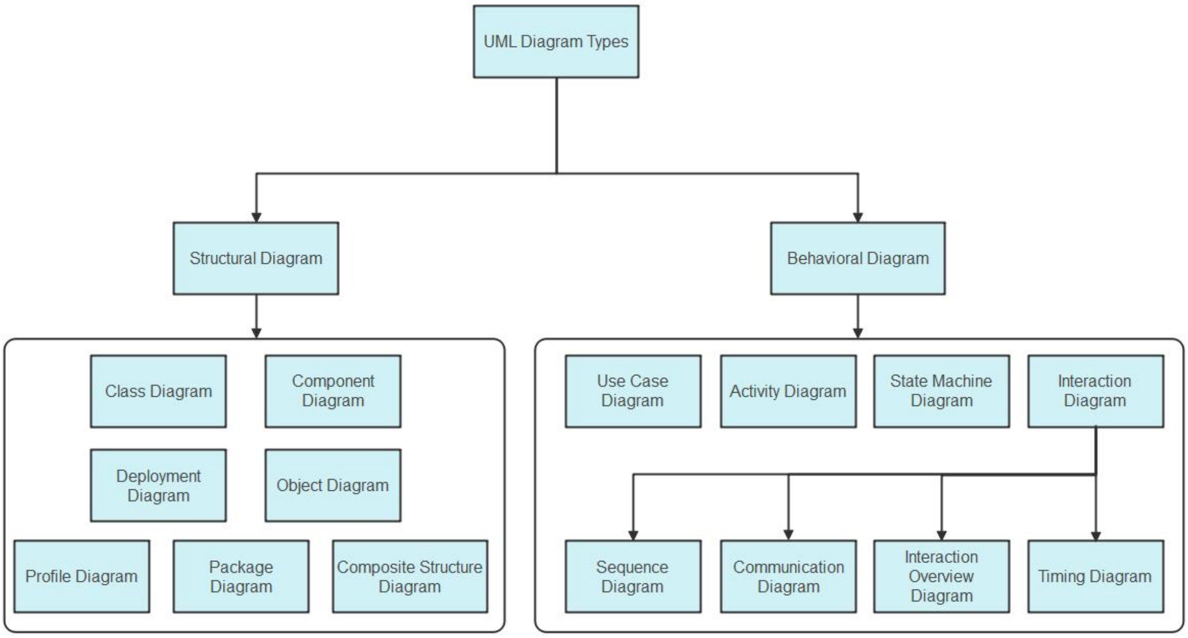


Figure 3.0.2: UML Diagram Types

The methodology involves iterative processes, where prototypes are continually developed and refined based on feedback from users and supervisors, ensuring that the evolving prototypes meet all specified user needs. Testing phases aim to validate and verify the application, conducting usability and acceptance tests with users to identify operational issues. Finally, the implementation phase marks the transition to the production phase, involving construction, installation, and ongoing maintenance of the application. Throughout the development, prototypes are incrementally improved based on user recommendations, with the completion verified through user acceptance testing, ensuring the app meets all requirements and user expectations.

3.1 UML Diagrams

Use case Diagram

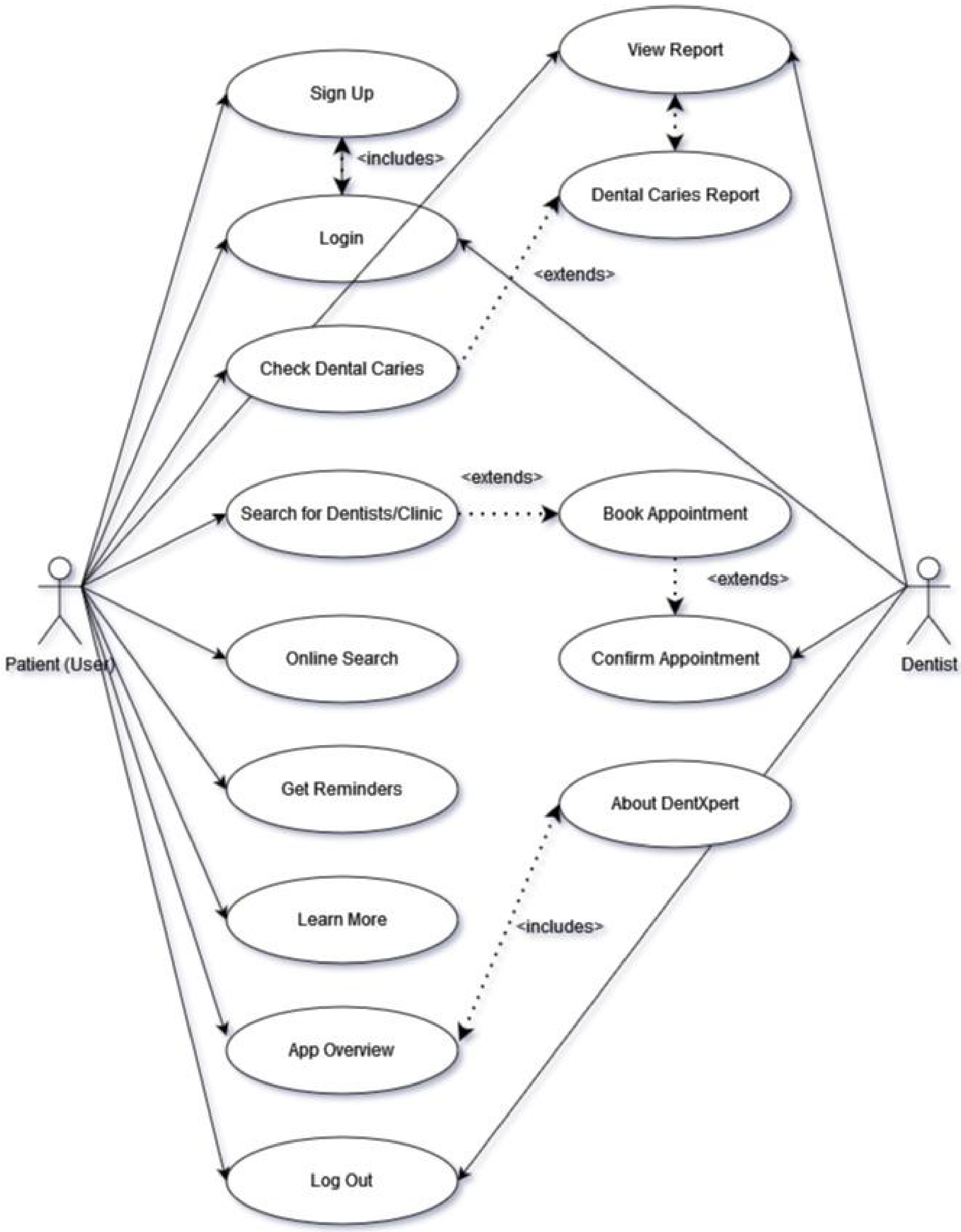


Figure 3.1.1: Use Case Diagram

Class Diagram

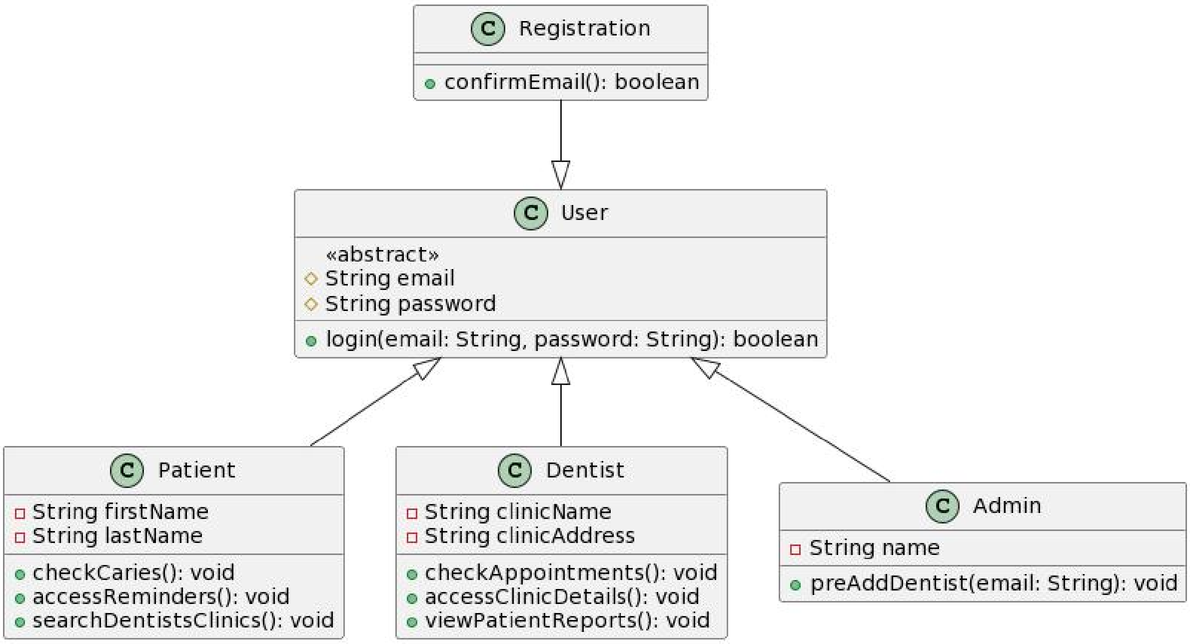


Figure 3.1.2: Class Diagram

Activity Diagram (Login):

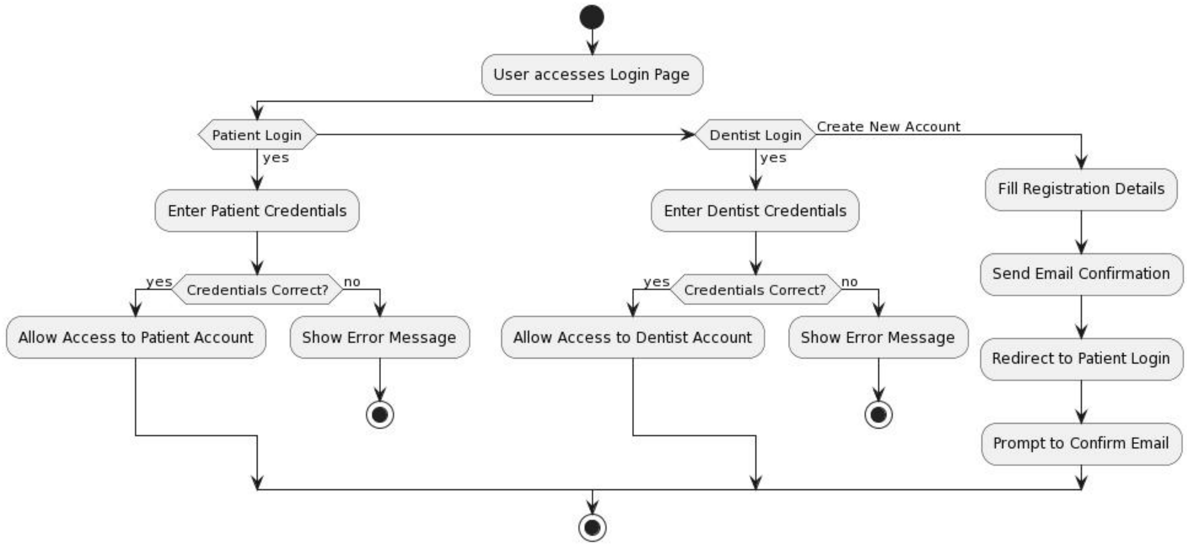


Figure 3.1.3: Activity Diagram (Login)

Activity Diagram (Caries Detection):

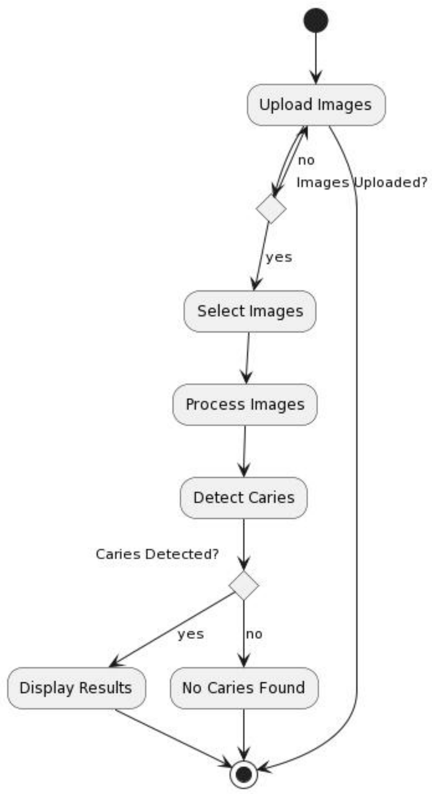


Figure 3.1.4: Activity Diagram (Caries Detection)

Component Diagram:

