Dermatology AI: Identifying Skin Injuries and Determining Burn Degrees using Deep Learning

Abdullaziz Alosayl [411109523],
Mohammed Alrushud[411107962],
Anas Shawky Alarbied [411116054],
Abdullah Huthlul Alhuthlul [411107932]

A project report submitted to Qassim University in partial fulfilment
of the requirements for the degree
of BSc in Computer Science
at the Department of Computer Science
College of Computer
Qassim University
Saudi Arabia

May 2024

Supervisor Dr. Mohammed Alsuhaibani

Acknowledgements

First and foremost, we praise Allah Almighty for His guidance. Secondly, we express our gratitude to everyone who contributed to the success and completion of this project. Special thanks go to our project supervisor, Dr.Mohammed Alsuhaibani, for his support and guidance.

Abdulaziz, Anas, Mohammed, Abdullah

Dedication

We are bachelor's students at the Department of Computer Science, College of Computer at Qassim University. We would like to dedicate this project to College of Computer at Qassim University's, which provided unwavering support throughout the different phases of this project.

Abdulaziz, Anas, Mohammed, Abdullah

Abstract

This project introduces an AI-based model leveraging deep learning for the diagnosis and severity assessment of skin burns. Globally, millions suffer from burns annually, with current diagnostic methods relying heavily on subjective visual examinations by healthcare professionals. This conventional approach often leads to inconsistent treatment recommendations and patient care. Our model aims to address this process by using algorithms like YOLO for accurate burn analysis in images, categorizing burns into first-degree, second-degree, and third-degree. The model will be trained on a diverse dataset of burn images, ensuring precision and reliability across various demographics and burn stages. This approach promises to improve diagnostic accuracy by reducing (not eliminating) reliance on human expertise and mitigating bias.

List of Figures

Contents

Chapter 1

Introduction

1.1 Introduction

Skin burns, resulting from exposure to extreme temperatures, are among the most common and painful of all injury types. Globally, millions suffer from burn injuries annually, with a significant proportion experiencing long-term physical and psychological consequences. Timely and accurate diagnosis of burn severity is not just pivotal but is often the linchpin in ensuring optimal therapeutic outcomes [4].

However, a notable challenge in the field of burn treatment is the diagnosis of burn severity. Traditionally, healthcare professionals have depended on visual examinations to measure burn severity. This approach involves judging the burn's appearance to estimate its depth and extent. Despite its widespread use, this method has limitations, it's subjective, varies based on the examiner's experience, and lacks a standardized assessment protocol. Consequently, this can lead to divergent treatment recommendations and inconsistency in patient care, sometimes resulting in sub-optimal outcomes or complications [5].

Therefore, the imperative for a more objective, reliable, and scalable solution becomes clear. This project thus proposes a transformative approach: leveraging Artificial Intelligence (AI) and Machine Learning (ML) in particular for burn analysis and detection. Central to our proposal is the adaptation of deep learning techniques, such as YOLO [6] algorithm designed for object detection. By utilising such approaches, our project aims to discern burns in visual media, be it images for example, and provide a reliable severity assessment.

1.2 Project Scope

The primary objective of this project is to develop an AI-based model that specializes in distinguishing between first-degree, second-degree, and third-degree burns [7]. To ensure the model's precision and reliability, we will train the model on a diverse dataset, encompassing diverse burn images from various patient demographics, burn sources, and stages of injury. This dataset will serve as the foundation upon which our ML model will be trained, enabling it to understand the visual characteristics unique to each burn degree.

Once trained, as usual with ML-powered models, the model will be adept at processing and categorizing the severity of burns in new, unseen images, positioning it as a valuable adjunct to healthcare professionals in their assessment and diagnosis of burn injuries.

1.3 Motivation

ML-based systems have demonstrated the ability to improve the accuracy and expediency of burn detection, thereby ensuring the timely provision of essential care to patients [8, 9]. These systems alleviate the dependence on human expertise, particularly in remote or underserved regions, effectively addressing healthcare disparities. With their objectivity and consistency, ML-based systems offer assessments that mitigate the inherent risks of bias and diagnostic errors. Furthermore, these systems hold promise for early detection, facilitating timely interventions and fostering improved patient outcomes.

Our main drive to develop a ML-based model for burn detection comes from a few key reasons. Firstly, we want to offer a faster way to check burns. Secondly, this tool can help doctors give advice over the phone or online, especially in places where getting to a hospital is hard. Lastly, it is also a useful aid for those who might find it hard to see burns clearly or can't easily move around. Overall, by using ML for burn care, we believe both patients and healthcare workers can benefit greatly.

1.4 Aim and Objectives

1.4.1 Aim

Our primary goal is to develop an AI system that utilizes deep learning methodologies to detect and categorize burn injuries. While there are various algorithms available, we are initially exploring the capabilities of the You Only Look Once (YOLO) [10] model as an exemplary object detection model. This exploration will guide us in determining the most optimal approach for our specific needs in burn identification from images, ensuring both efficiency and precision.

1.4.2 Objectives

Data Collection

The objective of data collection is to gather a large and diverse dataset of burn-related images and videos. This involves conducting thorough research and sourcing data from various reliable and relevant sources. The collected data should include examples of different types of burns, such as first-degree, second-degree, and third-degree burns.

Model Development

The objective of model development is to design and develop the architecture of the burn detection model. This includes selecting an appropriate deep-learning framework and defining the network architecture. The model should be capable of accurately detecting and classifying burns in images. The architecture may involve using convolutional neural networks (CNNs) [11].

Model Training and Evaluation

Once the model architecture is defined, the objective is to train the model using the collected dataset. This involves feeding the dataset into the model and optimizing its parameters to improve its performance. The training process aims to enable the model to learn patterns and features that distinguish different types of burns. After training, the model needs to be evaluated using appropriate evaluation metrics, such as accuracy, precision, recall, and score.

User Interface

Ensuring our model is user-friendly is essential. We aim to design an intuitive web-based interface that integrates with our model. This approach aims to make the tool accessible and effective for both medical professionals and the wider public, facilitating rapid and accurate burn detection.

1.5 Project Plan and Schedule

Our plan for the project started with deciding on the appropriate idea for our project. Next, we ensured the idea's feasibility by searching for the appropriate datasets and tools. After that, we began writing an introduction and a literature review for the

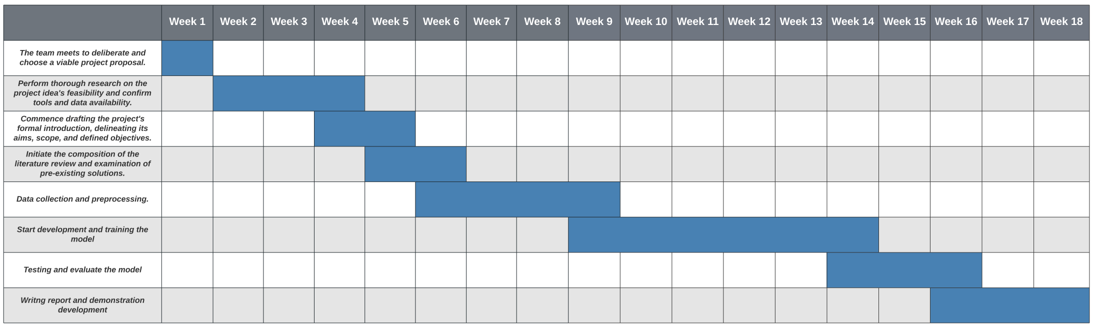

Figure 1.1: Time line

project. Following this, we initiated the work by collecting the datasets and commencing the development and testing of the model. Finally, we will compile a report summarizing our work. The detailed timeline of our project is illustrated in Figure 1.1.

