PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University of Amar Telidji, LAGHOUAT

FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

MASTER THESIS

Field : Mathematics and Computer Science
Option : Computer Science
Specialization : Information System and Decision

Submitted by : Asmaa NEDJEM

Theme

Multimodal-based Pediatric Diseases Diagnosis

Jury members :

Dr.	Narjes HAMINI	MC(B)	(University of Laghouat)	President
Dr.	Zohra ABDELHAFIDI	MC(A)	(University of Laghouat)	Examiner
Dr.	Amel BELABBACI	MC(B)	(University of Laghouat)	Examiner
Dr.	Leila BENAROUS	MC(B)	(University of Laghouat)	Advisor

Academic Year 2023/2024

Dedications

I would like to dedicate this thesis to my family.
A special feeling of gratitude to my loving mother, Fatima Kired
whose words of encouragement and push for tenacity ring in my ears.
My sisters and my brother have never left my side, they are very special.
Last but not least, I am dedicating this to my father Noureddine
gone forever away from our loving eyes and who left a void never to be filled in our lives.
Though your life was short with me, I will make sure your memory lives on as long as I shall live.
May Allah (SWT) grant you Jannah Firdaws. I love you all and thank you all beyond words.
…

Acknowledgments

First and foremost, I thank Allah (SWT) Almighty for letting me live to see this thesis through. I am forever indebted to Dr. Leila BENAROUS for her unwavering support, encouragements and patience through this study. I can never pay you back for all the help you have provided to me, the experience you have helped me to gain, the precious time you spent making sure my thesis is always on track and helping me in editing my many mistakes. I hope you find some kind of satisfaction in this modest thesis. Thank you so much!

My sincere thanks go to Dr. Narjes HAMINI, Dr. Zohra ABDELHAFIDI, and Dr. Amel BELABBACI for agreeing to serve on my committee with their expertise and precious time.

Finally, I would like to thank the teachers and administrators in our university department of computer science for their efforts to providing decent education. Their excitement and willingness to provide feedback made the completion of this research an enjoyable experience.

ملخص

في مشروعنا، استهدفنا حل حالات التشخيص المتأخر أو الخاطئ لأمراض الأطفال. بالنسبة لهذا النموذج الأولي، ركزنا على تصنيف خمس أمراض أطفال فقط لإثبات طريقة العمل. الأمراض المختارة هي التي لها أعراض مرئية متشابهة مثل الجدري المائي، مرض كاواساكي، الحصبة، الوردية، والحمى القرمزية. للتصنيف، استخدمنا الصور التي تلتقط الأعراض البصرية والنصوص التي تمثل الأعراض الداخلية والخارجية للمريض. تم جمع مجموعة بيانات الصور من الانترنت، بينما تم إنشاء مجموعة البيانات النصية بشكل اصطناعي. لتصنيف الأمراض باستخدام الصور، استخدمنا شبكة ResNet50 العصبية، ولتصنيف الأمراض باستخدام النصوص، استخدمنا نموذج الذاكرة الطويلة قصيرة الأمد (LSTM). كان الهدف من استخدام النموذجين هو دمجهما لإنشاء نموذج واحد يمكنه تصنيف الأمراض بناءً على نوعين من المعلومات. كانت نتائج النماذج المدربة ResNet50 و LSTM والنموذج المدمج جيدة (بدقة تزيد عن 90 %). تطبيقنا النهائي هو تطبيق للهاتف المحمول يعمل على نظام أندرويد يستخدم النموذج المدمج لتمكين الأطباء من إجراء تشخيص سريع ودقيق لأمراض الأطفال.

الكلمات المفتاحية: تصنيف أمراض الأطفال، الرؤية الحاسوبية ، ResNet50، LSTM، التعلم متعدد الوسائط ، دمج النماذج، تطبيق اندرويد للهاتف

Abstract

In our project, we aimed to solve the pediatric diseases delayed or mistaken diagnosis cases. For this prototype, we concentrated on only five pediatric diseases classification as a proof of concept. The selected have confusing visual symptoms, they are Chickenpox, Kawasaki, Measles, Roseola and Scarlet fever. For the classification, we use both the images capturing the visual symptoms and textual symptoms representing the internal and external state of the patient. The image dataset was collected from public repositories, while the textual dataset was synthetically constructed. For images disease classification we used Residual Network (ResNet50) and for text disease classification we used Long Short-Term Memory (LSTM). The purpose behind using two models was to create one model by fusing them that can classify diseases based on two types of information. The results for the trained models ResNET50, LSTM and their fusion were good (over 90% of accuracy). Our final application is an android mobile application that uses the fused model to allow doctors to do prompt accurate diagnosis of pediatric diseases.

