ENGR. MUHAMMAD ASAD ZEESHAN
09-244242-003

AI-Powered Framework for Real-Time Detection of Prostate Cancer from MRI Scans

Master of Science in Electrical Engineering

Supervisor: Dr Junaid Imtiaz

Co-Supervisor: Engr.Umair Shahid

Department of Electrical Engineering
Bahria University, Islamabad

June 2026

MS-13
Thesis Completion Certificate

Student Name: **Muhammad Asad Zeeshan** Registration Number: **71656**
Program of Study: **Master of Science in Electrical Engineering**
Thesis Title: **AI-Powered Framework for Real-Time Detection of Prostate Cancer from MRI Scans**

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 1 3 % that is within the permissible set by the HEC, for MS/MPhil.

I have also found the thesis in a format recognized by the BU for MS/MPhil thesis.

Principle Supervisor's Signature:____________________
Principle Supervisor's Name: Dr Junaid Imtiaz
June 30, 2026

MS-14A

Author’s Declaration

I, **Muhammad Asad Zeeshan** hereby state that my MSEE thesis titled **“AI-Powered Framework for Real-Time Detection of Prostate Cancer from MRI Scans”** is my own work and has not been submitted previously by me for taking any degree from **“Bahria University, Islamabad”** or anywhere else in the country / world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw cancel my MSEE degree.

MUHAMMAD ASAD ZEESHAN
09-244242-003
June 30, 2026

MS-14B
Plagiarism Undertaking

I, **Muhammad Asad Zeeshan** solemnly declare that research work presented in the thesis titled;

AI-Powered Framework for Real-Time Detection of Prostate Cancer from MRI Scans

is solely my research work with no significant contribution from any other person. Small contribution / help whenever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of Bahria University and the Higher Education Commission of Pakistan towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarised and any material used is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MSEE degree, the university reserves the right to withdraw / revoke my MS EE degree and HEC and the university has the right to publish my name on HEC / University Website on which name of students who submitted plagiarised thesis are placed.

MUHAMMAD ASAD ZEESHAN
09-244242-003
June 30, 2026

Acknowledgements

First and foremost, I express my deepest gratitude to Almighty Allah, the Most Gracious and the Most Merciful, for granting me the strength, wisdom, patience, and perseverance to successfully complete this research. Without His countless blessings and guidance, this work would not have been possible.

I would like to express my sincere appreciation to my respected supervisor, Dr. Junaid Imtiaz, for his invaluable guidance, continuous encouragement, constructive feedback, and unwavering support throughout this research. His expertise, insightful suggestions, and motivation played a vital role in shaping this work. I am also deeply thankful to my co-supervisor, Engr. Umair Shahid, for his technical guidance, valuable recommendations, and continuous assistance during every stage of this thesis.

I am grateful to the faculty members of the Department of Electrical Engineering, Bahria University, Islamabad Campus, for providing an excellent academic environment and the knowledge that contributed to my professional growth. I also extend my heartfelt thanks to my family, friends, and colleagues for their constant encouragement, understanding, and support throughout my academic journey. Their prayers, patience, and motivation have been a source of strength and inspiration in completing this milestone.

MUHAMMAD ASAD ZEESHAN
Islamabad, Pakistan
June 30, 2026

Abstract

Since abnormalities at an early stage of prostate cancer may be hard to spot from manual examination of MRI images, timely and accurate detection is essential. The aim of this thesis was to create an AI-based system for the real-time detection of prostate cancer from MRI images using a deep learning model based on YOLOv11 and a comprehensive clinical workflow. The proposed system will also allow suspicious cancer regions to be detected as well as providing a practical patient-to-physician review process.

The framework starts by preparing the MRI images, re-structuring the MRI datasets, annotations handling and image pre-processing for object detection. A model called YOLOv11 is used to detect cancerous areas in prostate MRI images and output bounding boxes and confidence scores. The trained model is deployed using a FastAPI backend, where uploaded MRI images are processed, validated, fed into the inference pipeline and transformed into clinically relevant output like annotated images and data for the diagnostic report.

For the system to be more practical, there are a separate patient portal and a separate physician portal. Patients register and login to the system, upload an MRI report, or a physician can log in and view the patient's uploaded MRI report, activate the AI model, review the annotated reports, approve and send the MRI report to the patient. The safe access of users is enabled using Firebase Authentication, users, cases, images, results of the detections and report records are stored using MongoDB and the final report delivery is automated using n8n after the physician has given his approval via email.

The framework developed shows how all these technologies deep learning, backend services, authentication, database storage, workflow automation, and clinical review can be integrated into a single AI-powered diagnostic system. Real time detection, explainable visual output, doctor control over when reports are released, and deployment readiness are the key highlights of the project. The system aims to assist radiologists and physicians in minimizing manual workload, optimizing workflow, and creating a structured environment for the AI-driven prostate cancer screening process.

Keywords: Prostate cancer, MRI, YOLOv11, object detection, ONNX, FastAPI, Firebase Authentication, MongoDB, n8n, clinical workflow automation.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the Curve
CNN	Convolutional Neural Network
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
DRE	Digital Rectal Examination
F1	Harmonic mean of precision and recall
FN	False Negative
FP	False Positive
GridFS	Grid File System in MongoDB
JWT	JSON Web Token
MRI	Magnetic Resonance Imaging
NMS	Non-Maximum Suppression
ONNX	Open Neural Network Exchange
PR	Precision-Recall
PSA	Prostate Specific Antigen
RBAC	Role Based Access Control
SMTP	Simple Mail Transfer Protocol
TN	True Negative
TP	True Positive
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Background

Prostate cancer is a major cancer in men and is a common cancer associated with aging [1, 2]. It can progress gradually and be asymptomatic in early stages, making possible early detection challenging without screening and the support of imaging. Common routes for diagnosis include prostate specific antigen (PSA) screening, digital rectal examination, biopsy, ultrasound, and magnetic resonance imaging (MRI) [3, 4]. Of these techniques, MRI plays an important part as it can provide information on the detailed prostate anatomy and possibly the suspicious soft tissue areas before an invasive confirmation is carried out. Medical imaging generates high dimensional visual data.

There can be many slices, different contrast within a single prostate exam, and changes in prostate tissue that might not be obvious. Such images will take trained expertise to be manually reviewed. Although expert review may be available, it can be time-consuming and the diagnosis might be based on experience, image quality, visibility of the lesion and clinical setting. These challenges provide the computer-aided systems with the potential to identify suspicious regions and to help physicians when reviewing the image.

To analyze medical images, deep learning has been a key technology that can learn visual patterns directly from image data [5, 6]. The previous systems tended to view prostate cancer analysis as an image classification problem. A classifier determines if an image is normal or abnormal, but it may not identify the area of interest. Localization is crucial for clinical review. A physician has to be informed about the location, size and certainty of the suspicious region in the model. Therefore, object detection models are more appropriate for the current task than just simple image classifiers.

YOLO (You Only Look Once) is a type of single-stage object detection model [7]. YOLO models make predictions in one pass of a forward process, which consist of bounding boxes and confidence scores. This means that they can be beneficial in locations where rapid detection is needed. The goal of this thesis is to get suspicious prostate cancer regions using YOLOv11 from MRI images. The model is not used as a final diagnostic authority. Rather, it is embedded within a patient and physician workflow and the physician is still responsible for reading and approving the report.

1.2 Clinical and Engineering Motivation

AI medical system is more than just an accurate model. It also needs to be compatible with the clinical pathway to which the model will be applied. Research works typically only present the accuracy of the models, confusion matrix or training curves. From an engineering point of view such results are useful, but short of the answers one might seek on how the image enters the system, how the output is evaluated, how the patient receives the result and how access is controlled.

This research can therefore be motivated by two reasons. The first one is technical and is to train and evaluate a YOLOv11 detector for suspicious cancer region localization in MRI images. The second is a pragmatic: to provide a working framework that will link patient upload, physician review, AI inference, report approval, and report emailing in a controlled workflow. This is more of a prototype of a deployable decision support system and less of a proof-of-concept machine learning experiment.

1.3 Problem Statement

Manual analysis of prostate MRI is very labor intensive and reliant on a trained specialist. Current AI-based approaches can help in prostate cancer analysis, but numerous of them are classification models or are just experimental scripts. They may not offer the option to deliver reviewed reports automatically, physician-controlled report approval or provide localized bounding boxes. In order to make this possible, there is a need for a practical AI system to identify suspicious prostate cancer areas in MRI scans, to funnel cases through a review process with a physician, to securely store the results, and to automatically generate reports once approval.

1.4 Research Aim

This study proposes, builds, and tests an AI-based system for real-time detection of prostate cancer from MR images using the YOLOv11 model and integrates the detector into a secure system communicating with a patient and a physician using FastAPI, Firebase Authentication, MongoDB, n8n, and email-based reporting.

1.5 Research Objectives

Specific aims of the present thesis are presented below.

1. To create prostate MRI object detection data set, including images with prostate cancer and images with background without cancer.
2. To record the split of the dataset for training, validation and testing following the standards of research.
3. To Train and test a YOLOv11 object detector for single class medical localization of cancer in MRI images.
4. To design an image preprocessing and post-processing pipeline for YOLOv11 inference, including resizing, normalization, tensor preparation, confidence filtering, and bounding box annotation.
5. To export the trained model into ONNX format for practical backend inference.
6. Implementing a FastAPI backend to verify Firebase tokens, handle user roles, store patient cases, trigger the inference, create reports and communicate with n8n.
7. To develop a patient portal for account registration, login, MRI upload, and report viewing.
8. To create a physician portal for case review, AI triggering, result inspection, report approval and final release to the patient.
9. To integrate n8n automation for physician approved report emailing.
10. To test the system implemented from the model metrics and qualitative detection examples and end-to-end workflow testing.

1.6 Research Questions

The following research questions will guide this thesis.

RQ1. Is it possible to achieve reasonable accuracy and sensitivity in identifying abnormal areas of prostate cancer in MRI scans using a YOLOv11 object detector?

RQ2. what value in the inclusion of negative MRI images do they provide to detection behavior that is more safe and why does this help to decrease over-detection risk?

RQ3. Can a trained YOLOv11 model be integrated into a complete patient and physician workflow using modern backend and automation tools?

RQ4. How to integrate physician approval so the AI answer can be reviewed prior to it being given to the patient?

RQ5. What are the main implementation challenges when combining model inference, authentication, database storage, workflow automation, and email delivery?

1.7 Hypothesis

The main idea behind the presented work is that a YOLOv11 detector trained on an adequately-prepared prostate MRI dataset, which is subsequently embedded in a physician-driven workflow, could facilitate rapid localization of suspicious regions of prostate cancer and a more streamlined reporting process compared to an automated model-only system, which would significantly benefit the safety of medical practice and reporting.

1.8 Research Contributions

The main contributions of this thesis are as follows.

1. A YOLOv11 based prostate MRI object detection pipeline that outputs bounding boxes and confidence values instead of only image-level labels.
2. A dataset preparation strategy that includes 415 negative MRI images to reduce false positive behavior in detection.
3. A detailed YOLOv11 preprocessing and post-processing pipeline for deployment-oriented inference.
4. A complete patient and physician workflow where the patient submits the MRI, the physician triggers the model, and the report is sent only after physician approval.
5. A working implementation using FastAPI, Firebase Authentication, MongoDB, n8n, and SMTP email delivery.
6. A structured evaluation using precision, recall, F1-score, mAP, confusion matrix, precision-recall curve, F1-confidence curve, qualitative detections, and workflow testing.

1.9 Scope of the Study

The scope of the study is limited to MRI image based detection of suspicious prostate cancer regions. The implemented model uses a single class named cancer. The work does not perform cancer grading, staging, survival prediction, recurrence prediction, or treatment recommendation. The project also does not replace radiologists or urologists. It is designed as a research prototype and decision support system. The physician remains responsible for reviewing and approving any result before it is released to the patient.

1.10 Project Workflow Summary

The workflow implemented is a controlled workflow sequence. Patient registers and uploads an MRI image using patient portal. Firebase Authentication authenticates the user and returns an identity token. FastAPI checks the identity of the token, saves the case in MongoDB, and exposes it to the physician portal. The doctor logs in, reads the case submitted, and initiates the inference of YOLOv11. The backend processes the image, runs the ONNX model, filters detections, creates the annotated output, and stores the report data. The physician reviews the output and approves the report. After approval, FastAPI triggers an n8n workflow. n8n validates the internal request, fetches the PDF report, emails it to the patient, and updates the case status.

1.11 Thesis Organization

Chapter 2 presents the literature review, including prostate cancer diagnosis, medical imaging, deep learning, object detection, YOLO based detection, workflow automation, and the identified research gap. Chapter 3 explains the methodology, dataset preparation, YOLOv11 processing pipeline, system design, database design, API design, and implementation tools. Chapter 4 presents the experimental results, training outputs, evaluation metrics, qualitative examples, portal implementation, report workflow, and discussion. Chapter 5 concludes the thesis, explains limitations, and presents future work. The appendices provide setup commands, API details, environment configuration, n8n workflow configuration, testing checklist, and supporting implementation material.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the background literature related to prostate cancer detection, MRI-based diagnosis, deep learning in medical image analysis, object detection, YOLO-based models, and clinical workflow automation. The purpose of the review is not only to show that deep learning can detect cancer from images, but also to identify the need for an integrated system that moves from image upload to physician review and report delivery.

2.2 Prostate Cancer as a Diagnostic Challenge

Prostate cancer develops in the prostate gland and is commonly associated with older age. It may remain asymptomatic in the early stage, which makes screening and early detection important. The Clinical diagnosis process typically starts with patient symptoms, PSA level or DRE. Imaging and biopsy can be used to assist in the diagnosis if there is any doubt. MRI is especially useful due to its high soft tissue contrast, which allowing the visualization of prostate anatomy and suspicion areas of prostate gland [3, 4]. One of the biggest problems with prostate cancer detection is that it may be very hard to find. The presence of a lesion can be influenced by scanner setting, image sequence, anatomical variation and lesion size. One of the difficulties is that of expert interpretation. Images must be scrutinised by a trained radiologist or physician and in busy clinical settings, the process can be time consuming. AI systems are thus appealing for use as decision support as they offer a second reading, can identify suspicious areas, and eliminate repetitive workload.

2.3 Conventional Diagnostic Methods

There are several tests that are utilized to assess prostate cancer. PSA blood levels are used as a screening indicator and are measured by PSAT. Digital rectal examination: physical exam to look for any abnormalities that can be felt. Biopsy is the definitive diagnostic technique, giving a tissue diagnosis. Imaging methods such as ultrasound, CT, and MRI provide visual information about the prostate gland and surrounding structures. Among these, MRI is widely used for lesion visualization and pre-biopsy decision support.

Table 2.1 summarizes common diagnostic methods and their relevance to this thesis.

Table 2.1: Common prostate cancer diagnostic methods and their relevance.

Method	Purpose	Relevance to this thesis
PSA test	Blood-based screening indicator	Provides clinical suspicion but does not localize lesions in images.
Digital rectal examination	Physical examination of prostate abnormalities	Depends on clinical examination and does not provide image localization.
Biopsy	Tissue sampling and confirmation	Important for final diagnosis, but invasive and outside the scope of this prototype.
MRI	Non-invasive visualization of prostate anatomy	Main input modality used for AI-assisted detection in this research.
Ultrasound	Imaging support, often used with biopsy	Not the selected modality in the implemented system.