1.6 Outline of The Project

After this introductory chapter, we will write three more chapters in this report as follows:

- **Chapter 2: Literature Review**. It will contain an introduction to this chapter, background information on the project's components, a section on existing related systems, and a concluding section summarizing the chapter.
- **Chapter 3: Problem Analysis**. This chapter will also include an introduction. We will proceed to specify a problem and provide a solution. Following that, we will analyze the system by collecting, analyzing, and organizing the requirements. Afterward, we will implement and evaluate a plan for the system, selecting technical tools and languages that are appropriate. Finally, we will provide a summary of the chapter.
- **Chapter 4: System Design**. Similar to the previous two chapters, it will begin with an introduction. It will then include the System Design Specification, followed by the drawing of the system architecture. Subsequently, it will feature pseudocode, algorithms, or scenarios of the model. Finally, it will conclude with a summary.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides an overview of existing research on automated skin burn classification. It focuses on studies that have explored machine learning and image analysis techniques for classification. The aim is to improve the efficiency and effectiveness of burn classification, ultimately enhancing patient care and outcomes. The review identifies gaps in current research and highlights areas for further investigation in this field.

2.2 Background

2.2.1 Skin Burns

Burns can be categorized into three degrees based on their severity [7]. First-degree burns are superficial and only affect the outermost layer of the skin, causing redness, pain, and mild swelling. These burns heal within a week without leaving scars. Second-degree burns extend beyond the epidermis and affect the underlying dermis. They are characterized by blistering, intense pain, redness, and swelling. Healing for second-degree burns may take several weeks and can result in scarring. Third-degree burns are the most severe, penetrating through all skin layers and affecting underlying tissues. Immediate medical attention is required for third-degree burns, often involving skin grafting for proper healing.

Accurate assessment of burn severity is crucial for determining appropriate treatment. AI-based solutions can assist in this assessment by providing precise and consistent evaluations based on factors such as burn depth and extent.

2.2.2 AI and Machine Learning in Skin Burns

Approaches to burn injury evaluation rely on healthcare professionals visually examining the affected skin to estimate the severity and extent of burns based on their visual characteristics. These methods do have some shortcomings, though, like a reliance on personal knowledge and inconsistent treatment suggestions. The use of ML AI techniques for analyzing and diagnosing skin burns is on the rise as a solution to these problems. These solutions' ability to provide accurate and consistent assessments of burn severity using AI and ML will improve diagnostic procedures, which will ultimately result in better patient outcomes.[8]

2.2.3 Image Classification

The process of image classification plays a vital role in diverse areas such as stock market prediction, weather forecasting, and medical diagnosis, presenting a complex challenge [12]. Among various techniques for image classification, Convolutional Neural Networks (CNNs) have demonstrated superior performance over traditional neural networks in computer vision tasks, primarily due to their ability to capture spatial dependencies in images. The architecture of CNNs includes multiple layers, like convolutional, pooling, and fully connected layers, which collectively enhance feature extraction and classification — key to accurate image analysis Figure 2.1. A notable design feature of CNNs is weight sharing, which considerably reduces the trainable parameters in the network, thereby improving generalization and preventing overfitting. This aspect makes CNNs particularly effective for image classification, a process involving multiple factors and techniques to ensure precision [1]. Challenges in image classification stem from both data characteristics and model capabilities. Methods like supervised and unsupervised learning are employed, with Support Vector Machine (SVM) being a prevalent model [13]. However, the advent of CNNs has been a game changer, as they have been successfully applied in image classification tasks, showing a significant improvement over traditional methods.

2.3 Existing Related Systems

Advancements in image classification technology have significantly impacted the medical field, offering tools for fast and accurate diagnosis of various conditions. These tools aim to assist radiologists, physicians, and researchers in disease detection from a variety of imaging modalities, including X-rays, MRIs, and CT scans, ultimately enhancing patient care.

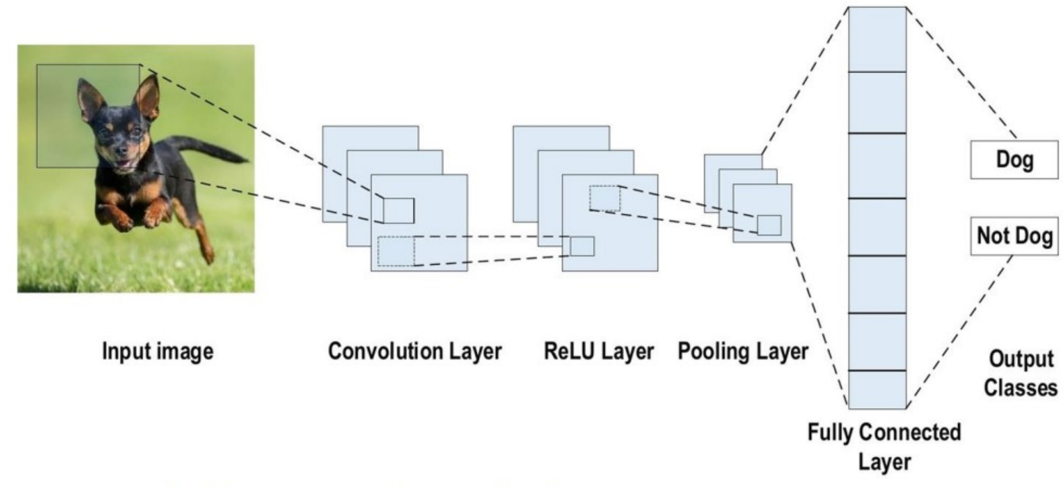

Figure 2.1: An example of CNN architecture for image classification[1]

Mateen et al. [14] demonstrated the application of image classification in diagnosing diabetic retinopathy from fundus images. Their approach integrated a Gaussian mixture model for precise region segmentation, VGGNet for high-dimensional feature extraction [15], and a combination of Principal Component Analysis and Singular Value Decomposition for efficient feature selection. Employing a dataset of over 35,000 fundus images, they showcased the superior performance of the VGG-19 Deep Neural Network compared to other models like AlexNet [16] and SIFT [17], along with a reduction in computational time.

In the domain of COVID-19 detection, Ozturk et al. [18] presented an automated approach utilizing deep neural networks, introducing the DarkCovidNet model inspired by the Darknet-19 architecture [19]. With the use of publicly available datasets, their model exhibited remarkable performance metrics, showcasing its potential as a tool for aiding healthcare professionals in efficient COVID-19 diagnosis.

Aishwarya et al. [20] focuses on developing an automated system for diagnosing skin cancer using the Yolo deep neural network. The study utilizes a dataset of skin cancer lesion images and trains the Yolo model to classify and detect different types of skin cancer. The performance of YoloV3 and YoloV4 models is evaluated, showing high classification scores and outperforming existing methods. Similarly, ALKolifi ALEnezi [21] build system employs image processing and machine learning to detect, extract, and classify skin diseases. It follows a process involving preprocessing to enhance image quality, feature extraction through resizing and CNN-based feature extraction, and classification using a multiclass SVM to identify various skin diseases. The system effectively identifies Eczema, Melanoma, and Psoriasis with 100% accuracy using a dataset of 100 images (80 for training and 20 for validation).

Rostami et al. [22] introduces an ensemble Deep Convolutional Neural Network (DCNN) classifier, merging patch-based classification with AlexNet, for image-wise classification of diverse wound types, addressing the challenges of manual wound classification with advanced Artificial Intelligence techniques. The study employs an ensemble (DCNN) classifier, combining patch-wise and image-wise classifications through Multilayer Perceptron (MLP), utilizing various deep architectures with transfer learning, and object localization algorithms, achieving superior multiclass wound image classification performance compared to methods discussed in the literature. The proposed ensemble DCNN-based classifier demonstrated high accuracy, achieving maximum values of 96.4% for binary and 91.9% for 3-class wound image classification, outperforming other deep classifiers.

In the specific domain of skin burn detection, various approaches have been explored to enhance classification and detection accuracy. E.B. et al. [23] conducts a comparative study of different segmentation algorithms in the classification of human skin burn depth. Eight hybrid segmentation algorithms are studied on a dataset of burn images categorized into three burn classes. The performance of the algorithms is evaluated by calculating the number of correctly segmented images for each burn depth. The study highlights the importance of using a good segmentation algorithm for accurate classification of skin burn depth.

