University of Macedonia

Department of Applied Informatics

Diabetic Foot Ulcer Detection using Deep Learning and Object Detection Models

BSc's Thesis of

Dimitrios Sparagis

Supervisor

Eftichios Protopapadakis

THESSALONIKI, JUNE 2025

Abstract

Diabetic Foot Ulcers (DFUs) are a serious complication affecting millions of patients globally. Early and accurate detection of ulcers can significantly reduce the risk of infection, amputation, and death. With the advent of artificial intelligence and deep learning, the development of automated ulcer detection systems has gained momentum, especially with object detection models.

This thesis explores the application of deep learning techniques, specifically object detection models such as YOLO, Faster R-CNN, and SSD, for the automated detection of foot ulcers. The models are trained and evaluated on a dataset of annotated ulcer images in COCO format and YOLO format (data.yaml). The pipeline includes preprocessing, data augmentation, model training, hyperparameter optimization, and evaluation. Quantitative evaluation using metrics such as mAP (mean Average Precision), precision, recall and statistical tests is used to compare the models.

The results demonstrate that object detection models, particularly Faster R-CNN, can be effectively adapted to the medical imaging domain to support clinical diagnosis. The thesis concludes with a discussion on practical deployment considerations and possible future extensions.

Dedication

I would like to formally dedicate this thesis to my family and friends, whose unwavering support, patience, and encouragement have been instrumental throughout the duration of this academic journey. I also extend my sincere gratitude to Professor Eftichios Protopapadakis for his guidance, constructive feedback, and valuable insights, all of which have been essential to the successful completion of this work.

Table of Contents

Abstract 2
Dedication 3
1. Introduction 7
2. Related Work on Diabetic Foot Ulcer Detection 9
3. Proposed Methodology 12
4. Implementation and results 19
4.1 Implementation 19
4.2 Evaluation 24
5. Conclusion and Future Work 31
6. References 33

List of Figures

Figure 1 Image of a Diabetic Foot Ulcer 7
Figure 2 Object Detection Example 12
Figure 3 YOLOv11 Architecture 13
Figure 4 Faster-RCNN Architecture 14
Figure 5 SSD Architecture 15
Figure 6 Model Evaluation Pipeline Flowchart 17
Figure 7 YOLO Training Loss Curve 22
Figure 8 Faster-RCNN Training Loss Curve 23
Figure 9 SSD Training Loss Curve 23
Figure 10 Per-Image mAP@0.5 Comparison Across Models 25
Figure 11 Distribution of mAP@0.5 per Image (Violin Plot) 26
Figure 12 YOLO Ulcer Detection Sample 1 27
Figure 13 YOLO Ulcer Detection Sample 2 27
Figure 14 SSD Ulcer Detection Sample 1 28
Figure 15 SSD Ulcer Detection Sample 2 28
Figure 16 Faster-RCNN Ulcer Detection Sample 1 29
Figure 17 Faster-RCNN Ulcer Detection Sample 2 29

List of Tables

Table 1 Model Comparison 10
Table 2 Model Input Size 19
Table 3 Applied Augmentations 20
Table 4 Model Training Parameters 21
Table 5 Model Results 24
Table 6 Statistical Testing Results 25

1. Introduction

Diabetic Foot Ulcers (DFUs) are a serious and growing public health concern, particularly among individuals living with diabetes mellitus. In Figure 1, there is a medical illustration showing the bottom (plantar surface) of a human foot with a diabetic foot ulcer. These ulcers significantly increase the risk of infection, hospitalization, and lower-limb amputation. Early and accurate detection is critical for improving patient outcomes and reducing healthcare costs. However, traditional diagnostic methods are time-intensive, rely heavily on clinical expertise, and are subject to human error. Recent advancements in artificial intelligence and deep learning have shown great promise in medical image analysis. In particular, object detection models are capable of identifying and localizing specific regions of interest within images, offering a powerful tool for automated ulcer detection. By leveraging these models, it is possible to create systems that assist clinicians in diagnosing foot ulcers more efficiently and consistently.

This thesis is motivated by the potential to bridge the gap between clinical need and technological capability, by applying state-of-the-art object detection models—such as YOLO, Faster R-CNN, and SSD—to the problem of diabetic ulcer detection.

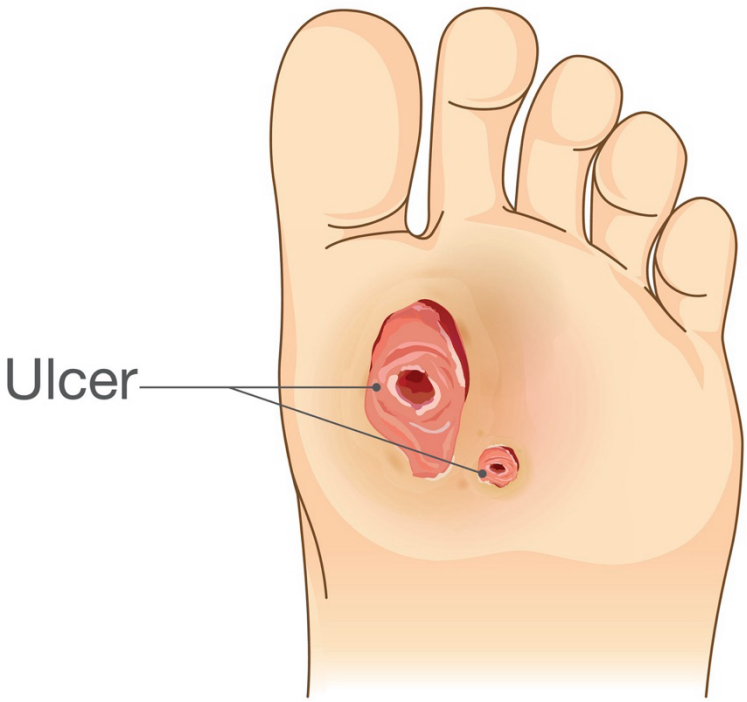

Figure 1 Image of a Diabetic Foot Ulcer

Despite the availability of deep learning technologies, their application in the medical field—especially for DFU detection—remains limited due to challenges such as small datasets, variability in image quality, and the complexity of ulcer characteristics. There is a pressing need for a robust, automated system that can detect and localize foot ulcers with high accuracy, minimal latency, and generalizability across different patient populations and clinical conditions. This thesis addresses the problem of developing and evaluating object detection models that can effectively identify diabetic foot ulcers in medical images. The goal is to assess their performance in a controlled experimental setting and to determine the most effective approach for potential clinical deployment.

The main objectives of this thesis are as follows:

- To implement and train three popular object detection models—YOLO, Faster R-CNN, and SSD—on a dataset of annotated diabetic foot ulcer images.
- To evaluate the models using quantitative metrics such as precision, recall, and mean Average Precision (mAP).
- To compare the models in terms of both detection accuracy and computational performance.
- To perform statistical testing to determine whether observed differences in model performance are significant.
- To explore the practical applicability of these models in a real-world clinical setting.

This thesis is organized into six chapters, each addressing a distinct aspect of the research conducted on the use of deep learning and object detection models for diabetic foot ulcer detection:

- **Chapter 1 – Introduction:** Introduces the research topic, outlines the motivation and problem statement, and defines the objectives and scope of the thesis.
- **Chapter 2 – Related Work on Diabetic Foot Ulcer Detection:** Provides an overview of existing research efforts in DFU detection, focusing on both classical image processing methods and recent advances in deep learning and object detection.
- **Chapter 3 – Proposed Methodology:** Describes the dataset formats, selected object detection models (YOLO, SSD, Faster R-CNN), and evaluation metrics used in this study.
- **Chapter 4 – Implementation and Results:** Presents the implementation details of each model, including training setup and software tools. This chapter also reports quantitative and qualitative results, along with a statistical comparison of model performance.
- **Chapter 5 – Conclusion and Future Work:** Summarizes the key findings, highlights the limitations of the current study, and suggests directions for future research.