2.4 MRI in Prostate Cancer Detection

MRI provides detailed soft tissue information and can help identify suspicious prostate regions. In prostate imaging, different sequences can provide complementary information about tissue appearance. A practical AI system can use image slices converted into PNG or JPEG format for model training. The thesis implementation uses MRI image frames prepared for object detection training. The image-based approach is suitable for YOLOv11 because the model expects a two-dimensional image and corresponding bounding box annotations.

The AI detection using MRI should be carefully preprocessed. Images should be in the correct size for the model input, be recognizable, be normalized, and be in the correct training format. The detector is trained to recognize the distribution and structure of the image in the pixels; any errors in the preprocessing may impact the performance of the model.

Therefore, image processing is not only considered as a minor step in the implementation of the proposed system but it is also considered an important part of the methodology.

2.5 Machine Learning and Deep Learning for Medical Images

Traditional machine learning approaches like support vector machines, decision trees, random forests and k-nearest neighbors can achieve a high accuracy when features are provided manually. But medical images are high-dimensional and with complex spatial patterns. Manual feature engineering could be time-consuming and may not include subtle lesion appearance. Deep learning overcomes this by learning the hierarchical visual features directly from the images themselves.

In medicine, convolutional neural networks (CNNs) have been extensively applied in classification and segmentation of medical images [5, 6, 9]. Some of the architectures used in medical imaging tasks include VGG, ResNet, EfficientNet, U-Net, Faster R-CNN and Mask R-CNN. These models can be used for classification, localization and segmentation. But the choice of a model will be dependent upon the task. The goal of a classification model is to predict a class label for the entire image. A segmentation model is used to predict the masks for individual pixels. An object detector predicts object bounding boxes and object class scores. The bounding box output is acceptable for this thesis because it provides a visualization of the region for which the physician has to review the image without needing masks at the pixel level.

2.6 Classification, Segmentation, and Object Detection

Medical image AI tasks can be divided into three broad categories. Classification answers whether a condition is present in an image. Segmentation outlines the exact region at pixel level. Object detection identifies approximate object locations using bounding boxes. Each approach has advantages and limitations.

Table 2.2: Comparison of common medical image AI tasks.

Task	Output	Strength	Limitation
Classification	Image-level label	Simple and efficient	Does not show lesion location.
Segmentation	Pixel-level mask	Highly detailed localization	Requires dense annotations and more effort.
Object detection	Bounding box and confidence	Localizes suspicious regions efficiently	Boundary is approximate rather than pixel-perfect.

The selected problem requires localization of suspicious cancer regions in MRI images. Object detection is therefore appropriate because it provides a region of interest that can be reviewed by the physician. This also supports report generation because bounding box coordinates and confidence values can be included in the report.

2.7 YOLO-Based Object Detection

YOLO models are single-stage object detectors designed for fast inference [7, 10]. Instead of generating region proposals in one stage and classifying them in another stage, YOLO predicts object boxes and class probabilities in a unified pass. This makes YOLO suitable for real-time systems and deployment scenarios where latency matters. The ability to output bounding boxes, class labels, and confidence values makes YOLO useful for lesion localization tasks.

In medical imaging, YOLO-based models have been used for lesion detection, cell detection, tumor localization, and anatomical abnormality detection [8, 10, 11]. And the reference work inspiring this thesis reported the use of YOLOv11 for prostate cancer detection and diagnosis [1]. The present work adopts the YOLOv11 framing but extends the contribution by implementing a patient and physician application around the model.

2.8 Why YOLOv11 for This Study

YOLOv11is chosen due to the need for local detection, quick inference, and seamless integration into a backend service. When the objective is to enable a physician dashboard, a real-time object detector is preferable over a heavy multi-stage pipeline. YOLOv11 is able to process an image and return the bounding boxes with a confidence score that is easy to display in the physician portal and put into a PDF report. The model is exported to ONNX for its inference on the backend [12]. ONNX is designed to deploy to other environments and enable the trained model to be deployed without keep-ing the entire training infrastructure within the production deployment. This makes the model more portable, and makes it easier to use the model in a web application.

2.9 Importance of Negative Samples

One of the typical issues in medical detection datasets is class imbalance. Positive images only can cause a model to believe that all images should have a lesion. This can result in false positive, which is an issue in medical screening as it can cause concern and extra workload for review. Empty label files are used for negative images, to teach the background appearance. The proposed dataset has 415 negative images added in 3 sets of training, validation and testing. As a result of this design, the model will learn to behave more safely, recognizing a valid MRI image to any detection area for cancer.

2.10 Evaluation Metrics in Object Detection

Object detection evaluation differs from simple classification because both class prediction and localization must be considered. A detection is considered correct only when the predicted bounding box sufficiently overlaps with the ground truth box and the class is correct. Intersection over union is used to measure overlap. Precision measures the fraction of predicted detections that are correct. Recall measures the fraction of true objects that were detected. F1-score combines precision and recall. Mean average precision summarizes detection quality across confidence thresholds and IoU thresholds.

Table 2.3: Evaluation metrics used in this thesis.

Metric	Meaning	Use in this thesis
Precision	Correct detections divided by all predicted detections	Measures false positive control.
Recall	Correct detections divided by all true objects	Measures missed lesion control.
F1-score	Harmonic mean of precision and recall	Summarizes balance between precision and recall.
mAP@50	Mean average precision at IoU 0.50	Measures detection quality at a common overlap threshold.
mAP@50-95	Mean average precision across multiple IoU thresholds	Provides stricter localization assessment.
Confusion matrix	Counts correct and incorrect detection outcomes	Provides interpretable error breakdown.
PR curve	Precision-recall trade-off across thresholds	Supports threshold understanding.
F1-confidence curve	F1-score across confidence thresholds	Supports confidence threshold selection.

2.11 Clinical Workflow Automation

A model alone is not enough for a useful healthcare prototype. A clinical workflow must support user roles, case submission, review, approval, reporting, and communication. In many AI papers, workflow integration is not considered. This creates a gap between experimental performance and practical deployment. The proposed system addresses this gap by including a patient portal, a physician portal, backend APIs, database storage, and n8n workflow automation.

The main use of n8n is it allows visual workflow orchestration. In this study, n8n receives an approved report trigger from FastAPI, validates the internal request, fetches report data, sends the report through email, and updates the case state. This separates email automation from the main inference backend and makes the reporting workflow easier to test and modify.

2.12 Authentication and Role Based Access

Security access control is a necessity in healthcare applications. Access to patients' cases and reports should be limited to patients themselves. Doctors should review or access cases which have been assigned or are available for review. Thus, Firebase Authentication is employed for user identity and role information is stored in the MongoDB [13, 14] database. With FastAPI, you can verify the Firebase token and protect endpoints with role based behavior. This is important because the results of models and medical images are sensitive data.

2.13 Research Gap Identified

The reviewed literature and project requirements show the following research gaps.

1. Many prostate cancer AI studies focus on classification and do not provide clear lesion localization.
2. Some detection studies report model metrics but do not implement a full patient-to-physician workflow.
3. Few systems combine AI detection with physician-controlled approval before patient report release.
4. Workflow automation with tools such as n8n is rarely integrated into prostate cancer detection prototypes.
5. Dataset imbalance and lack of negative images can increase false positive risk, but this issue is not always addressed in implementation-focused projects.
6. Deployment-oriented model serving using ONNX, FastAPI, authentication, storage, and report delivery is often missing from model-only studies.

2.14 Comparison with Related Approaches

Table 2.4 compares the present work with common categories of related systems.

Table 2.4: Comparison of related approaches and the proposed framework.

Approach	Main output	Strength	Limitation addressed by this thesis
CNN classifier	Image-level normal or abnormal label	Efficient classification	This thesis adds bounding box localization.
U-Net segmentation	Pixel-level lesion mask	Detailed boundary output	This thesis uses object detection to reduce annotation complexity and improve implementation simplicity.
Faster R-CNN or Mask R-CNN	Region proposals or instance masks	Strong localization capability	This thesis uses YOLOv11 for faster single-stage inference.
Model-only YOLO study	Detection metrics and plots	Demonstrates detection capability	This thesis adds patient portal, physician portal, storage, approval, and email workflow.
Manual clinical workflow	Physician review without AI support	Human judgment and clinical context	This thesis adds AI assistance and automation while preserving physician approval.
Proposed framework	Detection plus workflow automation	Localization, secure roles, physician review, report delivery	Provides an end-to-end prototype for clinical decision support research.

2.15 Summary

The literature supports the use of deep learning and YOLO-style detectors for fast medical image analysis. However, there remains a practical need for a complete workflow that connects the model with secure user access, case management, physician review, and report delivery. This thesis responds to that need by implementing a YOLOv11-based prostate MRI detection framework with a physician-controlled workflow and n8n-based automation.

Chapter 3

Research Methodology and System Design

3.1 Introduction

This chapter describes the research methodology followed to design, train, evaluate and implement the proposed prostate cancer detection framework. The methodology features data preparation, image preprocessing, training and exporting the YOLOv11 model, backend implementation, database design, frontend design, n8n automation and end-to-end testing. This study is an experimental and implementation study. The experimental part compares the results of the YOLOv11 detections. The implementation part shows the use of the trained model in a realistic patient/physician workflow.

3.2 Research Design

The research method is an applied experimental research. The input to the project is an MRI image dataset and the problem is reduced to a single-class object detection problem. The class name is cancer. The standard object detection metrics are used to train and evaluate the model. The trained detector is integrated in web based software prototype after model evaluation. This lets us also compare the performance of the model and its usability to the usability of the system.

The methodology is arranged into the following phases.

1. Dataset preparation and final split definition.
2. Image preprocessing and annotation format verification.
3. Training and validation of YOLOv11.
4. Test evaluation using curves, confusion matrix, and qualitative images.

5. ONNX export for backend inference.
6. FastAPI backend implementation.
7. Firebase Authentication integration.
8. MongoDB storage integration.
9. Patient and physician frontend implementation.
10. n8n workflow automation for approved report emailing.
11. End-to-end workflow testing.

3.3 Dataset Description

The Prostate MRI images that have been preprocessed for object detection are included in the data set used in this study. Cancer annotated images are considered positive images. The images that don't have a cancer annotation are called negative images. There are 3,930 images in the final dataset. Out of these, 3,515 are positive and 415 are negative. The negative images are also critical as they will help the model understand that some MR images are not cancerous. This decreases the likelihood of making predictions of cancer in every image.

3.4 Final Dataset Split

In this thesis, the final split of the dataset is presented in Table 3.1. Most images are included in the training subset and are used to update weights of the model. The validation subset is used to check out the behavior of the model during training and determine the best checkpoint. A test set is reserved for testing at the end of the training.

Table 3.1: the Final dataset split use in this study.

Split	Positive	Negative	Total	Split share	Negative share
Train	2,454	300	2,754	70.1%	10.9%
Validation	514	55	569	14.5%	9.7%
Test	547	60	607	15.4%	9.9%
Total	3,515	415	3,930	100%	10.6%

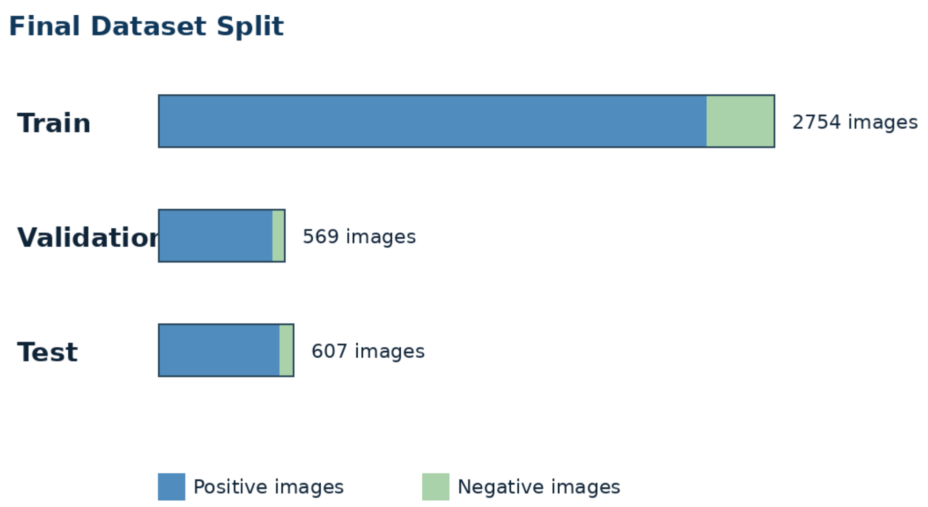


Figure 3.1: Final dataset distribution across training, validation, and test subsets.

3.5 Rationale for the Split

The The data split was optimized for model learning and to ensure good evaluation. The training set consists of 70.1 percent of the data which is sufficient to have enough samples for the detector to learn the appearance of the lesions and the patterns of the background. The validation set is included as The data amount of 14.5 percent is enough to be used for training progress monitoring and overtraining detection. The test set includes 15.4 per cent of the data and is used for final evaluation on new data.

The split also maintains the negative samples in each of the three subsets. This is significant because one should not only have negative data in training. When there are no negative images in the validation and test sets, the evaluation might not reflect the false positive behavior. Incorporating negative images with the validation and test sets makes the validation and test more realistic.

3.6 YOLO Annotation Format

YOLO object detection uses text files where each line represents one bounding box. In this study, the class ID 0 is used for cancer. Each annotation line contains the class ID, normalized x-center, normalized y-center, normalized width, and normalized height. The values are normalized by image width and height.

```
0 x_center y_center width height
```

For negative images, the label file exists but remains empty. This tells the model that the image is valid but contains no annotated object. Maintaining empty label files for negative images is important because missing labels may be interpreted as a dataset error by some training pipelines.

3.7 Data Quality Considerations

Dataset quality is critical in medical imaging. The following quality checks were considered during preparation.

1. Each image must have a corresponding label file.
2. Positive images must contain at least one bounding box annotation.
3. Negative images must contain an empty label file.
4. Images must be readable and convertible into the expected RGB or grayscale format.
5. Bounding boxes must remain inside image boundaries.
6. Train, validation, and test images must not be duplicated across splits.
7. Augmentation must be applied only to the training process, not by duplicating validation or test images.

3.8 DICOM to PNG Conversion

Mostly MRI scans can be saved as DICOM files. Typically, YOLO training pipelines use common image formats like PNG or JPEG. Thus, it is possible to convert the DICOM images in PNG format prior to training. The conversion process will read the DICOM pixel array, normalize the intensity, convert it to 8-bit representation, and save the image as a PNG. This step is crucial, since the range of the raw DICOM intensity values can be too large to be contained within the typical range (0 to 255) of common computer vision models.

For this thesis, the data for training was already available in image format. The methodology of the project is, however, conducive to future extension of the DICOM conversion. The DICOM conversion process should maintain the image quality and not alter the geometry of the lesions. Atypical conversion pipeline is presented in Appendix B.

3.9 Preprocessing Pipeline

The preprocessing pipeline transforms an uploaded MRI image into an image format suitable for inference using YOLOv11. It includes validation, decoding, resizing, normalization, dimension reordering, and tensor preparation. The purpose of preprocessing is to ensure that every input image has a consistent format before being passed into the model.

Table 3.2: Preprocessing steps used before YOLOv11 inference.