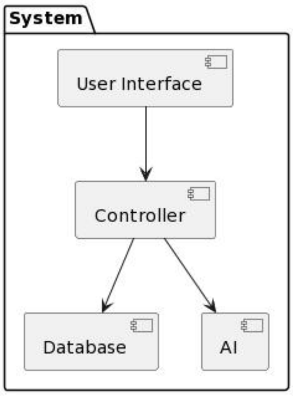


Figure 3.1.5: Component Diagram

Sequence Diagram

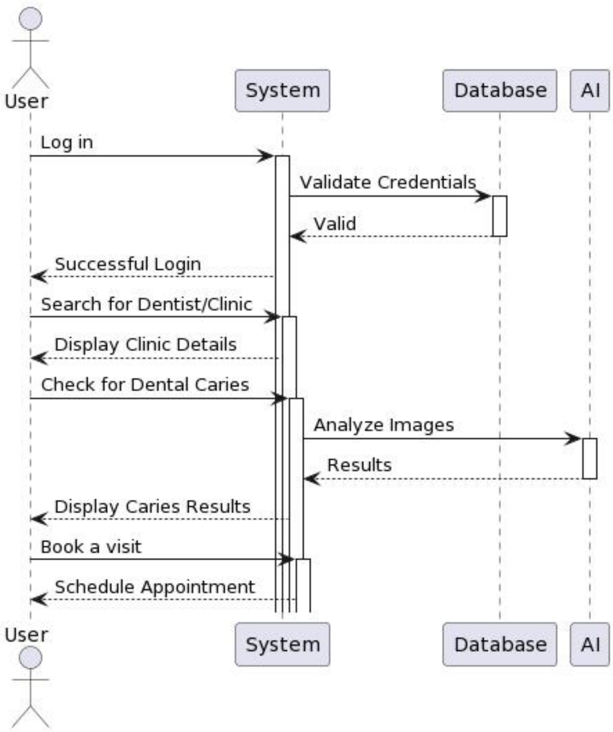


Figure 3.1.6: Sequence Diagram

3.2 Design

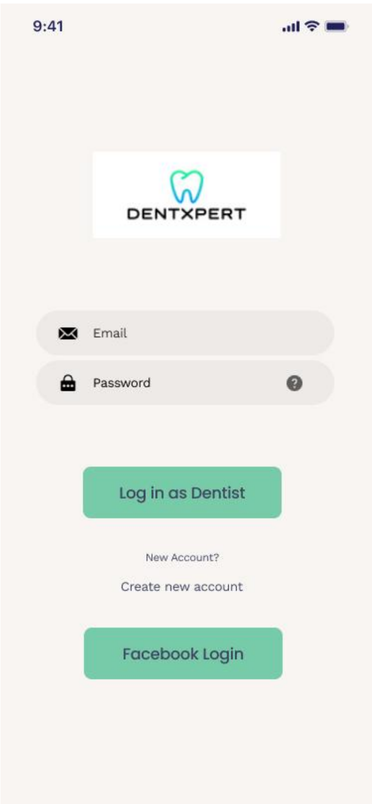
In this section, the methodology employed in designing the dental caries detection app is detailed. The design process encompasses the conceptualization and structuring of the app's interface, functionalities, and user interactions.

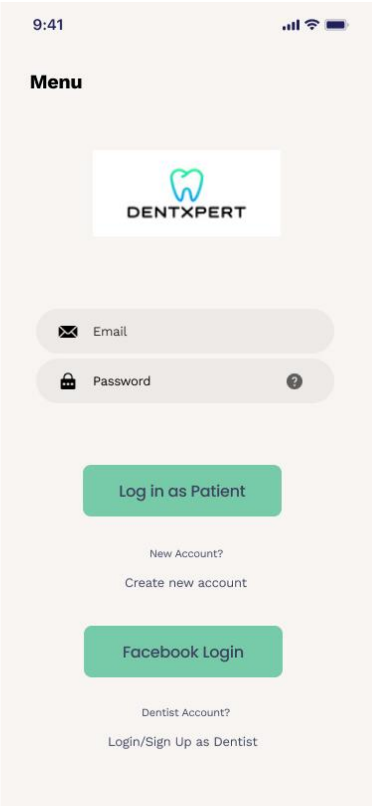
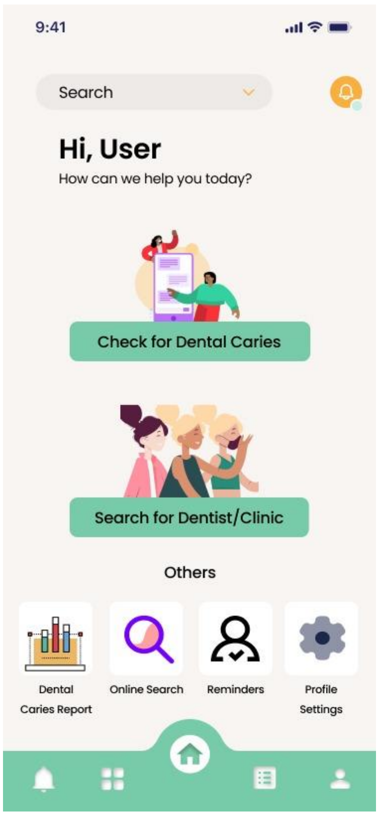
Conceptualization and Planning

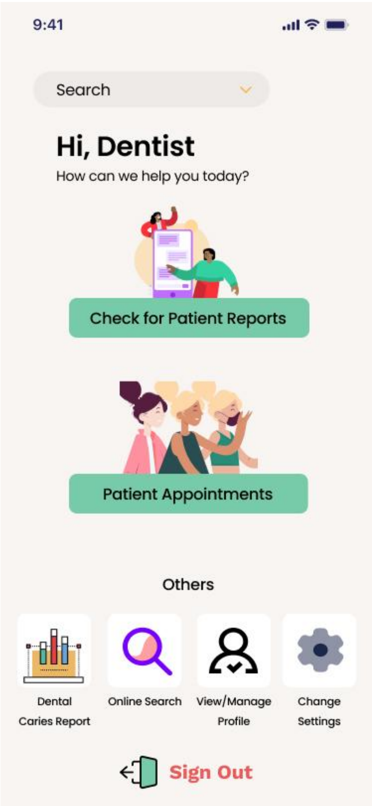
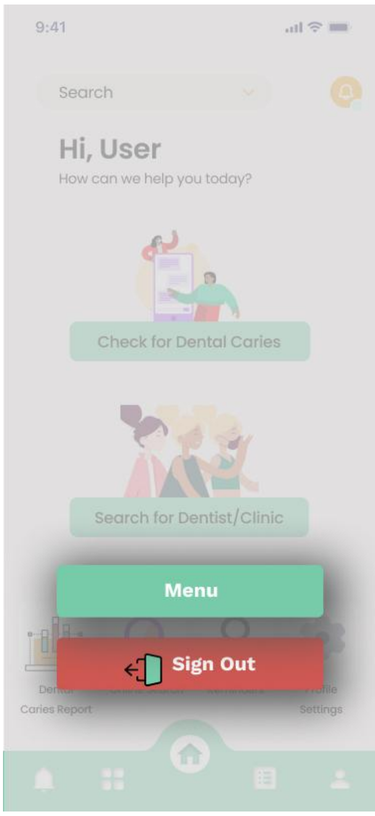
The design phase initiates with the conceptualization of the app's structure. It involves outlining the user flow, identifying key features, and wireframing the app's interface. This step aims to establish a blueprint for the app's development.

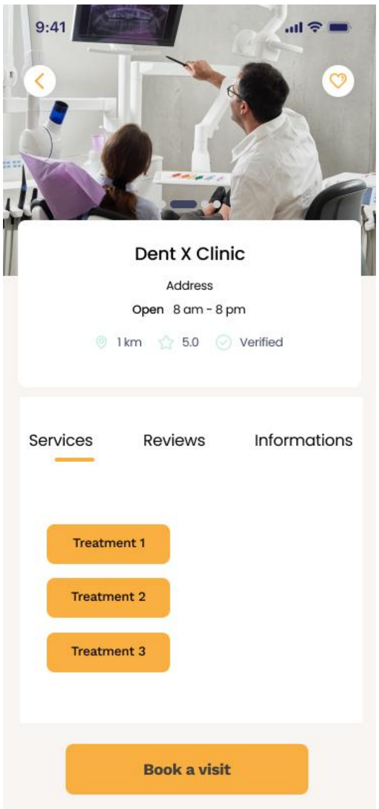
User Interface Design

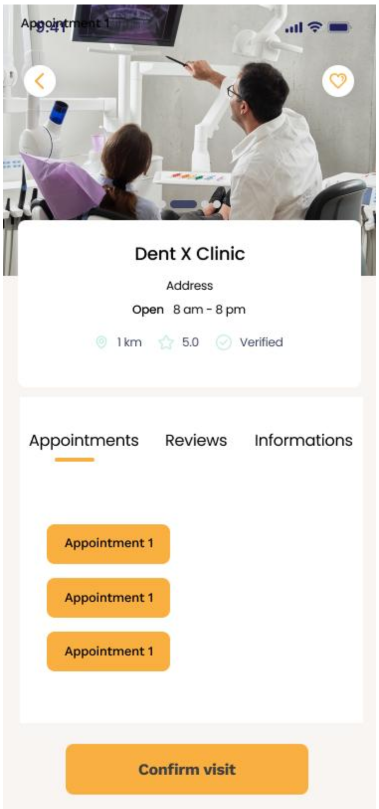
Following the conceptualization, the actual user interface (UI) design takes shape. Designers focus on creating an intuitive and visually appealing interface, incorporating elements for image upload, login mechanisms, and interactions with the YOLOv8 model for caries detection.

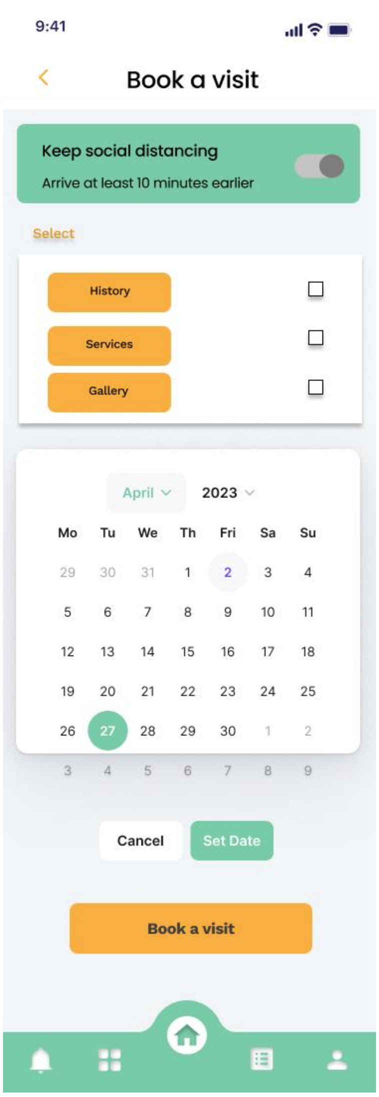
File	Description
	**Dentist Login Page:** Email input field Password input field Forgot password button Button Option: Log in as Dentist Optional: New Account Create new Account Button Option: Facebook Login

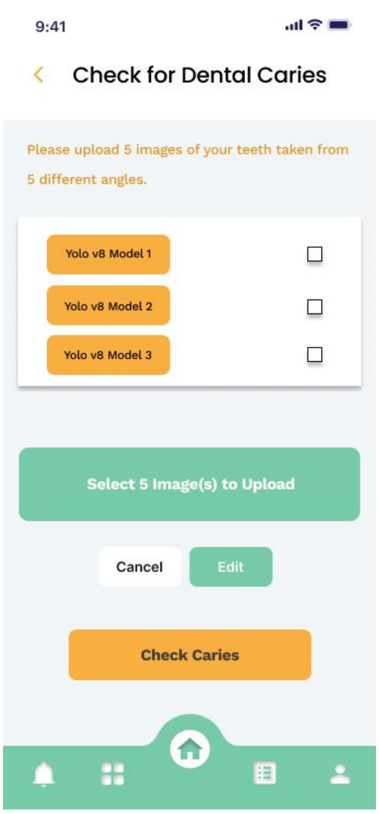
File	Description
	**Patient Login Page:** Email input field Password input field Forgot password button Button Option: Log in as Patient Optional: New Account Create new Account Button Option: Facebook Login Dentist Account option: Login/Sign up as Dentist
	**Patient Home Page:** Search input Notification button Button Option: Check for Dental Caries Button Option: Search for Dentist/Clinic Optional: Dental Caries Report | Online Search | Reminders | Profile Settings Navigation Bar: Notifications | Contents | Home | Learn | Profile

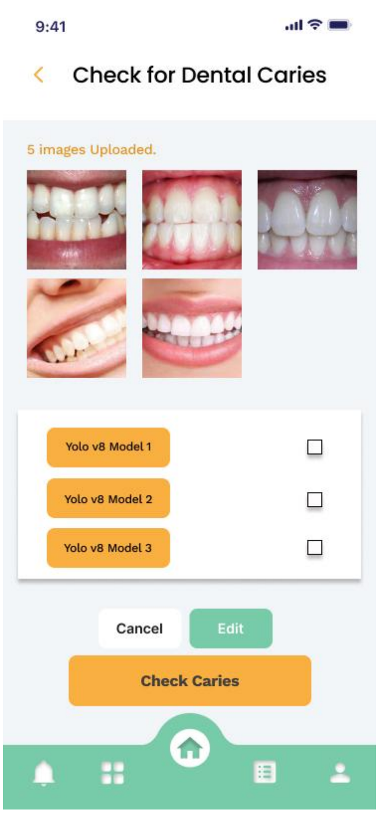
File	Description
	**Dentist Home page:** Search Input Button Option: Check for Patient Reports Button Option: Patient Appointments Optional: Dental Caries Report | Online Search | View\Manage Profile | Change settings Button Option: Sign Out
	**Log out Page:** Button Option: Menu Button Option: Sign out

