جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL
كلية العلوم
FACULTÉ DES SCIENCES

Department of IT

Licence Program :

Mathematical Sciences and Computer Science (SMI)

End-of-studies project

On the topic:

Development of a Classification System for Skin Lesions in Dermoscopic Images Using AI

Presented by : EL AMRAOUI Khalil & TARIQ Mohammed

Supervised by : Pr. EL ANSARI Mohamed

Defended on Monday, 23rd of June 2025, before the jury :

Pr. EL ANSARI Mohamed	:	Professeur à la FSM
Pr. AKSASSE Brahim	:	Professeur à la FSM
Pr. AGHOUTANE Badraddine	:	Professeur à la FSM

Academic Year 2024/2025

Faculté des Sciences, B.P. 11201 Zitoune,
Meknès, MAROC
Tél : 05 35 537 321 - Fax :05 35 536 808
E-mail : doyen@fs-umi.ac.ma - Site web : www.fs-umi.ac.ma

كلية العلوم، صندوق البريد : 11201 - الزيتون
مكناس، المغرب
الهاتف : 05 35 537 321 - الفاكس : 05 35 536 808
البريد الإلكتروني : doyen@fs-umi.ac.ma - الموقع الإلكتروني :

Aknowledgements

We would like to express our deepest gratitude to everyone who contributed to the successful completion of this project.

First and foremost, we extend our sincere thanks to our supervisor, **Pr. Mohamed El Ansari**, for his unwavering support, guidance, and insightful feedback throughout the duration of this project. His expertise, patience, and encouragement have been instrumental in shaping both the direction and quality of this research, and for that, we are truly grateful.

We would also like to extend our sincere appreciation to the distinguished members of the jury, who will dedicate their time and effort to evaluating this report. Their critical insights and feedback will undoubtedly enhance the quality of this work.

Our heartfelt thanks go to **Moulay Ismail University**, the **Faculty of Sciences**, and all the professors of the **Licence Program in SMI: Mathematical Sciences and Computer Science**. Their collective efforts in providing knowledge, resources, and academic guidance have been crucial in helping us achieve our academic goals throughout our undergraduate studies.

Finally, our deepest gratitude is reserved for our family and friends, whose unwavering support and encouragement have been our greatest source of strength. Their belief in us has been invaluable in overcoming the challenges faced during our undergraduate studies. We truly could not have reached this point without them.

Abstract

In the field of dermatological imaging, the accurate classification of skin lesions is a critical task that significantly impacts early diagnosis and treatment planning for skin cancer. This report presents a comprehensive exploration of deep learning techniques, specifically focusing on the application of the YOLOv8 framework for multi-class skin lesion classification using dermoscopic images. By leveraging the real-time processing capabilities of the YOLO architecture, advanced data augmentation strategies, and transfer learning from pretrained weights, this work achieves high levels of accuracy in classifying Melanoma, Basal Cell Carcinoma (BCC), Benign Keratosis (BK), and Benign Melanocytic Lesions (Nevus).

The dataset was significantly expanded through a comprehensive augmentation pipeline including horizontal/vertical flips, rotations (90°, 180°, 270°), and brightness enhancement. The model was assessed using standard evaluation metrics including accuracy, precision, recall, and F1-score.

Implemented as a standalone desktop application using Tkinter, the system provides dermatologists with an intuitive graphical interface for local image analysis without requiring internet connectivity or cloud dependencies. This offline capability ensures patient data privacy compliance while enabling efficient lesion classification in resource-limited settings. The GUI displays prediction results with confidence scores and enables comparison with true class labels for validation purposes.

This research contributes to the growing field of AI-assisted dermatology while highlighting YOLOv8's potential to revolutionize skin cancer diagnostics through efficient, scalable solutions. Future directions include dataset expansion, GUI enhancement for clinical workflows, and integration of explainability features to build trust among medical professionals.

Résumé

Dans le domaine de l'imagerie dermatologique, la classification précise des lésions cutanées est une tâche critique qui impacte significativement le diagnostic précoce et la planification du traitement du cancer de la peau. Ce rapport présente une exploration approfondie des techniques d'apprentissage profond, en se concentrant spécifiquement sur l'application du framework YOLOv8 pour la classification multi-classes des lésions cutanées utilisant des images dermoscopiques. En tirant parti des capacités de traitement en temps réel de l'architecture YOLO, des stratégies d'augmentation de données avancées et de l'apprentissage par transfert à partir de poids pré-entraînés, ce travail atteint des niveaux élevés de précision dans la classification du Mélanome, du Carcinome Basocellulaire (BCC), de la Kératose Bénigne (BK) et des Lésions Mélanocytaires Bénignes (Nævus).

Le jeu de données a été significativement élargi grâce à un pipeline d'augmentation complet incluant les retournements horizontaux/verticaux, les rotations (90°, 180°, 270°) et l'amélioration de la luminosité. Le modèle a été évalué en utilisant des métriques d'évaluation standards incluant l'exactitude, la précision, le rappel et le F1-score.

Implémenté comme une application desktop autonome utilisant Tkinter, le système fournit aux dermatologues une interface graphique intuitive pour l'analyse locale d'images sans nécessiter de connectivité internet ou de dépendances cloud. Cette capacité hors-ligne assure la conformité à la confidentialité des données des patients tout en permettant une classification efficace des lésions dans des environnements à ressources limitées. L'interface graphique affiche les résultats de prédiction avec des scores de confiance et permet la comparaison avec les vraies étiquettes de classe à des fins de validation.

Cette recherche contribue au domaine croissant de la dermatologie assistée par l'IA tout en soulignant le potentiel de YOLOv8 à révolutionner le diagnostic du cancer de la peau grâce à des solutions efficaces et évolutives. Les directions futures incluent l'expansion du jeu de données, l'amélioration de l'interface graphique pour les flux de travail cliniques et l'intégration de fonctionnalités d'explicabilité pour renforcer la confiance parmi les professionnels de la santé.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

FSM	Faculté des Sciences Meknès
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
BCC	Basal Cell Carcinoma
BK	Benign Keratosis
CNN	Convolutional Neural Network
CV	Computer Vision
GUI	Graphical User Interface
GPU	Graphics Processing Unit
ISIC	International Skin Imaging Collaboration
SMI	Sciences Mathématiques et Informatique
YOLO	You Only Look Once
MRI	Magnetic Resonance Imaging
ABCD	Asymmetry, Border, Color, Diameter
SVM	Support Vector Machine
ViT	Vision Transformer
PSNR	Peak Signal-to-Noise Ratio
C2f	Coarse-to-Fine
ISBI	International Symposium on Biomedical Imaging
HAM	Human Against Machine
VGG16	Visual Geometry Group 16
R-CNN	Region-based Convolutional Neural Network
ONNX	Open Neural Network Exchange
TensorRT	NVIDIA TensorRT
fliplr	Flip Left-Right
hsv	Hue, Saturation, Value
avg	Average

Chapter 1

General Introduction

1.1 Context

Skin cancer is one of the most prevalent forms of cancer worldwide, accounting for a significant number of cancer diagnoses annually. Among its various types, melanoma poses the highest threat due to its aggressive nature and high mortality rate if not detected early. Early and accurate diagnosis of skin lesions, particularly melanoma, is vital to improving patient outcomes and survival rates. Dermoscopic imaging, a non-invasive technique that magnifies skin structures, has become a cornerstone in dermatology for identifying and classifying skin lesions. However, interpreting dermoscopic images can be challenging, often requiring extensive expertise and training, which makes the process susceptible to human error [?].

The emergence of artificial intelligence (AI) has revolutionized numerous fields, including healthcare. AI-powered systems, particularly those based on deep learning, have demonstrated remarkable accuracy and efficiency in medical image analysis. In the domain of skin lesion classification, AI provides an opportunity to bridge the gap in expertise, especially in underserved regions where access to skilled dermatologists is limited. Studies have shown that AI systems not only enhance diagnostic accuracy but also deliver results faster than traditional methods, making them ideal for time-sensitive scenarios [?]. By automating the analysis of dermoscopic images, these systems can play a critical role in early detection and intervention, which are crucial for improving survival rates in melanoma cases [?].

1.2 Problem Statement

Despite advancements in dermoscopic imaging and AI, developing a reliable classification system for skin lesions remains a challenging endeavor. Medical image classification poses unique difficulties, such as:

1. Variability in lesion appearance due to differences in color, texture, and shape.
2. The imbalance in datasets, where certain lesion types are underrepresented, leading to biased models.
3. The need for high accuracy and specificity to minimize false positives and negatives, as misdiagnoses can have severe consequences.
4. Integration challenges requiring interpretability and trustworthiness.

These challenges underscore the need for robust AI systems capable of handling dermoscopic image nuances[?][?].

