Suez Canal University

Egyptian Chinese College for Applied Technology

Department of Communication Technology

Graduation Project Thesis

Chest X-Ray 14 Diseases Detection and classification System using AI

Prepared by

Salwa Mahmoud Ismail Ali

Supervised By

Ph.D. Ahmed Salem   Dr. Zakaria Hassan   Dr. Ahmed Magdy

Dr. Nariman Abdo   Dr. Emad Badry

June 2024

Ismailia, Egypt

Acknowledgements

I wish to express my sincere gratitude to the individuals and organizations who have supported me throughout this journey. Firstly, I will thank the great God (Allah) who gives us the ability to finish this work. I am deeply thankful to my parents for their constant encouragement and my two beloved sisters. Their support has been invaluable to my success.

I am indebted to my supervisors, Ph.D. Ahmed Salem, Dr. Ahmed Magdy, Dr. Zakaria Hassan, Dr. Nariman Abdo and Dr. Emad Badry, and for sure Dr. Ranya Moustafa. Their expertise, guidance, and encouragement have been instrumental in shaping this project. I am honored to have had the opportunity to work with them.

I also want to extend my appreciation to my teammates – Caroline Magdy, Yasser Mamdouh, Ahmed Abdulrahim, Yousef Sherif, and Yousef Ala'a. Their collaboration, insightful discussions, and suggestions have greatly enriched the research process. I am truly grateful for their contribution.

In conclusion, I am overwhelmed with gratitude for all those who have contributed to making this journey possible.

Thank you.

Abstracts

In today's fast-paced world, healthcare systems face numerous challenges, including long waiting times, inaccuracies in diagnosis, and the need for smarter and more efficient healthcare solutions. To address these challenges, we propose an innovative project titled "Autonomous Diagnosis Disease Detection and E-learning System." This project integrates advancements in Communications, Mechatronics, and Electronics to revolutionize the healthcare industry.

At the core of our system is a sophisticated robot capturing Chest x-ray images and make diagnosis of 14 Different Diseases and is equipped with a range of sensors capable of measuring patients' vital signs. These sensors provide real-time data, which is then seamlessly transmitted to a machine learning model or prediction system. Utilizing state-of-the-art algorithms, this system analyses the data to determine whether patients may have a disease or not.

One of the key features of our project is the integration of an intuitive mobile application called "VitoM" (Viral Monitoring). This application serves as a platform for uploading and accessing diagnosis results, allowing patients to receive timely feedback and recommendations. By automating the diagnosis process and eliminating the need for human intervention, we aim to reduce waiting times and improve the accuracy of diagnoses.

Furthermore, our project includes an innovative E-learning system designed to support undergraduate doctors and medical students. This platform offers access to advanced topics and allows users to analyse stored patient cases, which are saved as profiles for each patient. By leveraging technology to enhance medical education and training, we empower healthcare professionals to make informed decisions and deliver high-quality care.

Table of Contents

Chapter 1. Introduction to AI in healthcare 1

1. Introduction 2
2. What is Artificial Intelligence 3
3. Understanding the different types of artificial intelligence 3
4. Based on Capabilities 4
   - 4.A Artificial Narrow AI 4
   - 4.B General AI 4
   - 4.C Super AI 5
5. Based on Technologies 5
   - 5.A Machine Learning 5
   - 5.B Deep learning 5
   - 5.C Natural Language Processing 6
   - 5.D Computer vision 6
   - 5.E Robotics 6
   - 5.F Expert systems 7

Chapter 2. Literature review 8

1. Medical Robots 8
2. What can AI do to help diagnose diseases? 9
   - 2.A Medical Imaging Analysis: 9
   - 2.B Clinical Data Analysis: 9
   - 2.C Diagnostic Decision Support: 9
   - 2.D Screening Programs: 10
   - 2.E Natural Language Processing (NLP): 10

3. What is the need of using AI Robots? 11
3.A Automation of Routine Tasks 11
3.B Patient Monitoring and Engagement 11
3.C Efficient Data Collection and Analysis 11
3.D Helps med student in learning process 11
4. The Medical Segmentation Decathlon 11
5. Cardiovascular disease 12
5.A Coronary heart disease 13
5.B Strokes and TIAs 13
5.C Peripheral arterial disease 13
5.D Aortic disease 14
6. Causes of CVD 14
6.A High blood pressure 14
6.B Smoking 14
6.C High cholesterol 14
6.D Diabetes 14
6.E Inactivity 15
6.F Being overweight or obese 15
6.G Family history of CVD 15
6.H Ethnic background 15
7. The Study of Machine Learning 16
8. The Importance of Machine Learning 16
9. The most prominent ML projects in healthcare 17
9.A Digital Pathology Detection 17
9.B Medical Imaging Analytics 17

9.C Healthcare Operations Automation 17

9.D Genomic Medicine and Drug Discovery Process 17

9.E Predictive Analytics and Diagnostic 18

9.F Biomedical Informatics 18

9.G Acute Stroke Management 18

9.H Patient Care Coordination 18

9.I Medical Imaging Interpretation and Diagnostic Support 18

9.J Remote Patient Monitoring 18

Chapter 3. How does Artificial Intelligence work? 20

1. Problem statement 21

1.A Chest radiography 21

1.B Cardiovascular diseases 21

2. basics of data in ML 22

2.A Labeled data 22

2.B Unlabeled data 22

3. Data Acquisition 22

4. Why do we need Data Acquisition? 22

4.A General Data Science and Machine Learning Datasets 23

4.B Healthcare-Specific Datasets 23

5. Dataset Glossary (Column-wise) 23

6. What is data preprocessing? 24

7. Why is data preprocessing necessary? 25

8. Tools and Libraries 25

9. Purpose of data preprocessing 26

9.A Get Data Overview 26

9.B Identify Missing Data 26

9.C Identify Outliers or Anomalous Data 27

9.D Remove Inconsistencies 27

10. popular data pre-processing techniques 27

10.A Handling missing values 27

10.B Scaling 28

10.C Outlier treatment 29

10.D Feature encoding 30

Chapter 4. Project design 34

1. Project Workflow Diagram 34

2. Mechatronics Department 35

2.A Introduction about nursing robot 35

2.B Medical Nursing robot steps for designing 37

2.C Introduction of computer-aided Design (CAD) 40

2.D Evolution and History: 40

2.E Core Functionality: 40

2.F Types of CAD Software: 40

2.G Benefits of CAD: 41

2.H Applications: 41

2.I Advantages of Humanoid Robot Design for Biomedical Applications 42

3. Electronics Department 44

3.A Electronic Health Records 44

3.B Raspberry Pi 44

3.C Features of Raspberry Pi 45

3.D Warnings....47

3.E Safety Instructions....48

3.F How MAX30100 Pulse Oximeter and Heart Rate Sensor Works?....48

F.1 Heart Rate Measurement....49

3.G MAX30100 Module Hardware Overview....50

3.H Temperature Sensor....53

4. Communications Department....61

5. Biosensors and Electronics:....61

6. AI and robotics are transforming healthcare....61

6.A Keeping Well:....61

6.B Early Detection:....62

6.C Diagnosis:....62

6.D Decision Making:....62

6.E Treatment:....63

6.F Research:....63

6.G Training:....63

7. Tasks Machine Learning addresses in healthcare....64

7.A Classification:....64

7.B Recommendations:....64

7.C Clustering:....64

7.D Prediction:....64

7.E Anomaly detection:....64

7.F Automation:....64

7.G Ranking:....65

8. Application of Machine Learning in Healthcare Communication....65

8.A Patient Care 65
8.B Education and Knowledge Transfer System 65
8.C Emergency Response and COVID-19 66
9. Machine learning terminologies 67
9.A Features 67
9.B Target Variable 67
9.C Algorithm 67
9.D Classification 68
9.E Classification Threshold 68
9.F Dimension 68
9.G Hyperparameters 68
9.H Model 68
9.I Normalization 68
9.J Noise 69
9.K Overfitting 69
9.L Underfitting 69
9.M Precision 69
9.N Recall 70
9.O Recall vs Precision 70
9.P Specificity 70
9.Q ROC (Receiver Operating Characteristic) Curve 71
9.R PR AUC (area under the PR curve) 71
9.S test set 71
10. Machine learning basic classification 71
10.A Supervised learning 71

10.B Unsupervised learning 72

10.C Reinforcement learning 72

11. What is a machine learning algorithm? 72

12. How machine learning algorithms work 73

12.A Supervised learning algorithms 73

12.B Unsupervised learning algorithms 76

13. Applications of Computer Vision to Medical Diagnosis 76

13.A Dermatology 76

13.B Ophthalmology 78

13.C Histopathology 79

14. Chest X-Ray 14 Diseases Detection 81

14.A Definition of Deep Learning 81

14.B Choice of deep neural networks 82

14.C Image Classification 82

14.D Definition of Transfer Learning 83

14.E How Transfer Learning Works? 84

14.F Steps to Use Transfer Learning 85

14.G Features Extraction 85

14.H Chest X-Ray model Tools 87

14.I Prepare and load data for successful ML outcomes 88

14.J Build and fine-tune models with the TensorFlow ecosystem 89

Chapter 5. Implementation 90

1. System setup 90

2. Hardware Configuration 90

3. Software Environment 91

3.A Installing Anaconda........91

3.B Creating a Conda Environment........91

3.C Installing Required Libraries........91

4. Multi-Label Chest X-Ray Classification via Deep Learning........93

4.A NIH Chest X-ray Dataset........94

4.B Data limitations:........95

4.C Class descriptions........95

4.D Full Dataset Content........95

4.E Import Packages and Functions........96

4.F Load the Datasets........96

4.G Data Visualization........97

4.H Investigate a single image........97

4.I Preventing Data Leakage........98

4.J Model Development........100

4.K Addressing Class Imbalance........100

4.L Impact of class imbalance on loss function........102

4.M Computing Class Frequencies........103

4.N Best performing algorithm........105

4.O Image Classification and Class Imbalance........107

4.P Transfer Learning (pretraining Fine tuning)........107

4.Q Training procedure........107

4.R Training, Prediction, and Loss........109

5. Setting Up the Raspberry Pi........111

5.A Installing Raspbian OS........111

5.B Environment setup........111

5.C Installing Required Libraries 111

6. Camera Module 112

6.A Package installation code 112

6.B Capture an Image 112

7. Firebase Integration 112

7.A firebase-admin for database interaction 112

7.B Get Your Credentials JSON File 112

7.C Initialize Firebase Admin SDK 112

7.D Captured image uploading into a firebase cloud storage 113

7.E Access the Image 113

8. Robot design 114

8.A Sketches 114

8.B Fiberglass 116

8.C Creating Construction Planes and Sketches 119

8.D Adding Guide Rails 120

8.E Using the Loft Feature 121

8.F Employing the Shell Feature 122

8.G Applying the Extrude Cut Feature 125

8.H The Extrude Boss feature 128

8.I The Fillet features 130

8.J Robot Arms 132

8.K Inserting a Shelf 133

8.L Including Outside Doors 133

8.M Robot head 134

8.N Slider Mechanism 134

8.O Acrylic 136

8.P Drawer diffuser 138

8.Q Computer Numerical Control (CNC) 139

Chapter 6. Results and Discussion 142

1. Heart attack model Evaluation 142

1.A Best model selection 142

1.B Model saving 143

1.C Preventing CVD 144

2. Chest X-ray model Evaluation 146

2.A VGG16 algorithm 146

2.B Inception 148

2.C Applications of Inception: 149

2.D Comparison to Traditional CNNs: 149

2.E DensNet algorithm 151

2.F DenseNet Variants and Applications: 152

2.G Comparison to ResNet (another popular CNN): 152

2.H Chest X-ray model Results 152

2.I comparing performances with other Research papers 154

2.j Test Metrics 156

..... 158

3. Challenges of Deep Learning in healthcare applications 159

3.A Visualizing Learning with GradCAM 159

3.B GradCam flowchart 159

3.C Heat maps 160

3.D interpreting prediction Diseases (on Raspberry pi) 163

....163

.4 Challenges Faced 164

4.A Radiologist wrong Diagnosis...164

4.B Solution...164

C.4 Problems with the Dataset..164

5. Imbalance in the dataset 165

....165

5.A Solution 165

....166

6. Future Work...167

6.A Integration with CT Scans...167

6.B MRI Integration..167

6.C Enhancing Accuracy...167

Chapter 7. References 169

List of Figures

Figure 1 Medical robots.. **Error! Bookmark not defined.**

Figure 2 Medical Segmentation 12

Figure 3 Machine Learning model Lifecycle 20

Figure 4 Python and most common AI packages 26

Figure 5 Box plots can be used to identify the outliers.. 29

Figure 6 Input data for OneHotEncoder example 31

Figure 7 Final output after OneHotEncoder 31

Figure 8 Input data for Label Encoding example 32

Figure 9 Output data after Label Encoding.. 32

Figure 10 Input data for Frequency Encoding example 33

Figure 11 Output data after Frequency Encoding.. 33

Figure 12 Project Workflow Diagram 34

Figure 13 nursing robot 37

Figure 14 nursing robot in healthcare 39

Figure 15 Raspberry Pi model 4 B [22] 45

Figure 16 Raspberry Pi 4 with PoE HAT [23] 47

Figure 17 MAX30100-Pulse-Detection-Photoplethysmogram 48

Figure 18 Heart Rate Measurement.. 49

Figure 19 pulse Oximeter 50

Figure 20 MAX30100 Module Hardware Overview 51

Figure 21 power Requirement 51

Figure 22 MLX90614 Module Hardware 54

Figure 23 Built-In Optical Filter 55

Figure 24 Power Requirement 56

Figure 25 Infrared Thermometers... 57

Figure 26 field of view 1 58

Figure 27 field of view 2 58

Figure 28 Dermatology Detection... 77

Figure 29 inception algorithm for Dermatology Detection 78

Figure 30 Ophthalmology Detection-1 79
Figure 31 Ophthalmology Detection-2 79
Figure 32 Histopathology Detection 80
Figure 33 CNN Architecture 83
Figure 34 Transfer Learning 84
Figure 35 Feature extraction 86
Figure 36 Hardware setup for the healthcare monitoring system. 90
Figure 37 Heart Attack data 1 .... **Error! Bookmark not defined.**
Figure 38 Heart Attack data 2 .... **Error! Bookmark not defined.**
Figure 39 Heart Attack data 3 .... **Error! Bookmark not defined.**
Figure 40 Blood Pressure encoding .... **Error! Bookmark not defined.**
Figure 41 statistical analysis of categorical data .... **Error! Bookmark not defined.**
Figure 42 summary statistics for data 1 .... **Error! Bookmark not defined.**
Figure 43 summary statistic for data 2 .... **Error! Bookmark not defined.**
Figure 44 summary statistic for data 3 .... **Error! Bookmark not defined.**
Figure 45 summary statistic for data 4 .... **Error! Bookmark not defined.**
Figure 46 Distribution of Heart Attack risk .... **Error! Bookmark not defined.**
Figure 47 Male distributions of Heart Attack risk .... **Error! Bookmark not defined.**
Figure 48 Female distributions of Heart Attack .... **Error! Bookmark not defined.**
Figure 49 Age distributions .... **Error! Bookmark not defined.**
Figure 50 Diet distributions for Male .... **Error! Bookmark not defined.**
Figure 51 Diet distributions for Female .... **Error! Bookmark not defined.**
Figure 52 Alcohol consumption and Heart Attack risk .... **Error! Bookmark not defined.**
Figure 53 female smoking distributions .... **Error! Bookmark not defined.**
Figure 54 Triglycerides distributions .... **Error! Bookmark not defined.**
Figure 55 BMI distributions .... **Error! Bookmark not defined.**
Figure 56 Heart rate distribution .... **Error! Bookmark not defined.**
Figure 57 Triglyceride distribution .... **Error! Bookmark not defined.**
Figure 58 SMOTE to handle data imbalance .... **Error! Bookmark not defined.**
Figure 59 A chest x-ray identifies a lung mass. 94
Figure 60 A chest x-ray identifies a lung mass. 97

Figure 61 3 A chest x-ray image 98
Figure 62 processed chest x-ray image. 100
Figure 63 plot the frequency of each of the labels in our dataset 101
Figure 64 plot Class Frequencies... 103
Figure 65 plot Class Frequencies balancing 104
Figure 66 A deep DenseNet with three dense blocks. 105
Figure 67 Image Classification and Class Imbalance 107
Figure 68 Transfer learning 107
Figure 69 training the datas... 109
Figure 70 Initial design 115
Figure 71 A visualization of the final design 1 116
Figure 72 Robust frame: Crafted seamlessly from premium fiberglass. 118
Figure 73 construction plane 120
Figure 74 Take control of your creations: Guide rails in Fusion 360 for shaping lofts and sweeps. 121
Figure 75 Create Hollow Components with Precise Wall Thickness for Reduced Weight and Material Savings 124
Figure 76 The Extrude Cut Feature in Fusion 360 for Creating Complex Cutouts and Features in 3D Models 127
Figure 77 A physical model of our robot in development.. 130
Figure 78 Make models look nice and work better! Use round edges in Fusion 360........... 132
Figure 79 A detailed 3D rendering of a non-moving robotic arm....................................... 132
Figure 80 Inner shelf: Easy access to a robot's internal components. 133
Figure 81 Interior access panel of a robot, showcasing internal components.................... 133
Figure 82 Abstract robot head design 134
Figure 83 Precision in Motion: Exploring the Versatility of Slider Mechanisms.................. 135
Figure 84 Acrylic Drawer Diffuser 138
Figure 85 Crafted with precision using CNC technology: the robot's base. 141
Figure 86 Confusion Matrix for Random Forest (500 Estimators, Gini Criterion) - Best Model 143
Figure 87 Cardiovascular Problems and How to Prevent Them 145

Figure 88 Architecture of the modified VGG16 model 147
Figure 89 12 Inception-V3 architecture, trained for Chest X-ray images. The output of the architecture is further utilized for transfer learning, multi-layer feature fusion and selection 150
Figure 90 Test metrics overall summary.................................... **Error! Bookmark not defined.**
Figure 91 Confusion matrix for DenseNet. 156
Figure 92 AUC Curve for best performance.. 155
Figure 93 Biomedical project GitHub repo 176
Figure 94 Fablabs: Project partner 176

List of tables

Table 1 Temperature sensor's technical specifications 53

Table 2 MLX90614 specifications 56

Table 3 Chest X-ray Radiology report 96

Table 4 DensNet Architecture for ImageNet 106

Table 5 Performance comparison of presented model with state-of-the-art models on basis of AUC 154

Abbreviation list

AI	**Artificial intelligence**
ML	Machine learning
DL	Deep Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
NLP	Natural language processing
MSD	Medical Segmentation Decathlon
CXR	Chest X-ray
MRI	magnetic resonance imaging
NIH	National Institutes of Health
DensNet	Densely Connected Convolutional Networks
ECCAT	Egyptian Chinese College for Applied Technology
CAD	computer-aided detection and diagnosis
Relu	Rectified Linear Unit.
Grad-CAM	Gradient-weighted Class Activation Mapping.
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve

Chapter 1. Introduction to AI in healthcare

1. Introduction

Artificial Intelligence or AI is reshaping healthcare, making it more efficient and effective than ever. **AI** is being used in healthcare for everything from answering patient questions to assisting with surgeries and developing new pharmaceuticals. AI is getting increasingly sophisticated at doing what humans do, but more efficiently, more quickly and at a lower cost. The potential for both AI and robotics in healthcare is vast. Just like in our everyday lives, AI and robotics are increasingly a part of our healthcare eco-system.

The project presents the development and implementation of a robotic system equipped with artificial intelligence (AI) technology to detect chest diseases through the analysis of X-ray images. The robotic system integrates a camera for capturing X-rays and sensors for conducting comprehensive body checkups. The primary objective is to enhance medical education by providing medical students with a powerful tool to visualize and understand various chest diseases and associated physiological parameters.

The robot employs advanced AI algorithms to analyze chest X-ray images, identifying potential abnormalities indicative of a range of chest diseases, including but not limited to pneumonia, pulmonary fibrosis, pneumothorax, and cardiomegaly. The integration of sensors enables the robot to conduct a thorough body checkup by measuring vital signs, such as temperature, heart rate, and blood pressure. This holistic approach allows for a more comprehensive understanding of a patient's health status.

The project also includes the development of an intuitive and user-friendly application that serves as a visualization tool for medical students. The application provides a platform for medical students to observe and study real-world cases, enhancing their diagnostic skills and understanding of chest diseases. The robot's findings are presented through the application, allowing students to correlate clinical data, X-ray images, and sensor measurements for a more holistic patient assessment.

Key features of the system include its ability to provide real-time feedback, ensuring timely and accurate detection of potential chest diseases. Additionally, the robot and application combination facilitate remote learning, enabling medical students to access and analyze patient data from various locations. The project aims to bridge the gap between theoretical knowledge and practical application, fostering a deeper understanding of chest diseases and diagnostic processes among medical students.

The outcomes of this project are expected to contribute to the education and training of medical professionals, offering an innovative and interactive approach to chest disease diagnosis. The integration of AI, robotics, and medical imaging technologies holds great promise for enhancing medical education and ultimately improving patient care.

2. What is Artificial Intelligence

Artificial intelligence refers to computer systems that can perform tasks traditionally associated with human intelligence such as making predictions, identifying objects, interpreting speech, and generating natural language. AI systems learn how to do so by processing massive amounts of data and looking for patterns to model in their own decision-making. In many cases, humans will supervise an AI's learning process, reinforcing good decisions, and discouraging bad ones, but some AI systems are designed to learn without supervision [1].

3. Understanding the different types of artificial intelligence

Early iterations of the AI applications we interact with most today were built on traditional machine learning models. These models rely on learning algorithms that are developed and maintained by data scientists. In other words, traditional machine learning models need human intervention to process new information and perform any new task that falls outside their initial training. For example, Apple made Siri a feature of its iOS in 2011. This early version of Siri was trained to understand a set of highly specific statements and requests. Human intervention was required to expand Siri's knowledge base and functionality.

However, AI capabilities have been evolving steadily since the breakthrough development of artificial neural networks in 2012, which allow machines to engage in reinforcement learning and simulate how the human brain processes information. Unlike basic machine learning models, deep learning models allow AI applications to learn how to perform new tasks that need human intelligence, engage in new behaviors, and make decisions without human intervention. As a result, deep learning has enabled task automation, content generation, predictive maintenance, and other capabilities across industries.

Artificial Intelligence can be broadly classified into several types based on capabilities, functionalities, and technologies.

Here's an overview of the different types of AI:

4. Based on Capabilities

4.A Artificial Narrow AI

Artificial Narrow Intelligence, also known as Weak AI, what we refer to as Narrow AI is the only type of AI that exists today. Any other form of AI is theoretical. It can be trained to perform a single or narrow task, often far faster and better than a human mind can. However, it can't perform outside of its defined task. Instead, it targets a single subset of cognitive abilities and advances in that spectrum. Siri, Amazon's Alexa, and IBM Watson are examples of Narrow AI. Even OpenAI's ChatGPT is considered a form of Narrow AI because it's limited to the single task of text-based chat [2].

4.B General AI

Artificial General Intelligence (AGI), also known as Strong AI, is today nothing more than a theoretical concept. AGI can use previous learnings and skills to accomplish new tasks in a different context without the need for human beings to train the underlying models. This ability allows AGI to learn and perform any intellectual task that a human being can [2].

4.C Super AI

Super AI is commonly referred to as artificial superintelligence and, like AGI, is strictly theoretical. If ever realized, Super AI would think, reason, learn, make judgements, and possess cognitive abilities that surpass those of human beings. The applications possessing Super AI capabilities will have evolved beyond the point of understanding human sentiments and experiences to feel emotions, have needs and possess beliefs and desires of their own [2].

5. Based on Technologies

5.A Machine Learning

Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior. Artificial intelligence systems are used to perform complex tasks in a way that is similar to how humans solve problems.

There are different types of machine learning algorithms, but the most common are regression and classification algorithms. Regression algorithms are used to predict outcomes, while classification algorithms are used to identify patterns and group data.

Machine learning algorithms can be further divided into two categories: supervised and unsupervised. Supervised algorithms require a training dataset that includes both the input data and the desired output. Unsupervised algorithms do not require a training dataset, and instead rely on data to "learn" on its own [2].

5.B Deep learning

Deep learning is another subset of AI, and more specifically, a subset of machine learning. It has received a lot of attention in recent years because of the successes of deep learning networks in tasks such as computer vision, speech recognition, and self-driving cars.

Deep learning networks are composed of layers of interconnected processing nodes, or neurons. The first layer, or the input layer, receives input from the outside world, such as an image or a sentence. The next layer processes the input and passes it on to the next layer, and so on. These intermediate layers are often referred to as hidden layers [2].

5.C Natural Language Processing

Natural language processing (NLP) involves teaching computers to understand and produce written and spoken language in a similar manner as humans. NLP combines computer science, linguistics, machine learning and deep learning concepts to help computers analyze unstructured text or voice data and extract relevant information from it. NLP mainly tackles speech recognition and natural language generation, and it's leveraged for use cases like spam detection and virtual assistants [1].

5.D Computer vision

Narrow AI applications with computer vision can be trained to interpret and analyze the visual world. This allows intelligent machines to identify and classify objects within images and video footage.

Applications of computer vision include:

- Image recognition and classification
- Object detection
- Object tracking
- Facial recognition
- Content-based image retrieval

Computer vision is critical for use cases that involve AI machines interacting and traversing the physical world around them. Examples include self-driving cars and machines navigating warehouses and other environments [2].

5.E Robotics

Robots in industrial settings can use Narrow AI to perform routine, repetitive tasks that involve materials handling, assembly, and quality inspections. In healthcare, robots equipped with Narrow AI can assist surgeons in monitoring vitals and detecting potential issues during procedures. Agricultural machines can engage in autonomous pruning, moving, thinning, seeding, and spraying. And smart home devices such as the iRobot Roomba can navigate a home's interior using computer vision and use data stored in memory to understand its progress [2].

5.F Expert systems

Expert systems equipped with Narrow AI capabilities can be trained on a corpus to emulate the human decision-making process and apply expertise to solve complex problems. These systems can evaluate vast amounts of data to uncover trends and patterns to make decisions. They can also help businesses predict future events and understand why past events occurred [2].

Chapter 2. Literature review

1. Medical Robots

Medical robotics is causing a paradigm shift in therapy. The most widespread surgical robot, Intuitive Surgical's da Vinci system, has been discussed in over 4,000 peer-reviewed publications, was cleared by the United States' Food and Drug Administration (FDA) for multiple categories of operations, and was used in 80% of radical prostatectomies performed in the U.S. for 2008, just nine years after the system went on the market.

The rapid growth in medical robotics is driven by a combination of technological improvements (motors, materials, and control theory), advances in medical imaging (higher resolutions, magnetic resonance imaging, and 3D ultrasound), and an increase in surgeon/patient acceptance of both laparoscopic procedures and robotic assistance. New uses for medical robots are created regularly, as in the initial stages of any technology-driven revolution.

In 1979, the Robot Institute of America, an industrial trade group, defined a robot as "a reprogrammable, multi-functional manipulator designed to move materials, parts, tools, or other specialized devices through various programmed motions for the performance of a variety of tasks." Such a definition leaves out tools with a single task (e.g., stapler), anything that cannot move (e.g., image analysis algorithms), and nonprogrammable mechanisms (e.g., purely manual laparoscopic tools).

As a result, robots are generally indicated for tasks requiring programmable motions, particularly where those motions should be quick, strong, precise, accurate, untiring, and/or via complex articulations.

The downsides generally include high expense, space needs, and extensive user training requirements.

The greatest impact of medical robots has been in surgeries, both radiosurgery and tissue manipulation in the operating room, which are improved by precise and accurate motions of the necessary tools. Through robot assistance, surgical outcomes can be improved, patient trauma can be reduced, and hospital stays can be shortened, though the effects of robot assistance on long-term results are still under investigation.

Medical robots have been reviewed in various papers since the 1990s. Many such reviews are domain specific, for example, focusing on surgical robots, urological robots, spine robots, and so forth.

2. What can AI do to help diagnose diseases?

AI can play a significant role in diagnosing diseases by leveraging its ability to analyze large datasets, recognize patterns, and process information **at speeds beyond human capability**.

Here are several ways AI can assist in disease diagnosis:

2.A Medical Imaging Analysis:

Radiology: AI algorithms can analyze medical images, such as X-rays, MRIs, and CT scans, to detect abnormalities or potential signs of diseases. For example, AI can help identify tumors, fractures, or abnormalities in organs.

2.B Clinical Data Analysis:

AI can analyze patient data from electronic health records (EHRs) to identify patterns and correlations that may be indicative of specific diseases. This includes information such as symptoms, medical history, lab results, and vital signs.

2.C Diagnostic Decision Support:

AI can provide decision support to healthcare professionals by analyzing diagnostic data and offering suggestions or differential diagnoses. This assists doctors in making more informed decisions and reduces the risk of misdiagnosis.

2.D Screening Programs:

AI can be used in population-scale screening programs, analyzing large datasets to identify individuals at risk for certain diseases. This can be particularly useful for conditions with genetic predispositions or lifestyle-related factors.

2.E Natural Language Processing (NLP):

NLP algorithms can extract valuable information from unstructured clinical notes, research papers, and other textual sources, contributing to a more comprehensive understanding of a patient's health status.

3. What is the need of using AI Robots?

The use of AI robots in hospitals serves several important needs, contributing to enhanced efficiency, patient care, and overall healthcare management.

3.A Automation of Routine Tasks

AI robots can handle routine and repetitive tasks, such as delivering medications, transporting lab samples, or moving equipment. This allows healthcare staff to focus on more complex and patient-centered activities, improving overall workflow efficiency. (so that gives enough time for doctors to focus on the patients and learning)

3.B Patient Monitoring and Engagement

AI robots can be employed for patient monitoring, checking vital signs, and providing medication reminders. They can also engage with patients through interactive interfaces, offering information, entertainment, and companionship, especially for long-term hospital stays.

3.C Efficient Data Collection and Analysis

AI robots can collect and analyze large volumes of healthcare data efficiently. This data can be used for predictive analytics, identifying patterns and trends that may inform better decision-making for healthcare administrators and providers.

3.D Helps med student in learning process

AI robots could be programmed to examen med students on the Data collected by patients.

4. The Medical Segmentation Decathlon

The challenge comprised different target regions, modalities and challenging characteristics and was separated into seven known tasks (blue; the development phase: brain, heart, hippocampus, liver, lung, pancreas, prostate) and three mystery tasks (gray; the mystery phase: colon, hepatic vessels, spleen). MRI magnetic resonance imaging, MP-MRI multiparametric-magnetic resonance imaging, CT computed tomography. The following figure is an Overview of the ten different tasks of the Medical Segmentation Decathlon (MSD).

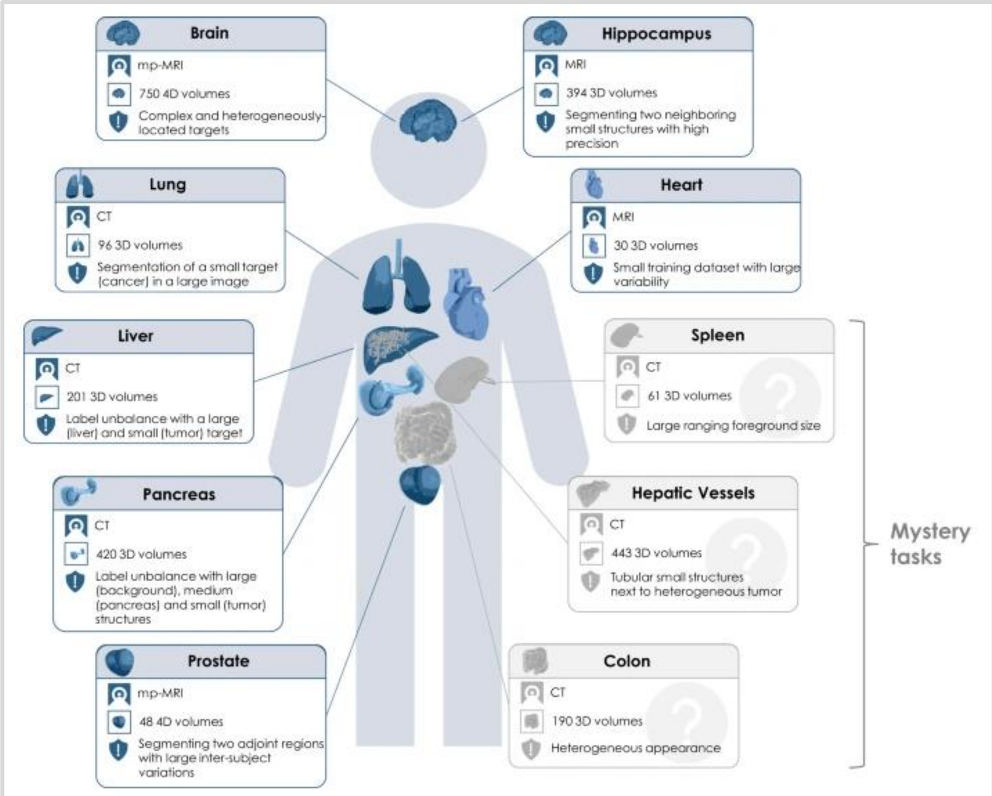


Figure 1 Medical Segmentation

5. Cardiovascular disease

Cardiovascular disease (CVD) is a general term for conditions affecting the heart or blood vessels.

It's usually associated with a build-up of fatty deposits inside the arteries (**atherosclerosis**) and an increased risk of blood clots.

It can also be associated with damage to arteries in organs such as the brain, heart, kidneys, and eyes.

CVD is one of the main causes of death and disability in the UK, but it can often largely be prevented by leading a healthy lifestyle [3].

There are many different types of CVD:

5.A Coronary heart disease

Coronary heart disease occurs when the flow of oxygen-rich blood to the heart muscle is blocked or reduced. This puts an increased strain on the heart, and can lead to:

- **Angina:** chest pain caused by restricted blood flow to the heart muscle.
- **Heart attacks:** where the blood flow to the heart muscle is suddenly blocked.
- **Heart failure:** where the heart is unable to pump blood around the body properly.

5.B Strokes and TIAs

A stroke is where the blood supply to part of the brain is cut off, which can cause brain damage and possibly death. A transient ischaemic attack (also called a TIA or mini stroke) is similar, but the blood flow to the brain is only temporarily disrupted. The main symptoms of a stroke or TIA can be remembered with the word FAST, which stands for:

- **Face:** the face may have drooped on one side, the person may be unable to smile, or their mouth or eye may have dropped.
- **Arms:** the person may not be able to lift both arms and keep them there because of arm weakness or numbness in one arm.
- **Speech:** their speech may be slurred or garbled, they may not be able to talk at all or they may not be able to understand what you are saying to them.
- **Time:** it's time to dial 999 immediately if you see any of these signs or symptoms.