2. Related Work on Diabetic Foot Ulcer Detection

This chapter presents a review of existing literature focused on the application of machine learning and computer vision techniques for the detection and classification of diabetic foot ulcers (DFUs). The review is organized into three main categories: traditional image processing techniques, classical machine learning approaches, and deep learning-based object detection methods. Initial attempts at automated DFU detection used classical image processing methods such as thresholding, edge detection, and morphological operations. Goyal et al. [35] applied simple color and texture analysis to isolate ulcer regions in RGB images. While computationally efficient, these methods were highly sensitive to lighting variations, background textures, and skin tone diversity, limiting their robustness.

As datasets grew in size, researchers turned to classical ML techniques like Support Vector Machines (SVM), Random Forests, and k-NN classifiers. These methods relied on handcrafted features such as:

- Local Binary Patterns (LBP)
- Histogram of Oriented Gradients (HOG)
- Color histograms and shape descriptors

For example, Wang et al. [8] trained an SVM using LBP features and achieved higher classification accuracy than thresholding methods. However, the dependency on feature engineering and lack of spatial context made these models less scalable.

With the advent of Convolutional Neural Networks (CNNs), DFU detection underwent a paradigm shift. Alzubaidi et al. [9] proposed a CNN-based classifier trained on a relatively small dataset of DFU images. Their results demonstrated superior performance over traditional methods, but lacked localization capabilities.

Goyal et al. [10] introduced an approach for real-time diabetic foot ulcer detection using the YOLOv2 object detection architecture. Their method emphasized low-latency inference while maintaining acceptable levels of accuracy, making it a suitable candidate for real-time clinical applications where speed is essential. Despite its performance advantages, the study highlighted the trade-off between model simplicity and detection precision, especially in complex ulcer presentations.

Zhang et al. [11] conducted a comparative analysis of two widely used object detection models, Faster R-CNN and SSD, applied to diabetic foot ulcer datasets. Their findings indicated that Faster R-CNN generally achieved higher precision in ulcer localization tasks but required significantly more computational resources and inference time compared to SSD. This contrast underscored the

performance-versus-efficiency dilemma that is often encountered when choosing a model architecture for medical image analysis.

Building upon these efforts, Kumar et al. [37] proposed a multi-class convolutional neural network designed not only for ulcer detection but also for severity grading. Their system was trained to classify ulcers into different stages based on visual features, demonstrating the potential to support triage-level decision-making in clinical workflows. This added granularity in classification aligns more closely with real-world clinical needs, where the severity of a lesion influences the urgency and type of treatment.

Study	Method	Strengths	Limitations
Alzubaidi et al. [9]	CNN classifier	Strong performance	No localization
Goyal et al. [10]	YOLOv2	Real-time performance	Lower small-object accuracy
Zhang et al. [11]	SSD, Faster R-CNN	High precision (Faster R-CNN)	Inference time
Kumar et al. [37]	Severity classifier	Multi-class output	Not focused on detection

Table 1 Comparison Across Studies

These insights directly informed the approach taken in this thesis: to implement and compare three leading object detection models using a unified dataset, with statistical and qualitative evaluation.

The automatic detection of diabetic foot ulcers (DFUs) from medical images presents several persistent challenges, both technical and clinical, that hinder the generalization and practical deployment of machine learning models in real-world healthcare settings.

One of the primary limitations in DFU detection research is the restricted availability of annotated datasets. Publicly accessible datasets are often small in scale, lack diversity in skin tones and lighting conditions, and display high intra-class variability. Furthermore, class imbalance is a

recurring issue, as ulcer instances are typically sparse compared to healthy skin regions. This lack of rich and representative data can lead to overfitting and poor generalization to unseen cases.

Compounding this issue is the inherent visual complexity and variation of DFUs. These ulcers vary significantly in size, shape, color, and stage of progression. They may also be occluded or surrounded by tissue that exhibits similar visual characteristics, such as necrosis or inflammation. These factors complicate both the annotation process and the model's ability to reliably differentiate ulcerated areas from the surrounding tissue.

Another important barrier to adoption in clinical practice is the lack of interpretability in deep learning models. Convolutional neural networks (CNNs), which form the backbone of most modern object detection systems, are often perceived as "black boxes." Their internal decision-making processes are not easily explainable, making it difficult for clinicians to trust or validate their outputs. This limitation is particularly problematic in medical applications, where explainability and accountability are critical.

In addition, many state-of-the-art object detection models are computationally intensive. They require substantial resources for both training and inference, which poses challenges for integration into real-time diagnostic tools or deployment on low-resource devices in clinical environments. Lightweight and efficient architectures are needed to bridge this gap without significantly compromising performance.

To address these challenges, this thesis offers several key contributions. First, it presents the implementation and comparative evaluation of three modern object detection models—YOLOv11, Faster R-CNN, and SSD—on a diabetic foot ulcer dataset, using unified data splits and consistent evaluation metrics. Second, the dataset was curated in both COCO and YOLO formats, with careful alignment of annotations across formats to ensure fair comparisons across model architectures. Third, in addition to standard metric-based evaluation, the thesis includes a rigorous statistical analysis of model performance using the Friedman test and the Wilcoxon signed-rank test, thereby enhancing the empirical validity of the findings. Lastly, a qualitative analysis is provided through visualization of model predictions, including examples of true positives, false positives, and missed detections, to offer interpretive context for the quantitative results.

3. Proposed Methodology

This chapter provides a detailed analysis of the deep learning models used in this thesis: YOLO, Faster R-CNN, SSD, and VGG16. It begins with an overview of Convolutional Neural Network architecture principles and then dives into the unique characteristics and mechanisms of each selected object detection model. Additionally, the chapter outlines key training techniques such as transfer learning and the use of pre-trained models. Finally, it presents the datasets and evaluation metrics that will be used in subsequent chapters.

In recent years, deep learning techniques have emerged as the dominant approach for visual recognition tasks, particularly in medical imaging. While traditional artificial intelligence (AI) and machine learning (ML) [1] methods provided early tools for pattern recognition, modern computer vision is largely driven by deep neural networks, especially Convolutional Neural Networks (CNNs) [3]. CNNs are specialized neural architectures that extract and process spatial hierarchies from image data. Their ability to learn meaningful visual features from raw pixels has made them foundational to tasks such as classification, segmentation, and—most importantly for this work—object detection. For this reason, this thesis focuses primarily on CNN-based object detection techniques rather than classical AI or ML paradigms.

As shown in Figure 2, Object detection involves identifying and localizing objects of interest within an image. Unlike simple classification, object detection models produce both:

- Bounding boxes (location),
- Class labels (what the object is).

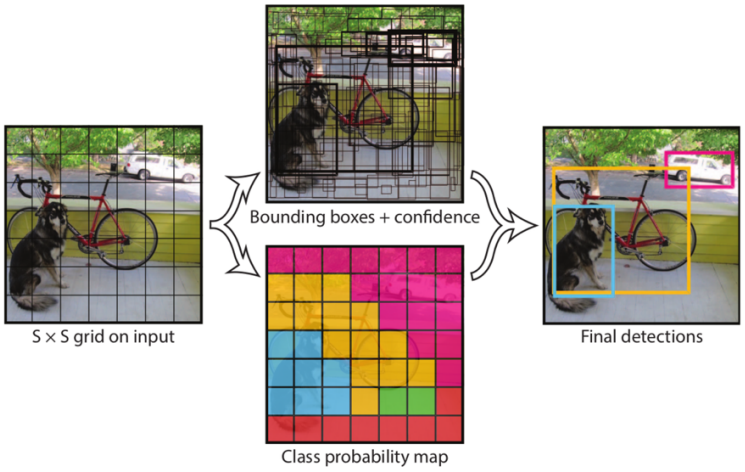

Figure 2 Object Detection Example

Two main categories of object detection models are:

- **Two-stage detectors** (e.g., Faster R-CNN): Propose regions and then classify them [4].
- **One-stage detectors** (e.g., YOLO, SSD): Perform detection and classification simultaneously, enabling faster inference [5][6].4.2 YOLO (You Only Look Once)

YOLO is a one-stage object detection architecture designed for real-time performance. Introduced by Redmon et al. [13], YOLO treats object detection as a single regression problem — predicting bounding boxes and class probabilities directly from full images in one evaluation. Unlike two-stage detectors that rely on region proposal steps, YOLO performs detection and classification simultaneously, making it significantly faster.