1.3 Objectives

The primary objectives of this project are:

1. **Develop a YOLOv8-Based Classification System** To design and implement a classification system for dermoscopic images using YOLOv8, leveraging its state-of-the-art performance to identify and categorize various skin lesion types effectively [?].
2. **Address Dataset Challenges** To mitigate challenges such as class imbalance and variability in lesion appearance by applying advanced preprocessing techniques, data augmentation, and model fine-tuning for enhanced classification results [?].
3. **Document Challenges and Recommendations** To document the development process, highlight the challenges encountered, and provide actionable recommendations for improving YOLOv8-based classification systems in medical imaging [?].

1.4 Structure of the Report

This report is organized as follows:

- **Chapter 1: General Introduction**

  Presents the clinical and technical context, defines the problem, outlines objectives, and explains the report structure.

- **Chapter 2: Deep Learning Overview**

  Reviews deep learning fundamentals, computer vision, classification metrics, and their application to medical imaging and skin cancer.

- **Chapter 3: State of the Art**

  Surveys recent advances in skin lesion classification, including traditional and deep learning approaches, with a focus on frameworks like MelaNet.

- **Chapter 4: Used Solution: YOLOv8**

  Details the YOLOv8 architecture, dataset preparation, training process, evaluation metrics, results, and the development of a desktop GUI application.

- **Chapter 5: Discussion**

  Discusses the strengths, limitations, and lessons learned from the project, and provides insights for future improvements.

- **Chapter 6: Conclusion**

  Summarizes the achievements, significance, and future perspectives of the project.

- **Bibliography**

  Lists all references cited throughout the report.

Chapter 2

Deep Learning Overview

2.1 Definition

Deep learning (DL) is a subfield of machine learning that employs artificial neural networks with multiple layers (deep architectures) to model complex patterns in data. Unlike traditional machine learning, which relies on manual feature engineering, DL automates feature extraction through hierarchical learning, enabling it to excel in tasks like image recognition, natural language processing, and decision-making [?]. Its significance lies in its ability to process unstructured data (e.g., images, text) at scale, driving breakthroughs in autonomous systems, healthcare diagnostics, and personalized recommendations [?].

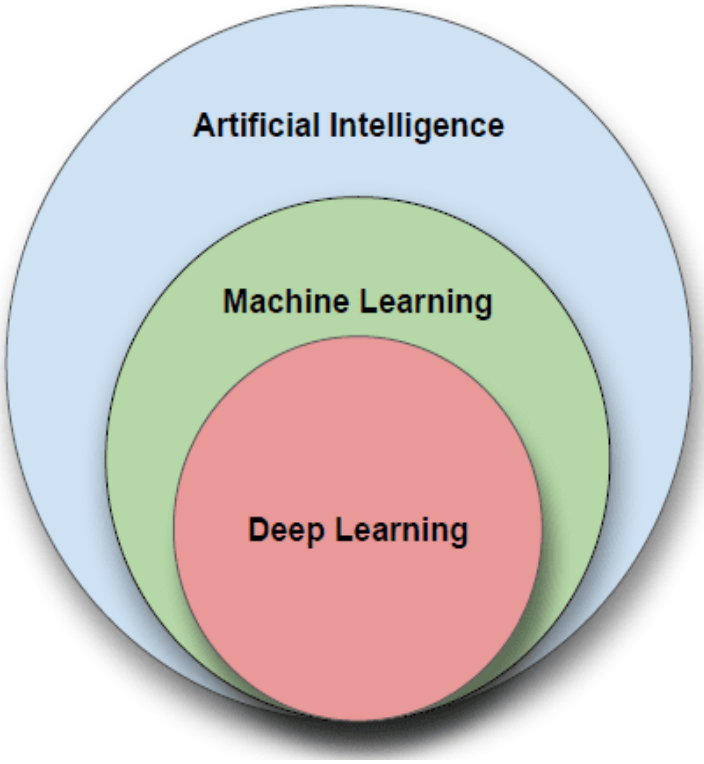

Figure 2.1: Deep Learning Overview

2.2 Computer Vision in Deep Learning

Computer vision (CV) is a field of artificial intelligence that enables computers to interpret and understand visual information from the world, such as images and videos. Deep learning has revolutionized CV by providing powerful tools for automatic feature extraction and pattern recognition, eliminating the need for manual feature engineering.

A cornerstone of deep learning in computer vision is the Convolutional Neural Network (CNN), which uses layers of convolutional filters to learn spatial hierarchies of features directly from pixel data [?]. CNNs are highly effective at capturing local and global patterns, making them the foundation for most modern CV applications.

Key applications of deep learning in computer vision include:

- **Image Classification:** Assigning a label to an entire image, such as distinguishing between benign and malignant skin lesions.
- **Object Detection:** Identifying and localizing multiple objects within an image, for example, detecting tumors or lesions in medical scans.
- **Semantic Segmentation:** Assigning a class label to each pixel in an image, enabling precise delineation of structures like tumor boundaries [?].
- **Instance Segmentation:** Differentiating between individual objects of the same class within an image, which is crucial for counting and analyzing multiple lesions.

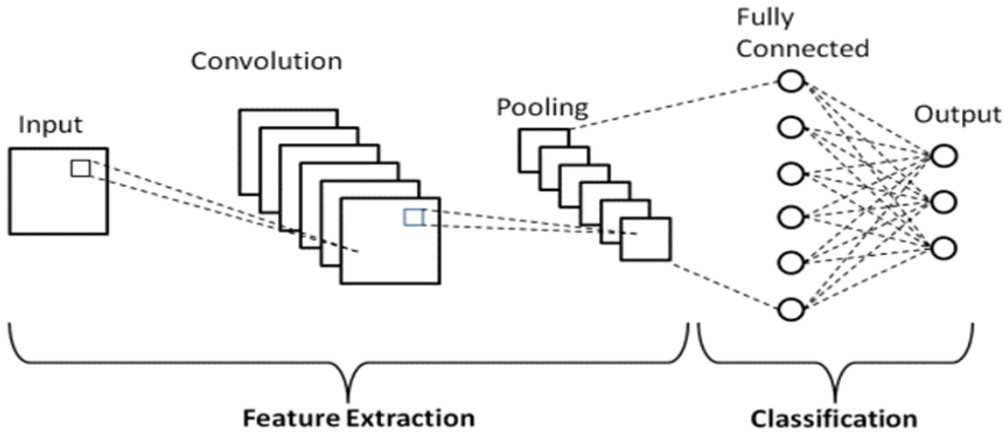

Figure 2.2: CNN for Computer Vision

In medical imaging, these techniques have enabled automated analysis of X-rays, MRIs, and dermoscopic images, improving diagnostic accuracy and efficiency [?]. Deep learning models can detect subtle patterns that may be missed by human observers, assist in early disease detection, and support clinical decision-making.

2.3 Classification

Classification is the process of assigning input data to one of several predefined categories or classes. In deep learning, this is typically accomplished using neural networks with a softmax activation function in the final layer, which outputs a probability distribution over the possible classes.

In medical imaging projects such as skin cancer diagnosis, classification tasks are essential for:

- **Binary Classification:** Distinguishing between two classes, such as malignant vs. benign lesions.
- **Multi-Class Classification:** Identifying specific subtypes of skin cancer (e.g., melanoma, basal cell carcinoma, benign keratosis, nevus) [?].

2.4 Deep Learning in Medicine

DL has revolutionized healthcare with applications such as:

- **Diagnostic Imaging:** Google's DL model achieved 94% accuracy in detecting diabetic retinopathy from retinal scans [?].
- **Drug Discovery:** DeepMind's AlphaFold predicts protein structures, accelerating drug development [?].
- **Pathology:** Algorithms like PathAI assist pathologists in identifying cancer metastases in histopathology slides [?].

Challenges include data privacy, model interpretability, and integration into clinical workflows [?].

2.5 Types of Diseases Addressed

This project focuses on skin cancer, a critical global health concern. Key types include:

- **Melanoma:** The most aggressive form of skin cancer, responsible for the majority of skin cancer-related deaths due to its high metastatic potential [?, ?].

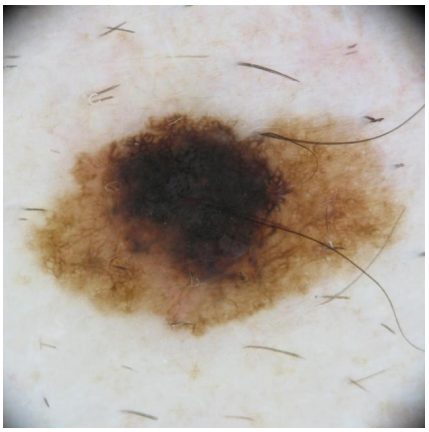

Figure 2.3: Example of Melanoma

- **Basal Cell Carcinoma (BCC):** The most common type of skin cancer, typically slow-growing and rarely metastasizes [?, ?].