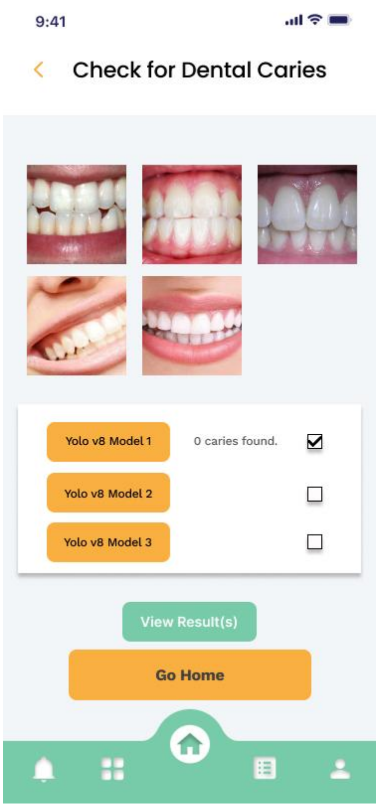
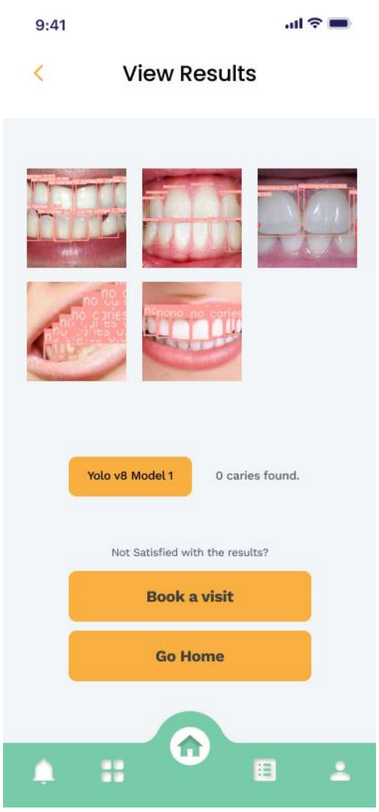
File	Description
	**Clinic details page:** Button: Back Clinic Details Clinic Name Clinic Address Open Status | Open hours Distant from location | Rating | Verified/Not-Verified Clinic Details Options: Services | Reviews | Information Button Option: Treatment 1 Button Option: Treatment 2 Button Option: Treatment 3 Button Option: Book a visit

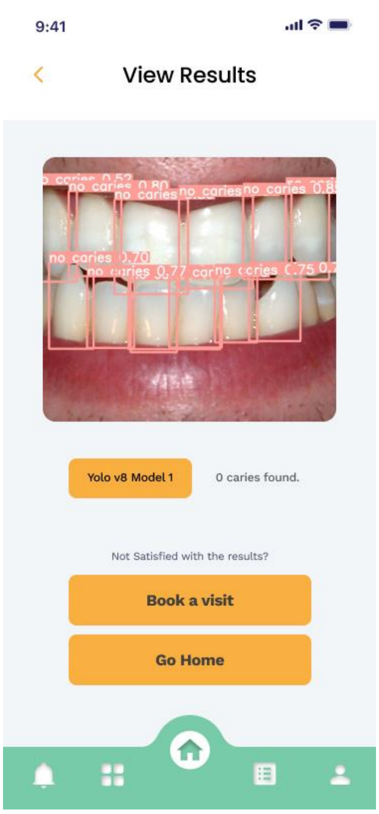
File	Description
 	**Clinic details - 1 page :** Button: Back image input : Clinic Details : Clinic Name Clinic Address Open Status | Open hours Distant from location | Rating | Verified/Not-Verified Clinic details option : Services | Reviews | Information Button Option: Appointment 1 Button Option: Appointment 1 Button Option: Appointment 1 Button Option: Confirm visit

File	Description
	**Book a Visit Page:** Button: Back Toggle button : Keep social distancing (note : Arrive at least 10 minutes earlier) Select option - Compliance Checklist : History Service Gallery Date Selection: Dropdown list: Months | Year Date picker: calendar widget Button Option: Cancel Button Option: Set Date Button Option: Book a visit Navigation bar: Notifications | Contents | Home | Learn | Profile

File	Description
	**Check for Dental Caries Page:** Button: Back Compliance Checklist: Yolo v8 Model 1 Yolo v8 Model 2 Yolo v8 Model 3 Button Option: Select 5 image(s) to Upload Button Option: Cancel Button Option: Edit Button Option: Check Caries Navigation bar: Notifications | Contents | Home | Learn | Profile

File	Description
	**Check for Dental Caries Page 2:** Button: Back (X) images uploaded: Display image uploaded compliance checklist ; Yolo v8 Model 1 Yolo v8 Model 2 Yolo v8 Model 3 Button Option: Select 5 image(s) to Upload Button Option: Cancel Button Option: Edit Button Option: Check Caries Navigation bar: Notifications | Contents | Home | Learn | Profile

File	Description
	**Check for Dental Caries Page 3:** Button: Back Display image uploaded : Compliance Checklist: Yolo v8 Model 1 Yolo v8 Model 2 Yolo v8 Model 3 Button Option: View Result(s) Button Option: Go Home Navigation bar: Notifications | Contents | Home | Learn | Profile
	**View Results Page :** Button: Back Display image : Result(s) of uploaded images Caries Result(s) : Yolo v8 Model (X) | Number of caries found Button Option: Book a visit Button Option: Go Home Navigation bar: Notifications | Contents | Home | Learn | Profile

<table>
<tr><th>File</th><th>Description</th></tr>
<tr><td>
</td><td>View Results - 1 page :

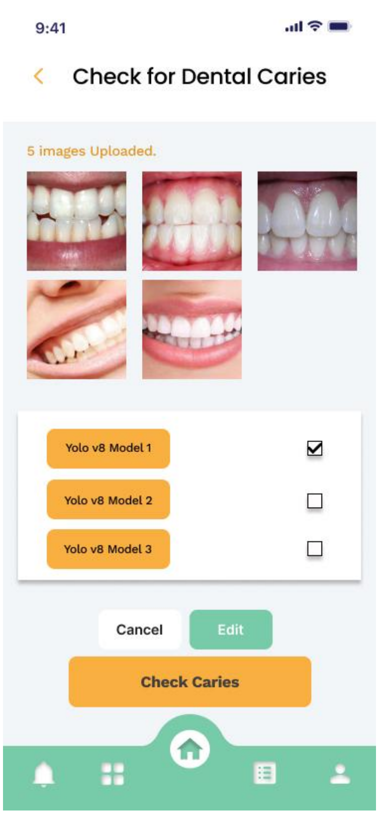
Button: Back

Display image : selected result of uploaded images

Caries Result(s) :
Yolo v8 Model (X) | Number of caries found

Button Option: Book a visit

Button Option: Go Home

Navigation bar: Notifications | Contents | Home | Learn | Profile</td></tr>
<tr><td>
</td><td>Check for Dental Caries - 4 page :

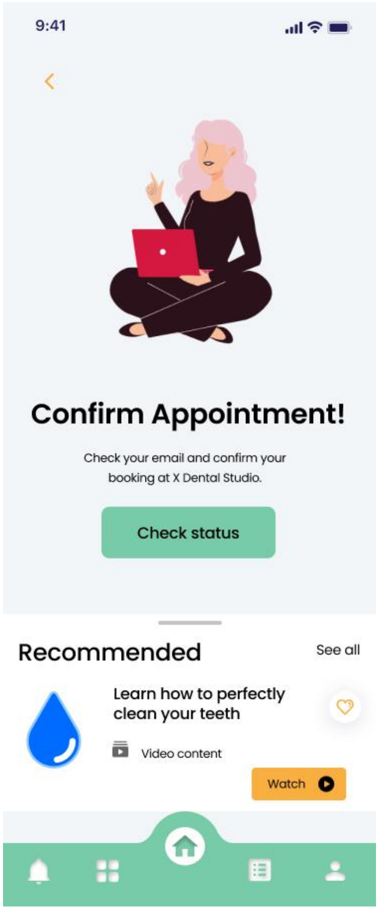
Button: Back

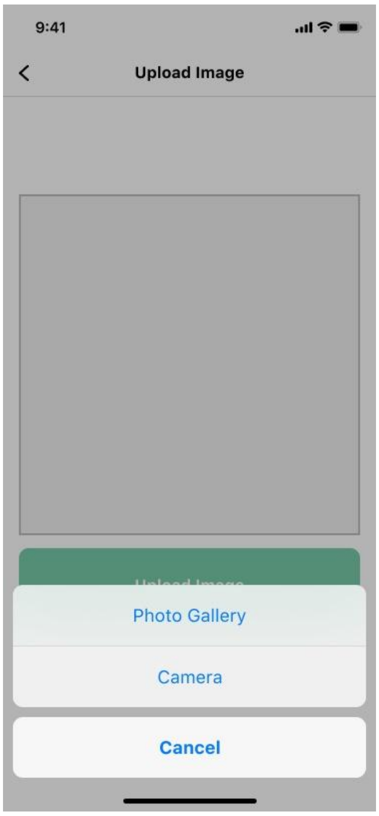
(X) images uploaded:
display image uploaded

Compliance Checklist:
Yolo v8 Model 1
Yolo v8 Model 2
Yolo v8 Model 3

Button Option: Select 5 image(s) to Upload

Button Option: Cancel
Button Option: Edit</td></tr>
</table>

File	Description
	Button Option: Check Caries Navigation bar: Notifications | Contents | Home | Learn | Profile
	**Confirmation Page :** Button: Back Button Option: Check Status Auto-suggest : Content title and Content link Navigation bar: Notifications | Contents | Home | Learn | Profile

File	Description
	**Upload image Page:** Display image Button Option: Upload image Device options: Photo Gallery Camera Cancel
	**Camera Upload:** Button: Flash - On/off Button: Zoom Button Option: Cancel | Shutter | Flip Camera

3.3 Implementation

The implementation phase details the process of transforming the design concepts into a functional dental caries detection app. This section encompasses the actual development of both front-end and back-end components.

The implementation phase marks the conclusive step in crafting this mobile application. It encompasses the actual creation and installation of the application itself. This stage signifies the transition from the developmental phase to the production phase, where the application reaches a stable state ready for installation on end-user devices. Maintenance and any subsequent updates for the mobile application are also part of this phase.

Given the selection of RAD (Rapid Application Development) methodology for this project, all prototypes developed are presented to end-users and domain experts for their insights and feedback. The mobile app undergoes incremental enhancements based on user suggestions and recommendations. Upon completion and successful construction of the mobile application, user acceptance testing becomes pivotal. This testing phase ensures that the mobile app aligns with all stipulated requirements, instilling confidence in developers. Summarized assessments from user acceptance testing are vital for documentation, validation, and verification of the mobile application's effectiveness.

Front-end Development

Front-end developers work on translating the UI design into code using suitable technologies. They create the visual elements, incorporate functionalities for image uploading, user authentication, and ensure seamless integration with the ML model.

Back-end Development

Simultaneously, the back-end development team builds the core infrastructure of the app. This involves setting up servers, databases, and APIs necessary for communication between the front-end and the YOLOv8 model. It ensures a robust foundation for data processing and retrieval.

3.4 Testing/Evaluation

The testing and evaluation phase focuses on assessing the app's functionality, performance, and accuracy in caries detection. It involves rigorous testing to ensure the app meets predefined standards and addresses user needs effectively.

Functional Requirements for the Dental Caries App:

1. User Authentication:

- ✓ Email and password fields for both dentist and patient logins.
- ✓ Forgot password functionality for account recovery.
- ✓ Log in options differentiated for Dentists and Patients.
- ✓ Social media integration for login via Facebook.

2. Account Creation and Management:

- ✓ Ability for new users to create accounts.
- ✓ Dentist-specific login/signup functionality.
- ✓ Differentiated modes (Family/Self) for patient accounts.
- ✓ Dentist-patient account interaction features.

3. Navigation and Home Page Features:

- ✓ Search functionalities for dental checks and clinics.
- ✓ Notification alerts for updates and reminders.
- ✓ Accessible options like Dental Caries Report, Online Search, Reminders, and Profile Settings.
- ✓ Segmented navigation bar for easy access to various sections.

4. Clinic Details & Booking:

- ✓ Detailed clinic information display including name, address, ratings, and services.
- ✓ Options to book appointments for various treatments.
- ✓ Multi-page wireframe displays for clinic details to streamline information.

5. Appointment Booking:

- ✓ Option to set dates, maintain social distancing preferences, and select specific treatments for clinic visits.
- ✓ Image upload functionality for dental check-up assessments.

6. Check for Dental Caries:

- ✓ Multi-page wireframe layouts for checking dental issues through image uploads.
- ✓ Compliance checklist for selecting specific dental models and uploaded images for examination.

7. Viewing Results:

- ✓ Pages displaying results from dental examinations using Yolo v8 Models.
- ✓ Display images and detailed results of caries found.
- ✓ Option to book visits based on the results.

8. Confirmation and Menu Pages:

- ✓ Confirmation page layout for checking status and suggested content.
- ✓ Menu wireframe for easy access to various app sections.
- ✓ Image upload functionality via camera or photo gallery.

Nonfunctional Requirements for the Dental Caries App:

1. Reliability and Availability:

- ✓ 95% availability to ensure users can access the app most of the time.
- ✓ Reliable user authentication for secure logins.