Keywords: Pediatric diseases classification, computer vision, ResNet50, LSTM, Multimodal Learning, Models Fusion, android mobile app.

Résumé

Dans notre projet, nous avons visé à résoudre les cas de diagnostics retardés ou erronés de maladies pédiatriques. Pour ce prototype, nous nous sommes concentrés uniquement sur la classification de cinq maladies pédiatriques à titre de preuve de concept. Les maladies sélectionnées présentent des symptômes visuels confus ; il s'agit de la varicelle, de la maladie de Kawasaki, de la rougeole, de la roséole et de la scarlatine. Pour la classification, nous utilisons à la fois des images capturant les symptômes visuels et des symptômes textuels représentant l'état interne et externe du patient. Le jeu de données d'images a été collecté à partir de dépôts publics, tandis que le jeu de données textuel a été construit de manière synthétique. Pour la classification des maladies à partir des images, nous avons utilisé le réseau de neurones résiduel (ResNet50) et pour la classification des maladies à partir des textes, nous avons utilisé la mémoire à long terme (LSTM). L'objectif derrière l'utilisation de ces deux modèles était de créer un seul modèle qui les fusionne, capable de classifier les maladies en se basant sur deux types d'informations. Les résultats des modèles entraînés ResNet50, LSTM et de leur fusion ont été bons (plus de 90 % de précision). Notre application finale est une application mobile Android qui utilise le modèle fusionné pour permettre aux médecins de poser un diagnostic rapide et précis des maladies pédiatriques.

Mots Clés : Classification des maladies pédiatriques, vision par ordinateur, ResNet50, LSTM, apprentissage multimodal, fusion de modèles, application mobile Android.

General Introduction

Health is always a top priority. However, medical errors are becoming more common in recent years putting patient life in danger. Consequences of these errors may be devastating especially for pediatric disease. A wrong or delayed diagnosis may be the reason of chronic diseases, permanent disabilities or even cause child's death.

Diagnostics intelligent apps have played an important role in solving this problem, but it is not intended by any mean to replace the doctor's role. Using pediatric diseases classification app, that take both visual and textual symptoms of a disease to recognize it the core aim of our developed application aiming the minimize the diagnostic errors in pediatric medicine.

The goal of this project is to create an android application that can classify the pediatric diseases by two types of inputs (texts and images). For this, we used complex type of deep learning known as multimodal learning. Initially, we used the ResNet50 and LSTM model separately, then combined them to obtain multimodal. Our thesis is divided into four chapters, the first two chapters are related to the medical and artificial intelligence theory and the last two chapters are for the contribution's backend and frontend development process, the content of each chapter is briefed below:

Chapter 1 focuses on analysis of statistics related to misdiagnosis cases and their fatalities as well as identify the diseases and their symptoms.

Chapter 2 provides a detailed overview of advances and achievements in computer vision and its techniques. It also includes explanation of multimodal learning, and its use in health sciences and medicine.

Chapter 3 explains our backend development process. From training separate unimodals to fusing them.

Chapter 4 depicts our final application and its development process. In which, we explain how we used the models exported from the backend in our android mobile application.