Moreover, Rangel-Olvera and Rosas-Romero [7] used sparse representation with over-redundant dictionaries to detect and classify burnt areas in color images,two methods were used to build dictionaries for burn severity classes: direct collection of feature vectors from patches in various images and locations, and collection of feature vectors followed by dictionary learning accompanied by K-singular value decomposition.The image dataset included 33 first-degree burns, 26 second-degree burns, and 56 third-degree burns, with a diversity of background artifacts,they achieved 95.65 percent sensitivity and 94.02 percent precision. Color and texture were important features, and a shadowed skin dictionary reduced false positives.

Suha and Sanam [24] proposes a Deep Convolutional Neural Network (DCNN) approach for real-time detection and categorization of burn severity, outperforming traditional methods by incorporating transfer learning, fine-tuning, and multiple convolutional layers for feature extraction and classification. The study compares (DCNN) by its effectiveness with traditional methods incorporating digital image processing and machine learning classifiers. And reveals that (DCNN) model's superior accuracy of 95.63% in detecting burn severity, outperforming traditional classifiers. Augmented data improved performance across models, underscoring the significance of dataset size, demonstrating the potential of intelligent technologies in burn assessment and treatment.

Furthermore, Abubakar et al. [25] addresses the challenges of accurate burn assessment by proposing machine learning techniques, specifically utilizing transfer learning, to discriminate burns from injured skin, aiming to overcome the limitations of human expertise and high assessment costs. The paper employs transfer learning with pre-trained deep learning models, ResNet50, ResNet101, and ResNet152, utilizing both fine-tuning and off-the-shelf features approaches to classify skin burn images and other injuries, enhancing discrimination accuracy. The study's approach, employing transfer learning with fine-tuned ResNet50, ResNet101, and ResNet152 models using Google Collab's GPU hardware, achieved exceptional accuracy of approximately 99.9% in discriminating burnt skin and injured skin, utilizing 200 training epochs and ensuring diverse data representation patterns.

On the other side, transitioning from specific applications to a broader perspective, the interplay between human expertise and machine learning algorithms becomes a focal point. For example, Haenssle et al. [26] showcased the prowess of a modified Google Inception v4 CNN in dermatological image analysis, outperforming dermatologists in both specificity and overall diagnostic accuracy. Dermatologists evaluated images at two levels: Level-I and Level-II. In Level-I, dermatologists correctly identified 86.6% of melanomas and 71.3% of benign lesions. In Level-II, their accuracy improved to 88.9% for melanomas and 75.7percent for benign lesions. The CNN outperformed dermatologists, with higher specificity (82.5%) in both levels and a greater area under the ROC curve (0.86 vs. 0.79). This interplay is further explored by Rao et al. [27], when they present the Global Filter Network GFNet, a computational efficient alternative to self-attention and MLP models, utilizing a 2D discrete Fourier transform by replacing self-attention layers in transformer architectures with a 2D discrete Fourier transform, element-wise multiplication with global filters, and inverse Fourier transform, evaluated on ImageNet and various datasets, demonstrating superior accuracy/complexity trade-offs, robustness, and generalization ability. The paper showcases GFNet's competitive edge in image classification by outperforming various architectures on ImageNet and demonstrating superior adversarial robustness and generalization abilities across multiple datasets, positioning it as a compelling alternative to transformer-style models and CNNs.

Corroborating the transition towards deep learning, Chaganti et al. [28] explores image classification, beginning with traditional machine learning like Support Vector Machines (SVM) despite the prevalence of Neural Networks (NN). The study initially utilized Support Vector Machines (SVM) for image classification, achieving 93% accuracy on a small dataset which reduced to 82% with data augmentation. Transitioning to Convolutional Neural Networks (CNN) resulted in an impressive 93.57% accuracy on the same dataset, emphasizing the superior potential of deep learning methods, facilitated by various data augmentation techniques.

Similarly, Rajpurkar et al. [29] introduce CheXNet which is a robust 121-layer convolutional neural network (CNN) based on the DenseNet[30] architecture, with the aim of identifying pneumonia from chest X-rays and surpassing the diagnostic capabilities of practicing radiologists. They used the ChestX-ray14 dataset which includes over 100,000 annotated X-ray images, CheXNet also pinpoints pertinent areas most indicative of pneumonia within the image. When it was evaluated against the annotations of four radiologists that had 4, 7, 25, and 28 years of experience, CheXNet achieved an impressive F1 score of 0.435, and it outperformed the average of the radiologist's performance.

Building on the momentum of deep learning applications, Abdelhalim et al. [31] focuses on data augmentation techniques to enhance skin disease detection using images. It proposes a method called SPGGAN for data augmentation in dermoscopy images. The SPGGAN generates synthetic skin lesion images to augment the training dataset, improving the performance of automated skin lesion detection systems. Experimental results show the effectiveness of the proposed method compared to traditional data augmentation approaches.

2.4 Contribution

Building upon existing research, which were thoroughly discussed in the above section, our project aims to bridge identified gaps in burn severity classification. Unlike prior studies that focused on specific aspects of skin disease classification, we plan to create a comprehensive solution by compiling a vast and diverse dataset, and conducting extensive comparisons of various machine learning and CNN-based models. Our goal is to enhance burn degree classifications and develop a user-friendly web-based platform, addressing the need for accessible and efficient diagnostic tools, and ultimately contributing to improved patient care and medical diagnostics.

2.5 Summary

In this chapter, we have given a background on the limitations of conventional burn assessment techniques and the potential benefits of AI and ML-based solutions. We also briefly explain the various degrees of burn severity and underline the importance of accurate assessment for an effective treatment. We then give a few examples of some existing work done in image classification in the medical field, in skin burn injuries and a comparison between machine learning techniques and some dermatologist experts.

Chapter 3

Problem Analysis

3.1 Introduction

This chapter starts by explaining the problem itself and suggesting a solution. We will also talk about other possible solutions and explain why our approach is beneficial. We will discuss the skills and expertise needed for this solution to work. Then specify how we are going to collect and analyze the requirements for our project and organize them in a clear way. We will also talk about our plan for making this solution and how we evaluate its success. Lastly, we explain the technical tools and programming languages that are going to be used and why we chose them.

3.2 Problem Specification

The automation of tasks using machine learning and AI has always been a significant challenge, resulting in varying outcomes influenced by many factors. In our case, we are focusing on classifying the severity of skin burns through image analysis. This involves compiling a comprehensive and diverse dataset and comparing various modern machine learning and CNN-based models. We expect to witness a notable improvement in the precision and accuracy of our skin burn classification system.

Also, we intend to develop a web-based platform to aid in the enhancement of patient care and medical diagnostics tools. Some required skills to effectively execute this are an understanding of CNN-Based models and their applications in image analysis, efficiency in data processing, model training, and the necessary skills for creating a web-based platform, which will be important for making this project.

3.3 System Analysis

3.3.1 Requirement Collection

To comprehensively gather requirements for our skin burn detection system, we conducted an extensive literature review of existing research in machine learning and deep learning applied to medical image classification. This review helped us identify significant gaps in current methodologies, specifically a need for enhanced speed, accuracy, user-friendliness, and stringent privacy standards. These insights have been instrumental in shaping our system's design and development, ensuring it not only meets technical classification needs but also addresses practical concerns of medical professionals and patients, especially regarding quick, precise, and confidential handling of sensitive medical data.

3.3.2 Requirement Analysis

The primary requirements for our system include:

- Developing a machine learning model aimed at accurate burn severity classification into distinct categories (1st, 2nd and 3rd degree).
- Creating a comprehensive and diverse dataset of skin burn images, accurately labeled for severity.
- Designing a user-friendly web interface for easy access and interaction with the classification model.
- Ensuring our classification performance meets or exceeds current benchmarks in skin burn severity classification.
- Integrating the system smoothly into existing clinical workflows to augment medical diagnostics.

3.3.3 Requirement Organization

Our system's requirements fall into two categories:

1. **Functional Requirements:**
   (a) Dataset Compilation (high priority): Compile a diverse dataset using machine learning techniques.
   (b) Data Preprocessing (high priority): Preprocess data using deep learning before training the proposed model.

(c) Model Comparison (high priority): Compare different machine learning and CNN-based models for burn classification.

(d) Training for YOLO Model (high priority): Utilize deep learning for YOLO model training, leveraging our dataset.