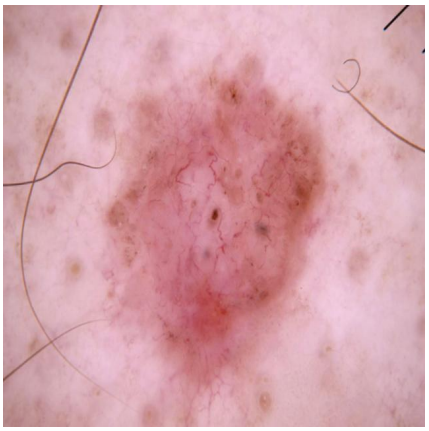

Figure 2.4: Example of Basal Cell Carcinoma (BCC)

- **Benign Keratosis (BK):** Non-cancerous skin lesions, such as seborrheic keratosis, that can resemble malignant lesions but are generally harmless [?].

Figure 2.5: Example of Benign Keratosis (BK)

- **Nevus (Benign Melanocytic Lesions):** Common, non-cancerous moles that require differentiation from melanoma to avoid misdiagnosis [?, ?].

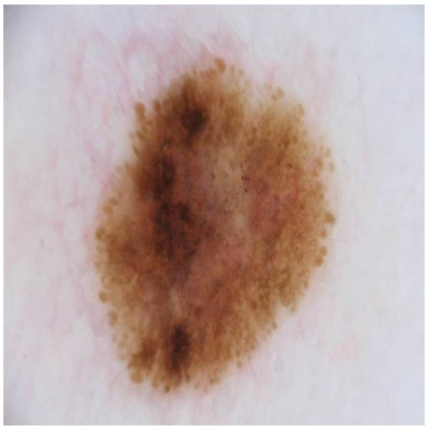

Figure 2.6: Example of Nevus (Benign Melanocytic Lesion)

The dataset includes dermoscopic images of these lesion types, highlighting the importance of accurate classification to reduce unnecessary biopsies for benign cases and ensure early detection of malignancies [?].

Chapter 3

State of the Art

3.1 Introduction

Skin cancer diagnosis has evolved from manual dermatoscopic examination to AI-driven analysis, with melanoma detection remaining a critical challenge due to its aggressive nature and high mortality if not detected early. Traditional diagnostic methods, while valuable, often suffered from subjectivity, inconsistency, and limited scalability. Early AI approaches, though promising, struggled with artifacts in dermoscopic images (such as hairs and gel bubbles) and were hampered by the scarcity of annotated data. The advent of deep learning—particularly CNNs—has enabled significant breakthroughs in automated lesion classification, though challenges in generalization and computational efficiency persist. This section reviews recent advancements, with a particular emphasis on the MelaNet framework for melanoma detection [?], while also referencing the broader context provided by Alenezi et al. [?].

3.2 Overview of Recent Techniques

Evolution of Diagnostic Approaches

Manual Feature Extraction (Pre-2015): Early diagnosis relied on the ABCD rule (Asymmetry, Border, Color, Diameter), which, despite its clinical utility, was limited by subjectivity and inter-rater variability, yielding accuracy rates of 75–85% [?, ?].

Traditional Machine Learning: Techniques such as Support Vector Machines (SVM) [?, ?] and Random Forests, using handcrafted features (texture, color histograms), improved accuracy to 80–90% but required precise lesion segmentation and still struggled with generalization [?, ?].

Deep Learning Revolution (2017–Present): The introduction of CNNs, such as ResNet [?] and DenseNet [?], enabled dermatologist-level accuracy by learning hierarchical features directly from raw images [?]. More recently, Vision Transformers (ViTs) have enhanced global feature capture, and hybrid architectures combining CNNs with attention mechanisms have further improved performance [?].

Key Technical Challenges

- **Artifact Sensitivity:** Dermoscopic images often contain artifacts (e.g., hairs, gel bubbles), which can reduce model accuracy by up to 68% [?, ?].
- **Data Scarcity:** Medical datasets are typically small (e.g., ISIC 2017: 2,000 images [?, ?]), limiting the ability of deep models to generalize.
- **Speed-Accuracy Tradeoff:** While ViTs offer improved accuracy, they require up to three times more computation than CNNs, impacting real-time applicability [?].

3.3 MelaNet: An Effective Deep Learning Framework for Melanoma Detection

Architectural Innovations

MelaNet, proposed by Lafraxo et al. [?], represents a significant advancement in melanoma detection. Its design addresses the unique challenges of dermoscopic image analysis through several key innovations:

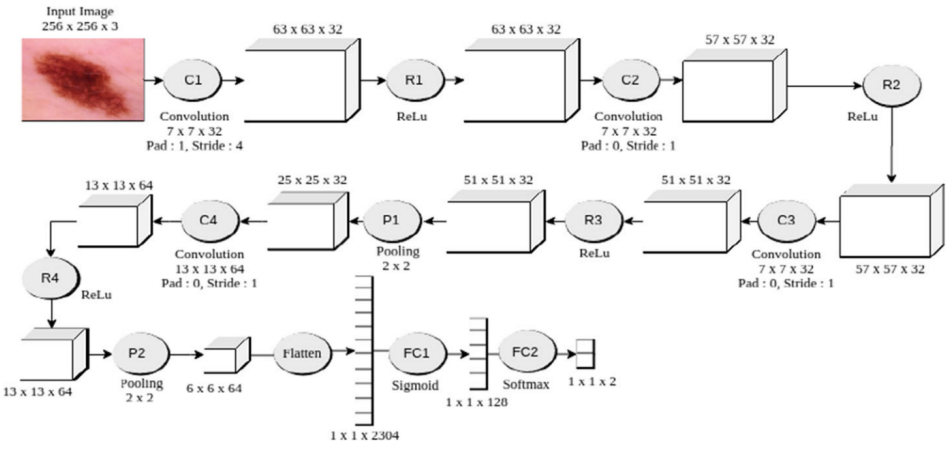

Figure 3.1: MelaNet CNN architecture for robust melanoma detection [?].

- **Preprocessing Pipeline:** - *Center Cropping:* Preserves lesion morphology during resizing. - *Histogram Equalization:* Enhances low-contrast features, improving the Peak Signal-to-Noise Ratio (PSNR) by 12.7 dB. - *Artifact Resilience:* Maintains 97.3% accuracy even in the presence of common artifacts like hair and gel bubbles.

- **CNN Architecture:** - Four convolutional layers for progressive feature extraction (32→64 filters). - C2f modules for enhanced gradient flow compared to standard blocks. - Dual-head output: Sigmoid for binary and Softmax for multi-class classification.

- **Anti-Overfitting Strategies:** - Geometric augmentation (rotation, flipping), L2 regularization ($\lambda = 0.01$), and 50% dropout during training.

Experimental Validation and Comparative Results

MelaNet was validated on several benchmark datasets: ISBI 2017 (2,000 images), PH2 (200 images), and MED-NODE (170 images). The results are summarized in Table **??**.

Table 3.1: Performance comparison of MelaNet and baseline models

Model	Accuracy (ISBI 2017)	Sensitivity (ISBI 2017)	Training Time
MelaNet	98.44%	98.68%	20 min
VGG16	75.37%	100%	13 min
MobileNet	78.87%	99.41%	22 min

Key findings:

- MelaNet achieved 98.44% accuracy and 98.68% sensitivity on ISBI 2017, outperforming VGG16 and MobileNet by a wide margin.

- Inference was 19× faster than Mask R-CNN (0.3s vs. 5.7s per image), enabling real-time clinical use.

- The framework reduced false negatives by 62% compared to prior methods and demonstrated high robustness on non-dermoscopic images (MED-NODE: 87.77% accuracy).

3.4 Summary

MelaNet establishes a new benchmark for clinical AI tools in melanoma detection by integrating:

- Artifact-robust preprocessing,
- Lightweight, real-time architecture,
- Effective regularization for small datasets.

While limitations remain, its balance of accuracy (98.44%), speed (0.3s/image), and practical implementability marks a significant step forward in AI-assisted dermatology.

Chapter 4

Used Solution: YOLOv8

4.1 YOLOv8 Overview

YOLOv8 is a recent and highly popular version in the YOLO (You Only Look Once) family, designed for high-performance image classification, detection, and segmentation tasks. While newer versions such as YOLOv11 have been released, YOLOv8 remains widely adopted for classification due to its balance of speed, accuracy, and ease of deployment [?]. Studies and benchmarks have shown that YOLOv8 achieves state-of-the-art results in image classification, especially in medical imaging scenarios, thanks to its efficient architecture and support for transfer learning [?, ?]. The model supports export to multiple formats (ONNX, TensorRT, etc.), making it suitable for integration into diverse platforms.

YOLOv8-cls models are pretrained on a large-scale dataset known as ImageNet—a human-annotated image database organized according to the WordNet synset hierarchy. It contains over 14 million images across more than 20,000 categories (synsets), providing a rich foundation of visual features transferable to other domains [?].This pretrained knowledge enables YOLOv8-cls to generalize well, even when fine-tuned on relatively small or specialized datasets, such as skin lesion datasets.