Step	Description
Input validation	Confirms that the uploaded file is an image and can be decoded.
Color handling	Converts the image into the expected channel format.
Resizing	Resizes the image to 416 x 416 pixels for model input.
Normalization	Scales pixel values to a numeric range suitable for neural network inference.
Tensor conversion	Converts the image into batch, channel, height, and width format.
ONNX input mapping	Sends the tensor to the ONNX Runtime session using the correct input name.

3.10 Training Augmentation Strategy

By using augmentation, the detector is given some variation in the image appearance, thus making it more general. Augmentation is a procedure that needs to be carefully chosen since MRI images depict anatomical structures. Overrotation or distortion may alter anatomical orientation and decrease realism. So, an augmentation of moderate intensity is presented.

Table 3.3: Augmentation strategy for training use.

Augmentation	Setting	Reason
Horizontal flip	Moderate probability	Helps the model learn symmetrical anatomical variations.
Small rotation	Limited angle	Simulates minor patient positioning variation without distorting anatomy.
Brightness adjustment	Controlled range	Simulates scanner intensity variation.
Scale variation	Small range	Supports robustness to lesion size differences.
Mosaic	Training only	Improves multi-scale generalization during training.
Heavy rotation	Disabled	Avoids unrealistic prostate orientation.

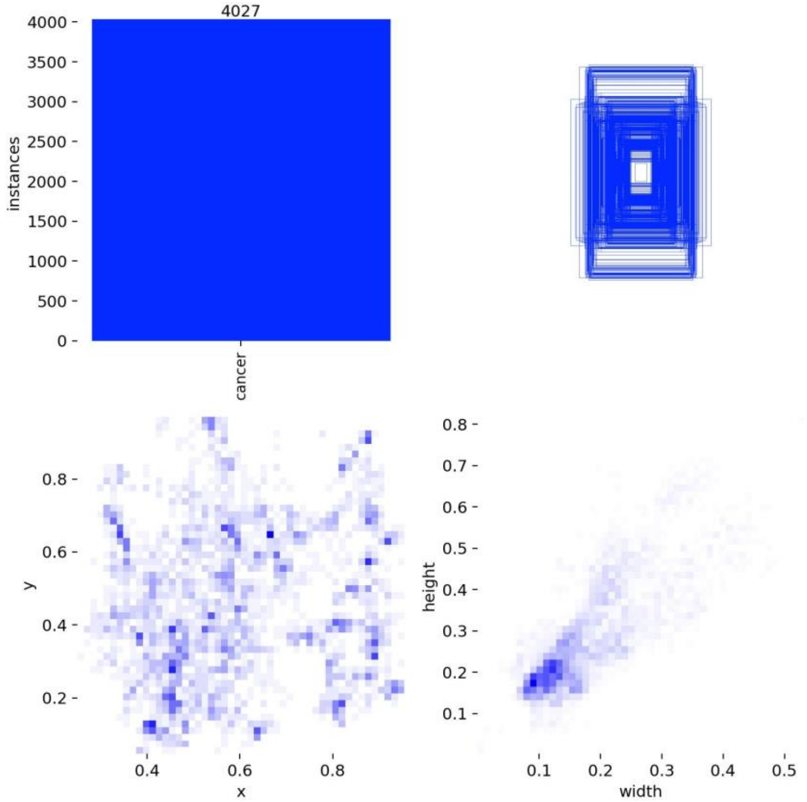
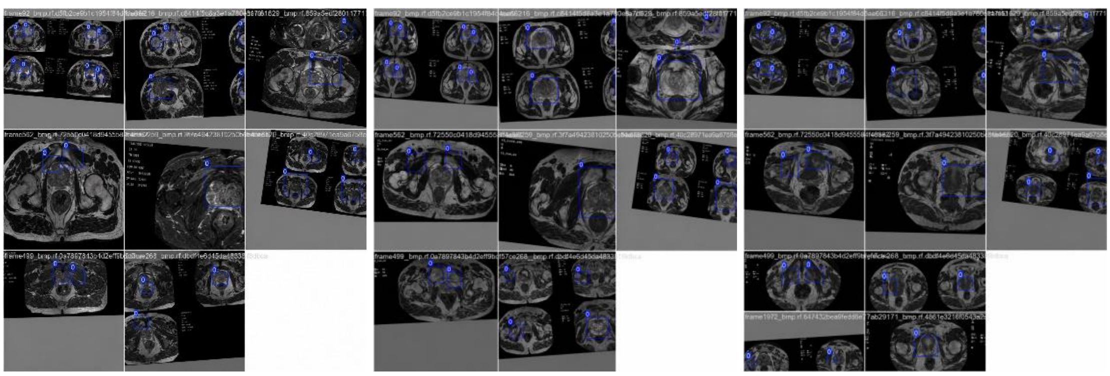


Figure 3.2: Label distribution and bounding box patterns generated from the training run.

3.11 Training Sample Visualization

Training batch visualization is used to ensure the dataset is being read properly, and that the bounding boxes match up with the suspicious areas. During the YOLO training run, some representative training batches are generated and shown in figure 3.3. These pictures will be helpful in the identification of annotation errors prior to use of numerical metrics.



(a) Batch 0 (b) Batch 1 (c) Batch 2

Figure 3.3: Representative training batches used to visually inspect image and label loading.

3.12 YOLOv11 Model Selection

YOLOv11 is chosen because the project does not aim to classify images, but to detect them locally in real time. The model identifies a "suspicious" area and provides a bounding box with a confidence score. This output can easily be shown in the doctor's portal and be added to the PDF report. A segmentation model might give finer-grained boundaries, though it would involve more fine-grained annotations, and a different evaluation process. YOLOv11 is a good balance between speed, localization and simplicity of implementation.

3.13 Training Configuration

The model was trained on the basis of one class. The resolution of input was 416 X 416 pixels. This resolution was chosen as a compromise for the delivery of inference speed and visibility. The higher the resolution, the better will be the localization of small lesions, but the amount of computation goes up. A lower resolution can lead to faster speeds, but may compromise the detection quality.

Table 3.4: YOLOv11 training configuration.

Parameter	Value
Task	Object detection
Class count	1
Class name	cancer
Input image size	416 x 416
Training images	2,754
Validation images	569
Test images	607
Confidence threshold	0.50
Export format	ONNX
Inference provider	CPUExecutionProvider
Primary backend framework	FastAPI
Workflow automation	n8n
Authentication	Firebase Authentication
Database	MongoDB Atlas

3.14 YOLOv11 Processing and Inference Pipeline

The processing pipeline, YOLOv11, is at the heart of the project. It converts a user uploaded MRI to an annotated result that can be shown to a physician. The pipeline is depicted in Fig.3.4.

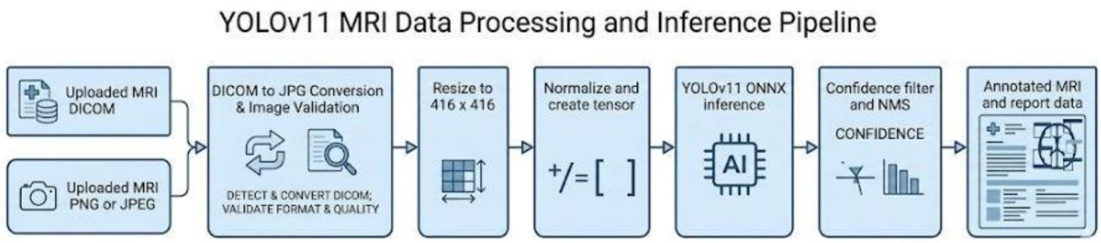


Figure 3.4: YOLOv11 data-processing and inference pipeline used in the proposed Design.

The pipeline madeuo of the following steps.

1. The patient uploads the MRI image and it becomes a case record.
2. The physician/Expert opens the case and punch the AI model.
3. The backend check the image format and decodes the image.
4. The image is resized to the model input resolution.
5. Pixel values are normalized and converted into an input tensor.
6. ONNX Runtime executes the YOLOv11 model.
7. Raw outputs are converted into bounding boxes, confidence scores, and class labels.
8. Low confidence predictions are filterd out using the selected confidence threshold.
9. The original image is annotated with bounding boxes.
10. The detection information and the annotated image are saved for the physician to review and report.

3.15 Post-Processing Strategy

Post-processing transforms raw outputs from the model to meaningful detections. It filters out boxes with confidence values which are below the threshold, maps back the predicted coordinates to the original image space, and eliminates overlapping and redundant boxes. The confidence threshold chosen is $0.50 for prototype implemented. This threshold is a compromise between sensitivity and false positive control. Lowering the threshold in medical screening can result in higher number of recalls and false-positive results. A higher threshold will decrease the number of false positives but may not detect subtle lesions. The model output is not automatically shared with the patient, the physician review stage helps with this trade-off.

3.16 ONNX Export and Inference

The trained model is converted to ONNX after training. The ONNX model can be served in its own environment in addition to the training environment. The ONNX Runtime loads the model file and infers on the CPU Execution Provider in the backend. This allows the local testing without using a GPU server. If the real-time throughput requirements are going to increase at deployment time, GPU acceleration can be added to the solution.

3.17 System Architecture

The proposed system consists of user, authentication, front end, back end, AI inference, database, automation and email layers. The overall architecture is depicted in Figure 3.5 shows the high-level architecture.

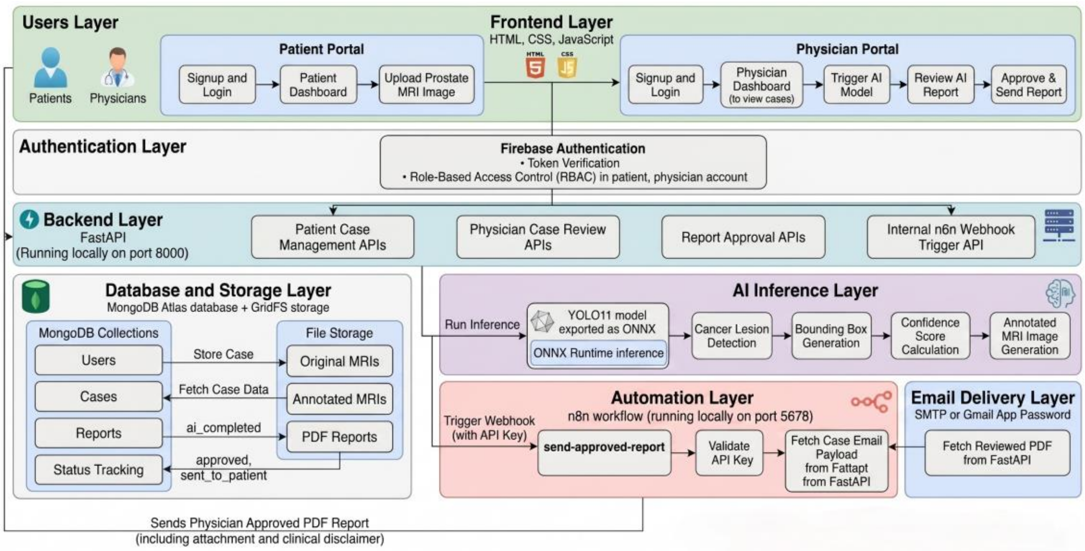


Figure 3.5: System architecture of the patient and physician prostate cancer detection workflow.

3.18 Technology Stack

The implementation uses a lightweight but complete technology stack. Table 3.5 summarizes the main tools.

Table 3.5: Technology stack used in the implementation.

Layer	Technology	Purpose
Frontend	HTML, CSS, JavaScript	Provides patient and physician portals.

Layer	Technology	Purpose
Authentication	Firebase Authentication	Handles signup, login, and identity token generation.
Backend	FastAPI	Provides APIs for cases, roles, inference, reports, and internal workflow triggers.
AI inference	YOLOv11 ONNX model	Detects suspicious cancer regions from MRI images.
Runtime	ONNX Runtime	Executes exported model for backend inference.
Image processing	OpenCV and Python utilities	Validates, resizes, normalizes, annotates, and encodes images.
Database	MongoDB Atlas	Stores user profiles, case records, detections, and report metadata.
File storage	GridFS or binary storage	Stores original images, annotated images, and PDF reports.
Automation	n8n	Sends approved reports to patients and updates case state.
Email	SMTP or Gmail App Password	Delivers reviewed PDF reports.

3.19 Backend API Design

FastAPI is used because it supports modern Python APIs, automatic documentation, request validation, and integration with machine learning code [15]. The backend verifies Firebase tokens and checks user roles before allowing actions. Table 3.6 summarizes the major API groups.

Table 3.6: Main API groups implemented in the backend.

Group	Example endpoint	Purpose
Health	`/health`	Confirms model loading, Firebase readiness, MongoDB status, and workflow configuration.
Authentication	`/auth/register-profile`	Stores user role profile after Firebase signup.
Patient	`/patient/cases`	Allows patient case upload and case history retrieval.

Group	Example endpoint	Purpose
Physician	`/physician/cases`	Allows physician case listing, claim, inference, approval, and sending actions.
Image	`/image/...`	Streams original or annotated case images.
Report	`/report/.../pdf`	Generates or serves PDF reports.
Internal	`/internal/cases/...`	Allows n8n to fetch report payloads and mark cases as sent using an internal API key.

3.20 Database Design

MongoDB stores the operational data required by the workflow [14]. The database design separates users, cases, detections, and report information. Binary data such as images and PDFs can be stored through GridFS or an equivalent binary storage strategy. Table 3.7 presents the key collections.

Table 3.7: MongoDB collection design.

Collection	Important fields	Purpose
users	uid, email, role, display name, created at	Stores Firebase user profile and application role.
cases	case id, patient id, physician id, status, timestamps	Tracks patient MRI cases and workflow state.
images	original image id, annotated image id	Stores or references uploaded and annotated images.
detections	class, confidence, bbox, model version	Stores model output for physician review.
reports	pdf id, approval status, email status	Stores generated report metadata and delivery state.
logs	event type, user id, case id, timestamp	Supports audit and debugging.

3.21 Case State Model

The workflow uses a state model to prevent uncontrolled report release. The case begins as submitted, then moves to claimed or under review, then AI completed, then approved, and finally sent to patient. Table 3.8 describes the states.

Table 3.8: Case status model used in the physician controlled workflow.

Status	Meaning
submitted	Patient uploaded the MRI image and the case is waiting for physician review.
claimed	Physician opened or accepted the case for review.
ai_completed	YOLOv11 inference was executed and detection results are available.
approved	Physician reviewed the result and approved the report for release.
sent_to_patient	n8n delivered the approved PDF report to the patient email address.
rejected	Physician rejected the AI result or report and did not release it.

3.22 Frontend Design

The front-end is created using HTML, CSS and JavaScript. It is enough to have a simple static frontend, as the system is based on Firebase as an authentication service and FastAPI as a backend logic service. Registration, logon to the patient portal, upload of MRI, and viewing of MRI report. Allows physician to view submitted cases, trigger models, review AI results, approve case, and send a report.

3.23 Patient Portal Workflow

The main goal of the patient portal is to maintain a simple patient interaction. The patient logs in, uploads MRI image and sees case status. The patient does not activate the AI model and the report is only sent out when the physician approves. This will not display the final result from AI that hasn't been reviewed.

3.24 Physician Portal Workflow

The physician portal is the control center for the diagnostic review process. The physician reviews cases submitted, makes inferences, reads annotated images, reads the detection confidence values, amends the report with clinical notes (if necessary), approves the report and forwards it to the patient. Such a role separation enables a safer AI-assisted workflow.