Lastly, we conclude the thesis and highlight our future perspectives

Contents

GENERAL INTRODUCTION....6
LIST OF FIGURES....VIII
LIST OF TABLES....VIII
CHAPTER 1. PEDIATRIC DISEASES....10
1.1 INTRODUCTION....11
1.2 STATISTICS RELATED TO MISDIAGNOSIS CASES AND THEIR FATALITIES....11
1.3 PEDIATRIC DISEASES GENERAL CLASSIFICATION....12
1.4 SELECTED DISEASES SYMPTOMS....12
1.5 MEDICAL FIELD INTERNSHIP EXPERIENCE....13
1.6 CONCLUSION....14
CHAPTER 2: COMPUTER VISION....15
2.1 INTRODUCTION....16
2.2 COMPUTER VISION EVOLUTION....16
2.3 COMPUTER VISION METHODS....17
2.3.1 CONVOLUTIONAL NEURAL NETWORKS....17
2.3.2 RECURRENT NEURAL NETWORK....20
2.4 COMPUTER VISION APPLICATIONS IN MEDICAL FIELD....23
2.5 MULTIMODAL LEARNING (MMDL)....23
2.5.1 MULTIMODAL LEARNING FUNCTIONING....23
2.5.2 MULTIMODAL LEARNING IN HEALTH SCIENCES AND MEDICINE....25
2.6 CONCLUSION....27
CHAPTER 3: MULTIMODAL TRAINING....28
3.1 INTRODUCTION....29
3.2 CONTRIBUTION EXPLANATION....29
3.2.1 Training the images model....31
3.2.2 Training the text model....32
3.3 SET UP (ENVIRONMENT)....33
3.3.1 Models Results....34
3.4 AI MULTIMODEL TRAINING....37
3.4.1 The proposed multimodal method....37
3.4.2 Fusion method application....37
3.5 EXPORTING THE MODEL....39
CHAPTER 4: FRONTEND DEVELOPMENT....40
4.1 AMELIORATION OF INITIAL DESIGN....41
4.2 CREATING THE FRONTEND....41
4.3 USING THE MODEL....43
4.4 EVALUATION IN TERM OF ACCURACY AND SPEED....43
4.5 CONCLUSION....44
GENERAL CONCLUSION....46
THE AUTO EVALUATION GRID....47
BIBLIOGRAPHY....48

List of Figures

Figure 1. 1. Initial design of our application. 14
Figure 2. 1. Computer Vision History [18]. 17
Figure 2. 2. Architecture of a CNN[22]. 18
Figure 2. 3. Architecture of Alexnet[23]. 18
Figure 2. 4. The Architecture of VGGNet (VGGNet-16) [24]. 19
Figure 2. 5. The Architecture of GoogleNet[25]. 19
Figure 2. 6. The Architecture of ResNet **[26].** 20
Figure 2. 7. (a) Mapping inside Residual block, (b) Simple direct mappings [21]. 20
Figure 2. 8. Recurrent neural network [28]. 21
Figure 2. 9. LSTM block[31]. 22
Figure 2. 10. Workflow of a typical multimodal. [48] 24
Figure 2. 11. Early, intermediate, and late fusion [51] 25
Figure 3. 1. Multimodal Pseudo Code 30
Figure 3. 2. The backend of our application training. 30
Figure 3. 3. Use case diagram 31
Figure 3. 4. Class diagram 31
Figure 3. 5. Training and Validation history 35
Figure 3. 6. Images Model Training Results 35
Figure 3. 7. Training and Validation history 36
Figure 3. 8. Text Model Training Results 36
Figure 3. 9. Multimodal Training History 38
Figure 3. 10. Multimodal Training Results 39
Figure 4. 1. Final design of our application. 41
Figure 4. 2. Enter Page. 42
Figure 4. 3. Switching between symptoms tabs. 42
Figure 4. 4. Project flow diagram. 43

List of Tables

Table 1. 1. The diseases symptoms. 13
Table 2. 1. Recapping table of the previous techniques. 22
Table 2. 2. Recap of related state of art 26
Table 3. 1. Images dataset 32
Table 3. 2. Our model hyper parameters 32
Table 3. 3. Texts dataset 33
Table 3. 4. Our model hyper parameters 33
Table 3. 5. The hardware characteristics 33
Table 3. 6. Backend Environment Setup 34
Table 3. 7. Our model hyper parameters 38
Table 4. 1. Colab Vs Android 44

Chapter 1. Pediatric Diseases

1.1 Introduction

Pediatric illnesses pose a challenge in term of diagnosis and treatment. Children are affected by the diseases more severely because they are still developing their immune systems. i.e., they are more vulnerable due to their immature immune response and their smaller bodies leading to serious reactions to pathogens.