5.C Peripheral arterial disease

Peripheral arterial disease occurs when there's a blockage in the arteries to the limbs, usually the legs. This can cause:

- dull or cramping leg pain, which is worse when walking and gets better with rest
- hair loss on the legs and feet
- numbness or weakness in the legs

- persistent ulcers (open sores) on the feet and legs

5.D Aortic disease

Aortic diseases are a group of conditions affecting the aorta. This is the largest blood vessel in the body, which carries blood from the heart to the rest of the body. One of most common aortic diseases is an aortic aneurysm, where the aorta becomes weakened and bulges outwards. This doesn't usually have any symptoms, but there's a chance it could burst and cause life-threatening bleeding.

6. Causes of CVD

The exact cause of CVD isn't clear, but there are lots of things that can increase your risk of getting it. These are called "risk factors [3].

The more risk factors you have, the greater your chances of developing CVD. If you're over 40, you'll be invited by your GP for an NHS Health Check every 5 years. Part of this check involves assessing your individual CVD risk and advising you how to reduce it, if necessary, the following are most CVD risk factors:

6.A High blood pressure

High blood pressure (hypertension) is one of the most important risk factors for CVD. If your blood pressure is too high, it can damage your blood vessels.

6.B Smoking

Smoking and other tobacco use is also a significant risk factor for CVD. The harmful substances in tobacco can damage and narrow your blood vessels.

6.C High cholesterol

Cholesterol is a fatty substance found in the blood. If you have high cholesterol, it can cause your blood vessels to narrow and increase your risk of developing a blood clot.

6.D Diabetes

Diabetes is a lifelong condition that causes your blood sugar level to become too high. High blood sugar levels can damage the blood vessels, making them more likely to become narrowed. Many people with type 2 diabetes are also overweight or obese, which is also a risk factor for CVD.

6.E Inactivity

If you don't exercise regularly, it's more likely that you'll have high blood pressure, high cholesterol levels and be overweight. All of these are risk factors for CVD. Exercising regularly will help keep your heart healthy. When combined with a healthy diet, exercise can also help you maintain a healthy weight.

6.F Being overweight or obese

Being overweight or obese increases your risk of developing diabetes and high blood pressure, both of which are risk factors for CVD. You're at an increased risk of CVD if:

- your body mass index (BMI) is 25 or above use the BMI healthy weight calculator to work out your BMI.
- you're a man with a waist measurement of 94cm (about 37 inches) or more, or a woman with a waist measurement of 80cm (about 31.5 inches) or more.

6.G Family history of CVD

If you have a family history of CVD, your risk of developing it is also increased. You're considered to have a family history of CVD if either:

- your father or brother were diagnosed with CVD before they were 55
- your mother or sister were diagnosed with CVD before they were 65

Tell your doctor or nurse if you have a family history of CVD. They may suggest checking your blood pressure and cholesterol level.

6.H Ethnic background

In the UK people of south Asian and Black African or African Caribbean background have an increased risk of getting CVD. This is because people from these backgrounds are more likely to have other risk factors for CVD, such as high blood pressure or type 2 diabetes.

Other factors that affect your risk of developing CVD include:

- **Age:** CVD is most common in people over 50 and your risk of developing it increases as you get older
- **Gender:** men are more likely to develop CVD at an earlier age than women
- **Diet:** an unhealthy diet can lead to high cholesterol and high blood pressure
- **Alcohol:** excessive alcohol consumption can also increase your cholesterol and blood pressure levels, and contribute to weight gain.

7. The Study of Machine Learning

Machine learning research generally falls into three aspects: machine learning methods, machine learning theory and machine learning application, which have different focuses. The research of machine learning methods aims at developing new learning methods. The study of machine learning theory explores the effectiveness and efficiency of machine learning methods and the fundamental theoretical issues of machine learning, yet the research of machine learning applications is concerned with applying machine learning methods to practical problems for problem-solving [4].

8. The Importance of Machine Learning

The last two decades have witnessed a remarkable development of machine learning in theory and application. Many significant breakthroughs have been marked by the successful application of machine learning in artificial intelligence, pattern recognition, data mining, natural language processing, speech processing, computational vision, information retrieval, biological information, and many other computer application fields. It has become the core technology in these areas. Furthermore, it is convinced that machine learning will play an increasingly crucial role in future scientific development and technology applications [4].

The importance of machine learning in science and technology is mainly reflected in the following aspects:

1. Machine learning is an effective way of massive data processing. We live in an era of information explosion, where the processing and utilization of massive data is an inevitable demand. Data, in reality, is not only large in scale but also

often uncertain. Machine learning tends to be the most powerful tool for processing this type of data.

2. Machine learning is an effective means of computer intelligentization. Intelligentization is the inevitable trend of computer development as well as the primary goal of computer technology research and development. In recent decades, research in artificial intelligence and other fields has proved that the application of machine learning in imitating human intelligence is the most effective means despite certain limitations.
3. Machine learning is a crucial component of the development of computer science. It can be considered that computer science is composed of three dimensions: system, computation, and information. Machine learning mainly belongs to the information dimension and plays a central role.

9. The most prominent ML projects in healthcare

9.A Digital Pathology Detection

PathAI: offers robust AI-powered products for pathology assessment across research and clinical development. Their solutions are for the accurate diagnosis of diseases from tissue samples, including cancer research, early detection, and grading [5].

9.B Medical Imaging Analytics

Aidoc: delivers a powerful platform for the analysis of medical images, including radiology scans, CT scans, and X-rays, to assist radiologists in identifying critical findings and abnormalities [6].

9.C Healthcare Operations Automation

Bautomate: offers a business intelligence platform for healthcare operations that aims to automate routine tasks [7].

9.D Genomic Medicine and Drug Discovery Process

Deep Genomics: leverages revolutions in AI, RNA biology, and automation to enable a new approach to drug discovery and development. Their AI Platform untangles the complexity of RNA biology, identifies novel targets, and evaluates thousands of possibilities to identify the best therapeutic candidates [8].

9.E Predictive Analytics and Diagnostic

Prognos Health: leverages AI and Big Data analysis to generate insights from clinical laboratory data, supporting early diagnosis, risk stratification, and treatment optimization [9].

9.F Biomedical Informatics

BenchSci: offers the disease biology generative AI platform for pharmaceutical companies' R&D that accelerates biomedical research and drug discovery by analyzing scientific literature, experimental data, and molecular pathways. The core of the solution is ASCEND, powered by their evidence engine, built from ontologies and multi-modal AI to understand the entire history of biomedical experiments and clinical trials with explainability and domain expertise [10].

9.G Acute Stroke Management

Viz.ai: offers an AI-based solution for the quick identification and triage of acute stroke cases on medical imaging, enabling timely intervention and treatment. The solution helps to significantly save time and improve patient and economic outcomes for organizations [11].

9.H Patient Care Coordination

Innovaccer: offers an AI-enabled patient care coordination platform that integrates data from multiple sources, including EHRs, claims data, and wearables, to provide a comprehensive view of patient health and already helping 96,000+ medical professionals. The solutions help to unify Health Plans, DMEs, Claims, EHRs, Labs, Pharmacy, and SDoH into unified patient records [12].

9.I Medical Imaging Interpretation and Diagnostic Support

Caption Care: delivers AI-driven medical imaging software that enables clinicians to perform and interpret ultrasound exams more accurately and efficiently [13].

9.J Remote Patient Monitoring

Current Health: offers a remote patient monitoring platform powered by AI and predictive analytics, enabling proactive Healthcare interventions and care coordination. The platform offers Healthcare providers services like Virtual Wards, Transitional Care, and Chronic Care, while pharmaceutical companies can benefit from Decentralized Clinical Trials and Commercial Deployment [14].

Chapter 3. How does Artificial Intelligence work?

The aim of AI science is to create a computer system which can model human behavior in such a way that it can be thus used for solving complex issues with the help of human-like thinking processes. AI systems work by merging bulky sets of data with intelligent processing algorithms and run through multiple tasks extremely quickly in very little time.

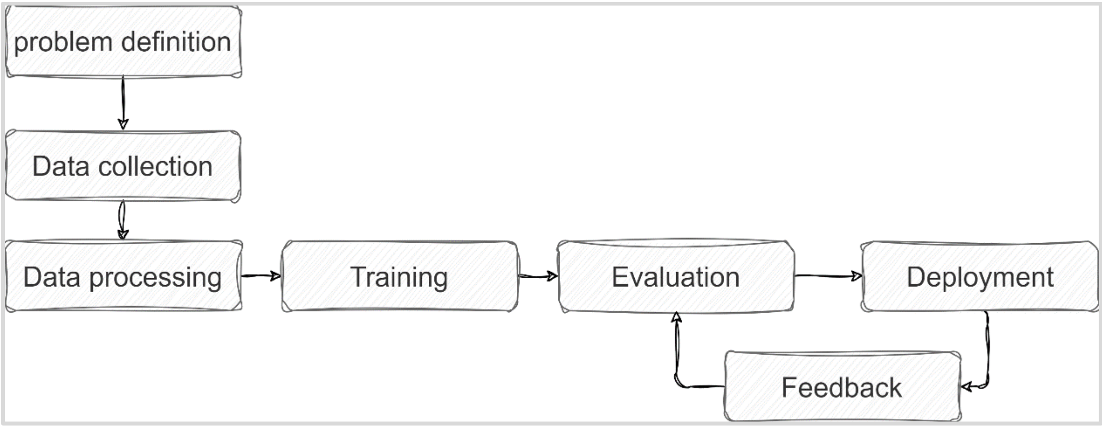


Figure 2 Machine Learning model Lifecycle

Artificial intelligence systems work by using algorithms and data. First, a massive amount of data is collected and applied to mathematical models, or algorithms, which use the information to recognize patterns and make predictions in a process known as training. Once algorithms have been trained, they are deployed within various applications, where they continuously learn from and adapt to new data. This allows AI systems to perform complex tasks like image recognition, language processing and data analysis with greater accuracy and efficiency over time [1].

The diagram we reviewed outlined the typical stages of an ML project, starting with defining a problem. The following headlines will delve into the specific problem we're tackling and then explore the steps involved in building and deploying a successful ML model to address it [15].

1. Problem statement

1.A Chest radiography

The most used imaging test worldwide is chest radiography, which is essential for the early detection, diagnosis, and treatment of many serious illnesses. Chest illnesses are a major public health concern in the country. Radiology is currently a growing field in Egypt. Egypt has evolved over the years, with the introduction of new technologies and advanced imaging techniques. However, there are still challenges that need to be addressed, such as the shortage of trained radiologists. which is insufficient to service the entire country's population. the lack of radiologists is extremely severe, making it challenging for patients to get diagnostic imaging services. The availability of skilled radiologists is essential for the difficult process of identifying various chest ailments from X-rays.

1.B Cardiovascular diseases

In Egypt, cardiovascular diseases (CVDs) pose significant health risks, often remaining undetected until symptoms become severe. Establishing effective predictive models for CVDs necessitates addressing several challenges, including limited awareness of cardiovascular risk factors and reluctance to seek medical attention until symptoms become debilitating. Developing user-friendly and accessible predictive tools can empower individuals to proactively assess their cardiovascular health and seek timely interventions. By bridging the gap between awareness and action, such models hold promise for improving heart health outcomes and reducing the burden of CVDs in the population.

2. basics of data in ML

Data is generally divided into two categories in ML: labeled and unlabeled.

2.A Labeled data

Is given tags (or labels) that indicate the desired output for the ML model to predict. This type of data plays a role in supervised training, where the model is taught to make predictions based on input information. For instance, in a dataset aimed at identifying spam emails, messages would be marked as either "spam" or "not spam" to later train the models to sort emails correctly [16].

2.B Unlabeled data

As you can tell from its name, lacks associated labels. It's used in unsupervised learning scenarios where the model seeks to find patterns and structures within the data without instructions on what to predict. Unlabeled data is more abundant and simpler to gather compared to labeled data but poses challenges when it comes to deriving meaningful insights without predefined guidance [16].

3. Data Acquisition

Data acquisition is one of the most important steps in a machine learning algorithm. It's used to collect data on how your model performs on new datasets.

Data acquisition is simply collecting new data and transforming it into a format your machine learning algorithm can use. Once you've acquired some training data, your model can learn from it and improve its performance on new tasks [17].

4. Why do we need Data Acquisition?

For most machine learning algorithms, you need to acquire training data before using them for prediction. This training data can be provided by humans or other machines (e.g., from web scraping). The goal is to have a large enough sample size that your model can learn from effectively but not so large that it takes too much time to train (and possibly overfit) the available data [17].

Access to authorized open databases is essential for conducting research in data science and machine learning. Here are some reputable open-access databases and resources:

4.A General Data Science and Machine Learning Datasets

- Kaggle Datasets: Kaggle offers a vast collection of datasets across various domains that are open for use.
- UCI Machine Learning Repository: The University of California, Irvine provides a collection of datasets specifically for machine learning applications.
- Google Dataset Search: A tool provided by Google to find datasets stored across the web. Google Dataset Search
- Data.gov: A comprehensive repository of datasets provided by the US government, covering various sectors.

4.B Healthcare-Specific Datasets

- MIMIC-III (Medical Information Mart for Intensive Care): A freely accessible critical care database containing de-identified health-related data. MIMIC-III
- HealthData.gov: A collection of datasets on a wide array of health-related topics provided by the US Department of Health & Human Services.
- OpenfMRI: An open repository of functional magnetic resonance imaging (fMRI) datasets.

The dataset used for developing the heart attack risk prediction model was sourced from Kaggle, specifically the 'Heart Attack Risk Prediction Dataset' by SOURAV BANERJEE [18].

5. Dataset Glossary (Column-wise)

- Patient ID: Unique identifier for each patient
- Age: Age of the patient
- Sex: Gender of the patient (Male/Female)
- Cholesterol: Cholesterol levels of the patient
- Blood Pressure: Blood pressure of the patient (systolic/diastolic)
- Heart Rate: Heart rate of the patient

- Diabetes: Whether the patient has diabetes (Yes/No)
- Family History: Family history of heart-related problems (1: Yes, 0: No)
- Smoking: Smoking status of the patient (1: Smoker, 0: Non-smoker)
- Obesity: Obesity status of the patient (1: Obese, 0: Not obese)
- Alcohol Consumption: Level of alcohol consumption by the patient (None/Light/Moderate/Heavy)
- Exercise Hours Per Week: Number of exercise hours per week
- Diet: Dietary habits of the patient (Healthy/Average/Unhealthy)
- Previous Heart Problems: Previous heart problems of the patient (1: Yes, 0: No)
- Medication Use: Medication usage by the patient (1: Yes, 0: No)
- Stress Level: Stress level reported by the patient (1-10)
- Sedentary Hours Per Day: Hours of sedentary activity per day
- Income: Income level of the patient
- BMI: Body Mass Index (BMI) of the patient
- Triglycerides: Triglyceride levels of the patient
- Physical Activity Days Per Week: Days of physical activity per week
- Sleep Hours Per Day: Hours of sleep per day
- Country: Country of the patient
- Continent: Continent where the patient resides
- Hemisphere: Hemisphere where the patient resides
- Heart Attack Risk: Presence of heart attack risk (1: Yes, 0: No)

6. What is data preprocessing?

A considerable chunk of any data-related project is about data preprocessing and data scientists spend around 80% of their time on preparing and managing data. Data preprocessing is the method of analyzing, filtering, transforming, and encoding data so that a machine learning algorithm can understand and work with the processed output [19].

7. Why is data preprocessing necessary?

Algorithms that learn from data are simply statistical equations operating on values from the database. So, as the popular saying goes, “if garbage goes in, garbage comes out”. Your data project can only be successful if the data going into the machines, is high quality.

In data extracted from real-world scenarios, there’s always noise and missing values. This happens due to manual errors, unexpected events, technical issues, or a variety of other obstacles. Incomplete and noisy data can’t be consumed by algorithms, because they’re usually not designed to handle missing values, and the noise causes disruption in the true pattern of the sample.

Data preprocessing aims to solve these problems by thorough treatment of the data at hand [19].

8. Tools and Libraries

Data preprocessing steps can be simplified through tools and libraries that make the process easier to manage and execute. Without certain libraries, one liner solutions can take hours of coding to develop and optimize.

Data Preprocessing with Python: Python is a programming language that supports countless open-source libraries that can compute complex operations with a single line of code. For instance, for the smart imputation of missing values, one needs only use Scikit-Learn’s impute library package. Or, for scaling datasets, just call upon the MinMaxScaler function from the preprocessing library package. There are countless data preprocessing functions available in the preprocessing library package [19].

Figure 3 Python and most common AI packages

9. Purpose of data preprocessing

After you have properly gathered the data, it needs to be explored, or assessed, to spot key trends and inconsistencies [19].

The main goals of Data Quality Assessment are:

9.A Get Data Overview

Understanding the data formats and overall structure is crucial. This involves:

- Data Formats: Reviewing how data is stored (e.g., CSV, JSON).
- Data Structure: Examining the organization of data, such as rows and columns in a table.
- Statistical Properties: Calculating and understanding key statistics like mean, median, quantiles, and standard deviation. These details can help identify irregularities and provide insights into data distribution.

9.B Identify Missing Data

Missing data is common in most real-world datasets and can disrupt true data patterns. Steps include:

- Detection: Identifying missing values within the dataset.

- Impact Assessment: Understanding how missing data can affect analysis and predictions.
- Handling Missing Data: Employing strategies like imputation (replacing missing values with mean/median) or using algorithms that handle missing values natively.

9.C Identify Outliers or Anomalous Data

Some data points may deviate significantly from the overall pattern. Handling these outliers involves:

- Detection: Using statistical methods or visualization tools (e.g., box plots) to spot outliers.
- Assessment: Determining if outliers are due to errors or natural variations.
- Action: Deciding whether to remove, transform, or keep outliers based on their impact on model performance.

9.D Remove Inconsistencies

Real-world data often contains inconsistencies that need correction, including:

- Incorrect Spellings: Standardizing text data to fix typos.
- Misplaced Values: Correcting values entered in the wrong columns.
- Duplicated Data: Identifying and removing duplicate records to avoid skewing results.
- Manual Checks: While some inconsistencies can be resolved through automated scripts, others may require manual review to ensure accuracy.

10. popular data pre-processing techniques

10.A Handling missing values

Missing values are a recurrent problem in real-world datasets because real-life data has physical and manual limitations. For example, if data is captured by sensors from a particular source, the sensor might stop working for a while, leading to missing data. Similarly, different datasets have different issues that cause missing data points [19].

- **Drop samples with missing values:** this is instrumental when both the number of samples is high, and the count of missing values in one row/sample is high. This is not a recommended solution for other cases since it leads to heavy data loss.
- **Replace missing values with zero:** sometimes this technique works for basic datasets, since the data in question assumes zero as a base number, signifying that the value is absent. However, in most cases, zero can signify a value in itself. For example, if a sensor generates temperature values and the dataset belongs to a tropical region. Similarly, in most cases, if missing values are populated with 0, then it would be misleading to the model. 0 can be used as replacement only when the dataset is independent of its effect. For example, in phone bill data, a missing value in the billed amount column can be replaced by zero, since it might indicate that the user didn't subscribe to the plan that month.
- **Replace missing value with mean, median or mode:** you can deal with the above problem, resulting from using 0 incorrectly, by using statistical functions like mean, median or mode as a replacement for missing values. Even though they're also assumptions, these values make more sense and are closer approximations when compared to one single value like 0.

10.B Scaling

Different columns can be present in different ranges. For example, there can be a column with a unit of distance, and another with the unit of a currency. These two columns will have starkly different ranges, making it difficult for any machine learning model to reach an optimal computation state [19]. Some popular scaling techniques are:

- Min-Max Scaler: min-max scaler shrinks the feature values between any range of choice. For example, between 0 and 5.
- Standard Scaler: a standard scaler assumes that the variable is normally distributed and then scales it down so that the standard deviation is 1 and the distribution is centered at 0.

- Robust Scaler: robust scaler works best when there are outliers in the dataset. It scales the data with respect to the inter-quartile range after removing the median.
- Max-Abs Scaler: like min-max scaler, but instead of a given range, the feature is scaled to its maximum absolute value. The sparsity of the data is preserved since it does not center the data.

10.C Outlier treatment

Outliers are data points lying far away from the majority of other data points, outliers in the data that is not normally distributed do not require identification. As most statistical tests assume that data are normally distributed [19].

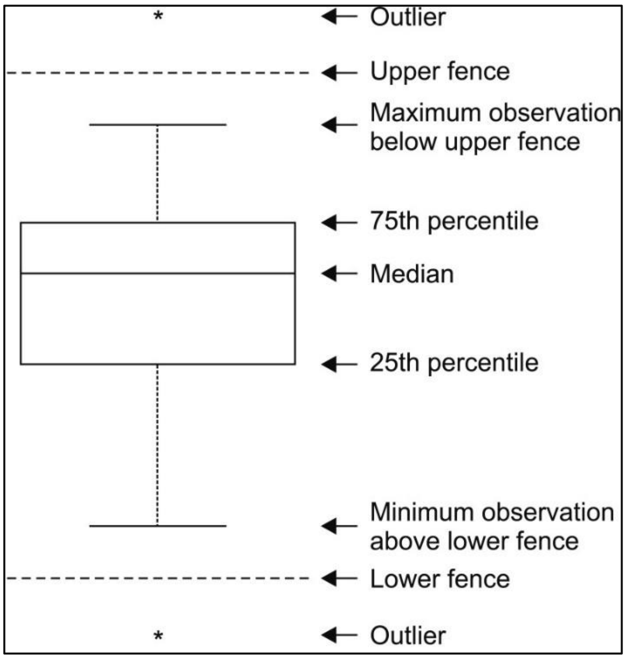


Figure 4 Box plots can be used to identify the outliers

Outlier identification should precede data analysis. Different methods are used to identify outliers in a normal distribution. One of the methods measures the distance between a data point and the center of all data points to determine an outlier. Based on this method, the data points that do not fall within three SD of the mean are identified as outliers. However, this method is not considered appropriate because the mean and SD are statistically sensitive to the presence of outliers. Alternatively, the median and quartile range are more useful because these statistics are less sensitive to outliers.

10.D Feature encoding

Sometimes, data is in a format that can't be processed by machines. For instance, a column with string values, like names, will mean nothing to a model that depends only on numbers. So, we need to process the data to help the model interpret it. This method is called categorical encoding. But before we dive into different encoding methods, let's take a step back. Understanding the type of categorical data, you're dealing with is crucial.

Understanding whether it's nominal or ordinal is crucial because it determines the types of statistical analyses that can be performed on the data. For nominal data, you can perform frequency counts and measures of central tendency like mode, while for ordinal data, you can also perform median and quartiles calculations in addition to what you can do with nominal data [19].

D.1 One-Hot Encoding

One-Hot Encoding takes a single categorical column and creates a separate binary column for each of the variables contained within. This works best with nominal data when categories have no identifiable order. However, when you have a large number of unique categorical variables, you will need to be careful using this technique as it could lead to a high dimensional representation, making the data more sparse and, therefore, more difficult to find patterns. This could lead to poor generalizations and risks of overfitting. The below example shows how the column 'Furniture' is split into binary columns for each individual item [20].

ID	Furniture
1	Chair
2	Table
3	Desk
4	Sofa
5	Sofa

Figure 5 Input data for OneHotEncoder example

ID	Chair	Desk	Sofa	Table
1	1	0	0	0
2	0	0	0	1
3	0	1	0	0
4	0	0	1	0
5	0	0	1	0

Figure 6 Final output after OneHotEncoder

D.2 Label Encoding

Label Encoding assigns a unique number to each category. This technique assumes an ordinal relationship between the categories, assuming a ranking or order to the categories. Education levels show a good use case for this technique due to the natural progression order. However, if we were to use this on our furniture example from One-Hot Encoding, there is no meaningful order; this would likely lead to incorrect assumptions in the machine learning model. The below example shows education levels where the encoded number increases depending on the level obtained [20].

ID	Categorical Data
1	GCSE
2	A Level
3	Bachelor's Degree
4	Master's Degree
5	PHD
6	A Level
7	Bachelor's Degree

Figure 7 Input data for Label Encoding example

ID	Categorical Data	Label Encoded
1	GCSE	1
2	A Level	2
3	Bachelor's Degree	3
4	Master's Degree	4
5	PHD	5
6	A Level	2
7	Bachelor's Degree	3

Figure 8 Output data after Label Encoding

D.3 Frequency Encoding

Frequency Encoding assigns a value based on a count of a category's occurrence in the data set. This technique should be used with caution, especially when the frequency of the categories is imbalanced. However, this can be resolved by normalizing the values after encoding. The below example shows the encoded frequency based on a color column [20].

ID	Color
1	Green
2	Green
3	Red
4	Purple

Figure 9 Input data for Frequency Encoding example

ID	Color	Frequency Encoding
1	Green	2
2	Green	2
3	Red	1
4	Purple	1

Figure 10 Output data after Frequency Encoding

Chapter 4. Project design

1. Project Workflow Diagram

To provide a comprehensive overview of the project, we begin with the project workflow diagram. This diagram illustrates the contributions from various departments and outlines the step-by-step processes involved in the development and implementation of the Robot.

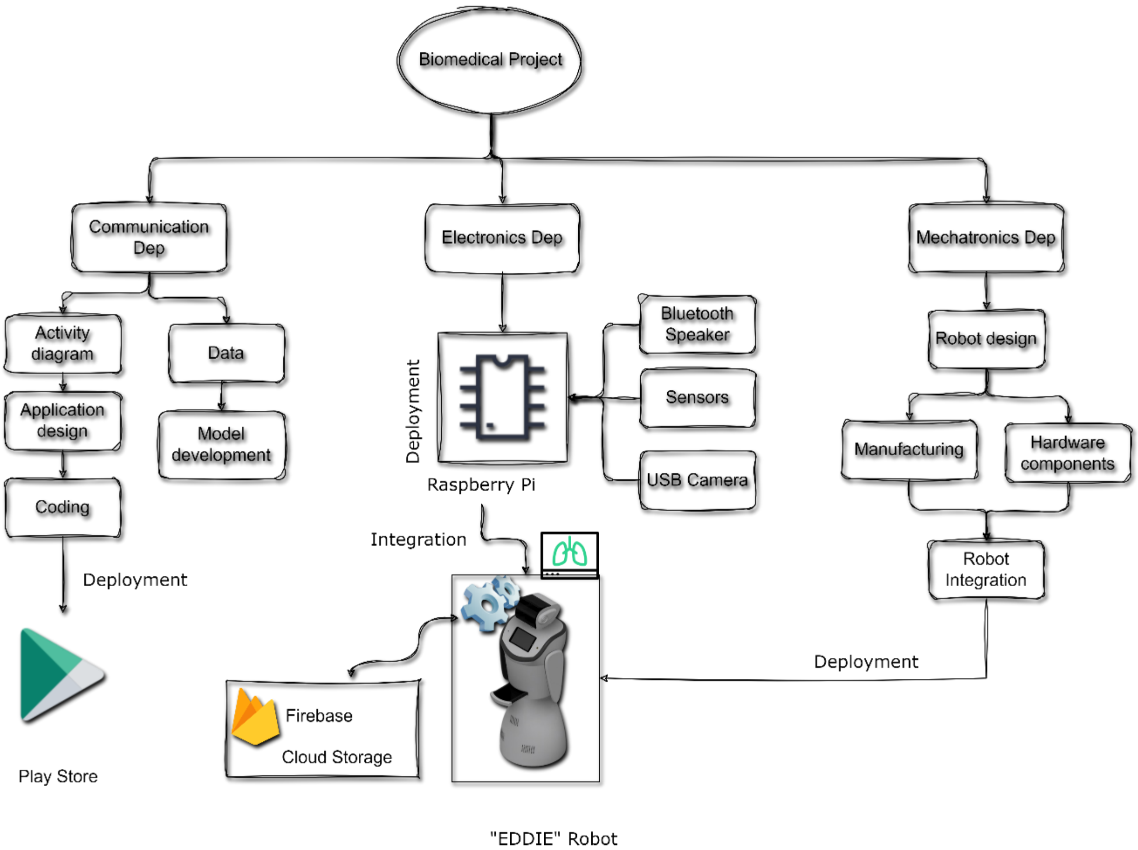


Figure 11 Project Workflow Diagram

The workflow diagram above provides a visual representation of the entire project lifecycle. It highlights the integrated efforts of different departments, ensuring a seamless development process.

2. Mechatronics Department

Responsible for designing and manufacturing the robotic framework, ensuring mechanical components are robust and reliable for medical applications.

2.A Introduction about nursing robot

In an era defined by rapid technological advancements and an ever-growing demand for high-quality healthcare services, the integration of robotics into the medical field stands as a transformative frontier. At the forefront of this technological revolution lies the emergence of nursing robots, sophisticated marvels of engineering and artificial intelligence designed to revolutionize patient care delivery and redefine the roles of healthcare professionals.

Nursing robots represent the culmination of decades of research, innovation, and collaboration between experts in robotics, healthcare, and AI. These cutting-edge machines embody the fusion of precision engineering, intricate sensor systems, and adaptive algorithms, all orchestrated to augment and enhance the capabilities of healthcare providers while optimizing patient outcomes. As stalwart allies in the modern healthcare ecosystem, nursing robots embody the ethos of efficiency, accuracy, and compassion, heralding a new era where technology and humanity converge to redefine the boundaries of medical care.

Stepping into the bustling corridors of hospitals, clinics, and long-term care facilities, nursing robots stand poised to revolutionize every facet of patient care delivery. From the moment a patient is admitted to their journey towards recovery, these tireless mechanical assistants serve as vigilant guardians, monitoring vital signs, administering medications, and providing invaluable support to healthcare professionals every step of the way. With their dexterous manipulators and intricate array of sensors, nursing robots navigate the complex labyrinth of healthcare environments with unparalleled precision, seamlessly integrating into interdisciplinary care teams to provide a seamless continuum of care.

The capabilities of nursing robots transcend the confines of traditional healthcare paradigms, encompassing a diverse array of functions designed to address the multifaceted needs of patients and providers alike. Whether it be the meticulous administration of medications, the gentle assistance with activities of daily living, or the intricate support during surgical procedures, these versatile machines epitomize the epitome of technological sophistication and human-centric design. Through their unwavering dedication and tireless efforts, nursing robots not only alleviate the burden on healthcare professionals but also empower patients to reclaim control over their health and well-being.

Beyond their instrumental role in clinical care delivery, nursing robots also serve as harbingers of profound societal and ethical implications, prompting reflections on the evolving nature of human-machine interaction and the ethical considerations surrounding their integration into healthcare settings. As these machines continue to proliferate and permeate every facet of healthcare delivery, questions regarding autonomy, accountability, and equity loom large, challenging stakeholders to navigate the complex landscape of technological innovation with wisdom and foresight.

In conclusion, the advent of nursing robots represents a watershed moment in the annals of healthcare history, ushering in an era of unprecedented possibilities and unparalleled advancements. As these remarkable machines continue to evolve and mature, they hold the promise of revolutionizing patient care delivery, transcending the confines of conventional practice, and ushering in a future where the union of technology and compassion transforms the landscape of healthcare as we know it.

2.B Medical Nursing robot steps for designing

Designing a nursing robot involves a multifaceted approach that integrates various disciplines such as robotics, healthcare, human factors engineering, and artificial intelligence. Here's a detailed description of the process:

B.1 Needs Assessment and Goal Definition:

The first step is to identify the specific needs and goals for the nursing robot. This involves consulting with healthcare professionals, patients, and other stakeholders to understand the tasks the robot will perform and the challenges it needs to address. For example, robots may need to assist with patient care tasks such as monitoring vital signs, administering medication, or providing companionship to patients.

B.2 Conceptual Design:

Based on the needs assessment, designers create a conceptual design for the nursing robot. This includes defining its physical appearance, functionality, and key features. Designers may use sketches, renderings, and 3D modeling software to visualize the robot's form and function.

Figure 12 nursing robot

B.3 Mechanical Design:

Mechanical engineers then design the robot's physical structure and components. This involves selecting materials, determining dimensions, and designing mechanisms for movement and manipulation. The robot's design must prioritize safety, durability, and ease of maintenance, especially in a healthcare setting.

B.4 Electrical and Electronic Design:

Electrical engineers design the robot's electrical and electronic systems, including sensors, actuators, and control systems. These components enable the robot to perceive its environment, make decisions, and perform tasks autonomously or under human supervision. Power management and energy efficiency are critical considerations, as the robot may need to operate for extended periods without recharging.

B.5 Software Development:

Software developers write the code that enables the robot to perform its intended tasks. This includes programming the robot's behaviors, decision-making algorithms, and user interface. Machine learning and AI techniques may be employed to enable the robot to learn from experience and adapt to different situations over time.

B.6 Human-Robot Interaction Design:

Human factors engineers design the user interface and interaction modalities to ensure that the robot is easy to use and intuitive for both healthcare professionals and patients. This may involve touchscreen interfaces, voice commands, or gestures, depending on the robot's capabilities and the preferences of its users.

B.7 Testing and Iteration:

Throughout the design process, prototypes of the nursing robot are built and tested in simulated healthcare environments. This allows designers to identify and address any issues with performance, safety, or usability. Feedback from healthcare professionals and end-users is used to refine the design through iterative testing cycles.

B.8 Regulatory Compliance:

Designers must ensure that the nursing robot complies with relevant regulatory standards and requirements for medical devices and robotics. This may involve conducting safety assessments, obtaining certifications, and documenting the robot's design and manufacturing processes.

B.9 Manufacturing and Production:

Once the design is finalized and all necessary approvals are obtained, the nursing robot moves into production. This involves sourcing components, manufacturing the robot's physical structure, and assembling the various subsystems. Quality control processes are implemented to ensure that each robot meets the required standards of performance and reliability.

B.10 Deployment and Training:

Finally, the nursing robot is deployed in healthcare facilities where it will be used to assist with patient care. Healthcare professionals receive training on how to interact with the robot and integrate it into their workflow effectively. Ongoing support and maintenance services are provided to ensure that the robot continues to operate smoothly and reliably over time.

Figure 13 nursing robot in healthcare

2.C Introduction of computer-aided Design (CAD)

Computer-Aided Design (CAD) revolutionized the way engineers and designers conceptualize, create, and analyze products and structures. In essence, CAD refers to the use of computer software to facilitate the design process across various industries, including mechanical, architectural, electrical, and civil engineering.

2.D Evolution and History:

CAD emerged in the 1960s and 1970s as a response to the limitations of traditional manual drafting methods. Early CAD systems were primitive compared to modern software but laid the groundwork for the development of more sophisticated tools. Over the decades, CAD has evolved rapidly, driven by advancements in computing power, graphics technology, and user interfaces.

2.E Core Functionality:

CAD software enables engineers and designers to create precise digital representations of objects, parts, assemblies, or structures. Users can manipulate geometric shapes, define dimensions, specify materials, and apply various visual properties such as color and texture. CAD tools offer a range of features for modeling, drafting, rendering, simulation, and analysis, empowering users to explore design alternatives and optimize their solutions.

2.F Types of CAD Software:

There are various types of CAD software tailored to specific industries and applications.

- **2D CAD:** Primarily used for creating 2-dimensional drawings and schematics, such as architectural floor plans or mechanical blueprints.
- **3D CAD:** Allows for the creation of three-dimensional models with depth, enabling users to visualize and manipulate objects in virtual space.
- **Parametric CAD:** Utilizes parameters and constraints to define the relationships between geometric elements, facilitating design modifications and updates.