3.25 n8n Automation Design

After a physician's approval, n8n will used. The process starts with a webhook called send-approved-report. The case identifier and internal key are sent to the webhook by FastAPI. The validation node performs its own key validation. Then n8n takes the email content and PDF report from FastAPI, sends it via SMTP and marks the case as sent through FastAPI. Figure 3.6 shows the implemented n8n workflow.

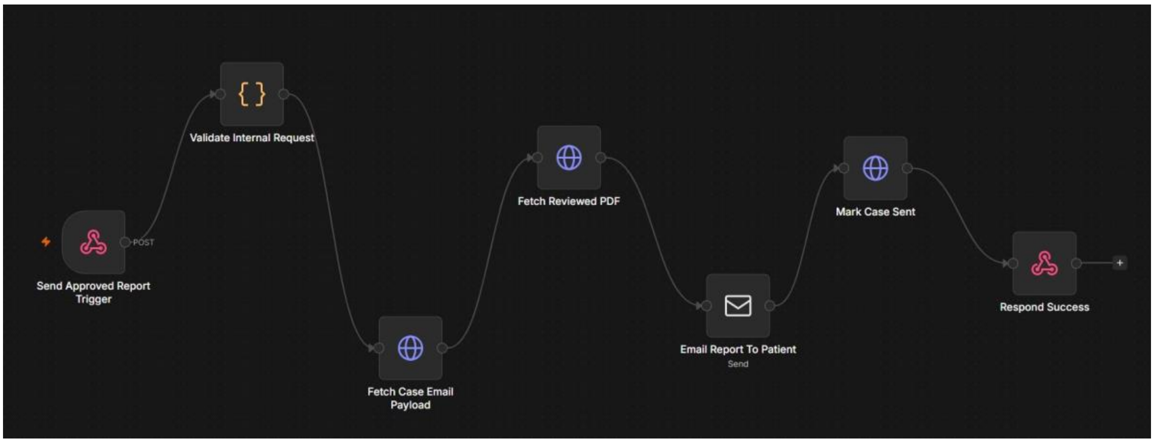


Figure 3.6: n8n automation workflow used for physician approved report delivery.

3.26 Security Design

It has internal API key protection, role based authorization, and Firebase authentication. Firebase offers user identity [13]. MongoDB stores the roles of the users. n8n internal endpoints are protected by an API key, and access to the report payloads is restricted by fastAPI. In the production environment, additional requirements would include HTTPS, more robust secrets management, audit logs, encryption at rest, and limited network access.

3.27 Ethical Considerations

The system is designed for decision support and research. It does not provide an independent medical diagnosis. The physician is responsible for interpretation and report approval. The data set comprises image data, which is prepared for research purposes. Proper data governance, patient consent (where appropriate), institutional approval and clinical validation would be needed for future clinical use.

3.28 Testing Methodology

Testing was performed at model and system levels. Model testing included training curve inspection, validation metrics, test confusion matrix, PR curve, F1-confidence curve, and qualitative prediction visualization. System testing included Firebase login, patient signup, physician signup, MRI upload, case visibility, AI inference, report approval, n8n trigger, email delivery, and final case status update. This two-level testing approach ensures that the model works and that the surrounding application can execute the complete workflow.

3.29 Summary

This chapter described the full methodology used to build the proposed framework. The method combines dataset design, YOLOv11 detection, ONNX inference, FastAPI services, Firebase authentication, MongoDB storage, frontend portals, and n8n automation. The next chapter presents the implementation outputs, model results, workflow validation, and discussion.

Chapter 4

Implementation, Results, and Discussion

4.1 Introduction

This chapter presents the implementation of the proposed prostate cancer detection framework and discusses the experimental results. The chapter covers backend implementation, frontend implementation, n8n automation, model training behavior, validation metrics, test evaluation, qualitative outputs, and workflow testing. The purpose is to show both the performance of the YOLOv11 detector and the working behavior of the complete patient and physician system.

4.2 Implementation Overview

The system was implemented as a local prototype with three main running services. The FastAPI backend runs on port 8000, the frontend runs on port 5500, and n8n runs on port 5678. The YOLOv11 model is loaded as an ONNX file from `models/best.onnx`. Firebase Authentication is enabled for login and token verification. MongoDB stores users, cases, reports, and detection metadata. n8n performs report email delivery after physician approval.

The local service endpoints are summarized in Table 4.1.

Table 4.1: Local service endpoints used during implementation and testing.

Service	URL	Purpose
FastAPI	`http://127.0.0.1:8000`	Backend APIs and model inference
Frontend	`http://127.0.0.1:5500`	Patient and physician portals
n8n	`http://127.0.0.1:5678`	Workflow automation and email delivery

4.3 Implemented End-to-End Workflow

Thefinal workflow is physician controlled, which was intentional to prevent the automatic release of AI-generated results to patients without medical review. This system starts with patient creating an account using Firebase Authentication which is then followed by uploading a prostate MRI image using the patient portal. The backend is created using FastAPI and verifies the authentication token and securely stores the case in MongoDB. Once the data is saved, a doctor logs into the doctor portal and accesses the submitted case. The doctor then manually activates the inference based on the YOLOv11 model for analysis.

After being triggered, the back-end will undertake a series of operations such as MRI image preprocessing, model prediction with ONNX and post-processing and annotation of the predicted regions. The detection outputs generated are then presented in the physician portal for the physician to review. The doctor checks the outcome and either accepts or rejects the report that is generated. Once this report is approved, FastAPI will trigger an n8n webhook that will start a workflow that will pull the report down and send it to the patient's email address. Lastly, the system changes the case status to "sent_to_patient", and tracks the case through a completed workflow.

Patient uploads a prostate MRI image from the patient portal.

4.4 Backend Health Verification

The back-end has a health check endpoint to validate major integration. The health response checks if the model is loaded, authentication is enabled, MongoDB is configured, Firebase is ready, and if the n8n workflow settings are present. It was helpful during development as it enabled fast identification of the missing model files, incorrect Firebase credentials, missing MongoDB connection, and n8n webhook configuration problems..

Table 4.2: Backend health checks used during implementation.

Health field	Expected meaning
`model_loaded`	Confirms that the ONNX model file is loaded successfully.
`auth_enabled`	Confirms that protected endpoints require Firebase verification.
`firebase.ready`	Confirms that Firebase Admin credentials are valid.
`mongo_enabled`	Confirms that MongoDB support is enabled in configuration.
`mongodb.connected`	Confirms that the backend can connect to MongoDB Atlas.
`n8n_send_report_configured`	Confirms that the approved report webhook URL is configured.
`internal_api_key_configured`	Confirms that internal n8n communication is protected.

4.5 Model Training Results

The training results display the model's progress in training. The training loss reduced over time and validation precision, recall, and mean average precision (mAP) increased. The export training curves are shown in figure 4.1. The curves show evidence of the model learning useful features and stable validation performance.

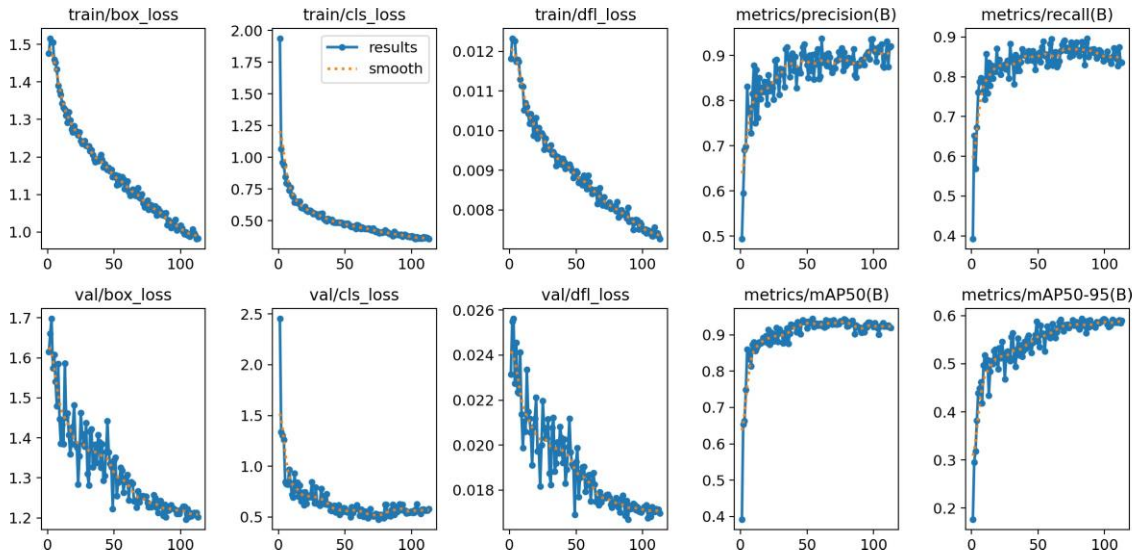


Figure 4.1: YOLOv11 training and validation curves generated during model training.

4.6 Best Validation Metrics

The Table 4.3. presents the best validation metrics obtained during the training process. These values are a summary of the performance of the model on the validation set. Precision means the number of cancer diagnoses that are identified correctly. Recall: Number of true cancer objects detected. The detection quality is measured by mAP@50, with IoU threshold set as 0.50. mAP@50-95 is a stricter criterion as it is an average of several IoU thresholds.

Table 4.3: Best validation metrics observed during YOLOv11 training.

Metric	Best value
Precision	93.79%
Recall	89.68%
mAP@50	94.43%
mAP@50-95	59.36%

The precision value is high, indicating that the model has high accuracy in identifying false positive detections. Most of the lesions are detected as shown in the recall value with some missed detections. The difference between mAP@50 and mAP@50-95 indicates the model has good ability to locate the overall lesion area but is less efficient at localization when the condition is set to be more stringent. This is normal in medical images where there can be variations in annotations and subtle boundaries between lesions.

4.7 Training Loss Interpretation

There are several loss components to YOLO training. The localization error is measured by box loss. The classification loss is a measure of class prediction error. Bounding box quality is affected by distribution focal loss. A training model that is useful should have a decrease in loss values and an increase in validation metrics over time. The training and the validation curves in this study show that the model did not just fall apart and make random predictions and thus learned the task.

Table 4.4: Interpretation of YOLO training outputs.

Output	Interpretation
Box loss	Lower value indicates improved bounding box localization.
Class loss	Lower value indicates improved cancer class prediction.
DFL loss	Supports more accurate bounding box distribution learning.
Precision curve	Shows how correct predictions improve during training.
Recall curve	Shows how missed detections reduce during training.
mAP curve	Shows detection quality across confidence thresholds.

4.8 Confusion Matrix Analysis

The confusion matrix provides an interpretable summary of correct and incorrect predictions. Figure 4.2 shows the test confusion matrix. The model produced 842 true cancer detections, 91 false positive cancer detections, and 118 false negative missed detections. These values were used to calculate object-level precision, recall, and F1-score.

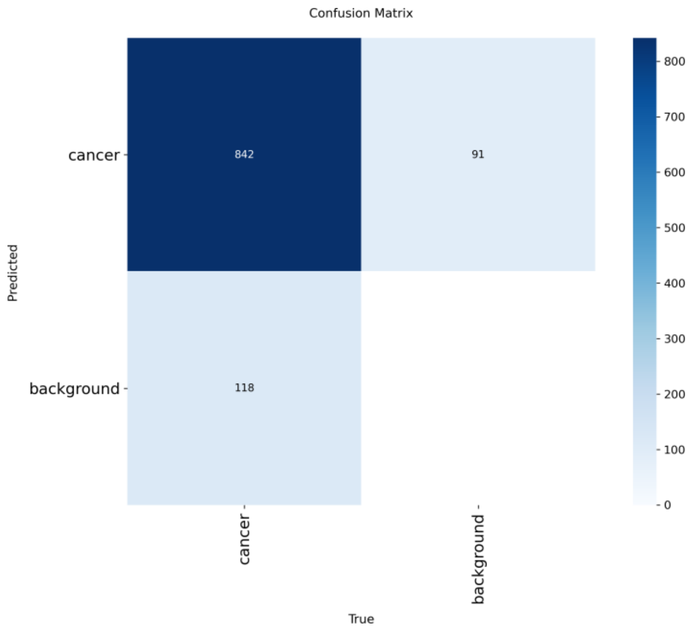


Figure 4.2: Test evaluation confusion matrix.

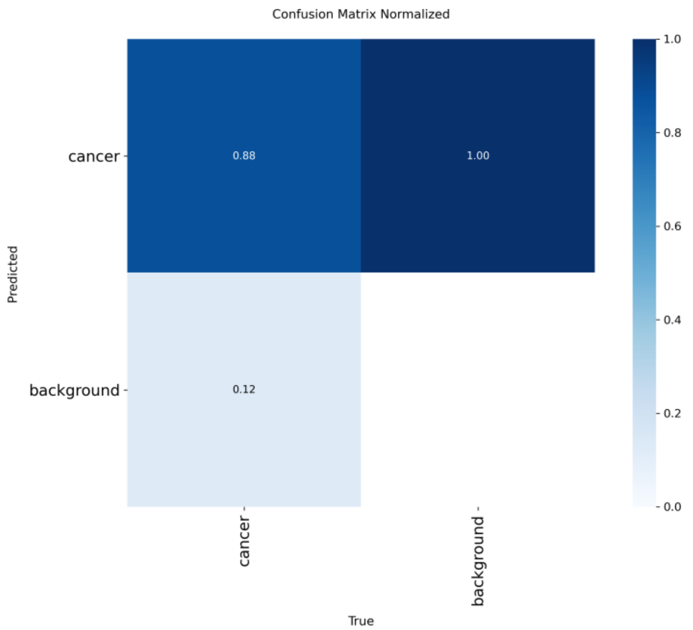


Figure 4.3: Normalized test evaluation confusion matrix.

4.9 Object-Level Test Metrics

Using the test confusion matrix values, object-level performance was calculated as follows.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Where $TP = 842$, $FP = 91$, and $FN = 118$. The resulting metrics are shown in Table 4.5.

Table 4.5: Object-level test metrics computed from the confusion matrix.

Metric	Value
True positive detections	842
False positive detections	91
False negative detections	118
Precision	90.25%
Recall	87.71%
F1-score	88.96%

Results from the test indicate that the detector is not perfect, but it is a good detector. A false positive is something that occurs when the model classifies a region as cancer when it is actually not. With a false negative, there are portions of the annotated cancer area that are not detected. In cases of clinical decision support system, both errors are of significance. Physician review is therefore necessary as it checks the reports prior to their release, ensuring they are verified by humans..

4.10 Precision-Recall Curve

The precision-recall curve depicts the relationship between precision and recall for various confi-dence thresholds. It has proved to be useful for imbalanced detection tasks, as it highlights the balance between the number of lesions detected and the number of false alarms. Figure 4.4 shows the precision-recall curve for the test evaluation.

