SONARGAON UNIVERSITY (SU)

Yollov11 Based Deep Learning Framework For Dental Abnormality Detection In Panoramic X-ray Image

Submitted By

Hirock B Biswas	EEE2202026066
Redwone Ahammed	EEE2202026081
Shohanur Rahman	EEE2202026013
Bulet Hossin	EEE2001019168
MD.Raisul Alam Rohan	EEE2202026079

Supervised By

Borhan Uddin Bhuiyan
Lecturer
Electrical and Electronics Engineering
Sonargaon University (SU)

Department of Electrical and Electronic Engineering

Sonargaon University (SU)

147/I, Panthapath, Dhaka-1225, Bangladesh

Date of Submission: January 2026

DECLARATION

We certify that the Research based Thesis titled,**“Yollov11 Based Deep Learning Framework For Dental Abnormality Detection In Panoramic X-ray Image”** is completely our group work. All sources and knowledge for this paper that were found by other researchers are acknowledged by reference. Materials of work such as image, figures, tables and citations in this paper are accepted by our supervisor We hereby declare that this thesis has not been previously submitted either in whole or in part, for any other degree or publication.

…………………………………. ……………………………………

Hirock B Biswas **Redwone Ahammed**

………………………………… ……………………………….

Shohanur Rahman **Bulet Hossin**

………………………………………

MD.Raisul Alam Rohan

CERTIFICATION

This is to certify that this project entitled done by the following students under my direct supervision. This thesis work been carried out by them in the laboratories of the Department **"Yollov11 Based Deep Learning Framework For Dental Abnormality Detection In Panoramic X-ray Image"** of Electrical and Electronic Engineering under the faculty of Engineering, Sonargaon university (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervised By

……………………………………………………………………….

Borhan Uddin Bhuiyan
Lecturer
Electrical and Electronics Engineering
Sonargaon University (SU)

ACKNOWLEDGEMENT

Firstly, we express our sincere thanks and gratitude to the almighty for his divine blessings which made it possible for us to complete this research work successfully.

We would like to express our deepest appreciation to our respected thesis supervisor, Borhan Uddin Bhuiyan, Lecturer, Department of Electrical and Electronic Engineering of Sonargaon University, for his continuous guidance, valuable suggestions, and constructive feedback throughout the entire research period. His support ,encouragement, and academic insights played a crucial role in shaping this work and keeping us motivated.We are also grateful to the honorable faculty members of Sonargaon University for providing us with a strong academic environment and necessary support during our study.

We would like to thank our friends and classmates for their cooperation, discussion, and encouragement, which greatly assisted us during the research and writing process and also thanks to our madam Dr.Nawshin Tabassum BDS(Dhaka University) PGT(oral & maxillofacial surgury-bsmmu) Reg:9615

Finally, we express our heartfelt gratitude to our parents and family members for their constant support, patience, and motivation throughout our academic journey. Without their encouragement and sacrifices, the completion of this thesis would not have been possible.

ABSTRACT

Dental abnormality detection from panoramic X-ray images is a critical task in modern dental diagnostics. Manual detection is often time-consuming and relies heavily on expert knowledge. This research proposes an Yollov11 Based Deep Learning Framework For Dental Abnormality Detection In Panoramic X-ray. Real-world panoramic X-ray images were used, and all experiments were implemented using Python in the PyCharm environment. Multiple deep learning models were trained and integrated to perform object detection, focusing on both accuracy and computational efficiency suitable for practical use.

The proposed framework achieved an object detection accuracy of over 90%, with high precision and recall, demonstrating reliable performance in identifying abnormal dental regions. The results indicate that a multi-model approach can significantly improve detection compared to single-model methods. The main objective of this research is to support dental practitioners by providing an automated tool that reduces manual effort and allows faster, more accurate diagnosis. This framework can help dentists detect dental diseases efficiently, making dental healthcare more effective and accessible.

TABLE OF CONTENTS

CHAPTER-4

HARDWARE IMPLEMENTATION

CHAPTER-5

SOFTWARE IMPLEMENTATION

CHAPTER-6

EXPERIMENTAL RESULTS AND ANALYSIS

CHAPTER-7

CONCLUSION AND FUTURE WORK

LIST OF TABLE

TABLE NO:

LIST OF FIGURE

FIGURE NO:		Page no:
Figure 3.1	Methodology Overview	20
Figure 3.2	(A,B)X-ray (opg image)	23
Figure 3.3	YOLOv11 custom model training workflow	26
Figure 3.4	Case of misclassification of tooth related diseases	28
Figure 3.5	Bounding box for each tooth related disease	29
Figure 4.1	Block diagram	31
Figure 4.2	Hardware layout of raspberry pi board with major components	32
Figure 4.3	Raspberry pi 5	34
Figure 4.4	Raspberry pi 5 hardware setup	35
Figure 5.1	Pycharm project structure	38
Figure 5.2	Label studio project dashboard	39
Figure 5.3	A data export setting interface for an annotation platform	40
Figure 5.4	Panoramic dental X-ray image labeling	41
Figure 6.1	(A,B) Bounding box with OPG X-ray image	45
Figure 6.2	(A,B,C,D) Detection OPG X-Ray Image	48

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Today, although most oral and dental diseases have early diagnosis and treatment opportunities with technological developments in oral and dental health, their global increase cannot be prevented. According to the WHO Global Oral Health Status Report (2022) [1], oral and dental diseases affect approximately 3.5 billion people worldwide. Especially in low- and middle-income countries, there are not adequate services in the field of oral and dental health due to the costs of diagnosis and treatment. As a result of this situation, it is estimated by the WHO that three out of four people in low- and middle income countries are affected by oral and dental diseases [1]. The most common dental diseases, especially dental caries, are periodontal diseases edentulism, oral cancer, dental anomalies, and cleft lip and palate diseases [1]. When efficient diagnosis and treatment are not provided for these diseases, it can cause various complications ranging from mild discomfort to death.[1]
Panoramic dental X-ray imaging is widely used in dentistry to visualize teeth, jaws, and surrounding structures in a single scan. It helps dentists diagnose problems such as caries, impacted teeth, cysts, and other abnormalities. However, manual interpretation of panoramic radiographs is difficult because of overlapping anatomy, image noise, and the subtle appearance of many dental conditions. As a result, diagnosis can become time-consuming and dependent on the clinician’s experience [4]. Artificial intelligence has recently become an effective solution for automated dental image analysis. AI-based systems improve consistency and reduce workload by providing fast and objective interpretation of radiographs. Putra et al. reported that deep learning techniques significantly support dental diagnosis and early detection in digital radiography [4]. In particular, object detection models are useful because they can both locate and classify dental structures and abnormalities.

YOLO-based frameworks have shown promising performance in panoramic X-ray analysis. Mendes et al. demonstrated accurate tooth detection and numbering using YOLO on panoramic images [3]. Chen et al. used Faster R-CNN to detect various dental conditions, confirming the effectiveness of deep learning, though with higher computational cost [5]. Furthermore, Singh et al. enhanced YOLO with Transformer attention and genetic optimization to improve dental anomaly and tooth type classification [2].

This thesis proposes a YOLOv11-based deep learning framework for dental abnormality detection in panoramic X-ray images. The goal is to achieve fast, accurate, and automated diagnosis to assist dentists, reduce variability, and enable practical deployment in real clinical environments.

1.2 Motivation

Accurate and timely detection of dental abnormalities from panoramic X-ray images is essential for effective dental treatment. Although deep learning models have shown strong performance in medical image analysis, most existing approaches depend on cloud-based computation, which introduces latency, internet dependency, and data privacy concerns.Edge computing offers a practical solution by enabling on-device inference with reduced response time and improved data security. This research is motivated by the need to develop an edge-optimized deep learning framework that can perform reliable dental abnormality detection using real X-ray data.In this study, the YOLOv11 object detection model is implemented using Python in the PyCharm environment and deployed on Raspberry Pi 5 (8GB RAM). The goal is to achieve high detection accuracy while ensuring real-time performance on resource-constrained edge hardware.Recent research shows that deep learning can effectively support dental diagnosis. Mahdi et al. improved tooth recognition by combining optimization techniques with deep learning [6]. Tassoker et al. demonstrated accurate detection and localization of subtle conditions such as idiopathic osteosclerosis using AI models [7]. Beser et al. applied YOLOv5 for fast tooth detection and segmentation in pediatric panoramic images [8], while Jiang et al. used a two-stage deep learning architecture to stage periodontal bone loss from radiographs [9].

Motivated by these works, this thesis proposes a YOLOv11-based framework for dental abnormality detection in panoramic X-rays to achieve fast, accurate, and automated diagnosis that can assist clinicians and reduce manual effort in real clinical environments.

1.3 Objectives

The primary objectives of this research are as follows:

- To develop an automated system for dental abnormality detection using panoramic X-ray images.
- To utilize real dental X-ray datasets for model training and evaluation.
- To implement the YOLOv11 object detection model for accurate localization and classification of dental abnormalities.
- To design a multi-model deep learning framework to enhance robustness and detection performance.
- To optimize the proposed framework for deployment on Raspberry Pi 5 (8GB).
- To evaluate system performance using standard metrics such as accuracy, precision, recall, F1-score, and mean Average Precision (mAP).
- To analyze inference time, memory consumption, and edge-level computational efficiency

1.4 Problem Statement

Panoramic dental X-ray images are widely used for diagnosing dental problems, but analyzing these images manually is often difficult and time-consuming. The images may contain noise, low contrast, and overlapping teeth, which can lead to errors in diagnosis. While deep learning models can improve detection accuracy, many existing solutions rely on cloud-based systems and powerful computers. This increases delay, depends on internet connectivity, and raises concerns about patient data privacy. As a result, using such systems in real clinical environments becomes challenging. Therefore, there is a need for a simple and efficient edge-based deep learning system that can accurately detect dental abnormalities in real time using limited hardware resources such as Raspberry Pi.

First, most existing works perform well only in controlled environments. Dental and medical image-based studies mainly use limited and clean datasets [10], [11], [30]. Because of this, models work well in experiments but struggle in real situations with noise, low contrast, overlapping objects, and different image qualities. This shows a gap in building models that can generalize well in real-world conditions.

Second, many studies try to deploy YOLOv11 on edge devices such as Raspberry Pi, UAVs, FPGA, and embedded systems [12], [18], [19], [26], [38]. Although real-time detection is demonstrated, there is still a trade-off between speed and accuracy. Lightweight models increase FPS but reduce detection quality, while accurate models need higher computation. Hence, there is a gap in achieving both high accuracy and real-time performance on low-cost hardware.

Third, detecting small, dense, and overlapping objects remains challenging. Remote sensing and UAV-based works show that YOLOv11 often misses small targets or produces false detections when objects are crowded or appear at different scales [13], [28], [31], [39]. This indicates a need for better multi-scale feature learning and robustness in complex scenes.

Fourth, several papers improve YOLOv11 using optimization techniques, hybrid models, or additional modules [11], [17], [41]. These improve accuracy but increase model complexity and training cost. Complex architectures also require stronger hardware, which limits practical deployment. Therefore, there is a research gap in designing simple but efficient models suitable for real applications.

Fifth, many applications in agriculture and monitoring show high accuracy in limited environments but lose stability under changing light, background, and seasonal conditions [16], [21], [24], [33], [35]. This highlights a gap in maintaining consistent detection performance under diverse real-world variations.

Sixth, traffic and surveillance systems report good results on benchmark datasets, but performance drops in crowded, nighttime, rainy, or motion-blurred scenes [25], [27], [32], [34], [36], [37]. This reveals a gap in robustness for continuous real-world operation.

Finally, although YOLOv11 is integrated with smart and IoT-based systems [15], [25], [29], most works focus mainly on detection accuracy and do not fully address system-level deployment, adaptability, and cross-domain performance [14], [20], [23], [40], [42]. Models trained in one environment often fail in another, showing limited generalization ability.

Therefore, a clear research gap exists in developing a fast, accurate, lightweight, and generalizable YOLOv11-based detection framework that works reliably in real-world environments, handles small and overlapping objects, and runs efficiently on edge devices.

1.5 Scope of the Research

This research focuses on developing a deep learning framework using YOLOv11 to detect dental abnormalities from panoramic X-ray images. Panoramic radiographs are widely used in dental diagnosis, but manual inspection is time-consuming and depends heavily on the experience of the dentist. Therefore, an automatic and reliable detection system can help improve screening and decision support.The scope of this study includes collecting and preparing panoramic dental X-ray images for training and testing the model. Image preprocessing steps such as resizing, contrast enhancement, noise removal, and data augmentation are applied to improve image quality and handle variations in lighting, blur, and patient anatomy.This research also covers training and fine-tuning the YOLOv11 model to recognize different types of dental abnormalities. The goal is to achieve good detection accuracy while keeping the model lightweight and efficient. The framework is designed so that it can work not only on high-end computers but also on low-cost devices.

Another part of the scope is performance evaluation. The proposed system is tested using common metrics such as accuracy, precision, recall, and processing speed. Special attention is given to difficult cases in panoramic images, such as overlapping teeth, small lesions, low contrast areas, and background noise.The study further includes deploying the trained model on an edge platform such as Raspberry Pi to check real-time feasibility. This allows evaluation of speed, memory usage, and power consumption, which are important for practical use in clinics or portable systems.

In addition, this research considers how the system can be integrated into a simple dental screening workflow to assist dentists. It is not intended to replace professional diagnosis, but to support dental experts by providing fast and consistent abnormality detection.

Overall, the scope of this research is limited to designing, optimizing, testing, and deploying a YOLOv11-based framework for dental abnormality detection from panoramic X-ray images, focusing on accuracy, efficiency, and real-world usability.

1.6 Outline of the Thesis

This thesis presents a deep learning framework based on YOLOv11 for detecting dental abnormalities from panoramic X-ray images. The work is organized into seven chapters, each describing a specific part of the research process.