The algorithm divides the input image into an S ×S grid. Each grid cell is responsible for predicting a fixed number of bounding boxes, as well as associated confidence scores and class probabilities. After prediction, a non-maximum suppression (NMS) step is applied to eliminate redundant detections and retain the most confident bounding boxes.

YOLO offers several advantages. It is extremely fast, making it ideal for real-time detection tasks, and its architecture enables end-to-end training. However, the model also has limitations. It tends to struggle with detecting small objects and objects that are close together in the image, as the grid-based approach can limit the spatial resolution of detections in densely populated scenes.

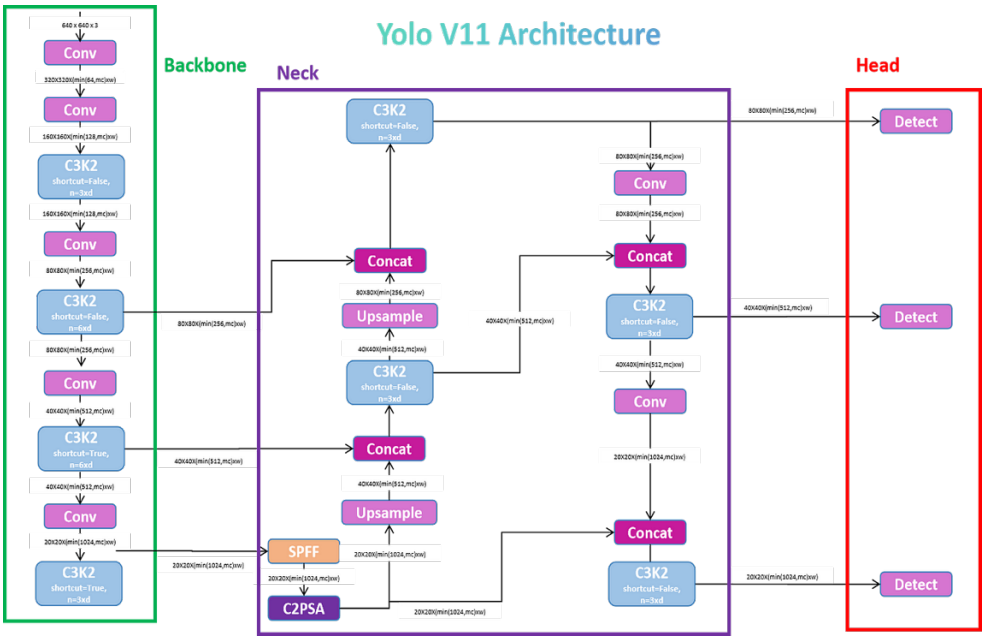

Figure 3 YOLOv11 Architecture

As shown in Figure 3, the YOLOv11 architecture is structured into three main components: Backbone, Neck, and Head. The Backbone extracts features from the input using convolutional layers and CSP blocks. The Neck fuses multi-scale features via upsampling, concatenation, and specialized modules like SPPF, enhancing spatial information and context. Finally, the Head applies detection layers at different scales to output bounding boxes, objectness scores, and class predictions. This architecture is optimized for both speed and accuracy in object detection tasks.

Faster R-CNN is a two-stage object detection architecture introduced by Ren et al. [14] and is widely recognized for its accuracy and robustness in complex visual tasks. The model operates by first generating a set of candidate object regions using a dedicated Region Proposal Network (RPN). These proposals are then refined and classified by a second-stage convolutional neural network, which outputs the final object detections.

The architecture consists of several key components. A backbone network—typically a pretrained CNN such as ResNet or VGG—is used to extract feature maps from the input image. The RPN then slides over these feature maps to generate anchor boxes and assign scores indicating the likelihood of object presence. Regions of interest (RoIs) are subsequently processed through an RoI pooling layer, which crops and resizes the regions into a fixed size suitable for classification. Finally, the refined regions are passed to a fully connected layer that predicts both the object class and the precise bounding box coordinates.

Faster R-CNN is known for its high detection accuracy and its effectiveness in handling small and overlapping objects. However, these benefits come at the cost of computational efficiency. Compared to one-stage models like YOLO and SSD, Faster R-CNN is slower and more complex to train and deploy, making it less suitable for real-time applications.

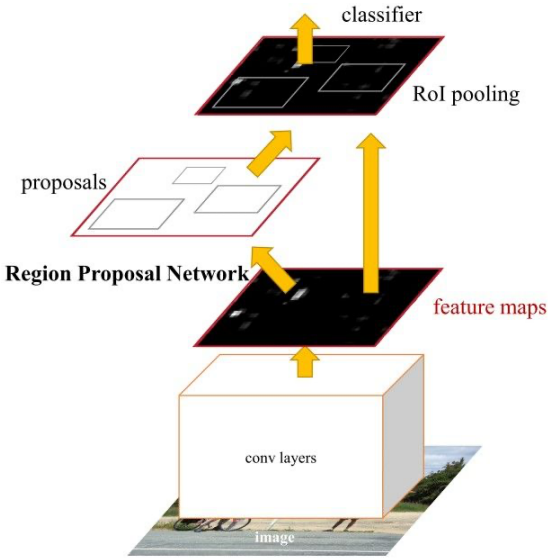

Figure 4 Faster-RCNN Architecture

Figure 4 illustrates the architecture of Faster R-CNN**,** a two-stage object detection framework. First, an input image is passed through convolutional layers to generate feature maps. These maps are sent to the Region Proposal Network (RPN), which suggests candidate object regions (proposals). These proposals are then processed by RoI (Region of Interest) Pooling, which extracts fixed-size feature representations for each region. Finally, these features are fed into a classifier to predict object classes and refine bounding box coordinates. This architecture efficiently combines region proposal and classification into a unified pipeline.

Single Shot MultiBox Detector (SSD) is a one-stage object detection architecture introduced by Liu et al. [15]. Unlike two-stage detectors, SSD performs object localization and classification in a single forward pass, making it significantly faster and well-suited for real-time applications. The model extends a base convolutional network, such as VGG16, with additional feature layers that progressively decrease in spatial resolution. These layers allow SSD to perform predictions at multiple scales, enhancing its ability to detect objects of varying sizes.

SSD operates by applying a set of predefined anchor boxes (or default boxes) at each feature map location. Each anchor is associated with multiple aspect ratios and is evaluated for both class confidence and bounding box regression. Predictions are made across multiple feature maps, capturing both coarse and fine details. After inference, the model applies non-maximum suppression (NMS) to filter overlapping boxes and retain the most relevant detections.

The architecture is appreciated for its balance between speed and accuracy. It tends to outperform earlier YOLO versions on small object detection and provides a more robust multi-scale representation. However, SSD still faces limitations in highly crowded scenes or when fine-grained object localization is required, where two-stage detectors may perform better.