(e) Inference (high priority): Implement deep learning for real-time burn detection and classification.

(f) Evaluation of Accuracy (medium priority): Utilize various metrics for assessing burn detection precision.

(g) Optimization (medium priority): Use advanced ML techniques to improve YOLO model accuracy and performance.

(h) User Interface (low priority): Develop an intuitive interface for user interaction during training and inference.

2. **Non-functional Requirements:**

   (a) Performance and Efficiency (high priority): Focus on efficient data processing for handling diverse datasets.

   (b) Security, Privacy, and Compliance (high priority): Implement robust security and comply with privacy regulations.

   (c) Usability and Accessibility (medium priority): Ensure the web platform is user-friendly and accessible.

3.4 Implementation and Evaluation Plan

Firstly, we will collect a diverse set of images and data related to skin burn degrees. This dataset will include examples of different burn degrees and types of skin injuries. We will then proceed to label the data accurately, identifying the regions suspected of injuries and the corresponding burn degrees.

Next, we will split the dataset into training, testing, and evaluation sets. This division will allow us to assess the performance of the system accurately. We will select an appropriate deep learning YOLO [6] model and configure it to train on the available data. The model will undergo extensive training, during which we will adjust its parameters and optimize it for optimal performance.

Once the model is trained, we will evaluate its performance using the testing dataset. We will measure metrics such as detection accuracy and classification accuracy to assess the system's effectiveness in identifying skin injuries and determining burn degrees.

Moving on to the evaluation phase, we will conduct a final performance evaluation of the trained model. This evaluation will involve assessing its performance on the testing and evaluation datasets. We will compare the model's results with those of healthcare professionals to validate its accuracy.

If necessary, we will refine and adjust the model based on the evaluation results. This iterative process will involve modifying the model and repeating the training and evaluation steps until we achieve satisfactory performance.

3.5 Technical Tools and Programming Languages

In this section, we will list the main programming languages and tools we will use in our system:

- **Python programming language:** We will use python programming language because it is widely used in machine learning for its simplicity and readability, facilitating efficient development and collaboration. Its extensive ecosystem, including libraries like TensorFlow and PyTorch, provides powerful tools for data manipulation, model training, and deployment.

- **Pytorch framework:** We will use PyTorch [32] for its dynamic computational graph and intuitive, Pythonic syntax, enabling flexible and seamless development of deep learning models, especially in research and experimentation contexts. Its active community and growing ecosystem further contribute to its popularity for academic and industrial applications.

- **YOLO algorithm:** We will use the modern object detection algorithm YOLO [6] for real-time object detection, as it efficiently processes images in a single pass, providing high accuracy with precise object localization.

- **Django:** We chose Django [33] for our API development due to its ability to handle high web traffic and accommodate user growth, as well as its robust security features. Django's ORM provides a simplified approach to database interactions, and its active community and comprehensive documentation make it an ideal choice for building scalable web applications.

In summary, Python, PyTorch, YOLO algorithm, and Django are chosen to leverage their respective strengths Python for its simplicity and extensive ecosystem, PyTorch for flexible deep learning model development, YOLO for efficient real-time object detection, and Django for efficient API development. Together, these tools form a powerful stack for building and deploying image classification projects with high performance and accuracy.

3.6 Summary

In this chapter, we have specified the problem of classifying the severity of skin burns through image analysis and provided a brief explanation. We then proceeded to analyze the system, starting with collecting requirements by analyzing previous studies. Following that, we selected our main requirements by filtering those suitable for our system. Subsequently, we outlined our functional and nonfunctional requirements. We presented a step-by-step plan and implementation of our system. Lastly, we detailed the technical tools, including frameworks, algorithms, and programming languages, that we will use in our system.

Chapter 4

System Design

4.1 Introduction

This chapter delves into the design aspects of the project, "Skin Determining Burn Degrees using Deep Learning." A comprehensive system design is crucial for the successful implementation of the proposed solution. This chapter will outline the System Design Specification, Design Architecture, and Scenario to give a detailed view of the project's design.

4.2 System Design Specification

The website features a single-page design, serving as the home page where users can directly upload images. Upon uploading, users immediately receive the results on the same page. This streamlined approach simplifies interaction and enhances user experience by consolidating all functionalities into one accessible location.

4.2.1 Home Page

On this home page Figure 4.1, users can upload an image either by browsing from their computer when accessed through a website, or by taking a picture directly or uploading from the photo gallery if accessed via a mobile phone. After uploading the image and pressing the 'Classify Burn' button, results are displayed immediately on the same page Figure 4.2. This setup provides a seamless and efficient experience, allowing users to quickly obtain results without navigating through multiple pages.

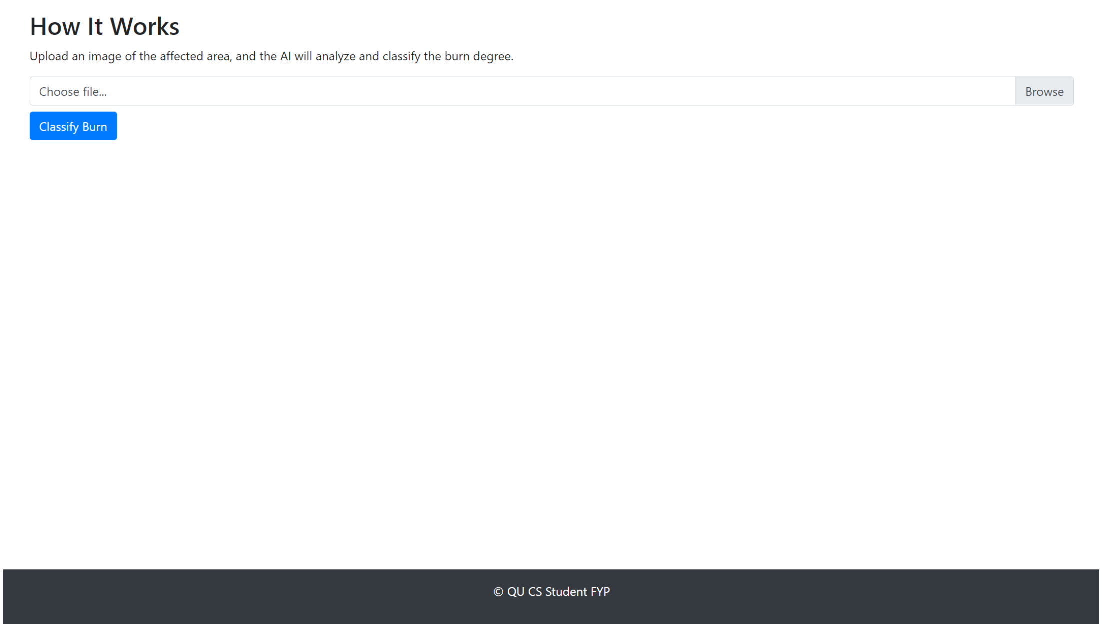

Figure 4.1: Home page

Figure 4.2: result

4.3 Design Architecture

4.3.1 Flow chart:

The system workflow Figure 4.3 initiates with user interactions, encompassing activities like logging in or accessing the home page. Following this, user inputs, including image uploads, undergo a meticulous validation process to ensure data integrity and accuracy. The validated data then advances to AI processing, where dedicated components execute tasks such as image preprocessing and deep learning analysis. The conclusive phase involves displaying the results of the analysis to the user, providing a clear and comprehensible presentation that outlines the identified type and degree of the skin burn. This systematic approach ensures a robust and user-centric experience throughout the various stages of the system's operation.