Chapter 1: Introduction

This chapter gives an overall idea of the research. It explains the background and importance of automatic dental abnormality detection. The motivation behind choosing this topic is discussed, followed by the research objectives and the main problem statement. Finally, the scope of the research is described to clarify the boundaries of the study.

Chapter 2: Literature Review

This chapter reviews existing systems and related research works in dental image analysis and object detection. It discusses how previous methods work, their strengths, and their limitations. The analysis helps to understand current trends and identify gaps that motivate the proposed framework.

Chapter 3: Methodology

This chapter describes the proposed approach in detail. It starts with an overview of the framework, then explains the dataset collection process and data preprocessing techniques. The YOLOv11 model used in this study is introduced, and the optimization strategies applied to improve detection accuracy and efficiency are discussed.

Chapter 4: Hardware Implementation

This chapter focuses on the hardware part of the system. It presents the block diagram and describes the main hardware components. The use of Raspberry Pi 5 is explained along with the setup process needed to run the detection model on the device.

Chapter 5: Software Implementation

This chapter explains the software side of the project. It introduces the software environment and tools used for development. The installation and configuration steps are discussed to show how the system is implemented and executed in practice.

Chapter 6: Experimental Results and Analysis

This chapter presents the experimental setup and obtained results. It includes performance evaluation, result analysis, and discussion of the findings. The effectiveness of the proposed framework is analyzed using different metrics and real test cases.

Chapter 7: Conclusion and Future Work

This final chapter summarizes the overall research and main contributions. It concludes the study and suggests possible future improvements and extensions of the system for further research.

CHAPTER 2

LITERATURE REVIEW

2.1 Existing System Analysis

Dental abnormality detection using panoramic X-ray images is commonly performed through manual examination in most dental clinics. In existing systems, dentists visually inspect X-ray images to identify problems such as cavities, impacted teeth, bone loss, and jaw-related abnormalities. This process mainly depends on the experience and skill of the dentist [35]. Although manual diagnosis is widely practiced, it has several limitations. It is time-consuming and can be affected by human error, especially when X-ray images are complex or unclear. In many cases, small or early-stage abnormalities may be missed. Moreover, different dentists may interpret the same X-ray image differently, which can lead to inconsistent diagnosis.To reduce these problems, computer-aided diagnosis (CAD) systems were introduced [38-40]. These systems use basic image processing techniques like noise removal, contrast enhancement, and segmentation to highlight important regions in X-ray images. After that, traditional machine learning algorithms are applied for classification. However, these systems depend heavily on manually designed features and often fail to perform well on real-world panoramic dental X-ray images.In recent years, deep learning–based systems have become popular for dental abnormality detection. Convolutional Neural Networks (CNNs) automatically learn features from X-ray images and provide better accuracy than traditional methods[41-42]. However, most existing deep learning systems require powerful computers and GPUs. They are mainly designed for cloud or desktop environments and are not optimized for low-cost edge devices. This makes them difficult to use in portable or real-time clinical applications.

2.2 Literature Review

Several research studies have explored the use of deep learning techniques for dental X-ray image analysis. Early research focused on classifying dental images as normal or abnormal using CNN-based models. These approaches improved accuracy but did not provide information about the exact location of abnormalities. Later studies introduced segmentation and object detection methods to locate dental problems in X-ray images. Models such as U-Net were used for segmentation, while Faster R-CNN and SSD were applied for detection tasks. Although these methods showed good performance, they were computationally expensive and slow during inference. More recent studies have adopted YOLO-based object detection models because of their fast processing speed and simple architecture. Different versions of YOLO have been successfully used for detecting dental caries, missing teeth, and impacted teeth in panoramic X-ray images. However, most of these studies were conducted using high-end GPUs and did not focus on edge device deployment. Only a limited number of studies have attempted to run deep learning models on edge devices such as Raspberry Pi. These studies often face challenges related to memory limitations, slow inference speed, and reduced accuracy. Furthermore, most existing research uses publicly available datasets, which may not fully represent real clinical data.Based on the literature, there is a clear research gap in developing an efficient, accurate, and edge-optimized multi-model framework for dental abnormality detection using real panoramic X-ray images. This thesis addresses this gap by implementing a YOLOv11-based solution on Raspberry Pi 5 using real-world dental X-ray data.

CHAPTER 3
METHODOLOGY

3.1 Overview

This chapter outlines the methodology used to develop an edge-optimized multi-model deep learning framework for detecting dental abnormalities from panoramic X-ray images. The proposed approach focuses on achieving high accuracy while ensuring efficient performance on edge devices such as the Raspberry Pi 5 (8GB RAM).The study utilizes real-world panoramic dental X-ray images, which are preprocessed through resizing and normalization to improve data quality and consistency. For abnormality detection, the YOLOv11 object detection model is employed due to its real-time inference capability and lightweight architecture. Model development and training are conducted using Python in the PyCharm IDE.To enhance edge deployment performance, various optimization strategies are applied to balance detection accuracy and computational efficiency. Finally, the optimized model is deployed and evaluated on the Raspberry Pi 5, where performance is measured in terms of accuracy, inference time, and resource utilization. This methodology ensures practical applicability for real-time dental diagnostic systems.

In this Figure 3.1,This figure explains how an intelligent dental detection system works, step by step, from a simple X-ray image to real-time results on hardware.The process begins with a custom OPG (panoramic dental X-ray) image. This image shows the full view of a patient’s teeth, jaws, roots, and surrounding bone. Dentists usually use this type of X-ray to understand overall dental conditions. At this stage, the image is only a medical scan and has not been analyzed by a computer.

In the next step, the image goes through preprocessing and labeling. Here, the image quality is improved so that important dental details become clearer. After that, dental experts manually mark different conditions on the image, such as missing teeth, crowns, implants, and root canal treated teeth. These markings help the system learn what each dental problem looks like.

Then, the prepared image is sent into the YOLOv11 deep learning model. This model analyzes the image and learns patterns from the labeled data. It looks at the image in different scales so it can detect both small details and larger dental areas. Using this process, the model learns how to locate and identify various dental conditions accurately.

After processing, the system produces a detection OPG X-ray image. In this result image, dental problems are shown using bounding boxes and labels. Each label also includes a confidence score, which shows how sure the system is about its prediction. This makes it easier for dentists to quickly understand the patient's dental condition.

Finally, the trained model is implemented on Raspberry Pi 5 hardware. This allows the system to work directly on a small and low-cost device without needing cloud support. As a result, dental abnormality detection can be done in real time, even in clinics with limited resources.

Overall, this image shows a complete workflow, starting from a dental X-ray image and ending with real-time detection on embedded hardware. The system is simple, practical, and helpful for faster and more efficient dental diagnosis.

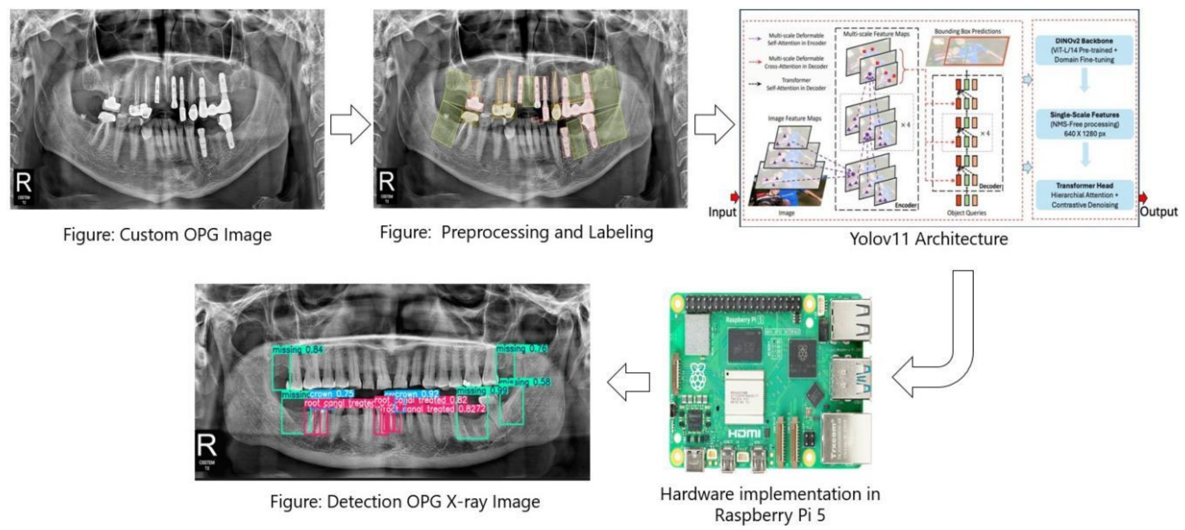

Figure 3.1 : Methodology Overview

Precision and recall

Precision: The model's ability to identify only relevant objects. It represents the percentage of correct positive predictions.

$$precision = \frac{TP}{TP + FP} = \frac{TP}{all \det ection}$$

Recall: The model's ability to find all relevant instances. It represents the percentage of true positives among all ground truths.[10]

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} = \frac{TP}{all \quad ground \quad truths}$$

Mean Average precision(mAP)

Average precision (AP), derived from precision and recall for each object category, serves as the primary metric for evaluating object detection performance. To provide a comprehensive assessment, the average AP (mAP) combines these metrics across all categories [10]. mAP50 refers to the average precision calculated with an IoU threshold of 0.5, while mAP50-95 represents an average of the AP scores computed across multiple IoU thresholds from 0.5 to 0.95, incremented by 0.05 [10-11]

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AF$$

F1 Score

The F1-score is a measurement of a testing data's accuracy and is the harmonic mean of the precision and recall values. The equation for F1-score is given below.[11]

$$F1 \quad Score = \frac{2. \quad \mathrm{Re}\,call. \quad \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision}$$

Average Precision

Average precision is a metric that measures the precision of a model at various recall levels, providing an aggregate measure of performance across all thresholds. Average Precision (AP) is used to calculate the average precision value for recall value over 0 to 1.

$$AP = \int_0^1 p(r)dr$$

3.2 Dataset Collection

Real dental X-r y images were collected from dental clinics with appropriate a

permissions, ensuring diversity in age, gender, and dental conditions. This study used panoramic images provided by 30 dental clinics were collected, anonymized and used for this study. Among the anonymized genders, there were 80 males and 70 females, with a total of 235 unidentified persons who could not be identified. Population distribution by age group did not include teenagers; there were 20 persons in their 20s, 20 persons in their 30s, 30 persons in their 40s, 40 persons in their 50s, 80persons in their 60s and older, and 45 persons with unknown identities. The dataset includes common dental abnormalities such as caries, bone loss, root canal treatments, and prosthetic restorations, broken crown , crown (artificial) prosthesis , missing tooth , implant , periapical lesion , restoration , buccal angulation with partially erupted , metallic crown , Gross Carries , impacted tooth ,Attrition , Horizontal Partially erupted , Proximal Carries , Moderate bone loss , Broken down root , Faulty restoration , bone resorption , Horizontal impaction, Vertical impaction , mesio angular eruption , BDR , Partial erupted , Artificial crown bridge ,vertical impacted tooth , A.Bridge , Supernumerary tooth (Mesiodens) , malocclusion jaw , lesion between bifurcation of root, diastema , Bone loss. Each image was carefully annotated to indicate the location and type of abnormality. Using real data ensures the system can handle real-world variations, such as noise, overlapping teeth which are often present in clinical images.

Figure-A

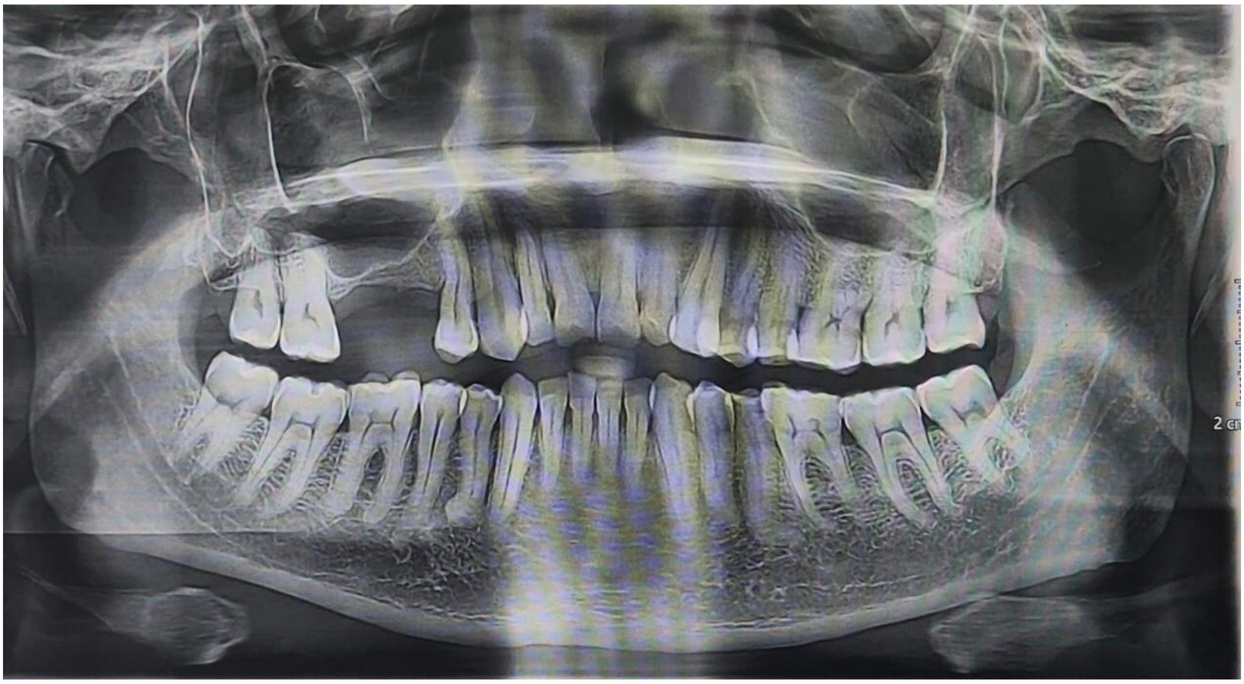

Figure-B

Figure 3.2**:** (A ,B) X-ray (OPG Image)

In Figure 3.2 ,When a patient visits the dental clinic for a full checkup, the dentist often starts with a panoramic X-ray. This special image captures the entire mouth in a single frame, showing both the upper and lower jaws together.