As such, it is essential to diagnose the diseases the earliest and the diagnosis should be accurately correct to save the children's lives and protect them for long term consequences. A critical aspect of this understanding involves recognizing the complexities of misdiagnosis. The use of automated smart classification system to identify the diverse children's ailments and their symptoms is fundamentally important. The field of artificial intelligence (AI) is emerging as a powerful tool in the fight against childhood illness. AI technology has the potential to improve diagnostic accuracy, leading to earlier detection and more effective treatment for children.

In this chapter, we explain the importance in numbers of having these systems. Moreover, we illustrate the diseases general classification as well as the symptoms of the diseases subject to our study. Subsequently, we resume our internship experience that we had in 240 hospitals and its take outs related to understanding the diseases symptoms and the medical systems particular traits and needs. Lastly, we conclude the chapter.

1.2 Statistics related to misdiagnosis cases and their fatalities

Error in diagnosis known as misdiagnosis is the most identified reason in malpractice claims, because it constitutes a sizable proportion of medical errors and are responsible for significant costs and harm. Misdiagnoses can be the result of wrong or delayed understanding of symptoms especially for diseases with confusing signs [1]. In order to enhance our understanding about the relative prevalence of diagnostic errors, contributory factors, and potential preventive strategies, we resume the study recording diagnostic errors of pediatricians, between November 2008 and May 2009. For the experience, 1,362 survey were sent to physicians including academic physicians, trainees, and community-based physicians at : University of Texas, Baylor College of Medicine, and University of Cincinnati.

The overall response rate was 53%, 54% among which admitted committing diagnosis errors at least once or twice per month, trainee marked higher rate at 77%. As for the rate of diagnostic errors that harmed patients at least once or twice per year was reported to be 45% [1]. The findings of this study, although old, highlights the frequent occurrence of diagnosis errors in pediatric care which can potentially harm patients. This further underlines the importance of having assisted intelligent tools to aid in providing accurate and timely diagnosis.

1.3 Pediatric diseases general classification

Children grow and their bodies change rapidly, and the growth rate differs from one child to another, as such the reactions to the pediatric diseases and severity differs as well. This presents a challenge to do classification for these diseases, therefore, a systematic classification based on selected criteria is needed, among these are the symptoms patterns and signs of the disease, hereditary (genetic), etiology (causes) -infections and contaminations, gender and environment [2]. In addition to, the affected organ system, age of onset, and disease severity. By employing a multi-layered classification system based on multiple criteria, the understanding of these diseases becomes easier, ultimately paving the way for more effective diagnosis, treatment, and improved health outcomes for children.

1.4 Selected diseases symptoms

For the subject of this study, we selected 5 diseases that infect the children which are Kawasaki, Scarlet Fever, measles, chickenpox, roseola. These diseases vary in their severity from mild to perilous leading to death. They impact children whose age ranges from 6 months old to 15 years old. Moreover, these diseases have internal and external symptoms such as skin rashes. A rash is a common symptom, but the specific characteristics of the rash, such as color, texture, and shape, can provide valuable clues for distinguishing between these different diseases. Beyond the rash, other symptoms like fever, sore throat, and swollen lymph nodes also play an important role in the diagnostic process. By understanding these key indicators, we can be more vigilant in identifying potential diseases in children and seek appropriate medical attention. We collected both of these symptoms (textual and images) to use them in our classification, Table 1.2 resumes them.

- **Kawasaki :** also known as Kawasaki syndrome, was first described in Japan by Tomisaku Kawasaki in 1967. It is an acute non contagious febrile illness [3] [4].
- **Scarlet fever :** is a bacterial illness group A streptococcus, that develops in some people who have strep throat. It is an acute infectious disease, the infection spreads from person to person by droplets released when an infected person coughs or sneezes [5] [6].
- **Measles :** is a highly contagious viral disease that often affects children and can cause very serious complications caused by a virus [7] [8].
- **Chickenpox :** is an illness caused by the varicella-zoster virus. Chickenpox spreads very easily to unvaccinated people and those who never had this disease before [9] [10].
- **Roseola :** is an illness caused by the varicella-zoster virus. Chickenpox spreads very easily to unvaccinated people and those who never had this disease before [11] [12].