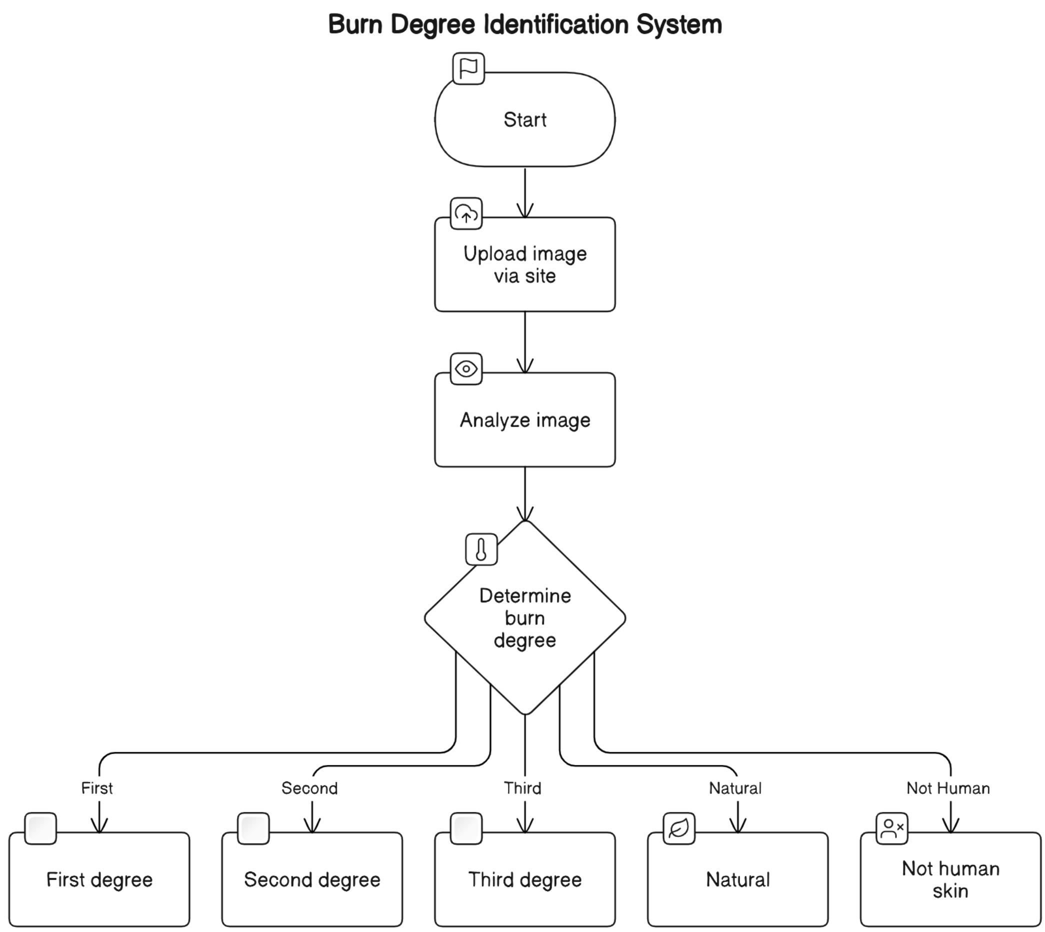

Figure 4.3: Flow chart

4.3.2 Sequence Diagram:

The user initiates the process by uploading an image of a skin injury, marking the commencement of the analysis. Subsequently, the uploaded data undergoes a series of processing steps, encompassing image prepossessing and feature extraction. The processed data then interacts with the deep learning model, initiating an AI algorithm that comprehensively analyzes the skin injury image. The final step involves presenting the results of the analysis to the user, providing detailed insights into the type and degree of the identified skin burn.This sequential workflow Figure 4.4 ensures a comprehensive and user-friendly experience, from input to result presentation.

4.4 Scenario

The user interaction scenario with the web page (as shown in Figure 4.5), a user begins by accessing the web page and proceeds to upload an image. Once the user uploads the image, it is transmitted from the client-side (user's browser) to the web server for processing. The uploaded image undergoes a series of preprocessing steps, including cropping, resizing, filtering, and normalization. These steps are essential for optimizing the image and preparing it for the model classification process.

The preprocessed image is then forwarded to the model classification step, utilizing a pre-trained model (YOLO) for efficient and accurate skin burn degree classification. The model, represented by a neural network, processes the image, culminating in a fully connected layer (FC) that outputs the result.

Finally, the classified result is displayed to the user on the web page, providing information about the skin burn degree identified by the model. This comprehensive process illustrates how a user can seamlessly interact with the web page to receive predictions about his skin health.

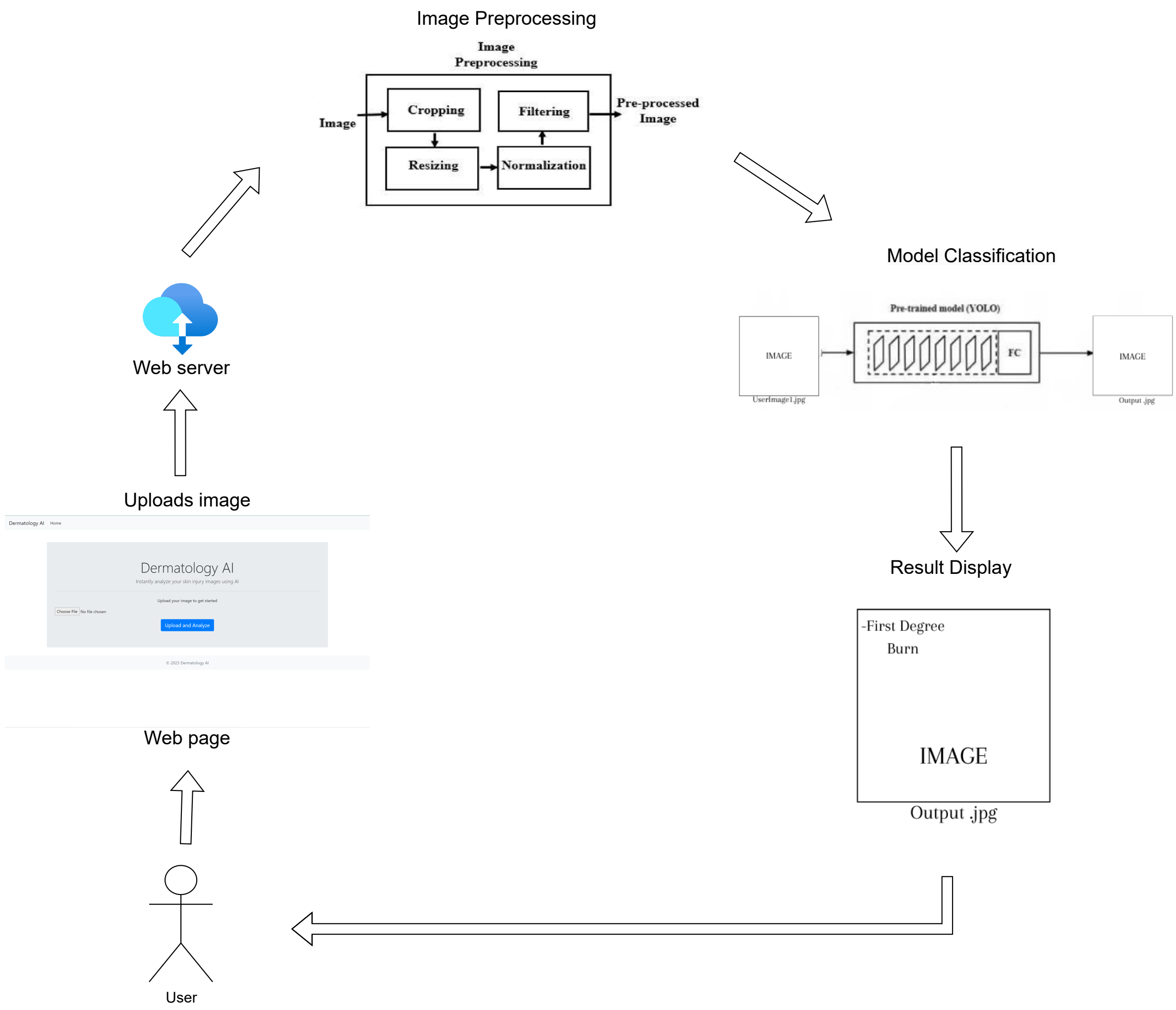

Figure 4.5: Simplified Scenario

4.5 Summary

To sum up this chapter, we mainly discussed the system design of our project. We began with an introduction, followed by specifying the system design and providing the interfaces of our website. We then discussed the design architecture by drawing some charts. Finally, we concluded the chapter by presenting a scenario that demonstrates how the user will interact with the website.