2. Performance and Compatibility:

- ✓ User-friendly interface design for easy navigation.
- ✓ Support for a minimum of 100 concurrent users without performance issues.
- ✓ Platform compatibility to run on Android 2.3 and higher versions.

3. Regular Updates and Maintenance:

- ✓ Weekly updates of the medicine database to reflect the latest information.
- ✓ App maintenance and updates for enhanced performance.

4. User Experience:

- ✓ Structured wireframe pages for clear and intuitive navigation.
- ✓ Timely notifications and reminders for improved user experience.

Functional Requirements	**Description**
User Authentication	- Email/password fields for both dentist and patient logins - Forgot password functionality - Differentiated logins for Dentists and Patients - Social media integration (Facebook)
Account Creation & Management	- New user account creation - Dentist-specific login/signup - Differentiated patient account modes (Family/Self) - Dentist-patient account interaction features
Navigation & Home Page	- Search for dental checks & clinics - Notification alerts - Accessible options like Dental Caries Report, Online Search, Reminders, Profile Settings - Segmented navigation bar
Clinic Details & Booking	- Detailed clinic information display - Booking options for treatments - Multi-page wireframes for clinic details
Appointment Booking	- Date setting, social distancing preferences - Image upload for dental check-ups
Check for Dental Caries	- Multi-page layouts for image uploads - Compliance checklist for dental models/images
Viewing Results	- Pages displaying dental examination results - Display images and detailed caries findings - Visit booking based on results
Confirmation & Menu Pages	- Confirmation page for status checks - Menu wireframe for app sections - Image upload via camera or gallery

Figure 3.4.1: Functional Requirements

Non-Functional Requirements	**Description**
Reliability & Availability	- 95% availability - Secure and reliable user authentication
Performance & Compatibility	- User-friendly interface - Support for 100 concurrent users - Compatibility with Android 2.3+
Regular Updates & Maintenance	- Weekly medicine database updates - Ongoing app maintenance and performance updates
User Experience	- Structured wireframes for intuitive navigation - Timely notifications and reminders

Figure 3.4.2: Non-Functional Requirements

Functionality Testing

Quality assurance experts conduct extensive testing, evaluating the app's functionalities such as image uploads, login mechanisms, and the integration with the YOLOv8 model. This step aims to identify and rectify any bugs or issues within the app.

Model Integration Testing

Specific attention is given to the integration of the YOLOv8 model within the app's back-end. The accuracy and efficiency of caries detection from uploaded images are verified and validated.

User Experience Evaluation

User feedback and usability testing are essential components. Gathering user insights helps in refining the app's usability, ensuring an intuitive and user-friendly experience.

This comprehensive methodology outlines the sequential process from design conception to app implementation and thorough testing. It ensures the development of a robust and efficient dental caries detection app aligned with defined objectives and user requirements.

Figure 3.4.1: Requirements Engineering

3.5 Deep Learning Model

3.5.1 Objectives

- ✓ Annotate a comprehensive dataset to facilitate model training.
- ✓ Develop a YOLOv8-based deep learning model for dental caries detection.
- ✓ Deploy a test web service for real-time object detection in dental images.

3.5.2 Data Preparation

Dataset Overview

The dataset, available at Roboflow Universe, comprises 7000+ images with annotations for dental caries.

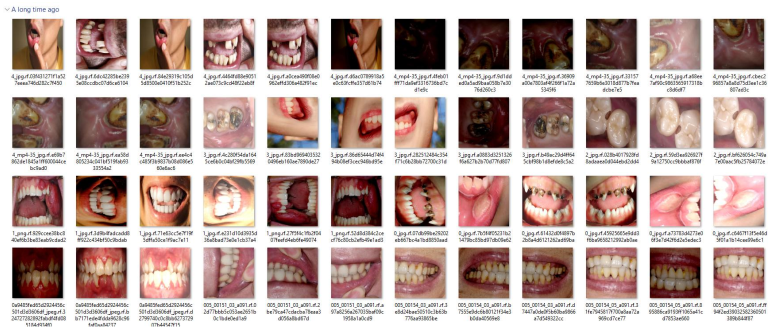


Figure 3.5.2.1: The Dataset containing 7000+ images

Image Labeling

Image labeling was performed using PyCharm, providing accurate annotations for dental caries in each image. The labeling process included defining bounding boxes around caries regions.

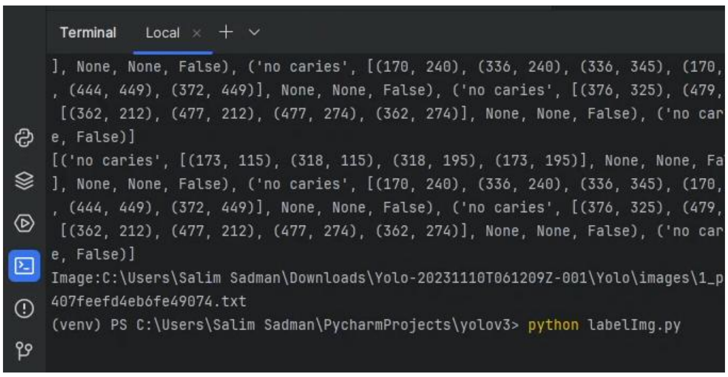


Figure 3.5.2.2: Running labelImg library on PyCharm Terminal

Figure 3.5.2.3: Running labelImg Window

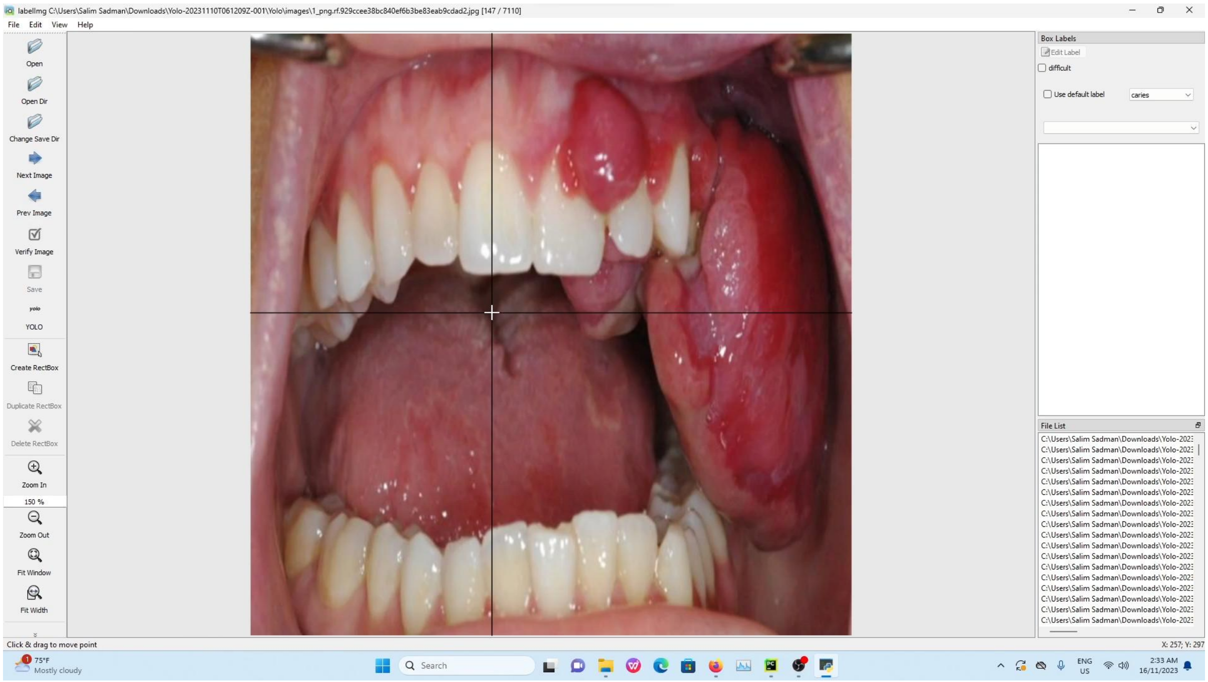


Figure 3.5.2.4: Loading an image into the labelImg Window

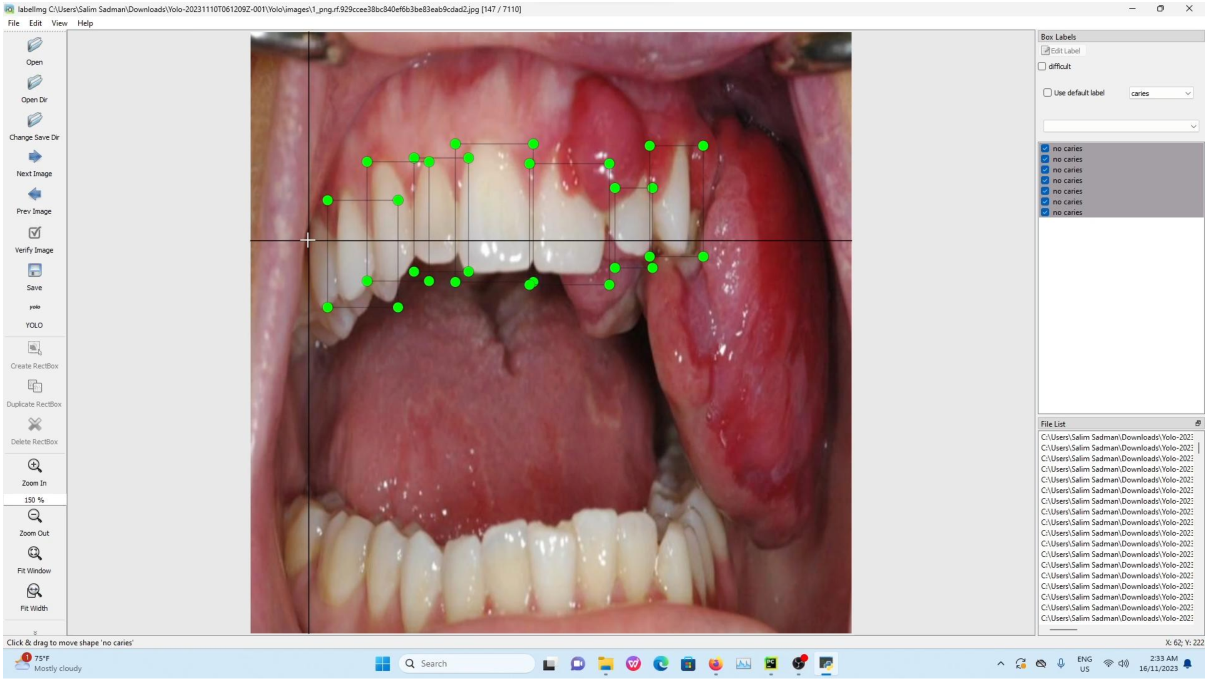


Figure 3.5.2.5: Labelling the image with classes (caries, no-caries) in labelImg Window

```
File    Edit    View

0 0.592187 0.346094 0.215625 0.151562
0 0.657031 0.471094 0.164062 0.110937
0 0.644531 0.619531 0.235937 0.148438
0 0.692969 0.754687 0.229687 0.143750
0 0.752344 0.845313 0.142187 0.075000
0 0.582031 0.192969 0.164062 0.132812
0 0.243750 0.300781 0.212500 0.170313
0 0.296875 0.147656 0.181250 0.198437
```

Figure 3.5.2.6: .txt file of the labels saved alongside the images from the dataset

3.5.3 Model Training

Environment Setup

An Anaconda Navigator environment was created, incorporating the Ultralytics YOLOv8 library for model training.