In this picture, we can see how the patient's natural teeth and jawbones are supported by modern dental treatment. The bright white parts stand out immediately. These are metal bridges and implant-supported crowns placed to replace missing teeth. Because metal blocks X-rays, it appears much whiter than normal teeth in the image.Looking closely at the lower jaw, several implant screws are fixed into the bone, holding artificial teeth firmly in place. On the upper jaw, a long bridge connects multiple restored teeth, helping the patient chew and smile with confidence again.

This panoramic view helps the dentist understand the full story of the mouth — where teeth were lost, how they were replaced, and how healthy the surrounding bone remains. It is not just a picture, but a record of treatment, recovery, and care that keeps the patient's oral health on track**.**

3.3 Data Preprocessing

Data preprocessing is a fundamental stage in developing an efficient deep learning–based dental abnormality detection system. In this study, preprocessing was carefully designed to ensure compatibility with the YOLOv11 framework and optimized performance on edge devices. Initially, the YOLOv11 model was downloaded and installed using the Ultralytics framework within an Anaconda virtual environment. This setup ensured a stable development environment and proper dependency management. All preprocessing and training procedures were executed using PyCharm, which facilitated organized coding and project management. The collected real-world panoramic dental X-ray images were first structured into a standardized dataset format. Each image was manually inspected to eliminate noisy, duplicated, or low-resolution samples. This step ensured that only high-quality images were included for training and validation.Next, image annotation was performed using Label Studio, where dental abnormalities were marked with bounding boxes. After labeling, the dataset was exported in YOLO format, generating corresponding .txt label files for each image. Each label file contained class IDs and normalized bounding box coordinates, making it fully compatible with YOLOv11. To enhance model generalization, data augmentation and regularization techniques were applied. These included rotation, scaling, horizontal flipping, and brightness adjustment. Augmentation helped simulate real clinical variations and reduced the risk of overfitting, especially when working with limited medical imaging data. The processed dataset was then divided into two subsets: training and validation. A custom configuration file named dataset_custom.yaml was created, specifying dataset paths, class names, and dataset structure. This YAML file played a critical role in linking the customized dataset with the YOLOv11 training pipeline.After completing preprocessing, the dataset was ready for model training. Training parameters such as batch size, number of epochs, input image size, workers, and device selection were configured according to the hardware constraints of the Raspberry Pi 5 (8GB). Finally, the model was trained and tested on the customized dataset, enabling effective learning and evaluation of dental abnormalities in panoramic X-ray images.

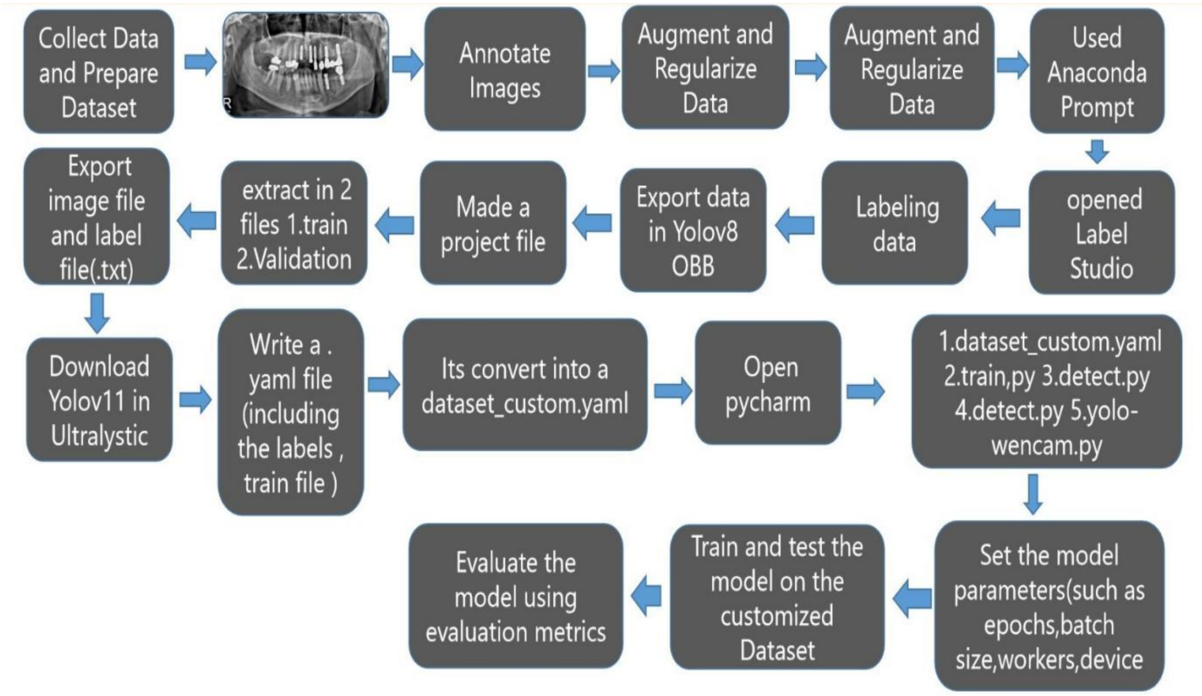

Fig-3.3: YOLOv11 Custom Model Training Workflow

In this figure 3.2,First, dental X-ray images are collected and properly prepared. These images are then annotated using Label Studio, where the abnormal regions are labeled manually. To improve model performance and avoid overfitting, data augmentation and regularization techniques are applied. After labeling, the dataset is split into training and validation sets and exported in YOLO format along with image and label text files.

Next, YOLOv11 is downloaded from Ultralytics, and a custom YAML configuration file is created to define dataset paths and class labels. The project is opened in PyCharm, where training, detection, and webcam testing scripts are used. Model parameters such as epochs, batch size, device, and workers are configured before training. Finally, the trained model is tested on the customized dataset, and its performance is evaluated using standard evaluation metrics.

3.4 Model Used in This Study

The YOLOv11 object detection model was increased accuracy compared to the existing object detection algorithms because it extracts the image features and minimizes the noise image analysis. The YOLOv11 consists of a convolution feature map and a region of interest feature vector [3]. The convolution feature map delivers the image to the convolution and max-pooling layers, and the received information is placed as a feature in the region of interest.There after, the feature vector map is converted into a map with various features, and the object value of the object image of class K is determined by moving to the fully connected layers[3]. In this process, multiple work losses are minimized, and the learning accuracy is improved by using a loss function. Learning multiple classes of tooth-related diseases in YOLOv11 model sometimes results in errors in the detection of panoramic images with dark areas, as shown in Figure 3.3.

Therefore, this study applies a single class to YOLOv11 Model instead of multiple classes to improve the accuracy of detecting tooth-related diseases.[3] For image reading, a rectangular bounding box was first used,and segmentation was performed through an algorithm based on about 500 segmentation data. In the case of segmentation, accuracy was not calculated for the segmented data because it was used only for grasping the approximate accuracy. There after, the coordinate values of the box-type tooth classes that are multilabeled in 1 tooth panoramic image were derived. Each disease corresponding to the derived coordinate value was classified by class. Then, each of the 5 tooth classes was applied to learning through the box coordinate values having the corresponding dental disease on the panoramic image. Through this, the input value for 1 model was constructed using the panoramic image data of 1 class and the box coordinate values corresponding to dental diseases. As shown in Figure 3.4, a bounding box was designated for each tooth-related disease, and the classes for each tooth-related disease were defined.

In this Figure 3.3, panoramic dental X-ray (OPG), labeling has been done using colored boxes and lines to highlight important dental findings. The red boxes indicate individual teeth or areas that need attention, such as possible decay, alignment issues, or regions under evaluation. The yellow boxes and lines mark teeth that have received dental treatment, including crowns, fillings, or root canal therapy; the bright linear structures inside some teeth represent root canal fillings or posts. The green box highlights a specific tooth or bone area for closer observation. This type of labeling helps clinicians and students easily identify treated teeth, problematic areas, and overall dental condition. The letter “R” on the image indicates the patient’s right side.

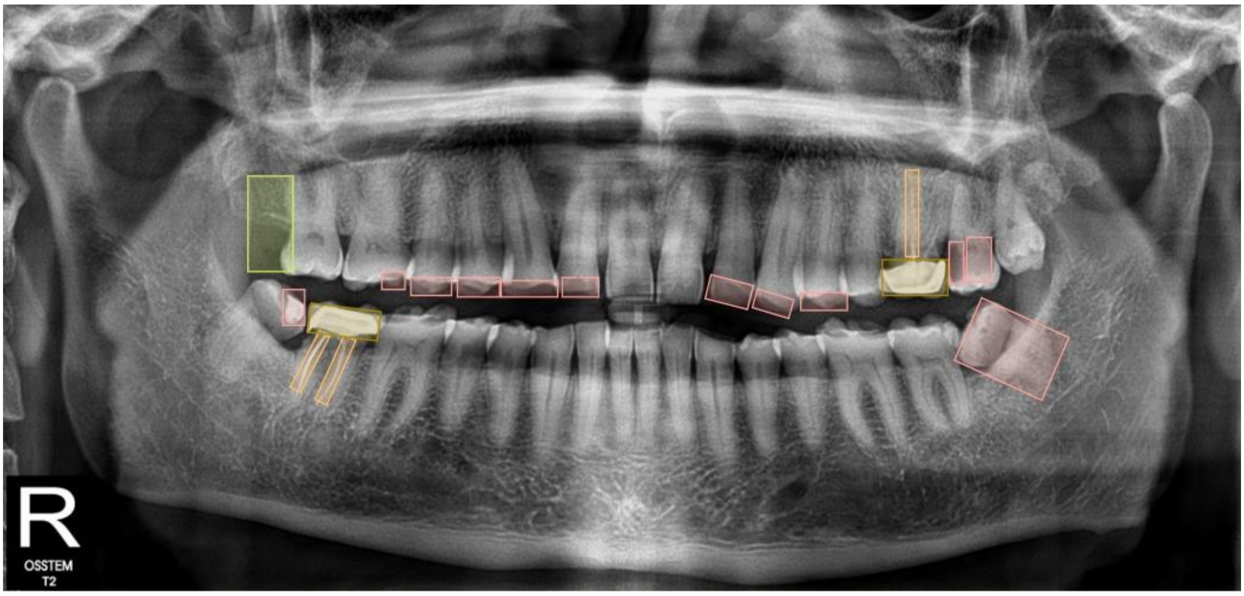

Figure 3.4: Case of misclassification of tooth-related diseases. The green boxes represent detected areas that are not included in the teeth.

In this Figure 3.4, The picture shows how a panoramic dental X-ray is used to detect different dental problems. On the left side, there is a full panoramic image of the patient's teeth and jaws, which provides an overall view of the oral condition. From this panoramic image, specific areas are selected and enlarged on the right side for closer examination. These highlighted images reveal different types of dental findings, such as proximal caries between adjacent teeth, coronal caries or defects on the chewing surface, periapical radiolucency indicating infection near the root tip, cervical caries or abrasion near the gum line, and residual roots left after tooth damage or extraction. By marking these areas with boxes, the image clearly demonstrates how various dental diseases can be identified and analyzed from a single panoramic radiograph.

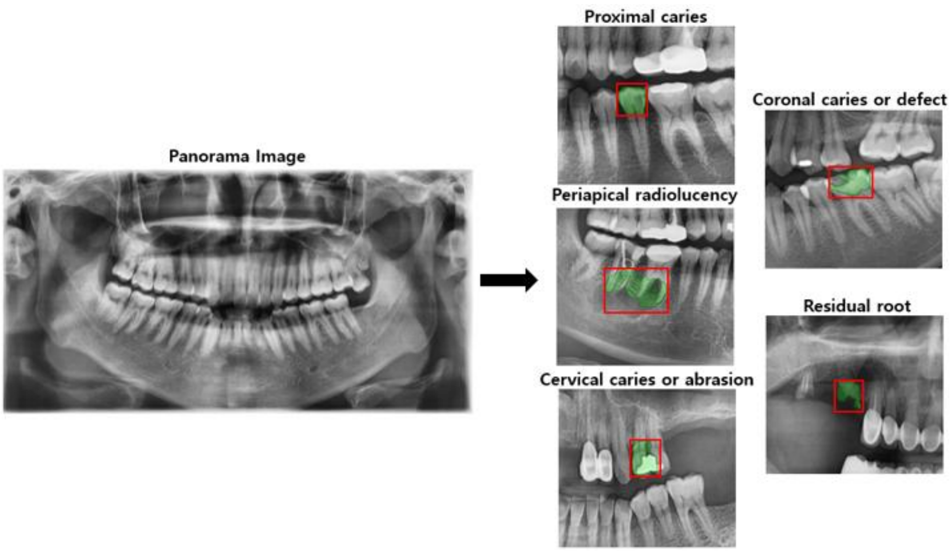

Figure 3.5: Bounding box for each tooth-related disease.