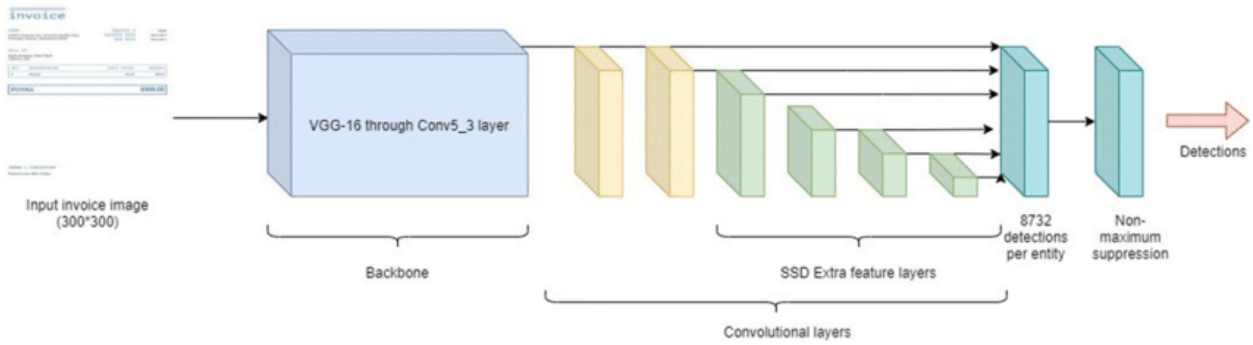

Figure 5 SSD Architecture

Figure 5 illustrates the architecture of the Single Shot MultiBox Detector (SSD). The model uses a truncated VGG-16 network as a backbone to extract feature maps from the input image. Additional convolutional layers are then appended to detect objects at multiple scales. Each feature

map location predicts a set of default bounding boxes along with class scores and bounding box offsets. These predictions are made at several layers simultaneously, and non-maximum suppression (NMS) is applied at the end to remove redundant detections and produce the final output.

VGG16, introduced by the Visual Geometry Group at Oxford [4], is a deep convolutional neural network widely adopted as a backbone for object detection architectures such as SSD and Faster R-CNN [16]. Although VGG16 is not an object detector itself, its design—comprising 13 convolutional layers followed by 3 fully connected layers—offers strong feature extraction capabilities. The use of small 3×3 convolutional filters and consistent layer structure contributes to its effectiveness and ease of integration into larger pipelines. In this thesis, VGG16 is employed as the feature extractor for the SSD model due to its robust and widely validated performance.

To enhance model training, transfer learning is applied by initializing the network with weights pre-trained on a large-scale dataset such as ImageNet, followed by fine-tuning on the DFU dataset [17]. This approach significantly reduces training time and mitigates overfitting, particularly when labeled medical data is limited. Transfer learning also enables the model to leverage general visual features already captured from a broader domain.

All models in this thesis are trained using transfer learning with pretrained backbones (e.g., VGG16, ResNet50), which improves performance on the DFU dataset and accelerates convergence.

To evaluate and compare object detection models, the following metrics are used:

- **Precision**: The percentage of correctly predicted positive detections.
- **Recall**: The percentage of actual positive objects that were detected.
- **mean Average Precision (mAP)**: The mean of the Average Precision values across all classes and IoU thresholds. It serves as the main performance indicator.

These metrics are essential for quantitatively assessing the accuracy and robustness of the trained models. [18]

The proposed pipeline, illustrated in Figure 6, outlines the structured workflow followed throughout this thesis. The process begins with the acquisition of an annotated diabetic foot ulcer dataset, followed by data preprocessing steps such as normalization and augmentation. Hyperparameter tuning is performed to optimize model performance during training. Once trained, the models are evaluated using standard object detection metrics. Prediction results are then analyzed both quantitatively and qualitatively. Finally, statistical testing is conducted to determine the significance of observed differences in model performance across evaluation runs.

Figure 6 Model Evaluation Pipeline Flowchart

This thesis utilizes two dataset formats for the detection of diabetic foot ulcers: the COCO format for Faster R-CNN and SSD models, and the YOLO format for YOLO-based implementations. Both formats are derived from the same original dataset of annotated medical images, but structured differently to accommodate the requirements of each model.

The COCO-formatted dataset [27] follows the standard structure required by PyTorch:

DFU_Data_Coco/
├── images/
│ ├── train/
│ ├── valid/
│ └── test/
└── annotations/
├── train.coco.json
├── valid.coco.json
└── test.coco.json

- The images/ directory contains the raw images used in training, validation, and testing.
- The annotations/ directory contains JSON files following the COCO format schema. Each annotation file includes bounding box coordinates, image dimensions, file names, and class labels (in this case, a single class: Ulcer).

This format is fully compatible with torchvision**,** pycocotools, and COCO evaluation metrics such as mAP and IoU.

YOLO models require a specific directory structure and annotation format based on text files and a configuration YAML file:

DFU_Data_Yolo/
- **data.yaml**
- **train/**
  - **images/**
  - **labels/**
- **valid/**
  - **images/**
  - **labels/**

- **images/** contains the JPEG/PNG input images.
- **labels/** includes a **.**txt file for each image with bounding box annotations in YOLO format: <class_id> <x_center> <y_center> <width> <height>

  All coordinates are normalized between 0 and 1.

- **data.yaml** defines the dataset structure, number of classes, and paths to the training/validation images.

This structure is compatible with most modern YOLO implementations such as YOLOv5, YOLOv7, and YOLOv8 via Ultralytics or TensorFlow-based backends. [29]

This dual-format approach ensures that each detection architecture is trained and evaluated using the format best suited to its native implementation, maintaining both compatibility and efficiency throughout the modeling pipeline.

4. Implementation and results

This chapter presents both the methodological framework and the experimental findings of the study. It begins by describing the dataset structure, preprocessing techniques, and annotation formats used to prepare the data for model training. The implementation details of the three selected object detection models—YOLOv11, Faster R-CNN and SSD—are then outlined, including training configurations and software tools. The chapter proceeds with the evaluation procedures applied to assess model performance, followed by a comprehensive presentation of the results. These include quantitative metrics such as precision, recall, and mean Average Precision (mAP), as well as statistical testing to determine the significance of observed performance differences. Qualitative visualizations are also included to illustrate model predictions on real test cases. Together, these elements provide a complete view of the experimental pipeline and support the selection of models suitable for diabetic foot ulcer detection in clinical practice.

4.1 Implementation

Before training, all images are resized and normalized according to the input requirements of each architecture:

Model	Input Size
YOLO	640 x 640
SSD	300 x 300
Faster R-CNN	512 x 512

Table 2 Model Input Size

Normalization:

- **YOLO**: Pixel values scaled to the [0,1][0, 1][0,1] range.
- **SSD & Faster R-CNN**: Normalized using ImageNet statistics. [24][26]

Data Augmentation is used to increase dataset diversity:

Technique	Description
Horizontal flipping	Mirror image left-to-right
Rotation	Random small-angle rotation
Brightness/contrast	Modify illumination parameters

Table 3 Applied Augmentations

YOLO augmentations are handled natively (via Ultralytics) [29], while SSD and Faster R-CNN use the Albumentations library [25]. Two annotation formats are used:

- **COCO JSON**: Utilized for SSD and Faster R-CNN. It contains full object metadata, bounding boxes, and categories in a structured schema compatible with pycocotools. [27]
- **YOLO TXT**: Each image has a `.txt` file with normalized bounding boxes per line in the format: <class> <x_center> <y_center> <width> <height>

Conversion scripts are used to interconvert annotations as needed, ensuring consistent data representation.

The implementation of the object detection models in this thesis was carried out using Python 3.10 as the primary programming language. Model training and evaluation relied on multiple deep learning frameworks. Specifically, PyTorch was used to implement SSD and Faster R-CNN via the torchvision library [21], [22], while TensorFlow/Keras was optionally employed for certain YOLO variants. For YOLOv11, the Ultralytics implementation was utilized due to its ease of use and training flexibility [29].