- **BIM (Building Information Modeling):** Focuses on architectural and construction projects, integrating 3D modeling with data-rich information about building components and materials.
- **CAM (Computer-Aided Manufacturing):** Links CAD software with manufacturing processes, enabling the generation of toolpaths and instructions for CNC machining, 3D printing, or other fabrication methods.

2.G Benefits of CAD:

CAD offers numerous advantages over traditional drafting methods, including:

- Increased productivity and efficiency through automation and parametric Modeling.
- Improved accuracy and precision, reducing errors and rework.
- Enhanced visualization and communication of design concepts.
- Streamlined collaboration and document sharing among team members.
- Integration with simulation and analysis tools for performance evaluation.
- Faster prototyping and product development cycles, accelerating time to market.

2.H Applications:

CAD is widely used across industries for a diverse range of applications, ***including:***

- Product design and development in manufacturing and engineering sectors.
- Architectural and building design for residential, commercial, and infrastructure projects.
- Electrical and electronic circuit design and PCB layout.
- Automotive and aerospace engineering for vehicle and aircraft design.
- Civil engineering for infrastructure planning, road design, and urban development.

2.I Advantages of Humanoid Robot Design for Biomedical Applications

When it comes to designing robots for specific applications, humanlike robots have several advantages, especially in the context of biomedical applications. Here's why human-robot design is often considered the best option for biomedical robots:

I.1 Operability in Human Environments:

Humanlike robots are designed to operate in environments where humans live and work. Their form factor, movement capabilities, and interaction mechanisms are optimized for compatibility with human spaces.

In medical settings, such as hospitals or rehabilitation centers, humanlike robots can navigate corridors, interact with medical staff, and assist patients more effectively.

I.2 Understanding Human Movements:

By developing humanoid robots, researchers gain insights into the complexities of human movement. Studying how these robots walk, grasp objects, and perform tasks helps us understand biomechanics better.

This understanding can lead to improved prosthetic devices for people with injuries or disabilities. Humanoids serve as a bridge between robotics and rehabilitation medicine.

I.3 Social Interaction and Acceptance:

Humanlike robots can engage in social interactions with patients, caregivers, and medical professionals. Their expressive faces, gestures, and natural movements create a sense of familiarity.

For patients, interacting with a humanoid robot can be less intimidating than interacting with a traditional machine. This acceptance is crucial for successful human-robot collaboration.

1.4 Promising Applications:

Humanoids find practical applications in healthcare. For instance:

- Assisting Autistic Children: Studies show that humanoid robots like Nao, Bandit, Kaspar, and RoboKind Zeno positively impact autistic children. These robots can provide social training and emotional support.
- Space Exploration: NASA's Robonaut, a humanoid robot, assists astronauts on the International Space Station. Its humanlike dexterity and adaptability make it valuable for complex tasks.
- Rehabilitation and Physical Therapy: Humanoids can guide patients through exercises, monitor progress, and provide motivation during rehabilitation.
- Surgical Training: Simulating surgical procedures using humanoid robots allows medical students and surgeons to practice in a controlled environment.

3. Electronics Department

Integrates biosensors to collect vital patient data and configures the Raspberry Pi and camera for accurate image detection and data processing.

3.A Electronic Health Records

The availability of digital health data is the basis to apply Machine Learning algorithm for medical applications. The main digital data sources are the electronic health record (EHR) or electronic medical record (EMR). The data elements included in an electronic health/medical record can vary a lot. It can be something as basic as the name and date of birth or more details around for example historical doctor's office visit records including all diagnoses. Here we would like to focus on implementations of EHRs that would allow the use of Machine Learning. This means they must contain information in a digital format. Therefore, several data standards were developed. A well-known international standard is the HL7 (Health Level 7) standard. This standard includes multiple concepts to store images within. Mainly, it recommends storing images under the DICOM (Digital Imaging and Communications in Medicine) standard. A supplement for dermoscopic images shall be included in this standard. The FHIR (Fast Healthcare Interoperable Resources) standard is an implementation of the HL7 Standard [21].

3.B Raspberry Pi

The Raspberry Pi is a single-board computer about the size of a credit card. The Pi runs a version of Linux/android that was customized to work on the ARM processor that drives it. With Linux on board, the functionality of this small device is enhanced that it can be utilized for automating systems. It can work with sensors and external devices and has an inbuilt display and Lan port for communication purposes. The added advantage is that it can be programmed with python.

The Raspberry Pi boards are designed by a subsidiary of the Raspberry Pi Foundation, a charitable organization dedicated to advancing computer science education, and manufactured at a Sony factory in South Wales.

Since its launch, the Pi has been adopted by many schools, and its availability has also coincided with almost tripling the number of people applying to study computer science at Cambridge.

The Raspberry Pi 4 Model B is the latest version of the low-cost Raspberry Pi computer.

Figure 14 Raspberry Pi model 4 B [22]

3.C Features of Raspberry Pi

C.1 CPU

The Raspberry Pi 4 is equipped with a Quad-core ARM Cortex-A72 processor, operating at either 1.5 GHz or 1.8 GHz, depending on the revision of the BCM2711 SoC on the board. The earlier B0 stepping was initially utilized for the Pi 4, which was later replaced with the newer C0-stepped chip starting in mid-2021. The higher clock rate of the C0 stepping is attributed to slightly improved thermals. Additionally, the processor contains 32 KB of data L1 cache, 48 KB of instruction L1 cache, and 1 MB of shared L2 cache.

C.2 GPU

The Broadcom BCM2711 SoC boasts an enhanced GPU compared to previous Raspberry Pi iterations, transitioning from the VideoCore IV clocked at 400 MHz to the VideoCore VI clocked at 500 MHz. Alongside the increased clock rate, the VideoCore VI incorporates its own memory manager, enabling it to access more memory than its predecessor. Moreover, the VideoCore VI is compliant with OpenGL ES 3.1 and Vulkan 1.2 standards.

C.3 RAM

The Raspberry Pi 4 offers various RAM configurations, replacing the 1 GB of LPDDR2 with options for 1 GB, 2 GB, 4 GB, or 8 GB of 3200 MHz LPDDR4. Notably, the 8 GB Pi 4 was released one year after the other models.

C.4 IO and Connectivity

Significant enhancements were made to IO functionality with the Raspberry Pi 4. The total USB bandwidth was upgraded, transitioning from four USB 2.0 ports to two USB 3.0 ports and two USB 2.0 ports. A dedicated ethernet controller on the BCM2711 SoC facilitated the addition of gigabit ethernet. Moreover, the singular full-sized HDMI port on the Raspberry Pi 3 was replaced with dual micro-HDMI connectors. Bluetooth was upgraded from 4.2 on the Raspberry Pi 3 to 5.0 on the Raspberry Pi 4. Additionally, the power supply connector was changed from Micro-USB to USB-C. The ethernet port features Power over Ethernet IEEE 802.3af (802.3at Type 1) compatibility, akin to the Raspberry Pi 3 Model B+.

Figure 15 Raspberry Pi 4 with PoE HAT [23]

3.D Warnings

- This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum1. Any external power supply used with Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors and mice when used in conjunction with Raspberry Pi.
- Where peripherals are connected that do not include the cable or connector, the cable or connector must offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

3.E Safety Instructions

To avoid malfunction or damage to this product please observe the following:

- Do not expose to water, moisture, or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; Raspberry Pi 4 Model B is designed for reliable operation at normal ambient room temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the printed circuit board whilst it is powered and only handle by the edges to minimise the risk of electrostatic discharge damage.

3.F How MAX30100 Pulse Oximeter and Heart Rate Sensor Works?

The MAX30100, or any optical pulse oximeter and heart-rate sensor for that matter, consists of a pair of high-intensity LEDs (RED and IR, both of different wavelengths) and a photodetector. The wavelengths of these LEDs are 660nm and 880nm, respectively.

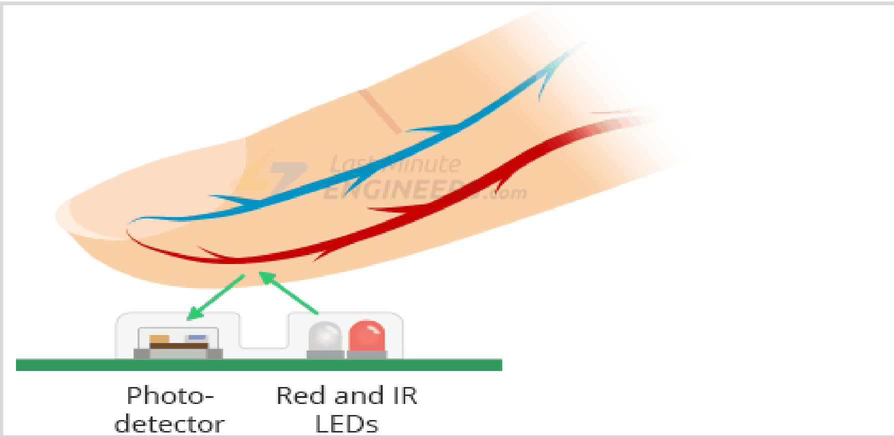


Figure 16 MAX30100-Pulse-Detection-Photoplethysmogram

The MAX30100 works by shining both lights onto the finger or earlobe (or essentially anywhere where the skin isn't too thick, so both lights can easily penetrate the tissue) and measuring the amount of reflected light using a photodetector. This method of pulse detection through light is called Photoplethysmography.

The working of MAX30100 can be divided into two parts: Heart Rate Measurement and Pulse Oximetry (measuring the oxygen level of the blood).

F.1 Heart Rate Measurement

The oxygenated hemoglobin (HbO2) in the arterial blood has the characteristic of absorbing IR light. The redder the blood (the higher the hemoglobin), the more IR light is absorbed. As the blood is pumped through the finger with each heartbeat, the amount of reflected light changes, creating a changing waveform at the output of the photodetector. As you continue to shine light and take photodetector readings, you quickly start to get a heartbeat (HR) pulse reading.

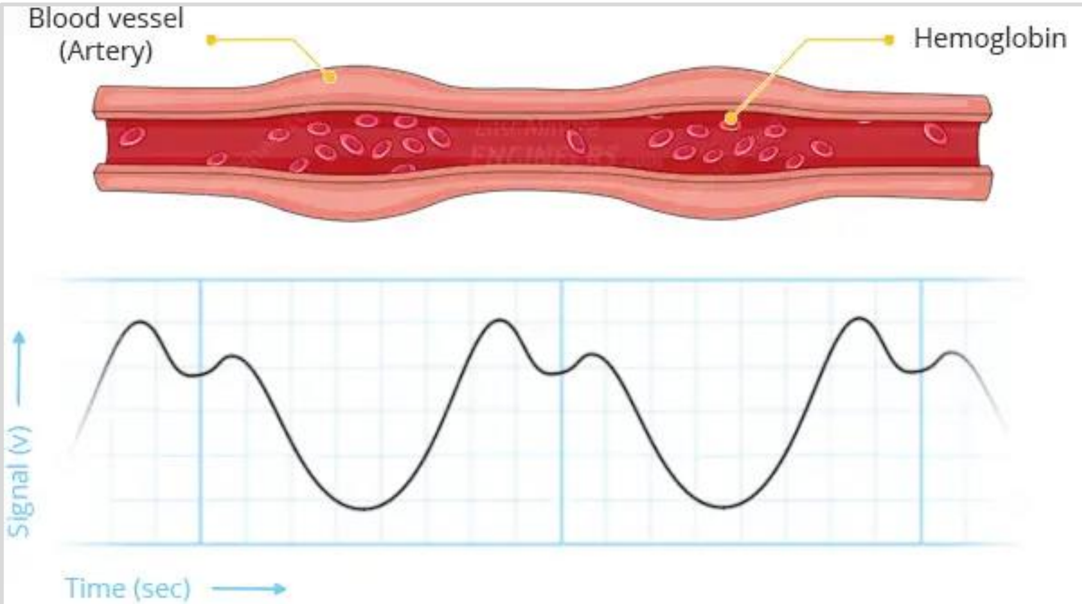


Figure 17 Heart Rate Measurement

F.2 Pulse Oximetry

Pulse oximetry is based on the principle that the amount of RED and IR light absorbed varies depending on the amount of oxygen in your blood. The following graph is the absorption-spectrum of oxygenated hemoglobin (HbO2) and deoxygenated hemoglobin (Hb).

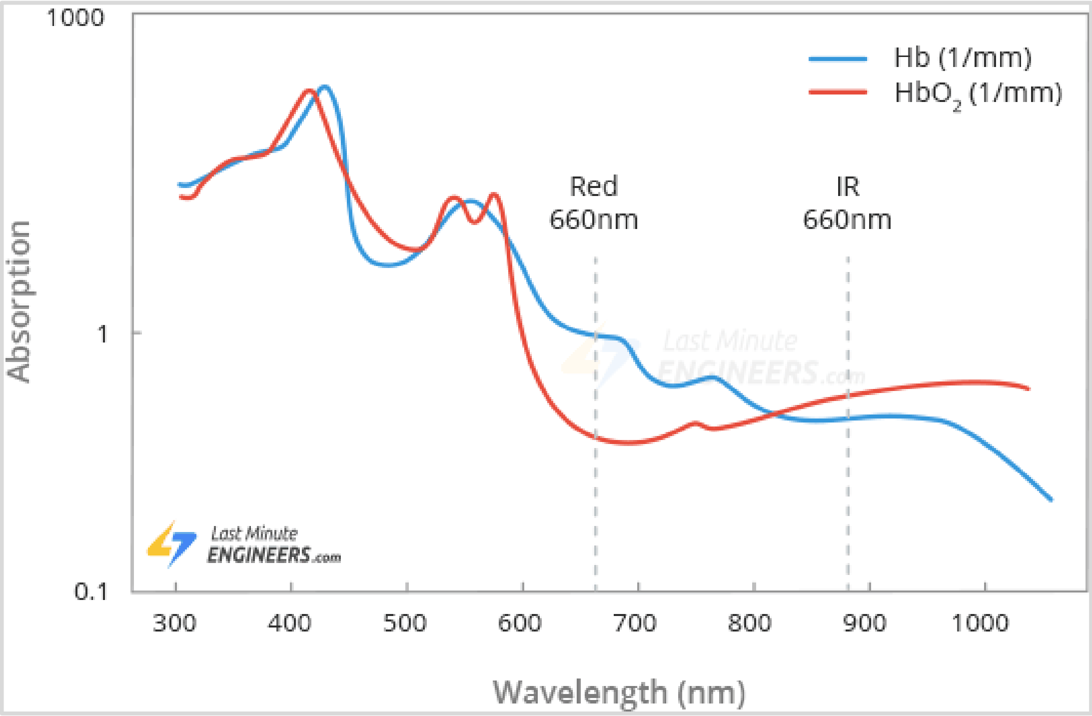


Figure 18 pulse Oximeter

As you can see from the graph, deoxygenated blood absorbs more RED light (660nm), while oxygenated blood absorbs more IR light (880nm). By measuring the ratio of IR and RED light received by the photodetector, the oxygen level (SpO2) in the blood is calculated.

3.G MAX30100 Module Hardware Overview

The module features the MAX30100 a modern, integrated pulse oximeter and heart rate sensor IC, from Analog Devices. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry (SpO2) and heart rate (HR) signals.

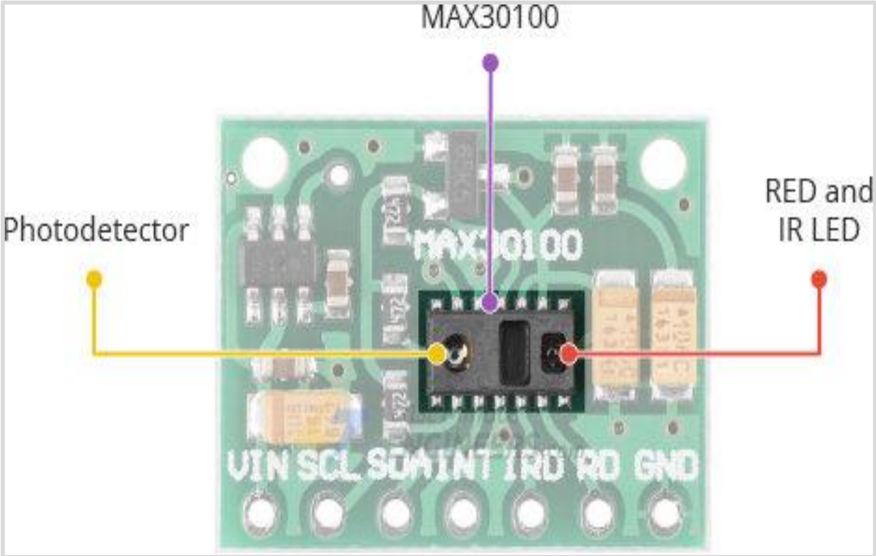


Figure 19 MAX30100 Module Hardware Overview

On the right, the MAX30100 has two LEDs – a RED and an IR LED. And on the left is a very sensitive photodetector. The idea is that you shine a single LED at a time, detecting the amount of light shining back at the detector, and based on the signature, you can measure blood oxygen level and heart rate.

G.1 Power Requirement

The MAX30100 chip requires two different supply voltages: 1.8V for the IC and 3.3V for the RED and IR LEDs. So, the module comes with 3.3V and 1.8V regulators. This allows you to connect the module to any microcontroller with 5V, 3.3V, even 1.8V level I/O.

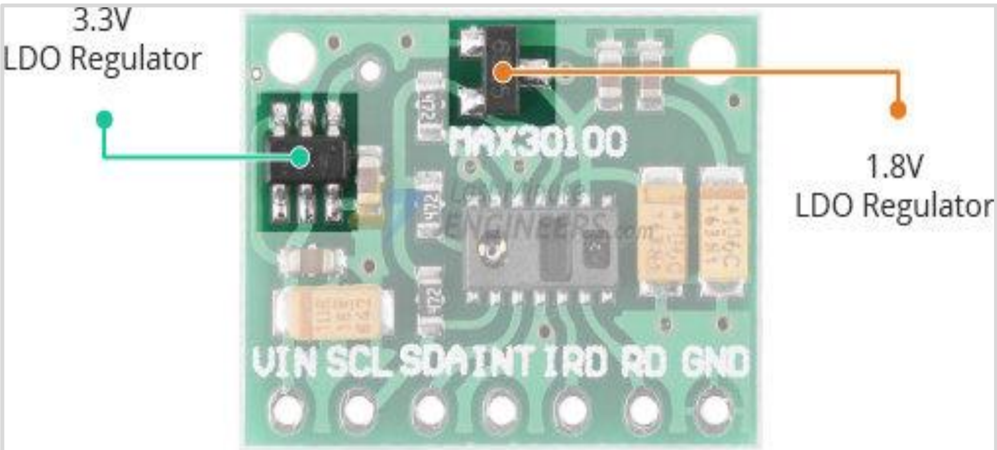


Figure 20 power Requirement

One of the most important features of the MAX30100 is its low power consumption: the MAX30100 consumes less than 600µA during measurement. Also, it is possible to put the MAX30100 in standby mode, where it consumes only 0.7µA. This low power consumption allows implementation in battery powered devices such as handsets, wearables, or smart watches.

G.2 On-Chip Temperature Sensor

The MAX30100 has an on-chip temperature sensor that can be used to compensate for the changes in the environment and to calibrate the measurements.

This is a reasonably precise temperature sensor that measures the 'die temperature' in the range of -40°C to +85°C with an accuracy of ±1°C.

G.3 I2C Interface

The module uses a simple two-wire I2C interface for communication with the microcontroller. It has a fixed I2C address: $0xAE_{HEX}$ (for write operation) and $0xAF_{HEX}$ (for read operation).

G.4 FIFO Buffer

The MAX30100 embeds a FIFO buffer for storing data samples. The FIFO has a 16-sample memory bank, which means it can hold up to 16 SpO2 and heart rate samples. The FIFO buffer can offload the microcontroller from reading each new data sample from the sensor, thereby saving system power.

G.5 Technical Specifications

Table 1 Temperature sensor's technical specifications

Power supply	**3.3V to 5.5V**
Current draw	~600µA (during measurements)
	~0.7µA (during standby mode)
Red LED Wavelength	660nm
IR LED Wavelength	880nm
Temperature Range	-40˚C to +85˚C
Temperature Accuracy	±1˚C

3.H Temperature Sensor

The MLX90614 non-contact temperature sensor is placed near the human body, and it detects the temperature values without physical contact with patients. The MLX90614 is integrated with a low noise amplifier, 17-bit ADC and powerful DSP unit thus achieving high accuracy and resolution of the thermometer. The schematic of sensor MLX90614 is shown in Fig.4.1.

H.1 MLX90614 Module Hardware Overview

At the heart of the module is a high precision non-contact infrared temperature sensor from Melexis MLX90614. Unlike most temperature sensors, this sensor measures temperature without being physically touched. This can be very useful for monitoring the temperature of something moving like a spinning motor shaft or objects on a conveyor belt for example. Simply point the sensor at what you want to measure, and it will detect the temperature by absorbing the emitted IR waves.

Figure 21 MLX90614 Module Hardware

H.2 Capabilities

The MLX90614 generates two temperature measurements: an object temperature and an ambient temperature. The object temperature is the non-contact measurement ‘observed’ from the sensor, while the ambient temperature measures the temperature on the die of the sensor. Ambient temperature can be used to calibrate the data, but what we really need comes from object temperature measurements.

Because it does not have to touch the object being measured, it can sense a wider range of temperatures than most digital sensors: object temperature measurements range from -70 to 382.2°C, while ambient temperature measurements range from -40 to 125°C. Both the ambient temperature and the object temperature have a resolution of 0.02°C with a standard accuracy of 0.5°C around room temperatures.

H.3 Built-In Optical Filter

The MLX90614 has a built-in optical filter that cuts off visible and near-infrared light, reducing their effect on measurements. It also provides immunity against ambient light and sunlight.

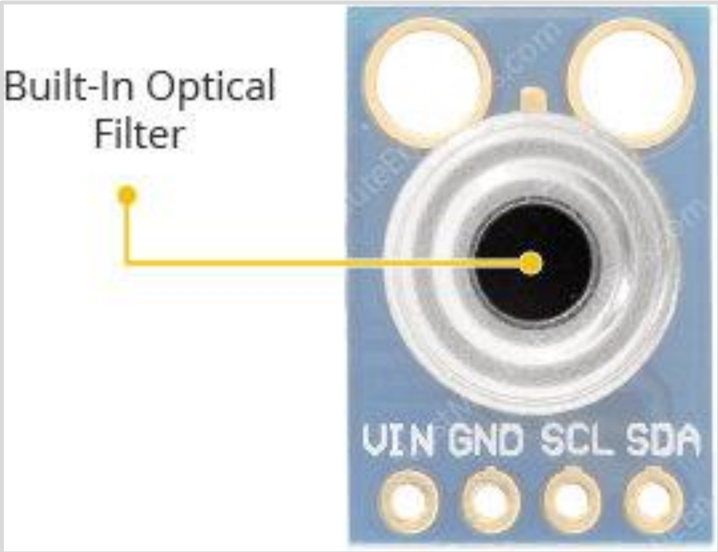


Figure 22 Built-In Optical Filter

H.4 Power Requirement

The module comes with a 662K 3.3V precision voltage regulator and voltage level translator, so you can use it with your favorite 3.3V or 5V microcontroller without any worries.

The MLX90614 consumes less than 2mA during measurement. This low power consumption allows implementation in battery powered devices such as handheld thermal scanners.

Figure 23 Power Requirement

H.5 Module specifications:

Table 2 MLX90614 specifications

Object temperature	**-70°C to 382.2°C**
Ambient temperature	-40°C to 85°C
Accuracy	±0.5°C (around room temperatures)
Resolution	±0.2°C
Field of view	90°
Supply voltage	3.3 to 5.5V
Operating Current	2mA

H.6 How Does the Infrared Thermometers Work?

If you've ever used or seen someone, use an infrared thermometer before, you may have found yourself wondering, "How is this style of measurement even possible?"

Well, Infrared thermometers like MLX90614 take advantage of the fact that any object, including humans, above absolute zero (0°K or -273°C) temperature, emits (not visible to the human eye) light in the infrared spectrum that is directly proportional to its temperature. Refer to the **Stefan–Boltzmann law.**

Internally, the MLX90614 is a pair of two devices: an infrared thermopile detector and an ASSP (Signal-Conditioning Application Processor). Here is the internal block diagram of the MLX90614 showing both the thermopile and the ASSP.

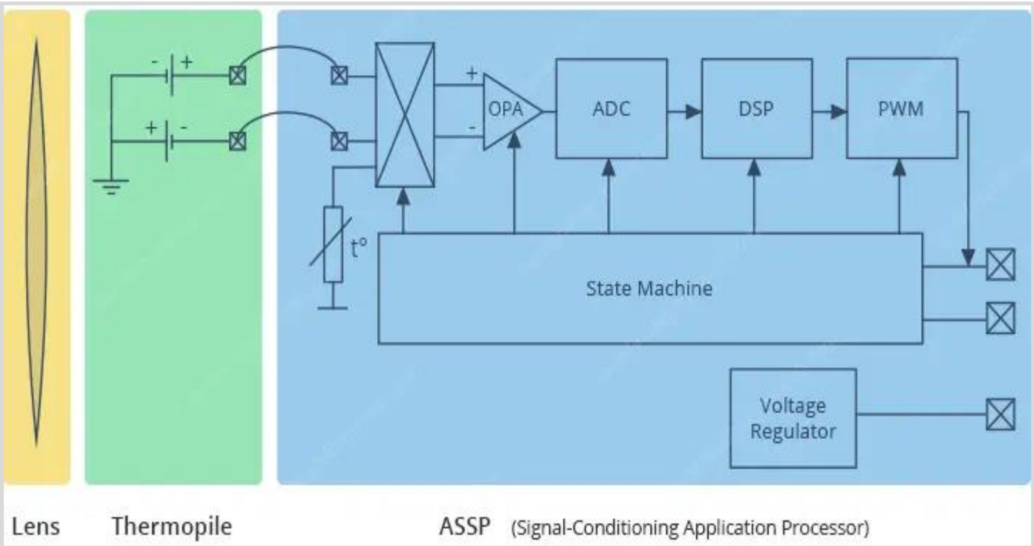


Figure 24 Infrared Thermometers

The IR radiation emitted by an object or human is first focused by a converging (convex) lens onto a special infrared detector called a Thermopile. The thermopile senses how much infrared energy is being emitted by objects in its field-of-view (FOV) and generates an electrical signal proportional to that.

The voltage produced by the thermopile is picked up by the ASSP's 17-bit ADC and then processed before passing to the microcontroller. And the best part is that this whole process is achieved in a fraction of a second.

H.7 Field of View (FOV)

An IR thermometer's field-of-view (FOV) is one of the most important metrics to be aware of. It is determined by the angle in which the sensor is sensitive to thermal radiation. This means that the sensor will detect all objects in the field-of-view and return the average temperature of all objects in it.

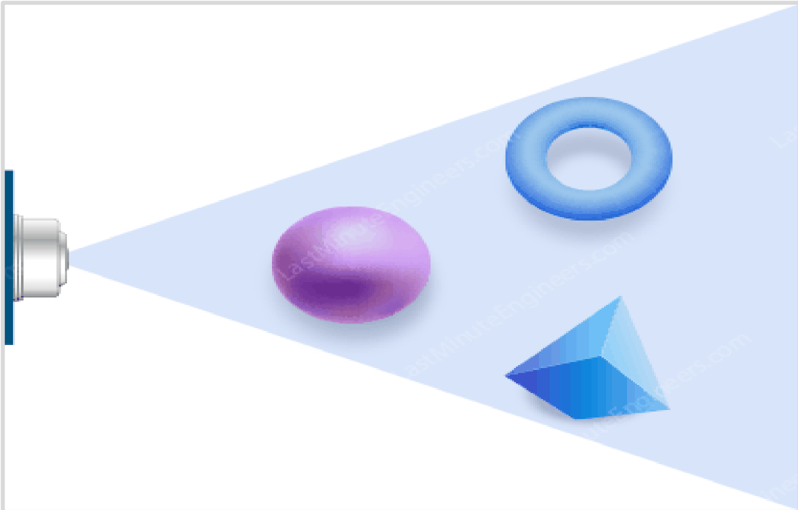


Figure 25 field of view 1

It is important that the measured object completely fills the field-of-view. Otherwise, the sensor may detect objects that are not supposed to be measured, resulting in inaccurate measurements.

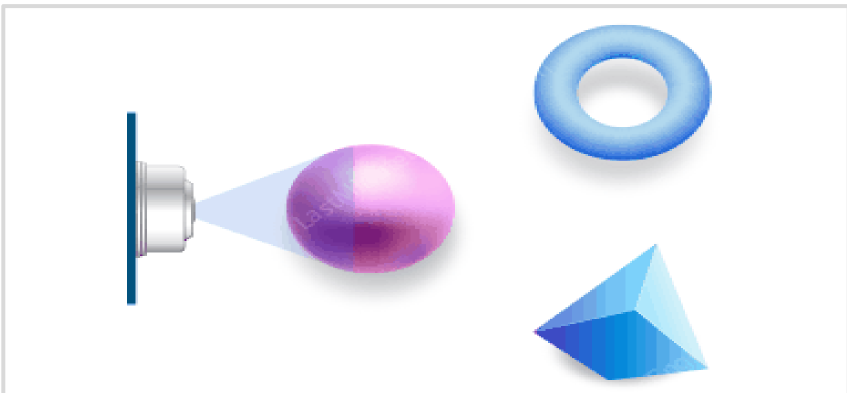


Figure 26 field of view 2

The field-of-view also determines the relationship between the distance from an object and the sensing area. If the sensor is near the object, its sensing area is very narrow, but gets increasingly wider as it moves farther away.

The field-of-view of the MLX90614 is cone-shaped and relatively wide: 90°. This means that for every 1cm you move away from an object, the sensing area increases by 2cm. If you are one-foot 30cm (approx. 1 foot) away from an object, the sensing area will be 60cm (approx. 2 feet).

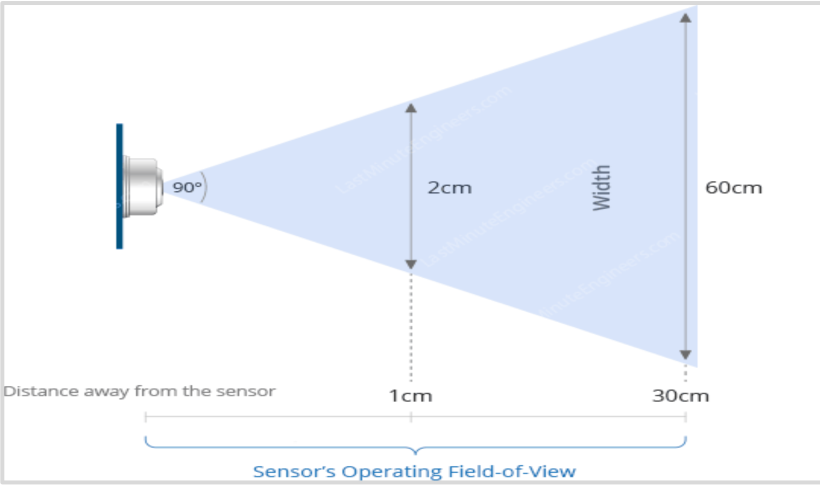


Figure 5.8 sensor operating field of view.

H.8 MLX90614 Output Interfaces

The MLX90614 supports two interfaces; though you will need one to access the other. The 2-wire SMBus interface is the primary means of communicating with the MLX90614. Once the SMBus interface is set up, you can later configure the MLX90614 to produce a PWM (pulse-width-modulated) signal representing the measured temperature.

H.9 SMBus Interface

The primary interface to the MLX90614 is the 2-wire SMBus interface which is basically the same as I2C (a slightly non-standard type of I2C called "repeated-start") and uses the same two signals – SDA and SCL – to carry data and clock signals respectively. A master device controls the clock signal, while the data signal is controlled by-directionally.

Every MLX90614 has a default I2C address of 0x5A. However, it can be programmed to have one of 127 I2C addresses so that you can add up to 127 devices to the same bus to get a larger temperature map.

H.10 PWM Interface

The MLX90614's data can also be read through the PWM interface. Note that in order to use the PWM interface, the MLX90614 must first be configured over the SMBus.

Once configured, the MLX90614 outputs a continuous 10-bit PWM signal on the SDA pin that represents the measured object temperature. By default, the PWM signal covers the range of -20°C to 120°C with an output resolution of 0.14°C, but this can also be adjusted via SMBus.

H.11 Thermal Relay/Thermal Switch

By configuring this range (setting minimum and maximum temperature values) the PWM output can be turned into a "Thermal Relay/Thermal Switch" signal.

So, when the temperature exceeds the set threshold, the PWM pin is triggered which can be used as an interrupt source or can be used to directly control a relay. Note that the output drive capability is 25mA only.

4. Communications Department

Develops sophisticated algorithms to analyze medical data, trains AI models using the collected and preprocessed data to ensure accurate diagnostics, predictions and creates a mobile application for user interaction.

5. Biosensors and Electronics:

The rapid advancement of medical imaging technology has opened new possibilities for early disease detection and diagnosis. In this thesis, we explore the development of an automated lung disease detection system utilizing a combination of hardware and AI techniques. Our system employs a **Raspberry Pi 4 Model B** to control a camera that captures lung X-ray images. These images are subsequently uploaded to a cloud storage for comparison with existing data. The objective is to detect and classify lung diseases automatically, without any human intervention. Patients receive the results through a dedicated application.

In the domain of biomedical engineering, merging technology with healthcare has created exciting new possibilities for patient monitoring and data analysis. This project utilizes the **Raspberry Pi 4 Model B**, a powerful and versatile microcomputer, along with two specialized sensors the **MAX30100** and the **MLX90614** to monitor key health parameters. The **MAX30100** sensor measures blood oxygen saturation (SpO2) and heart rate, while the **MLX90614** sensor accurately captures body temperature without requiring direct contact with the skin.

6. AI and robotics are transforming healthcare

6.A Keeping Well:

One of AI's biggest potential benefits is to help people stay healthy, so they don't need a doctor, or at least not as often. The use of AI and the Internet of Medical Things (IoMT) in consumer health applications is already helping people. Technology applications and apps encourage healthier behavior in individuals and help with the proactive management of a healthy lifestyle. It puts consumers in control of health and well-being. Additionally, AI increases the ability for healthcare professionals to better understand the day-to-day patterns and needs of the people they care for, and with that understanding they can provide better feedback, guidance, and support for staying healthy [24].

6.B Early Detection:

AI is already being used to detect diseases, such as cancer, more accurately and in their early stages. According to the American Cancer Society, a high proportion of mammograms yield false results, leading to 1 in 2 healthy women being told they have cancer. The use of AI is enabling review and translation of mammograms 30 times faster with 99% accuracy, reducing the need for unnecessary biopsies.

The proliferation of consumer wearables and other medical devices combined with AI is also being applied to oversee early-stage heart disease, enabling doctors and other caregivers to better monitor and detect potentially life-threatening episodes at earlier, more treatable stages [24].