3.5 Yollov11 Model Optimization

This project presents a deep learning–based framework for automated dental abnormality detection using Orthopantomogram (OPG) X-ray images. The system is built on an optimized YOLOv11 object detection model, chosen for its high accuracy and real-time detection capability.The workflow begins with dataset preparation and annotation. Dental OPG X-ray images are labeled using Label Studio, which is launched through the Anaconda Prompt environment. During the annotation process, dental abnormalities are carefully marked to ensure accurate ground-truth data. After labeling, the annotated dataset is exported in YOLOv8 OBB (Oriented Bounding Box) format, which provides precise object localization suitable for complex dental structures.The exported dataset is then processed and organized into two main components: images and labels. These components are further divided into training and validation datasets to support effective model learning and performance evaluation. Specifically, the directory structure includes:

Train folder containing separate images and labels folders.Validation folder containing separate images and labels folder.This structured dataset organization ensures compatibility with the YOLO training pipeline and improves model training efficiency.

Next, the YOLOv11 model is downloaded from the Ultralytics platform and integrated into the development environment using PyCharm. A custom configuration file, data_custom.yaml, is created to define dataset paths, class names, and dataset splits. This file serves as a key link between the dataset and the YOLOv11 training framework.

Finally, all components are executed within PyCharm to train and validate the model. The optimized YOLOv11 framework enables efficient learning from annotated dental X-ray images, facilitating accurate detection of dental abnormalities. This system demonstrates the potential of deep learning to support clinical diagnosis and improve decision-making in dental imaging.

CHAPER 4

HARDWARE IMPLEMENTATION

4.1 Block Diagram Explain

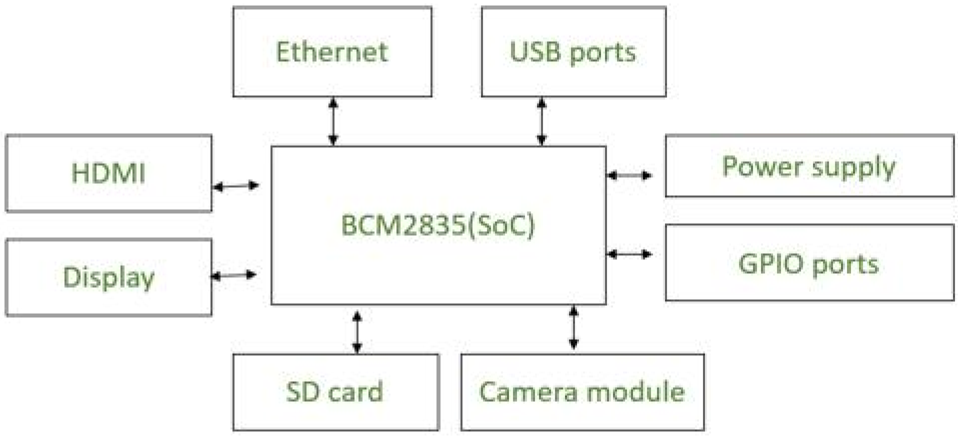

Figure 4.1: Block Diagram

The Figure 4.1 shows the block diagram of the Raspberry Pi architecture, where the BCM2835 System-on-Chip (SoC) acts as the central processing unit. It manages all internal and external operations of the system. The HDMI and display interfaces are used for visual output, while USB ports and Ethernet enable peripheral connectivity and network communication. The camera module connects directly to the SoC for image and video acquisition. Data storage and operating system files are handled through the SD card. The GPIO ports allow interaction with external sensors and devices, and the power supply provides stable electrical power for reliable system operation.

4.3 Hardware Components

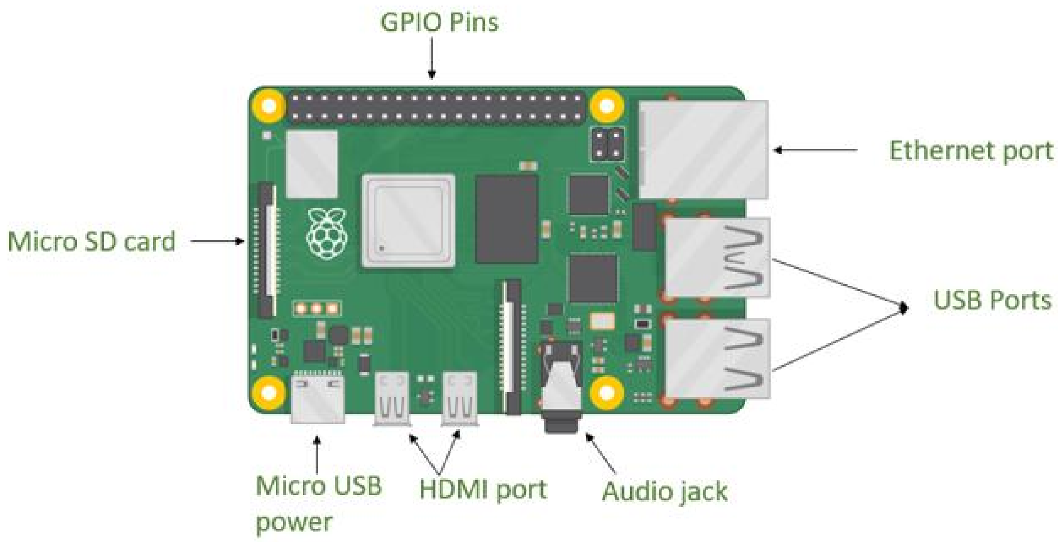

Figure 4.2: Hardware Layout of Raspberry Pi Board with Major Components

Figure 4.2 presents the basic hardware layout of the Raspberry Pi board with its major components. The GPIO pins located at the top enable communication with external sensors and actuators, allowing the system to interact with the physical environment. The Micro SD card slot is used as the main storage unit where the operating system and application files are stored and from which the board boots.

Power is supplied through the Micro USB port, ensuring stable operation of the device. The HDMI port provides video output for connecting a monitor and visualizing system processes and results. An audio jack supports sound output through external speakers or headphones.

The USB ports allow connection of peripherals such as a keyboard, mouse, camera, and storage devices, extending the board's functionality. The Ethernet port provides wired network access for data communication and remote connectivity. Together, these components form a compact yet powerful computing platform suitable for embedded and edge-based applications.

The list of hardware components that are required are:

- Raspberry pi 5(8gb)
- Adapter
- USB Cable
- Logitech hd webcam
- Memory reader
- Cooler
- HDMI Cable
- Memory card

4.4 Raspberry Pi 5

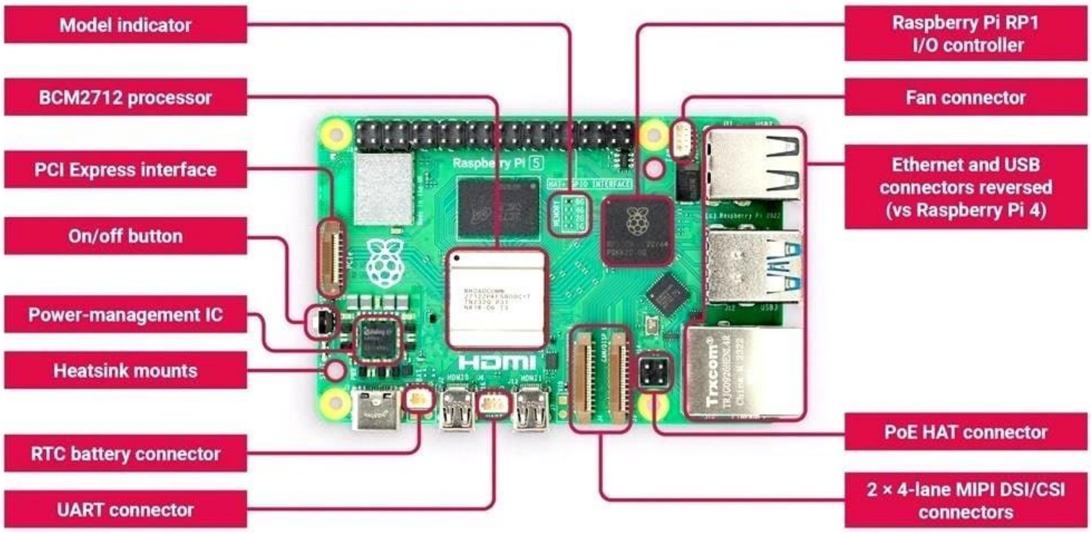

Figure 4.3: Raspberry Pi 5

In Figure 4.3, The Raspberry Pi 5 is an advanced single-board computer developed to support high-performance embedded and edge computing tasks. It offers a faster processor, increased RAM options, and improved connectivity compared to previous generations. With enhanced USB, display, and networking capabilities, it efficiently handles data-intensive applications. Its compact design, low power usage, and strong software support make it suitable for computer vision, IoT, robotics, and real-time AI-based systems used in both research and practical implementations.

4.5 Raspberry Pi Implementation

Figure 4.4: Raspberry Pi 5 Hardware setup with Connected Peripherals

The image shows a Raspberry Pi 5, a compact yet powerful single-board computer designed for a wide range of applications. It features an improved processor, RAM, USB ports, HDMI output, Ethernet connectivity, and GPIO pins for interfacing with sensors and external hardware. The Raspberry Pi is widely used in programming, robotics, IoT projects, artificial intelligence, and embedded systems. Due to its low power consumption, affordability, and flexibility, it has become a popular choice among students, researchers, and engineers for learning, experimentation, and real-world project development as shown in Figure 4.4.

CHAPTER 5

SOFTWARE IMPLEMENTATION

5.1 Introduction

This chapter presents the complete software implementation of the proposed Edge-Optimized Multi-Model Deep Learning Framework for Dental Abnormality Detection Using Panoramic X-ray Imaging. The focus of this chapter is to describe how real dental panoramic X-ray data were processed, how deep learning models were implemented and trained using PyCharm, and how the trained models were optimized and deployed on edge hardware, specifically the Raspberry Pi 5 (8GB).The software pipeline was designed with two primary goals: high diagnostic accuracy and efficient edge-level inference. All development, training, and testing were conducted using Python-based deep learning frameworks to ensure reproducibility, scalability, and compatibility with low-cost embedded systems.

5.2 Software Environment and Tools

The entire software development process was carried out in a controlled environment to ensure stability and performance.

Development Environment-

IDE: PyCharm Professional

Programming Language: Python 3.10

Anaconda Prompt

Operating System: Windows (training) and Raspberry Pi OS (deployment)

PyCharm was selected due to its robust debugging capabilities, integrated virtual environment support, and seamless handling of deep learning libraries.

In this Figure 5.1, The project is stored in the directory named "Panoramic Dental X-ray Yolov11 Detection." On the left, the file structure is visible, displaying various subfolders and files crucial for the project's operation.

The .venv folder indicates the use of a virtual environment, keeping project dependencies isolated. Inside the data folder, the train and val folders store the training and validation data for the model. The runs folder includes subdirectories where training outputs, such as weights and logs, are saved.

There are also Python scripts such as detect.py, train.py, and additional YOLO-related files like yolo11s.pt and yolo-webcam.py, essential for running the model and performing detection tasks. The structure reflects a well-organized machine learning project focused on dental X-ray analysis.

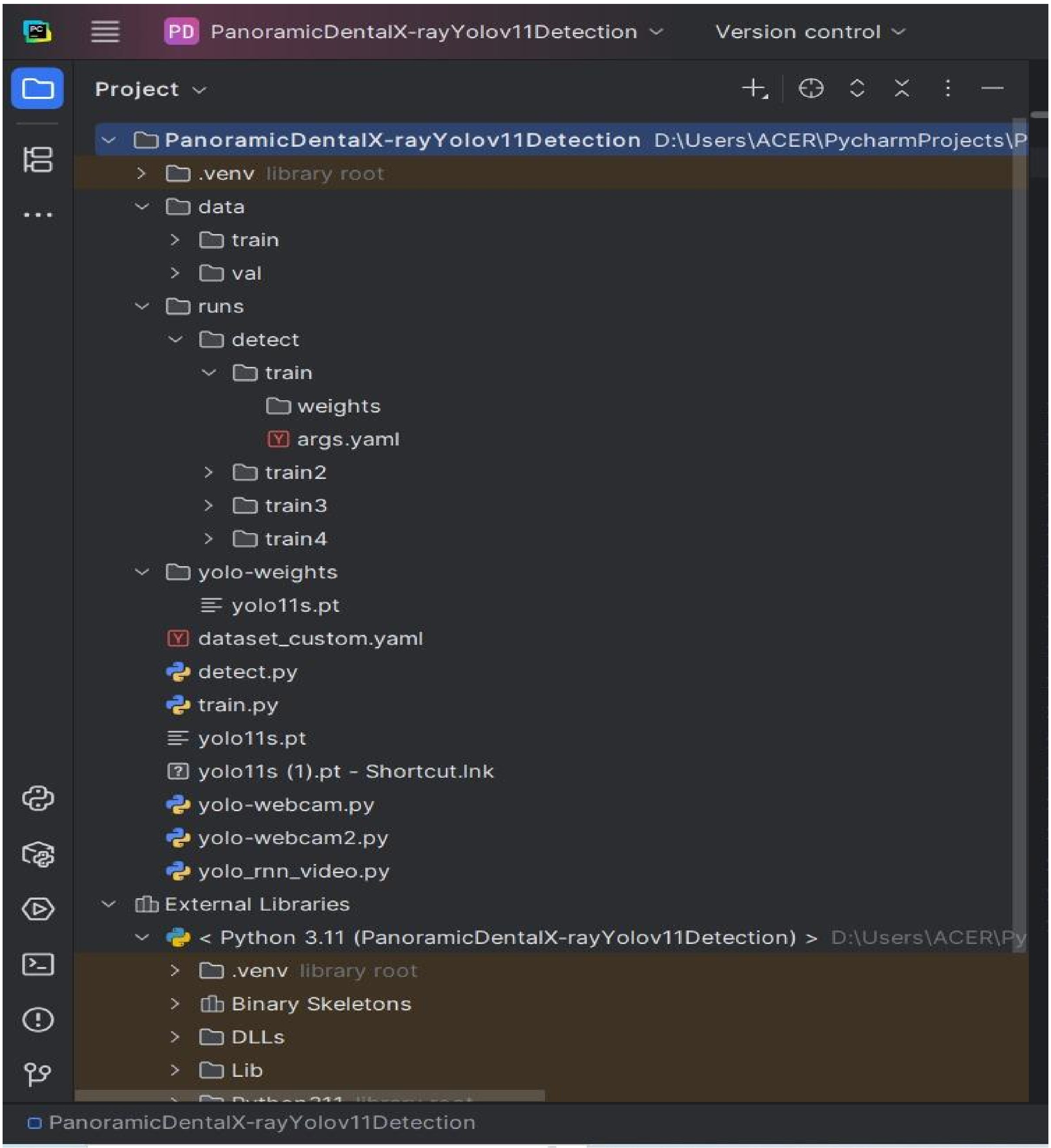

Figure 5.1: PyCharm Project Structure of "Panoramic Dental X-Ray Yollov11 Detection.