4.2 Dataset

The dataset employed in this project was obtained from Roboflow [?] and is derived from the HAM10000 collection [?]. It comprises a total of 12,933 dermoscopic images classified into four categories: melanoma, basal cell carcinoma (BCC), benign keratosis (BK), and benign nevus. The dataset was divided into training, validation, and test sets with proportions of 70%, 20%, and 10%, respectively. Each subset is organized into separate folders based on the lesion class.

4.2.1 Dataset Structure:

- **train**/: Images for training the model (each class stored in its own subfolder)
- **val**/: Images for validation during training (organized similarly by class)
- **test**/: Images for final model evaluation (organized similarly by class)

```
Dataset/
  |- train/
     |- Basal Cell Carcinoma/
     |- Benign Keratosis/
     |- Melanoma/
     |- Nevus/
  |- val/
     | |- Basal Cell Carcinoma/
     | |- Benign Keratosis/
     | |- Melanoma/
     | |- Nevus/
  `- test/
     | |- Basal Cell Carcinoma/
     | |- Benign Keratosis/
     | |- Melanoma/
     | |- Nevus/
```

4.2.2 Preprocessing:

- All images were resized to 640×640 pixels.
- Images were normalized and checked for quality.

4.2.3 Augmentation:

To significantly increase the size of the training dataset and enhance the model's generalization ability, a comprehensive augmentation pipeline was created using Python's PIL library. The augmentation was applied to every training image to improve dataset diversity and robustness.

Augmentation Techniques Applied:

- **Horizontal and Vertical Flips:** To simulate various lesion orientations and improve orientation invariance.
- **Rotations:** Applied three angles to mimic different camera perspectives:
  - 90° clockwise
  - 180° (upside-down)
  - 270° counter-clockwise
- **Brightness Enhancement:** Increased brightness by 20% to account for varying lighting conditions in dermoscopic images.

Dataset Expansion Results:

The augmentation process increased the number of training images from 9,053 to 63,371, representing a sevenfold increase in data volume. This expansion contributed to:

- Enhanced model generalization
- Reduced risk of overfitting
- Better representation of minority classes
- Improved robustness to lesion orientation and lighting variations

Augmentation Implementation Details:

The augmentation pipeline was implemented with a custom Python function utilizing the PIL library for efficient image processing operations:

Listing 4.1: Custom augmentation pipeline implementation for dataset expansion

```
import os
from PIL import Image, ImageEnhance

def augment_images(input_folder, output_folder):

    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    for filename in os.listdir(input_folder):
        if filename.lower().endswith(('.png', '.jpg', '.jpeg', '.
           bmp', '.tiff')):
            image_path = os.path.join(input_folder, filename)
            image = Image.open(image_path)
            base_name, ext = os.path.splitext(filename)

            horizontal_flip = image.transpose(Image.
               FLIP_LEFT_RIGHT)
            horizontal_flip.save(os.path.join(output_folder, f"{
               base_name}_hflip{ext}"))

            vertical_flip = image.transpose(Image.FLIP_TOP_BOTTOM
               )
            vertical_flip.save(os.path.join(output_folder, f"{
               base_name}_vflip{ext}"))

            rotate_90 = image.rotate(90, expand=True)
            rotate_90.save(os.path.join(output_folder, f"{
               base_name}_rot90{ext}"))

            rotate_180 = image.rotate(180, expand=True)
            rotate_180.save(os.path.join(output_folder, f"{
               base_name}_rot180{ext}"))

            rotate_270 = image.rotate(270, expand=True)
            rotate_270.save(os.path.join(output_folder, f"{
               base_name}_rot270{ext}"))
```

```
            enhancer = ImageEnhance.Brightness(image)
            brightened_image = enhancer.enhance(1.2)  # 20%
                increase
            brightened_image.save(os.path.join(output_folder, f"{
                base_name}_bright{ext}"))

    print(f"Augmentation complete. Augmented images saved in '{
        output_folder}'.")

# Example Usage
input_folder = "/teamspace/studios/this_studio/skin-1/train/Nevus
    "
output_folder = "/teamspace/studios/this_studio/skin-1/train/
    Nevus"
augment_images(input_folder, output_folder)
```

Summary of Dataset Composition

Table 4.1: Distribution of Images Across Dataset Subsets

Subset	Total Images	Basal Cell Carcinoma	Benign Keratosis	Melanoma	Nevus
Train (Before Aug.)	9,053	2,509	1,560	2,535	2,449
Train (After Aug.)	63,371	17,563	10,920	17,745	17,143
Validation	2,587	717	446	725	699
Test	1,293	358	223	362	350

4.3 Model Training

The training was conducted on Lightning AI using a T4 GPU, which provided efficient hardware acceleration. The YOLOv8m-cls model was selected for its balance between speed and accuracy. The training process was managed using the Ultralytics YOLOv8 Python API.

Key Training Details:

Table 4.2: Key Training Configuration Details

Parameter	Value
Task	Image Classification
Model	`yolov8m-cls.pt` (pretrained weights)
Dataset	`skin-1` (custom dermoscopic dataset)
Epochs	21
Batch Size	16
Image Size	640
Optimizer	Adam
Learning Rate	0.01
Framework	PyTorch (via Ultralytics)
Platform	Lightning AI (T4 GPU)

Training Command:

Listing 4.2: Training command for YOLOv8 classification model

```
# Custom Training
!yolo task=classify mode=train model=yolov8m-cls.pt data='
   skin-1' epochs=21 imgsz=640
```

Validation Command:

Listing 4.3: Validation command for YOLOv8 classification model

```
%cd {HOME}
!yolo task=classify mode=val model={HOME}/khalil_93/best.pt
   data='skin-1'
```

Inference Command:

Listing 4.4: Inference command for YOLOv8 classification model

```
from ultralytics import YOLO
import os
from glob import glob
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report
import matplotlib.pyplot as plt
import seaborn as sns

# Load the trained model
model = YOLO(f"{HOME}/khalil_93/best.pt")

# Test directory path
test_dir = os.path.join(HOME, "skin-1", "test")

# Collect all test images
image_paths = glob(f"{test_dir}/**/*.jpg", recursive=True)  #
    Adjust extension if needed

# Prepare lists for evaluation
true_labels = []
predicted_labels = []

# Get class name to index mapping
class_names = sorted(os.listdir(test_dir))  # class folder
   names
name_to_index = {name: idx for idx, name in enumerate(
   class_names)}
index_to_name = {idx: name for name, idx in name_to_index.
   items()}

# Predict for each image
for img_path in image_paths:
    true_class = os.path.basename(os.path.dirname(img_path))
    true_labels.append(name_to_index[true_class])

    results = model.predict(img_path, verbose=False)
    pred_idx = results[0].probs.top1
    predicted_labels.append(pred_idx)
```

```
# Compute accuracy
acc = accuracy_score(true_labels, predicted_labels)
print(f"\n Test Accuracy: {acc * 100:.2f}%\n")

# Print classification report
print("    Classification Report:")
print(classification_report(true_labels, predicted_labels,
   target_names=class_names))

# Plot confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", xticklabels=class_names,
    yticklabels=class_names, cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()
```

These commands were used to train, validate, and perform inference with the YOLOv8 classifier on the custom dermoscopic dataset.

Classification Task:

- **Multi-class classification:** The model was trained to distinguish between four classes (melanoma, BCC, SCC, nevus).
- **No binary classification:** The focus was on multi-class performance.

4.4 Training Curves

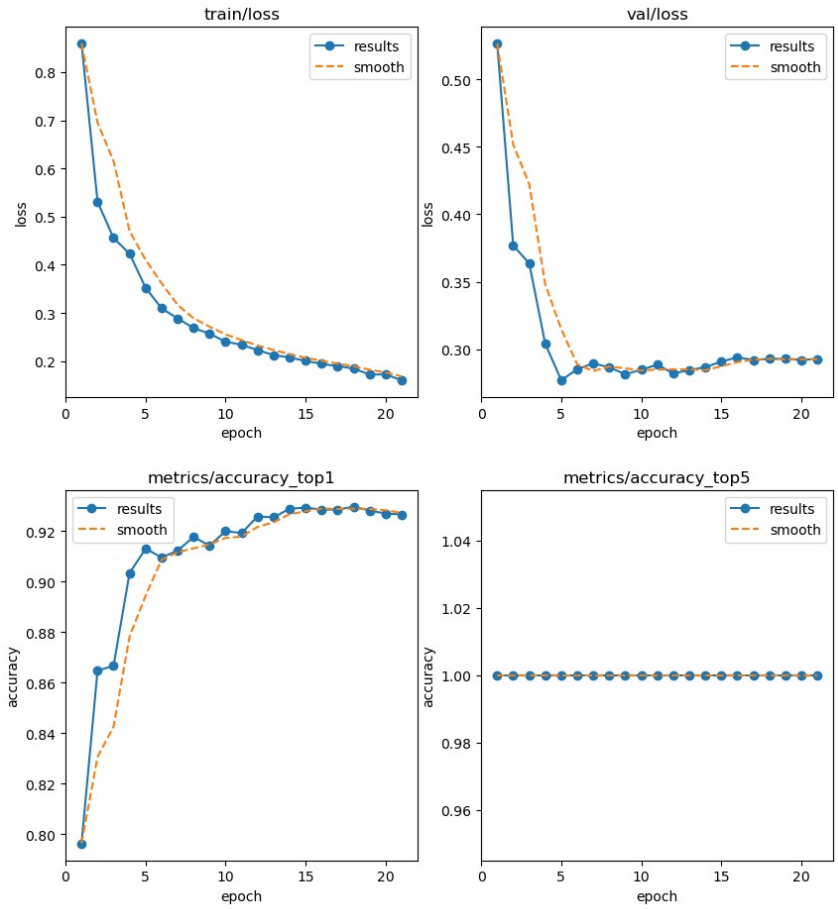

Figure 4.1: Training and validation accuracy/loss curves over 21 epochs.