6.C Diagnosis:

IBM’s Watson for Health is helping healthcare organizations apply cognitive technology to unlock vast amounts of health data and power diagnosis. Watson can review and store far more medical information – every medical journal, symptom, and case study of treatment and response around the world – exponentially faster than any human.

Google’s DeepMind Health is working in partnership with clinicians, researchers, and patients to solve real-world healthcare problems. Technology combines machine learning and systems neuroscience to build powerful general-purpose learning algorithms into neural networks that mimic the human brain [24].

6.D Decision Making:

Improving care requires the alignment of big health data with appropriate and timely decisions, and predictive analytics can support clinical decision-making and actions as well as prioritize administrative tasks.

Using pattern recognition to identify patients at risk of developing a condition – or seeing it deteriorate due to lifestyle, environmental, genomic, or other factors – is another area where AI is beginning to take hold in healthcare [24].

6.E Treatment:

Beyond scanning health records to help providers identify chronically ill individuals who may be at risk of an adverse episode, AI can help clinicians take a more comprehensive approach for disease management, better coordinate care plans and help patients to better manage and comply with their long-term treatment programs.

Robots have been used in medicine for more than 30 years. They range from simple laboratory robots to highly complex surgical robots that can either aid a human surgeon or execute operations by themselves. In addition to surgery, they're used in hospitals and labs for repetitive tasks, in rehabilitation, physical therapy and in support of those with long-term conditions [24].

6.F Research:

The path from research lab to patient is a long and costly one. According to the California Biomedical Research Association, it takes an average of 12 years for a drug to travel from the research lab to the patient. Only five in 5,000 of the drugs that begin preclinical testing ever make it to human testing and just one of these five is ever approved for human usage. Furthermore, on average, it will cost a company US $359 million to develop a new drug from the research lab to the patient.

Drug research and discovery is one of the more recent applications for AI in healthcare. By directing the latest advances in AI to streamline the drug discovery and drug repurposing processes there is the potential to significantly cut both the time to market for new drugs and their costs [24].

6.G Training:

AI allows those in training to go through naturalistic simulations in a way that simple computer-driven algorithms cannot. The advent of natural speech and the ability of an AI computer to draw instantly on a large database of scenarios, means the response to questions, decisions or advice from a trainee can challenge in a way that a human cannot. And the training program can learn from previous responses from the trainee, meaning that the challenges can be continually adjusted to meet their learning needs.

And training can be done anywhere; with the power of AI embedded on a smartphone, quick catch-up sessions, after a tricky case in a clinic or while travelling, will be possible [24].

7. Tasks Machine Learning addresses in healthcare

Machine learning techniques can be applied to solve a wide variety of tasks. When it comes to applications of machine learning in healthcare, these tasks include [25]:

7.A Classification:

Machine learning algorithms can help to determine and label the kind of disease or medical case you're dealing with.

7.B Recommendations:

Machine learning algorithms can offer necessary medical information without the need to actively search for it.

7.C Clustering:

Machine learning can help to group together similar medical cases to analyze the patterns and conduct research in the future.

7.D Prediction:

Using current data and common trends, machine learning can make a prognosis on how future events will unfold.

7.E Anomaly detection:

Using machine learning in healthcare, you can find the things that stand out from common patterns and determine whether they require any actions to be performed.

7.F Automation:

machine learning can handle standard repetitive tasks that take too much time and effort from doctors, like data entry, appointment scheduling, inventory management, etc. For example, chatbots based on natural language processing can relieve the burden on medical staff through communication with patients.

7.G Ranking:

Machine learning can put the relevant information first, making the search for it easier.

8. Application of Machine Learning in Healthcare Communication

Machine learning (ML) is a branch of artificial intelligence (AI) and computer science that focuses on the using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy.

Machine learning comprises a series of algorithms to analyze data, learn from it, and make informed selections based on statistics [26].

8.A Patient Care

Healthcare chatbots have high potential in medical communication by improving communication between clinic–patient and doctor–patient. It can help fulfill the high demand for health services through remote testing, monitoring of medication follow-up, or telephone consultations. A chatbot can conduct fast and easy health surveys, set up personal health-related reminders, communicate with clinical teams, book appointments, and retrieve and analyze health data. Chatbots can provide fast or instant responses to patients' healthcare-related questions while looking for specific symptoms or patterns in predicting disease [26].

8.B Education and Knowledge Transfer System

To provide a practical collaborative healthcare workforce, interprofessional education is essential. Interprofessional education occurs when more than one profession learns with, or from, or about each other. Its main purpose is to increase collaboration and improve healthcare quality. It provides a collaborative framework and provides insight into how each discipline contributes without losing its own identity. Text mining and computational linguistics are two computationally intensive fields where more options are becoming available to study large text corpora and implement corpora for various purposes. In these systems, the analysis of text does not require a deep understanding technique. AIML is the assembly language for the AI conversational agent, such as a chatbot in most cases.

In the last few years, there was a rapid increase in chatbots in various fields, such as health care, education, marketing, cultural heritage, supporting system, entertainment, and many others. The chatbot in the educational domain has high potential, especially when delivering medical science knowledge. Fabio's paper shows the implementation of such a chatbot as a teaching medium and demonstrates its utility. A chatbot can be used in instructional situations for educational purposes due to its interactive ability as opposed to traditional e-learning modules. Students can continuously interact with the chatbot. They can ask questions related to a specific field. A chatbot can also be used to learn or study a new language. It can be used to visualize a corpus's content, a tool to access information systems, and a tool to give answers to questions in a specific domain [26].

8.C Emergency Response and COVID-19

In emergencies, a few minutes can save a life. Quicker health care and accessibility systems can save many lives in many cases. A semi-automated emergency paramedical end-to-end response system was proposed. It can distribute medical supplies on-site in case of emergencies. This system utilizes decentralized distribution and does not involve any third-party institutions to ensure security. The response system can be used in urban, semi-urban, and rural areas. It allows community hospitals to provide specialized healthcare despite the absence of a specialized doctor. The response system uses drones to access remote areas that are difficult for a human to access. The deep neural network allows the drone to detect objects, making it more accurate and reducing failure significantly. The chatbot takes user responses and evaluates them, then, if need be, passes them to the administration. The chatbot uses NLP to refine input responses. The response system can also be utilized to detect stroke using CT scans.

Coronavirus disease 2019 (COVID-19) has become a major global concern since January 2020. It primarily affects the cardiovascular system and requires sensitive, fast, and specific tools to identify the disease early, and better preventative measures can be applied. COVID-19 is a worldwide crisis. Over 100 million people have been infected already. It has caused over 2 million deaths worldwide. Many countries have overstretched their healthcare resources to mitigate the spread of the pandemic. AI was implemented to monitor and control the COVID-19 pandemic in several critical areas. It was also used to predict the risk of developing the disease, hospital admission, and progression in those areas. AI was also used for early detection and diagnosis. DNN can be used in conjunction with x-ray or CT to diagnose COVID-19 detection and automate the process to keep up with the overwhelming demand [26].

9. Machine learning terminologies

While going through the book, we will get many terminologies related to machine learning. The terminologies that we need to understand are discussed in this section.

9.A Features

Features, also known as attributes, predictor variables, or independent variables, are simply the characteristics or labels of the dataset. Suppose we have information on the height and the mass of sixty students in a class. The height and the mass are known as the features within the dataset. These features are extracted from the raw dataset and are fed to models as inputs [27].

9.B Target Variable

Simply put, target variables are the outputs that the models should give. For instance, a movie review must be classified as positive or negative. Here, the variable positive/negative is the target variable in this case. First, this target variable needs to be determined by the user. Then, after the target variable is determined, the relationship between the features and the target variable must be understood to perform further operations [27].

9.C Algorithm

A method, function, or series of instructions used to generate a machine learning model. Examples include linear regression, decision trees, support vector machines, and neural networks [27].

9.D Classification

Predicting a categorical output.

- Binary classification predicts one of two possible outcomes (e.g. is the email spam or not spam?)
- Multi-class classification predicts one of multiple possible outcomes (e.g. is this a photo of a cat, dog, horse or human?) [27].

9.E Classification Threshold

The lowest probability value at which we're comfortable asserting a positive classification. For example, if the predicted probability of being diabetic is > 50%, return True, otherwise return False [27].

9.F Dimension

The dimensions for machine learning and data scientist is different from physics. Here, the dimension of data means how many features you have in your data ocean(dataset). e.g. in case of object detection application, flatten image size and color channel (e.g. 28*28*3) is a feature of the input set. In case of house price prediction (maybe) house size is the data-set so we call it 1 dimensional data [27].

9.G Hyperparameters

Hyperparameters are higher-level properties of a model such as how fast it can learn (learning rate) or complexity of a model. The depth of trees in a Decision Tree or number of hidden layers in a Neural Networks are examples of hyper parameters [27].

9.H Model

A data structure that stores a representation of a dataset (weights and biases). Models are created/learned when you train an algorithm on a dataset [27].

9.I Normalization

Restriction of the values of weights in regression to avoid overfitting and improve computation speed [27].

9.J Noise

Any irrelevant information or randomness in a dataset which obscures the underlying pattern [27].

9.K Overfitting

Overfitting occurs when your model learns the training data too well and incorporates details and noise specific to your dataset. You can tell a model is overfitting when it performs great on your training/validation set, but poorly on your test set (or new real-world data) [27].

9.L Underfitting

Underfitting occurs when your model over-generalizes and fails to incorporate relevant variations in your data that would give your model more predictive power. You can tell a model is underfitting when it performs poorly on both training and test sets [27].

9.M Precision

In the context of binary classification (Yes/No), precision measures the model's performance at classifying positive observations (i.e. "Yes"). In other words, when a positive value is predicted, how often is the prediction correct? We could game this metric by only returning positive for the single observation we are most confident in [27].

$$P = \frac{TruePositives}{TruePositives + FalsePositives}$$

9.N Recall

Also called sensitivity. In the context of binary classification (Yes/No), recall measures how "sensitive" the classifier is at detecting positive instances. In other words, for all the true observations in our sample, how many did we "catch." We could game this metric by always classifying observations as positive [27].

$$R = \frac{TruePositives}{TruePositives + FalseNegatives}$$

9.O Recall vs Precision

Say we are analyzing Brain scans and trying to predict whether a person has a tumor (True) or not (False). We feed it into our model and our model starts guessing.

- **Precision** is the % of True guesses that were actually correct! If we guess 1 image is True out of 100 images and that image is actually True, then our precision is 100%! Our results aren't helpful however because we missed 10 brain tumors! We were super precise when we tried, but we didn't try hard enough.
- **Recall**, or Sensitivity, provides another lens which with to view how good our model is. Again let's say there are 100 images, 10 with brain tumors, and we correctly guessed 1 had a brain tumor. Precision is 100%, but recall is 10%. Perfect recall requires that we catch all 10 tumors! [27].

9.P Specificity

In the context of binary classification (Yes/No), specificity measures the model's performance at classifying negative observations (i.e. "No"). In other words, when the correct label is negative, how often is the prediction correct? We could game this metric if we predict everything as negative [27].

$$S = \frac{TrueNegatives}{TrueNegatives + \ FalsePositives}$$

9.Q ROC (Receiver Operating Characteristic) Curve

A plot of the true positive rate against the false positive rate at all classification thresholds. This is used to evaluate the performance of a classification model at different classification thresholds. The area under the ROC curve can be interpreted as the probability that the model correctly distinguishes between a randomly chosen positive observation [27].

9.R PR AUC (area under the PR curve)

Area under the interpolated precision-recall curve, obtained by plotting (recall, precision) points for different values of the classification threshold. Depending on how it's calculated, PR AUC may be equivalent to the average precision of the model [27].

9.S test set

A subset of the dataset reserved for testing a trained model. Traditionally, you divide examples in the dataset into the following three distinct subsets [27]:

- **training set**
- **validation set**
- **test set**

10. Machine learning basic classification

Machine learning is generally categorized into three major groups: supervised learning, unsupervised learning, and reinforcement learning. Sometimes it also includes semi-supervised learning and active learning [4].

10.A Supervised learning

Supervised learning is a machine learning method in which the prediction models are acquired from labeled data. The labeled data represent the relationship between input and output, and the prediction model generates the corresponding output for a given input. The essence of supervised learning is to learn the statistical law of input–output mapping [4].

10.B Unsupervised learning

Unsupervised learning refers to the machine learning problem of learning predictive models from unlabeled data. Unlabeled data is naturally obtained data, and the predictive model represents the data's category, conversion, or probability. The essence of unsupervised learning is to learn statistical laws or potential structures in the data [4].

10.C Reinforcement learning

Reinforcement learning is also different from what machine learning researchers call unsupervised learning, which is typically about finding structure hidden in collections of unlabeled data. The terms supervised learning and unsupervised learning appear to exhaustively classify machine learning paradigms, but they do not. Although one might be tempted to think of reinforcement learning as a kind of unsupervised learning because it does not rely on examples of correct behavior, reinforcement learning is trying to maximize a reward signal instead of trying to find hidden structure. Uncovering structure in an agent's experience can certainly be useful in reinforcement learning, but by itself does not address the reinforcement learning agent's problem of maximizing a reward signal. We therefore consider reinforcement learning to be a third machine learning paradigm, alongside supervised learning, unsupervised learning, and perhaps other paradigms as well [28].

11. What is a machine learning algorithm?

A machine learning algorithm is a set of rules or processes used by an AI system to conduct tasks most often to discover new data insights and patterns, or to predict output values from a given set of input variables. Algorithms enable machine learning (ML) to learn.

Most often, training ML algorithms on more data will provide more accurate answers than training on less data. Using statistical methods, algorithms are trained to determine classifications or make predictions, and to uncover key insights in data mining projects. These insights can subsequently improve your decision-making to boost key growth metrics [29].

12. How machine learning algorithms work

The learning system of a machine learning algorithm can be divided into three main parts [30]:

- **A decision process:** In general, machine learning algorithms are used to make predictions or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.
- **An error function:** An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
- **A model optimization process:** If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met.

12.A Supervised learning algorithms

Supervised learning can be separated into two types of problems when data mining

- **Classification** uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, K-nearest neighbor and random forest, which are described in more detail below.
- **Regression** is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

Various algorithms and computations techniques are used in supervised machine learning processes:

1. **AdaBoost or gradient boosting:** Also called adaptive boosting7, this technique boosts an underperforming regression algorithm by combining it with weaker ones to create a stronger algorithm that results in fewer errors. Boosting combines the forecasting power of several base estimators.
2. **Artificial neural networks:** Also known as ANNs, neural networks or simulated neural networks (SNNs), are a subset of machine learning techniques and are at the heart of deep learning algorithms. The learner algorithm recognizes patterns in input data using building blocks called neurons, approximating the neurons in the human brain, which are trained and modified over time. (More in "neural networks.")
3. **Decision tree algorithms:** Used for both predicting numerical values (regression problems) and classifying data into categories, decision trees use a branching sequence of linked decisions that may be represented with a tree diagram. One of the advantages of decision trees is that they are easy to validate and audit, unlike the black box of a neural network.
4. **Dimensionality reduction:** When a selected data set has a high number of features7 , it has high dimensionality. Dimensionality reduction then cuts down the number of features, leaving only the most meaningful insights or information. An example is principal component analysis.
5. **K-nearest neighbor:** Also known as KNN, this non-parametric algorithm classifies data points based on their proximity and association to other available data. It assumes that similar data points can be found near each other. As a result, it seeks to calculate the distance between data points, usually through Euclidean distance, and then it assigns a category based on the most frequent category or average.
6. **Linear regression:** Linear regression is used to identify the relationship between a dependent variable and one or more independent variables and is typically leveraged to make predictions about future outcomes. When there is only one independent variable and one dependent variable, it is known as simple linear regression.
7. **Logistic regression:** While linear regression is leveraged when dependent variables are continuous, logistic regression is selected when the dependent variable is categorical, meaning there are binary outputs, such as "true" and "false"

or "yes" and "no." While both regression models seek to understand relationships between data inputs, logistic regression is mainly used to solve binary classification problems, such as spam identification.

8. **Neural networks:** Primarily leveraged for deep learning algorithms, neural networks process the input training data by mimicking the interconnectivity of the human brain through layers of nodes. Each node is made up of inputs, weights, a bias (threshold) and an output. If that output value exceeds a given threshold, it “fires” or activates the node, passing data to the next layer in the network. Neural networks learn from adjustments based on the loss function through the process of gradient descent. When the cost function is at or near zero, you can be confident in the model’s accuracy.
9. **Naïve Bayes:** This approach adopts the principle of class conditional independence from the Bayes Theorem. This means that the presence of one feature does not impact the presence of another in the probability of a given outcome, and each predictor has an equal effect on that result. There are three types of Naïve Bayes classifiers: Multinomial Naïve Bayes, Bernoulli Naïve Bayes and Gaussian Naïve Bayes. This technique is primarily used in text classification, spam identification and recommendation systems.
10. **Random forests:** In a random forest, the machine learning algorithm predicts a value or category by combining the results from a number of decision trees. The "forest" refers to uncorrelated decision trees, which are assembled to reduce variance and enable more accurate predictions.
11. **Support vector machines (SVM):** This algorithm may be used for both data classification and regression, but typically for classification problems, constructing a hyperplane where the distance between two classes of data points is at its maximum. This hyperplane is known as the decision boundary, separating the classes of data points (such as oranges vs. apples) on either side of the plane.

12.B Unsupervised learning algorithms

Unlike supervised learning, unsupervised learning uses unlabeled data. From that data, the algorithm discovers patterns that help solve clustering or association problems. This is particularly useful when subject matter experts are unsure of common properties within a data set. [29] Common algorithms are:

- **Clustering:** These algorithms can identify patterns in data so that it can be grouped. Algorithms can help data scientists by identifying differences between data items that humans have overlooked.
- **Hierarchical clustering:** This groups data into a tree of clusters8. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes these steps: 1) identify the two clusters which can be closest together, and 2) merge the two maximum comparable clusters. These steps continue until all the clusters are merged together.
- **K-means clustering:** This identifies groups within data without labels9 into different clusters by finding groups of data which are similar to one another. The name "K-means" come from the k centroids that it uses to define clusters. A point is assigned to a particular cluster if it is closer to that cluster's centroid than any other centroid.

13. Applications of Computer Vision to Medical Diagnosis

13.A Dermatology

Dermatology is the branch of medicine dealing with the skin. One of the tasks dermatologists perform is to look at a suspicious region of the skin to determine whether a mole is skin cancer or not. Early detection could likely have an enormous impact on skin cancer outcomes. The five-year survival rate of one kind of skin cancer drops significantly if detected in its later stages.

In this study, an algorithm is trained to determine whether a region of skin tissue is cancerous or not.

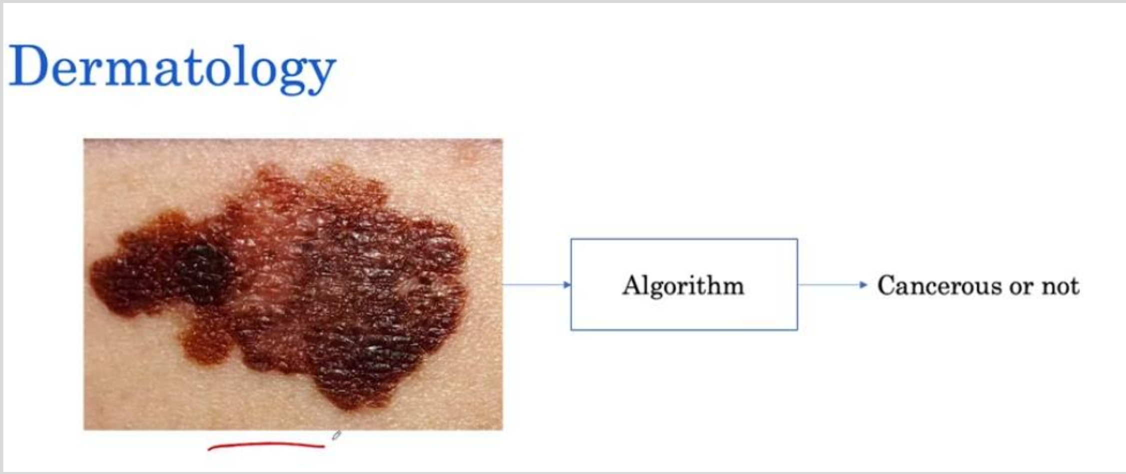


Figure 27 Dermatology Detection

Dermatologist-level classification of skin cancer with deep neural networks Skin cancer, the most common human malignancy, is primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis, a biopsy and histopathological examination. Automated classification of skin lesions using images is a challenging task owing to the fine-grained variability in the appearance of skin lesions. Deep convolutional neural networks (CNNs) show potential for general and highly variable tasks across many fine-grained object categories

Classification technique is a deep CNN. Data flow is from left to right: an image of a skin lesion (for example, melanoma) is sequentially warped into a probability distribution over clinical classes of skin disease using Google Inception v3 CNN architecture pretrained on the ImageNet dataset (1.28 million images over 1,000 generic object classes) and fine-tuned on our own dataset of 129,450 skin lesions comprising 2,032 different diseases. The 757 training classes are defined using a novel taxonomy of skin disease and a partitioning algorithm that maps diseases into training classes (for example, acrolentiginous melanoma, amelanotic melanoma, lentigo melanoma). Inference classes are more general and are composed of one or more training classes (for example, malignant melanocytic lesions the class of melanomas). The probability of an inference class is calculated by summing the probabilities of the training classes according to taxonomy structure (see Methods).

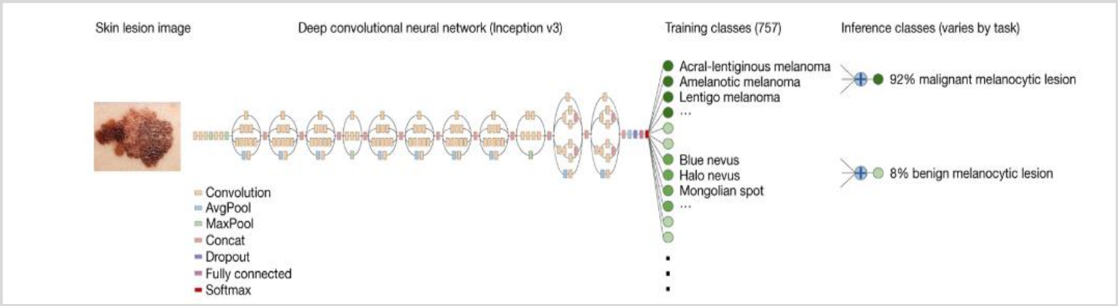


Figure 28 inception algorithm for Dermatology Detection

Skin cancer, the most common human malignancy, is primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis, a biopsy and histopathological examination. Automated classification of skin lesions using images is a challenging task owing to the fine-grained variability in the appearance of skin lesions. Deep convolutional neural networks (CNNs) show potential for general and highly variable tasks across many fine-grained object categories. Here we demonstrate classification of skin lesions using a single CNN, trained end-to-end from images directly, using only pixels and disease labels as inputs. We train a CNN using a dataset of 129,450 clinical images

13.B Ophthalmology

which deals with the diagnosis and treatment of eye disorders. One well-known study in 2016 looked at retinal fundus images, which photographed the back of the eye. One disease or pathology to look at here is diabetic retinopathy, which is damage to the retina caused by diabetes and is a major cause of blindness. Currently, detecting DR is a time-consuming and manual process that requires a trained clinician to examine these photos. In this study, an algorithm was developed to determine whether patients had diabetic retinopathy by looking at such photos.

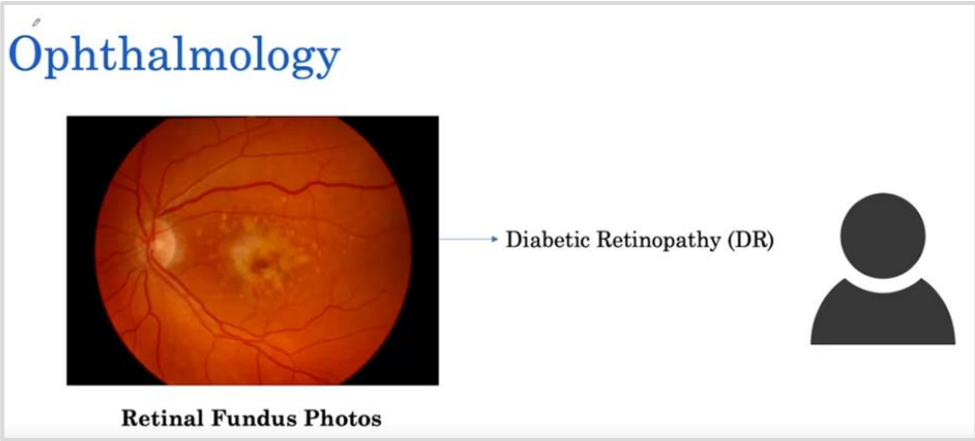


Figure 29 Ophthalmology Detection-1

This study used over 128,000 images of which only 30 percent had diabetic retinopathy. We'll look at this data imbalanced problem, which is prominent in medicine and in many other fields with real-world data. We will see some methods for tackling this challenge.

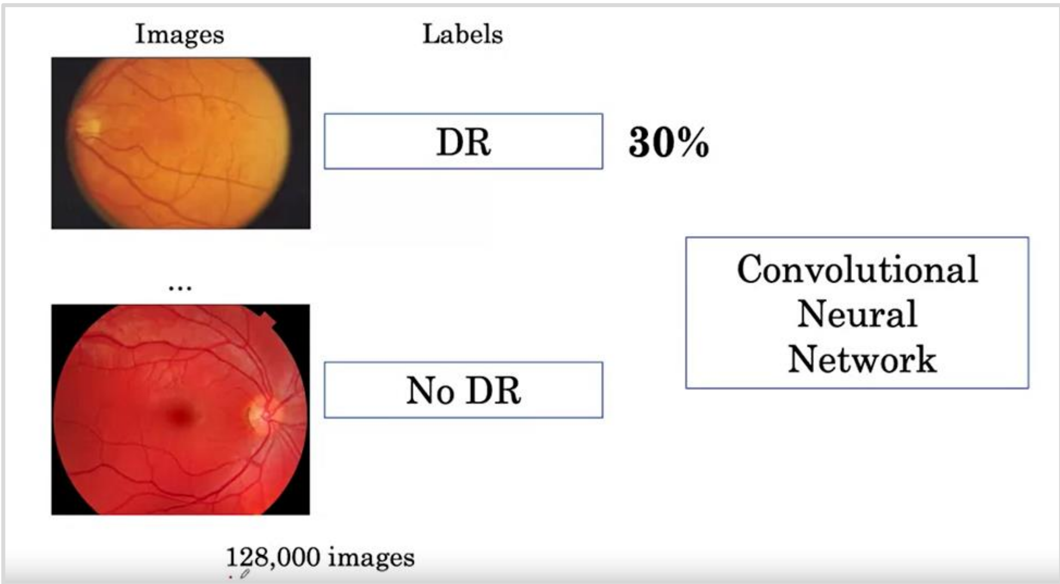


Figure 30 Ophthalmology Detection-2

13.C Histopathology

This study showed that the performance of the resulting algorithm was comparable to ophthalmologists. In the study, a majority vote of multiple ophthalmologists was used to set the reference standard or ground truth, which is a group of experts, best guess of a right answer. Later this week in the course, we'll look at how ground truth can be set in such medical AI studies. Our third example is in histopathology, a medical specialty involving examination of tissues under the microscope. One of the tasks that pathologists do is look at scanned microscopic images of tissue called whole slide images, and determine the extent to which a cancer has spread. This is important to help plan treatment, predict the course of the disease, and the chance of recovery.

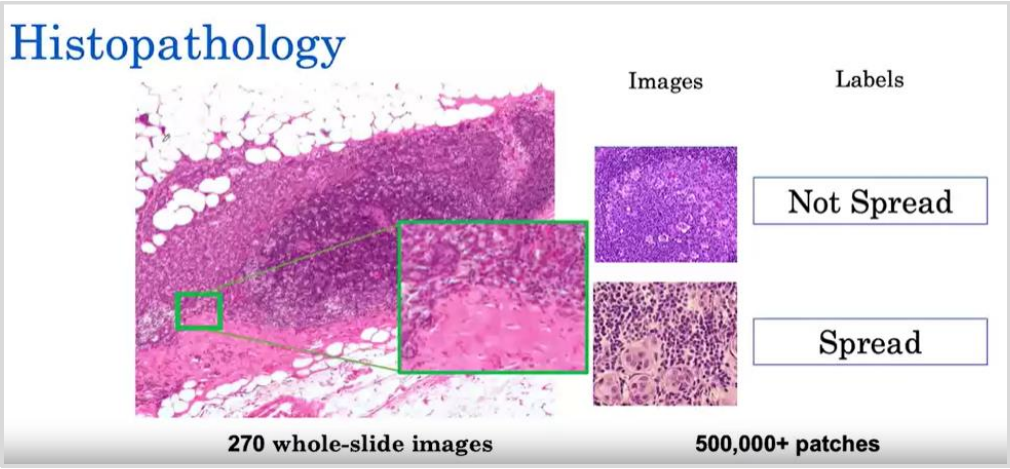


Figure 31 Histopathology Detection

In one study in 2017, using only 270 whole slide images, AI algorithms were developed and then evaluated against pathologists. It was found that the best algorithms performed as well as the pathologists did. Now in histopathology, the images are very large and cannot be fed directly into an algorithm without breaking them down. The general setup of these studies is that instead of feeding in one large, high resolution digital image of the slide, several patches are extracted at a high magnification and used to train a model. These patches are labeled with the original label of the whole slide image and then fed into a deep learning algorithm. In this way, the algorithm can be trained on hundreds of thousands of patches. In this course, you will apply a similar idea of breaking down a large image into smaller images for model training to the task of brain tumor segmentation.

14. Chest X-Ray 14 Diseases Detection

Deep learning networks have gained much popularity in recent years with state-of-the-art results in various machine learning problems. Improvement in hardware, choice of optimization techniques, availability of large datasets, and many theoretical breakthroughs have led to the success of deep learning. One significant area in machine learning that has greatly benefited from the success of deep learning is transfer learning, where a model trained for one task is used for a different but related task. The effectiveness of transfer learning has been widely studied with traditional machine learning methods. However, recent advancements in deep learning have caused many researchers to focus on how to effectively use deep learning for transfer learning, gradually displacing traditional transfer learning methods. Transfer learning in deep learning can be divided into four categories based on the scenarios: 1) Fixed feature extractor, 2) Pretrained Initialization, 3) Domain-specific feature spaces, and 4) Deep model cascades.

14.A Definition of Deep Learning

Deep learning technology has truly blossomed with most of the amazing success stories. Starting with the convolutional networks, the real progress was made in 2012 when AlexNet largely showed a certain class of images recognized using those convolutional networks. This was because the proficiency of deep learning largely extends from having nested body representations and fairly conducting progress in the last layer in a supervised fashion. Following the success of AlexNet, just started to have all forms of practitioners and researchers jumping in and improve the convolutional networks and methodically get progress in different classes of images.

When discussing deep learning, we are talking about the application of deep neural networks to specified problems, resulting in the need to train these models which then are able to offer estimations on new data. Deep learning is a subset of machine learning which is a subset of artificial intelligence. Artificial intelligence, machine learning, and deep learning are substitute terms for closely related subjects, and they make many famous algorithms according to their manner of learning some things from the data. It becomes increasingly harder to find a definite edge or the particular elements in order to categorize the three classes or to understand what the innovative understanding of the most pervasive machine learning terminologies is.

14.B Choice of deep neural networks

Image classification is one of the earliest and most studied tasks in **deep learning**. Thus, for medical image classification, especially 2D medical image classification, instead of designing **neural networks** by ourselves, we may consider directly applying the existing models. Some early works include AlexNet , VGGNet , and GoogleNet . Compared to non-deep **learning approaches**, these models show exceptional performance on the ImageNet **classification task** [46]. Therefore, they had been soon after adopted to solve medical problems such as skin disease/lesion classification , pathological chest X-ray classification, **mitosis detection** , and nucleus classification . Despite the popularity, these early **deep neural networks**, however, are outdated and do not retain competitive performance compared with the later models.

14.C Image Classification

Image classification is the primary domain, in which deep neural networks play the most important role of medical image analysis. The image classification accepts the given input images and produces output classification for identifying whether the disease is present or not. E. Kim et al. proposed a CNN method which outperforms perfect image classification accuracy in cytopathology. Inception v3 architecture is one of the best methods for medical data analysis and has accomplished proficient human performance. The CNN architecture proposed by E. Hosseini-AsL et al. uses three-dimensional convolutions to classify the Alzheimer disease. J. Kawahara et al. proposed a CNN-like architecture used for predicting the development of the brain. In image classification, CNNs are the recent state-of-the-art methods. The CNNs learned about natural images, showing strong performance and encountering the accuracy of human **expert systems**. Finally, these statements conclude that CNNs can be improved to control the essential architecture of medical images.

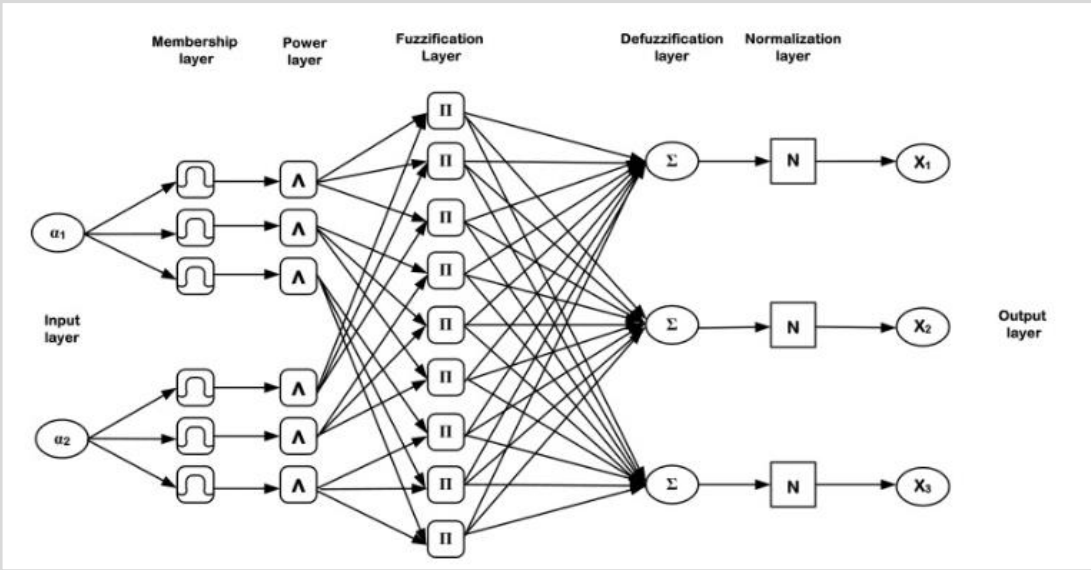


Figure 32 CNN Architecture

The figure is Hybrid computer-aided classification system design using end-to-end CNN-based deep feature extraction and ANFC-LH classifier for chest radiographs.