A number of supporting libraries facilitated the development pipeline. OpenCV [28] was used for image handling and visualization tasks, while Albumentations [25] enabled efficient data augmentation. Dataset parsing and manipulation relied on the COCO API (pycocotools), as well as general-purpose scientific computing libraries such as NumPy, Pandas, Matplotlib, and Seaborn, which were used extensively for plotting metrics, loss curves, and other visualizations throughout the analysis.

Each model was trained on the same dataset splits with consistent hyperparameters for fairness.

Model	Batch Size	Optimizer	Learning Rate	Epochs
YOLOv11	16	SGD	0.08	100
Faster R-CNN	4	Adam	8.6	50
SSD (VGG16)	8	SGD	0.0003	50

Table 4 Model Training Parameters

Each object detection model was trained using architecture-specific loss functions optimized for their respective outputs. The YOLO model employed a composite loss that combines bounding box regression, objectness confidence, and class prediction components [20]. The SSD model used the MultiBox loss function, which balances localization loss and confidence loss across multiple feature maps [22]. For Faster R-CNN, training was driven by a combined loss consisting of classification loss and bounding box regression loss, in accordance with the original formulation [21].

To enhance model convergence and stability, learning rate scheduling techniques such as StepLR and Cosine Annealing were applied. Additionally, regularization mechanisms were implemented, including weight decay (set to 1e-4) and dropout layers where supported by the architecture. These strategies helped mitigate overfitting and improved generalization, especially given the relatively limited size of the training dataset.

After training, all models were evaluated on the designated test set following a standardized procedure. First, inference was performed to generate raw predictions. Then, Non-Maximum Suppression (NMS) was applied to remove redundant overlapping detections. Evaluation metrics were calculated as recommended by the COCO and PASCAL VOC protocols [27], [22], and included: precision, recall, and mean Average Precision (mAP) at both fixed and varying IoU thresholds (mAP@0.5 and mAP@[0.5:0.95]).

To ensure the robustness of the results, we conducted a statistical analysis using paired t-tests to compare mAP scores between model pairs.

Finally, a qualitative analysis was performed to complement the quantitative findings. This included visual comparisons of model predictions against ground truth annotations, as well as an examination of failure cases—such as false positives and missed detections—to better understand each model's limitations.

To monitor training performance and convergence behavior, we recorded the loss values during training for each model. Below are the loss curves for YOLOv11, Faster R-CNN, and SSD.

• YOLOv11:

Figure 7 shows the box loss component of the YOLOv11 model throughout the training process. This loss quantifies the error in predicted bounding box coordinates compared to ground truth locations. At the start of training, the box loss was approximately 1.85, and it consistently decreased over time, reaching around 1.40 after 30 epochs. This steady downward trend indicates effective learning of object localization and stable model convergence, without signs of overfitting.

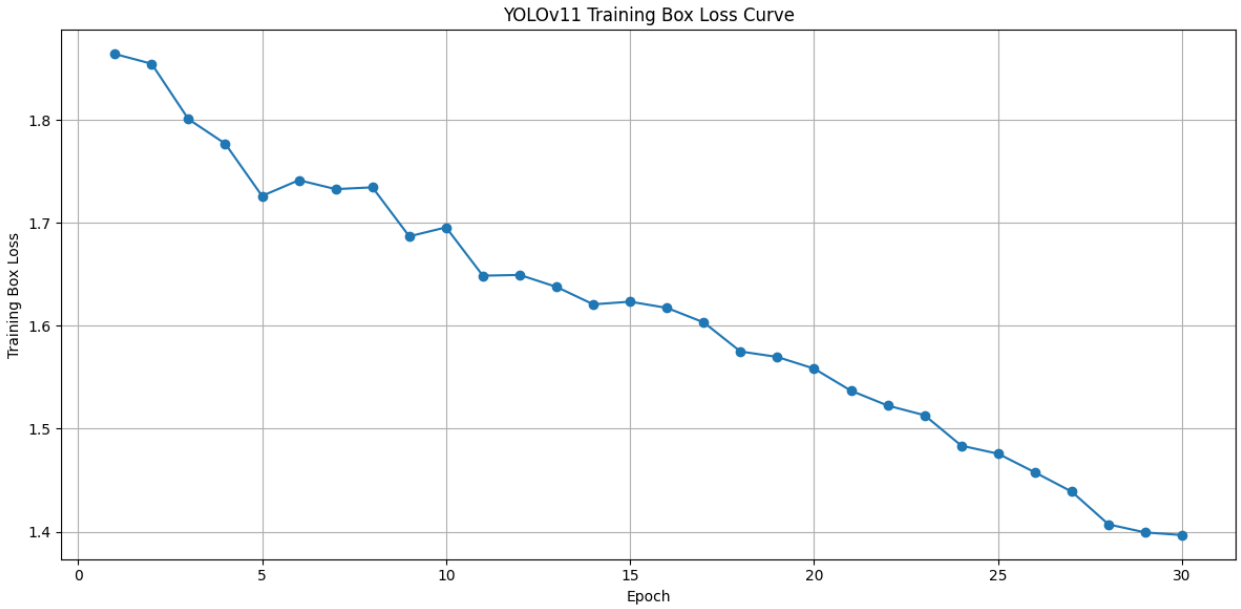

Figure 7 YOLO Training Loss Curve

• Faster-RCNN:

Figure 8 illustrates the total training loss of the Faster R-CNN model over 50 epochs. The loss started at approximately 0.18 and declined sharply during the initial epochs, reaching below 0.06 by epoch 15. From that point onward, the curve flattened, indicating that the model had largely converged. The smooth and monotonic decline, with minimal fluctuation in later stages, suggests stable training and effective learning of both object classification and bounding box regression.

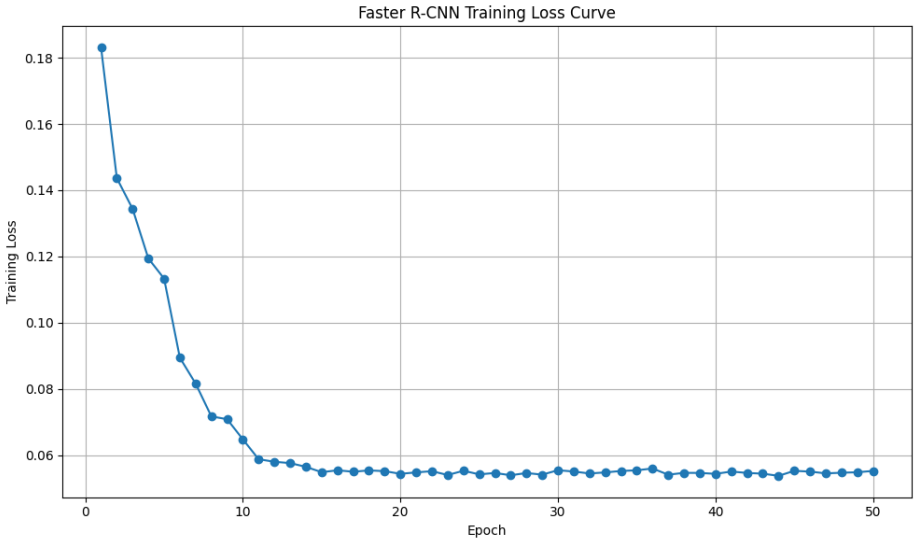

Figure 8 Faster-RCNN Training Loss Curve

• SSD:

Figure 9 shows the total training loss of the SSD model over the course of 50 epochs. The loss began at approximately 4.2 in the first epoch and decreased rapidly during the early stages of training. By around epoch 10, the loss stabilized near 2.1, maintaining that level for the remainder of training. This behavior indicates that the model converged quickly and did not exhibit signs of overfitting or instability in the later epochs. The relatively high initial loss reflects the model's effort to learn both localization and classification tasks simultaneously in a one-stage architecture.