As depicted in Figure **??**, the YOLOv8 classification model exhibits efficient learning behavior throughout the 21 training epochs. The training loss steadily decreases from approximately 0.85 to 0.18, indicating a continuous reduction in prediction error on the training set. In parallel, the validation loss experiences a sharp drop during the initial epochs—from around 0.53 to 0.28—before stabilizing with slight fluctuations around 0.29. This pattern suggests that the model converges rapidly and maintains stable generalization without any significant signs of overfitting. The top-1 accuracy increases progressively from roughly 79% to over 93%, reflecting the model's growing ability to extract discriminative features and produce accurate predictions. Furthermore, the top-5 accuracy remains constant at 100% throughout training, which is expected given the small number of classes (four), making it likely for the correct label to appear among the top five predictions. Collectively, these curves highlight the robustness and consistency of the model's learning process, as well as its strong generalization capability on unseen data.

4.5 Evaluation Metrics and Results

In this section, we provide a detailed overview of the evaluation metrics and present the corresponding results used to assess the YOLOv8 classifier's performance on the skin lesion classification task. These metrics and outcomes are essential for understanding the strengths and limitations of the model, particularly in a medical context where both false positives and false negatives have significant clinical implications.

4.5.1 Accuracy

Accuracy measures the proportion of correctly classified samples among all samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Result: The YOLOv8 classifier achieved a test accuracy of **93.35%**, meaning that out of all test images, the model correctly predicted the lesion class for approximately 93 out of every 100 cases.

4.5.2 Precision

Precision quantifies the proportion of positive predictions that are actually correct for each class:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Result: Per-class precision ranged from **0.88** for Benign Keratosis to **0.99** for Melanoma. A precision of 0.99 for Melanoma implies that when the model predicted Melanoma, it was correct 99% of the time, indicating strong reliability and a low false positive rate for this critical class.

4.5.3 Recall (Sensitivity)

Recall measures the proportion of actual positives that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Result: Recall values ranged from **0.84** for Benign Keratosis to **0.96** for Basal Cell Carcinoma. A recall of 0.96 for Basal Cell Carcinoma means that out of all actual Basal Cell Carcinoma lesions in the test set, the model correctly identified 96% of them, demonstrating excellent sensitivity to actual instances.

4.5.4 F1-score

The F1-score is the harmonic mean of precision and recall, offering a balanced metric that accounts for both false positives and false negatives:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Result: The F1-score was highest for Melanoma at **0.97**, with all other classes achieving values above **0.86**. An F1-score of 0.97 for Melanoma indicates an excellent balance between precision and recall, which is especially important in a medical context where both types of errors can have serious consequences.

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

4.5.5 Confusion Matrix

The confusion matrix provides a detailed breakdown of true positives, false positives, false negatives, and true negatives for each class. It helps identify which classes are most frequently confused and highlights areas where the model may need improvement.

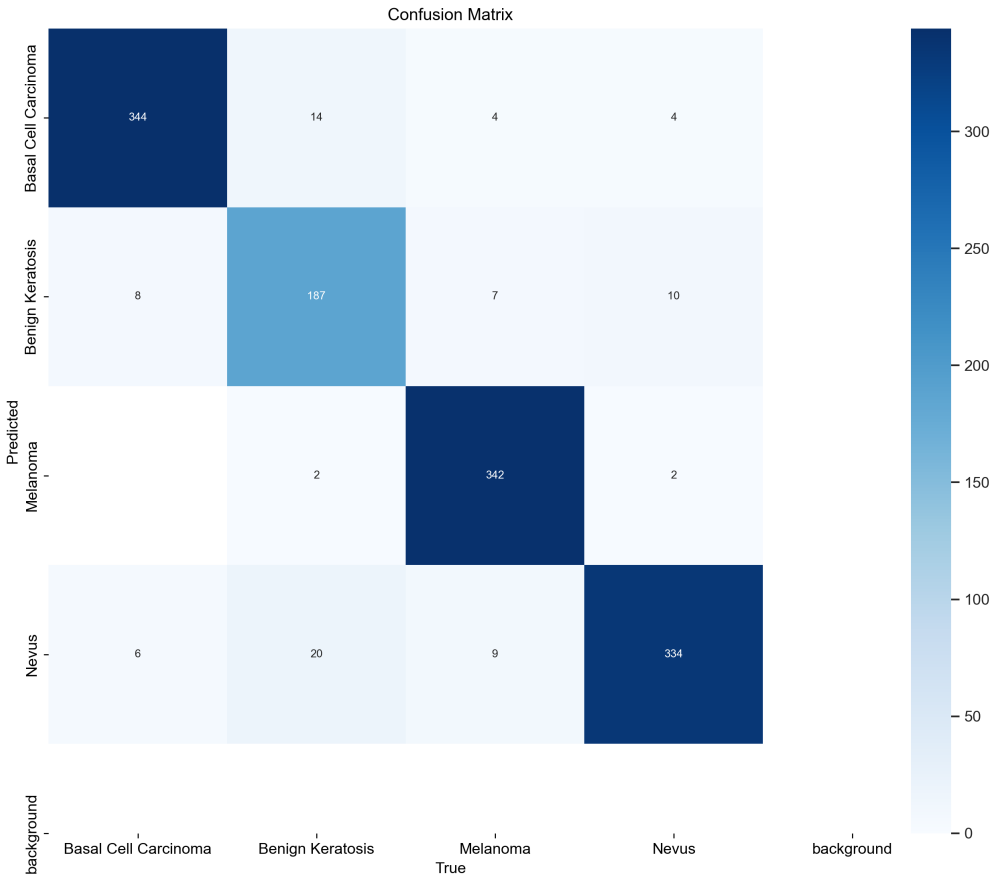

Figure 4.2: Confusion Matrix of YOLOv8 Classification Results

As shown in Figure **??**, the YOLOv8 model demonstrates strong classification performance across the four skin lesion categories: Basal Cell Carcinoma, Benign Keratosis, Melanoma, and Nevus. Most predictions fall along the diagonal, indicating accurate classifications. For Basal Cell Carcinoma, 344 out of 358 cases were correctly identified, yielding a sensitivity of approximately 96%. Melanoma, a high-risk class, was correctly predicted in 342 out of 362 cases, with only minimal confusion with other categories. Nevus was correctly classified in 334 of 350 cases, showing similar robustness. The most frequent misclassifications occurred with Benign Keratosis, where 187 out of 223 instances were correctly predicted, and the rest were confused primarily with Nevus. Despite this, overall per-class performance remains high, and the model effectively distinguishes between lesion types, especially those of high clinical relevance.

4.5.6 Macro and Weighted Averages

To further evaluate overall performance, both macro and weighted averages of the metrics were calculated. The **macro average** represents the unweighted mean of precision, recall, and F1-score across all classes, treating each class equally. In this case, macro-averaged

metrics all approximate **0.93**, indicating consistently strong performance regardless of class frequency. The **weighted average**, on the other hand, adjusts for class imbalance by weighting each class's contribution by its number of samples. The weighted precision, recall, and F1-score are also around **0.93**, suggesting that the model maintains reliable performance even when class distributions are uneven. The close agreement between macro and weighted scores reinforces the model's robustness across both frequent and rare classes, which is particularly important in medical applications where minority classes may have critical clinical implications.

4.5.7 Summary Table of Results

Table 4.3: Detailed classification metrics for each class on the test set.