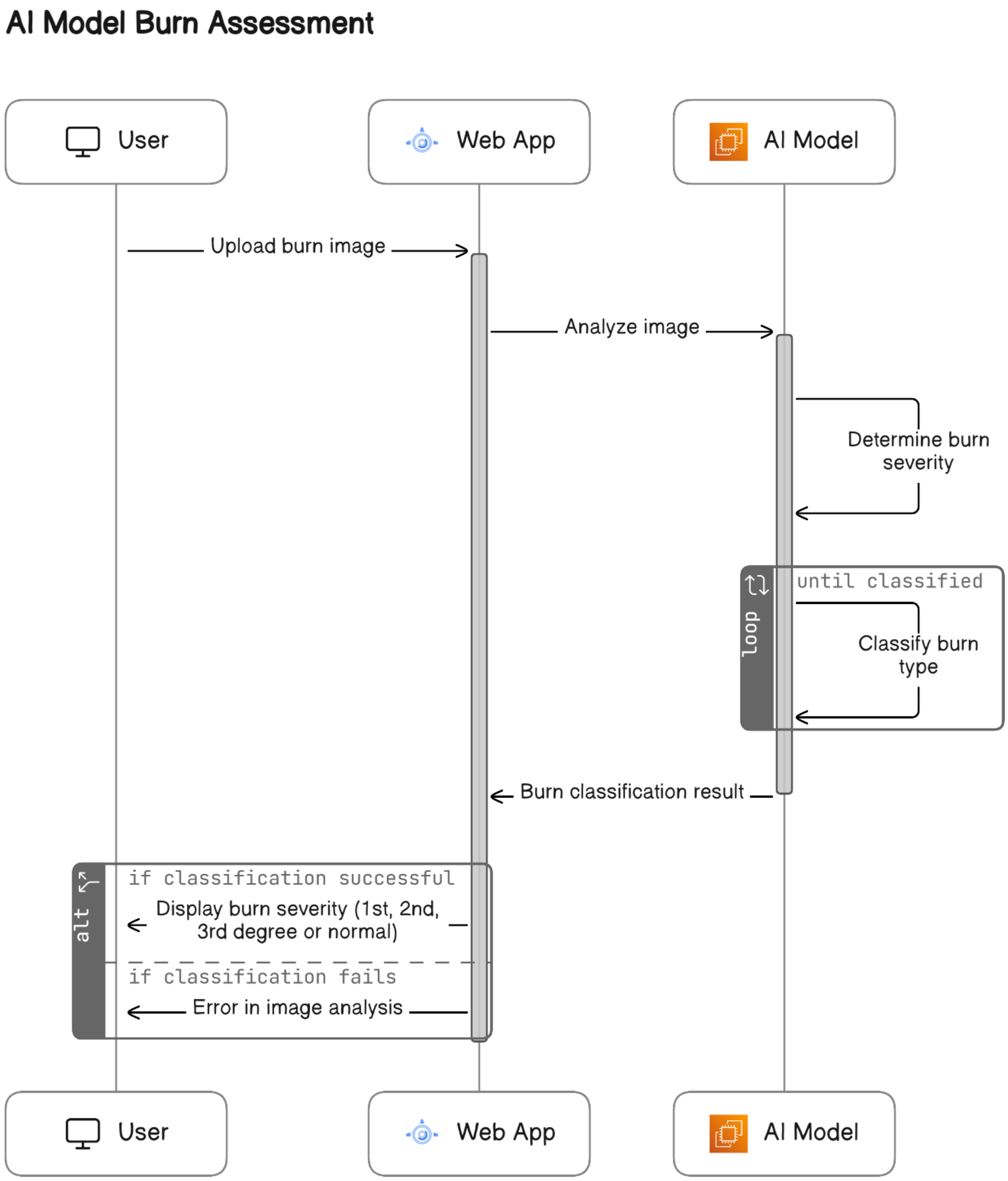

Figure 4.4: Sequence Diagram

Chapter 5

System Implementation

5.1 Introduction

In this chapter, we will represent the implementation process of the project. First, a detailed explanation of the implementation phases is presented. Then the technical tools that were needed for our project are discussed.

5.2 Implementation and Evaluation Plan

We have built our YOLO model using ultralytics in Python and other technical tools highlighted section 3.5. In the following sections, we will shed more light on the built model. For the model evaluation,we used graphs and compared it with other models on the same dataset.

5.3 Dataset

One of the key contributions section 2.4 of this project is the creation of a comprehensive and diverse dataset. To address the challenges posed by varying degrees of burns, we have meticulously collected a total of 2,700 images manually, divided into three categories based on the degree of burn severity. This dataset serves as the foundation for developing a robust solution.

5.3.1 Data Preparation

To prepare the dataset for training the Yolov8 model, the following steps were taken to ensure data quality and relevance:

Data Collection: A diverse dataset was collected by combining publicly available datasets with internally collected data.

Data Cleaning: Duplicated images and images of lower quality were removed from the dataset to mitigate bias and enhance model generalization.

5.3.2 Pre-Processing

Before training the Yolov8-based model, the dataset underwent pre-processing to ensure compatibility with the model architecture and improve its robustness. The following steps were performed:

Image Resizing : All images in the dataset were resized to a resolution of 224x224 pixels to standardize input dimensions for the Yolov8 model.
Data Augmentation: Various data augmentation techniques were employed to augment the dataset and enhance model generalization. These techniques included : Flip, 90-Degree Rotation, Random Rotation, Crop, Shear, Blur, and Random Noise.

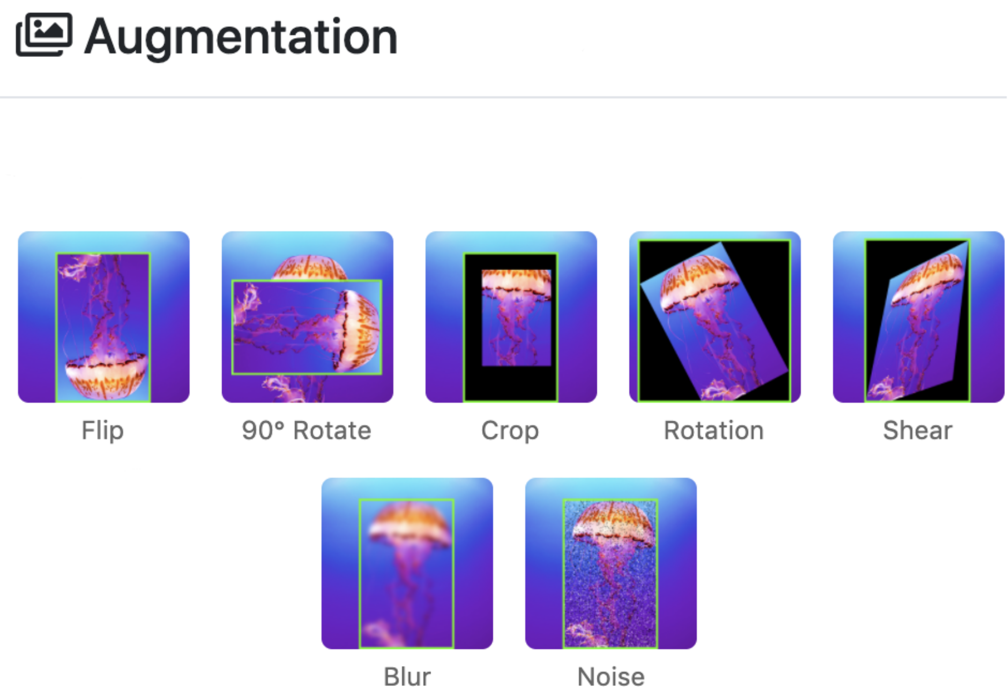

Figure 5.1: Example of the augmentation techniquesZ[2]

5.4 Model Structure

In this section, we describe the technical implementation of the Yolov8 model for image classification. The section is segmented into three main stages: Data Preparation, Pre-Processing, and Model Implementation.

5.4.1 Model Overview

Our approach to training the convolutional neural network model involves several steps (Figure 5.2). The process starts with data collection, followed by data preprocessing which includes image augmentation and resizing to normalize the input data. The dataset was then split into 85% training set and 15% test set.