14.D Definition of Transfer Learning

Devising learning algorithms in the transfer learning setting is natural for human beings. By observing or experiencing one thing, humans can easily apply the learned knowledge to a different but related problem. While broad definitions of transfer learning, including domain adaptation, multitask learning, and others, have been studied in various disciplines such as statistics, machine learning, and computer vision, transfer learning itself is not a new concept. Using knowledge from the source task to improve learning the target task has a long history: it has been even independently invented many times in the studies of statistical learning, psychometrics, and cognitive psychology. In parallel, in the machine learning and computer vision community, earlier notions of transfer learning were called instance-based learning, incremental learning, and framework integration, which are studies on transfer learning without using many algorithms we currently associate with transfer learning.

In statistics, transfer learning refers to the situation in which what has been learned in one setting is exploited to improve generalization in another setting. In specific domains, the definition is formalized in a related but different way. To solve the target learning problem, we have the source learning problem allowing to help the target learning, which is particularly investigated by a probabilistic model-based framework. For the supervised learning task, machine learning techniques, theoretically, need training and test data following the same input and output distributions. However, in reality, the training and test data may be from different feature spaces or probability distributions. The assumption that training examples and test examples are independent and identically distributed (i.i.d.) is a standard assumption made in many statistical learning problems. Unfortunately, in some cases, this assumption may not always hold: when a domain is changed, transfer learning is required.

14.E How Transfer Learning Works?

In computer vision, neural networks typically aim to detect edges in the first layer, forms in the middle layer, and task-specific features in the latter layers.

The early and central layers are employed in transfer learning, and the latter layers are only retrained. It makes use of the labelled data from the task it was trained on.

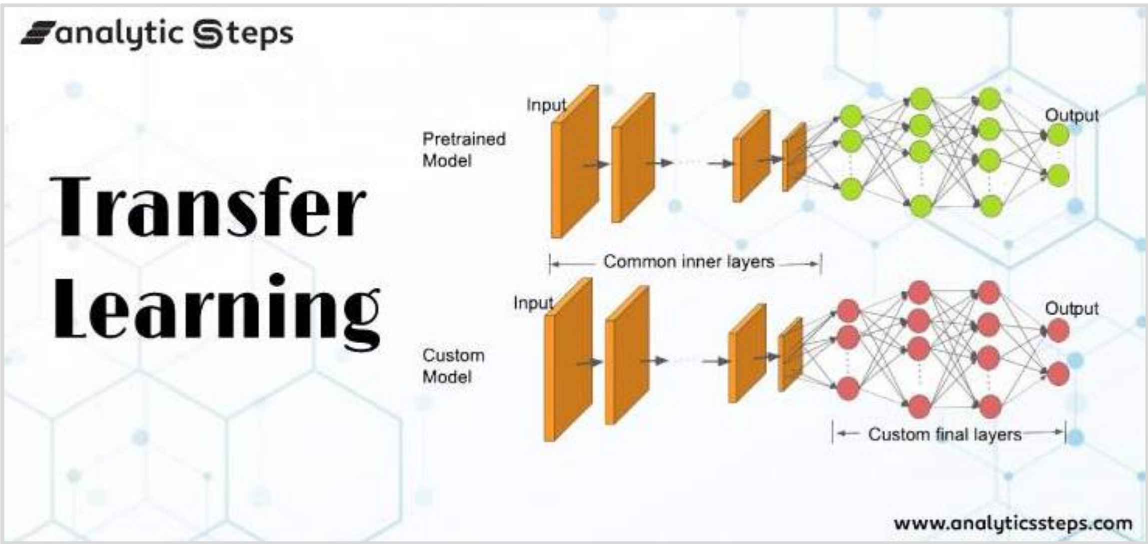


Figure 33 Transfer Learning

14.F Steps to Use Transfer Learning

1. Training a Model to Reuse it

Consider the situation in which we wish to tackle Task A but lack the necessary data to train a deep neural network. Finding a related task B with a lot of data is one method to get around this. Utilize the deep neural network to train on task B and then use the model to solve task A.

The problem we are seeking to solve will decide whether we need to employ the entire model or just a few layers. If the input in both jobs is the same, we might reapply the model and make predictions for my new input. Changing and retraining distinct task-specific layers and the output layer, on the other hand, is an approach to investigate.

2. Using a Pre Trained Model

The second option is to employ a model that has already been trained. There are a number of these models out there, so do some research beforehand. The number of layers to reuse and retrain is determined by the task. Keras consists of nine pre-trained models used in transfer learning, prediction, fine-tuning. These models, as well as some quick lessons on how to utilise them, may be found here.

Many research institutions also make trained models accessible. The most popular application of this form of transfer learning is deep learning.

14.G Features Extraction

Another option is to utilize deep learning to identify the optimum representation of my problem, which comprises identifying the key features. This method is known as representation learning, and it can often produce significantly better results than hand-designed representations. Feature creation in machine learning is mainly done by hand by researchers and domain specialists. Deep learning, fortunately, can extract features automatically. Of course, this does not diminish the importance of feature engineering and domain knowledge; we must still choose which features to include in our network.

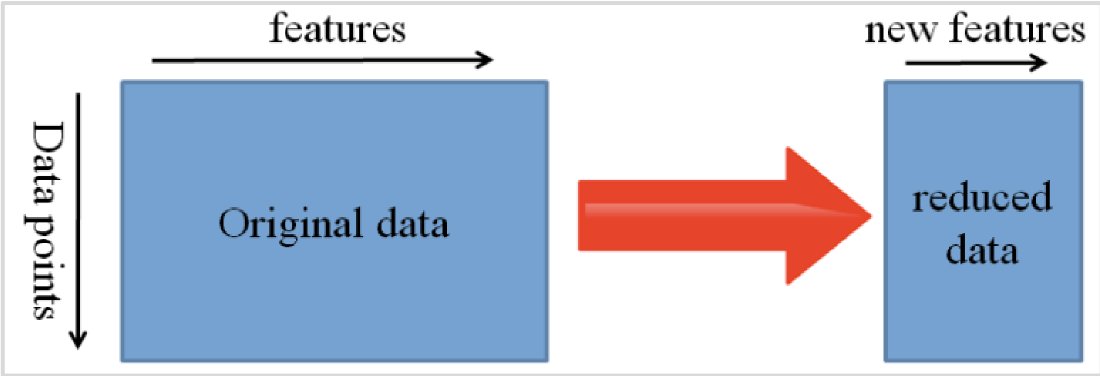


Figure 34 Feature extraction

G.1 Extraction of Features in Neural Networks

Neural networks, on the other hand, have the ability to learn which features are critical and which aren't. Even for complicated tasks that would otherwise necessitate a lot of human effort, a representation learning algorithm can find a decent combination of characteristics in a short amount of time.

The learned representation can then be applied to a variety of other challenges. Simply utilise the initial layers to find the appropriate feature representation but avoid using the network's output because it is too task specific. Instead, send data into our network and output it through one of the intermediate layers. The raw data can then be understood as a representation of this layer.

This method is commonly used in computer vision since it can shrink your dataset, reducing computation time and making it more suited for classical algorithms.

G.2 What is a Pre-trained Model?

A pre-trained model has been previously trained on a dataset and contains the weights and biases that represent the features of whichever dataset it was trained on. Learned features are often transferable to different data. For example, a model trained on a large dataset of bird images will contain learned features like edges or horizontal lines that you would be transferable your dataset.

G.3 Why use a Pre-trained Model?

Pre-trained models are beneficial to us for many reasons. By using a pre-trained model, you are saving time. Someone else has already spent the time and compute resources to learn a lot of features and your model will likely benefit from it.

14.H Chest X-Ray model Tools

H.1 Jupyter

Jupyter is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It's a powerful tool for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Jupyter is composed of several components, including:

- Jupyter Notebook: An interactive document that contains code, text, and other elements.
- Jupyter Kernel: A program responsible for handling various types of requests (code execution, code completions, inspection), and providing a computational engine for Jupyter.
- JupyterHub: A multi-user version of the notebook designed for companies, classrooms and research labs.

H.2 TensorFlow

TensorFlow makes it easy for beginners and experts to create machine learning models for desktop, mobile, web, and cloud. See the sections below to get started.

- Easy model building
  TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy. If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

- Robust Machine learning production

  TensorFlow has always provided a direct path to production. Whether it's on servers, edge devices, or the web, TensorFlow lets you train and deploy your model easily, no matter what language or platform you use. Use TFX if you need a full production ML pipeline. For running inference on mobile and edge devices, use TensorFlow Lite. Train and deploy models in JavaScript environments using TensorFlow.js.

- Powerful experimentation for research

  Build and train state-of-the-art models without sacrificing speed or performance. TensorFlow gives you the flexibility and control with features like the Keras Functional API and Model Subclassing API for creation of complex topologies. For easy prototyping and fast debugging, use eager execution.

  TensorFlow also supports an ecosystem of powerful add-on libraries and models to experiment with, including Ragged Tensors, TensorFlow Probability, Tensor2Tensor and BERT.

14.I Prepare and load data for successful ML outcomes

Data can be the most important factor in the success of your ML endeavors. TensorFlow offers multiple data tools to help you consolidate, clean and preprocess data at scale:

- Standard datasets for initial training and validation
- Highly scalable data pipelines for loading data
- Preprocessing layers for common input transformations
- Tools to validate and transform large datasets

Additionally, responsible AI tools help you uncover and eliminate bias in your data to produce fair, ethical outcomes from your models.

14.J Build and fine-tune models with the TensorFlow ecosystem

Explore an entire ecosystem built on the **Core framework** that streamlines model construction, training, and export. TensorFlow supports distributed training, immediate model iteration and easy debugging with **Keras**, and much more. Tools like model Analysis and TensorBoard help you track development and improvement through your model's lifecycle. To help you get started, find collections of pre-trained models at TensorFlow Hub from Google and the community, or implementations of state-of-the art research models in the Model Garden. These libraries of high-level components allow you to take powerful models and fine-tune them on new data or customize them to perform new tasks.

Chapter 5. Implementation

1. System setup

To implement the healthcare monitoring system, several hardware and software components were utilized. This chapter outlines the setup process, configuration details, and code implementation for both the lung disease and heart attack detection models. For complete source code and additional resources, please visit the project's **GitHub** repository [31].

2. Hardware Configuration

The primary hardware components used in this project include the Raspberry Pi 4 Model B, a compatible camera module, the MAX30100 pulse oximeter, and the MLX90614 temperature sensor. The Raspberry Pi 4 Model B serves as the central processing unit, managing data acquisition and processing tasks.

Hardware Components:

- Raspberry Pi 4 Model B
- Camera Module
- MAX30100 Pulse Oximeter
- MLX90614 Temperature Sensor

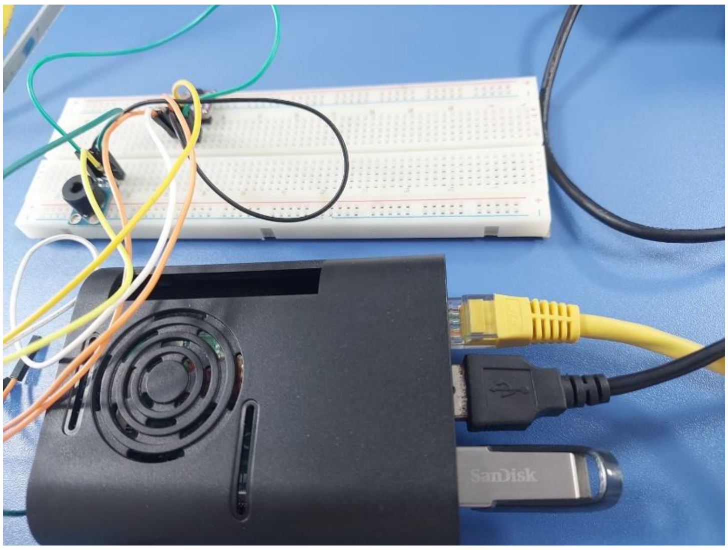


Figure 35 Hardware setup for the healthcare monitoring system.

3. Software Environment

The software environment includes the Raspbian operating system, Python programming language, and several libraries essential for data processing, image analysis, and communication with sensors.

Software Components:

- Raspbian OS
- Python 3
- Numpy
- Pandas
- Matplotlib and Seaborn
- Scikit-learn
- ML Algorithms like SVM, Random Forest, Logistic Regression
- Imbalanced-learn
- OpenCV
- Firebase Admin SDK
- TensorFlow/Keras

To set up the software environment on a local machine, follow these steps:

3.A Installing Anaconda

Anaconda is a popular distribution for Python and R that simplifies package management and deployment. Download and install Anaconda from the official website.

3.B Creating a Conda Environment

Create a new Conda environment with the required packages:

```
conda create -n healthcare_env python=3.8
conda activate healthcare_env
```

3.C Installing Required Libraries

Use Conda and pip to install the required libraries:

```
conda install -c conda-forge opencv
conda install -c conda-forge tensorflow
conda install -c conda-forge imbalanced-learn
pip install joblib
pip install scikit-learn
```

If you're not using Conda, you can install all necessary packages using pip:

```
pip install opencv-python tensorflow imbalanced-learn firebase-
admin joblib sciki
```

4. Multi-Label Chest X-Ray Classification via Deep Learning

Chest radiography is the most common type of imaging examination in the world, with over 2 billion procedures performed each year. This technique is critical for screening, diagnosis, and management of thoracic diseases, many of which are among the leading causes of mortality worldwide. A computer system to interpret chest radiographs as effectively as practicing radiologists could thus provide substantial benefit in many clinical settings, from improved workflow prioritization and clinical decision support to large-scale screening and global population health initiatives.

Recent advancements in deep learning and large datasets have enabled algorithms to match the performance of medical professionals in a wide variety of other medical imaging tasks, including diabetic retinopathy detection, skin cancer classification , and lymph node metastases detection . Automated diagnosis from chest imaging has received increasing attention , with specialized algorithms developed for pulmonary tuberculosis classification and lung nodule detection, but the use of chest radiographs to discover other pathologies such as pneumonia and pneumothorax motivates an approach that can detect multiple pathologies simultaneously. Only recently have the computational power and availability of large datasets enabled the development of such an approach. The National Institutes of Health's release of ChestX-ray14 led to many more studies that use deep learning for chest radiograph diagnosis. However, the performance of these algorithms has not been compared to that of practicing radiologists.

In this work, we aimed to assess the performance of a deep learning algorithm to automatically interpret chest radiographs. We developed a deep learning algorithm to concurrently detect the presence of 14 different disease classes in chest radiographs and evaluated its performance against practicing radiologists.

4.A NIH Chest X-ray Dataset

National Institutes of Health Chest X-Ray Dataset

Chest X-ray exams are one of the most frequent and cost-effective medical imaging examinations available. However, clinical diagnosis of a chest X-ray can be challenging and sometimes more difficult than diagnosis via chest CT imaging. The lack of large publicly available datasets with annotations means it is still very difficult, if not impossible, to achieve clinically relevant computer-aided detection and diagnosis (CAD) in real world medical sites with chest X-rays. One major hurdle in creating large X-ray image datasets is the lack resources for labeling so many images. Prior to the release of this dataset, Openi was the largest publicly available source of chest X-ray images with 4,143 images available.

This NIH Chest X-ray Dataset is comprised of 112,120 X-ray images with disease labels from 30,805 unique patients. To create these labels, the authors used Natural Language Processing to text-mine disease classifications from the associated radiological reports. The labels are expected to be >90% accurate and suitable for weakly-supervised learning. The original radiology reports are not publicly available but you can find more details on the labeling process in this Open Access paper: "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases." (Wang et al.) [8]

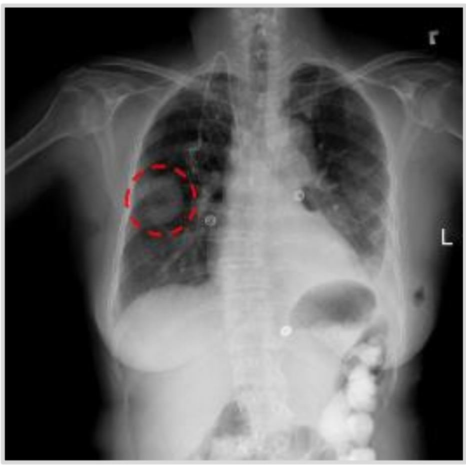


Figure 36 A chest x-ray identifies a lung mass.

4.B Data limitations:

- The image labels are NLP extracted so there could be some erroneous labels but the NLP labeling accuracy is estimated to be >90%.
- Very limited numbers of disease region bounding boxes (See BBox_list_2017.csv)
- Chest x-ray radiology reports are not anticipated to be publicly shared. Parties who use this public dataset are encouraged to share their “updated” image labels and/or new bounding boxes in their own studied later, maybe through manual annotation

4.C Class descriptions

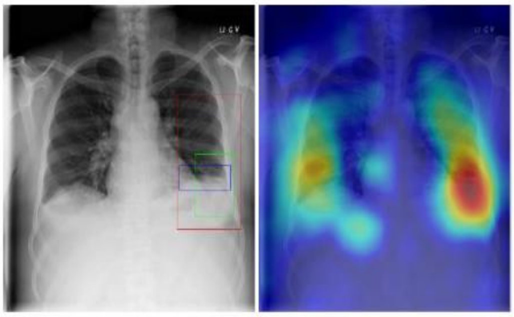
There are 15 classes (14 diseases, and one for "No findings"). Images can be classified as "No findings" or one or more disease classes:

1. Atelectasis
2. Consolidation
3. Infiltration
4. Pneumothorax
5. Edema
6. Emphysema
7. Fibrosis
8. Effusion
9. Pneumonia
10. Pleural_thickening
11. Cardiomegaly
12. Nodule Mass
13. Hernia

4.D Full Dataset Content

There are 12 zip files in total and range from ~2 gb to 4 gb in size. Additionally, we randomly sampled 5% of these images and created a smaller dataset for use in Kernels. The random sample contains 5606 X-ray images and class labels.

Table 3 Chest X-ray Radiology report

Radiology report	Keyword	Localization Result
findings include: 1. left basilar atelectasis/consolidation. 2. prominent hilum (mediastinal adenopathy). 3. left pic catheter (tip in atriocaval junction). 4. stable, normal appearing cardiomediastinal silhouette. impression: small right pleural effusion otherwise stable abnormal study including left basilar infiltrate/atelectasis, prominent hilum, and position of left pic catheter (tip atriocaval junction).	Effusion; Infiltration; Atelectasis	

A sample of chest x-ray radiology report mined disease keywords and localization result from the "Atelectasis" Class. Correct bounding. box (in green), false positives (in red) and ground truth (in blue) are plotted over the original image.

4.E Import Packages and Functions

We'll make use of the following packages:

- numpy and pandas is what we'll use to manipulate our data
- matplotlib.pyplot and seaborn will be used to produce plots for visualization
- util will provide the locally defined utility functions that have been provided for this assignment

We will also use several modules from the Keras framework for building deep learning models.

4.F Load the Datasets

The ChestX-ray14 dataset [5] was used to develop the deep learning algorithm. The dataset is currently the largest public repository of radiographs, containing 112,120 frontal-view (both posteroanterior and anteroposterior) chest radiographs of 30,805 unique patients. Each image in ChestX-ray14 was annotated with up to 14 different thoracic pathology labels that were chosen based on frequency of observation and diagnosis in clinical practice. The labels for each image were obtained using automatic extraction methods on radiology reports, resulting in 14 binary values per image, where 0 indicates the absence of that pathology and 1 denotes the presence (multiple pathologies can be present in each image). We partitioned the dataset into training, tuning, and validation.

- Each image in the data set contains multiple text-mined labels identifying 14 different pathological conditions.
- These in turn can be used by physicians to diagnose 8 different diseases.
- We will use this data to develop a single model that will provide binary classification predictions for each of the 14 labeled pathologies.
- In other words, it will predict 'positive' or 'negative' for each of the pathologies.

we downloaded the entire dataset for free [6].

4.G Data Visualization

We can Display Random Images

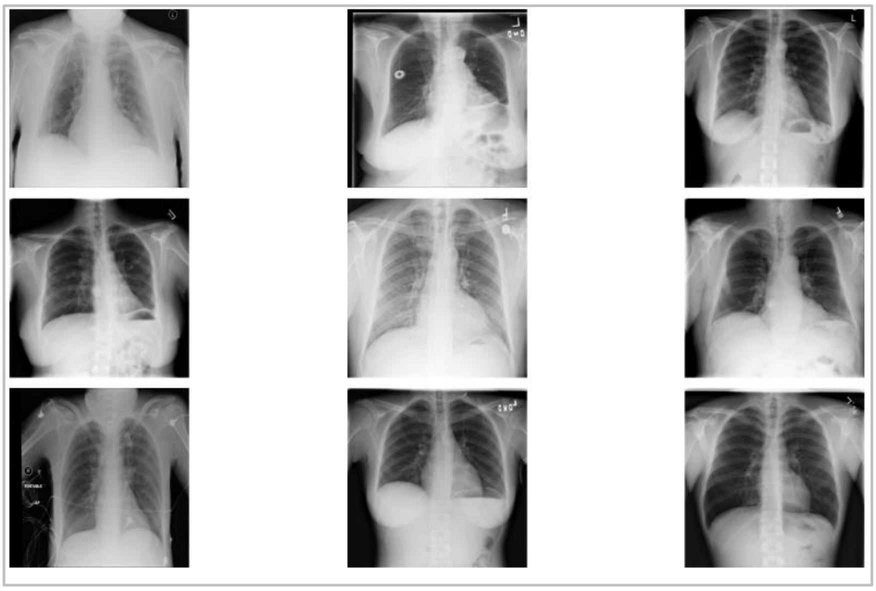


Figure 37 A chest x-ray identifies a lung mass.

4.H Investigate a single image

Look at the first image in the dataset and print out some details of the image contents. The dimensions of the image are **1024** pixels width and **1024** pixels height, one single color channel The maximum pixel value is **0.9804** and the minimum is **0.0000** The mean value of the pixels is **0.4796** and the standard deviation is **0.2757**

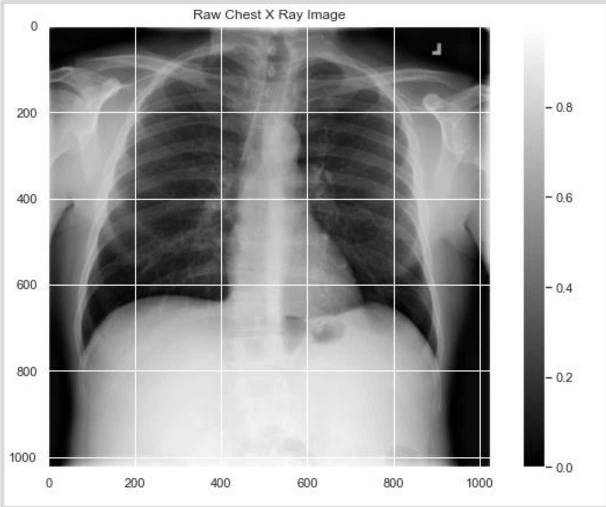


Figure 38 3 A chest x-ray image

4.I Preventing Data Leakage

It is worth noting that our dataset contains multiple images for each patient. This could be the case, for example, when a patient has taken multiple X-ray images at different times during their hospital visits. In our data splitting, we have ensured that the split is done on the patient level so that there is no data "leakage" between the train, validation, and test datasets.

I.1 Preparing Images

With our dataset splits ready, we can now proceed with setting up our model to consume them.

- For this we will use the off-the-shelf ImageDataGenerator class from the Keras framework, which allows us to build a "generator" for images specified in a DataFrame.
- This class also provides support for basic data augmentation such as random horizontal flipping of images.
- We also use the generator to transform the values in each batch so that their mean is 00 and their standard deviation is 1.
  This will facilitate model training by standardizing the input distribution.
- The generator also converts our single channel X-ray images (grayscale) to a three-channel format by repeating the values in the image across all channels.

We will want this because the pre-trained model that we'll use requires three-channel inputs.

Since it is mainly a matter of reading and understanding Keras documentation, we have implemented the generator for you. There are a few things to note:

1. We normalize the mean and standard deviation of the data.
2. We shuffle the input after each epoch.
3. We set the image size to be 320px by 320px.

I.2 Build a separate generator for valid and test sets

Now we need to build a new generator for validation and testing data.

Why can't we use the same generator as for the training data?

Look back at the generator we wrote for the training data.

a. It normalizes each image per batch, meaning that it uses batch statistics.
b. We should not do this with the test and validation data, since in a real life scenario we don't process incoming images a batch at a time (we process one image at a time).
c. Knowing the average per batch of test data would effectively give our model an advantage.

The model should not have any information about the test data.

What we need to do is normalize incoming test data using the statistics computed from the training set.

- There is one technical note. Ideally, we would want to compute our sample mean and standard deviation using the entire training set.
- However, since this is extremely large, that would be very time consuming.
- In the interest of time, we'll take a random sample of the dataset and calcualte the sample mean and sample standard deviation.

what the generator gives our model during training and validation. We can do this by calling the __get_item__(index) function:

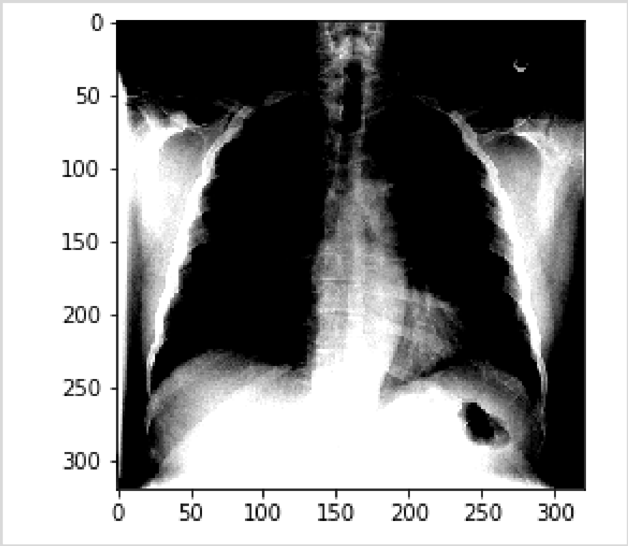


Figure 39 processed chest x-ray image.

4.J Model Development

Now we'll move on to model training and development. We have a few practical challenges to deal with before actually training a neural network, though. The first is class imbalance.

4.K Addressing Class Imbalance

One of the challenges with working with medical diagnostic datasets is the large class imbalance present in such datasets.

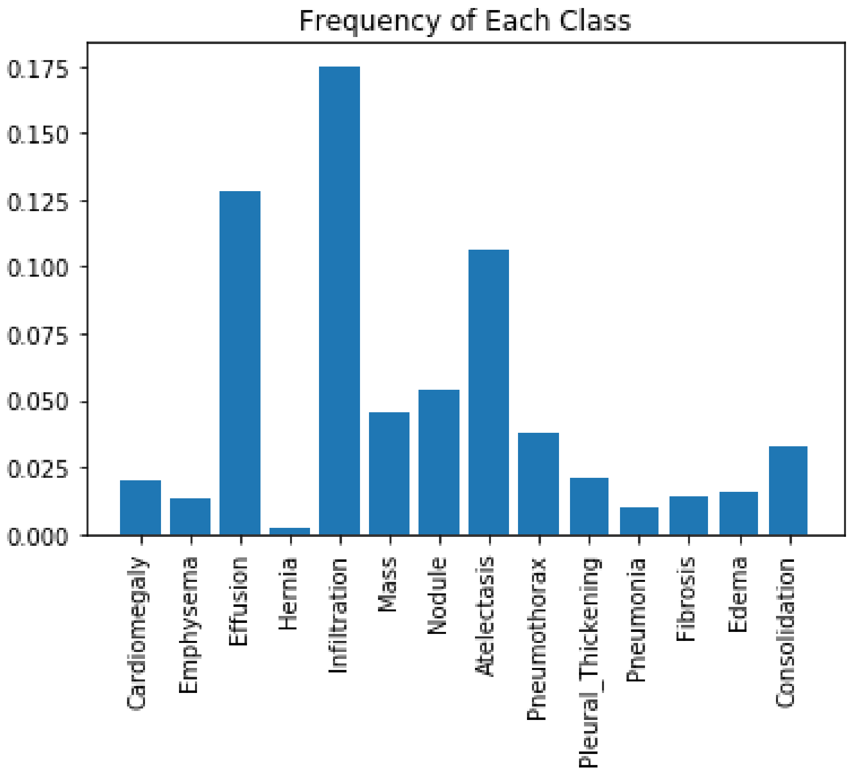


Figure 40 plot the frequency of each of the labels in our dataset

We can see from this plot that the prevalence of positive cases varies significantly across the different pathologies. (These trends mirror the ones in the full dataset as well.)

- The Hernia pathology has the greatest imbalance with the proportion of positive training cases being about 0.2%.
- But even the Infiltration pathology, which has the least amount of imbalance, has only 17.5% of the training cases labelled positive.

Ideally, we would train our model using an evenly balanced dataset so that the positive and negative training cases would contribute equally to the loss.

If we use a normal cross-entropy loss function with a highly unbalanced dataset, as we are seeing here, then the algorithm will be incentivized to prioritize the majority class (i.e. negative in our case), since it contributes more to the loss.

4.L Impact of class imbalance on loss function

we Assume that we would have used a normal cross-entropy loss for each pathology.

We recall that the cross-entropy loss contribution from the *ith* training data case is:

$$\mathcal{L}_{cross-entropy}(x_i) = -(y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i))),$$

where xi and yi are the input features and the label, and $f(x_i)$ is the output of the model, i.e. the probability that it is positive.

And for any training case, either y_i=0 or else (1−y_i)=0, so only one of these terms contributes to the loss (the other term is multiplied by zero, and becomes zero).

We can rewrite the overall average cross-entropy loss over the entire training set $\mathcal{D}$ of size N as follows:

$$\mathcal{L}_{cross-entropy}(\mathcal{D}) = -\frac{1}{N}\Big(\sum_{\text{positive examples}} \log(f(x_i)) + \sum_{\text{negative examples}} \log(1 - f(x_i)) \Big).$$

Using this formulation, we can see that if there is a large imbalance with very few positive training cases, for example, then the loss will be dominated by the negative class. Summing the contribution over all the training cases for each class (i.e. pathological condition), we see that the contribution of each class (i.e. positive or negative) is:

$$freq_p = \frac{\text{number of positive examples}}{N}$$

and

$$freq_n = \frac{\text{number of negative examples}}{N}.$$

4.M Computing Class Frequencies

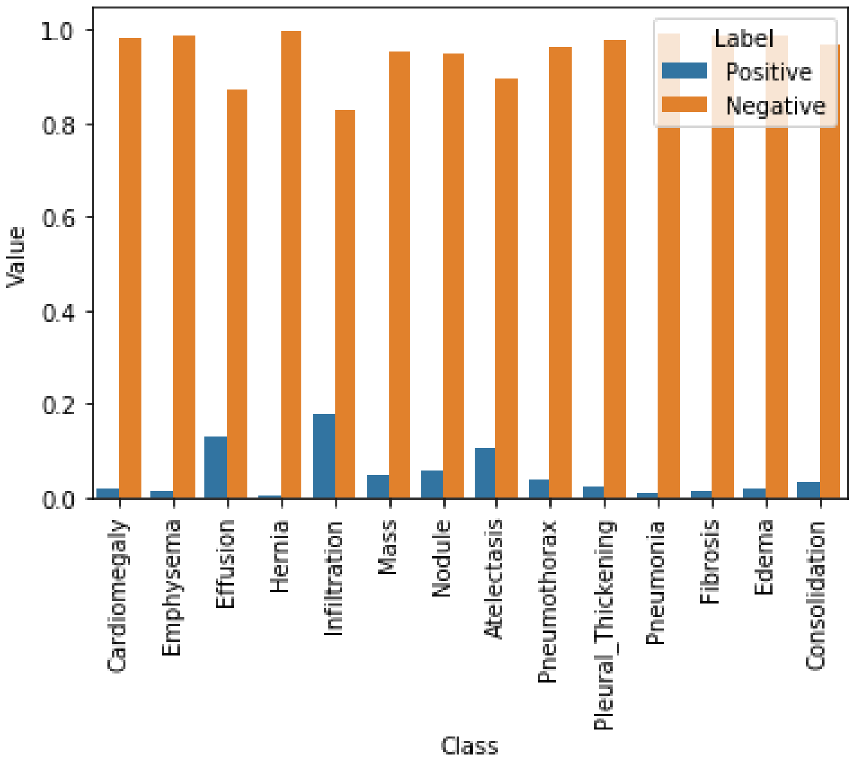


Figure 41 plot Class Frequencies

The contributions of positive cases is significantly lower than that of the negative ones. However, we want the contributions to be equal. One way of doing this is by multiplying each example from each class by a class-specific weight factor, w_{pos} and w_{neg}, so that the overall contribution of each class is the same

To have this, we want

$$w_{pos} \times freq_p = w_{neg} \times freq_n,$$

which we can do simply by taking

$$w_{pos} = freq_{neg}$$

$$w_{neg} = freq_{pos}$$

This way, we will be balancing the contribution of positive and negative labels.

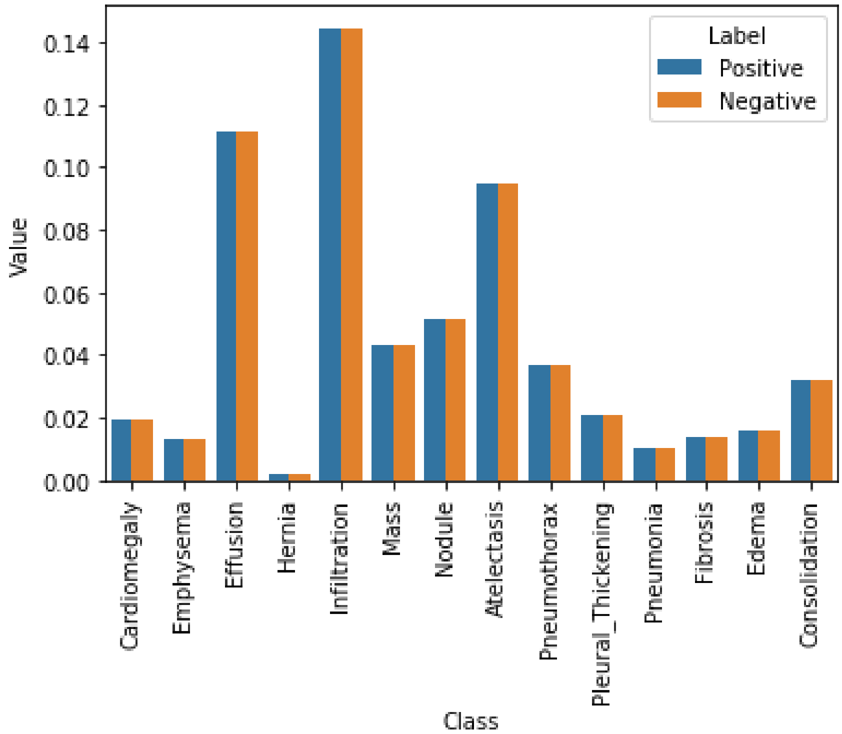


Figure 42 plot Class Frequencies balancing

As the above figure shows, by applying these weightings the positive and negative labels within each class would have the same aggregate contribution to the loss function. Now let's implement such a loss function.