Class	Precision	Recall	F1-score	Support
Basal Cell Carcinoma	0.94	0.96	0.95	358
Benign Keratosis	0.88	0.84	0.86	223
Melanoma	0.99	0.94	0.97	362
Nevus	0.91	0.95	0.93	350
Accuracy			**0.93**	1293
Macro avg	0.93	0.92	0.93	1293
Weighted avg	0.93	0.93	0.93	1293

The classification report confirms that the model performs well across all four classes, with precision and recall scores exceeding 0.84 for each. Melanoma, the most clinically significant class, achieved the highest F1-score of 0.97, reflecting exceptional precision and sensitivity. Basal Cell Carcinoma and Nevus were also classified with high reliability, reaching F1-scores of 0.95 and 0.93, respectively. Benign Keratosis, while slightly lower in recall, still maintained a respectable F1-score of 0.86. Together, the macro and weighted averages reinforce the model's balanced and robust performance, while the overall accuracy of 93.35% supports its suitability for deployment in real-world clinical settings.

4.5.8 Sample Predictions

To qualitatively evaluate the performance of the YOLOv8 classifier, we present sample predictions from the test set. Each image displays the true lesion class, the predicted class, and the associated confidence score, offering insight into the model's decision-making process and its level of certainty.

```
image 1/1 /teamspace/studios/this_studio/skin-1/test/Basal Cell Carcinoma
Speed: 8.0ms preprocess, 9.8ms inference, 0.1ms postprocess per image at
Image: ISIC_0024431_jpg.rf.5b00ead1de3b83905d9bea25a0b9ed43.jpg
  True Class:      Basal Cell Carcinoma
  Predicted Class: Nevus
  Confidence:      0.54
```

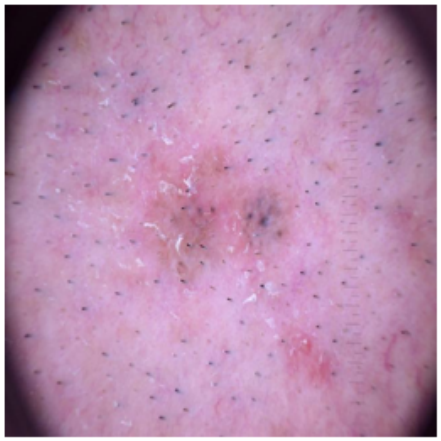

Figure 4.3: Sample prediction with a confidence score of 0.54. The model incorrectly classified the image, and the moderate confidence score reflects a degree of uncertainty. Such cases often arise when the lesion shares visual features with other classes, making it more challenging for the model to distinguish accurately.

```
image 1/1 /teamspace/studios/this_studio/skin-1/test/Melanoma
Speed: 9.1ms preprocess, 9.8ms inference, 0.1ms postprocess p
Image: ISIC_0065997_JPG.rf.b26b06138c9e14e49d1169734d00bf07.j
  True Class:      Melanoma
  Predicted Class: Melanoma
  Confidence:      1.00
```

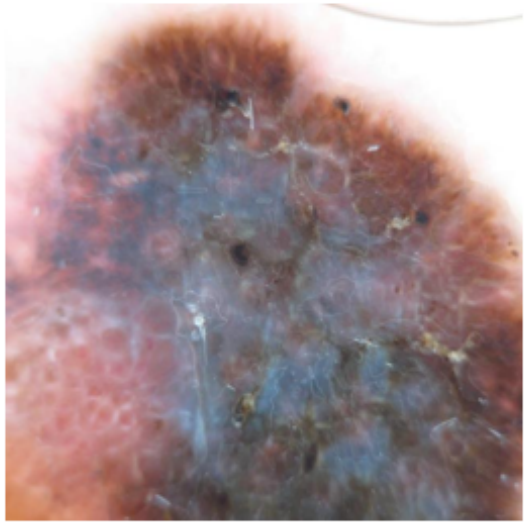

Figure 4.4: Sample prediction with a confidence score of 1.00. In this example, the model produced a highly confident prediction, indicating strong certainty in the extracted features and their correspondence with the predicted class.

4.5.9 Exported Model

The best-performing model was exported in `.pt` (PyTorch) format using the standard YOLOv8 export command. This export was a necessary step to enable integration of the trained model into GUI application, which is discussed in the following section.

Listing 4.5: Export command for YOLOv8 model to PyTorch format

```
!yolo export model={HOME}/runs/classify/train/weights/best.pt
    format=pt
```

4.6 Desktop GUI Application for Skin Lesion Classification

To facilitate practical and user-friendly deployment of the trained YOLOv8 classifier, we developed a standalone desktop application with a graphical user interface (GUI) using Python's Tkinter library. The GUI allows dermatologists and clinicians to easily upload dermoscopic images, run classification locally, and instantly view the predicted class, confidence score, and comparison with the true class label.

Frameworks and Tools Used:

- **Tkinter:** Python's standard library for building desktop GUIs, chosen for its simplicity and cross-platform compatibility.
- **Pillow:** For image loading and display within the GUI.
- **Ultralytics YOLO:** For loading the trained YOLOv8 model and performing inference.
- **ttk (Themed Tkinter):** For modern, styled widgets such as tables and buttons.

Motivation for an Offline Desktop GUI:

- **Data Privacy:** All image analysis is performed locally, ensuring patient data never leaves the user's computer and complying with medical privacy regulations.
- **Accessibility:** The application can be used in clinics or remote locations without internet access, making it suitable for resource-limited settings.

- **Ease of Use:** The GUI provides an intuitive workflow for clinicians, requiring no command-line interaction or programming knowledge.

How the GUI Works:

1. The user clicks the "Upload Image" button to select a dermoscopic image from their computer.
2. The selected image is displayed in the GUI.
3. The YOLOv8 model performs classification on the image, and the predicted class, confidence score, and true class (if available) are shown in a results table.
4. A message indicates whether the prediction matches the true class, providing immediate feedback.

Implementation Code:

Listing 4.6: Implementation of the Tkinter-based desktop GUI for skin lesion classification

```
import tkinter as tk
from tkinter import filedialog, ttk
from PIL import Image, ImageTk
from ultralytics import YOLO
import os
# Load your trained model
model = YOLO('best.pt')
#     result_label.config(text=result_text)
def open_image():
    # Open file dialog to select an image
    filepath = filedialog.askopenfilename(
        filetypes=[("Image␣files", "*.jpg␣*.jpeg␣*.png"), ("
           All␣files", "*.*")]
    )
    if not filepath:
        return
    # Run inference
    results = model(filepath)
    # Display original image without prediction overlays
    image = Image.open(filepath).convert("RGB")
    image.thumbnail((400, 400))  # Maintain original size
    img_tk = ImageTk.PhotoImage(image)
    image_label.config(image=img_tk)
    image_label.image = img_tk
    # Extract prediction
    probs = results[0].probs
    top1 = probs.top1
    confidence = probs.top1conf.item()
    class_name = results[0].names[top1]
    # Get true class from folder name (e.g., path/to/test/
       ClassX/image.jpg      ClassX)
    true_class = os.path.basename(os.path.dirname(filepath))
```

```
    # Update table with results
    for child in result_table.get_children():
        result_table.delete(child)
    result_table.insert("", "end", values=(true_class,
       class_name, f"{confidence:.2%}"))
    # Check if prediction is correct
    if true_class == class_name:
        result_message.config(text="Success: Prediction
           matches the true class!", fg="green")
    else:
        result_message.config(text="Error: Prediction does
           not match the true class.", fg="red")
# GUI setup
root = tk.Tk()
root.title("Cancer Detection GUI")
root.geometry("900x750")
root.config(bg="#FFF7E6")  # Creamy background
# Title Label
title_label = tk.Label(root, text="Skin Cancer Detection",
   font=("Helvetica", 28, "bold"), fg="white", bg="#0277BD")
    # Blue color
title_label.pack(pady=10, fill=tk.X)
# Upload Button
style = ttk.Style()
style.configure("Custom.TButton",
                font=("Helvetica", 14),
                padding=10,
                foreground="#F57C00",
                borderwidth=2,
                relief="raised")
style.map("Custom.TButton",
          bordercolor=[("!disabled", "#F57C00")])
upload_button = ttk.Button(root, text="Upload Image", command
   =open_image, style="Custom.TButton")
upload_button.pack(pady=20)
```

```
# Main Frame for Image and Results
main_frame = tk.Frame(root, bg="#FFF7E6")
main_frame.pack(pady=10, expand=True, fill=tk.BOTH)
# Left Frame for Image
left_frame = tk.Frame(main_frame, bg="#FFF7E6")
left_frame.pack(side=tk.LEFT, padx=10, pady=10)
image_label = tk.Label(left_frame, bg="#E3F2FD", width=400,
    height=400)  # Light blue background
image_label.pack()
# Right Frame for Results
right_frame = tk.Frame(main_frame, bg="#FFF7E6")
right_frame.pack(side=tk.RIGHT, padx=10, pady=10)
# Create a style for the Treeview
style = ttk.Style()
style.configure("Treeview.Heading", font=("Helvetica", 12, "
    bold"))  # Font for headings
style.configure("Treeview", font=("Helvetica", 10, "bold"))
    # Font for rows
# Result Table
columns = ("True␣Class", "Predicted␣Class", "Confidence")
result_table = ttk.Treeview(right_frame, columns=columns,
    show="headings", height=1)
result_table.heading("True␣Class", text="True␣Class")
result_table.heading("Predicted␣Class", text="Predicted␣Class
    ")
result_table.heading("Confidence", text="Confidence")
result_table.column("True␣Class", width=150, anchor="center")
result_table.column("Predicted␣Class", width=150, anchor="
    center")
result_table.column("Confidence", width=150, anchor="center")
result_table.pack(pady=20)
# Add Message Label
result_message = tk.Label(right_frame, text="", font=("
    Helvetica", 16), bg="#FFF7E6")
result_message.pack(pady=10)
```

```
# Footer
footer_label = tk.Label(root, text="Developed␣by:␣El␣Amraoui␣
   Khalil␣&␣Tariq␣Mohammed", font=("Helvetica", 12, 'bold', '
   italic'), bg="#FFF7E6", fg="#F57C00")
footer_label.pack(side=tk.BOTTOM, pady=10)
# Run the GUI
root.mainloop()
```

The full implementation is provided in the gui.py file. This approach ensures that the AI-powered skin lesion classifier is accessible, secure, and practical for real-world clinical use.