Model Compilation and Training

The CNN model was compiled using various combinations of loss functions, optimizers, and activation functions to explore their impact on model performance. we tested two types of loss functions:

- **VFL Loss Function**: Varifocal loss function[34] is a dynamically scaled cross entropy loss function .
- **Categorical Crossentropy**: Standard for multiclass classification problems, providing a measure of the difference between the predicted probability and the actual class.

In our experiments with optimizers, we tested:

- **RAdam, AdamW, and NAdam**: These variants of Adam optimizer adjust the learning rates based on different heuristic methods.
- **Stochastic Gradient Descent (SGD)**: A traditional optimizer that utilizes a fixed learning rate and momentum to converge to the optimal weights.

Metrics and Model Evaluation
To monitor and evaluate the model's performance, we focused on two primary metrics:

- **Training Accuracy & Validation Loss**: Training Accuracy helped to gauge the model's effectiveness in learning from the training dataset, while Validation Loss helps in determining how well the model generalizes to new, unseen data.

- **Hyperparameter Tuning**: For optimizing our model's hyperparameters, we used a mutation algorithm from the Ray Tune Python library. This algorithm begins by randomly selecting initial candidate solutions, which represent various sets of hyperparameters. Each candidate is then assessed using a fitness function that evaluates their performance in terms of model effectiveness. The most successful solutions—those which has the highest fitness scores—are selected for mutation.

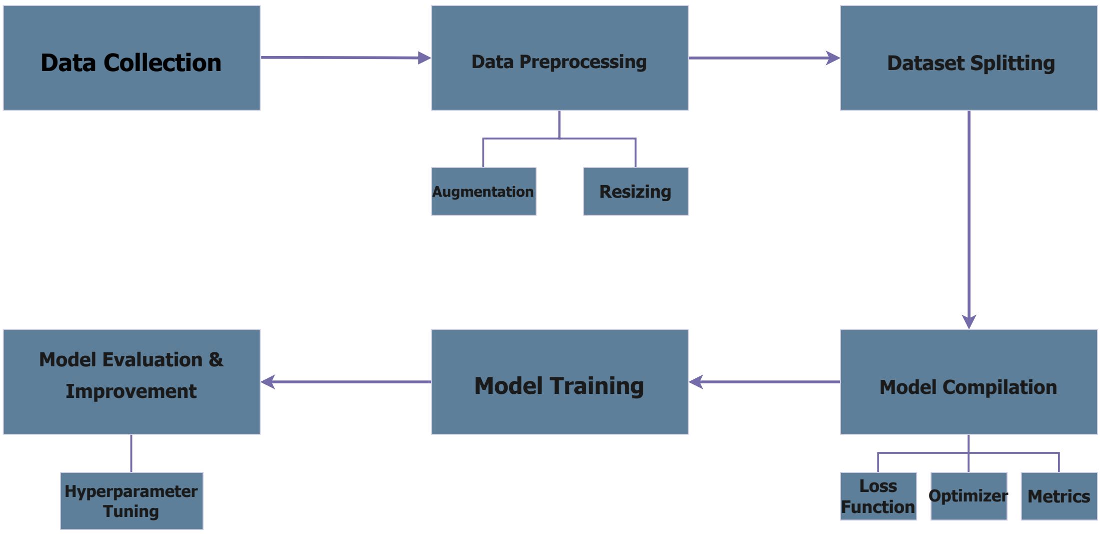

Figure 5.2: General overview of the model workflow

5.4.2 Model Implementation

The architecture of a basic Convolutional Neural Network (CNN) for image classification (Figure 5.3) consists of three main components: Feature Extraction, Pooling, and Classification.

Feature Extraction (Convolutional Layers): The Convolutional layers are responsible for extracting features from the input image. These layers apply convolution operations to the input image, creating feature maps that capture various aspects such as edges, textures, and patterns. This process helps in learning spatial hierarchies of features, which are crucial for image recognition tasks.

Pooling: Following the convolutional layers, pooling layers (max pooling for example) are used to reduce the spatial dimensions of the feature maps. Pooling operations help in down-sampling the feature maps, making the model more computationally efficient and reducing the risk of overfitting. Pooling retains the most significant features while discarding less important information.
Classification (Fully Connected Layers): These layers take the high-level features extracted by the convolutional and pooling layers and use them to perform the classification task. The output from the fully connected layers is passed through an activation function (such as softmax for example) to produce the final class probabilities, enabling the model to make predictions about the input image.

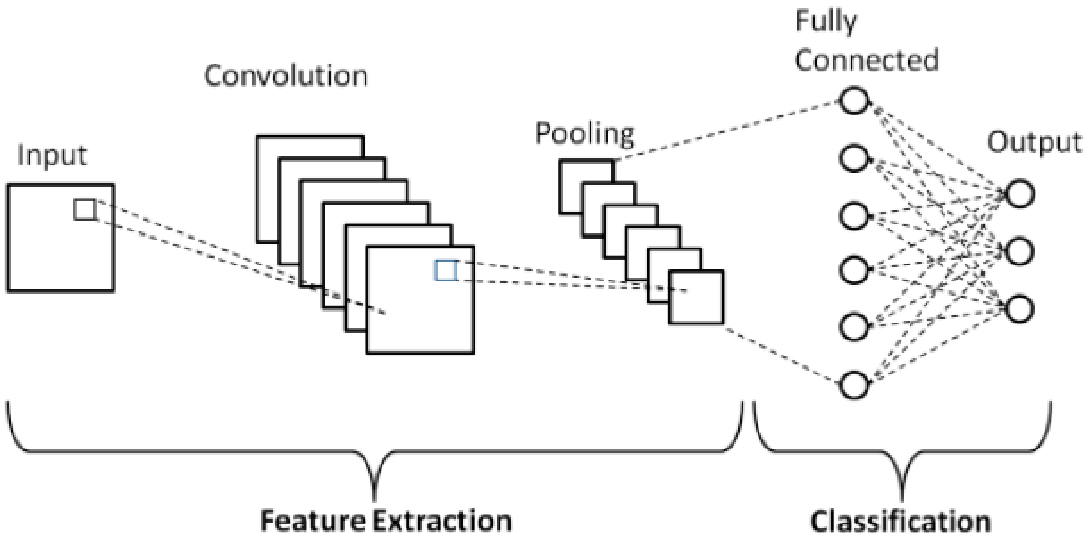

Figure 5.3: Diagram of a basic convolutional neural network (CNN) architecture [3] .

5.5 DJANGO

We build our project using the Django framework as we highlighted in section 3.5, which is one of the most popular web frameworks. Here are the key reasons for our choice:

Rapid Development: Offers numerous built-in features for faster project completion.
Well-Defined Structure: Promotes reusability and maintainability of code.
Strong Community Support: Provides extensive third-party packages and resources.
Built-in Security Features: Includes protections against SQL injections, XSS, and CSRF.
Scalability: Effectively handles high traffic and complex data operations.

5.6 Summary

In this chapter we detailed the implementation of our project, focusing on the YOLOv8 model for image classification. We discussed the creation and preparation of a dataset with 2,700 images categorized by burn severity, including steps like image resizing and data augmentation to improve model robustness.

We outlined the general CNN model architecture and training process, comparing different loss functions and optimizers. Metrics such as training accuracy and validation loss were used to evaluate performance, and hyperparameter tuning .

Finally, we highlighted the use of the Django framework for deployment, noting its rapid development capabilities, well-defined structure, strong community support, built-in security features, and scalability.

Chapter 6

experiments and Result Discussions

6.1 Introduction

In this chapter, we will explore the various experiments that we conducted in our quest to achieve the best possible result and increase the efficiency of the model used, then we will review the final result achieved.

6.2 experiments

6.2.1 Dataset:

We conducted experiments using a variety of datasets, some sourced from the internet and others compiled by us, featuring diverse volumes of data. Unfortunately, these datasets did not yield encouraging results. Consequently, we focused on gathering the best possible images to enhance the quality and relevance of our dataset. Ultimately, we successfully compiled a collection of 2700 images, categorized into three burn severity levels.