After computing the weights, our final weighted loss for each training case will be

$$\mathcal{L}^{w}_{cross-entropy}(x) = -(w_p y \log(f(x)) + w_n(1-y)\log(1-f(x))).$$

4.N Best performing algorithm

N.1 DenseNet121

Use a pre-trained **DenseNet121** model. Densenet is a convolutional network where each layer is connected to all other layers that are deeper in the network

- The first layer is connected to the 2nd, 3rd, 4th etc.
- The second layer is connected to the 3rd, 4th, 5th etc.

N.2 Densely Connected Convolutional Networks

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections - one between each layer and its subsequent layer - our network has L(L+1)/2 direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. We evaluate our proposed architecture on four highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet). DenseNets obtain significant improvements over the state-of-the-art on most of them, whilst requiring less memory and computation to achieve high performance. [7]

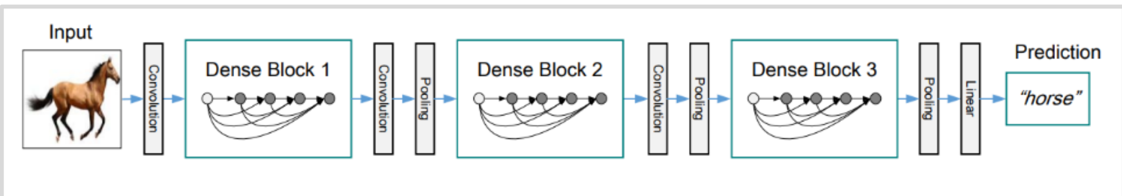


Figure 43 A deep DenseNet with three dense blocks.

The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

N.3 DenseNet Architectures

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Table 4 DensNet Architecture for ImageNet

DenseNet architectures for ImageNet. The growth rate for all the networks is k = 32. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

After we are using a pre-trained DenseNet121 model which we can load directly from Keras and then add two layers on top of it:

a. A GlobalAveragePooling2D layer to get the average of the last convolution layers from DenseNet121.
b. A Dense layer with sigmoid activation to get the prediction logits for each of our classes.

We set our custom loss function for the model by specifying the loss parameter in the compile () function.

4.O Image Classification and Class Imbalance

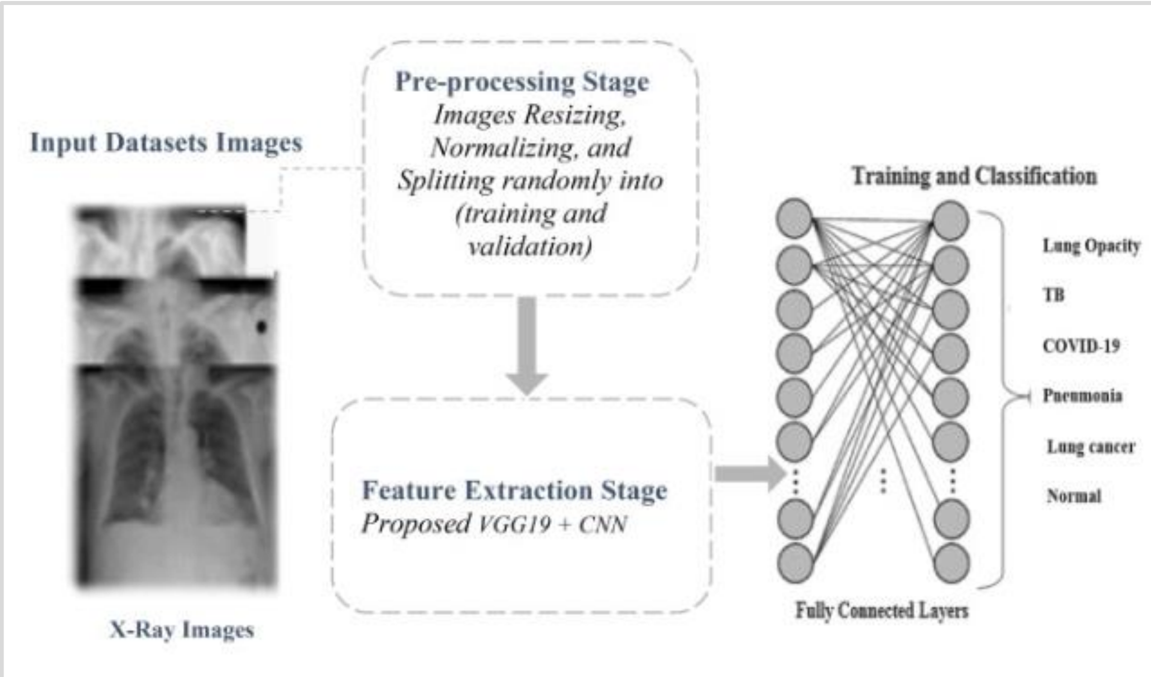


Figure 44 Image Classification and Class Imbalance

4.P Transfer Learning (pretraining Fine tuning)

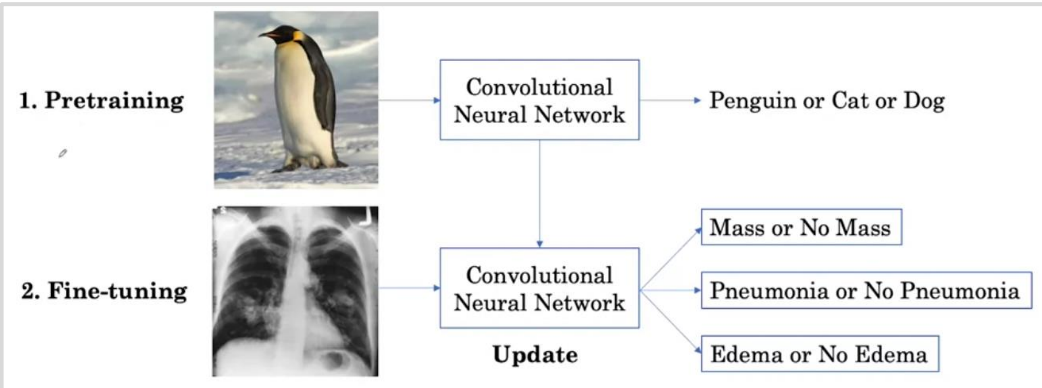


Figure 45 Transfer learning

4.Q Training procedure

The training process consisted of 2 consecutive stages to account for the partially incorrect labels in the ChestX-ray14 dataset. First, multiple networks were trained on the training set to predict the probability that each of the 14 pathologies is present in the image. Then, a subset of those networks, each chosen based on the average error on the tuning set, constituted an ensemble that produced predictions by computing the mean over the predictions of each individual network. The ensemble was used to relabel the training and tuning sets as follows: first, the ensemble probabilities were converted to binary values by computing the threshold that led to the highest average F1 score on the tuning set across all pathologies. Then, the new label was taken to be positive if and only if either the original label was positive, or the ensemble prediction was positive. Finally, new networks were trained on the relabeled training set, and a subset of the new networks was selected based on the average error on the relabeled tuning set. The final network was an ensemble of 10 networks trained on the relabeled data, where again the predictions of the ensemble were computed as the mean over the predictions of each individual network.

Before both stages of training, the parameters of each network were initialized with parameters from a network pretrained on ImageNet [9]. The final fully connected layer of the pretrained network was replaced with a new fully connected layer producing a 14-dimensional output, after which the sigmoid was applied to each of the outputs to obtain the predicted probabilities of the presence of each of the 14 pathology classes. Before inputting the images into the network, the images were resized to 320 pixels by 320 pixels and normalized based on the mean and standard deviation (SD) of images in the ImageNet training set. For each image in the training set, a random lateral inversion was applied with 50% probability before being fed into the network. The networks were updated to minimize the sum of per-class weighted binary cross entropy losses, where the per-class weights were computed based on the prevalence of that class in the training set. All parameters of the networks were trained jointly using Adam with standard parameters. Adam is an effective variant of an optimization algorithm called stochastic gradient descent, which iteratively applies updates to parameters to minimize the loss during training. We trained the networks with minibatches of size 8 and used an initial learning rate of 0.0001 that was decayed by a factor of 10 each time the loss on the tuning set plateaued after an epoch (a full pass over the training set). To prevent the networks from overfitting, early stopping was performed by saving the network after every epoch and choosing the saved network with the lowest loss on the tuning set. No other forms of regularization, such as weight decay or dropout, were used. Each stage of training completed after around 20 hours on a single NVIDIA GeForce GTX TITAN Black. Each network had 6,968,206 learnable parameters, and the final ensemble had 69,682,060 parameters.

4.R Training, Prediction, and Loss

During training, an algorithm is shown images of chest X-rays labeled with whether they contain a mass or not. The algorithm learns using these images and labels. The algorithm eventually learns to go from a chest X-ray input to produce the output of whether the X-ray contains mass. And this algorithm can go by different names.

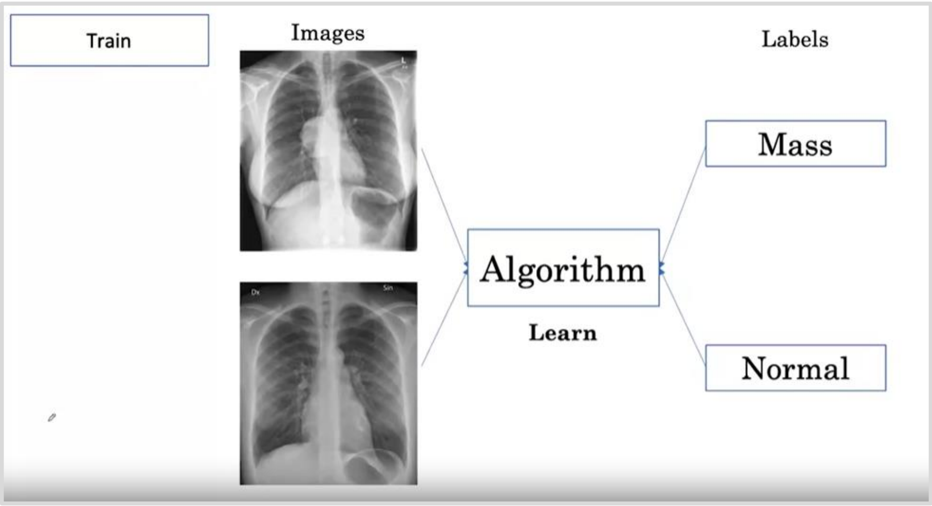


Figure 46 training the datas

The model architecture for our pre-trained model is exactly the same, but we used a few useful Keras "callbacks" for this training. Do spend time to read about these callbacks at your leisure as they will be very useful for managing long-running training sessions:

a. we can use ModelCheckpoint callback to monitor our model's val_loss metric and keep a snapshot of your model at the point.
b. we can use the TensorBoard to use the Tensorflow Tensorboard utility to monitor our runs in real-time.

c. we can use the ReduceLROnPlateau to slowly decay the learning rate for our model as it stops getting better on a metric such as val_loss to fine-tune the model in the final steps of training.
d. we can use the EarlyStopping callback to stop the training job when our model stops getting better in it's validation loss. we can set a patience value which is the number of epochs the model does not improve after which the training is terminated. This callback can also conveniently restore the weights for the best metric at the end of training to your model.

5. Setting Up the Raspberry Pi

The initial step involves setting up the Raspberry Pi with the necessary operating system and libraries.

5.A Installing Raspbian OS

Download the latest version of Raspbian OS from the official Raspberry Pi website and use tools like Balena Etcher to flash the image onto an SD card. Insert the SD card into the Raspberry Pi and power it up.

5.B Environment setup

B.1 upgrade the system packages

```
sudo apt update
sudo apt upgrade -y
```

B.2 Install Python and pip

```
sudo apt install python3 python3-pip -y
```

B.3 Virtual Environment

Before running a Python script, it is important to set up a virtual environment. This keeps dependencies required by different projects in separate places by creating virtual Python environments for them. Here's how to set up a virtual environment:

5.C Installing Required Libraries

Install the required Python libraries:

C.1 OpenCV for image processing

```
pip3 install opencv-python
```

C.2 Keras to facilitates the building, training, and inference of deep learning models

```
Pip install keras
```

C.3 Tensorflow

```
pip install tensorflow
```

C.4 Pillow (PIL Fork)

Used for opening, manipulating, and saving many different images file formats.

```
pip install pillow
```

6. Camera Module

To use a USB webcam with your Raspberry Pi, you can utilize various packages and libraries that allow you to control and interact with the webcam. One of the most common packages for this purpose is **fswebcam**, which is a simple and straightforward tool to capture images from a USB webcam.

6.A Package installation code

```
sudo apt-get install fswebcam
```

6.B Capture an Image

To take a picture and save the captured image as image.jpg.

```
fswebcam -r 640x480 --no-banner /path/to/your/image.jpg
```

7. Firebase Integration

Firebase is a comprehensive app development platform provided by Google. It offers a suite of backend cloud computing services and tools to help developers build, manage, and grow their applications.

7.A firebase-admin for database interaction

```
pip install firebase-admin
```

7.B Get Your Credentials JSON File

a. Go to the Google Cloud IAM & Admin Console.
b. Select your project and create a new service account.
c. Download the credentials JSON file for the service account.

7.C Initialize Firebase Admin SDK

Use the credentials JSON file to initialize the Firebase Admin SDK in your Python script.

```
import firebase_admin
from firebase_admin import credentials, storage
cred = credentials.Certificate("path/to/your/credentials.json")
firebase_admin.initialize_app(cred, {"storageBucket": "your-app-
name.appspot.com"}
```

7.D Captured image uploading into a firebase cloud storage

Firebase provides a cloud storage service where you can store files. You'll need to set up Firebase for your project if you haven't already.

1. Define the local path to your image file.

```
local_image_path = "/path/to/your/image.jpg"
```

2. Create a reference to the Firebase Storage bucket.

```
bucket = storage.bucket()
```

3. Specify the desired storage path within the bucket.

```
storage_path = "images/my-image.jpg"
```

4. Upload the image.

```
Blob = bucket.blob(storage_path)
blob.upload_from_filename(local_image_patyh)
print (f"Image uploaded to Firebase Storage: {blob.public_url}"
```

7.E Access the Image

The uploaded image can be accessed via its public URL provided by Firebase Storage.

8. Robot design

To enhance user experience with our healthcare solution, it is important to minimize the complexity and technical involvement for patients. By designing a humanoid robot, we can integrate all electronic components and wiring within a user-friendly and visually appealing exterior. This ensures that patients do not need to handle or even see electronic parts, such as jumpers or sensors, making the interaction with the healthcare solution straightforward and intuitive. The humanoid design simplifies the user experience and increases acceptance of advanced technology in healthcare settings, ultimately improving patient compliance and satisfaction.

8.A Sketches

Introduction to Sketching: Sketching is a fundamental process that bridges imagination and reality. Whether you’re an artist, designer, or simply someone with creative ideas, sketching allows you to express your thoughts visually.

A.1 Conceptual Sketching:

- **The Art of Rough Ideas:** Sketching involves roughly scribbling an idea on paper. It’s not about perfection; instead, it’s about capturing the essence of your concept swiftly.
- **Translating Abstract Thoughts:** Imagine your mind’s eye translating abstract thoughts into tangible lines and shapes. Sketches serve as the bridge between imagination and execution.

A.2 Why Sketch?

- **Brainstorming:** Sketches help you explore and invent concepts rapidly. They allow you to record ideas, even if they’re not fully formed.
- **Communication:** When discussing ideas with others, sketches make it easier to convey your vision. They provide a visual language that transcends words.
- **Iterative Process:** Sketches evolve. They blend in or fade away as you refine your concept. Embrace mistakes they’re part of the exploration.

A.3 Sketching steps:

Exploring Initial Designs for Our Project: Ideas and Sketches

During the early stages of our project, we embarked on a journey of creativity and problem-solving. Our goal was to find the most suitable robot design that would fulfill the project's requirements. Through brainstorming sessions and collaborative thinking, we generated a variety of preliminary shapes and concepts.

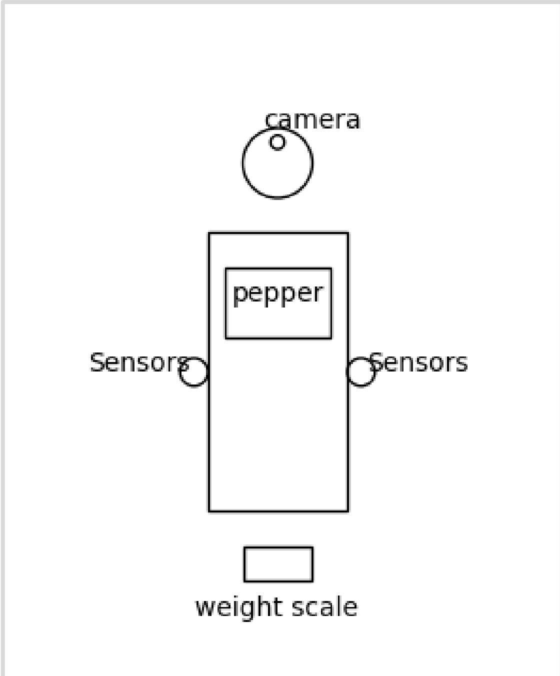


Figure 47 Initial design

These sketches represent our exploration of different possibilities, each aimed at addressing specific project objectives. As we move forward, we now shift our focus to the practical aspects how to manufacture each robot based on these initial designs.

Figure 48 A visualization of the final design 1

8.B Fiberglass

Also known as glass fiber, is a versatile material made from extremely fine fibers of glass. It's commonly used in a wide range of applications due to its exceptional properties:

B.1 Uses:

- **Construction:** Fiberglass is extensively used in construction for insulation, roofing, and reinforcement in materials like composites.
- **Automotive:** It's used in automobile bodies, as well as in components like bumpers and panels.

- **Aerospace:** Fiberglass composites are used in plane components due to their lightweight and high strength.
- **Marine:** In boat building, fiberglass is a popular material for hulls, decks, and other structural components.
- **Sports and Recreation:** Fiberglass is found in sporting equipment like fishing rods, surfboards, and hockey sticks due to its strength and flexibility.
- **Wind Energy:** Fiberglass composites are used in wind turbine blades due to their lightweight and durable nature.

B.2 Physical Properties:

- Lightweight: Fiberglass is notably lightweight, making it easy to manage and transport.
- Transparency: In its pure form, fiberglass is transparent, allowing it to be used in applications where visibility is necessary.
- Heat Resistance: Fiberglass has good thermal insulation properties, making it useful for applications requiring resistance to elevated temperatures.
- Electrical Insulation: It's an excellent electrical insulator, making it suitable for use in electrical applications.
- Chemical Resistance: Fiberglass has resistance to many chemicals, making it suitable for use in corrosive environments.

B.3 Mechanical Properties:

- Strength: Fiberglass is renowned for its high tensile strength, meaning it can withstand pulling forces without breaking.
- Flexibility: While fiberglass is strong, it also has a degree of flexibility, which allows it to bend without breaking.
- Stiffness: It has a high stiffness-to-weight ratio, providing structural integrity while keeping weight low.
- Fatigue Resistance: Fiberglass shows good fatigue resistance, making it suitable for applications subjected to repeated stress.

B.4 Chemical Properties:

- Inertness: Fiberglass is inert to many chemicals, making it resistant to corrosion and degradation.
- Water Resistance: It is generally waterproof and does not absorb water, which is essential for many outdoor applications.
- UV Resistance: Fiberglass can be formulated to resist degradation from ultraviolet (UV) radiation, prolonging its lifespan in outdoor applications.
- Acid and Alkali Resistance: It is resistant to both acids and alkalis, making it suitable for use in environments where these substances are present.

Overall, fiberglass's combination of physical, mechanical, and chemical properties makes it a highly versatile material with applications across various industries.

Figure 49 Robust frame: Crafted seamlessly from premium fiberglass.

8.C Creating Construction Planes and Sketches

C.1 Construction Planes

Construction planes in Fusion 360 are reference surfaces that serve as a foundation for creating sketches and features. These planes can be created perpendicular or parallel to existing geometry, or at specific angles. They provide visual reference and help keep design intent throughout the modeling process. Construction planes can be positioned and oriented precisely to ease the creation of complex geometry.

C.2 Sketches

Sketches in Fusion 360 are 2D profiles created on construction planes. They are used as the basis for creating 3D features such as extrusions, revolves, lofts, and sweeps. Sketches can consist of lines, arcs, circles, rectangles, and other geometric shapes. Fusion 360 offers a variety of sketching tools for creating and editing geometry, including constraints and dimensions to ensure precise control over the sketch.

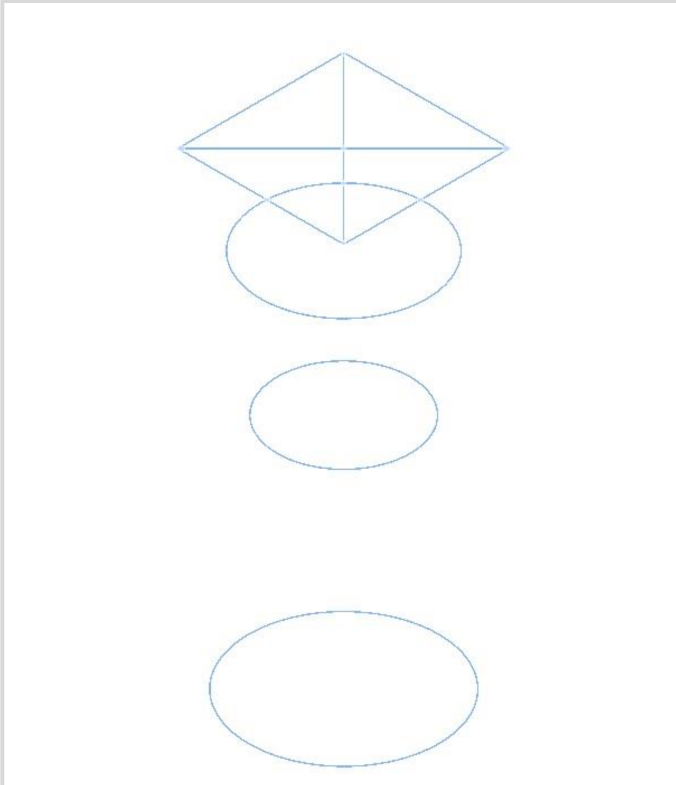


Figure 50 construction plane

8.D Adding Guide Rails

Guide rails in Fusion 360 are essentially curves or edges that influence the shape of a loft or sweep operation. They provide a path or boundary for the lofted or swept feature to follow, ensuring that the resulting geometry conforms to a specified trajectory.

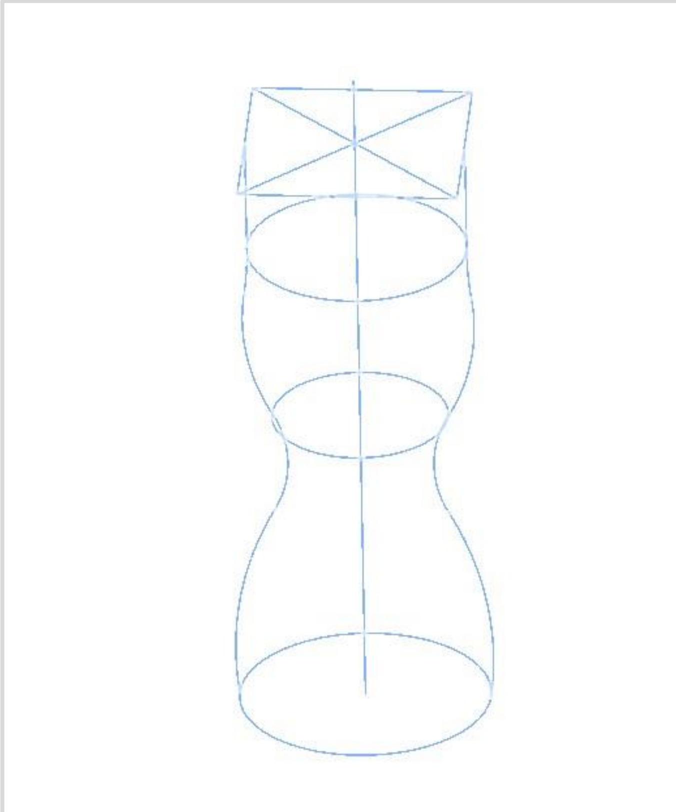


Figure 51 Take control of your creations: Guide rails in Fusion 360 for shaping lofts and sweeps.

8.E Using the Loft Feature

The Loft feature in Fusion 360 is a powerful tool used to create smooth transitions between two or more profiles. It allows users to generate complex 3D shapes by interpolating between different cross-sections or sketches.

E.1 Functionality

The Loft feature creates a surface or solid body by blending between two or more profiles along a specified path. It smoothly interpolates between the profiles, creating a continuous and organic transition between them.

E.2 Usage:

Lofting is commonly used to create shapes such as transitions, transitions between geometric features, or to create organic forms. It's particularly useful when designing products with complex curves or surfaces, such as automotive bodies, consumer products, or aerospace components.

E.3 Profiles

To use the Loft feature, users need to define two or more profiles. These profiles can be sketches drawn on different planes or existing edges or curves. Fusion 360 allows for flexibility in profile selection, enabling users to create a variety of shapes and transitions.

E.4 Path

In addition to profiles, users can specify a path along which the lofted feature will be created. The path helps define the trajectory of the loft operation, influencing the shape and orientation of the resulting geometry. The Loft feature offers options for linear, curved, or custom paths, providing control over the shape of the final design.

E.5 Controls

Fusion 360 provides controls for adjusting the loft operation, including options for controlling continuity, tangency, and curvature between profiles. These controls allow users to fine-tune the shape of the lofted feature to achieve the desired design intent.

E.6 Workflow:

Using the Loft feature typically involves selecting the profiles and path, specifying loft options and controls, and generating the lofted geometry. Fusion 360 offers a straightforward workflow for creating lofted features, making it accessible to both beginners and experienced users.

E.7 Benefits:

The Loft feature in Fusion 360 offers several benefits, including the ability to create complex shapes with ease, flexibility in profile and path selection, and precise control over the loft operation. It streamlines the design process, enabling users to iterate quickly and explore unique design variations.

8.F Employing the Shell Feature

The Shell feature in Fusion 360 is a tool used to hollow out solid bodies, leaving behind a specified wall thickness. It's commonly employed in product design, manufacturing, and engineering to reduce material usage, decrease weight, and create hollow components with internal cavities.

F.1 Functionality

The Shell feature removes material from the interior of a solid body while supporting its external geometry, effectively creating a hollowed-out shell. This process helps perfect designs by reducing weight and material usage without compromising structural integrity.

F.2 Usage

Shell operations are often applied in various industries, including automotive, aerospace, consumer products, and industrial machinery. It's used to create hollow components such as enclosures, housings, molds, and lightweight structural elements.

F.3 Wall Thickness

When using the Shell feature, users can specify the desired wall thickness for the hollowed-out section. Fusion 360 allows for precise control over the thickness of the shell, ensuring that it meets design requirements and manufacturing constraints.

F.4 Interior Features

The Shell feature preserves exterior features and surfaces while removing material from the interior of the solid body. This means that internal features, such as holes, pockets, or cavities, can be kept or changed as needed during shell operation.

F.5 Workflow

To use the Shell feature in Fusion 360, users select the solid body to be shelled, specify the desired wall thickness, and show any interior features to be preserved. Fusion 360 then automatically generates the hollowed-out shell based on the defined parameters.

F.6 Benefits

The Shell feature offers several benefits, including the ability to perfect designs for weight reduction and material savings, the preservation of external geometry and features, and the creation of internal cavities for functional purposes.

F.7 Applications

Shell operations find applications in a wide range of industries and design scenarios. They are used to create lightweight components, reduce material costs, and improve the performance and efficiency of products and systems.

Figure 52 Create Hollow Components with Precise Wall Thickness for Reduced Weight and Material Savings

8.G Applying the Extrude Cut Feature

The Extrude Cut feature in Fusion 360 is a fundamental tool used to remove material from a solid body by extruding a 2D profile or sketch along a specified direction. It's commonly employed in subtractive manufacturing processes to create holes, slots, and other cutouts in 3D models.

G.1 Functionality

The Extrude Cut feature removes material from a solid body by cutting away portions of the geometry. It uses similarly to the Extrude feature but in reverse, pushing material inward instead of outward to create voids or recesses within the model.

G.2 Usage

Extrude cuts are widely used in mechanical design, product development, and engineering to create features such as holes, pockets, channels, and complex geometries. They are essential for manufacturing processes such as milling, drilling, and CNC machining.

G.3 Profile Selection

To perform an extrude cut, users select a 2D profile or sketch that defines the shape of the cutout. This profile can be drawn directly on a face or plane of the solid body or selected from existing geometry.

G.4 Direction and Depth

Users specify the direction and depth of the extrude cut, deciding how far the selected profile will extend into the solid body. Fusion 360 provides options for extruding cuts in various directions, including symmetrically, asymmetrically, or to a specific depth.

G.5 End Conditions

Extrude cuts can be defined using different end conditions, such as blind, through-all, up to next, or up to selected. These options allow users to control where the extrude cut ends are compared to the model's geometry.

G.6 Additional Features

Fusion 360 offers more features and options for extrude cuts, including draft angles, chamfers, fillets, and taper angles. These features enable users to create complex and precise cutouts with specific design requirements.

G.7 Workflow

Using the Extrude Cut feature in Fusion 360 involves selecting the profile to be cut, specifying the direction and depth of the cut, choosing end conditions, and applying any other features or modifications as needed.

G.8 Benefits

Extrude cuts provide several benefits, including the ability to create precise cutouts and features within solid bodies, support for a wide range of geometries and end conditions, and integration with other modeling and manufacturing tools in Fusion 360.

In summary, the Extrude Cut feature in Fusion 360 is a versatile tool for subtractive manufacturing processes, allowing users to create complex cutouts and features within solid models with precision and control. It plays a crucial role in mechanical design, product development, and engineering workflows, enabling the creation of functional and manufacturable designs.

Figure 53 The Extrude Cut Feature in Fusion 360 for Creating Complex Cutouts and Features in 3D Models

8.H The Extrude Boss feature

The Extrude Boss feature in Fusion 360 is a fundamental tool used to add material to a solid body by extruding a 2D profile or sketch along a specified direction. It's a core component of the parametric modeling workflow and is commonly used in various design and engineering applications to create protrusions, ribs, and other geometric features.

H.1 Functionality

The Extrude Boss feature adds material to a solid body by extending a 2D profile or sketch outward in a specified direction. It allows users to create solid geometry by extruding profiles along linear, curved, or custom paths.

H.2 Usage

Extrude bosses are widely used in mechanical design, product development, and engineering to create features such as raised surfaces, ribs, tabs, and complex geometries. They are essential for additive manufacturing processes such as 3D printing and rapid prototyping.

H.3 Profile Selection

To perform an extrude boss, users select a 2D profile or sketch that defines the shape of the extrusion. This profile can be drawn directly on a face or plane of the solid body or selected from existing geometry.

H.4 Direction and Depth

Users specify the direction and depth of the extrude boss, deciding how far the selected profile will extend outward from the original geometry. Fusion 360 provides options for extruding bosses in various directions, including symmetrically, asymmetrically, or to a specific depth.

H.5 End Conditions

Extrude bosses can be defined using different end conditions, such as blind, through-all, up to next, or up to selected. These options allow users to control where the extrude boss ends compared to the model's geometry.

H.6 Additional Features

Fusion 360 offers added features and options for extrude bosses, including draft angles, chamfers, fillets, and taper angles. These features enable users to create complex and precise extrusions with specific design requirements.

H.7 Workflow

Using the Extrude Boss feature in Fusion 360 involves selecting the profile to be extruded, specifying the direction and depth of the extrusion, choosing end conditions, and applying any added features or modifications as needed.

H.8 Benefits

Extrude bosses provide several benefits, including the ability to create complex geometric features within solid bodies, support for a wide range of profiles and end conditions, and integration with other modeling and manufacturing tools in Fusion 360.

In summary, the Extrude Boss feature in Fusion 360 is a versatile tool for adding material to solid models, allowing users to create protrusions and geometric features with precision and control. It plays a crucial role in the design, engineering, and manufacturing workflows, enabling the creation of functional and aesthetically pleasing designs.

Figure 54 A physical model of our robot in development.

8.I The Fillet features

The Fillet feature in Fusion 360 is a tool used to create rounded or chamfered edges on solid models, enhancing their appearance, improving ergonomics, and reducing stress concentrations. It's an essential component of the design process, providing users with the ability to add aesthetic and functional details to their models. Here's a brief overview:

I.1 Functionality

The Fillet feature adds rounded or chamfered edges to sharp corners and edges of solid bodies, creating smooth transitions between adjacent faces. It helps improve the visual appeal of models, enhance their tactile feel, and perfect their structural integrity.

I.2 Usage

Fillets are widely used in mechanical design, product development, and engineering to create aesthetically pleasing designs, reduce manufacturing costs, and improve part performance. They are essential for cutting stress concentrations, improving flow characteristics, and enhancing the overall user experience.

1.3 Profile Selection

To apply a fillet, users select the edges or corners of a solid model where they want to add the rounded or chamfered features. Fusion 360 provides tools for selecting individual edges, multiple edges, edge loops, or entire faces, giving users flexibility and control over the fillet application.

1.4 Radius and Constraints

Users specify the radius or size of the fillet, figuring out the extent to which the edges or corners will be rounded or chamfered. Fusion 360 offers options for specifying fillet radii numerically or interactively, as well as constraints to ensure fillets meet specific design requirements.

1.5 Workflow

Using the Fillet feature in Fusion 360 involves selecting the edges or corners to be filleted, specifying the fillet radius or size, applying constraints if necessary, and generating the filleted geometry. Fusion 360 provides real-time feedback and visualization tools to aid users in creating precise and visually appealing fillets.

1.6 Benefits

The Fillet feature offers several benefits, including the ability to create smooth and visually pleasing transitions between faces and edges, improve part ergonomics and usability, and reduce stress concentrations and material fatigue. It enhances the overall quality and functionality of models while streamlining the design process.

In summary, the Fillet feature in Fusion 360 is a versatile tool for adding rounded or chamfered edges to solid models, enabling users to create aesthetically pleasing designs with improved performance and functionality. It's an essential component of the design workflow, providing users with the ability to refine and optimize their models for various applications.

Figure 55 Make models look nice and work better! Use round edges in Fusion 360.

8.J Robot Arms

Figure 56 A detailed 3D rendering of a non-moving robotic arm.

8.K Inserting a Shelf

Figure 57 Inner shelf: Easy access to a robot's internal components.

8.L Including Outside Doors

Figure 58 Interior access panel of a robot, showcasing internal components.