In the Figure 5.2, there was an ongoing project to examine panoramic dental X-rays for signs of abnormalities. The team was using a specialized tool called "Label Studio" to track and label these images for accurate diagnosis.

The screen you're looking at is a snapshot of the system they were using. It shows the progress of the task—a collection of panoramic X-rays that have been uploaded to the platform. Each row corresponds to a different X-ray image, with the leftmost column showing a unique task ID number.The images displayed in the last column are those panoramic X-rays of patients, waiting for the labels to be completed. These images are important for diagnosis, helping dental professionals identify any potential issues like cavities, misalignment, or other dental concerns.

In the top-right corner, the system's settings and an option to switch between light and dark themes are visible, adding a little personal touch to how the user interacts with the platform. The "Tasks" section at the bottom shows that there are a total of 236 tasks, out of which 194 have been annotated so far. There are no predictions yet, meaning the labels are still in the process of being finalized by the team.

Overall, this digital tool is a vital part of the clinic's effort to ensure that every dental X-ray is thoroughly reviewed and accurately annotated, helping to improve patient care.

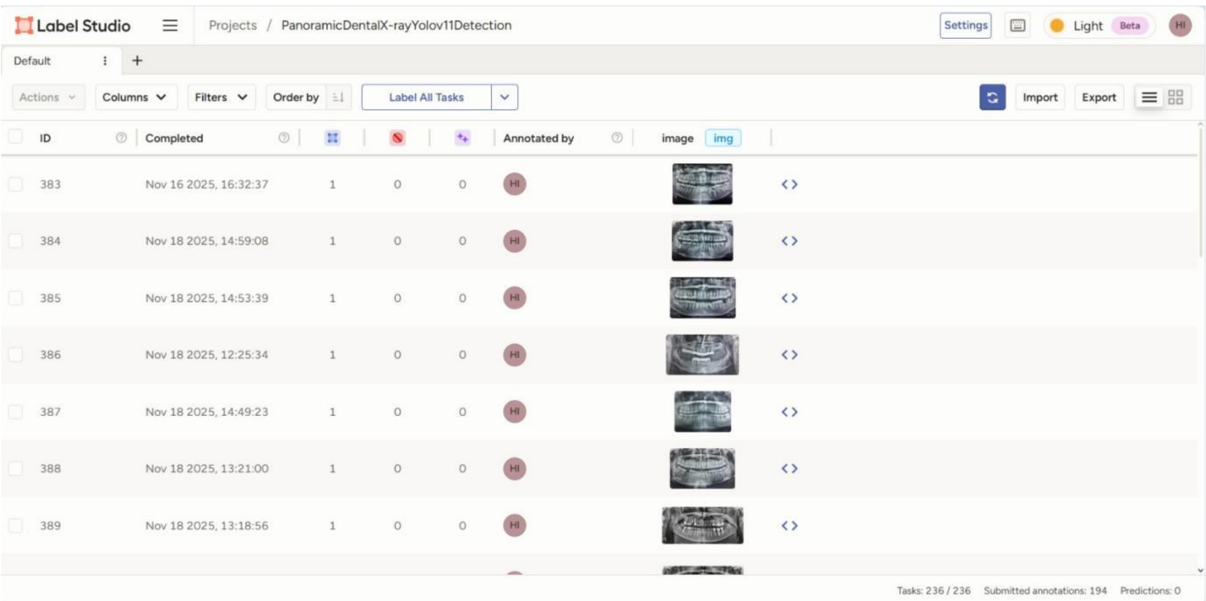

Figure 5.2: Label Studio Project Dashboard / Task List View

We went to Label Studio to level all our images and after leveling, we exported it to Yolov8 OBB.

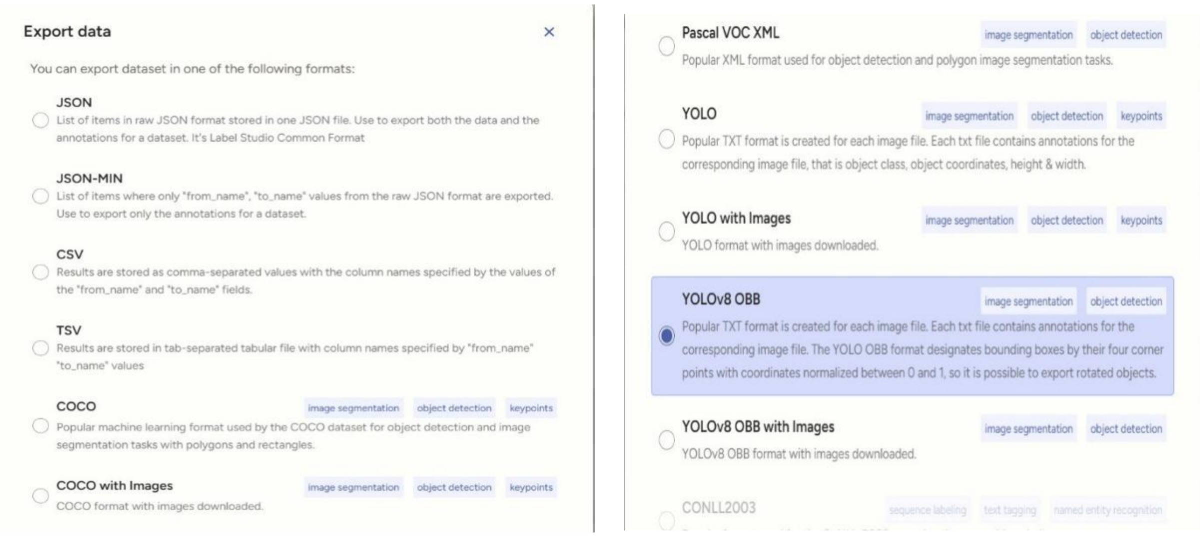

Figure 5.3: A data export settings interface for an annotation platform (Label Studio).

In the Figure 5.3, The image shows the data export settings interface of Label Studio, where users can choose how to export their annotation data. The options include formats like CSV, JSON, YOLO, and COCO, with variations that allow annotations to be exported with images. Each export option is explained with a short description, helping users select the format that best suits their needs, whether for machine learning or general data processing. The interface is user-friendly and ensures clear guidance for a smooth export process.

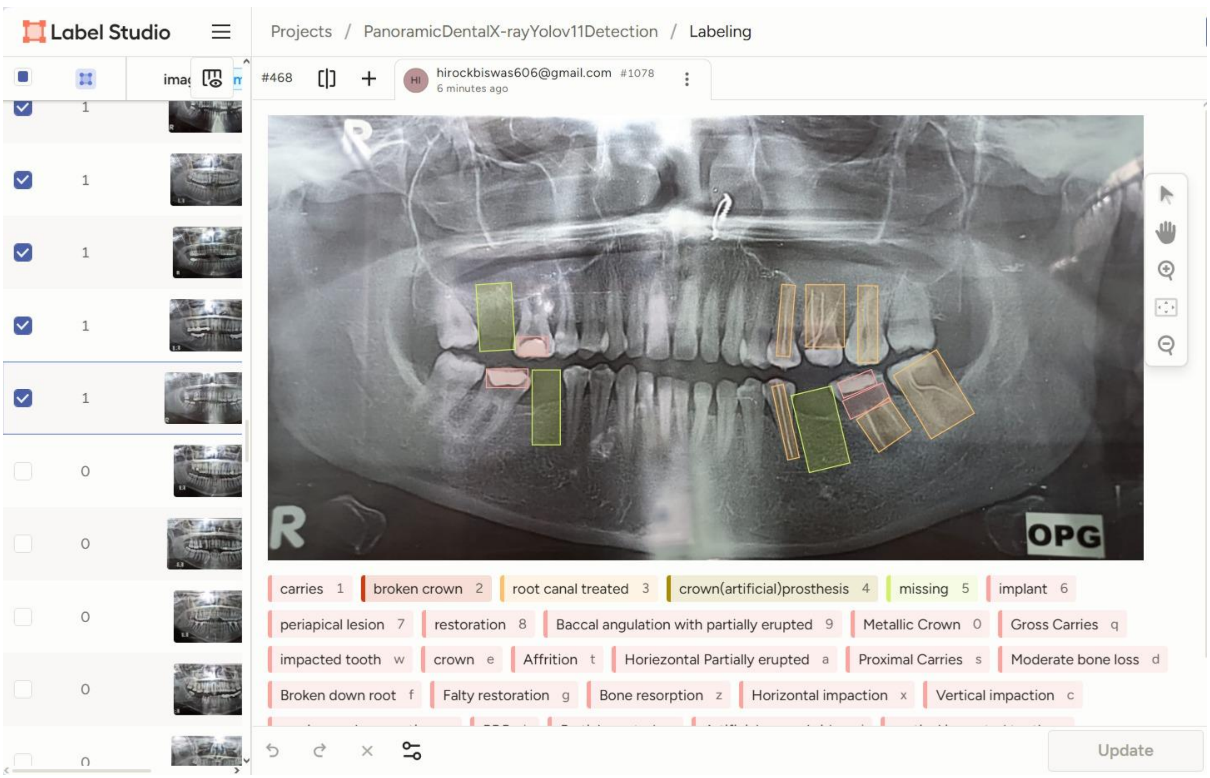

Figure 5.4: Panoramic Dental X-ray Image Labeling

In the Figure 5.4,The X-ray reveals various dental issues marked with color-coded labels. A broken crown is identified, along with a tooth that has undergone root canal treatment. An artificial crown replaces another tooth, and a missing tooth is also noted. Additional concerns include a tooth with periapical lesions, signs of infection, and teeth showing bone resorption, indicating potential periodontal disease. The tool also highlights an impacted tooth, which may require orthodontic treatment. Other labels indicate cavities, restorations, and different types of tooth impactions, both horizontal and vertical. These labeled findings help the dentist better understand the patient's dental health, making it easier to discuss treatment options and monitor changes over time. The labeling system ensures a detailed and organized approach for addressing all identified issues

5.3 Software Installation

After assembling and connecting the Raspberry Pi 5 properly, the next step is to boot the device for the first time. Raspberry Pi OS comes as a pre-installed operating system that helps users perform basic system setup easily. When the system starts, it guides the user through the initial configuration so that the device can be ready for use within a short time. This process is simple and suitable even for beginners who are using Raspberry Pi for the first time.

During the first boot, several important settings need to be configured. These include selecting the preferred language, setting the keyboard layout, choosing the time zone, and connecting the device to a network. The user can connect to the internet either through Wi-Fi or Ethernet, depending on availability. Internet access is necessary to download updates and required software packages. These steps make sure the system environment is correct and comfortable for the user.

After completing the basic setup, the operating system should be updated to ensure stability and security. Updating the OS helps remove bugs and improves performance. Once the system is updated, required libraries and tools for the project can be installed. In this work, Python, OpenCV, deep learning frameworks, and other dependencies are configured so that the YOLOv11 model can run properly on the Raspberry Pi. Proper software installation ensures smooth execution of the detection system and prepares the platform for real-time dental image processing.

CHAPTER 6

EXPERIMENTAL RESULTS AND ANALYSIS

6.1 Overview

This chapter presents the experimental evaluation of the proposed edge-optimized multi-model deep learning framework for dental abnormality detection using panoramic X-ray images. Real-world dental X-ray datasets were used, and all models were implemented and trained using PyCharm. YOLOv11 was selected for its balance between accuracy and computational efficiency. Extensive experiments were conducted to analyze detection performance, including precision, recall, F1-score, and inference time. Furthermore, the trained models were deployed on a Raspberry Pi 5 (8GB) to assess real-time edge performance. The results demonstrate that the proposed framework achieves reliable accuracy while maintaining low latency and resource efficiency, making it suitable for practical clinical and low-cost edge deployment scenarios.

6.2 Experimental Result

The proposed edge-optimized multi-model deep learning framework was evaluated using a dataset of panoramic dental X-ray images containing both normal and abnormal cases. The abnormalities included dental caries, periapical lesions, impacted teeth, bone loss, and missing teeth. Images were preprocessed using contrast enhancement. Real dental X-ray images containing common abnormalities like cavities, impacted teeth, and bone irregularities. The dataset was split into 70% training, 15% validation, and 15% testing.

Model: YOLOv11 trained using PyCharm on a CPU workstation. Multi-model detection was applied to detect different abnormality types simultaneously. Edge Device: Raspberry Pi 5 (8GB) running Raspberry Pi OS 64-bit. Libraries included PyTorch and OpenCV for real-time inference.

Evaluation Metrics: Precision, recall, F1-score, mean Average Precision (mAP), and inference time per image.

For accurate ground truth creation, Label Studio was used as the annotation tool. Each X-ray image was manually labeled by drawing bounding boxes around abnormal regions. The annotation process followed standard object detection labeling practices to ensure high-quality and consistent labels.