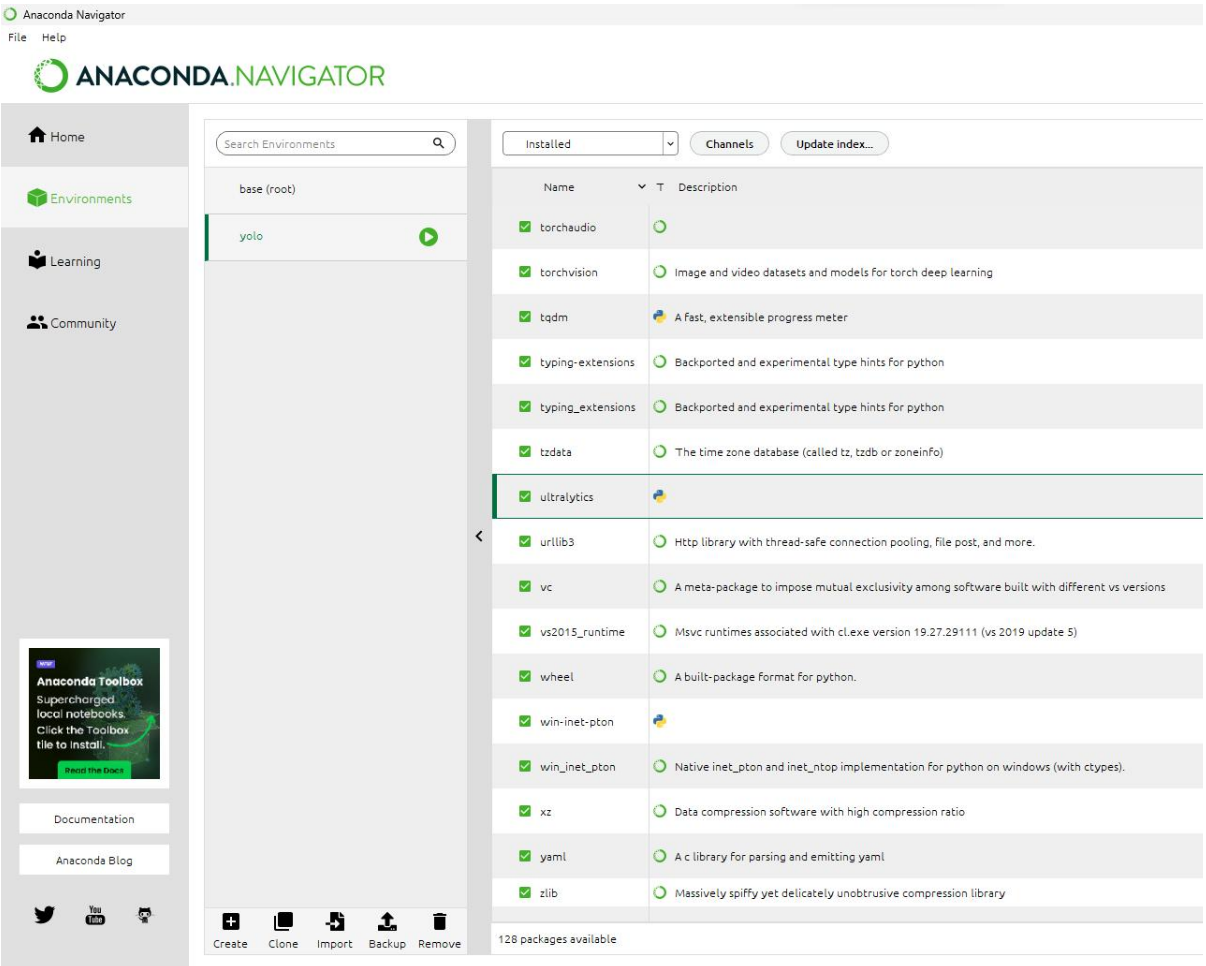
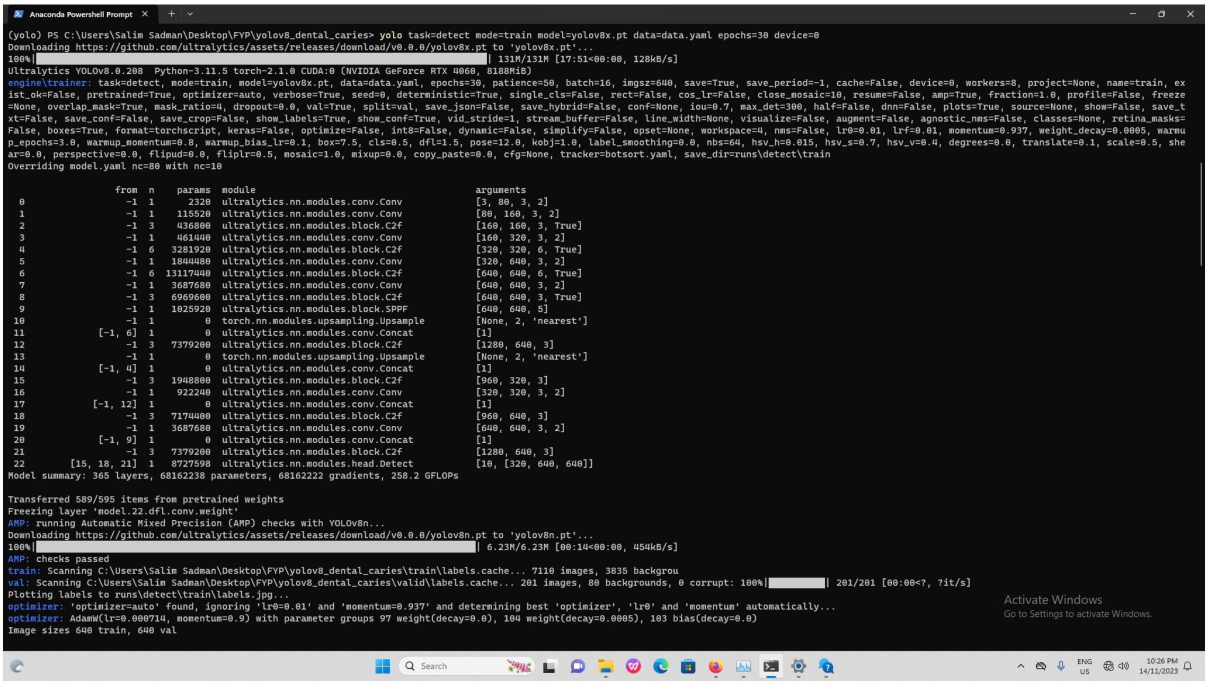


Figure 3.5.3.1: Creating yolo environment in Anaconda Navigator

Model Configuration

The YOLOv8 model was configured with appropriate hyperparameters, including the choice of YOLOv8 variant and training epochs.



```
Anaconda Powershell Prompt
(yolo) PS C:\Users\Salim Sadman\Desktop\FYP\yolov8_dental_caries> yolo task=detect mode=train model=yolov8x.pt data=data.yaml epochs=30 device=0
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8x.pt to 'yolov8x.pt'...
100%|                                                            | 131M/131M [17:51<00:00, 128kB/s]
Ultralytics YOLOv8.0.208  Python-3.11.5 torch-2.1.0 CUDA:0 (NVIDIA GeForce RTX 4060, 8188MiB)
engine\trainer: task=detect, mode=train, model=yolov8x.pt, data=data.yaml, epochs=30, patience=50, batch=16, imgsz=640, save=True, save_period=-1, cache=False, device=0, workers=8, project=None, name=train, exist_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single_cls=False, rect=False, cos_lr=False, close_mosaic=10, resume=False, amp=True, fraction=1.0, profile=False, freeze=None, overlap_mask=True, mask_ratio=4, dropout=0.0, val=True, split=val, save_json=False, save_hybrid=False, conf=None, iou=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, show=False, save_txt=False, save_conf=False, save_crop=False, show_labels=True, show_conf=True, vid_stride=1, stream_buffer=False, line_width=None, visualize=False, augment=False, agnostic_nms=False, classes=None, retina_masks=False, boxes=True, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify=False, opset=None, workspace=4, nms=False, lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label_smoothing=0.0, nbs=64, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0, cfg=None, tracker=botsort.yaml, save_dir=runs\detect\train
Overriding model.yaml nc=80 with nc=10

                   from  n    params  module                                       arguments
  0                  -1  1      2320  ultralytics.nn.modules.conv.Conv             [3, 80, 3, 2]
  1                  -1  1    115520  ultralytics.nn.modules.conv.Conv             [80, 160, 3, 2]
  2                  -1  3    436800  ultralytics.nn.modules.block.C2f             [160, 160, 3, True]
  3                  -1  1    461440  ultralytics.nn.modules.conv.Conv             [160, 320, 3, 2]
  4                  -1  6   3281920  ultralytics.nn.modules.block.C2f             [320, 320, 6, True]
  5                  -1  1   1844480  ultralytics.nn.modules.conv.Conv             [320, 640, 3, 2]
  6                  -1  6  13117440  ultralytics.nn.modules.block.C2f             [640, 640, 6, True]
  7                  -1  1   3687680  ultralytics.nn.modules.conv.Conv             [640, 640, 3, 2]
  8                  -1  3   6969600  ultralytics.nn.modules.block.C2f             [640, 640, 3, True]
  9                  -1  1   1025920  ultralytics.nn.modules.block.SPPF            [640, 640, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  3   7379200  ultralytics.nn.modules.block.C2f             [1280, 640, 3]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  3   1948800  ultralytics.nn.modules.block.C2f             [960, 320, 3]
 16                  -1  1    922240  ultralytics.nn.modules.conv.Conv             [320, 320, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  3   7174400  ultralytics.nn.modules.block.C2f             [960, 640, 3]
 19                  -1  1   3687680  ultralytics.nn.modules.conv.Conv             [640, 640, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  3   7379200  ultralytics.nn.modules.block.C2f             [1280, 640, 3]
 22        [15, 18, 21]  1   8727598  ultralytics.nn.modules.head.Detect           [10, [320, 640, 640]]
Model summary: 365 layers, 68162238 parameters, 68162222 gradients, 258.2 GFLOPs

Transferred 589/595 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to 'yolov8n.pt'...
100%|                                                            | 6.23M/6.23M [00:14<00:00, 454kB/s]
AMP: checks passed
train: Scanning C:\Users\Salim Sadman\Desktop\FYP\yolov8_dental_caries\train\labels.cache... 7110 images, 3835 backgrou
val: Scanning C:\Users\Salim Sadman\Desktop\FYP\yolov8_dental_caries\valid\labels.cache... 201 images, 80 backgrounds, 0 corrupt: 100%|          | 201/201 [00:00<?, ?it/s]
Plotting labels to runs\detect\train\labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.000714, momentum=0.9) with parameter groups 97 weight(decay=0.0), 104 weight(decay=0.0005), 103 bias(decay=0.0)
Image sizes 640 train, 640 val
```

Figure 3.5.3.2: Setting up the model on "Train Mode", device set to 0 (RTX 4060)

Training Process

The model was trained on the annotated dataset, optimizing for box and class losses. The training loop involved random batch extraction, forward pass through the model, loss calculation, and optimization using the Stochastic Gradient Descent (SGD) optimizer.

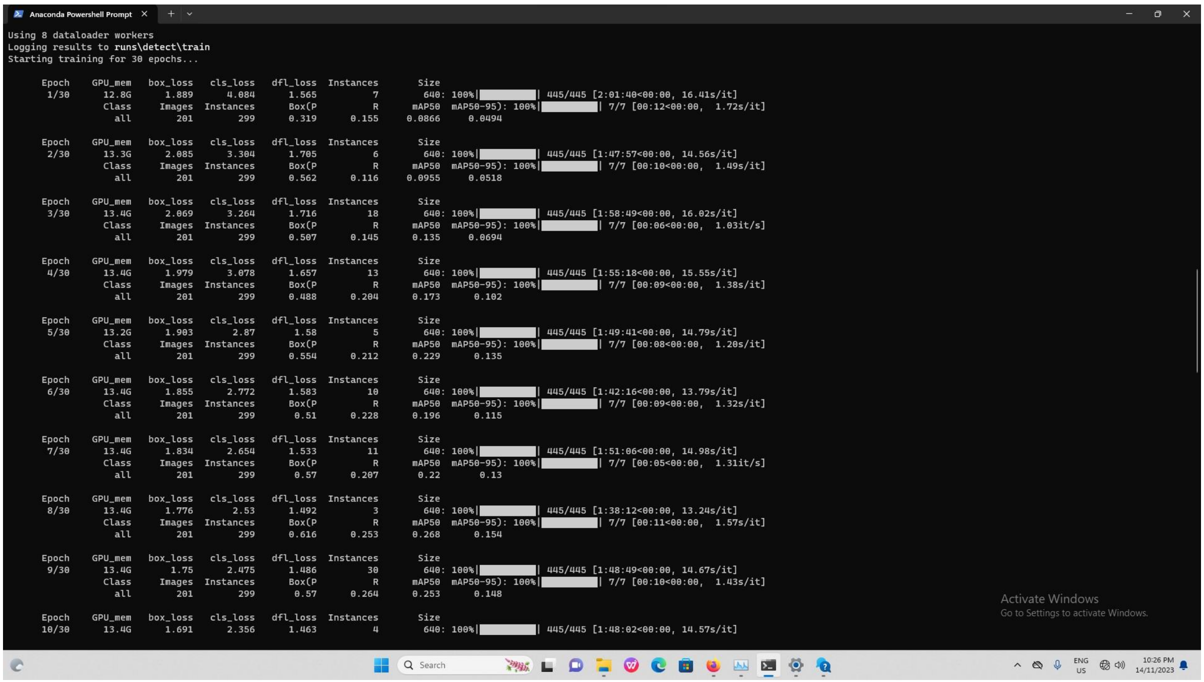


Figure 3.5.3.3: Epochs running on device 0 (RTX 4060)

3.5.4 Model Evaluation

Metrics such as box_loss, cls_loss, and mAP50-95 were monitored during training to evaluate the model's performance. The process iteratively improved the model's ability to detect dental caries.