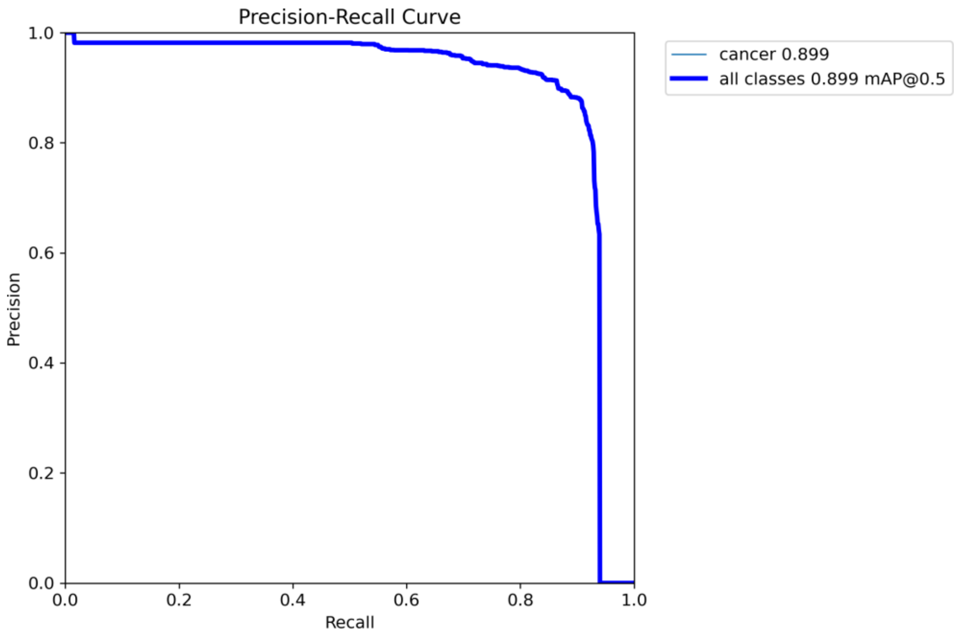


Figure 4.4: Precision-recall curve for test evaluation.

A model that has high precision and high recall for a variety of thresholds is desired. The curve helps in the selection of thresholds. A 0.50 was chosen for this prototype as a compromise between sensitivity and precision. This threshold may be modified according to the clinical validation preferences and screening goals in future clinical validation.

4.11 F1-Confidence Curve

F1-confidence curve is the curve that illustrates the evolution of F1 scores with the increase of confidence. It assists in establishing a threshold that is balanced with false positives and false negatives. Figure 4.5 shows the F1-confidence curve.

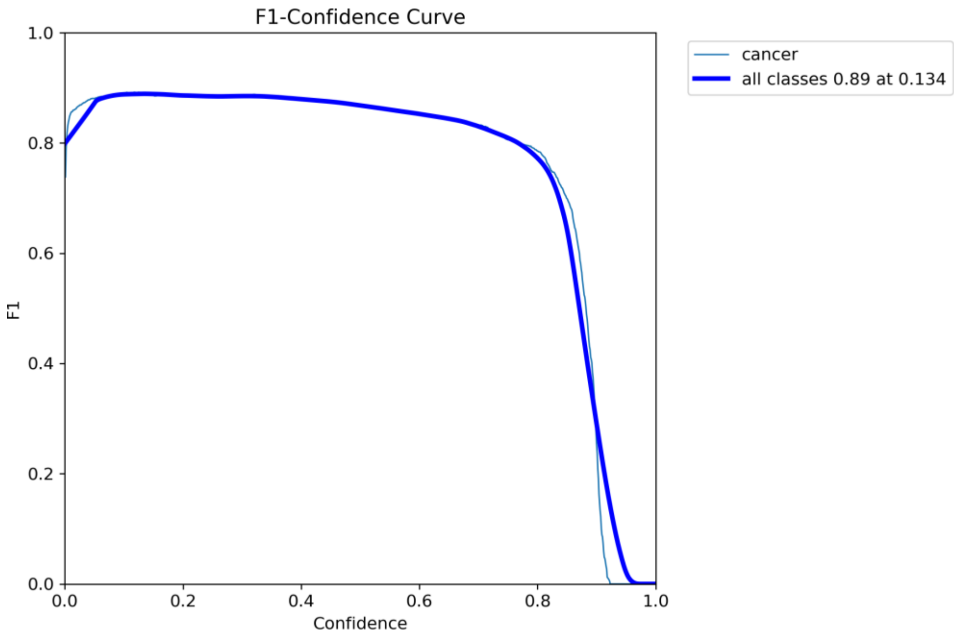
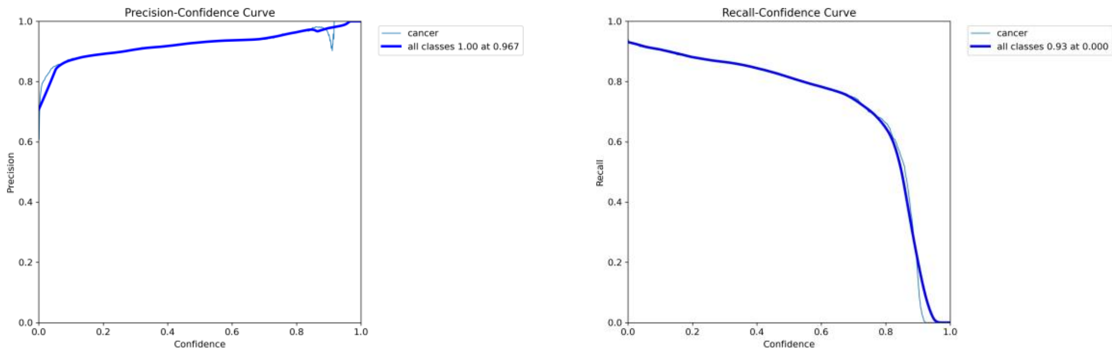


Figure 4.5: F1-confidence curve generated during test evaluation.

The F1 curve is helpful because the precision and recall are usually inversely related. Lower confidence threshold tends to be associated with a higher level of recall, but may also result in more false positives. A higher threshold tends to decrease recall and increase precision. The chosen threshold should correspond to the purpose for which the sensor will be used. If there is a need for screening support, then sensitivity would be the criterion that is important. Physician review is an extra safety net for report release.

4.12 Precision and Recall Confidence Curves

Precision and recall confidence curves show a more detailed picture of the behavior of the thresholds. The precision and recall curves of the test evaluation are displayed in Figure 4.6. These curves are used to understand the change in model behavior with varying confidence thresholds.



(a) Precision curve

(b) Recall curve

Figure 4.6: Precision and recall confidence curves for test evaluation.

4.13 Qualitative Detection Results

While quantitative metrics are required, there are also qualitative examples that are of importance for medical image detection. It is important that the doctor be able to visually check the predicted box and make sure that it is clinically sensible. Figure 4.7 presents sample predictions from the test set. Here are some predictions of how model is able to draw bounding boxes over suspicious areas.

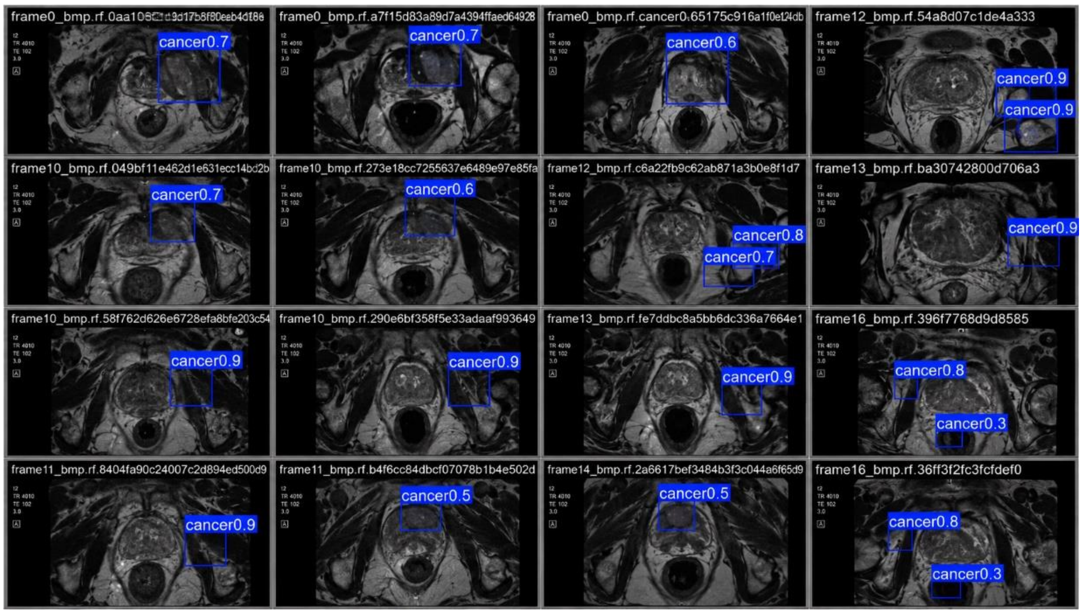
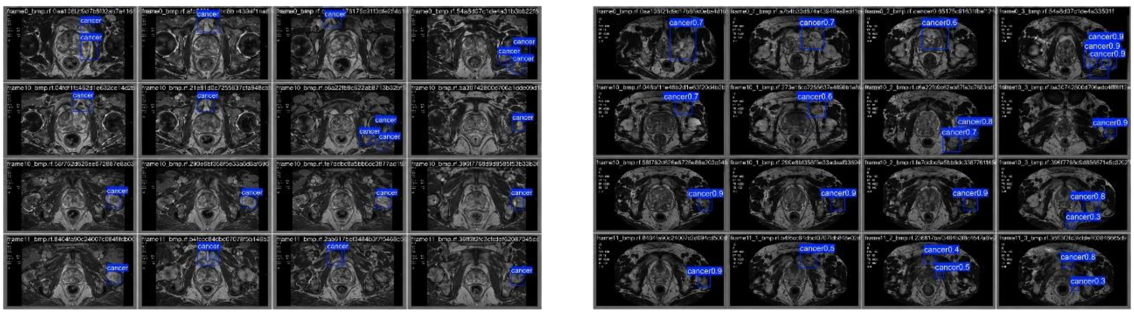


Figure 4.7: Sample test predictions generated by the YOLOv11 detector.

4.14 Ground Truth and Prediction Comparison

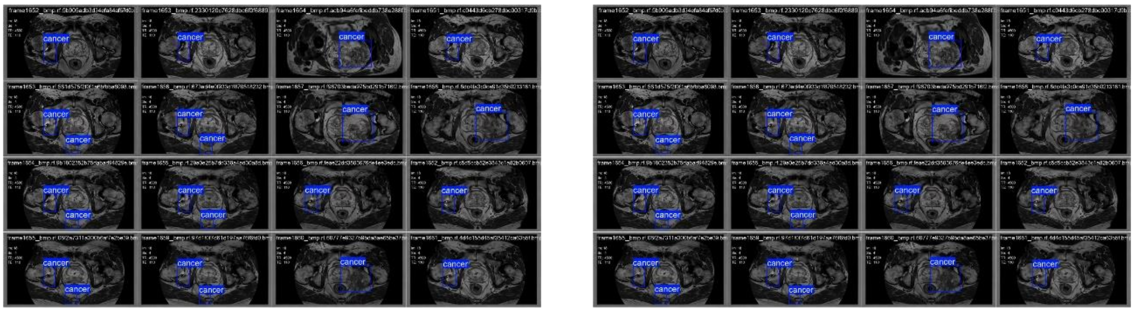
Compare the ground truth data with the predictions.Compare the predicted to the ground truth data. One way to check the behavior of a model is to compare the ground truth labels with the model's outputs. Figures 4.8, 4.9, and 4.10 show test batch label and prediction comparisons. These numbers provide information on the model's ability to match detections to annotated regions, as well as to see if additional false detection events are occurring.



(a) Ground truth labels

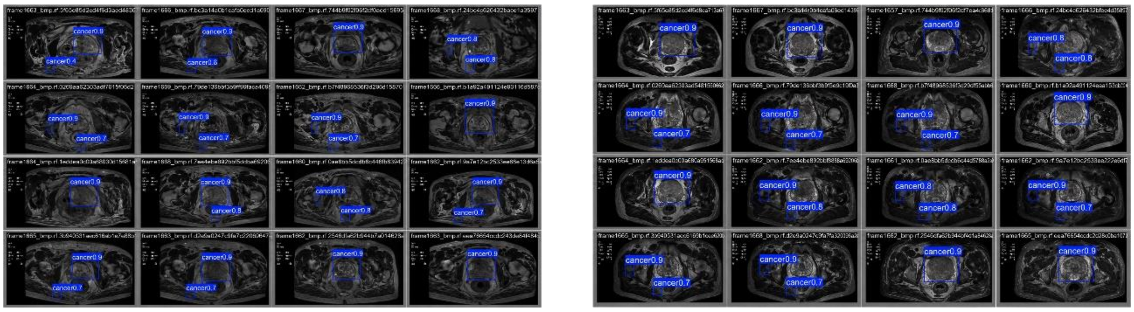
(b) Model predictions

Figure 4.8: Ground truth and prediction comparison for test batch 0.



(a) Ground truth labels (b) Model predictions

Figure 4.9: Ground truth and prediction comparison for test batch 1.

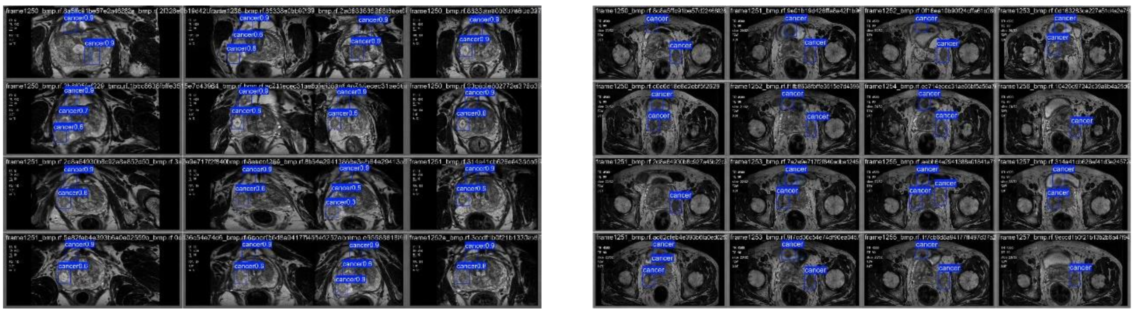


(a) Ground truth labels (b) Model predictions

Figure 4.10: Ground truth and prediction comparison for test batch 2.

4.15 Validation Batch Inspection

During training, the validation batch inspection was also employed. The results of the training run and the validation labels are presented in Figure 4.11 This inspection assisted in ensuring that the model was learning meaningful localization, not spurious detections.



(a) Validation labels (b) Validation predictions

Figure 4.11: Validation label and prediction comparison from the training run.

4.16 Patient Portal Implementation

The patient portal enables users to sign up, log in, upload MRI pictures and observe case status. It's deliberately stripped down as the patient shouldn't have to deal with the model parameters or trigger the diagnosis. Patient uploads case, the physician assumes responsibility for review. Figure 4.12 shows the implemented patient portal.