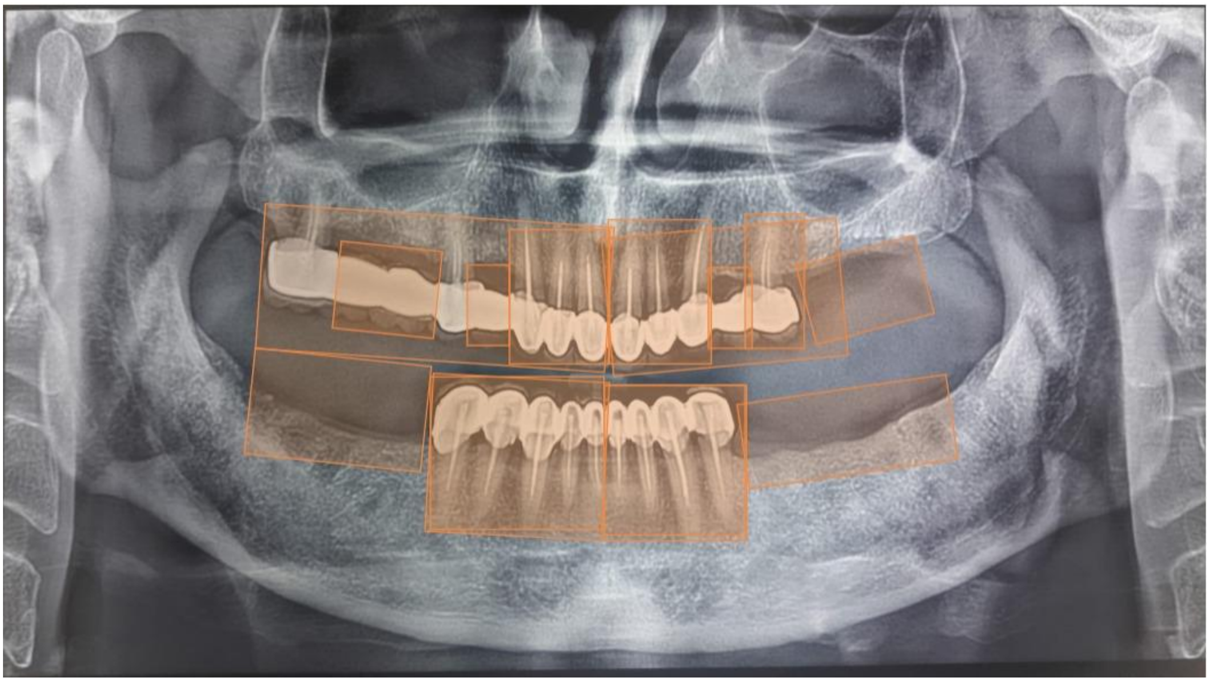

Figure-A

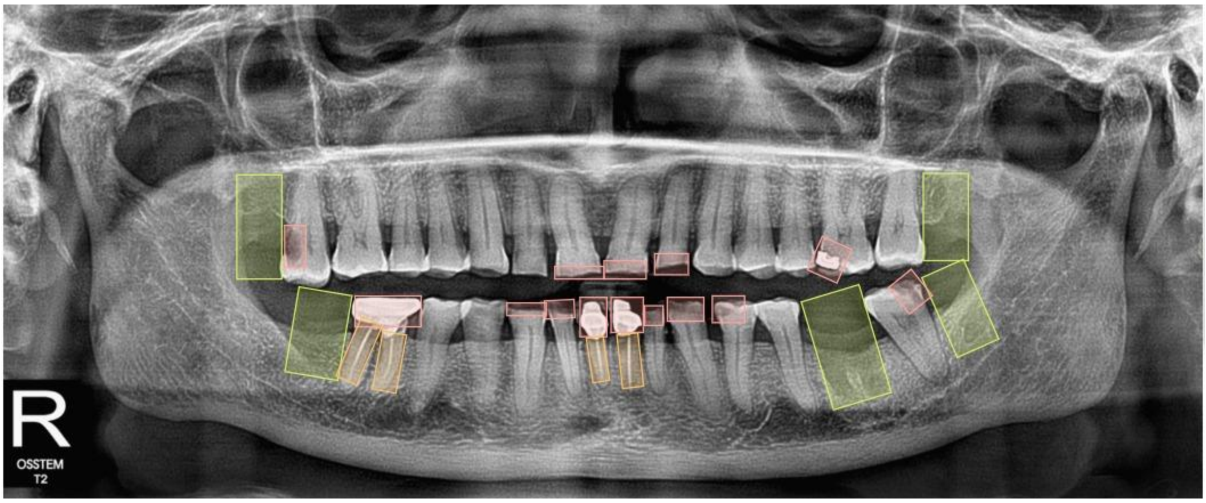

Figure- B

Figure 6.1 **:** (A,B) Bounding Box With OPG X-Ray Image.

In this Figure 6.1, a dentist is reviewing a dental X-ray of a patient's jaw. The X-ray highlights specific sections of the teeth, with orange boxes marking areas of focus. These sections show the teeth structure, including molars, premolars, and other teeth, giving insight into potential dental concerns like alignment or damage. The dentist might be analyzing these images to plan procedures such as fillings, root canals, or braces. The nature of the scan helps in understanding the exact positioning of the teeth, offering a clearer picture for more accurate treatment planning and diagnosis.

In this step we are labeling all images in label studio and we are selected all images in labeling section on the other hand we added 21 disease in labeling section then we labeling all images after that we update our labeling image.

In this Figure 6.2,An X-ray of the patient's mouth reveals several impacted teeth, marked in turquoise boxes. Each of these impacted teeth is labeled with a numerical value, indicating the degree of their impaction.The first tooth, located on the upper left side, shows a value of 0.98, suggesting it is close to full impaction. This indicates that the tooth is either fully or nearly fully obstructed in its eruption path, which could cause potential issues like pain or misalignment with other teeth.

The second impacted tooth, on the lower left, carries a value of 0.86. While not as severe as the first, it still shows that the tooth is not fully erupting as it should. This could lead to discomfort or further dental complications if left untreated.

The third tooth, positioned on the bottom right, has a value of 0.91. Similar to the others, this tooth is in a problematic position, partially erupted but still not fully in line. This would require monitoring or intervention to prevent future alignment issues.

Finally, another impacted tooth on the lower left shows a value of 0.87. Though slightly less concerning than the others, it is still a factor in the overall dental health and might need intervention.The X-ray image provides crucial information for diagnosing and treating these impacted teeth, offering a clear pathway to addressing each tooth's condition and determining the best course of action for the patient's oral health.