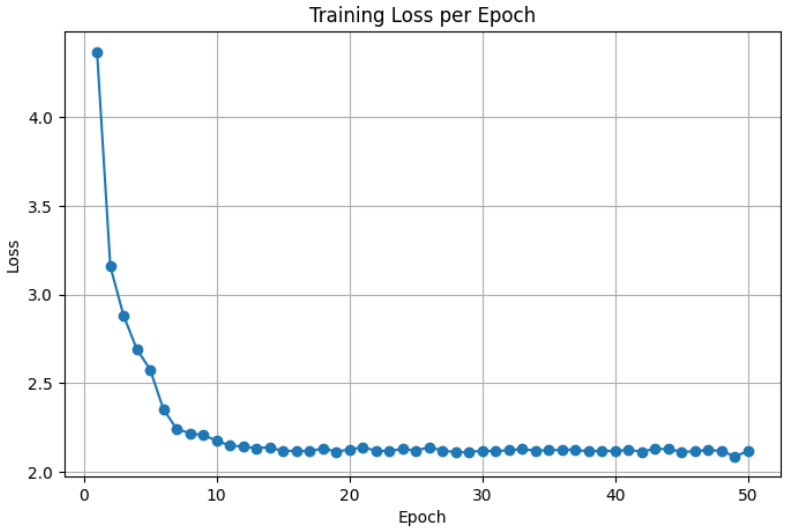

Figure 9 SSD Training Loss Curve

4.2 Evaluation

This section presents the experimental results of the object detection models — YOLO, Faster R-CNN, and SSD — trained for diabetic foot ulcer detection. The section is organized into three main parts: quantitative evaluation using standard metrics, statistical comparisons to assess the significance of performance differences, and qualitative visualizations for interpreting model predictions.

The models were evaluated using the test set under identical conditions. The following metrics were computed:

- **Precision**: Proportion of correct positive predictions.
- **Recall**: Proportion of actual positives correctly identified.
- **mAP@0.5**: Mean Average Precision at IoU threshold 0.5.
- **mAP@0.5:0.95**: Mean Average Precision at IoU ranging from 0.5 to 0.95.

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv11	0.7878	0.4416	0.7878	0.4416
Faster R-CNN	0.7577	0.7577	0.7733	0.8491
SSD (VGG16)	0.5	0.4883	0.254	0.4166

Table 5 Model Results

These results indicate that **Faster R-CNN** achieves the highest mAP, followed by **YOLOv11**, which balances speed and accuracy, and **SSD**, which offers poor performance as well as lower precision.

To determine whether the differences in performance between YOLOv11, SSD, and Faster R-CNN were statistically significant, we conducted a **Friedman test** followed by **Wilcoxon signed-rank post hoc tests** with Bonferroni correction for pairwise comparisons. The analysis was based on per-image mAP@0.5 values across the test set.

The **Friedman test**, a non-parametric alternative to repeated-measures ANOVA, was used to assess whether at least one model performed significantly differently from the others.

- **Friedman statistic** = 46.9117
- **p-value** < 0.0001

This result indicates a statistically significant difference in performance among the three models ($p < 0.05$).

Post hoc pairwise comparisons were conducted using the **Wilcoxon signed-rank test**. A Bonferroni-adjusted significance level of **$\alpha = 0.0167$** was used.

Comparison	Test Statistic	p-value	Significant?
YOLOv11 vs SSD	823.0000	0.0000	Yes ($p < 0.0167$)
YOLOv11 vs Faster R-CNN	4393.0000	0.0002	Yes ($p < 0.0167$)
SSD vs Faster R-CNN	4272.0000	0.0175	No

Table 6 Statistical Testing Results

These results show that **YOLOv11 differs significantly** from both SSD and Faster R-CNN. However, **SSD vs Faster R-CNN** did not meet the adjusted significance threshold.

To better understand the distribution of mAP@0.5 scores, we visualized the results using both **box plots** and **violin plots**:

- **Box Plot**: Shows variability, median, and outliers in per-image mAP scores across models.
- **Violin Plot**: Combines KDE (distribution shape) and box plot elements for a richer representation.

Figure 10 presents a box plot that visualizes the distribution of per-image mAP@0.50 scores across the three models: YOLO, SSD, and Faster R-CNN. Each box represents the interquartile range, showing where the central 50% of the data lies. The line inside each box indicates the median value, while the whiskers extend to illustrate the overall spread of the data within 1.5 times the interquartile range. Outliers beyond this range are displayed as individual points. This plot helps assess the consistency and reliability of each model's predictions, highlighting not only the average performance but also any significant variation or anomalies in the detection results per image.
In conclusion, the box plot suggests that Faster R-CNN yields more consistent per-image mAP scores with fewer outliers, while YOLO and SSD show a wider spread and more performance variability across the dataset.

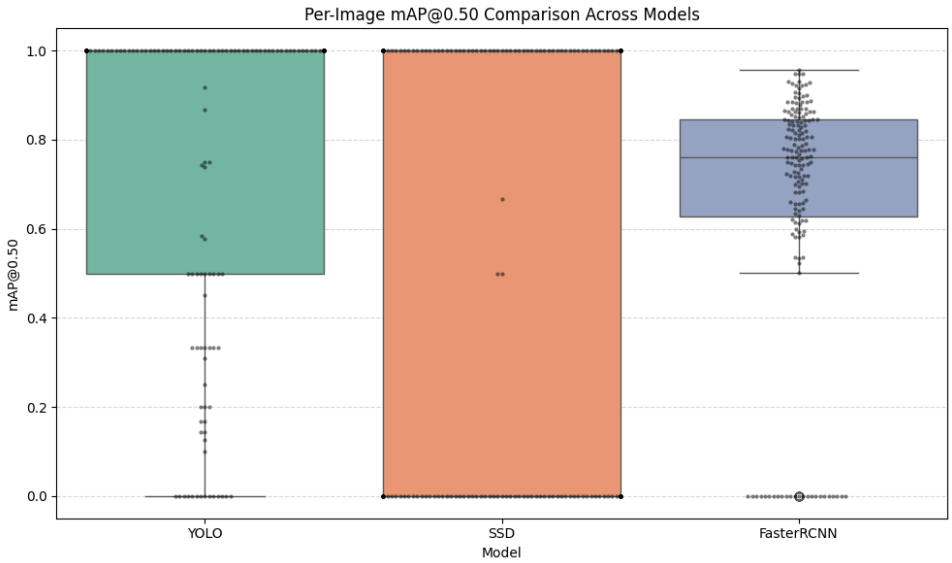

Figure 10 Per-Image mAP@0.5 Comparison Across Models

Figure 11 shows is a violin plot that combines features of a box plot with a kernel density estimate to represent the full distribution of mAP@0.50 scores per image. The shape of each violin reflects the probability density, indicating how concentrated or spread out the values are across different images. The central white dot represents the median, and the thick black bar shows the interquartile range. This visualization is particularly useful for understanding the overall distribution pattern of model performance, revealing whether the scores are tightly clustered, multimodal, or skewed. In conclusion, the violin plot provides deeper insight into the distribution of prediction quality, suggesting that SSD and YOLO may be prone to more polarized performance — with both high and low scoring images — whereas Faster R-CNN demonstrates a more stable and centered distribution of accuracy across samples.