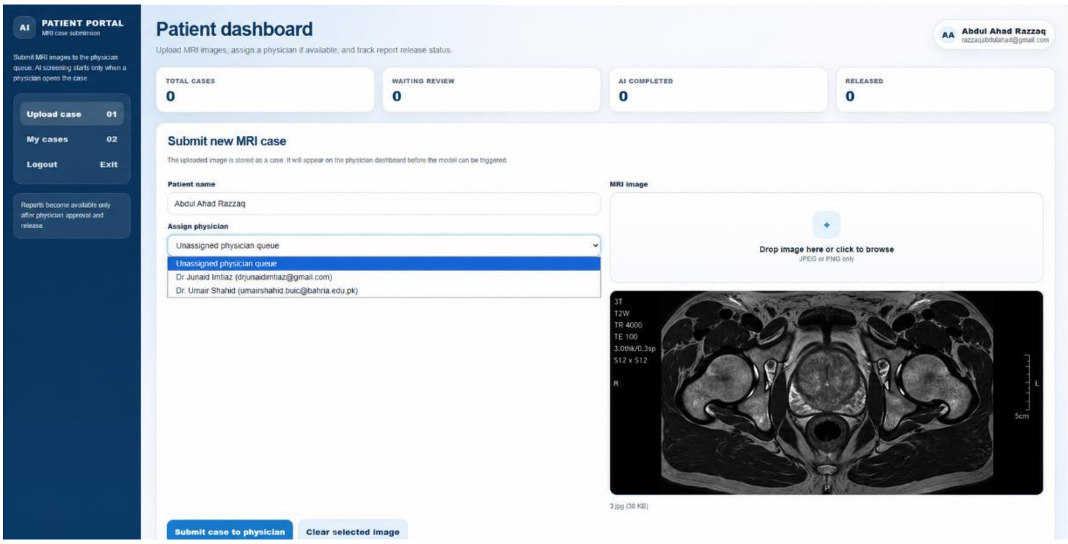


Figure 4.12: Patient portal used for MRI upload and case status viewing.

4.17 Physician Portal Implementation

The physician portal is for clinical review. It enables the doctor to check submitted cases, launch AI inference, check on the annotated image, view the detection values, approve the report, reject the report, and send the final report to the patient. Figure 4.13 shows the implemented physician portal.

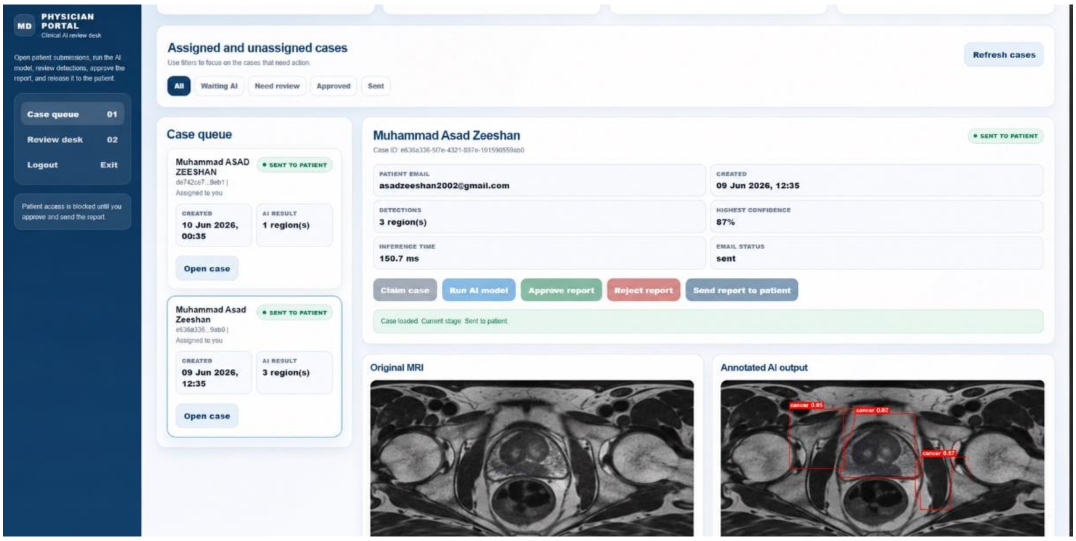
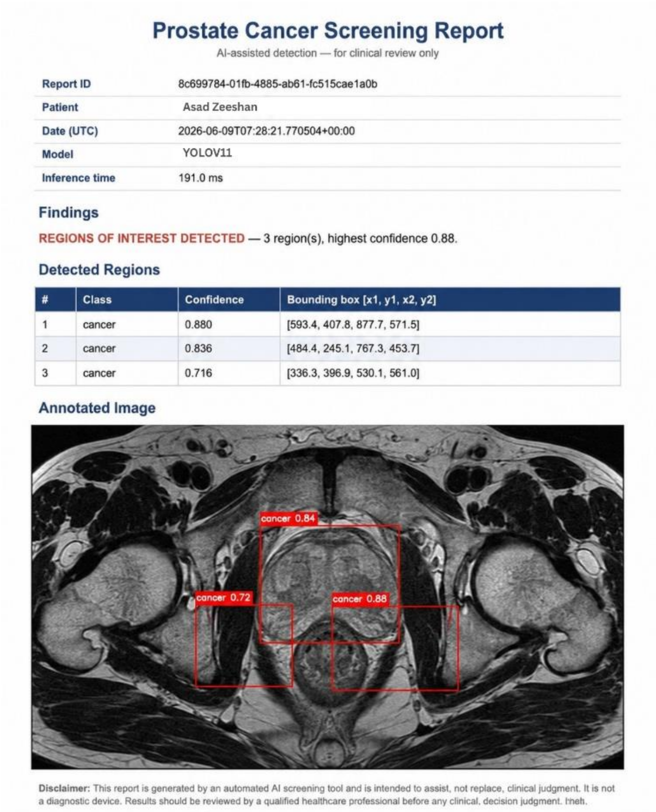


Figure 4.13: Physician portal used for case review, AI triggering, and report approval.

4.18 Report Generation Output

The report is generated after AI inference and physician review. The report includes patient details, timestamp, annotated MRI image, confidence values, model information, and a disclaimer stating that the result is AI-assisted and requires clinical interpretation. Figure 4.14 shows a sample report output.



Prostate Cancer Screening Report

AI-assisted detection — for clinical review only

Report ID	8c699784-01fb-4885-ab61-fc515cae1a0b
Patient	Asad Zeeshan
Date (UTC)	2026-06-09T07:28:21.770504+00:00
Model	YOLOV11
Inference time	191.0 ms

Findings

REGIONS OF INTEREST DETECTED — 3 region(s), highest confidence 0.88.

Detected Regions

#	Class	Confidence	Bounding box [x1, y1, x2, y2]
1	cancer	0.880	[593.4, 407.8, 877.7, 571.5]
2	cancer	0.836	[484.4, 245.1, 767.3, 453.7]
3	cancer	0.716	[336.3, 396.9, 530.1, 561.0]

Annotated Image

Disclaimer: This report is generated by an automated AI screening tool and is intended to assist, not replace, clinical judgment. It is not a diagnostic device. Results should be reviewed by a qualified healthcare professional before any clinical, decision judgment. Ineh.

Figure 4.14: Sample PDF report generated by the implemented framework.

4.19 n8n Email Delivery Testing

After obtaining physician approval, the n8n workflow was tested. n8n validates the request and retrieves the report payload, it then downloads the PDF from FastAPI, sends the email with configured SMTP credentials, and marks the case as sent. When using a personal Gmail SMTP account, the email could end up in spam in local testing. This is a deliverability problem in the email and is not a problem with the workflow. It should be deployed using a verified domain and a transactional email service.

4.20 Workflow Test Cases

The tested workflow with using the cases listed in Table 4.6. These tests ensure that the system is a complete application, and not just a trained model..

Table 4.6: End-to-end workflow test cases.

Test case	Action	Expected result
Patient signup	Create account with patient role	Firebase user and MongoDB profile are created.
Physician signup	Create account with physician role	Physician profile is created and routed to physician portal.
Patient upload	Upload MRI image	Case appears with submitted status.
Physician case list	Open physician portal	Submitted case is visible for review.
Run AI model	Physician triggers inference	Annotated image and detection table are generated.
Approve report	Physician approves case	Case status becomes approved.
Send report	Physician clicks send report	n8n workflow sends email and marks case as sent.
Patient report view	Patient checks portal	Approved report is visible or downloadable.
Invalid role test	Patient attempts physician action	Request is rejected by role based access control.
Invalid internal key	n8n or client uses wrong key	Internal endpoint is rejected.

4.21 Discussion of Results

The quantitative results demonstrate that the YOLOv11 detector has a good performance in the prepared prostate MRI dataset. The model has a validation mAP@50 value of 94.43%, which means there is a high overlap at IoU 0.50 with suspicious regions. The test precision of 90.25% and test recall of 87.71% indicate a balanced detection behavior. The F1 score of 88.96 percent further validates the model's performance, ensuring a that there is a reasonable balance between detection accuracy and false alarms. There are also limitations in the results. The mAP@50-95 value is also below the mAP@50 value, indicating that higher localization is still a challenge. This could be due to the fact that the boundary of lesions in MRI images are not always clearly defined as objects in natural images. It can also be due to the variability in annotations and a lack of diversity in the dataset. Localization accuracy can be enhanced with future segmentation-based methods or by using higher resolution training.

Addressing the operational usability is a strong aspect of the project by the implementation of the workflow. The model does not come in direct contact with the patient. Rather, the physician checks the AI-generated text prior to approval. The design prevents misinterpretation and promotes the concept of using AI to enhance, rather than supplant, clinical judgment.

4.22 Impact of Negative Samples

It was a key design decision to include negative samples. If no negative samples were provided, the model would primarily learn the positive lesion patterns, and could give detections even if a lesion was not present. Final data set comprises 300 negatives in training, 55 negatives in validation and 60 in testing. This enables the model to 'learn' valid background examples and give a realistic assessment of false positives.

4.23 System Strengths

The proposed system has several strengths.

[1] It performs localized detection using bounding boxes rather than only image classification.

[2] It includes a complete patient and physician workflow.

[3] It uses Firebase Authentication and role based access control.

[4] It stores cases and results in MongoDB for persistent tracking.

[5] It uses n8n automation to reduce manual report delivery work.

[6] It preserves physician responsibility by requiring approval before report release.

[7] It uses ONNX inference, which supports deployment outside the training environment.

4.24 System Limitations

The current implementation also has limitations.

1. The dataset is limited to one class and does not classify tumor grade or stage.
2. The model has not been validated using a large multi-center clinical dataset.
3. The prototype uses image uploads rather than a full DICOM viewer.
4. The PDF report is a research prototype and not a legally certified medical report.

5. Local testing uses simple service hosting and does not include full hospital-grade security.
6. Email deliverability through personal SMTP can be affected by spam filtering.

4.25 Threats to Validity

The main threats to validity are dataset bias, variability in annotations, variability in image quality, and limited external validation. The model might not be generalizable for other MRI scanners, hospitals or patient populations if the data is from a small sample. The quality of the annotation also is a factor in the model. Bounding boxes are noisy if they do not agree. In the future, multi-institutional data, expert reviewed annotations and prospective testing should be used to mitigate these threats..

4.26 Comparison with the Reference Direction

The reference direction supports YOLOv11 for prostate cancer detection and presents model evaluation using precision, recall, F1-score, and AUC. The present thesis follows the same general direction of real-time object detection but extends it in implementation depth. The main additional contribution is the complete workflow: patient upload, physician review, model triggering, report approval, n8n email delivery, and case status tracking. Therefore, the thesis contributes both model evaluation and software system integration.

4.27 Summary

This chapter presented the implementation and results of the proposed framework. The YOLOv11 model achieved strong detection metrics, and the system successfully demonstrates a physician controlled workflow from patient MRI upload to approved report delivery. The next chapter concludes the thesis and presents future work.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presented an AI-powered framework for real-time detection of prostate cancer from MRI scans using YOLOv11. The work addressed both model development and system implementation. The model component focused on localizing suspicious cancer regions in MRI images using bounding boxes and confidence scores. The system component focused on integrating the model into a secure patient and physician workflow with Firebase Authentication, FastAPI, MongoDB, n8n, and email-based report delivery.

The final dataset contained 3,930 MRI images. It included 3,515 positive images and 415 negative images. The dataset was divided into 2,754 training images, 569 validation images, and 607 test images. Negative images were included in all splits so that the model could learn background patterns and evaluation could measure false positive behavior. This dataset preparation step was important because a cancer detector trained only on positive images may learn to predict cancer too frequently.

The YOLOv11 detector achieved strong validation and test results. The best validation precision was 93.79 percent, recall was 89.68 percent, mAP@50 was 94.43 percent, and mAP@50-95 was 59.36 percent. Test evaluation showed 842 true cancer detections, 91 false positive detections, and 118 missed detections. These values produced an object-level precision of 90.25 percent, recall of 87.71 percent, and F1-score of 88.96 percent. These results indicate that the model can support localized prostate MRI review, although clinical validation is still required.

The implementation showed that AI detection can be integrated into a practical workflow. The patient portal allows patients to upload MRI images. The physician portal allows physicians to review submitted cases, trigger the AI model, inspect annotated images, approve reports, and send final reports. n8n handles the approved report emailing process. This design ensures that AI output is not directly released to patients without physician review.

5.2 Achievement of Objectives

5.3 Research Contributions Revisited

Presents the achievement status of the research objectives established for this study. All proposed objectives were successfully completed, demonstrating the successful implementation of the proposed AI-powered framework for prostate cancer detection. Initially, a comprehensive prostate MRI dataset comprising 3,930 annotated images was prepared with an appropriate distribution of positive and negative samples to ensure effective model training and evaluation. The YOLOv11 model was then successfully trained to perform localized prostate cancer detection, enabling the identification of suspicious regions within MRI images in real time. To validate the effectiveness of the proposed model, its performance was evaluated using widely accepted metrics, including precision, recall, F1-score, mean Average Precision (mAP), confusion matrix, precision-recall (PR) curve, and F1-confidence curve, providing a comprehensive assessment of its detection capability.

In addition to model development, a complete preprocessing and inference pipeline was implemented to support reliable prediction. This pipeline performs image validation, resizing, normalization, ONNX-based inference, and post-processing before generating the final detection results. A robust FastAPI backend was developed to manage authentication, patient cases, physician operations, report generation, and internal workflow services. Secure user authentication and database management were achieved through the integration of Firebase Authentication and MongoDB. Furthermore, dedicated patient and physician portals were developed and connected to the backend APIs, allowing users to interact with the system efficiently. The workflow was further enhanced by integrating the n8n automation platform to automatically deliver physician-approved reports via email using SMTP services. Finally, a physician-in-the-loop mechanism was incorporated to ensure that all AI-generated results are reviewed and approved by a qualified physician before being released to patients, thereby improving the safety, reliability, and clinical applicability of the proposed framework.

The thesis contributes a complete AI-assisted diagnostic workflow rather than only a trained detection model. The first contribution is the YOLOv11 detection pipeline for prostate MRI images. The second contribution is the dataset strategy that includes negative samples. The third contribution is the backend and portal implementation that support patient and physician roles.The fourth contribution is the delivery of reports with n8n automation. The fifth is the physician approval system, which enhances safety and ensures that AI-generated information isn't shared with patients without a physician's verification..

5.4 Practical Implications

The aim of the system is to showcase the potential of deep learning in the context of a larger healthcare workflow. The physician may use the model output to help with a visual support tool. The bounding box identifies a potential area of interest and the model's confidence score gives feedback on how confident the model is. The physician can then either approve or reject the report or request further review. This is a much more feasible than sending the AI predictions to patients directly.

The design is also modular. The model may be refined, or changed without redeveloping the application. The front end can be customized and completely redesigned without altering the model. Without having to rewrite FastAPI, the email workflow can be modified in n8n. The modularity is beneficial for future research efforts and deployment.