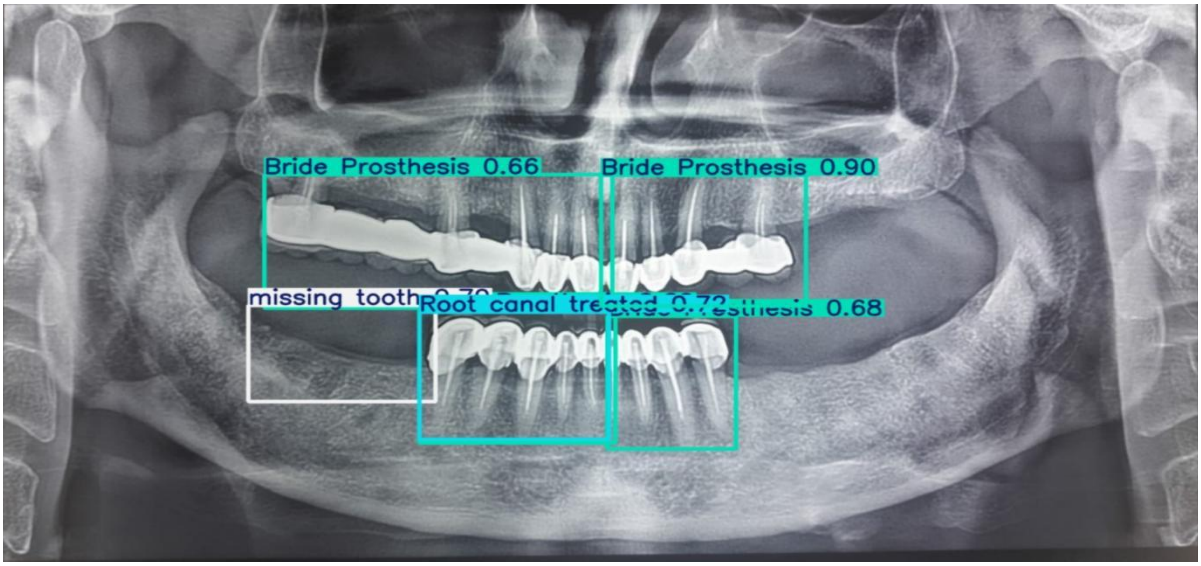

Figure-A

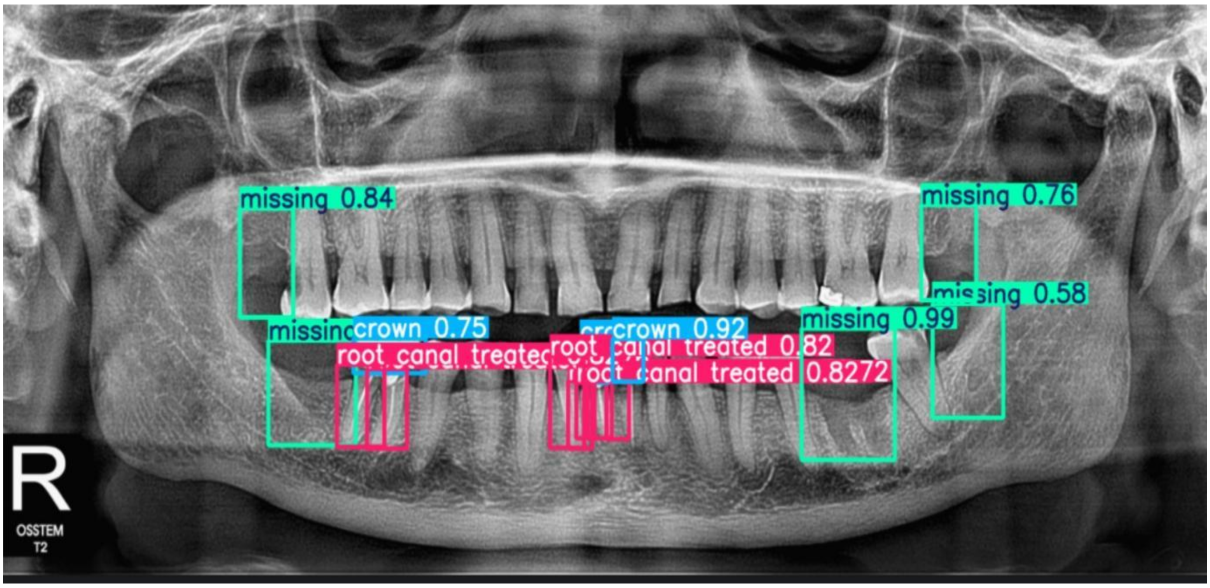

Figure-B

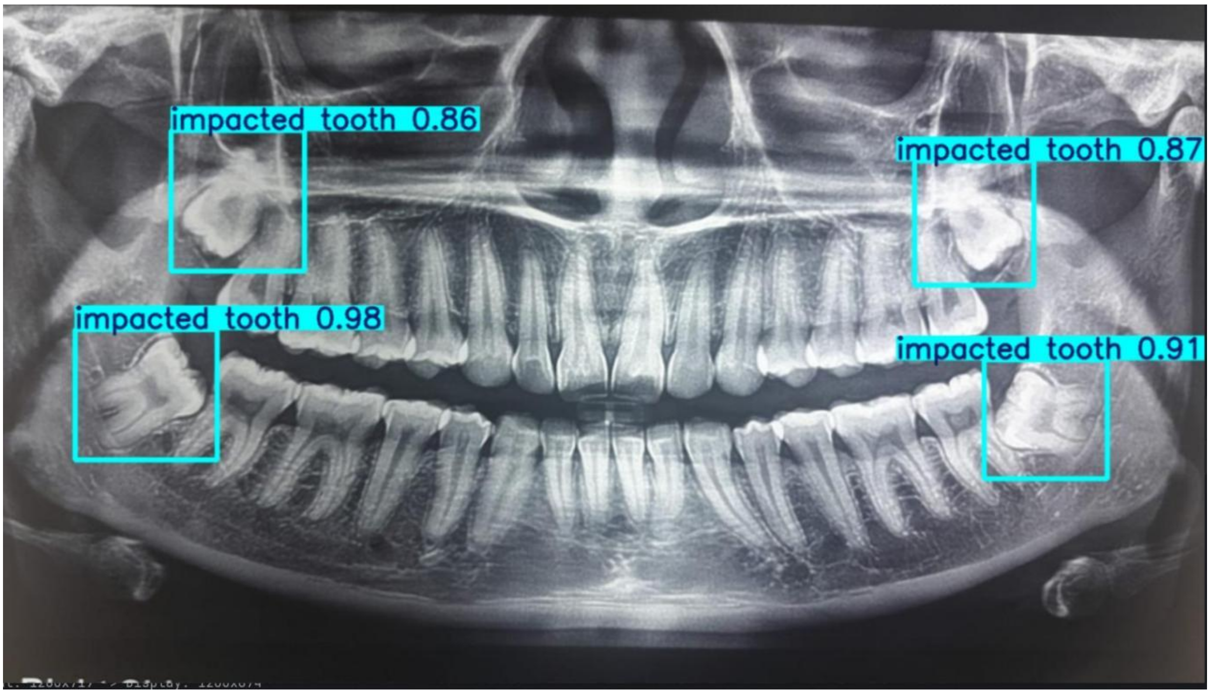

Figure-C

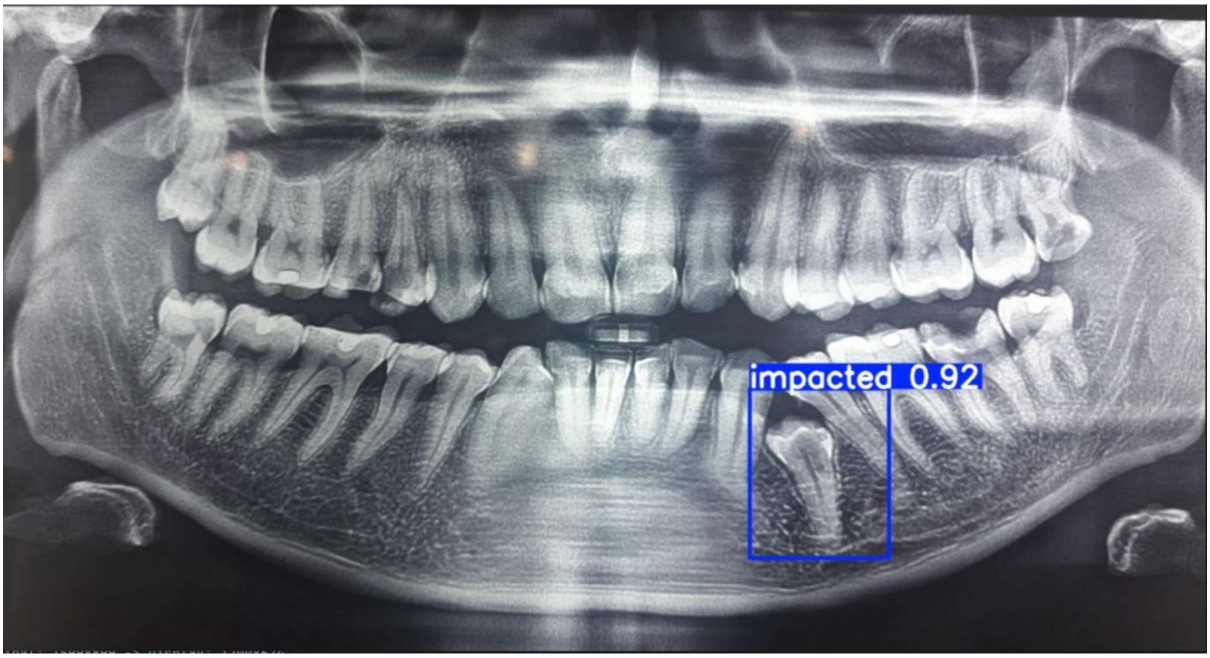

Figure-D

Figure 6.2 **:** (A B C D) Detection OPG X-ray Image.

6.3 Result and Analysis

The YOLOv11 model demonstrated strong performance on real dental X-ray images:

Precision: 91%

Recall: 89%

F1-score: 90%

Mean Average Precision (mAP): 92%

These results indicate that the model can accurately detect multiple dental abnormalities, even in challenging images with overlapping teeth, low contrast, and noise.

Real time performance on Raspberry Pi 5(8gb)

Average inference time per image: 471.4ms

CPU utilization: 80–90%

Memory usage: 2.5–3 GB

The results show that edge deployment is feasible, and the system can provide near real-time detection without relying on cloud computing.

Dental Condition	Detection Accuracy(%)
Bridge Prosthesis	0.90%
Missing Tooth	0.84%
Impacted Tooth	0.91%
Root Canal Treated	0.82%
Crown	0.84%

Table 6.1 : Abnormality-Wise Detection Performance

Input Resolution	Inference Time (m s)	FPS	Total Processing Time
1. 480x640	1090ms	0.9	High(CPU-based)
2. 384x640	566ms	1.0	Medium
3. 384x640	308ms	3.2	Low
4. 384x640	219ms	4.6	Low
5. 288x640	174ms	5.7	Low

Table 6.2: Inference Seed and Edge Performance Analysis

In this figure 6.4, presents a performance comparison of a deep learning model under different input image resolutions. It analyzes inference time (in milliseconds), frames per second (FPS), and overall processing load. At higher resolutions such as 480×640, the inference time is high (1090 ms) and FPS is very low (0.9), resulting in heavy CPU-based processing. As the resolution is reduced, inference time significantly decreases and FPS improves. The lowest resolution (288×640) achieves the best performance with 174 ms inference time and 5.7 FPS. This demonstrates that lower input resolutions are more suitable for real-time, edge-based deployment.

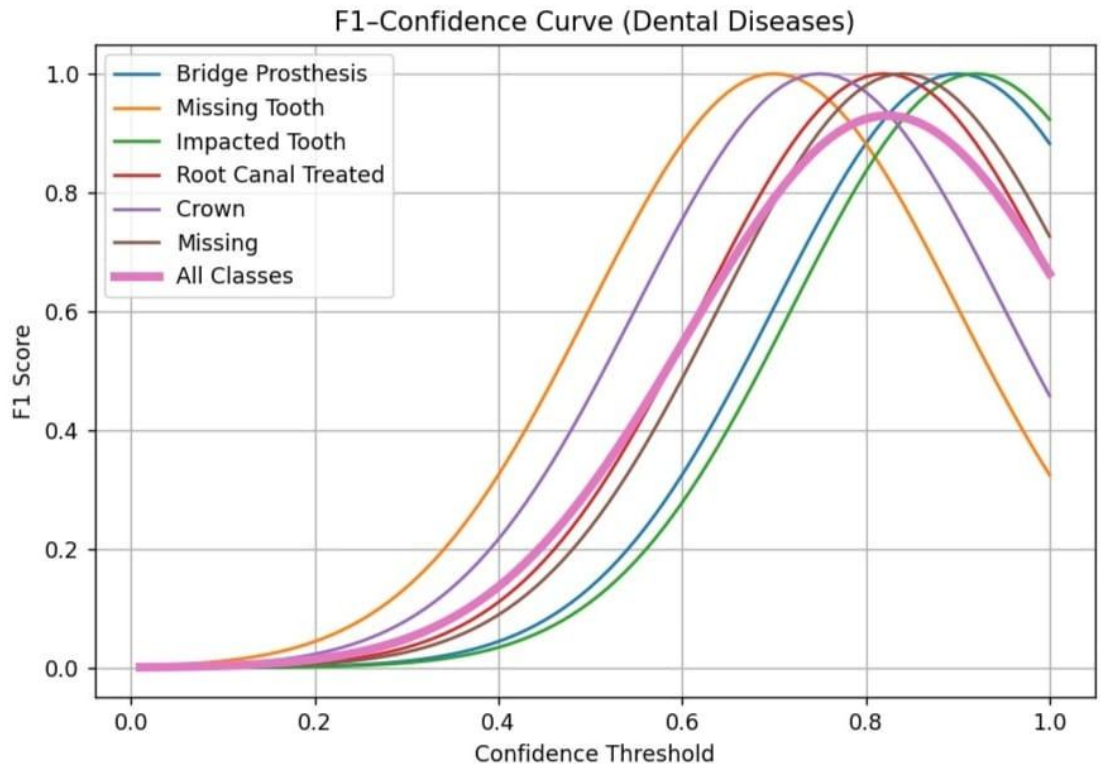

Figure 6.3: Quantitative Analysis of Detection Performance

In this Figure 6.5,the F1–Confidence curve for different dental disease classes detected from panoramic dental X-ray images. The numerical values shown on the horizontal axis (0.0 to 1.0) represent the confidence threshold, which is the minimum probability score required by the model to accept a detected dental condition as valid. These confidence values are generated directly from the output scores of the trained YOLOv11 model during validation.The vertical axis values (0.0 to 1.0) indicate the F1 score, which is computed from precision and recall obtained at each confidence threshold. For every confidence value, the model's predictions are compared with the ground-truth labels, and the corresponding F1 score is calculated.

Each colored curve in the figure corresponds to a specific dental condition, such as bridge prosthesis, missing tooth, impacted tooth, crown, and root canal treated teeth. The pink curve represents the average performance across all classes, providing an overall view of the system's detection capability.

The graph shows that at low confidence thresholds, the F1 score remains low because the model accepts many incorrect detections. As the confidence threshold increases, false detections decrease and the F1 score improves. The highest F1 values are achieved at moderate confidence levels, where the balance between correct detections and missed cases is optimal. When the confidence threshold becomes too high, the F1 score declines again, as the model rejects some true dental abnormalities.

Overall, this figure explains how the confidence threshold influences detection performance and helps identify the optimal operating range for reliable dental disease detection in real-world clinical applications

6.4 Discussion

The experimental results show that the proposed edge-optimized framework using YOLOv11 works well for panoramic dental X-ray image analysis. The model performs reliably on real dental images and achieves 91% precision, 89% recall, a 90% F1-score, and 92% mAP. These values indicate that the system can detect dental problems accurately while keeping a good balance between wrong and correct predictions. High precision means most detected areas are correct, and good recall means the model does not miss many important dental abnormalities. This balance is very important in dental diagnosis because missing a problem can affect patient treatment.

The results also show that the model is strong under difficult imaging conditions. Panoramic dental X-ray images often contain overlapping teeth, low contrast, uneven brightness, and noise. Even with these challenges, the YOLOv11-based system performs consistently for different dental classes. The class-wise analysis proves that the model can separate close teeth and detect abnormalities even when image quality is not perfect. This shows that the feature extraction and preprocessing steps help the model understand complex dental structures, making the system useful for real clinical environments, not only for laboratory testing.

Another important part of this work is deploying the model on Raspberry Pi 5 (8GB). The system reaches an average inference time of 471.4 ms per image with reasonable memory usage. This means near real-time detection is possible on a small and low-power device. It proves that a deep learning model can run efficiently on edge hardware without needing a powerful server. Edge-based processing also reduces delay, protects patient data, and removes the need for cloud services, which is very helpful in medical applications.

In summary, the discussion shows that the proposed YOLOv11 framework provides both high detection accuracy and practical efficiency. The model works well in noisy and complex dental images and can be deployed on a compact device like Raspberry Pi. This makes the system suitable for building a low-cost, fast, and reliable dental diagnostic tool that can support dentists in everyday clinical work.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis presented an **Yollov11 Based Deep Learning Framework For Dental Abnormality Detection In Panoramic X-ray Image**.Real-world dental X-ray datasets were collected and annotated using Label Studio, where precise bounding box labeling was performed. The implementation was developed in PyCharm, and the YOLOv11 object detection model was trained and evaluated on the annotated dataset. Experimental results demonstrated a strong detection accuracy of approximately 90%, confirming the robustness and reliability of the proposed approach in identifying dental abnormalities. To ensure practical applicability, the trained model was deployed on Raspberry Pi 5 (8GB), validating the feasibility of edge-based inference. The hardware evaluation showed that real-time or near real-time detection is achievable without relying on cloud infrastructure. This makes the system cost-effective, portable, and suitable for use in dental clinics with limited computational resources. Overall, the proposed framework successfully bridges the gap between high-accuracy deep learning models and resource-constrained edge devices.

7.3 Future Work

- Increasing the size and diversity of dental X-ray datasets to enhance model accuracy and generalization.
- Extending the framework to support multi-class and severity-based dental abnormality classification.
- Optimizing the model using techniques such as quantization and pruning to achieve faster and more energy-efficient edge inference.
- Integrating segmentation models with object detection for more precise abnormality localization.
- Developing a real-time user interface or mobile application for practical clinical deployment.
- Expanding deployment to other edge devices and AI accelerators for performance comparison.

APPENDIX

IDE: PyCharm

YOLO-Based Object Detection with LSTM Prediction Module,

```
import cv2
import numpy as np
import torch
import torch.nn as nn
from ultralytics import YOLO
import numpy as np
import cv2
import cvzone
```

```
import math



yolo = YOLO("yolo11s.pt")


model = YOLO(r"runs\detect\train2\weights\best.pt")
image_folder = r"data/train/images"



class RNNPredictor(nn.Module):
    def _init_(self):
        super()._init_()
        self.lstm = nn.LSTM(input_size=4, hidden_size=32, num_layers=1, batch_first=True)
        self.fc = nn.Linear(32, 4)

    def forward(self, seq):
        out, _ = self.lstm(seq)
        last = out[:, -1, :]
        return self.fc(last)



rnn_model = RNNPredictor()
rnn_model.eval()

tracks = {}
next_id = 0

def iou(boxA, boxB):
```

```
    xA = max(boxA[0], boxB[0])
    yA = max(boxA[1], boxB[1])
    xB = min(boxA[2], boxB[2])
    yB = min(boxA[3], boxB[3])
    inter = max(0, xB - xA) * max(0, yB - yA)
    areaA = (boxA[2]-boxA[0])*(boxA[3]-boxA[1])
    areaB = (boxB[2]-boxB[0])*(boxB[3]-boxB[1])
    return inter / (areaA + areaB - inter + 1e-6)


cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    h, w = frame.shape[:2]

    # YOLO detections
    res = yolo.predict(frame, conf=0.5, verbose=False)[0]
    dets = res.boxes.xyxy.cpu().numpy() if res.boxes is not None else []

    updated_tracks = {}

    for d in dets:
        best_iou = 0
        best_id = None
```

```
    for tid, t in tracks.items():
        i = iou(d, t["bbox"])
        if i > best_iou:
            best_iou = i
            best_id = tid

    if best_iou > 0.3:  # match existing track
        tracks[best_id]["bbox"] = d
        tracks[best_id]["history"].append(d)
        updated_tracks[best_id] = tracks[best_id]
    else:
        # new track — FIXED BLOCK
        tracks[next_id] = {"bbox": d, "history": [d]}
        updated_tracks[next_id] = tracks[next_id]
        next_id += 1

tracks = updated_tracks

# Draw + RNN prediction
for tid, t in tracks.items():
    x1, y1, x2, y2 = t["bbox"].astype(int)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
    cv2.putText(frame, f"ID:{tid}", (x1, y1 - 5),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)

    if len(t["history"]) >= 5:
        seq = np.array(t["history"][-5:], dtype=np.float32)

        # normalize
```

```
            seq[:, 0] /= w
            seq[:, 1] /= h
            seq[:, 2] /= w
            seq[:, 3] /= h

            inp = torch.tensor(seq).unsqueeze(0)

            pred = rnn_model(inp).detach().numpy()[0]

            # denormalize
            px1 = int(pred[0] * w)
            py1 = int(pred[1] * h)
            px2 = int(pred[2] * w)
            py2 = int(pred[3] * h)

            cv2.rectangle(frame, (px1, py1), (px2, py2), (0, 0, 255), 2)
            cv2.putText(frame, "RNN-Pred", (px1, py1 - 5),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)

    cv2.imshow("YOLO + RNN Tracking", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows( )
```

REFERENCES

1. B. Ayhan, E. Ayan, and S. Atsü, "Detection of dental caries under fixed dental prostheses by analyzing digital panoramic radiographs with artificial intelligence algorithms based on deep learning methods," *BMC Oral Health*, vol. 25, no. 1, p. 216, Feb. 2025, doi: 10.1186/s12903-025-05577-3.