3.5.5 Trained Model Visualization

Confusion Matrix

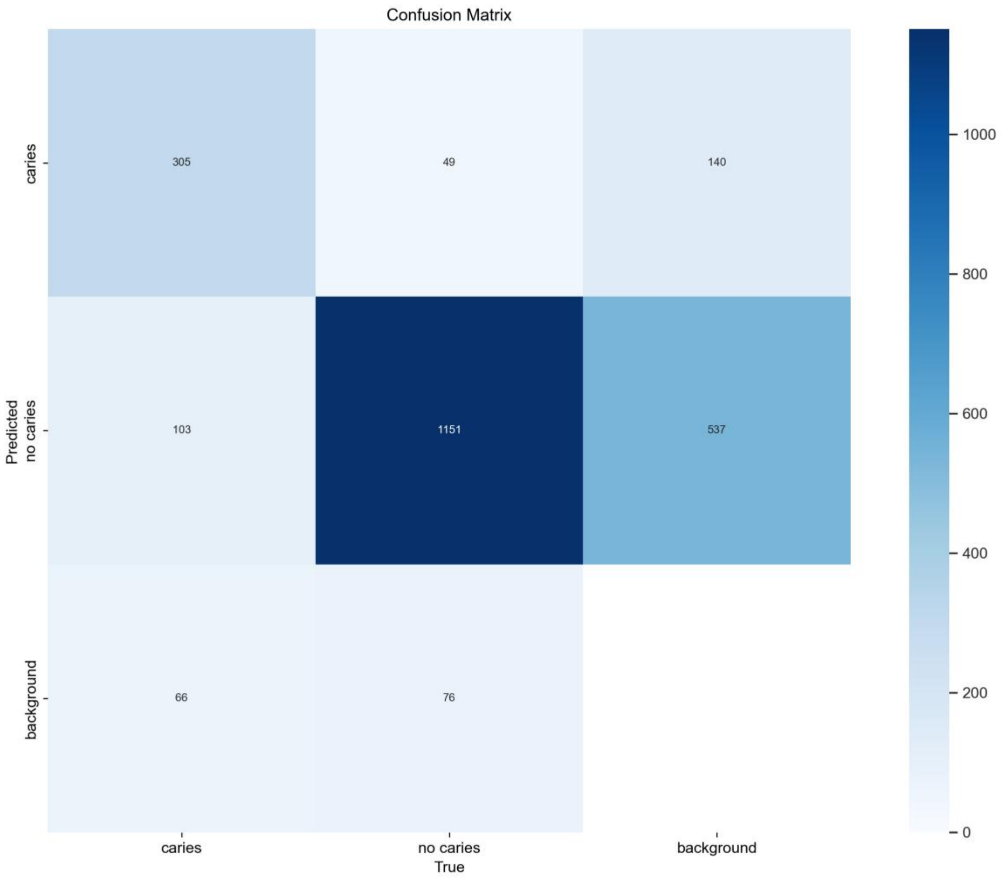


Figure 3.5.5.1: Confusion Matrix of the trained model

The confusion matrix is a table that shows how many times the model correctly classified an image, and how many times it made a mistake. In this case, the rows of the table represent the actual labels of the images, and the columns represent the labels that the model predicted. For example, the top left cell of the table shows that there were 1000 images that actually contained caries, and the model correctly classified all of them. The bottom right cell of the table shows that there were 600 images that actually contained background, and the model correctly classified 537 of them. The other cells in the table show the number of times that the model made mistakes. For example, the cell in the second row and second column shows that there were 103 images that actually did not contain caries, but the model predicted that they did. The confusion matrix can be used to calculate a number of different metrics that measure the performance of the model, such as accuracy, precision, recall, and F1 score. These metrics can be used to compare the performance of different models, or to track the performance of a single model over time. In the specific case of the confusion matrix, it appears that the model is doing a good job of classifying images that contain caries. However, it is making more mistakes on images that do not contain caries. This could be because there are more images of non-caries in the dataset, or because the model is not as good at distinguishing between non-caries and background images.

Loss Visualization

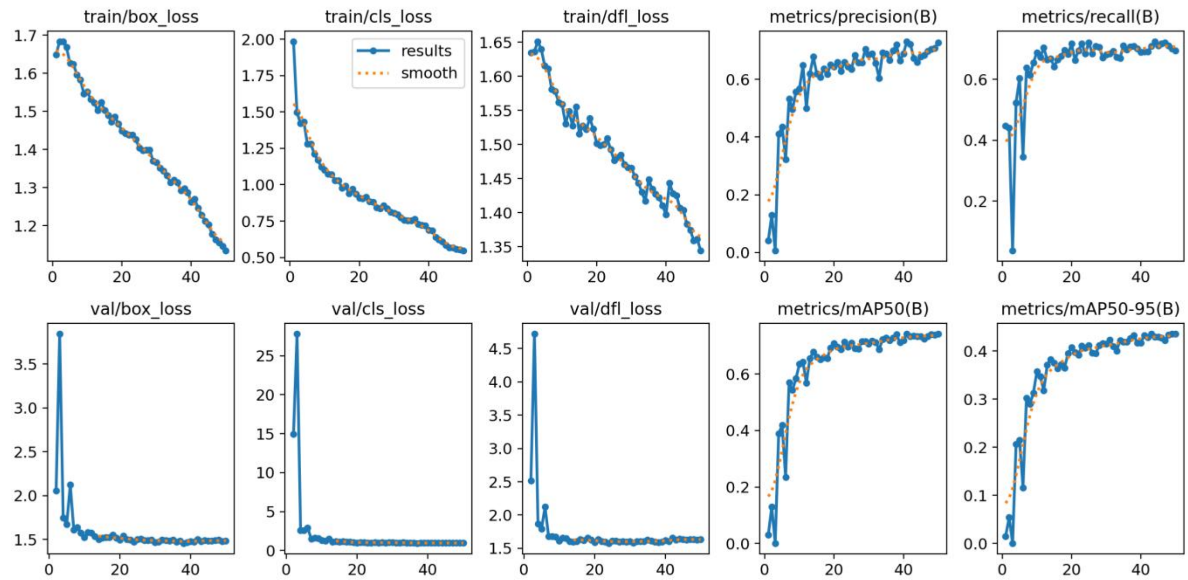


Figure 3.5.5.2: Loss Visualization of the trained model

Top Left Graph

Title: train/box loss

X-axis: Epochs (training iterations)

Y-axis: Loss value

Line: Loss curve during training

Observation: The loss steadily decreases as the model trains, indicating it's learning to make better predictions.

Top Middle Graph

Title: train/cls loss

X-axis & Y-axis: Same as previous graph

Line: Classification loss curve during training

Observation: Similar to the box loss, the classification loss also decreases, suggesting improvement in classifying objects.

Top Right Graph

Title: train/dfl loss

X-axis & Y-axis: Same as previous graphs

Line: Difficult example focus loss curve during training

Observation: The loss related to difficult examples shows some fluctuations but seems to trend downwards, implying the model is getting better at handling challenging cases.

Bottom Left Graph

Title: metrics/precision(B)

X-axis: Epochs

Y-axis: Precision value (0-1)

Line: Precision curve for a specific class (class B)

Observation: The precision for class B starts lower but gradually increases to around 0.6, indicating the model is becoming more accurate in identifying class B objects.

Bottom Middle Graph

Title: metrics/recall(B)

X-axis & Y-axis: Same as previous graph

Line: Recall curve for class B

Observation: The recall for class B also starts lower and climbs to around 0.6, suggesting the model is improving at finding all instances of class B.

Bottom Right Graph

Title: results

X-axis: Epochs

Y-axis: Values related to model performance (unclear without more context)

Lines: Multiple lines representing different metrics

It's hard to discern specific trends without knowing the exact metrics plotted, but it seems some values are improving with training. The graphs suggest that the machine learning model is successfully learning from the training data and improving its performance in object detection or classification.

F1-Confidence Curve

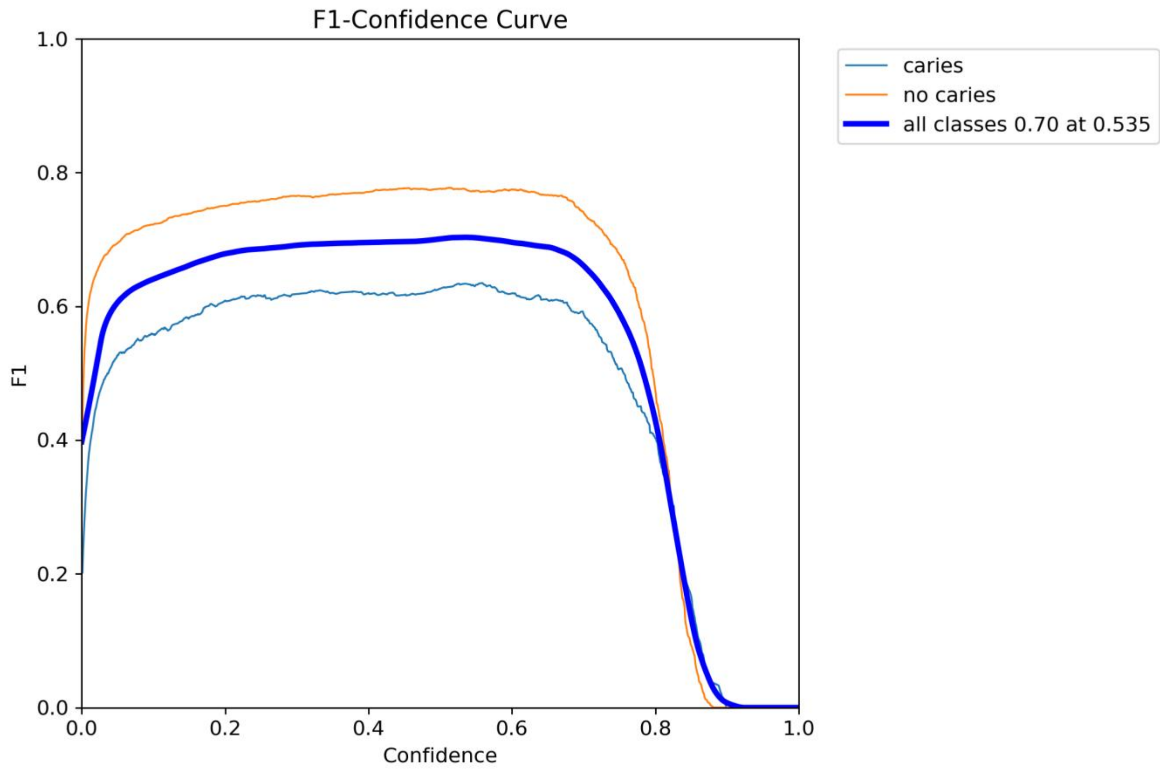


Figure 3.5.5.3: F1-Confidence Curve of the trained model

The curve visualizes the performance of a machine learning model in classifying dental images as containing caries (tooth decay), no caries, or all classes. It depicts the F1 score, which is a harmonic mean of precision and recall, at different confidence thresholds.
X-axis: Confidence threshold (0 to 1). This represents the model's certainty in its predictions. A higher threshold means the model is only considering predictions it's very confident about.
Y-axis: F1 score (0 to 1). This metric combines precision (correctly identified caries) and recall (all actual caries identified) into a single measure of the model's effectiveness. A score of 1 indicates perfect precision and recall. Curve: The blue line shows the F1 score for all

caries classes (no differentiation between types). The green and red lines represent specific caries classes, but without labels in the image, it's unclear which classes they correspond to. Overall, the F1 score decreases as the confidence threshold increases. This means the model is more likely to make mistakes when it's only considering very confident predictions. The blue curve peaks at around 0.535, with an F1 score of approximately 0.70. This indicates that the model achieves its best balance of precision and recall at this confidence level, considering all caries classes together. The green and red curves have different shapes and peak at different confidence thresholds. This suggests that the model's performance varies depending on the specific caries class being considered.

Precision-Confidence Curve

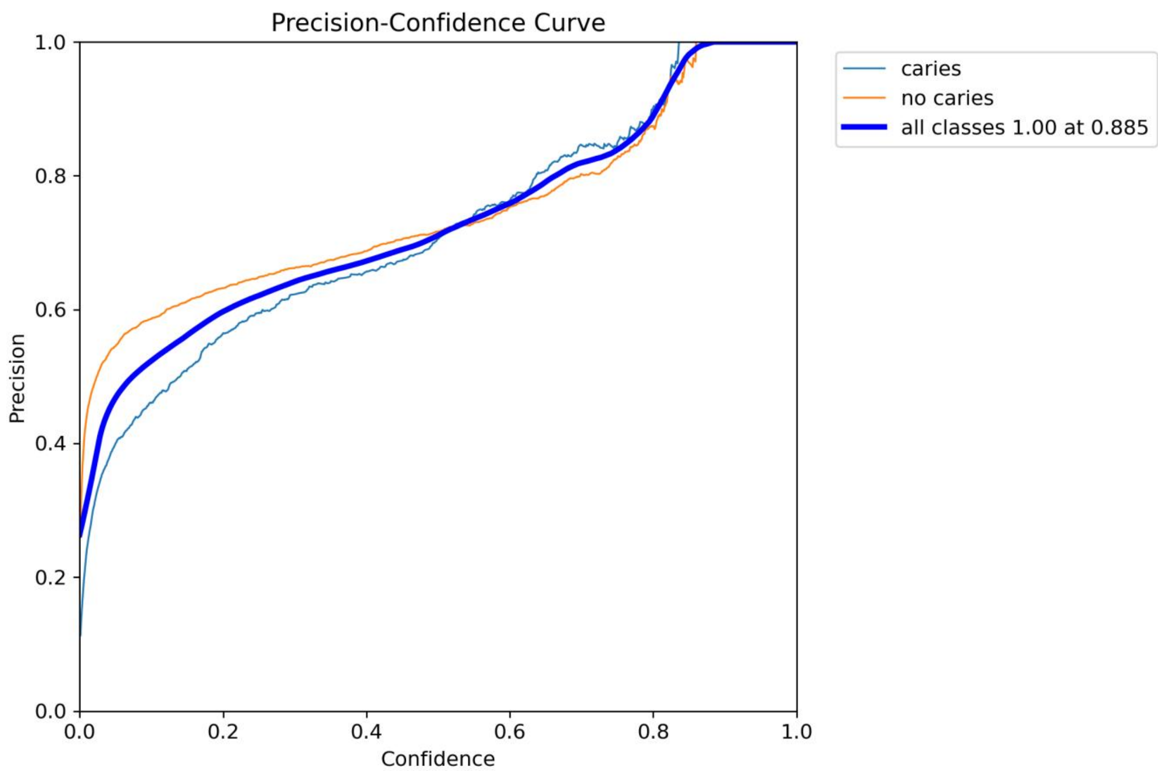


Figure 3.5.5.4: Precision-Confidence Curve of the trained model

The x-axis represents the confidence level, ranging from 0% to 100%. The y-axis represents the precision, which is the percentage of correctly identified caries cases out of all cases the model identified as caries. The blue line shows the overall precision-confidence curve. The dotted lines at 0.2, 0.4, 0.6, and 0.8 confidence levels are likely for reference. The precision for all caries classes (blue line) starts low at 0% confidence and gradually increases to around 80% at 60% confidence. Then, it starts to decrease again, reaching around 60% at 100% confidence. This indicates that the model is more likely to be accurate when it is less confident in its predictions. At higher confidence levels, the model is more likely to make mistakes. The precision for individual caries classes (not shown in the image but mentioned in the text) might differ from the overall precision. The text mentions that for all classes, the precision is 1.00 at 0.885 confidence, which means the model perfectly identified all caries cases at that confidence level. The precision-confidence curve provides insights into the performance of the dental caries detection model at different confidence levels. It helps users understand the trade-off between accuracy and certainty in the model's predictions.