8.M Robot head

Figure 59 Abstract robot head design

8.N Slider Mechanism

A slider mechanism is a mechanical device used to create linear motion or movement along a defined path. It typically consists of two or more components that slide or glide relative to each other, enabling translation in one or more directions. Here's a brief overview:

- **Functionality:** The primary function of a slider mechanism is to facilitate linear motion by translating input force or energy into controlled movement along a specified path. It allows for back-and-forth or side-to-side motion without rotation.
- **Components:** A slider mechanism typically consists of at least two components: a base or stationary element and a sliding element. The sliding element moves relative to the base, guided by tracks, grooves, or other guiding features to ensure smooth and precise motion.

N.1 Slider mechanism applications

- **Mechanical Engineering:** Slider mechanisms are used in machine designs, such as linear motion systems, robotic arms, and assembly lines, to facilitate precise positioning and movement of components.
- **Consumer Products:** They are used in products like drawer slides, sliding doors, telescopic antennas, and camera sliders, enabling smooth and controlled linear motion in everyday items.

- **Automotive:** Slider mechanisms are employed in vehicle seats, sunroofs, and adjustable pedals to provide adjustable positioning and comfort for occupants.
- **Medical Devices:** They are used in medical equipment such as adjustable beds, surgical tables, and patient lifts to facilitate patient positioning and movement during procedures.
- **Electronics:** Slider mechanisms are incorporated into devices like smartphones, laptops, and digital cameras for features such as slide-out keyboards, retractable screens, and adjustable stands.

N.2 Slider mechanism Design Considerations

- **Materials:** Choice of materials based on factors such as strength, durability, friction, and weight.
- **Guiding Mechanism:** Choosing proper guiding features such as tracks, rails, or grooves to ensure smooth and stable motion.
- **Load Capacity:** Designing the mechanism to support the anticipated load while maintaining desired performance.
- **Friction and Lubrication:** Addressing frictional forces and incorporating lubrication or bearing systems to minimize wear and ensure smooth operation.
- **Safety:** Incorporating safety features to prevent pinching, trapping, or accidental release of the sliding components.

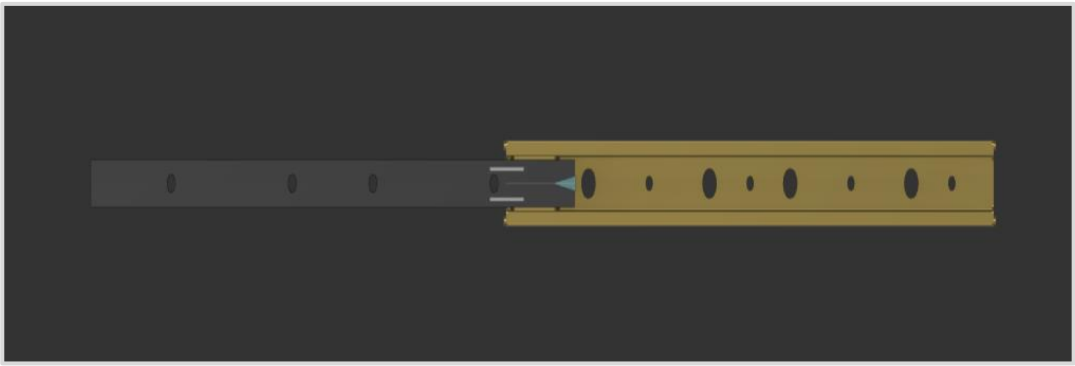


Figure 60 Precision in Motion: Exploring the Versatility of Slider Mechanisms

8.0 Acrylic

also known as polymethyl methacrylate (PMMA) or acrylic glass, is a transparent thermoplastic material with a wide range of applications. Here's an overview of its uses, physical, mechanical, and chemical properties:

0.1 Uses:

- **Signage and Displays:** Acrylic is commonly used for signage, display cases, point-of-sale displays, and retail fixtures due to its clarity, lightweight, and ease of fabrication.
- **Construction and Architecture:** It's used in construction for skylights, windows, doors, and architectural panels, providing transparency, weather resistance, and durability.
- **Automotive:** Acrylic is used in automotive applications such as headlight lenses, taillight covers, and instrument panels due to its optical clarity, impact resistance, and weatherability.
- **Lighting:** Acrylic is used as a diffuser material in LED lighting fixtures, light covers, lenses, and light pipes to distribute and transmit light effectively.
- **Furniture and Decor:** It's used in furniture design, decorative panels, tabletops, and shelving due to its aesthetic appeal, versatility, and ease of fabrication.
- **Medical Devices:** Acrylic is used in medical equipment and devices such as incubators, medical displays, and prosthetic devices due to its biocompatibility and optical clarity.

0.2 Physical Properties:

- **Transparency:** Acrylic offers excellent optical clarity, similar to glass, allowing high levels of light transmission.
- **Lightweight:** It's lighter than glass, making it easier to handle and install in various applications.
- **Weather Resistance:** Acrylic has good weatherability and UV resistance, making it suitable for outdoor applications without yellowing or deteriorating over time.

- **Impact Resistance:** Acrylic is more impact resistant than glass, making it less prone to breakage and shattering.
- **Thermal Properties:** Acrylic has moderate thermal conductivity and can withstand a wide temperature range, but it softens at high temperatures.

0.3 Mechanical Properties:

- **Strength:** Acrylic has good tensile and flexural strength, although not as high as some other engineering plastics.
- **Hardness:** It has a relatively high surface hardness, which provides resistance to scratching and abrasion.
- **Flexibility:** Acrylic can be thermoformed or bent to certain extents without cracking, allowing for some degree of flexibility in design.
- **Dimensional Stability:** Acrylic has low shrinkage and good dimensional stability, maintaining its shape and size over time.

0.4 Chemical Properties:

- **Chemical Resistance:** Acrylic is resistant to many chemicals, although it can be attacked by certain solvents and strong acids.
- **Solubility:** It's soluble in some organic solvents, such as acetone and dichloromethane, which can be used for solvent bonding and polishing.
- **Flammability:** Acrylic is flammable and may burn when exposed to a flame, although flame-retardant grades are available.

In summary, acrylic is a versatile material with a wide range of applications due to its transparency, weather resistance, impact resistance, and ease of fabrication. It's used in industries ranging from signage and construction to automotive and medical devices, offering both aesthetic appeal and functional performance.

8.P Drawer diffuser

Acrylic diffusers play a crucial role in various lighting applications, offering a combination of optical clarity, durability, and design versatility. As a transparent thermoplastic material, acrylic allows for high levels of light transmission while diffusing and distributing light evenly, making it ideal for creating soft and uniform illumination. Its lightweight nature, ease of fabrication, and resistance to weathering and impact further enhance its suitability for lighting fixtures, signage, displays, and architectural elements. Whether used in commercial, residential, or industrial settings, acrylic diffusers contribute to creating aesthetically pleasing and functional lighting solutions. With ongoing advancements in material technology and manufacturing processes, acrylic diffusers continue to evolve, offering improved performance, efficiency, and design possibilities for the lighting industry.

Figure 61 Acrylic Drawer Diffuser

8.Q Computer Numerical Control (CNC)

technology revolutionized manufacturing by automating and enhancing precision in machining processes. Here's a brief overview:

- **Functionality:** CNC technology involves the use of computer-controlled machines to perform various manufacturing operations, such as cutting, milling, drilling, and turning. These machines follow instructions from computer programs to precisely control the movement of cutting tools and workpieces, allowing for the accurate production of complex parts and components.

  Components: A typical CNC system consists of several key components:
- **Computer:** Controls the CNC machine and executes the machining program.

  Controller: Interprets the machining program and generates commands to drive the machine's motors and actuators.
- **Machine Tool:** Performs the actual machining operations, such as cutting, drilling, or milling. Cutting Tools: Tools such as drills, end mills, and lathe tools are mounted on the machine and used to shape the workpiece.
- **Workpiece:** The material being machined, which is typically metal, plastic, wood, or composite material.
- **Operation:** In CNC machining, the process begins with the creation of a part program, usually written in a language like G-code or CAM (Computer-Aided Manufacturing) software. This program specifies the toolpaths, cutting parameters, and other instructions for machining the part. The program is then transferred to the CNC machine's controller, which executes the instructions to move the cutting tool relative to the workpiece, removing material as per the programmed design.
- **Benefits:** CNC technology offers several advantages over conventional machining methods:
- **Precision:** CNC machines can achieve extremely tight tolerances and produce complex geometries with high accuracy.

- **Automation:** CNC machines can run continuously with minimal human intervention, increasing productivity and efficiency.
- **Flexibility:** CNC machines can be reprogrammed quickly to produce different parts, making them suitable for small-batch and custom manufacturing.
- **Quality:** CNC machining produces consistent results, reducing errors and rework compared to manual machining.
- **Versatility:** CNC machines can work with a wide range of materials, including metals, plastics, wood, and composites.
- **Applications:** CNC technology is used in various industries and applications, including:
- **Aerospace:** Manufacturing aircraft components such as engine parts, structural components, and aircraft interiors.
- **Automotive:** Producing automotive parts such as engine blocks, transmission components, and chassis components.
- **Medical:** Manufacturing medical devices, implants, and prosthetics with high precision and quality.
- **Electronics:** Fabricating electronic components, circuit boards, and enclosures for electronic devices.
- **Prototyping:** Rapid prototyping and product development for testing and validation of new designs.

In summary, CNC technology has transformed manufacturing by enabling precise, automated, and versatile machining processes. It has become an indispensable tool in modern manufacturing, driving efficiency, quality, and innovation across industries.

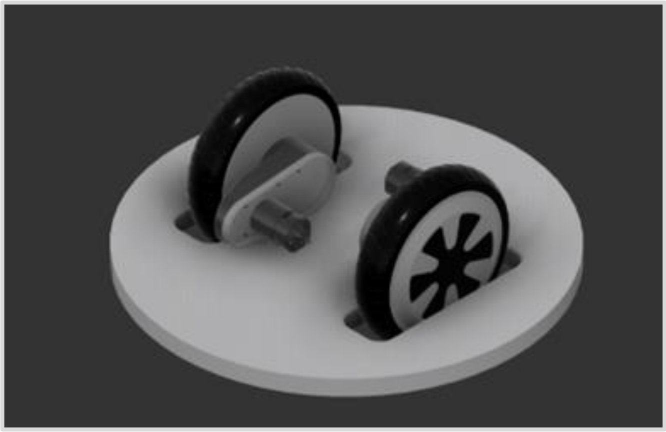


Figure 62 Crafted with precision using CNC technology: the robot's base.

In this exploration, we delved into various aspects of manufacturing and materials, highlighting their significance and impact across industries. We began by examining CNC (Computer Numerical Control) technology, a pivotal innovation that has revolutionized manufacturing processes. By automating machining operations with precision and efficiency, CNC technology enables the production of intricate parts and components with unparalleled accuracy. Its versatility, coupled with benefits such as flexibility, automation, and quality assurance, has propelled it to the forefront of modern manufacturing.

Transitioning to materials, we explored the properties and applications of aluminum and acrylic. Aluminum, prized for its lightweight nature, corrosion resistance, and versatility, finds widespread use in transportation, construction, consumer goods, and industrial applications. Its exceptional physical, mechanical, and chemical properties make it a material of choice for engineers and designers seeking strength, durability, and sustainability.

Acrylic, alternatively, shines as a transparent thermoplastic material renowned for its optical clarity, weather resistance, and ease of fabrication. Widely employed as acrylic diffusers in lighting fixtures, signage, displays, and architectural elements, it offers uniform light distribution while enhancing aesthetics and functionality. With its lightweight design, durability, and ability to withstand various environmental conditions, acrylic continues to shape modern lighting solutions across residential, commercial, and industrial settings.

Chapter 6. Results and Discussion

1. Heart attack model Evaluation

After training each model (Hear attack risk), we evaluated their performance using various metrics such as accuracy, precision, recall, and the F1 score. The model that achieved the highest overall performance was selected.

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
accuracy = accuracy_score(y_test, rf_pred)
precision = precision_score(y_test, rf_pred)
recall = recall_score(y_test, rf_pred)
f1 = f1_score(y_test, rf_pred)
print(f'Accuracy: {accuracy:.2f}')
print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
print(f'F1 Score: {f1:.2f}')
```

1.A Best model selection

Among all the models tested, the Random Forest classifier achieved the highest performance with an accuracy of **0.713** precision of **0.851**, recall of **0.521**, and an F1 score of **0.646.** Thus, it was selected as the final model for predicting heart attack risk.

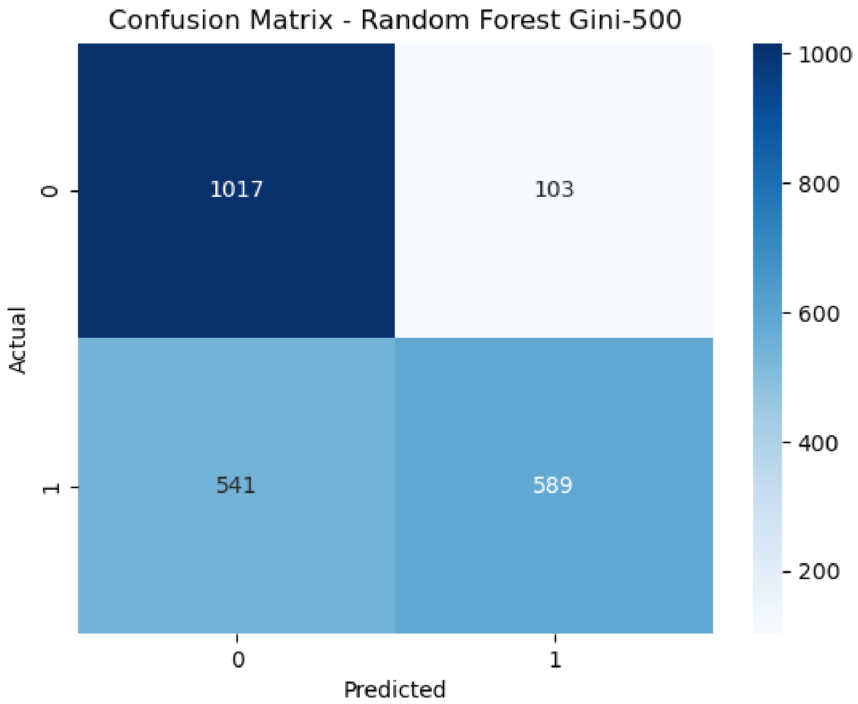


Figure 63 Confusion Matrix for Random Forest (500 Estimators, Gini Criterion) - Best Model

1.B Model saving