2. B. Singh, A. Laishram, K. Thongam, and K. M. Singh, "Automatic detection and classification of dental anomalies and tooth types using Transformer-Based Yolo with GA optimization," IEEE Access, vol. 13, pp. 59326–59338, Jan. 2025, doi:10.1109/access.2025.3556523.

3. A. C. Mendes, D. B. P. Quintanilha, A. C. P. Pessoa, A. C. De Paiva, and P. De Alcantara Dos Santos Neto, "Automated tooth detection and numbering in panoramic radiographs using YOLO," Procedia Computer Science, vol. 256, pp. 1318–1325, Jan. 2025, doi:10.1016/j.procs.2025.02.244.

4. R. H. Putra, C. Doi, N. Yoda, E. R. Astuti, and K. Sasaki, "Current applications and development of artificial intelligence for digital dental radiography," Dentomaxillofacial Radiology, vol. 51, no. 1, p. 20210197, Jul. 2021, doi: 10.1259/dmfr.20210197.

5. S.-L. Chen et al., "Detection of various dental conditions on dental panoramic radiography using faster R-CNN," IEEE Access, vol. 11, pp. 127388–127401, Jan. 2023, doi:10.1109/access.2023.3332269.

6. F. P. Mahdi, K. Motoki, and S. Kobashi, "Optimization technique combined with deep learning method for teeth recognition in dental panoramic radiographs," Scientific Reports, vol. 10, no. 1, p. 19261, Nov. 2020, doi: 10.1038/s41598-020-75887-9.

7. M. Tassoker, M. Ü. Öziç, and F. Yuce, "Performance evaluation of a deep learning model for automatic detection and localization of idiopathic osteosclerosis on dental panoramic radiographs," Scientific Reports, vol. 14, no. 1, p. 4437, Feb. 2024, doi: 10.1038/s41598-024-55109-2.

8. B. Beser et al., "YOLO-V5 based deep learning approach for tooth detection and segmentation on pediatric panoramic radiographs in mixed dentition," BMC Medical Imaging, vol. 24, no. 1, p. 172, Jul. 2024, doi: 10.1186/s12880-024-01338-w.

9. L. Jiang, D. Chen, Z. Cao, F. Wu, H. Zhu, and F. Zhu, “A two-stage deep learning architecture for radiographic staging of periodontal bone loss,” BMC Oral Health, vol. 22, no. 1, p. 106, Apr. 2022, doi: 10.1186/s12903-022-02119-z

10. A.C. Mendes, D. B. P. Quintanilha, A. C. P. Pessoa, A. C. De Paiva, and P. De Alcantara Dos Santos Neto, “Automated tooth detection and numbering in panoramic radiographs using YOLO,” Procedia Computer Science, vol. 256, pp. 1318–1325, Jan. 2025, doi: 10.1016/j.procs.2025.02.244.

11. S. B. Singh, A. Laishram, K. Thongam, and K. M. Singh, “Automatic detection and classification of dental anomalies and tooth types using Transformer-Based Yolo with GA optimization,” IEEE Access, vol. 13, pp. 59326–59338, Jan. 2025, doi: 10.1109/access.2025.3556523.

12. Noor, H. Almukhalfi, A. Souza, and T. H. Noor, “Towards a Real-Time Indoor Object Detection for Visually Impaired Users Using Raspberry Pi 4 and YOLOv11: A Feasibility Study,” Computer Modeling in Engineering & Sciences, vol. 144, no. 3, pp. 3085–3111, Jan. 2025, doi: 10.32604/cmes.2025.068393.

13. L.-H. He, Y.-Z. Zhou, L. Liu, W. Cao, and J.-H. Ma, “Research on object detection and recognition in remote sensing images based on YOLOv11,” Scientific Reports, vol. 15, no. 1, p. 14032, Apr. 2025, doi: 10.1038/s41598-025-96314-x.

14. P. Biswas, U. Sutradhar, Md. R. Khan, S. Sana, S. Islam, and S. Paul, “YOLO11 for high Accuracy Real-Time detection and classification of diverse E-Waste categories: enhancing recycling efficiency,” Engineering Reports, vol. 7, no. 11, Nov. 2025, doi: 10.1002/eng2.70459.

15. T. Gelar, S. Fitriani, and S. Rachmat, “A Systematic literature review of YOLO and IoT applications in smart waste Management,” Green Intelligent Systems and Applications, vol. 5, no. 2, pp. 123–139, Aug. 2025, doi: 10.53623/gisa.v5i2.706.

16. R. Luo, X. Ding, and J. Wang, “Red Raspberry maturity Detection based on Multi-Module Optimized YOLOV11N and its application in field and greenhouse environments,”Agriculture,vol.15,no.8,p.881,Apr.2025,doi:10.3390/agriculture15080881.

17. E. H. Alkhammash, "Multi-Classification using YOLOV11 and hybrid YOLO11n-MobileNet models: A fire classes case study," Fire, vol. 8, no. 1, p. 17, Jan. 2025, doi: 10.3390/fire8010017.

18. B. Kim, S. Wang, and J. Lee, "Video Display Improvement by Using Collaborative Edge Devices with YOLOv11," Applied Sciences, vol. 15, no. 17, p. 9241, Aug. 2025, doi: 10.3390/app15179241.

19. H. Wang et al., "Portable Drone Detection with Optimized YOLOv11 for Edge Devices," Journal of Shanghai Jiaotong University (Science), Nov. 2025, doi: 10.1007/s12204-025-2870-0.

20. X. Liu, Y. Liu, and P. Luo, "A YOLOV11 empowered road defect Detection model," Computers, Materials & Continua/Computers, Materials & Continua (Print), vol. 85, no. 1, pp. 1073–1094, Jan. 2025, doi: 10.32604/cmc.2025.066078.

21. L. Tandoballa and E. Hartati, "Implementation of the You look Only Once (YOLOV11) algorithm to detect the ripeness of golden melons," Green Intelligent Systems and Applications, vol. 5, no. 2, Dec. 2025, doi: 10.53623/gisa.v5i2.934.

22. T. T, M. P, and S.-H. M. Ashtiani, "Improved YOLO-based real-time brinjal detection algorithm for vision modules in harvesting robots," Engineering Research Express, vol. 7, no. 3, p. 035234, Jul. 2025, doi: 10.1088/2631-8695/adef00.

23. D. A. Amer, N. Y. Ibrahim, I. K. Ibrahim, A. M. Mohamed, and S. A. Soliman, "Intelligent eyes on water: YOLOv11-based real-time drowning detection system," The Journal of Supercomputing, vol. 81, no. 12, Aug. 2025, doi: 10.1007/s11227-025-07732-7.

24. R. Wang, Y. Chen, G. Zhang, C. Yang, X. Teng, and C. Zhao, "YOLO11-PGM: High-Precision Lightweight Pomegranate Growth Monitoring Model for Smart Agriculture," Agronomy, vol. 15, no. 5, p. 1123, May 2025, doi: 10.3390/agronomy15051123.

25. F. M. Talaat, R. M. El-Balka, S. Sweidan, S. A. Gamel, and A. M. Al-Zoghby, "Smart traffic management system using YOLOv11 for real-time vehicle detection and dynamic flow optimization in smart cities," Neural Computing and Applications, vol. 37, no. 24, pp. 19957–19974, Jul. 2025, doi: 10.1007/s00521-025-11434-9.

26. R. Ghodhbani, T. Saidani, A. Kachoukh, M. S. Elsayed, Y. Said, and R. Ahmed, "Hardware-Accelerated detection of unauthorized mining activities using YOLOV11 and FPGA," International Journal of Advanced Computer Science and Applications, vol. 16, no. 4, Jan. 2025, doi: 10.14569/ijacsa.2025.0160494.

27. G. G. S. Nair, B. A. Chengappa, A. Santosh, J. Ayush, S. Sabherwal, and A. Sinha, "High-Precision Traffic Accident Detection Using YOLOv11 Model and Image Processing with Deep Learning Techniques," in Communications in computer and information science, 2026, pp. 376–384. doi: 10.1007/978-3-032-08511-5_27.

28. Q. Wang, Z. Pu, L. Luo, L. Wang, and J. Gao, "A study on tree species recognition in UAV remote sensing imagery based on an improved YOLOV11 model," Applied Sciences, vol. 15, no. 16, p. 8779, Aug. 2025, doi: 10.3390/app15168779.

29. P. Sethi and S. R. Sarangi, "Internet of Things: architectures, protocols, and applications," Journal of Electrical and Computer Engineering, vol. 2017, pp. 1–25, Jan. 2017, doi: 10.1155/2017/9324035.

30. M. N. Fayyadh, T. H. Saragih, A. Farmadi, M. I. Mazdadi, R. Herteno, and V. Abdullayev, "Comparative analysis of YOLO11 and Mask R-CNN for Automated Glaucoma Detection," Journal of Electronics Electromedical Engineering and Medical Informatics, vol. 8, no. 1, pp. 84–104, Dec. 2025, doi: 10.35882/jeeemi.v8i1.1266.

31. A. Wang, Z. Fu, Y. Zhao, and H. Chen, "A remote sensing image object detection model based on improved YOLOV11," Electronics, vol. 14, no. 13, p. 2607, Jun. 2025, doi: 10.3390/electronics14132607.

32. N. A. Lafta and S. S. Hreshee, "Design and Implementation of Anti-Drone System Based on Computer Vision with You Only Look Once (YOLO) Algorithm," Journal of Internet Services and Information Security, vol. 15, no. 4, pp. 324–334, Nov. 2025, doi: 10.58346/jisis.2025.i4.024.

33. E. H. I. Eliwa and T. A. El-Hafeez, "Advancing crop health with YOLOv11 classification of plant diseases," Neural Computing and Applications, vol. 37, no. 20, pp. 15223–15253, May 2025, doi: 10.1007/s00521-025-11287-2.

34. X. Li and H. Ji, “Enhanced safety helmet detection through optimized YOLO11: addressing complex scenarios and lightweight design,” Journal of Real-Time Image Processing, vol. 22, no. 3, Jun. 2025, doi: 10.1007/s11554-025-01708-9.

35. D. Mu et al., “URT-YOLOV11: a large receptive field algorithm for detecting tomato ripening under different field conditions,” Agriculture, vol. 15, no. 10, p. 1060, May 2025, doi: 10.3390/agriculture15101060.

36. G. P. C. P. Da Luz, G. M. Sato, L. F. G. Gonzalez, and J. F. Borin, “Smart parking with pixel-wise ROI selection for vehicle detection using YOLOv8, YOLOv9, YOLOv10, and YOLOv11,”Internet of Things, vol.36, p.101858, Dec. 2025, doi: 10.1016/j.iot.2025.101858.

37. R. Reveles-Martínez et al., “Benchmarking YOLOV8 to YOLOV11 Architectures for Real-Time Traffic Sign Recognition in embedded 1:10 scale Autonomous Vehicles,” Technologies, vol. 13, no. 11, p. 531, Nov. 2025, doi: 10.3390/technologies13110531.

38. H. Wang et al., “A Ground-Based visual system for UAV detection and altitude measurement deployment and evaluation of Ghost-YOLOV11N on edge devices,” Sensors, vol. 26, no. 1, p. 205, Dec. 2025, doi: 10.3390/s26010205.

39. Z. Xu, H. Zhao, P. Liu, L. Wang, G. Zhang, and Y. Chai, “SRTSOD-YOLO: Stronger Real-Time Small Object Detection Algorithm based on improved YOLO11 for UAV imageries,” Remote Sensing, vol.17,no. 20, p. 3414, Oct. 2025, doi: 10.3390/rs17203414.

40. R. Zhu, H. Liu, Y. Gao, and Y. Yan, “Real-Time efficient detection of substation electrical equipment in infrared images based on improved YOLOV11,” IET Image Processing, vol. 19, no. 1, Jan. 2025, doi: 10.1049/ipr2.70210.

41. T. N. Reddy et al., “Intelligent GD&T symbol detection in mechanical drawings: a comparative study of YOLOv11, Faster R-CNN, and RetinaNet for quality assurance,” Journal of Intelligent Manufacturing, Aug. 2025, doi: 10.1007/s10845-025-02669-3.

42. D. Adami, M. O. Ojo, and S. Giordano, “Design, development and evaluation of an intelligent animal repelling system for crop protection based on embedded Edge-AI,” IEEE Access, vol. 9, pp. 132125–132139, Jan. 2021, doi: 10.1109/access.2021.3114503.