Precision-Recall Curve

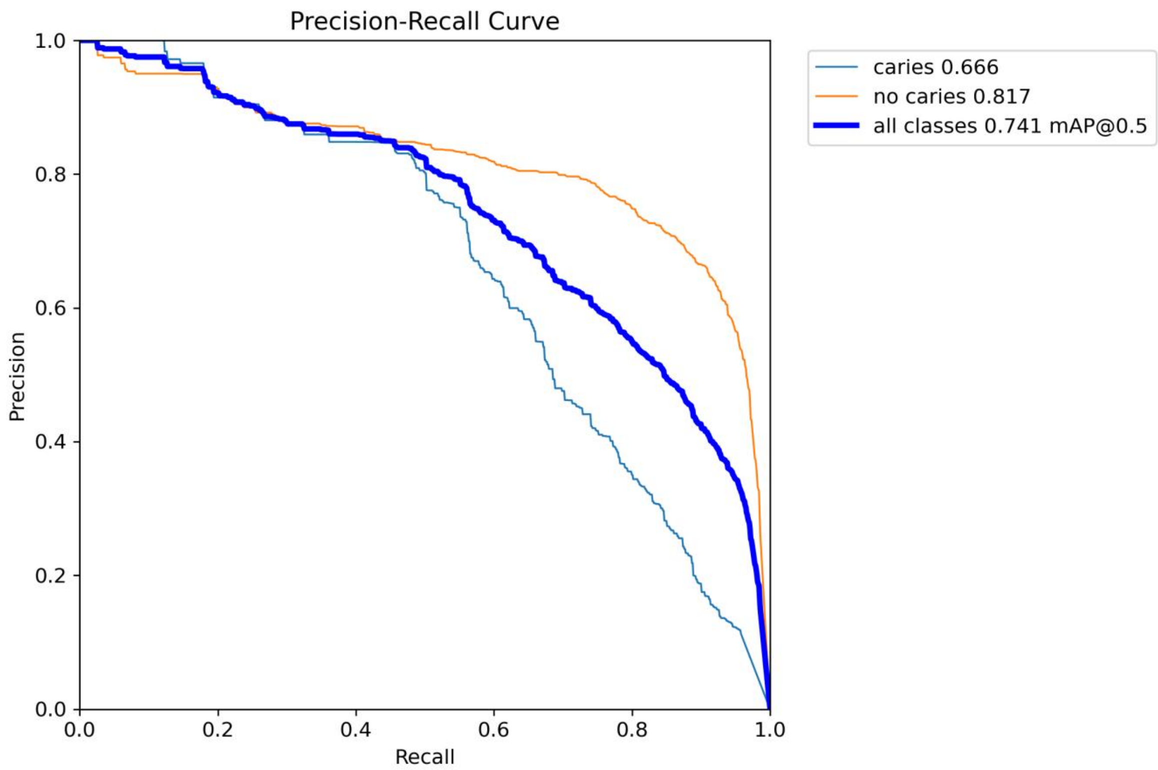


Figure 3.5.5.5: Precision-Recall Curve of the trained model

X-axis: Recall, ranging from 0 to 1. This represents the proportion of actual caries cases that the model correctly identified.
Y-axis: Precision, ranging from 0 to 1. This represents the proportion of cases identified as caries by the model that were actually caries.
Blue line: Precision-recall curve for all caries classes combined.
Green line: Precision-recall curve for a specific caries class (unlabeled in the image).
Orange line: Precision-recall curve for another specific caries class (unlabeled).
Dashed lines: Reference lines at recall values of 0.2, 0.4, 0.6, and 0.8.
The precision-recall curves show the trade-off between correctly identifying caries cases (recall) and avoiding false positives (precision) at different thresholds. The blue curve for all caries classes starts high in precision at low recall, indicating the model is good at identifying some caries with high confidence. As recall increases, precision drops, meaning the model starts including more false positives as it tries to identify all caries cases. The green and orange curves for specific caries classes have different shapes and positions, suggesting the model's performance varies depending on the type of caries. The average precision values for each class provide a summary of the model's performance. For example, the caries class has an average precision of 0.666, meaning on average, 66.6% of the cases the model identified as caries were actually caries. The mAP@0.5 value of 0.741 indicates the average precision across all classes at a recall of 0.5. This means that on average, the model has a precision of 74.1% when it identifies half of the actual caries cases. The precision-recall curve and additional information provide a comprehensive view of the dental caries detection model's performance. By analyzing these curves and metrics, researchers and dentists can understand the model's strengths and weaknesses, and make informed decisions about its potential use in clinical settings.

Recall-Confidence Curve

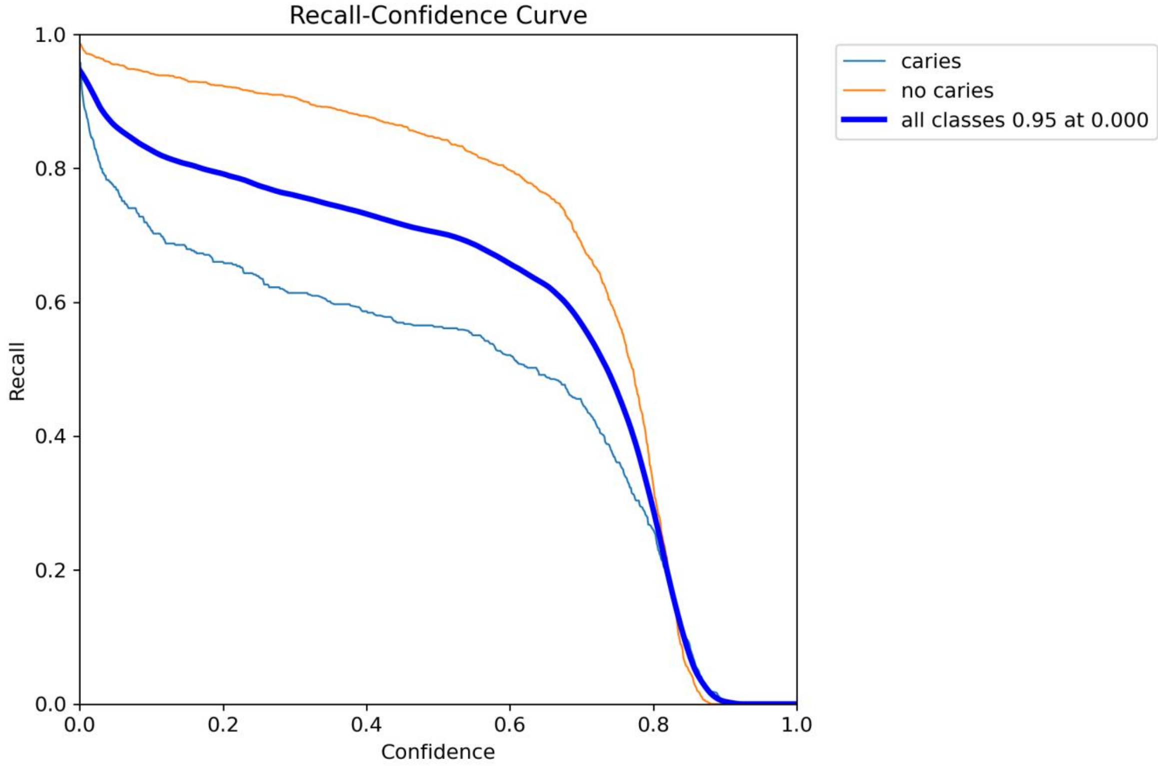


Figure 3.5.5.6: Recall-Confidence Curve of the trained model

Recall-confidence curves are used in machine learning to show the trade-off between the recall and confidence of a model's predictions. Recall is the fraction of positive cases that the model correctly identified, while confidence is the model's certainty in its predictions.
The blue line in the graph shows the recall of the model, which is the fraction of correct positive identifications. The red line shows the confidence of the model, which is the average probability that the model assigns to its positive predictions. The curve shows that as the confidence of the model increases, the recall of the model decreases. This is because the model is more likely to make confident predictions on easy cases, but it is less likely to make confident predictions on hard cases. The point where the two lines intersect is the point where the model has the best balance of recall and confidence. This is the point where the model is most likely to make correct predictions on both easy and hard cases. In the specific case of the graph you sent me, the text at the bottom shows that the model has a recall of 0.95 at a confidence of 0.000. This means that the model is able to correctly identify 95% of the positive cases, but it is only confident in its predictions 0% of the time. This suggests that the

model is very good at identifying positive cases, but it is not very good at making confident predictions. This could be because the model is trained on a dataset that is imbalanced, with many more negative cases than positive cases.

Label Analysis

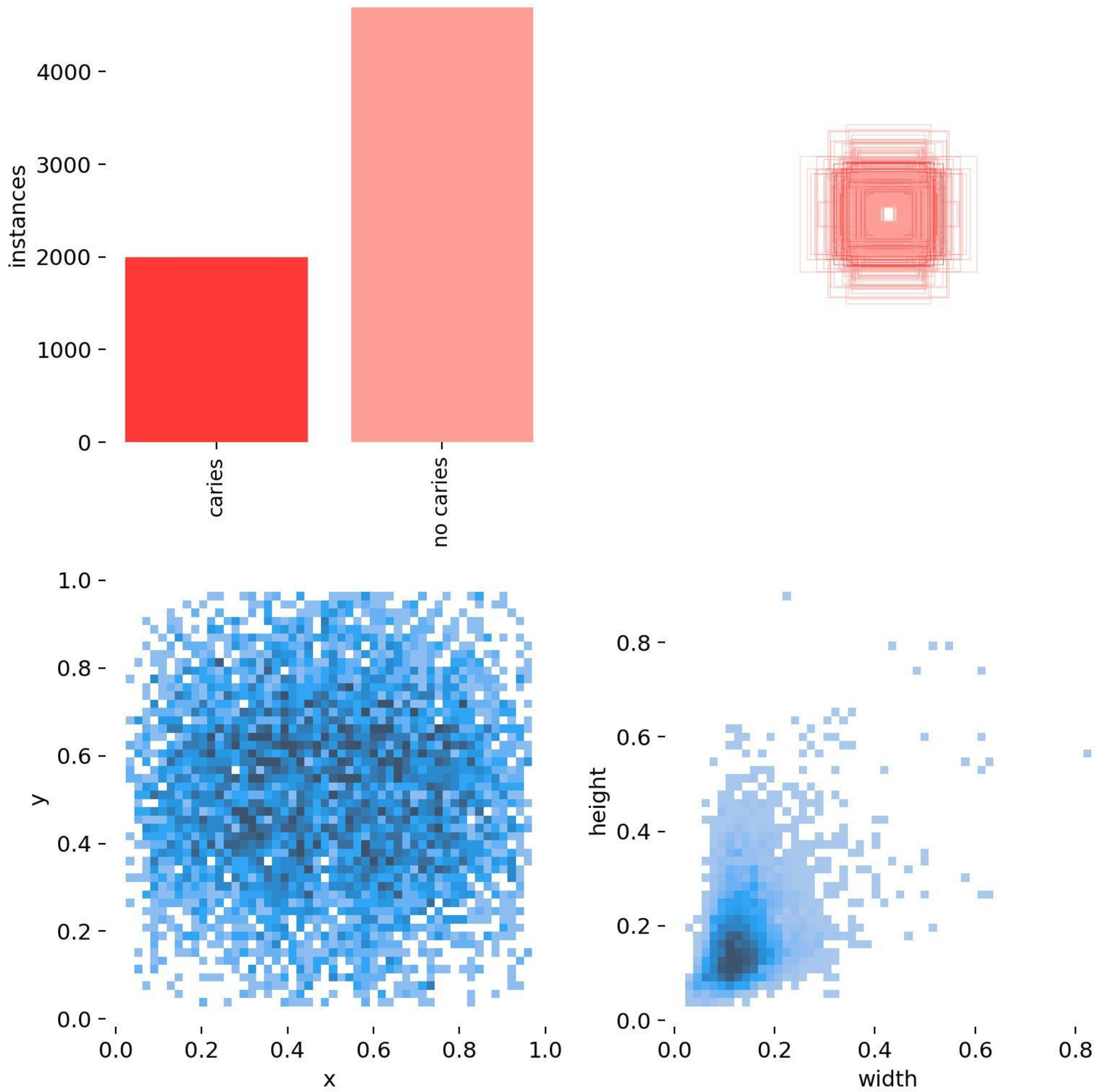


Figure 3.5.5.7: Label Diagram of the trained model

The text on the y-axis of the graph is "instances", and the text on the x-axis is "height". The numbers on the y-axis range from 0 to 4000, and the numbers on the x-axis range from 0 to 1. The blue pixelated sphere is located in the upper right corner of the graph. It is difficult to say for sure how many cells are in the sphere, but it appears to be more than 2000.