```
joblib.dump(rf_gin_model, 'heart_attack_RFGin500_model.pkl')
```

1.C Preventing CVD

A healthy lifestyle can lower your risk of CVD. If you already have CVD, staying as healthy as possible can reduce the chances of it getting worse [3].

C.1 Stop smoking

If you smoke, you should try to give up as soon as possible. The NHS Better Health website can provide information, support, and advice to help.

Your GP can also provide you with advice and support. They can also prescribe medication to help you quit.

C.2 Have a balanced diet

A healthy, balanced diet is recommended for a healthy heart which include the followings:

- low levels of saturated fat try to include healthier sources of fat, such as oily fish, nuts and seeds, and olive oil, and avoid unhealthy fats such as fatty cuts of meat, lard, cream, cakes, and biscuits
- low levels of salt – aim for less than 6g (0.2oz or 1 teaspoon) a day
- low levels of sugar
- plenty of fiber and wholegrain foods
- plenty of fruit and vegetables – eat at least 5 portions of fruit and vegetables a day.

C.3 Exercise regularly

Adults are advised to do at least 150 minutes of moderate activity a week, such as cycling or brisk walking. If you find it difficult to do this, start at a level you feel comfortable with and gradually increase the duration and intensity of your activity as your fitness improves. Visit your GP for a health check if you haven't exercised before or you're returning to exercise after a long break.

C.4 Maintain a healthy weight

If you're overweight or obese, a combination of regular exercise and a healthy diet can help you lose weight. If you're struggling to lose weight, your GP or practice nurse can help you come up with a weight loss plan and recommend services in your area.

C.5 Cut down on alcohol

If you drink alcohol, try not to exceed the recommended limit of 14 alcohol units a week for men and women. If you do drink this much, you should aim to spread your drinking over 3 days or more.

A unit of alcohol is roughly equivalent to half a pint of normal-strength lager or a single measure (25ml) of spirits. A small glass of wine (125ml) is about 1.5 units. Your GP can give you help and advice if you're finding it difficult to cut down on your drinking.

C.6 Medicine

If you have a particularly high risk of developing CVD due to high blood cholesterol, your GP may recommend taking medicines called statins to reduce your risk.

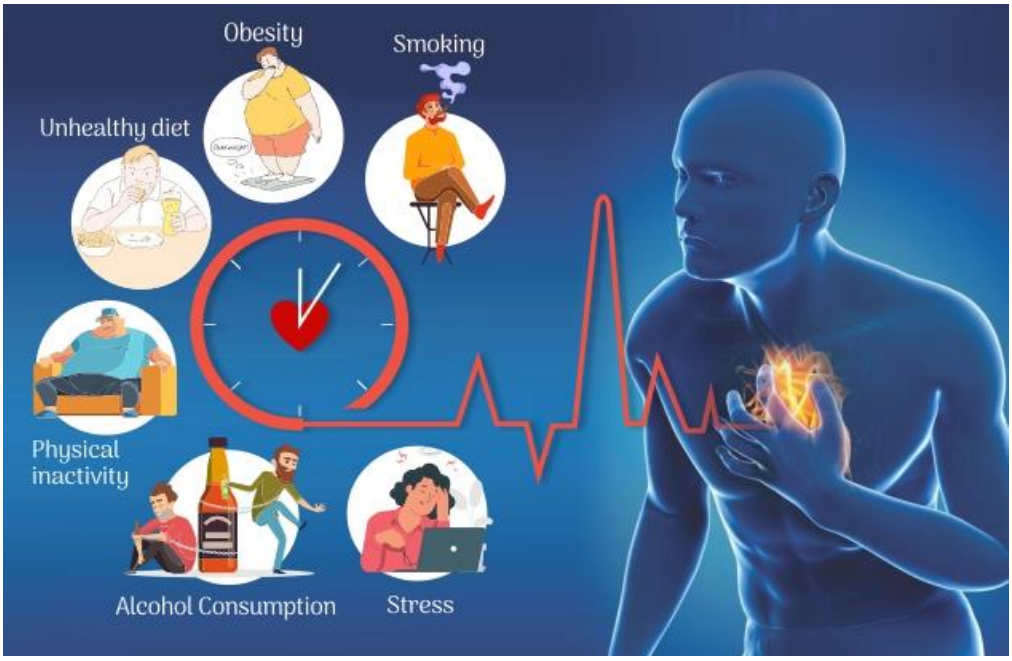


Figure 64 Cardiovascular Problems and How to Prevent Them

2. Chest X-ray model Evaluation

2.A VGG16 algorithm

VGG16, also known as VGGNet, is a convolutional neural network (CNN) architecture designed for image recognition tasks. It was developed by the Visual Geometry Group (VGG) at the University of Oxford and introduced in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" in 2014. Here's a breakdown of its key features:

A.1 Depth and Simplicity:

- VGG16 is known for its depth, consisting of 16 layers – 13 convolutional layers and 3 fully-connected layers. This depth allows the network to learn complex features from images.
- Interestingly, VGG16 achieves this depth with a relatively simple architecture. It primarily uses 3x3 filters in all convolutional layers, stacked together with max-pooling layers for downsampling. This repetitive structure makes it easy to understand and implement.

A.2 Performance and Transfer Learning:

- VGG16 achieved impressive results on the ImageNet image classification task in 2014, demonstrating the effectiveness of deep convolutional networks.
- Due to its strong performance, VGG16 has become a popular choice for transfer learning. In transfer learning, a pre-trained VGG16 model (trained on a massive dataset like ImageNet) is used as a feature extractor, with a new classification head added on top for a specific task. This approach leverages the learned features from the pre-trained model and reduces training time for new tasks.

A.3 Advantages and Disadvantages:

A.4 Advantages:

- Achieves high accuracy on image classification tasks.
- Simple and well-understood architecture.
- Pre-trained models available for transfer learning.

A.5 Disadvantages:

- Can be computationally expensive to train due to its depth.
- Requires a large amount of data for training from scratch.
- May not be the most efficient architecture for resource-constrained environments compared to newer models.

Overall, VGG16 is a significant milestone in the development of deep convolutional neural networks. While it may not be the most cutting-edge architecture today, its depth, simplicity, and effectiveness in transfer learning make it a valuable tool in the computer vision field.

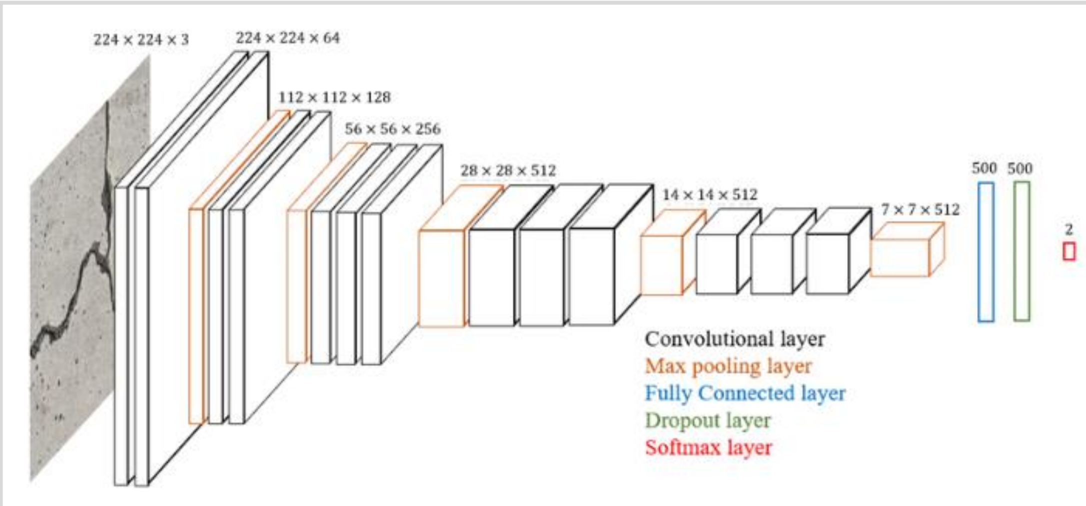


Figure 65 Architecture of the modified VGG16 model

2.B Inception

The Inception architecture is a convolutional neural network (CNN) design known for its efficient approach to feature extraction in images. It was developed by researchers at Google and introduced in the paper "Going Deeper with Convolutions" by Szegedy et al. in 2014.

Here's a breakdown of the key concepts:

B.1 Inception Modules:

- The core building block of the Inception architecture is the inception module. This module combines several convolutional filters with different kernel sizes (1x1, 3x3, 5x5) within a single layer.
- These filters capture features at various scales and resolutions within the image. The outputs from these filters are then concatenated, allowing the network to learn a richer and more comprehensive representation of the image.

B.2 Benefits of Inception Modules:

a. **Efficient Feature Extraction:** By combining filters of different sizes in a single module, Inception allows the network to efficiently learn a wider range of spatial and frequency information from the input image.

b. **Reduced Parameters:** Compared to traditional CNNs that use separate layers for different filter sizes, Inception can achieve similar performance with fewer overall parameters. This makes it more efficient in terms of memory usage and potentially faster to train.

c. **Multi-Scale Processing:** The use of various filter sizes enables the network to capture features at different scales, which is crucial for tasks like object recognition where objects can appear in different sizes within an image.

B.3 Inception Network Architectures:

a. The Inception architecture has been used in various configurations, with the most well-known being the GoogLeNet model, which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014.

b. Newer versions, like Inception v2, Inception v3, and Inception-ResNet v2, have introduced further improvements and optimizations to the basic Inception module concept.

2.C Applications of Inception:

Inception architectures have been successfully applied to various computer vision tasks, including:

- Image classification (ImageNet)
- Object detection
- Image segmentation
- Video classification

2.D Comparison to Traditional CNNs:

- Compared to traditional CNNs that stack convolutional layers with a single filter size, Inception offers a more efficient way to extract features at different scales within a single layer.
- While Inception can be computationally expensive during training due to the use of multiple filters, its potential benefits in parameter efficiency and feature extraction make it a valuable architecture for image recognition tasks.

Overall, the Inception architecture has been a significant contribution to the development of deep convolutional neural networks. Its focus on efficient feature extraction with inception modules has led to advancements in various computer vision applications.

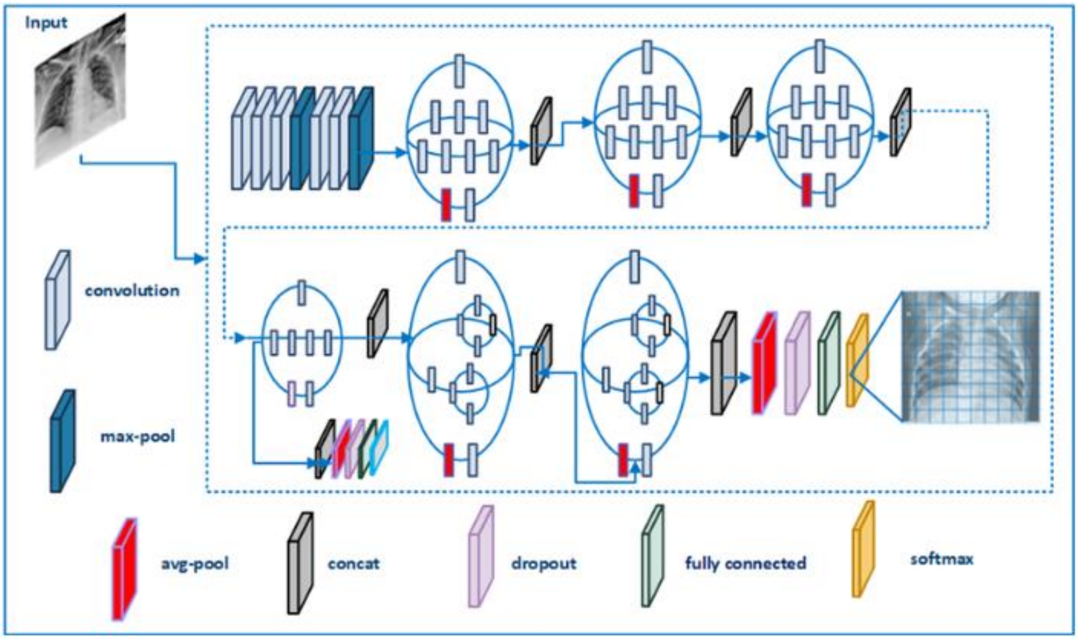


Figure 66 12 Inception-V3 architecture, trained for Chest X-ray images. The output of the architecture is further utilized for transfer learning, multi-layer feature fusion and selection

2.E DensNet algorithm

DenseNet, also known as Dense Convolutional Network, is a relatively recent convolutional neural network (CNN) architecture designed for image recognition and classification tasks. It was introduced in the paper "Densely Connected Convolutional Networks" by Huang et al. in 2016. Here's what makes DenseNet unique:

E.1 Dense Connections:

- Unlike traditional CNNs where each layer receives input only from the previous layer, DenseNet connects each layer to all preceding layers in a feed-forward fashion. This approach fosters feature reuse and information flow throughout the network.
- Each layer receives the combined feature maps from all preceding layers, concatenating them as input. This allows later layers to access and potentially improve upon the features learned by earlier layers.

E.2 Benefits of Dense Connections:

a. **Improved Feature Propagation:** Information learned by earlier layers is directly accessible to later layers, potentially leading to better feature extraction.
b. **Reduced Vanishing Gradients:** Dense connections can help alleviate the vanishing gradient problem, a common issue in deep networks where gradients become too small during backpropagation, hindering training.
c. **Potentially Fewer Parameters:** DenseNet can achieve similar performance with fewer parameters compared to traditional CNNs due to the efficient feature reuse enabled by dense connections.

E.3 Dense Blocks and Transition Layers:

- DenseNet is typically built using dense blocks, which are groups of convolutional layers with dense connections.
- Transition layers are inserted between dense blocks. These layers perform operations like bottlenecking (reducing the number of feature maps) and downsampling to control network size and introduce some level of hierarchy.

2.F DenseNet Variants and Applications:

- DenseNet comes in various configurations with different numbers of layers and growth rates (determining how many feature maps are added in each layer). Popular variants include DenseNet-121, DenseNet-169, and DenseNet-201.
- DenseNet has shown promising results in various image recognition tasks, including object classification, image segmentation, and medical image analysis.

2.G Comparison to ResNet (another popular CNN):

- Both DenseNet and ResNet (another widely used CNN architecture) aim to address the vanishing gradient problem in deep networks.
- DenseNet achieves this through direct connections, while ResNet uses skip connections that jump over a few layers.
- DenseNet can be more efficient in terms of parameter usage, but its dense connections also make it computationally expensive during training.

Overall, DenseNet offers a compelling architecture for image recognition tasks with its focus on dense connections and efficient feature reuse. While it may require more computational resources during training compared to some alternatives, its potential benefits in accuracy and parameter efficiency make it a valuable tool in the computer vision toolbox.

2.H Chest X-ray model Results

Predicting for Model DenseNet121

- **AUROC:** 0.7838944810043479
- **Accuracy:** 0.8661118865509148
- **f1 score:** 0.2781139232578845
- **precision score:** 0.3769061182384384
- **recall score:** 0.2405177476737821

Predicting for Model Vgg16

- **AUROC:** 0.7160723914295691
- **Accuracy:** 0.8380253234105122
- **f1 score:** 0.12822206598069283
- **precision score:** 0.18616550288586275
- **recall score:** 0.10502495781061488

Predicting for Model Inception

- **AUROC :** 0.6921098095693519
- **Accuracy :** 0.8297820273226124
- **f1 score :** 0.1167743126554978
- **precision score :** 0.16528302942802703
- **recall score :** 0.09623468806091708

2.I comparing performances with other Research papers

DensNet121 has achieved best AUC between the three algorithms.

Condition	Wang	Yao	CheXNeXt	DCXNet	Our model
Atelectasis	0.716	0.772	0.862	0.804	0.896
Cardiomegaly	0.807	0.904	0.831	0.897	0.796
Effusion	0.784	0.859	0.901	0.874	0.652
Infiltration	0.609	0.695	0.721	0.706	0.802
Mass	0.706	0.792	0.909	0.792	0.712
Nodule	0.671	0.717	0.894	0.735	0.675
Pneumonia	0.633	0.713	0.851	0.735	0.857
Pneumothorax	0.806	0.841	0.944	0. 876	0.751
Consolidation	0.708	0.788	0.893	0.823	0.763
Edema	0.835	0.882	0.924	0.89	0. 775
Emphysema	0.815	0.829	0.704	0.907	0.82
Fibrosis	0.769	0.767	0.806	0.81	0.656
Plural_Thickening	0.708	0.765	0.798	0.782	0.728

Table 5 Performance comparison of presented model with state-of-the-art models on basis of AUC

1.1 AUROC metrics for more training Epoch

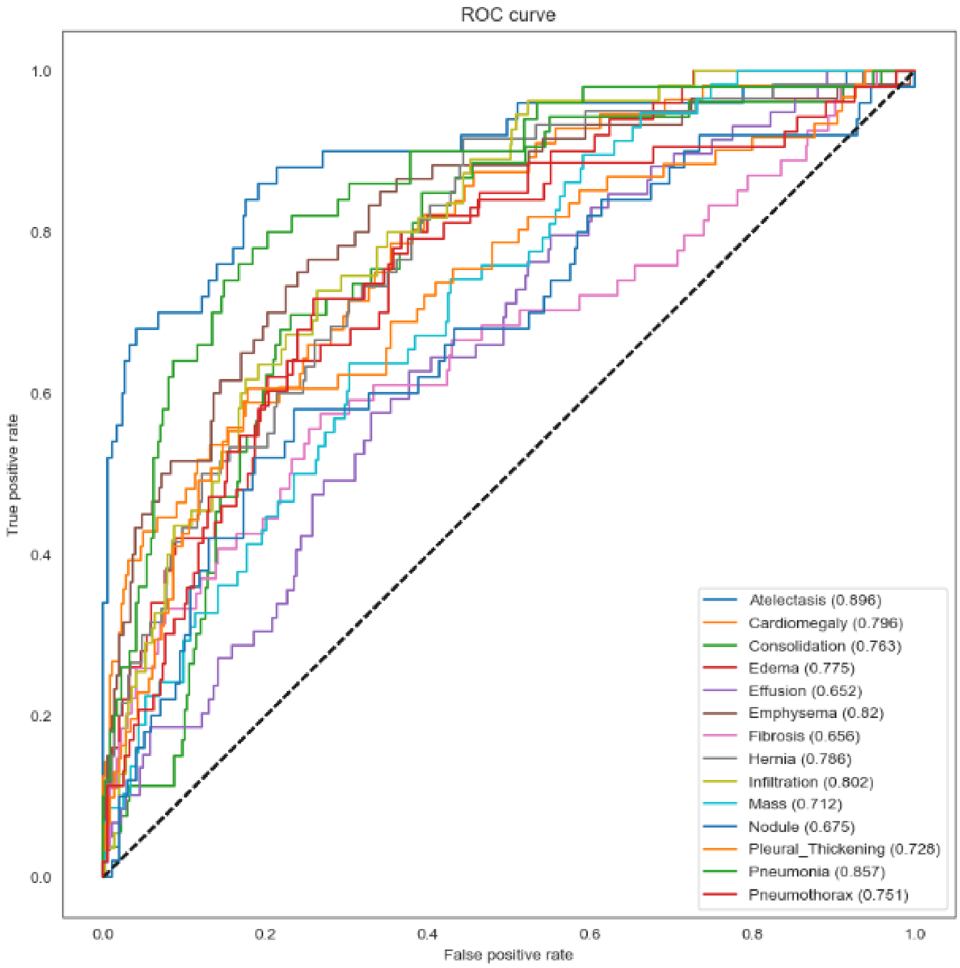


Figure 67 AUC Curve for best performance

- To compare our model to previous algorithms, a single diagnostic performance measure, AUC was used. AUC is a metric that indicates how well a model fits. In the medical field, this value also represents the likelihood that a patient who suffered a condition would have a greater risk score than a patient who didn't experience the event, if chosen at random. It summarizes the model's performance across various thresholds and provides a reliable indication of its ability to distinguish between different cases. AUC was chosen because it does not require a threshold value and we do not need to convert our model outputs to binary predictions as the model outputs a vector between 0−1 for every class which indicates the probability of every class for a provided image. This AUC was calculated using sklearn library which plots a ROC curve and then calculates the area under the curve using Simpson's rule.

- The performance of the presented model was compared to Wang[10] andYao[11], CheXNeXt[12] and DCXNet[13] in Table 1.

Our model has higher performance on Pneumonia class(AUC ≥ 0.1) and Atelectasis class(AUC ≥ 0.092)

2.j Test Metrics

Model performance validated against success metrics. Details available are in Figure 9. Based on overall test metrics, DenseNet121 achieved the best performance. This model achieved ROC score of **0.78** and accuracy of **87%**. Other models ROC values ranged from **0.69** to **0.75** and accuracy from **83%** to **86%**.

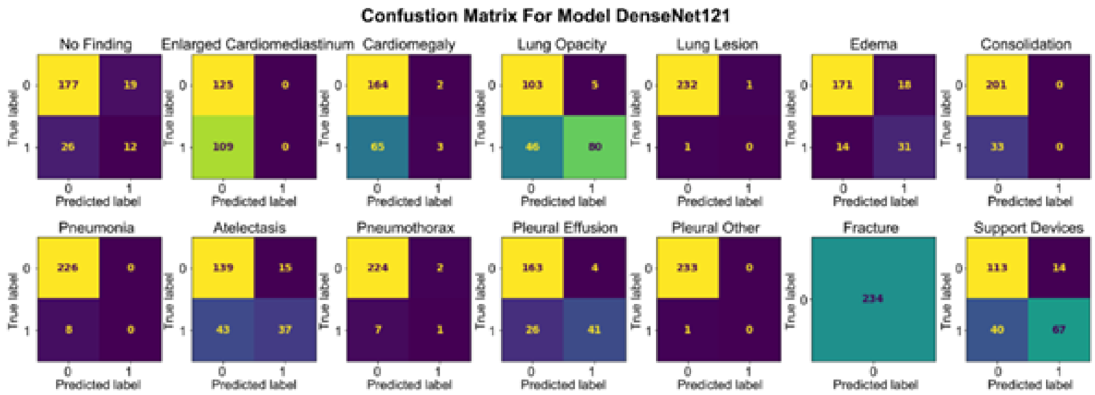
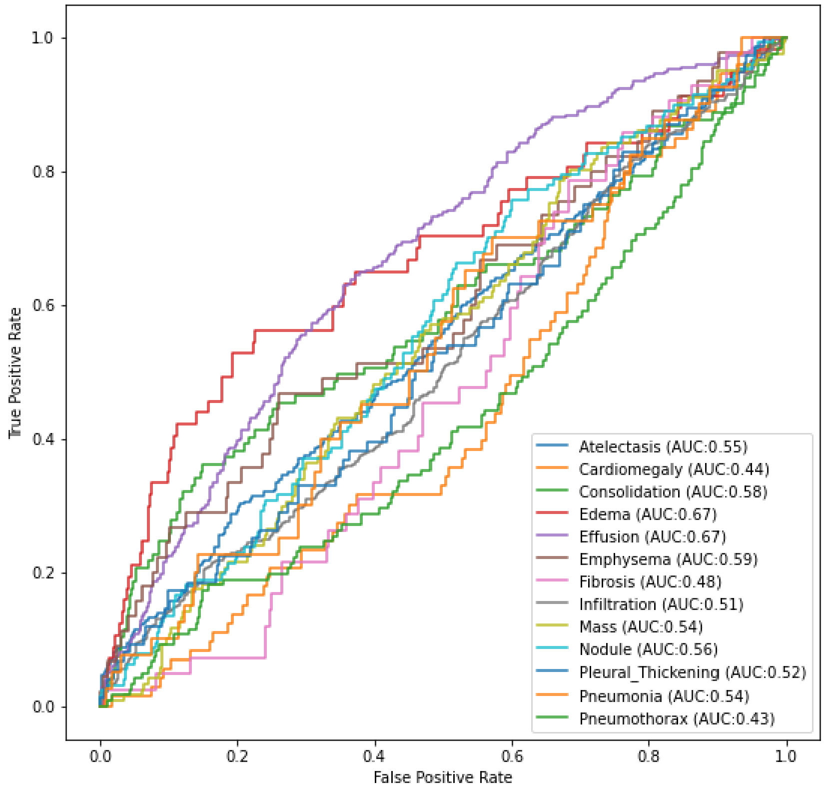


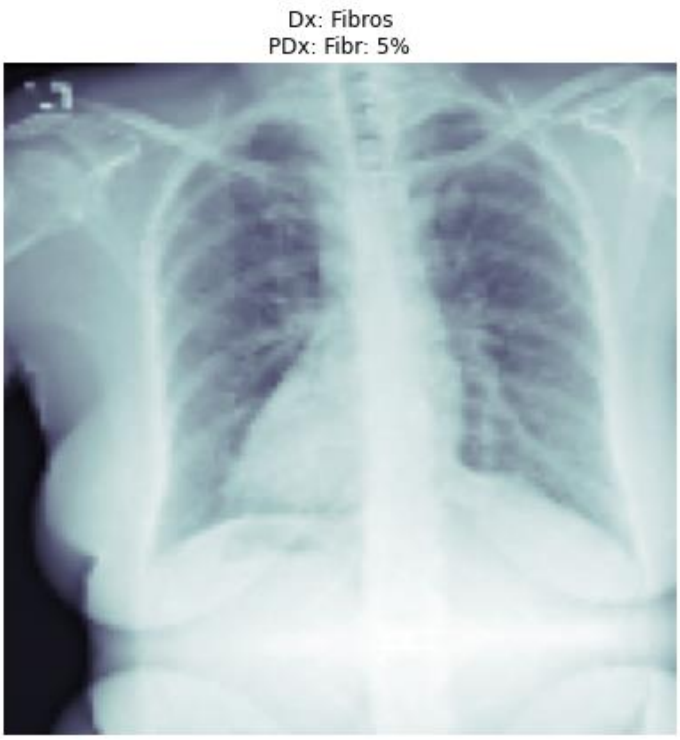
Figure 68 Confusion matrix for DenseNet

first training Evaluation AUC metric :

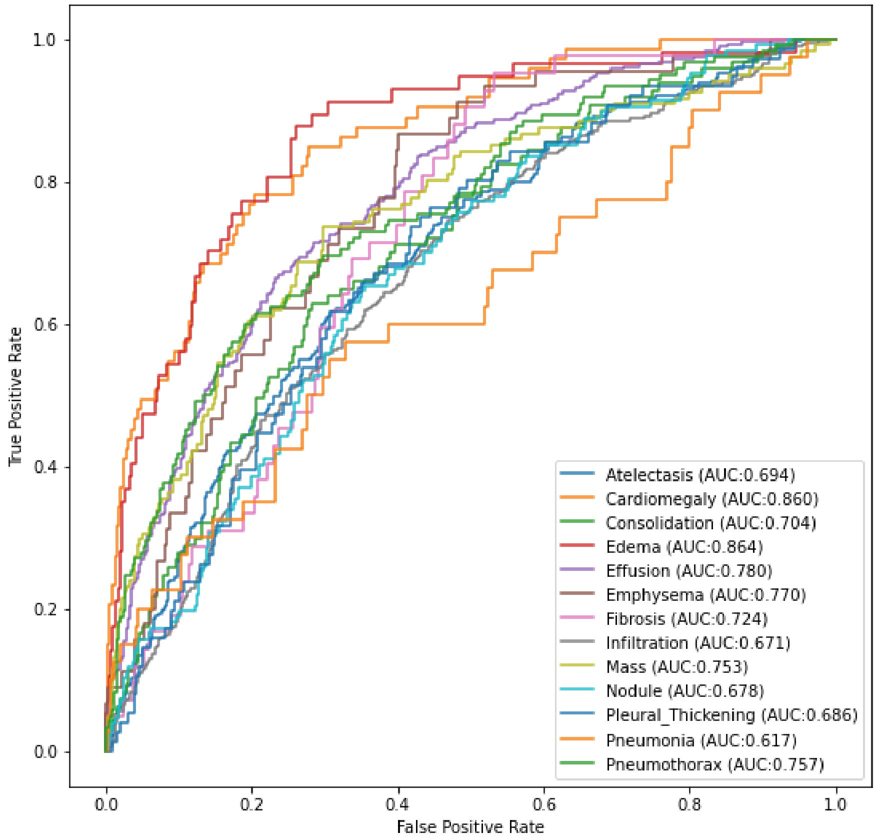


Prediction :

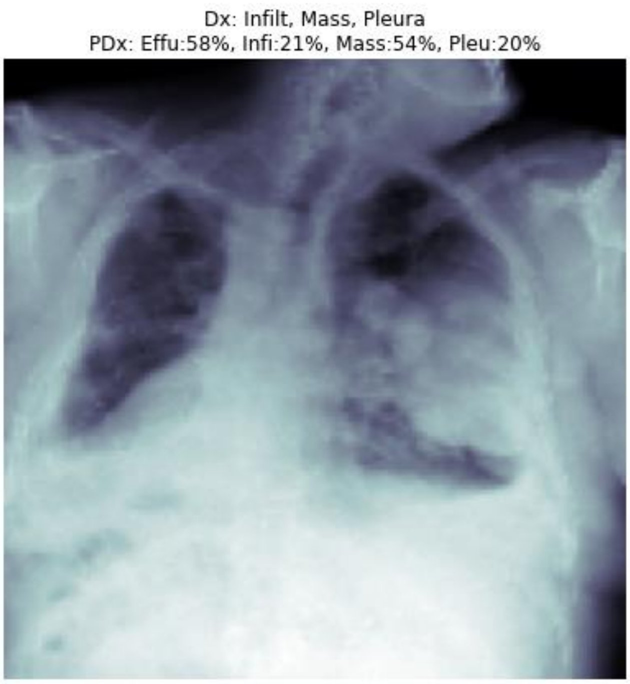
Not accurate as respected



Second training Evaluation AUC metric :



Prediction 2



3. Challenges of Deep Learning in healthcare applications

3.A Visualizing Learning with GradCAM

One of the challenges of using deep learning in medicine is that the complex architecture used for neural networks makes them much harder to interpret compared to traditional machine learning models (e.g. linear models).

One of the most common approaches aimed at increasing the interpretability of models for computer vision tasks is to use Class Activation Maps (CAM).

- Class activation maps are useful for understanding where the model is "looking" when classifying an image.

In this section we will use a **GradCAM's** technique to produce a heatmap highlighting the important regions in the image for predicting the pathological condition.

- This is done by extracting the gradients of each predicted class, flowing into our model's final convolutional layerIt is worth mentioning that GradCAM does not provide a full explanation of the reasoning for each classification probability.
- However, it is still a useful tool for "debugging" our model and augmenting our prediction so that an expert could validate that a prediction is indeed due to the model focusing on the right regions of the image.

  Interpretation tools like this one can be helpful for discovery of markers, error analysis, and even in deployment.

3.B GradCam flowchart

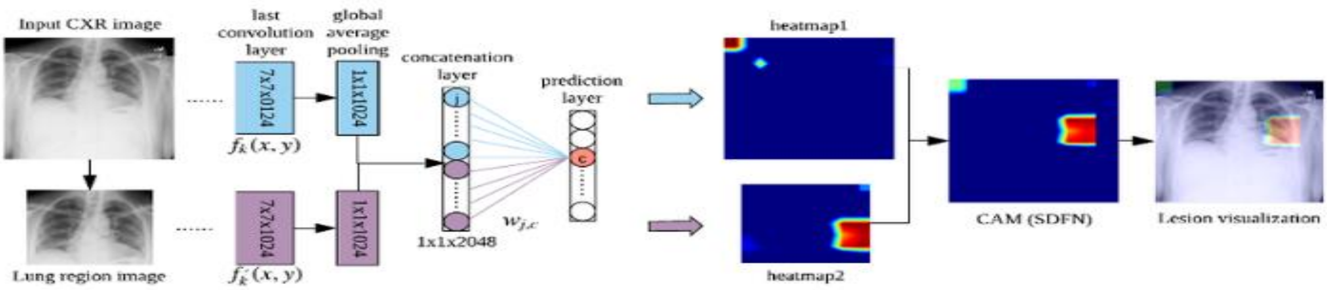


Figure4.1 The flowchart illustrating the generation of the class activation map (CAM) for the proposed SDFN method.

3.C Heat maps

In this chest x-ray, the patient has an enlarged heart also called cardiomegaly.

Our algorithm recognizes that it's cardiomegaly with an output of 80% probability.

We will look at how we can also have the algorithm output a heat map over the image showing the parts of the image most indicative of cardiomegaly.

Here we can see that the model is looking in the heart region to make a decision.

So its heart's in the right place.

We'll look at the Grad-CAM method to make this output.

When an image is passed through a convolutional neural network, it is passed through a series of layers.

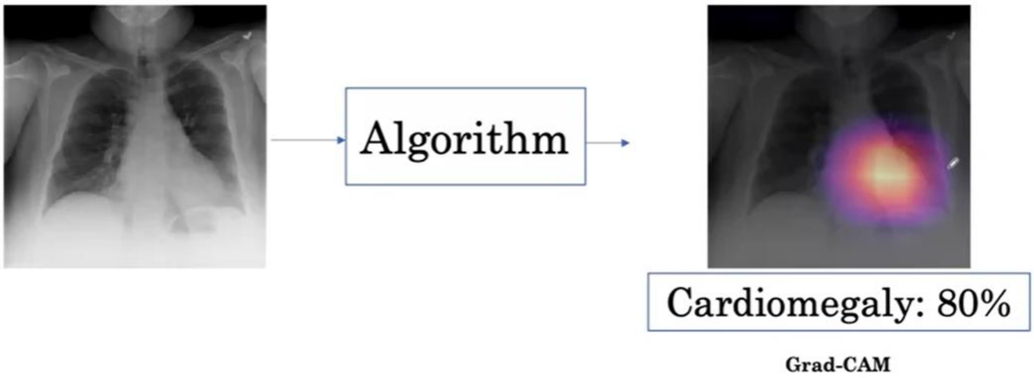


Figure 4.1 the patient has cardiomegaly

One final change we make in this computation is that we're only interested in features that have a positive influence on the score for cardiomegaly.

So here just expanding out the localization maps so we can see all of the numbers. We'll set the negative numbers here, in the localization map to zero and the positive numbers we keep as they are. This is applying the ReLu function, which all it's doing is deciding which is greater, the element or zero, and setting it to the max of that.

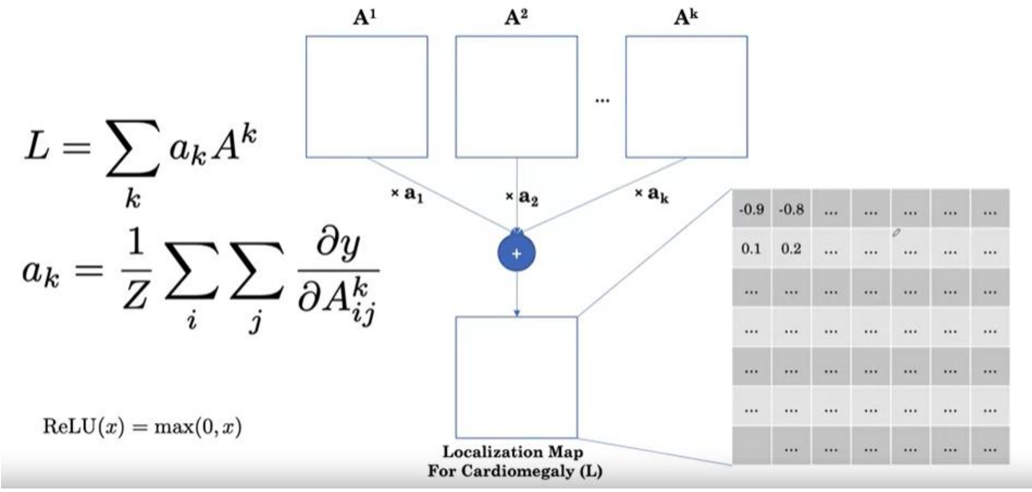
So we make that modification to the computation of the localization map L. Now, rather than look at the localization map as numbers on a table, we can visualize it using a heat map. A heat map is generated using a color map that translates numbers into color.

In this case, the more positive numbers are going to appear bright and the numbers close to zero appear black.

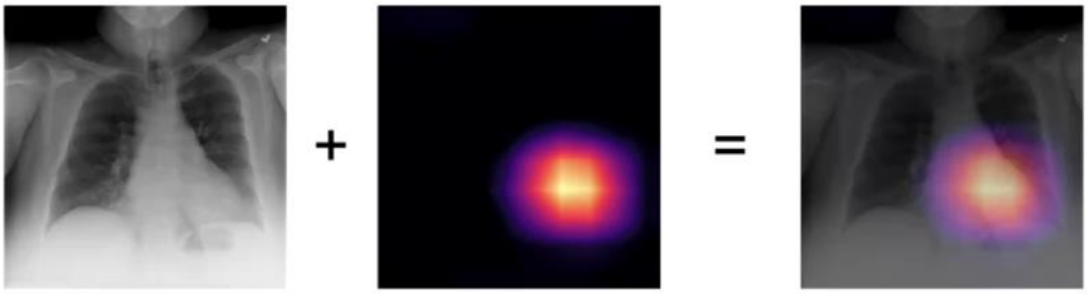
A. Breach notifications

B. Electronic Health Record (EHR) incentives

Enforcement: Office for Civil Rights (OCR), Health and Human Services (HHS) [37]



The heat map is generated at a low resolution because it's the same size as the spatial map which we saw earlier was 10 by 10. We usually have to resize it and usually have some kind of interpolation to fill in the gaps when we resize to bring it up to the size of the x-ray image.



Finally, we can now overlay the resized heat map on the original image.

Sometimes transparency is added to the heat map, sometimes proportional to the score for the class such that the heat map is brighter when the score is higher and less otherwise.

When we combine the two, the heat map and the x-ray image, we got this overlay which shows us where the model is looking to make a particular decision for cardiomegaly in this case.

Thus we can compute the GradCAM for cardiomegaly using this procedure on even new images.

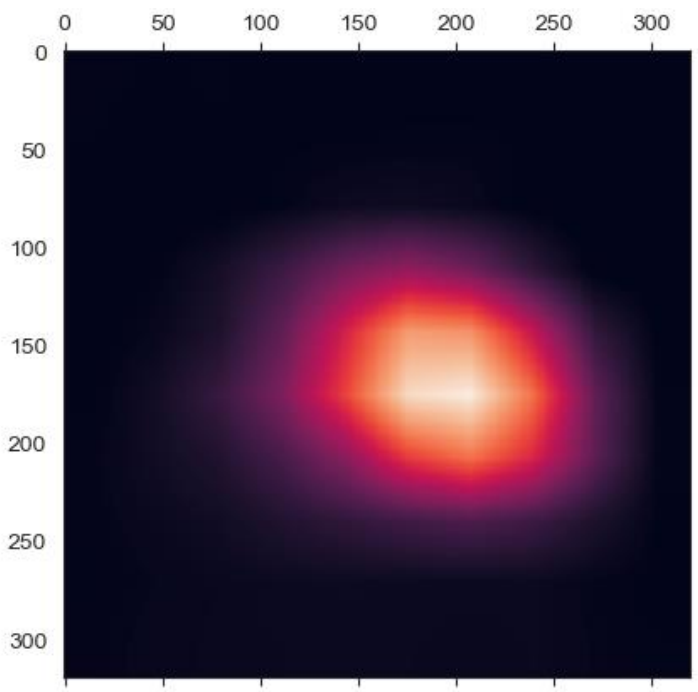
As a final extension, if the model has multiple possible output classes as your model for chest x-ray classification does, there is a different heat map you would want per disease.

3.D interpreting prediction Diseases (on Raspberry pi)

The dimensions of the image are 320 pixels width and 320 pixels height
The maximum pixel value is 0.9137 and the minimum is 0.0000
The mean value of the pixels is 0.4094 and the standard deviation is 0.2722

1/1 [==============================] - 2s 2s/step

Predictions: [Atelectasis]



4. Challenges Faced

4.A Radiologist wrong Diagnosis

Reading and diagnosing chest x-ray images may be a relatively simple task for radiologists but, in fact, it is a complex reasoning problem which often requires careful observation and knowledge of anatomical principles, physiology and pathology. Such factors increase the difficulty of developing a consistent and automated technique for reading chest X-ray images while simultaneously considering all common thoracic diseases.

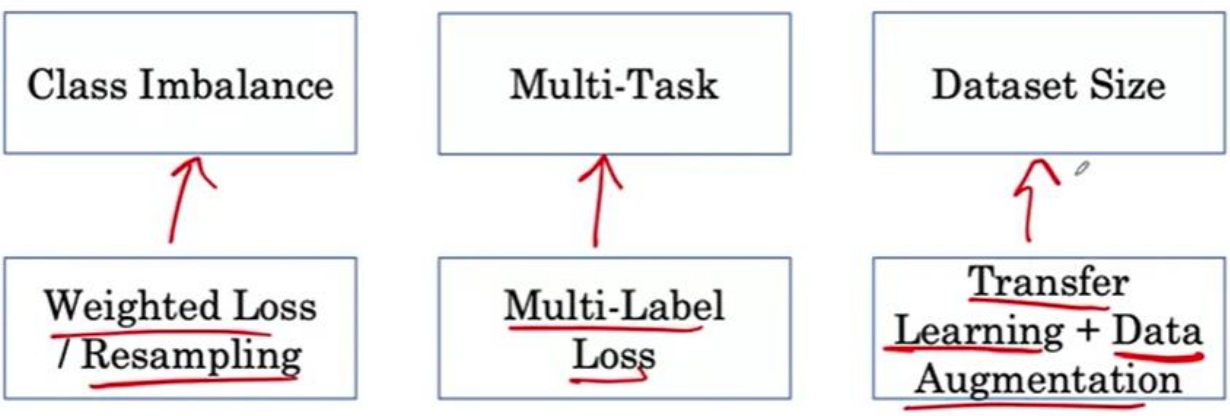
4.B Solution

By using this free dataset, the hope is that academic and research institutions across the country will be able to teach a computer to read and process extremely large amounts of scans, to confirm the results radiologists have found and potentially identify other findings that may have been overlooked.

In addition, this advanced computer technology may also be able to:

- help identify slow changes occurring over the course of multiple chest x-rays that might otherwise be overlooked
- benefit patients in developing countries that do not have access to radiologists to read their chest x-rays, and
- create a virtual radiology resident that can later be taught to read more complex images like CT and MRI in the future.

4.C Problems with the Dataset



5. Imbalance in the dataset

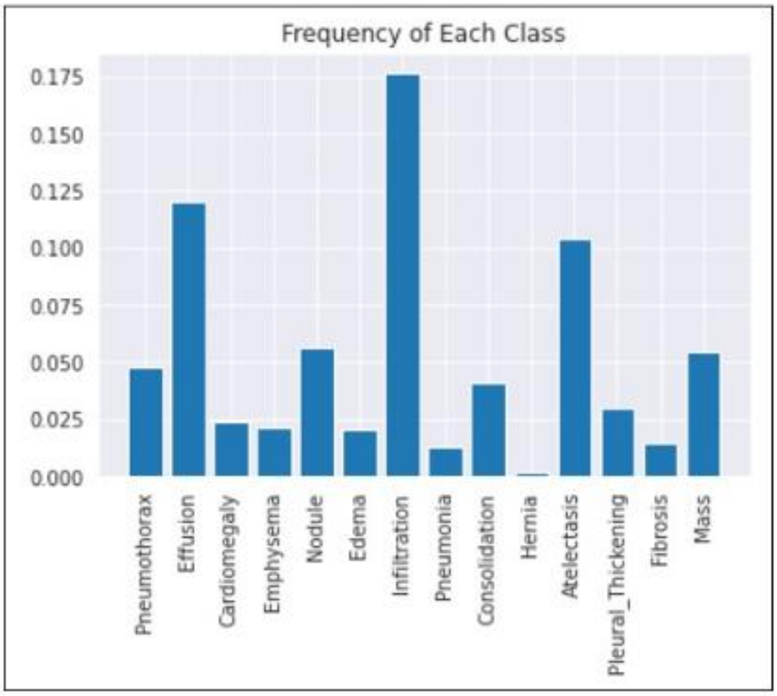


Figure 2: Imbalance in the dataset

5.A Solution

• The most unbalanced pathology is Hernia, with 0.1% of patients testing positive for training..

• However, only 17.5% of the training instances for the Infiltration pathology, which has the least degree of imbalance, have been classified as positive

This class imbalace issue doesnot allow for a normal cross-entropy loss for each class. For a balanced data set the loss function is:

$$\mathcal{L}(x_i) = -(y_i log(f(x_i)) + (1 - y_i) log(1 - f(x_i)))$$

where xi and yi are the input features and their corresponding labels and f(xi) is the output of the model which indicates the probability that it is positive. With the use of this formulation, we can observe that the loss will be dominated by the negative class in situations when there is a significant imbalance and there are few positive training events. One way of balancing such datasets require multiplying each class by a class-specific weight factors, wp and wn where wp is the frequency of negative samples and wn is the frequency of positive samples for each class. Then the previous unweighted loss function was modified as:

$$\mathcal{L}^{w}(x) = -(w_p y log(f(x)) + w_n(1-y) log(1-f(x)))$$

This equation was used to calculate the loss for each class and then the total loss can be calculated as the mean of the loss of each of the classes.

6. Future Work

6.A Integration with CT Scans

To enhance the robustness and comprehensiveness of our chest disease detection system, future work will focus on integrating Computed Tomography (CT) scans. CT scans provide detailed cross-sectional images of the body and offer more detailed information compared to traditional chest X-rays. By incorporating CT scan analysis, we aim to:

- **Improve Diagnostic Accuracy:** CT scans can help in detecting diseases that may not be visible on X-rays, such as early-stage cancers or small lesions.
- **Broaden Disease Detection:** Expand the range of detectable conditions by analyzing different tissue densities and structures that CT scans can reveal.
- **Enhance Model Training:** Utilize the richer data from CT scans to train more sophisticated models that can better understand complex patterns and anomalies.

6.B MRI Integration

Magnetic Resonance Imaging (MRI) offers another layer of diagnostic capability by providing high-contrast images of soft tissues. Incorporating MRI into our detection system will allow us to:

- **Detect Soft Tissue Abnormalities:** Identify diseases affecting soft tissues, such as tumors or inflammatory conditions, with higher precision.
- **Functional Imaging:** Utilize functional MRI techniques to study the physiological and metabolic changes associated with different chest diseases.
- **Comprehensive Patient Profiling:** Combine MRI data with X-ray and CT scan data to create a more comprehensive patient profile, improving overall diagnostic accuracy.

6.C Enhancing Accuracy

Improving the accuracy of our disease detection system is a continuous goal. To achieve higher accuracy, we plan to:

- **Advanced Algorithms:** Develop and implement more advanced machine learning and deep learning algorithms, such as convolutional neural networks (CNNs) and transformer models, specifically tailored for medical imaging.
- **Larger Datasets:** Curate and utilize larger, more diverse datasets to train our models, ensuring they generalize well across different populations and imaging conditions.

- **Multimodal Learning:** Explore multimodal learning techniques that combine information from X-rays, CT scans, and MRIs to enhance the system's diagnostic capabilities.
- **Real-Time Feedback:** Implement real-time feedback mechanisms that continuously learn from new data and improve model performance over time.

By focusing on these areas, we aim to develop a more accurate, reliable, and comprehensive system for detecting chest diseases, ultimately improving patient outcomes and healthcare efficiency.

Chapter 7. References

[1] Glover, Ellen, "Artificial Intelligence," built in, 2024. [Online]. Available: https://builtin.com/artificial-intelligence. [Accessed 2024].

[2] IBM, "Artificial Intelligence types," IBM, [Online]. Available: https://www.ibm.com/think/topics/artificial-intelligence-types. [Accessed 2024].

[3] "Cardiovascular Disease," NHS UK, [Online]. Available: https://www.nhs.uk/conditions/cardiovascular-disease/. [Accessed 2024].

[4] H. Li, Machine Learning Methods, Beijing, China: Springer, 2024.

[5] PathAI, "Path AI," [Online]. Available: https://www.pathai.com/. [Accessed 2024].

[6] "Aidoc," Aidoc, [Online]. Available: https://www.aidoc.com/. [Accessed 2024].

[7] "Healthcare," Bautomate, [Online]. Available: https://bautomate.com/healthcare/. [Accessed 2024].

[8] "deep genomics," [Online]. Available: https://www.deepgenomics.com/. [Accessed 2024].

[9] "Prognos Health," [Online]. Available: https://prognoshealth.com/. [Accessed 2024].

[10] "Benchsci," Benchsci, [Online]. Available: https://www.benchsci.com/. [Accessed 2024].

[11] Viz AI, "Viz AI," [Online]. Available: https://www.viz.ai/. [Accessed 2024].

[12] Innovaccer, "Innovaccer: Health Cloud Powerd by Unified patient data," Innovaccer, [Online]. Available: https://innovaccer.com/. [Accessed 2024].

[13] "Caption Care," Caption Care, [Online]. Available: https://www.caption-care.com/. [Accessed 2024].

[14] C. Health, "Current Health," [Online]. Available: https://www.currenthealth.com/.

[15] Harvard, "Machine Learning in Medicine 101: Health Care Executives," Harvard Medical School, [Online]. Available: https://corporatelearning.hms.harvard.edu/blog/machine-learning-medicine-101-health-care-executives. [Accessed 2024].

[16] A. Kornilov, "Data types in machine learning," Training Data, [Online]. Available: https://trainingdata.pro/data-types-in-machine-learning. [Accessed 2024].

[17] ICert Global, "Data Acquisition in Machine Learning," ICert Global, [Online]. Available: https://www.icertglobal.com/data-acquisition-in-machine-learning/detail. [Accessed 2024].

[18] S. Banerjee, "Heart Attack Prediction Dataset," 2021. [Online]. Available: https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset.

[19] "Data Preprocessing Guide," Neptune AI, 2023. [Online]. Available: https://neptune.ai/blog/data-preprocessing-guide. [Accessed 2024].

[20] "Encoding Techniques for Machine Learning in Alteryx: A Concise Guide," AlteryX, 2023. [Online]. Available: https://community.alteryx.com/t5/Data-Science/Encoding-Techniques-for-Machine-Learning-in-Alteryx-A-Concise/ba-p/1203894. [Accessed 2024].

[21] J. S. John Doe, "Implementation of Medical Machine Learning Applications," *Journal of Medical Research,* 2020.

[22] "Raspberri Pi," Adafruit, [Online]. Available: https://www.adafruit.com/product/4564.

[23] "Poe Plus Hat," raspberrypi, [Online]. Available: https://www.raspberrypi.com/products/poe-plus-hat/.

[24] PWC, "Transforming Healthcare: From AI and Robotics to New Health," PwC Global, [Online]. Available: https://www.pwc.com/gx/en/industries/healthcare/publications/ai-robotics-new-health/transforming-healthcare.html. [Accessed 2024].

[25] T. Natalie, "Machine Learning in Healthcare: 12 Real-World Use Cases to Know," Nix United, 2023. [Online]. Available: https://nix-united.com/blog/machine-learning-in-healthcare-12-real-world-use-cases-to-know/. [Accessed 2024].

[26] S. Sarkar, "Machine Learning in Healthcare Communication," 2021. [Online]. Available: https://www.mdpi.com/2673-8392/1/1/21. [Accessed 2024].

[27] Google, "Machine Learning Glossary," google developers, [Online]. Available: https://developers.google.com/machine-learning/glossary. [Accessed 2024].

[28] A. G. B. Richard S. Sutton, "Reinforcement Learning: An Introduction," [Online]. Available:

https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf. [Accessed 2024].

[29] IBM, "Machine Learning Types," [Online]. Available: https://www.ibm.com/blog/machine-learning-types/. [Accessed 2024].

[30] IBM, "How Machine Learning Algorithms Work," IBM, [Online]. Available: https://www.ibm.com/topics/machine-learning. [Accessed 2024].

[31] Biomedical team, "Biomedical project 2024," May 2024. [Online]. Available: https://github.com/Yasso-mamdouh/Biomedical2024-ECCAT.git.

[32] M. Davis, "Challenges and Barriers to Implementing AI in Healthcare," 2023. [Online]. Available: https://blogs.vmware.com/industry-solutions/2023/07/19/challenges-and-barriers-to-implementing-ai-in-healthcare/.

[33] L. Serhii and C. Roman, "Machine Learning in Healthcare," SPD Technology, [Online]. Available: https://spd.tech/machine-learning/machine-learning-in-healthcare/. [Accessed 2024].

[34] "Health Information Privacy," U.S. Department of Health and Human Services, [Online]. Available: https://www.hhs.gov/hipaa/index.html. [Accessed 2024].

[35] "General Data Protection Regulation," European Commission, [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en. [Accessed 2024].

[36] "A Guide for Businesses on PIPEDA," Office of the Privacy Commissioner of Canada, [Online]. Available: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-

electronic-documents-act-pipeda/pipeda-compliance-help/pipeda-compliance-and-training-tools/gd_pb_200807/. [Accessed 2024].

[37] "Health Information Technology for Economic and Clinical Health," U.S. Department of Health and Human Services, [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/special-topics/hitech-act-enforcement-interim-final-rule/index.html. [Accessed 2024].

[38] J. S. John Doe, "Impact of AI on Health Informatics," *Journal of Health Informatics in Developing Countries (JHIDC),* 2023.

[39] "Advances in AI Research," *Computers.*

[40] IBM, "Artificial intelligence," [Online]. Available: https://www.ibm.com/topics/artificial-intelligence.

[41] Google, "ML Solutions Overview," [Online]. Available: https://cloud.google.com/ai-platform/docs/ml-solutions-overview.

[42] J. Saltz, "AI lifecycle," 2024. [Online]. Available: https://www.datascience-pm.com/ai-lifecycle/.

[43] MIT, "Machine Learning Explained," 2021. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained.

[44] "Artificial Intelligence and Machine Learning in Healthcare," [Online]. Available: https://www.biomedcentral.com/collections/aiandmlinhealthcare.

[45] IEEE, "Machine Learning Applications in Healthcare," 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9964865.

[46] M. M. Amini, "Artificial Intelligence Ethics and Challenges in Healthcare Applications," 2023. [Online]. Available: https://www.mdpi.com/2504-4990/5/3/53.

[47] K. Staff, "5 Ethical Considerations for Implementing Machine Learning in Healthcare," 2023. [Online]. Available: https://kms-healthcare.com/5-ethical-considerations-for-implementing-machine-learning-in-healthcare/.

[48] "The Ethics of Machine Learning in Medical Sciences: Where Do We Stand Today?," *PubMed Central (PMC),* p. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7640783/, 2020.

[49] J. Bernal, "Transparency of Artificial Intelligence in Healthcare," 2022.

[50] "AI-powerd care coordination platform," [Online]. Available: https://www.viz.ai/.

[51] B. J. C. S. Senthil Pandi, Machine Learning, India: Shanlax Publications, 2021.

[52] "Glossary," ML Cheatsheet, [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/glossary.html. [Accessed 2024].

[53] J. S. John Doe, "Impact of AI on Health Informatics," *Journal of Health Informatics in Developing Countries,* no. https://jhidc.org/index.php/jhidc/article/view/389/388, 2023.

[54] IBM, "The benefits of AI in Healthcare," [Online]. Available: https://www.ibm.com/blog/the-benefits-of-ai-in-healthcare/. [Accessed 2024].

[55] "HITECH Act Summary," American Health Information Management Association, [Online]. Available: https://library.ahima.org/doc?oid=300027#.X4r9_lNKjBI. [Accessed 2024].

[56] "Understanding the HITECH Act," HealthIT.gov, [Online]. Available: https://www.healthit.gov/topic/privacy-security-and-hipaa/hitech-act-enforcement-interim-final-rule. [Accessed 2024].

Figure 69 Biomedical project GitHub repo

Figure 70 Fablabs: Project partner