Sample Predictions in GUI:

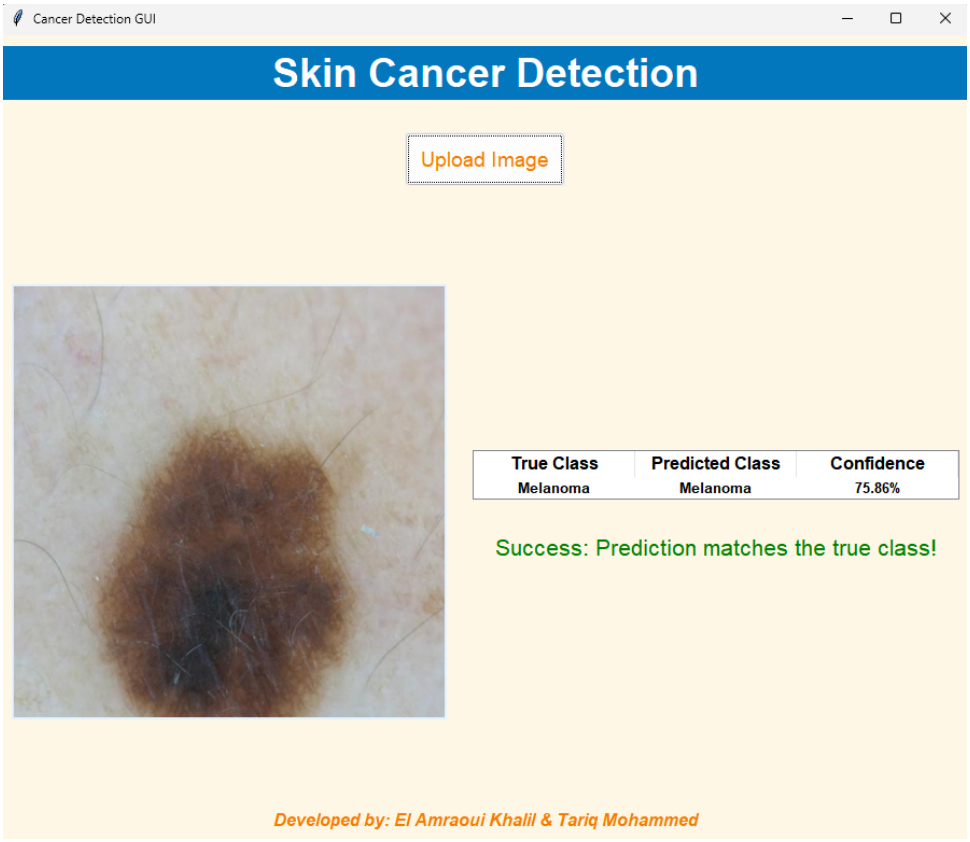

Figure 4.5: Sample predictions: True class and predicted class with confidence 0,75.

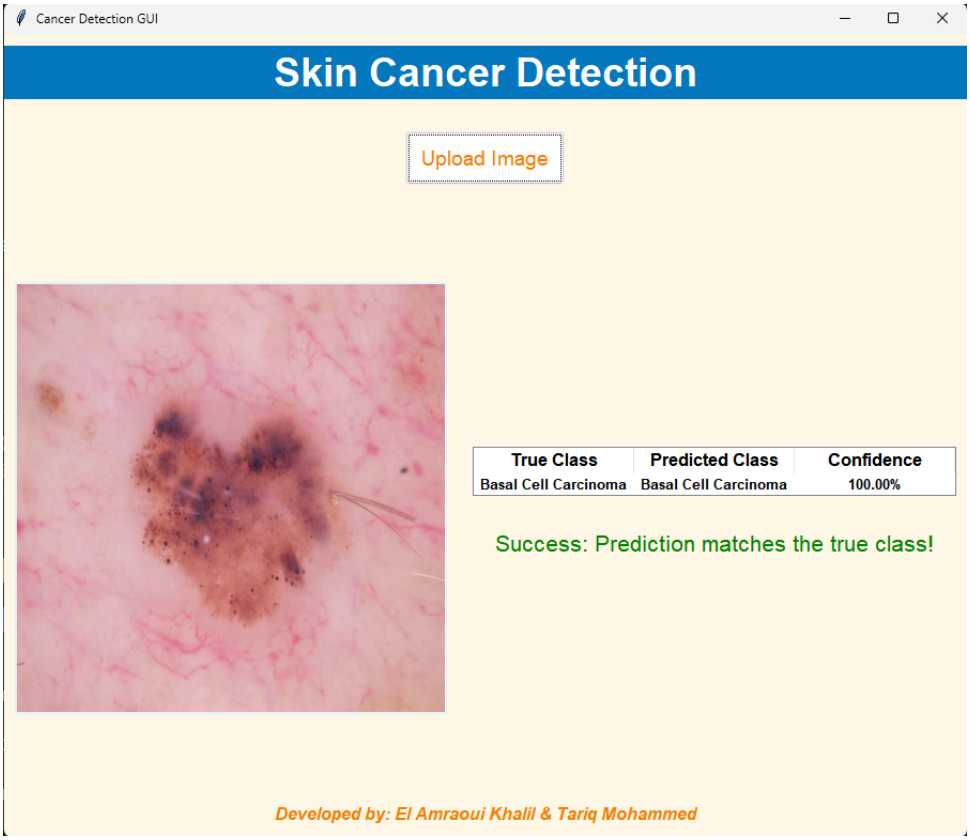

Figure 4.6: Sample predictions: True class and predicted class with confidence 1.

Chapter 5

Discussion

5.1 Strengths of the Approach

The application of YOLOv8 for skin lesion classification in dermoscopic images has demonstrated several notable strengths, both in terms of technical performance and practical deployment:

High Classification Accuracy

The YOLOv8-based classifier achieved a test accuracy of 93.35%, with an F1-score of 0.97 for melanoma. This high level of accuracy is particularly significant in the context of medical diagnostics, where false negatives can have severe consequences. The model's ability to distinguish between melanoma, basal cell carcinoma (BCC), benign keratosis (BK), and benign nevus demonstrates its robustness and generalizability across multiple lesion types.

Real-Time Inference and Efficiency

YOLOv8 is designed for real-time applications, and this project leveraged its efficient architecture to enable rapid inference. The model's lightweight design and optimized computation allow for deployment on standard desktop hardware without the need for specialized GPUs in the inference phase. This is crucial for clinical environments where quick decision-making is essential.

User-Friendly Deployment

The integration of the trained model into a standalone desktop application using Tkinter provides an intuitive graphical user interface (GUI) for dermatologists. This local deployment ensures that users can analyze images without internet connectivity, addressing privacy concerns and making the solution suitable for resource-limited settings.

Data Privacy and Compliance

By enabling offline analysis, the approach ensures compliance with medical data privacy regulations. Patient images are processed locally, eliminating the risk of data breaches associated with cloud-based solutions.

Transfer Learning and Pretrained Weights

Utilizing pretrained weights from YOLOv8m-cls.pt allowed for effective transfer learning, reducing the need for extensive labeled data and accelerating the training process. This is particularly beneficial in medical imaging, where annotated datasets are often limited.

Comprehensive Evaluation

The use of multiple evaluation metrics—accuracy, precision, recall, F1-score, and confusion matrix—provided a thorough assessment of model performance. Visualizations such as training curves and confusion matrix facilitated deeper insights into the learning dynamics and class-wise performance.

5.2 Challenges and Limitations

Despite the strengths, several challenges and limitations were encountered during the project:

Limited Dataset Size

The dataset, while based on the HAM10000 collection and expanded via Roboflow, still represents a limited sample of the diversity seen in real-world clinical practice. The model's generalizability to images from different devices, populations, or acquisition conditions may be constrained.