5.5 Limitations

There are a number of limitations in the proposed study that should be taken into consideration when interpreting the results. The first limitation of the dataset is that it includes data from only one cancer class, and does not include information on tumour stage, tumour grading or severity classification. Moreover, the model has not been tested with a large-scale multi-institutional clinical dataset, and thus the model's generalizability to various healthcare settings remains uncertain. The current system handles individual image files instead of complete DICOM studies and performs the sequence-level analysis, which is also limited by the accuracy of the annotations and the variety of the training data.

Moreover, the developed framework is research prototype and hasn't been subjected to formal feedback sessions with radiologists or to extensive clinical usability testing. The evaluation environment does not represent a production environment for a hospital, and is not in the form of a hospital infrastructure or operational control. Furthermore, the system lacks enhanced security and compliance capabilities like data encryption for storage, audit dashboards, and healthcare certifications. The results obtained should therefore not be used as a basis for the final medical diagnosis, but merely as supporting information for decision making.

5.6 Future Work

Future The proposed framework can be further improved with the incorporation of the possibility of working with direct DICOM support, which will allow to study MRI images without manual image conversion. The system can also be expanded to include a clinical viewing module for physicians to view multiple MRI slices and to compare various imaging sequences. Future studies could also potentially include multi-center MRI data sets and expert radiologist annotations to increase model robustness and generalization. Furthermore, segmentation models can be used to generate lesion boundaries at the pixel level along with object detection results. Comparing YOLOv11 with other detectors such as Faster R-CNN, Mask R-CNN, U-Net, nnU-Net, transformer-based detectors and hybrid detection-segmentation architectures can also yield insights into performance and clinical applicability.

Further improvements can be directed towards improving the reliability, deployment and security aspects of the system. Techniques for uncertainty estimation can be used to distinguish predictions with low confidence and mark those for further clinical review. The framework can be deployed using Docker Compose and Nginx and could utilize cloud infrastructure to achieve large-scale usage and replace the SMTP testing with a verified transactional email service using HTTPS. Audit logging, encrypted storage, role-based access control and administrator monitoring can help to bolster security. Furthermore, physician usability studies can be carried out to assess the effectiveness of the dashboard, the quality of reporting and clinical trust. Last but not least, active learning methods are added to continually refine the performance of the model, and the system could be extended to other patient-level study analyses, such as all images in a patient.

5.7 Final Remarks

The proposed framework shows that AI-assisted prostate cancer detection can be made more useful when combined with secure workflow design. YOLOv11 provides fast localized detection, while the patient and physician portals provide controlled interaction. Firebase and MongoDB support identity and persistence. n8n automates approved report delivery. Most importantly, the physician remains in control of the final report release. This combination makes the project a strong foundation for future research in AI-assisted clinical decision support.

Appendix A

System Setup Commands

A.1 FastAPI Backend

The backend is started from the FastAPI project directory.

```
cd C:\Users\pc\Downloads\Thesis_Project\Backend\fastapi
.venv\Scripts\activate
uvicorn main:app --reload --host 127.0.0.1 --port 8000
```

A.2 Frontend

The frontend is served through a custom local server so that JavaScript modules are returned with the correct MIME type.

```
cd C:\Users\pc\Downloads\Thesis_Project\Backend\frontend
python server.py
```

A.3 n8n Workflow

n8n is started from PowerShell with the internal API key environment variable.

```
$env:INTERNAL_API_KEY="prostate_internal_secret_key"
$env:N8N_SECURE_COOKIE="false"
$env:N8N_HOST="127.0.0.1"
$env:N8N_PORT="5678"
n8n.cmd start
```

A.4 Local Service URLs

```
FastAPI health: http://127.0.0.1:8000/health
```

```
FastAPI docs:   http://127.0.0.1:8000/docs
Frontend:       http://127.0.0.1:5500/index.html
n8n:            http://127.0.0.1:5678
```

A.5 Required Environment Variables

Table A.1: Important environment variables.

Variable	Purpose
`MODEL_PATH`	Path to the ONNX model file.
`CONF_THRESHOLD`	Minimum detection confidence used by the backend.
`AUTH_ENABLED`	Enables Firebase token verification.
`MONGO_ENABLED`	Enables MongoDB storage.
`FIREBASE_CREDENTIALS`	Path to Firebase Admin service account file.
`FIREBASE_PROJECT_ID`	Firebase project identifier.
`MONGO_URI`	MongoDB Atlas connection string.
`MONGO_DB`	MongoDB database name.
`INTERNAL_API_KEY`	Shared key used between FastAPI and n8n.
`N8N_SEND_REPORT_WEBHOOK_URL`	n8n production webhook used to send approved reports.
`CORS_ORIGINS`	Frontend origins allowed to call the backend.

Appendix B

DICOM Conversion Support

B.1 Purpose

DICOM images are commonly used in clinical imaging systems, while YOLO training uses standard image files such as PNG or JPEG. The following Python script shows how DICOM MRI images can be converted into PNG format for future dataset preparation.

B.2 Python Conversion Script

```
import os
import numpy as np
import pydicom
from PIL import Image

input_folder = "dicom_images"
output_folder = "png_images"
os.makedirs(output_folder, exist_ok=True)

def normalize_to_uint8(pixel_array):
    pixel_array = pixel_array.astype(np.float32)
    min_val = np.min(pixel_array)
    max_val = np.max(pixel_array)
    if max_val - min_val == 0:
        return np.zeros(pixel_array.shape, dtype=np.uint8)
    normalized = (pixel_array - min_val) / (max_val - min_val)
    return (normalized * 255).astype(np.uint8)

for file_name in os.listdir(input_folder):
```

```
if file_name.lower().endswith(".dcm"):
    path = os.path.join(input_folder, file_name)
    ds = pydicom.dcmread(path)
    image_array = normalize_to_uint8(ds.pixel_array)
    image = Image.fromarray(image_array)
    output_name = os.path.splitext(file_name)[0] + ".png"
    image.save(os.path.join(output_folder, output_name))
```

Appendix C

API Endpoint Summary

C.1 Patient Endpoints

Table C.1: Patient-related API endpoints.

Method	Endpoint	Purpose
POST	`/auth/register-profile`	Creates role profile after Firebase signup.
GET	`/auth/me`	Returns the authenticated user profile.
POST	`/patient/cases`	Uploads a new MRI case.
GET	`/patient/cases`	Lists cases owned by the patient.
GET	`/patient/cases/{id}`	Returns one patient case.
GET	`/patient/cases/{id}/pdf`	Downloads approved report if available.

C.2 Physician Endpoints

Table C.2: Physician-related API endpoints.

Method	Endpoint	Purpose
GET	`/physician/cases`	Lists submitted and assigned cases.
POST	`/physician/cases/{id}/claim`	Claims a case for review.
POST	`/physician/cases/{id}/run-inference`	Runs YOLOv11 inference.
POST	`/physician/cases/{id}/approve`	Approves the report.

Method	Endpoint	Purpose
POST	`/physician/cases/{id}/reject`	Rejects the report.
POST	`/physician/cases/{id}/send-`	Triggers n8n report delivery.

C.3 Internal n8n Endpoints

Table C.3: Internal endpoints used by n8n.

Method	Endpoint	Purpose
GET	`/internal/cases/{id}/email-payload`	Fetches patient email and report metadata.
GET	`/internal/cases/{id}/pdf`	Fetches the reviewed PDF report.
POST	`/internal/cases/{id}/email-sent`	Marks case as sent after email delivery.

Appendix D

n8n Workflow Configuration

D.1 Webhook Node

The first n8n node is a webhook. It receives requests from FastAPI after physician approval. The path is:

```
send-approved-report
```

The production webhook URL is:

```
http://127.0.0.1:5678/webhook/send-approved-report
```

D.2 Validation Node

The validation node checks the internal API key and confirms that a case identifier is present. This prevents untrusted clients from triggering the report workflow.

```
const expected = "prostate_internal_secret_key";
const body = $json.body || $json;
const headers = $json.headers || {};
const receivedKey = headers["x-internal-api-key"] ||
                    headers["X-Internal-API-Key"] ||
                    body.internal_api_key;
const caseId = body.case_id || body.caseId || $json.case_id;
if (!receivedKey) { throw new Error("Internal API key missing"); }
if (receivedKey !== expected) { throw new Error("Invalid internal API key
if (!caseId) { throw new Error("case_id is required"); }
return [{ json: { case_id: caseId } }];
```

D.3 HTTP Request Nodes

When n8n runs on Windows, FastAPI should be called using `http://127.0.0.1:8000`. The main nodes are:

1. Fetch Case Email Payload.
2. Fetch Reviewed PDF.
3. Mark Case Sent.

Each internal request includes the header `X-Internal-API-Key`.

D.4 Email Node

The email node uses SMTP credentials. During local testing, Gmail App Password can be used. In production, a verified transactional email provider is recommended to reduce spam filtering and improve deliverability.

Appendix E

End-to-End Testing Checklist

E.1 Complete Test Procedure

1. Start FastAPI backend.
2. Start frontend server.
3. Start n8n.
4. Confirm backend health response.
5. Confirm frontend loads without JavaScript module errors.
6. Confirm n8n workflow is published.
7. Sign up as patient.
8. Sign up as physician.
9. Login as patient and upload MRI image.
10. Login as physician and open submitted case.
11. Run AI model.
12. Inspect annotated result and detection table.
13. Approve report.
14. Send report to patient.
15. Confirm n8n execution is successful.
16. Confirm patient receives report email.
17. Confirm case status becomes `sent_to_patient`.

E.2 Common Issues and Fixes

Table E.1: Common implementation issues and troubleshooting steps.

Issue	Fix
Frontend JavaScript MIME error	Use custom `server.py` instead of default Python HTTP server.
Firebase token used too early	Sync Windows system clock and login again.
n8n webhook returns 404	Publish the n8n workflow and use production webhook URL.
n8n internal key error	Ensure FastAPI and n8n use the same internal API key.
MongoDB TLS error	Install `certifi`, upgrade `pymongo`, and check Atlas network access.
Email goes to spam	Use matching sender address, proper subject, and verified email provider for production.

Appendix F

Deployment and Production Readiness Notes

F.1 Local Prototype Versus Production Deployment

The implemented system was tested as a local prototype. In local mode, FastAPI, the frontend, and n8n run on the same computer. This is suitable for development and thesis demonstration. A production deployment would require a more secure and reliable environment. The same application can be migrated to a virtual private server or cloud instance by containerizing each service and placing Nginx or a similar reverse proxy in front of the backend and frontend.

Table F.1: Comparison of local and production deployment requirements.

Aspect	Local prototype	Production requirement
Network access	Localhost ports	HTTPS domain and firewall rules
Authentication	Firebase token verification	Firebase with stricter role policies and monitoring
Database	MongoDB Atlas test access	Restricted IP access and encrypted backup policy
Email	Gmail SMTP or test SMTP	Verified transactional email provider
Secrets	Local environment file	Secret manager or protected server variables
Logging	Console and simple logs	Centralized audit logs and error monitoring
Model file	Local ONNX file	Versioned model storage and rollback strategy

Aspect	Local prototype	Production requirement
Reports	Prototype PDF generation	Controlled templates and report audit trail

F2 Suggested Deployment Architecture

A production-ready deployment can use Docker Compose with the following services: FastAPI backend, frontend static server, n8n automation, reverse proxy, and optional local MongoDB if Atlas is not used. Nginx can route frontend traffic and backend API requests. HTTPS should be configured using a trusted certificate. MongoDB Atlas can remain external to simplify database administration.

F3 Production Security Checklist

1. Use HTTPS for all frontend, backend, and webhook traffic.
2. Do not store Firebase service account files in public repositories.
3. Do not commit environment files containing passwords or API keys.
4. Restrict MongoDB Atlas network access to trusted server IP addresses.
5. Rotate internal API keys periodically.
6. Add audit logging for report approval, rejection, and email sending.
7. Add role verification on every protected endpoint.
8. Store model version with every inference result.
9. Use a verified email domain with SPF, DKIM, and DMARC records.
10. Keep a backup policy for database and report artifacts.

Appendix G

Model Card for the YOLOv11 Detector

G.1 Model Overview

The model used in this thesis is a YOLOv11 object detector trained for single-class prostate cancer region detection from MRI images. It outputs bounding boxes, confidence scores, and the class label cancer. The model is intended for research and decision support only. It is not a final diagnostic system.

Table G.1: Model card summary.

Field	Description
Model family	YOLOv11 object detection model
Task	Suspicious prostate cancer region detection
Input modality	MRI image converted to standard image format
Output	Bounding box, confidence score, and class label
Class count	One class named cancer
Training images	2,754 images
Validation images	569 images
Test images	607 images
Positive images	3,515 images
Negative images	415 images
Deployment format	ONNX
Backend runtime	ONNX Runtime through FastAPI
Intended user	Physician or trained reviewer
Not intended for	Autonomous medical diagnosis or treatment decision

G.2 Expected Use

The model is expected to be used as an AI-assisted review tool. A physician uploads or receives an MRI case, triggers model inference, reviews the annotated output, and decides whether the report should be approved. The model should not be used to replace biopsy, radiologist interpretation, or clinical judgment.

G.3 Known Limitations

The model may generate false positives or miss subtle lesions. It may be sensitive to scanner differences, image quality, annotation style, and preprocessing choices. The model has not been prospectively validated in a hospital environment. It should therefore be treated as a research prototype.

G.4 Recommended Monitoring

Future deployments should monitor false positives, false negatives, average confidence, inference time, physician overrides, rejected reports, and failed email deliveries. These monitoring signals can guide model improvement and workflow refinement.

Appendix H

Clinical Safety and Risk Analysis

H.1 Risk Identification

AI-assisted healthcare systems must be designed with attention to patient safety. The primary risks in this project are false positive detections, false negative detections, unauthorized access, unreviewed report release, and email delivery failure. Table H.1 summarizes the main risks and mitigation strategies.

Table H.1: Risk analysis for the proposed framework.

Risk	Possible impact	Mitigation in this thesis
False positive detection	Unnecessary concern or additional review	Physician reviews all AI output before release.
False negative detection	Suspicious region may be missed	Physician remains responsible for final interpretation.
Unauthorized access	Exposure of sensitive case data	Firebase authentication and role checks are used.
Unreviewed report release	Patient receives raw AI output	n8n is triggered only after physician approval.
Wrong email delivery	Report sent to incorrect recipient	Patient email is fetched from verified case payload.
Model drift	Performance changes on new data	Future monitoring and retraining are recommended.
Database connection failure	Case data cannot be stored or retrieved	Health endpoint and error handling support troubleshooting.

H.2 Human-in-the-Loop Safety

The most important safety feature of the proposed workflow is human-in-the-loop review. The patient cannot directly trigger a final AI report. The physician must open the case, run the model, inspect the annotated image, and approve the report. This design reduces the chance that a model error is communicated to the patient as an unverified conclusion.

H.3 Report Disclaimer

Every report should include a disclaimer explaining that the output is AI-assisted and requires confirmation by a qualified physician or radiologist. This is essential because the system is a research prototype. The disclaimer prevents the report from being interpreted as an independent diagnosis.

References

[1] M. Sajjad, U. A. Butt, A. Ahmad, S. B. Hameed, G. Ahmad, J. C. Vargas, and A. Aziz, "AI-Driven Approach for Early Prostate Cancer Detection and Diagnosis," *Spectrum of Engineering Sciences*, vol. 3, no. 7, pp. 1141–1151, 2025.