Labels Correlogram

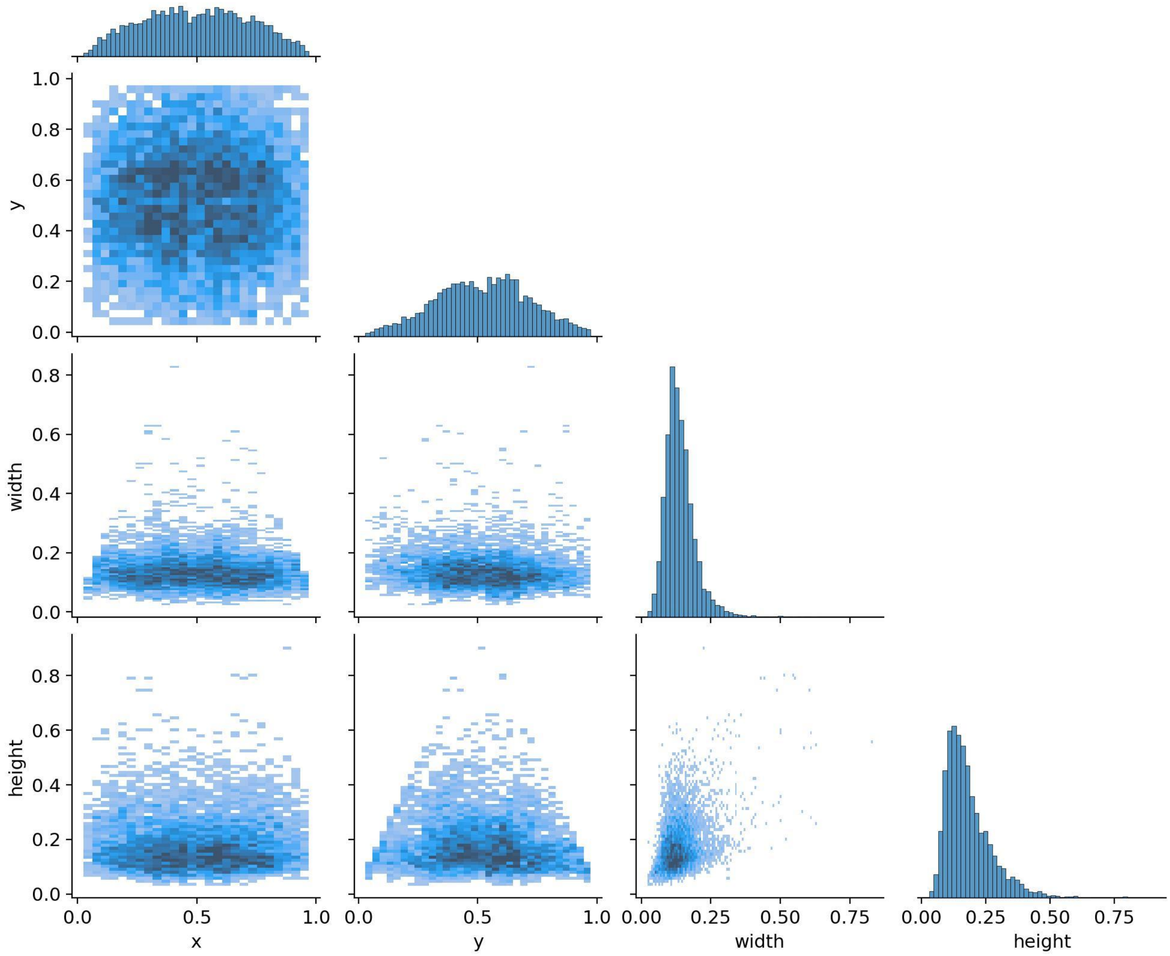


Figure 3.5.5.8: Label Correlogram of the trained model

Heatmap representing the number of cells in a dataset, where the x-axis represents height and the y-axis represents width. The color scale on the right-hand side of the image indicates the number of cells, with blue representing fewer cells and yellow representing more cells.

Training Batch

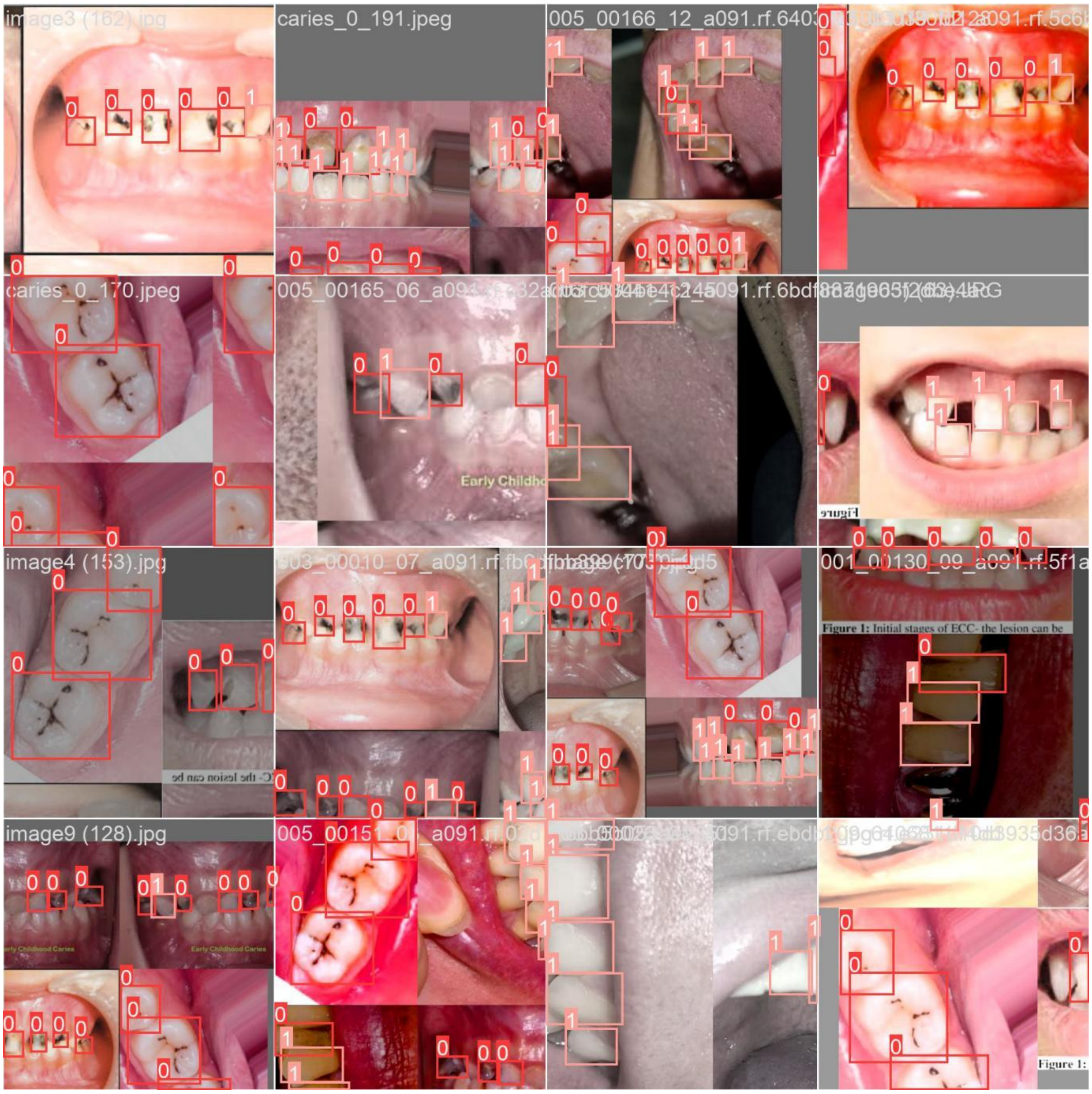


Figure 3.5.5.9: Training Batch 01 of the model

Labels & Predictions

Figure 3.5.5.9: Labels Batch 01 of the model

Figure 3.5.5.10: Prediction Batch 01 of the model

From these two image-sets, we can note that the predicted labels are pretty close and sometimes very confident which are close enough to the actual labels, which means the model is performing well.

3.5.6 Exporting Trained Model

The trained model was exported after every 10 epochs, with the best-performing model saved for deployment.

3.5.7 Implementation of the Model

Input

Figure 3.5.7.1: Inputting a Random teeth Image from the internet (Not present in the train/val set)

Output

Figure 3.5.7.1: Output of the Random teeth Image from the internet (With predicted labels: caries, no-caries)

REFERENCES

The future of dentistry, powered by Dental Ai. (no date) *Pearl*. Available at: https://www.hellopearl.com/ (Accessed: 30 October 2023).

Smart dental app uses AI and IOT for early disease detection (2020) *Innovate*. Available at: https://innovate.ieee.org/innovation-spotlight/smart-dental/ (Accessed: 30 October 2023).

'Caries detection' (2010) *Dental Abstracts*, 55(5), pp. 258–262. doi:10.1016/j.denabs.2010.06.027.

How to Detect Objects in Images Using the YOLOv8 Neural Network. (2023, May 4). FreeCodeCamp.org. https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/

APPENDICES

A. GANTT CHART

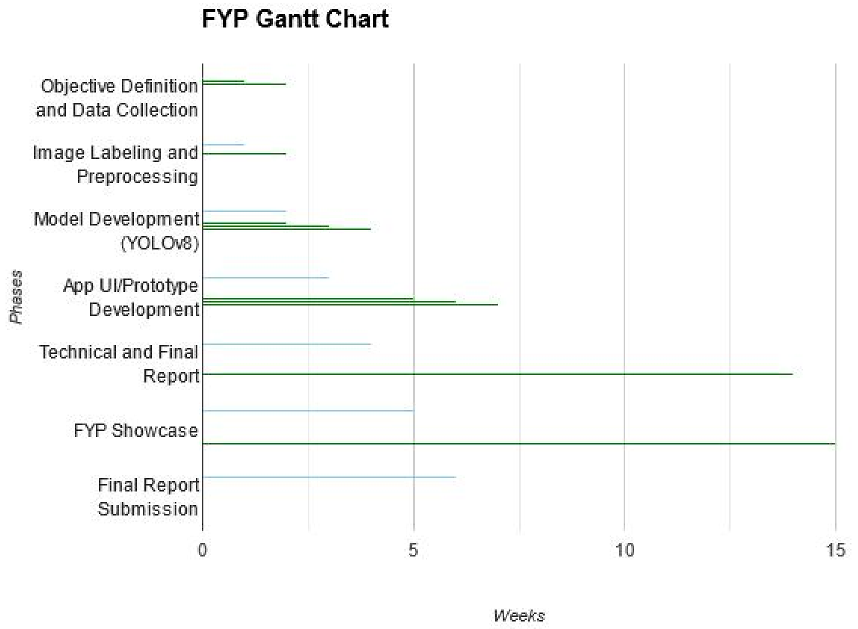


Figure x.0.1: FYP 1 Gantt Chart

B. INTERFACE

Refer to **3.1 Design**.

C. TEST SCRIPT

```
// Import necessary modules and functions for testingconst
{ authenticateUser, uploadImage, detectCaries } = require('./appFunctions');

// Import functions to be tested

// Test suite for user authenticationdescribe('User Authentication', () => {
  test('Valid user login', async () => {
    const user = { email: 'user@example.com', password: 'password123' };
    const isAuthenticated = await authenticateUser(user);
    expect(isAuthenticated).toBe(true);
```

```
  });

  test('Invalid user login', async () => {
    const user = { email: 'invalid@example.com', password: 'invalidpassword' };
    const isAuthenticated = await authenticateUser(user);
    expect(isAuthenticated).toBe(false);
  });

  // Additional test cases for forgot password functionality can be added here
});

// Test suite for image upload functionalitydescribe('Image Upload', () => {
  test('Upload valid image file', async () => {
    const imagePath = '/path/to/valid/image.jpg';
    const isUploaded = await uploadImage(imagePath);
    expect(isUploaded).toBe(true);
  });

  test('Upload invalid image file', async () => {
    const invalidImagePath = '/path/to/invalid/image.txt';
    const isUploaded = await uploadImage(invalidImagePath);
    expect(isUploaded).toBe(false);
  });

  // More test cases for various scenarios can be added for image upload
});

// Test suite for caries detection functionalitydescribe('Caries Detection', ()
=> {
  test('Detect caries in uploaded image', async () => {
    const uploadedImage = '/path/to/uploaded/image.jpg';
    const cariesDetected = await detectCaries(uploadedImage);
    expect(cariesDetected).toBe(true);
  });

  // Additional test cases for different scenarios in caries detection can be
added here
});
```