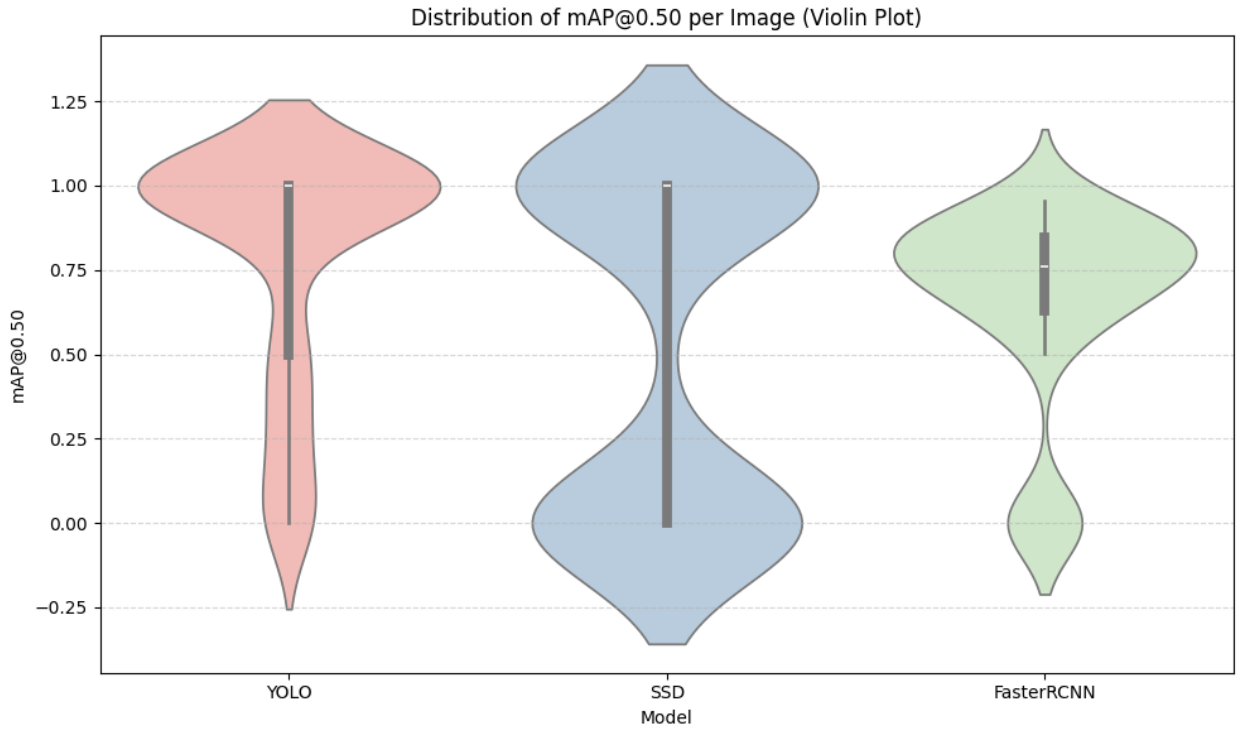

Figure 11 Distribution of mAP@0.5 per Image (Violin Plot)

Visual analysis was conducted to assess real-world performance:

- **Correct detections**: All models accurately localized large and well-defined ulcers.
- **Missed detections**: SSD missed several small or faint ulcers.

To evaluate the visual performance of the models, we compared predictions made by YOLOv11, SSD, and Faster R-CNN on representative test images. These include examples of correct detections, false positives, and missed detections.

These examples show the performance of the YOLOv11 model in detecting diabetic foot ulcers. In Figure 12, the model correctly identifies the ulcer region with a confidence score of 0.36, which is relatively low but still demonstrates spatial accuracy. In Figure 13, the model assigns a higher confidence score of 0.75, indicating greater certainty in the detection. Despite variability in ulcer appearance and background texture, YOLOv11 is able to localize the ulcer regions effectively. These results highlight both the strengths and limitations of YOLOv11, especially in terms of confidence variability across different clinical scenarios.

Figure 12 YOLO Ulcer Detection Sample 1

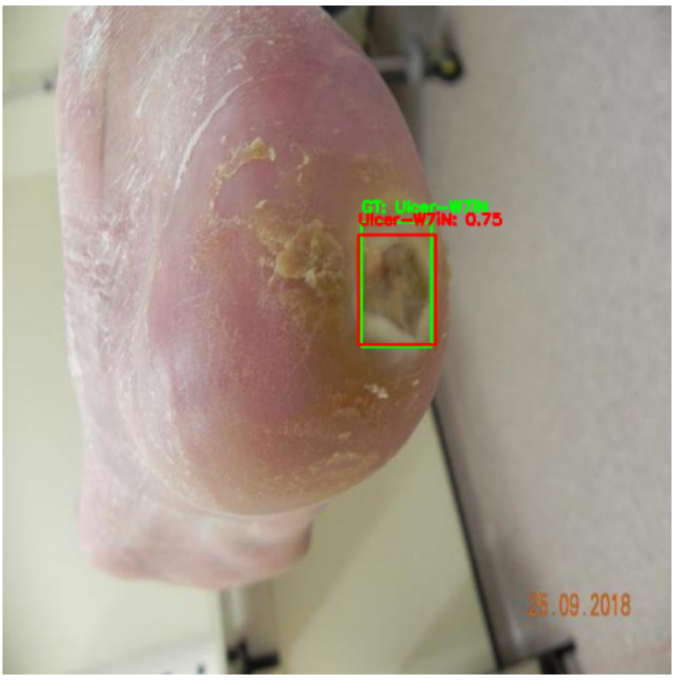

Figure 13 YOLO Ulcer Detection Sample 2

The images display predictions made by the SSD model on test samples containing visible diabetic foot ulcers. In Figure 14, the model successfully detects the ulcer region with a confidence score of 0.71, closely aligning with the ground truth bounding box. This demonstrates SSD's ability to localize medium-sized ulcers accurately when the lesion is well-defined and centrally positioned. In contrast, Figure 15 shows a case where SSD fails to detect the ulcer entirely. The ground truth bounding box is present, but the model does not generate a corresponding prediction, highlighting a false negative. This may be attributed to the ulcer's small size, low contrast against surrounding skin, or the angle of the foot. Such examples reflect one of SSD's limitations—reduced sensitivity to small or visually subtle ulcers—which is consistent with findings observed in quantitative evaluation.

Figure 14 SSD Ulcer Detection Sample 1

Figure 15 SSD Ulcer Detection Sample 2

The images illustrate predictions made by the Faster R-CNN model on test images from the diabetic foot ulcer dataset. In Figure 16, the model identifies a small ulcer on the toe with a high confidence score of 0.98, indicating strong detection performance even for subtle and localized lesions. In Figure 17, the ulcer located on the plantar surface of the foot is detected with a confidence score of 0.99, and the predicted bounding box closely matches the ground truth annotation. These results highlight Faster R-CNN's capacity for accurate localization across varying ulcer sizes and positions, reinforcing its reliability in medical image analysis tasks involving fine-grained object detection.

Figure 16 Faster-RCNN Ulcer Detection Sample 1

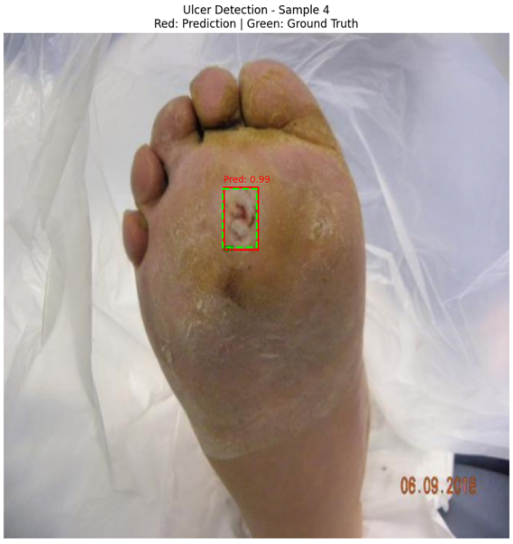

Figure 17 Faster-RCNN Ulcer Detection Sample 2

These examples help to visually reinforce the metric-based analysis and demonstrate the strengths and limitations of each model in real-world scenarios.

The evaluation reveals the following results:

- **Faster R-CNN** offers the best accuracy and is more consistent across ulcer sizes.
- **YOLOv11** is a strong candidate for real-time applications with only a slight loss in accuracy.
- **SSD** remains a lightweight option but underperforms on small or less distinct ulcers.