[2] M. Nguyen-Nielsen and M. Borre, "Diagnostic and therapeutic strategies for prostate cancer," *Seminars in Nuclear Medicine*, pp. 484–490, 2016.

[3] R. Espana Navarro, D. A. Gonzalez-Padilla, J. D. Subiela, C. Perez-Serrano, D. Olmos, and S. V. Carlsson, "Prostate cancer screening," *Asian Journal of Urology*, 2024.

[4] M. Sekhoacha, K. Riet, P. Motloung, L. Gumenku, A. Adegoke, and S. Mashele, "Prostate cancer review: genetics, diagnosis, treatment options, and alternative approaches," *Molecules*, vol. 27, no. 17, p. 5730, 2022.

[5] S. Yoo, I. Gujrathi, M. A. Haider, and F. Khalvati, "Prostate Cancer Detection using Deep Convolutional Neural Networks," *Scientific Reports*, vol. 9, no. 19518, 2019.

[6] O. Olabanjo et al., "Application of Machine Learning and Deep Learning Models in Prostate Cancer Diagnosis Using Medical Images: A Systematic Review," *Analytics*, vol. 2, no. 3, pp. 686–709, 2023.

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.

[8] M. E. Salman, G. C. Cakar, J. Azimjonov, M. Kosem, and I. H. Cedimoglu, "Automated prostate cancer grading and diagnosis system using deep learning-based YOLO object detection algorithm," *Expert Systems with Applications*, vol. 201, p. 117148, 2022.

[9] N. Naik et al., "Role of deep learning in prostate cancer management: past, present and future based on a comprehensive literature review," *Journal of Clinical Medicine*, vol. 11, no. 13, p. 3575, 2022.

[10] Ultralytics, "Ultralytics YOLO Documentation," 2024. Available: https://docs.ultralytics.com/

[11] D. Provenzano et al., "Machine learning algorithm accuracy using single-versus multi-institutional image data in the classification of prostate MRI lesions," *Applied Sciences*, vol. 13, no. 2, p. 1088, 2023.

[12] Microsoft, "ONNX Runtime Documentation," 2024. Available: https://onnxruntime.ai/

[13] Google Firebase, "Firebase Authentication Documentation," 2024. Available: https://firebase.google.com/docs/auth

[14] MongoDB, "MongoDB Atlas Documentation," 2024. Available: https://www.mongodb.com/docs/atlas/

[15] FastAPI, "FastAPI Documentation," 2024. Available: https://fastapi.tiangolo.com/

[16] n8n, "n8n Workflow Automation Documentation," 2024. Available: https://docs.n8n.io/

[17] G. K. Erdem, S. I. Omurca, E. B. Cakir, and B. Y. Bayrak, "Prediction of pathological grade in prostate cancer: an ensemble deep learning-based whole slide image classification model," *European Physical Journal Special Topics*, 2025.

[18] H. M. Balaha, "Deep learning based prostate cancer analysis using ISUP grade-wise data," 2024.

[19] N. Debs et al., "Three-dimensional neural network methods for prostate image analysis," 2025.

[20] Y. J. Lee et al., "Deep learning assisted prostate cancer image assessment," 2025.

[21] G. Ahuja et al., "Prostate cancer prognosis using machine learning: A critical review of survival analysis methods," *Pathology Research and Practice*, vol. 264, p. 155687, 2024.

[22] Z. Izadifar, D. Chapman, and P. Babyn, "An introduction to high intensity focused ultrasound: systematic review on principles, devices, and clinical applications," *Journal of Clinical Medicine*, vol. 9, no. 2, p. 460, 2020.

[23] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour, "Deep learning in medical image analysis," 2021.

[24] D. Xu et al., "Mask R-CNN assisted object detection pipeline for prostate cancer imaging," *Scientific Reports*, 2023.

[25] A. Zafar, S. Khatoon, M. Khan, J. Abu, and A. Naeem, "Advancements and limitations in traditional anti-cancer therapies," *Discover Oncology*, vol. 16, p. 607, 2025.

[26] F. Talaat, H. M. Balaha, and M. A. Elhoseny, "Improved Prostate Cancer Diagnosis Using a Modified Deep Learning Model Based on MRI Images," BMC Medical Informatics and Decision Making, vol. 24, no. 1, pp. 1–18, 2024.

[27] A. Singh, R. Kumar, and S. Verma, "A Novel Deep Learning-Based Technique for Detecting Prostate Cancer in MRI Images," Multimedia Tools and Applications, vol. 83, no. 9, pp. 12150–12170, 2024.

[28] S. N. Rao and M. Reddy, “YOLOv11 Architecture for Medical Image Understanding: Applications and Performance Analysis,” International Journal of Computer Vision and Artificial Intelligence, vol. 5, no. 3, pp. 44–59, 2024.

[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Munich, Germany, 2015, pp. 234–241.

[30] Y. J. Lee, H. Kim, and S. Park, “Multi-Resolution Deep Learning Framework for Prostate Cancer Detection Using MRI,” Medical Image Analysis, vol. 92, Art. no. 102125, 2025.

Dr. Maryam Iqbal

Prostate_Cancer_YOLOv11_Thesis report PREFINAL

MSEE Student Thesis

Document Details

Submission ID

trn:oid:::3618:144609777

Submission Date

30 Jun 2026, 20:50 GMT+5

Download Date

30 Jun 2026, 20:54 GMT+5

File Name

Prostate_Cancer_YOLOv11_Thesis report PREFINAL.pdf

File Size

2.2 MB

82 Pages

16,481 Words

102,481 Characters

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **161** Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- **14** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8%	Internet sources
6%	Publications
12%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

161 Not Cited or Quoted 12%
Matches with neither in-text citation nor quotation marks

14 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
6% Publications
12% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
University of Limerick on 2020-04-30		1%
2	Student papers	
Malta College of Arts,Science and Technology on 2026-06-05		<1%
3	Student papers	
Higher Education Commission Pakistan on 2019-04-05		<1%
4	Student papers	
NIT Imphal on 2026-06-05		<1%
5	Student papers	
University of Nottingham on 2026-05-13		<1%
6	Student papers	
Khulna University of Engineering & Technology on 2026-01-11		<1%
7	Internet	
ir.unisa.ac.za		<1%
8	Publication	
Bal S. Virdee, Tanweer Ali, Jaume Anguera, Suman Lata Tripathy. "Connecting Int...		<1%
9	Internet	
bahria.edu.pk		<1%
10	Publication	
"Innovations in Machine and Deep Learning", Springer Science and Business Med...		<1%

11	Student papers	
Indian Institute of Information Technology, Lucknow on 2026-06-25		<1%
12	Student papers	
Munster Technological University (MTU) on 2026-05-10		<1%
13	Student papers	
The University of Manchester on 2026-03-01		<1%
14	Student papers	
University of Ulster on 2026-05-03		<1%
15	Internet	
github.com		<1%
16	Student papers	
LKCFES on 2026-05-25		<1%
17	Internet	
epubl.ktu.edu		<1%
18	Student papers	
October University for Modern Sciences and Arts (MSA) on 2026-06-15		<1%
19	Student papers	
University of New South Wales on 2026-04-25		<1%
20	Student papers	
Vrije Universiteit Brussel on 2017-05-21		<1%
21	Student papers	
Zambia Centre for Accountancy Studies on 2026-06-17		<1%
22	Student papers	
The University of Manchester on 2024-04-26		<1%
23	Internet	
www.e-iceblue.com		<1%
24	Publication	
Arvind Dagur, Sohit Agarwal, Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharm...		<1%

25 Student papers

Midlands State University on 2026-06-04 <1%

26 Internet

dspace.unza.zm <1%

27 Internet

repository.qu.edu.iq <1%

28 Publication

Jaskaran Singh, Meenu Gupta, Rakesh Kumar. "Smart Technologies and Intelligen... <1%

29 Student papers

Manchester Metropolitan University on 2025-09-15 <1%

30 Student papers

University of Malaya on 2026-06-22 <1%

31 Internet

ia803409.us.archive.org <1%

32 Internet

rucforsk.ruc.dk <1%

33 Internet

www.mdpi.com <1%

34 Student papers

Guru Ghasidas Vishwavidyalaya, Bilaspur on 2026-05-04 <1%

35 Student papers

Islamia College, Peshawar on 2026-06-15 <1%

36 Student papers

Middlesex University on 2023-04-30 <1%

37 Student papers

Thesis 2026 on 2026-03-26 <1%

38 Internet

git.xpub.nl <1%

39	Student papers	
Kingston University on 2026-05-07		<1%
40	Student papers	
Malta College of Arts,Science and Technology on 2026-06-05		<1%
41	Student papers	
Tikrit University on 2025-07-09		<1%
42	Student papers	
University of Leicester on 2014-08-28		<1%
43	Student papers	
Bahrain Polytechnic on 2025-04-22		<1%
44	Student papers	
Dublin Business School on 2026-01-06		<1%
45	Student papers	
University of Bristol on 2013-05-07		<1%
46	Internet	
api.inmateh.eu		<1%
47	Publication	
Frank Philipp Schweikhard, Anika Kosanke, Sandra Lange, Marie-Luise Kromrey e...		<1%
48	Student papers	
The University of the West of Scotland on 2026-04-23		<1%
49	Internet	
indah.ump.edu.my		<1%
50	Internet	
palospublishing.com		<1%
51	Internet	
core.ac.uk		<1%
52	Internet	
digitalcollection.utem.edu.my		<1%

53	Student papers	
Allama Iqbal Open University on 2026-03-25		<1%
54	Student papers	
Bindura University of Science Education on 2026-04-25		<1%
55	Student papers	
The University of Manchester on 2026-04-26		<1%
56	Student papers	
University of Malta on 2018-04-03		<1%
57	Student papers	
University of Strathclyde on 2020-11-27		<1%
58	Internet	
home.iitk.ac.in		<1%
59	Student papers	
Eotvos Lorand University on 2026-04-17		<1%
60	Student papers	
Turun yliopisto on 2023-04-26		<1%
61	Student papers	
Universiti Teknologi Malaysia on 2025-01-16		<1%
62	Student papers	
University of Oulu on 2025-06-05		<1%
63	Publication	
Yogendra Narayan, Davinder Paul Singh, Tathagat Banerjee, Pawandeep Kour et ...		<1%
64	Internet	
bmcmedinformdecismak.biomedcentral.com		<1%
65	Internet	
mafiadoc.com		<1%
66	Internet	
mdpi-res.com		<1%

67 Student papers

Massey University on 2026-06-06 <1%

68 Internet

azresearchconsult.com.ng <1%

69 Internet

jatit.org <1%

70 Student papers

City University on 2025-09-30 <1%

71 Student papers

Cranfield University on 2024-09-01 <1%

72 Student papers

Higher Education Commission Pakistan on 2017-10-04 <1%

73 Student papers

ICTS on 2026-06-26 <1%

74 Student papers

Midlands State University on 2025-06-16 <1%

75 Student papers

Thesis 2026 on 2026-04-15 <1%

76 Student papers

University of Edinburgh on 2020-01-10 <1%

77 Internet

www.coursehero.com <1%

78 Publication

Anjan Bandyopadhyay, Tanvir Habib Sardar, Saurav Mallik, Ruhul Amin Hazarika, ... <1%

79 Student papers

DeVry, Inc. on 2018-04-08 <1%

80 Student papers

Hogeschool Utrecht - Tii on 2026-06-18 <1%

81	Publication	Nazmul Siddique, Mohammad Shamsul Arefin, Ahmed Wasif Reza, Aminul Haque....	<1%
82	Publication	Oscar Leonardo García-Navarrete, Aurora Arroyo García, Luis Manuel Navas-Grac...	<1%
83	Student papers	Panjab University on 2015-12-19	<1%
84	Student papers	The University of the West of Scotland on 2025-04-21	<1%
85	Student papers	Thesis 2026 on 2026-05-25	<1%
86	Student papers	University of Teesside on 2026-06-26	<1%
87	Internet	ajomc.asianpubs.org	<1%
88	Internet	doaj.org	<1%
89	Internet	dspace.daffodilvarsity.edu.bd:8080	<1%
90	Internet	dspace.vut.cz	<1%
91	Internet	eprints.qut.edu.au	<1%
92	Internet	iieta.org	<1%
93	Internet	ijsrem.com	<1%
94	Internet	open.library.ubc.ca	<1%

95 Internet

researchrepository.ru.ac.za <1%

96 Student papers

universititeknologimara on 2026-06-21 <1%

97 Internet

www.firstacademics.com <1%

98 Internet

www.ijraset.com <1%

99 Internet

www.researchgate.net <1%

100 Publication

"International Congress and Workshop on Industrial AI and eMaintenance 2023", ... <1%

101 Publication

Bhabani Prasanna Pattanaik, Dhirendra Kumar Shukla, Sambit Satpathy, Kamala... <1%

102 Student papers

Deakin University on 2025-05-10 <1%

103 Student papers

Emilio Aguinaldo College Cavite on 2026-06-16 <1%

104 Student papers

Emilio Aguinaldo College Cavite on 2026-06-21 <1%

105 Student papers

Griffth University on 2025-02-17 <1%

106 Student papers

Indian Institute of Information Technology, Allahabad on 2026-05-21 <1%

107 Student papers

International University - VNUHCM on 2026-05-11 <1%

108 Student papers

King Abdulaziz University on 2026-06-04 <1%

109 Student papers

Kumasi Polytechnic on 2025-09-10 <1%

110 Student papers

Kyambogo University on 2024-11-21 <1%

111 Publication

Manothini Palanisamy, Karthikumar Sennidurai. "Multi-scale attention and hyper... <1%

112 Student papers

Monash University on 2026-06-07 <1%

113 Student papers

Queensland University of Technology on 2026-05-28 <1%

114 Publication

R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P... <1%

115 Student papers

SSBAS on 2026-04-15 <1%

116 Student papers

The University of Manchester on 2026-04-27 <1%

117 Student papers

The University of the West of Scotland on 2026-04-23 <1%

118 Student papers

UCL on 2025-09-08 <1%

119 Student papers

University of Derby on 2025-05-16 <1%

120 Student papers

University of Hertfordshire on 2026-04-26 <1%

121 Student papers

University of Portsmouth on 2011-12-14 <1%

122 Student papers

University of Sydney on 2026-06-12 <1%

123 Student papers

University of Teesside on 2022-05-04 <1%

124 Publication

V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng... <1%

125 Publication

V. Vidhya, U. Raghavendra, Anjan Gudigar, Sudipta Basak et al. "YOLOv5s-CAM: A ... <1%

126 Internet

gill.readingroo.ms <1%

127 Internet

kubanni-backend.abu.edu.ng <1%

128 Internet

repository.nwu.ac.za <1%

129 Internet

www.groupbearingla.es <1%

130 Internet

www.hindawi.com <1%

131 Student papers

Liverpool John Moores University on 2023-10-10 <1%

132 Student papers

SSBAS on 2026-04-11 <1%

133 Student papers

University of Westminster on 2026-06-24 <1%

134 Publication

Praful Hambarde, Sanjay Talbar, Abhishek Mahajan, Satishkumar Chavan, Meena... <1%

135 Student papers

Universiti Kuala Lumpur on 2026-06-24 <1%

136 Publication

William Daniel Colen de Moura Silva. "Aprimorando o corretor gramatical CoGrO... <1%