6.2.2 CNN Models:

In this analysis, we compare the performance of various convolutional neural network (CNN) models on the same dataset to determine their efficacy in specific tasks. We focus on some well-known architectures which are the following: YOLO, DenseNet,

DenseNet169, InceptionV3, MobileNetV2, NASNetMobile, InceptionResNetV2, ConvNeXtXLarge, VGG19, and VGG16. By assessing their Top-1 accuracy, we aim to identify which model performs best in our scenario and how each model's unique architecture influences its accuracy.

Model	Top 1 Accuracy
YOLO	93%
VGG16	81%
VGG19	79%
DenseNet	86%
DenseNet169	88%
InceptionV3	89%
MobileNetV2	91%
NASNetMobile	87%
ConvNeXtXLarge	75%
InceptionResNetV2	91%

Table 6.1: Comparison of CNN Models on the Same Dataset

6.2.3 Model optimization:

Based on the results we achieved with the accuracy of the YOLO model (Table 6.1), we decided to conduct experiments by changing the optimization function on YOLO to see if there would be any impact on the results.

Optimizer	Accuracy	Validation Loss	Epochs
Stochastic Gradient Descent (SGD)	92%	0.63178	150
NAdam	91%	0.63607	150
RAdam	93%	0.62158	200
AdamW	91%	0.62988	300

Table 6.2: Comparing the Performance of Optimizers

As we can see in the table (Table 6.2), the best result obtained was 93% with RAdam, therefore it was adopted.

6.3 Results

After all the experiments outlined in the previous sections, the mentioned dataset consisting of 2700 images was adopted, and the YOLOV8 model was trained on it using the RAdam optimization function because it achieved the highest results and surpassed other comparisons.

Chapter 7

Conclusion

In this project, we explored the effectiveness of various convolutional neural network (CNN) models, including YOLOv8, for the diagnosis and severity assessment of skin burns, utilizing a carefully curated dataset of 2700 images. Our extensive experiments, as discussed in the previous chapters, highlight our methodological rigor and commitment to optimizing model performance. The adoption of the RAdam optimizer, following comparative analysis with other optimization functions, significantly enhanced the accuracy and reliability of the YOLOv8 model.

The results achieved underscore the potential of using deep learning in medical diagnostics to reduce dependence on subjective human evaluation, thereby increasing the consistency and speed of burn severity assessments. This approach not only improves patient outcomes by facilitating timely and accurate treatment but also holds promise for application in remote areas where specialized medical expertise may be lacking.

An integral part of our project's success is the web interface developed using Django. This interface provides an accessible platform for healthcare professionals to quickly upload images and receive immediate burn severity assessments, streamlining the diagnostic process in clinical settings.

Looking forward, we aim to expand our dataset and explore further enhancements in model architecture and training techniques to refine our predictions. Moreover, we aspire to develop capabilities to recommend the best possible medical interventions for each type of burn across different scenarios, enhancing the practical utility of our model in emergency medical situations. This ongoing development reflects our commitment to advancing medical technology through AI, improving healthcare accessibility and quality worldwide.

Bibliography

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-dujaili, Ye Duan, Omran Al-Shamma, José I. Santamaría, Mohammed Abdulraheem Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 2021. URL https://api.semanticscholar.org/CorpusID:232434552.

[2] Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow. URL https://roboflow.com.

[3] Van Hiep Phung and Eun Joo Rhee. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences*, 9 (21), 2019. ISSN 2076-3417. doi: 10.3390/app9214500. URL https://www.mdpi.com/2076-3417/9/21/4500.

[4] John P Abraham, Brian D Plourde, Lauren J Vallez, Brittany B Nelson-Cheeseman, John R Stark, Ephraim M Sparrow, and John M Gorman. Skin burns. *Theory and Applications of Heat Transfer in Humans*, 2:723–739, 2018.

[5] PN Kuan, S Chua, EB Safawi, HH Wang, and W Tiong. A comparative study of the classification of skin burn depth in human. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-10):15–23, 2017.

[6] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022.

[7] Brenda Rangel-Olvera and Roberto Rosas-Romero. Detection and classification of burnt skin via sparse representation of signals by over-redundant dictionaries. *Computers in Biology and Medicine*, 132:104310, 2021. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2021.104310. URL https://www.sciencedirect.com/science/article/pii/S0010482521001049.

[8] Nehemiah T. Liu and Jose Salinas. Machine learning in burn care and research: A systematic review of the literature. *Burns*, 41(8):1636–1641, 2015. ISSN 0305-4179. doi: https://doi.org/10.1016/j.burns.2015.07.001. URL https://www.sciencedirect.com/science/article/pii/S0305417915002004.

[9] Dhirendra Yadav, Ashish Sharma, Madhusudan Singh, and Ayush Goyal. Feature extraction based machine learning for human burn diagnosis from burn images. *IEEE Journal of Translational Engineering in Health and Medicine*, PP:1–1, 07 2019. doi: 10.1109/JTEHM.2019.2923628.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified,

real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[12] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

[13] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007. doi: 10.1080/01431160600746456. URL https://doi.org/10.1080/01431160600746456.

[14] Muhammad Mateen, Junhao Wen, Nasrullah, Sun Song, and Zhouping Huang. Fundus image classification using vgg-19 architecture with pca and svd. *Symmetry*, 11(1):1, 2018.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[17] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[18] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine*, 121:103792, 2020.

[19] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[20] N Aishwarya, K Manoj Prabhakaran, Frezewd Tsegaye Debebe, M Sai Sree Akshitha Reddy, and Posina Pranavee. Skin cancer diagnosis with yolo deep neural network. *Procedia Computer Science*, 220:651 – 658, 2023. ISSN 18770509.

[21] Nawal Soliman ALKolifi ALEnezi. A method of skin disease detection using image processing and machine learning. *Procedia Computer Science*, 163:85–92, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.12.090. URL https://www.sciencedirect.com/science/article/pii/S1877050919321295. 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence.

[22] Behrouz Rostami, DM Anisuzzaman, Chuanbo Wang, Sandeep Gopalakrishnan, Jeffrey Niezgoda, and Zeyun Yu. Multiclass wound image classification using an ensemble deep cnn-based classifier. *Computers in Biology and Medicine*, 134:104536, 2021.

[23] Safawi E.B., Wang H.H., Chua Stephanie, and Kuan P.N. A comparative study of segmentation algorithms in the classification of human skin burn depth. *International Journal on Advanced Science, Engineering and Information Technology*, 10:145, 2020. ISSN 2460-6952.

[24] Sayma Alam Suha and Tahsina Farah Sanam. A deep convolutional neural network-based approach for detecting burn severity from skin burn images. *Machine Learning with Applications*, 9:100371, 2022.

[25] Aliyu Abubakar, Mohammed Ajuji, and Ibrahim Usman Yahya. Comparison of deep transfer learning techniques in human skin burns discrimination. *Applied System Innovation*, 3(2):20, 2020.

[26] Holger Haenssle, Christine Fink, R Schneiderbauer, Ferdinand Toberer, Timo Buhl, A Blum, A Kalloo, A Hassen, Luc Thomas, A Enk, and Lorenz Uhlmann. Man against machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of oncology : official journal of the European Society for Medical Oncology*, 29, 05 2018. doi: 10.1093/annonc/mdy166.

[27] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in neural information processing systems*, 34:980–993, 2021.

[28] Sai Yeshwanth Chaganti, Ipseeta Nanda, Koteswara Rao Pandi, Tavva GNRSN Prudhvith, and Niraj Kumar. Image classification using svm and cnn. In *2020 International conference on computer science, engineering and applications (ICCSEA)*, pages 1–5. IEEE, 2020.

[29] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[30] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[31] Ibrahim Saad Aly Abdelhalim, Mamdouh Farouk Mohamed, and Yousef Bassyouni Mahdy. Data augmentation for skin lesion using self-attention based progressive generative adversarial network. *Expert Systems With Applications*, 165, 2021. ISSN 0957-4174.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[33] Django Software Foundation. Django. URL https://djangoproject.com.

[34] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8514–8523, 2021.