Augmentation Constraints

While standard augmentations (flips, rotations, brightness adjustments) were used, more advanced augmentation strategies (such as synthetic data generation or domain adaptation) could further enhance robustness. Mosaic augmentation, although available in YOLOv8, was not used due to its limited relevance for classification tasks.

Hyperparameter Optimization

The selection of hyperparameters (e.g., learning rate, batch size, optimizer) was based on standard practice. More extensive hyperparameter tuning or the use of automated optimization tools could potentially yield further improvements.

Interpretability and Explainability

Deep learning models, including YOLOv8, are often criticized for their "black box" nature. While the model provides high accuracy, understanding the specific features or regions influencing its decisions remains challenging. Incorporating explainability tools (e.g., Grad-CAM, SHAP) could enhance trust and adoption in clinical settings.

Hardware and Training Constraints

Although training was performed on a T4 GPU via Lightning AI, resource limitations (such as GPU memory and training time) restricted the exploration of larger models or

longer training schedules. In real-world deployments, hardware constraints may also limit the use of more complex architectures.

Potential for Overfitting

Given the limited dataset size and high model capacity, there is a risk of overfitting. Careful monitoring of validation metrics was employed, but further validation on external datasets is recommended.

5.3 Insights and Lessons Learned

Importance of Data Quality and Diversity

The quality and diversity of the training data are paramount in achieving robust model performance. Efforts to curate balanced, high-quality datasets directly impact the classifier's ability to generalize.

Transfer Learning Accelerates Development

Leveraging pretrained models significantly reduces the time and data required to achieve high accuracy. Transfer learning is especially valuable in medical imaging, where labeled data is scarce.

Evaluation Beyond Accuracy

Relying solely on accuracy can be misleading, especially in imbalanced datasets. The use of precision, recall, F1-score, and confusion matrices provided a more nuanced understanding of model strengths and weaknesses.

User-Centric Design Enhances Adoption

Developing a user-friendly desktop application with an intuitive GUI increases the likelihood of adoption by clinicians. Offline capability and data privacy are critical features for medical applications.

Continuous Validation is Essential

Ongoing validation with new and diverse datasets is necessary to ensure sustained model performance and to detect potential biases or failure modes.

Future Directions

Future work should focus on expanding the dataset, incorporating advanced augmentation techniques, and validating the model in real-world clinical settings. Integration of

lesion segmentation and multi-modal data (e.g., patient history) could further enhance diagnostic accuracy.

Collaboration with Domain Experts

Close collaboration with dermatologists and medical professionals is essential for dataset annotation, model validation, and interpretation of results. Their expertise ensures that the AI system addresses clinically relevant challenges and meets real-world needs.

Scalability and Deployment

The approach demonstrated here is scalable and can be adapted to other medical imaging tasks. The modular design of YOLOv8 and the flexibility of the deployment pipeline facilitate adaptation to new datasets and clinical requirements.

Conclusion of Discussion

In summary, the YOLOv8-based classification system for skin lesions has proven to be a powerful and practical tool, combining high accuracy with efficient deployment. While challenges remain, particularly regarding data diversity and model interpretability, the approach lays a strong foundation for future advancements in AI-assisted dermatology and medical image analysis.

Chapter 6

Conclusion

Recap of Project Objectives and Achievements

This project set out to develop an accurate and efficient classification system for skin lesions in dermoscopic images using deep learning, specifically leveraging the YOLOv8 framework. The main objectives included designing a robust multi-class classifier, optimizing model performance, and providing a practical deployment solution for clinical use.

Through careful dataset preparation, model selection, and training with advanced augmentation techniques, the YOLOv8-based classifier achieved a high test accuracy of 93.35% and an F1-score of 0.97 for melanoma. The integration of the trained model into a standalone desktop GUI application ensures accessibility, data privacy, and ease of use for clinicians, even in offline or resource-limited environments.

Importance of the Results in the Context of Healthcare

The results of this work demonstrate the potential of modern deep learning models to significantly enhance the accuracy and efficiency of skin cancer diagnosis. Early and reliable detection of melanoma and other skin cancers is critical for improving patient outcomes and survival rates. By providing a fast, accurate, and user-friendly tool for dermatologists, this project contributes to reducing diagnostic errors, supporting clinical decision-making, and expanding access to advanced diagnostic technology.

Furthermore, the offline capability and privacy-preserving design of the desktop application address key concerns in medical data handling, making the solution suitable

for real-world deployment. The methodologies and insights gained from this project can be extended to other medical imaging tasks, paving the way for broader adoption of AI-assisted diagnostics in healthcare.

Future Perspectives

Future work may focus on expanding the dataset, incorporating additional lesion types, enhancing the GUI for clinical workflows, and integrating explainability features to further build trust among medical professionals. Continuous validation and collaboration with healthcare experts will be essential to ensure the system remains reliable, interpretable, and impactful in diverse clinical settings.

Bibliography

[Esteva17] *Esteva, A., Kuprel, B., Novoa, R. A., et al. (2017). Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. Nature.*

[Pham21] *Pham, T. C., Luong, C. M., Hoang, V. D., et al. (2021). AI Outperformed Every Dermatologist in Dermoscopic Melanoma Diagnosis Using an Optimized Deep-CNN Architecture. Scientific Reports.*

[Liu24] *Liu, H., et al. (2024). Analysis of Artificial Intelligence-Based Approaches Applied to Non-Invasive Skin Imaging Modalities. Journal of Medical AI Research.*

[Tschandl18] *Tschandl, P. et al. (2018). The HAM10000 Dataset. Scientific Data.*

[Haenssle18] *Haenssle, H. A. et al. (2018). Man Against Machine. Journal of the European Academy of Dermatology and Venereology.*

[LeCun15] *LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.*

[Esteva17b] *Esteva, A., Kuprel, B., Novoa, R. A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.*

[Howard20] *Howard, J., and Gugger, S. (2020). Deep Learning for Coders with fastai and PyTorch. O'Reilly Media.*

[ISIC] *ISIC Archive: International Skin Imaging Collaboration. Retrieved from https://www.isic-archive.com*

[Goodfellow16] *Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.*

[LeCun15] *LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436–444.*

[Ronneberger15] *Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI.*

[Ronneberger15b] *Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI.*

[Esteva17b] *Esteva, A., et al. (2017). Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. Nature, 542(7639), 115–118.*

[He16] *He, K., et al. (2016). Deep Residual Learning for Image Recognition. CVPR.*

[Litjens17] *Litjens, G., et al. (2017). A Survey on Deep Learning in Medical Image Analysis. Medical Image Analysis, 42, 60–88.*

[Gulshan16] *Gulshan, V., et al. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 316(22), 2402–2410.*

[Jumper21] *Jumper, J., et al. (2021). Highly Accurate Protein Structure Prediction with AlphaFold. Nature, 596(7873), 583–589.*

[Bera19] *Bera, K., et al. (2019). Artificial Intelligence in Digital Pathology. Nature Reviews Cancer, 19(12), 703–715.*

[Topol19] *Topol, E. J. (2019). High-Performance Medicine: The Convergence of Human and Artificial Intelligence. Nature Medicine, 25(1), 44–56.*

[ACS23] *American Cancer Society. (2023). Skin Cancer Facts and Statistics.*

[Rogers15] *Rogers, H. W., et al. (2015). Incidence Estimate of Nonmelanoma Skin Cancer in the U.S., 2006. Archives of Dermatology, 146(3), 283–287.*

[Tschandl18b] *Tschandl, P., et al. (2018). The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. Scientific Data, 5, 180161.*

[Marchetti18] *Marchetti, M. A., et al. (2018). Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging Challenge:*

Comparison of the Accuracy of Computer Algorithms to Dermatologists for the Diagnosis of Melanoma from Dermoscopic Images. Journal of the American Academy of Dermatology, 78(2), 270–277.

[Lafraxo22] *Lafraxo, S., El Ansari, M., & Charfi, S. (2022). MelaNet: An Effective Deep Learning Framework for Melanoma Detection. Multimedia Tools and Applications.* https://www.researchgate.net/publication/358952296_MelaNet_an_effective_deep_learning_framework_for_melanoma_detection_using_dermoscopic_images

[Alenezi23] *Alenezi, F., et al. (2023). Deep Learning Approach for Skin Cancer Detection. Diagnostics, 13(19), 3147.* https://www.mdpi.com/2075-4418/13/19/3147

[Ultralytics23] *Ultralytics. (2023). YOLOv8: Cutting-Edge Object Detection Models.* https://docs.ultralytics.com/models/yolov8.

[Ultralytics24] *Ultralytics. (2024). Image Classification.* https://docs.ultralytics.com/tasks/classify/.

[Roboflow24] *Roboflow. (2024). Skin Lesion Classification (HAM10000-based) Dataset.* https://universe.roboflow.com/bigdata-xv5t7/skin-9nkfv.