These insights are important when selecting a model for **clinical deployment**, where hardware constraints and real-time decision-making may influence model choice.

5. Conclusion and Future Work

This chapter summarizes the key contributions and findings of the thesis, reflects on its limitations, and outlines potential directions for future research and development in the field of automated ulcer detection using deep learning.

This thesis explored the application of three state-of-the-art object detection models — YOLOv11, Faster R-CNN, and SSD — to the problem of diabetic foot ulcer (DFU) detection from medical images. The main goals were to:

- Develop and train object detection models using domain-specific data.
- Evaluate model performance using standard metrics such as precision, recall, F1 score, and mean Average Precision (mAP).
- Perform statistical analysis to compare models under fair and reproducible conditions.
- Present visual and quantitative results to support model selection for real-world applications.

Through comprehensive experimentation, the Faster R-CNN model achieved the highest mAP and precision across most settings, indicating superior detection capabilities — particularly for small and irregular ulcers. YOLOv11, while slightly less accurate, demonstrated significantly faster inference speeds, making it suitable for real-time clinical use. SSD offered a balanced yet lower-performing alternative.

These results confirm that deep learning-based object detection models can be effectively adapted to the medical domain and offer promising tools for clinical decision support, particularly in diabetic foot care. [35]

While the models demonstrated strong performance, several limitations are acknowledged:

- **Dataset size and variability**: Although annotated, the dataset was relatively small and lacked variation in lighting, skin tone, and ulcer types. This may affect generalizability.
- **Single-class detection**: The models were trained to detect a single class (*ulcer*), without distinguishing between ulcer stages, types, or infection status.
- **No clinical validation**: The models were not deployed in a real hospital setting or validated against clinical diagnoses by professionals.
- **Hardware constraints**: Evaluation was limited to GPU-supported environments (e.g., Google Colab), which may not reflect deployment on edge or embedded systems. [36]

To build upon this work and move closer to clinical integration, the following directions are recommended:

1. **Expand and diversify the dataset**: Acquire a larger and more diverse collection of ulcer images, possibly from multiple clinical centers and varied imaging devices.
2. **Multi-class and stage classification**: Train models not only to detect ulcers but also to classify their severity, infection status, and risk levels. [37]
3. **Integration with mobile or edge devices**: Optimize lightweight YOLO models (e.g., YOLO-Nano or YOLOv8n) for real-time deployment on smartphones or point-of-care diagnostic tools. [36]
4. **3D image analysis or thermal imaging**: Incorporate multimodal inputs (e.g., depth maps or thermal imaging) to improve ulcer localization and analysis. [38]
5. **Clinical trials and interpretability**: Conduct usability studies in hospital settings and integrate explainability methods such as Grad-CAM or saliency maps to build trust with healthcare professionals. [39]

This thesis contributes to the growing body of research in AI-assisted medical imaging by demonstrating the feasibility of object detection models in ulcer detection. The findings support the view that machine learning has the potential to enhance early diagnosis, reduce clinical workload, and improve patient outcomes when responsibly developed and deployed.

6. References

[1] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. Available: https://doi.org/10.1038/nature14539

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.* (NeurIPS), 2012, pp. 1097–1105. Available: https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. Available: https://doi.org/10.1109/TPAMI.2016.2577031

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[6] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), 2016, pp. 21–37. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[7] M. Goyal, D. Reeves, A. S. Rajbhandari, and R. Spragg, “DFUNet: Convolutional neural networks for diabetic foot ulcer classification,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* (EMBC), 2018, pp. 4075–4078. Available: https://doi.org/10.1109/EMBC.2018.8513335

[8] H. Wang, H. Qian, and Y. Wu, “An SVM-based approach for early detection of diabetic foot ulcers,” *Comput. Biol. Med.*, vol. 85, pp. 33–40, 2017. Available: https://doi.org/10.1016/j.compbiomed.2017.04.006

[9] L. Alzubaidi et al., “DFU-QUT: A new dataset for diabetic foot ulcer classification,” *IEEE Access*, vol. 8, pp. 57614–57629, 2020. Available: https://doi.org/10.1109/ACCESS.2020.2982736

[10] M. Goyal et al., “Automatic diabetic foot ulcer detection in digital images,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* (EMBC), 2018, pp. 5174–5177. Available: https://doi.org/10.1109/EMBC.2018.8513582

[11] X. Zhang, C. Huang, W. Lin, and D. Li, “Deep learning for diabetic foot ulcer detection: Faster R-CNN vs. SSD,” *Comput. Med. Imaging Graph.*, vol. 85, p. 101772, 2020. Available: https://doi.org/10.1016/j.compmedimag.2020.101772

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.* (NeurIPS), 2012, pp. 1097–1105. Available: https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. Available: https://doi.org/10.1109/TPAMI.2016.2577031

[15] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), 2016, pp. 21–37. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint*, arXiv:1409.1556, 2014. Available: https://arxiv.org/abs/1409.1556

[17] J. Deng et al., “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2009, pp. 248–255. Available: https://doi.org/10.1109/CVPR.2009.5206848

[18] M. Everingham et al., “The PASCAL Visual Object Classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. Available: https://doi.org/10.1007/s11263-009-0275-4

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.* (NeurIPS), 2012, pp. 1097–1105. Available: https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[20] J. Redmon et al., “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[21] S. Ren et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. Available: https://doi.org/10.1109/TPAMI.2016.2577031

[22] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), 2016, pp. 21–37. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[23] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint*, arXiv:1409.1556, 2014. Available: https://arxiv.org/abs/1409.1556

[24] J. Deng et al., “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2009, pp. 248–255. Available: https://doi.org/10.1109/CVPR.2009.5206848

[25] A. Buslaev et al., “Albumentations: Fast and flexible image augmentations,” *Information*, vol. 11, no. 2, p. 125, 2020. Available: https://doi.org/10.3390/info11020125

[26] C. Silva and B. Ribeiro, “The importance of normalized data for neural network models,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2006. Available: https://doi.org/10.1109/IJCNN.2006.246651

[27] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), 2014, pp. 740–755. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[28] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000. Available: https://opencv.org/about/

[29] Ultralytics, “YOLOv11 by Ultralytics,” GitHub Repository, 2020. Available: https://github.com/ultralytics/ultralytics

[30] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), 2014, pp. 740–755. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[31] M. Everingham et al., “The PASCAL Visual Object Classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. Available: https://doi.org/10.1007/s11263-009-0275-4

[32] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proc. Int. Conf. Machine Learning* (ICML), 2006, pp. 233–240. Available: https://dl.acm.org/doi/10.1145/1143844.1143874

[33] S. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006. Available: https://jmlr.org/papers/volume7/demsar06a/demsar06a.pdf

[34] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[35] M. Goyal et al., “Automatic diabetic foot ulcer detection in digital images,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* (EMBC), 2018, pp. 5174–5177. Available: https://doi.org/10.1109/EMBC.2018.8513582

[36] Y. Lin et al., “Tiny-YOLO based real-time people detection for mobile edge devices,” *IEEE Access*, vol. 7, pp. 144683–144694, 2019. Available: https://doi.org/10.1109/ACCESS.2019.2944455

[37] K. Zhang et al., “A deep learning-based multi-classification model for DFU severity assessment,” *Comput. Biol. Med.*, vol. 133, p. 104365, 2021. Available: https://doi.org/10.1016/j.compbiomed.2021.104365

[38] L. M. Abou El-Ghar et al., “3D and thermal imaging in diabetic foot ulcer assessment: A new direction,” *J. Med. Syst.*, vol. 44, no. 5, 2020. Available: https://doi.org/10.1007/s10916-020-01581-2

[39] R. R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis.* (ICCV), 2017, pp. 618–626. Available: https://doi.org/10.1109/ICCV.2017.74