	Disease	Scarlet fever	Kawasaki	Measles	Chickenpox	Roseola
	Image					
	Age rang	All ages, most children 5-15 years old	Kids until 15 years old	All ages	Kids 3-15 years old and maybe adult	Kids 6 months until 2 years old
General symptoms	Fever	39.5C°- 40 C°	39 C°- 40 C°	40.5 C°- 41 C°	38 C°- 40 C°	39.5 C°- 40 C°
	Fatigue	NO	NO	NO	YES	NO
	Chills	YES	NO	NO	NO	NO
	Pain	Chest pain, Headache and body-aches	Joint pain	NO	Headache	NO
Skin symptoms	Rash	Red lines, Red rash, Flushed face	Red rash, Red, Swollen skin-hands-feet, Peeling skin	A skin rash made up of large, flat blotches	Yes	Pink rash
	Tongue observation	Red tongue with white coating	Red, Swollen tongue	NO	NO	NO
	Spots	Patches throat	NO	Koplik's spots	Small, fluid-filled blisters	NO
	Lips observations	Pale ring around mouth	Red, Dry, Cracked lips	NO	NO	NO
	Eyes observations	NO	Red eyes with thick discharge	Inflamed eyes	NO	Swollen eyelids
Behavioral Symptoms	Irritability	NO	YES	NO	NO	YES
	Loss of appetite	NO	NO	NO	YES	YES
Respiratory symptoms	Nose discharges condition	NO	NO	Runny nose	NO	Runny nose
	Throat condition	Patches throat Sore, red throat	Coronary artery injury	Sore throat	NO	Sore throat
	Coughs	NO	NO	YES	NO	YES
	Breathing & Swallowing	Trouble-Rapid-Noisy breathing and Difficulty Inability swallowing	NO	NO	NO	NO
	Lymphatic Symptoms (Swollen Lymph Nodes)	Enlarged lymph	Enlarged lymph	NO	NO	Swollen lymph nodes
Gastrointestinal Symptoms	drooling	YES	NO	NO	NO	NO
	diaherra	YES	YES	NO	NO	YES
	vomiting	YES	YES	NO	NO	NO
	Abdominal pain	YES	YES	NO	NO	NO
	dehydration	YES	NO	NO	NO	NO
	Duration	1 week	1-2 week	2-3 week	4-6 week	1-2 week

Table 1. 1. The diseases symptoms.

1.5 Medical field internship experience

It is important to develop medical application, as they have a positive impact, which would provide accurate prompt diagnosis of diseases to doctors. These systems do not replace doctor's role, but they are made to assist them, facilitate their tasks, and populate their expertise. The establishment of these apps cannot be accomplished without the help and cooperation of doctors as machines cannot solve problems that human fail to solve, and any AI-app would not be possible without the human expertise that will be translated to the machine. In matter of fact, the AI-based apps are developed to generalize the expert knowledge and automate it to assist junior or non-specialized doctors with less experiences.

Thus, to be able to develop my application, it was mandatory for me to get a training and meet doctors to understand the symptoms and the diseases. For this purpose, I did two weeks internship at the pediatric service of the Mixed hospital, 240 beds, Laghouat. Through which I achieved my goals which were to confirm the symptoms and understand them and learn more about how to differentiate between them. After which I was able to establish my dataset (pictures and text). During the internship, I was introduced the medical software DEM available in most Algerian hospitals used to manage the medical files of patient and follow their progress, I got inspired from its interface to make my application design. Furthermore, I surveyed doctors' opinion regarding their readiness and willingness to use an artificial intelligence system to assist them in the diagnosis of diseases to which they showed their interest, excitement, and acceptance. Lastly, I sensed their requirements and preferences related to the design they expect for my application and showed them its interface (see Figure. 1.1) and ameliorated it according to their feedback. Noting that due to privacy and confidentiality regulations, it was not possible to obtain real datasets of patients related to these diseases as it infringes their rights and we could not obtain their consent to use it, because most of them were not present at the hospital and their contact information were not disclosed to us.

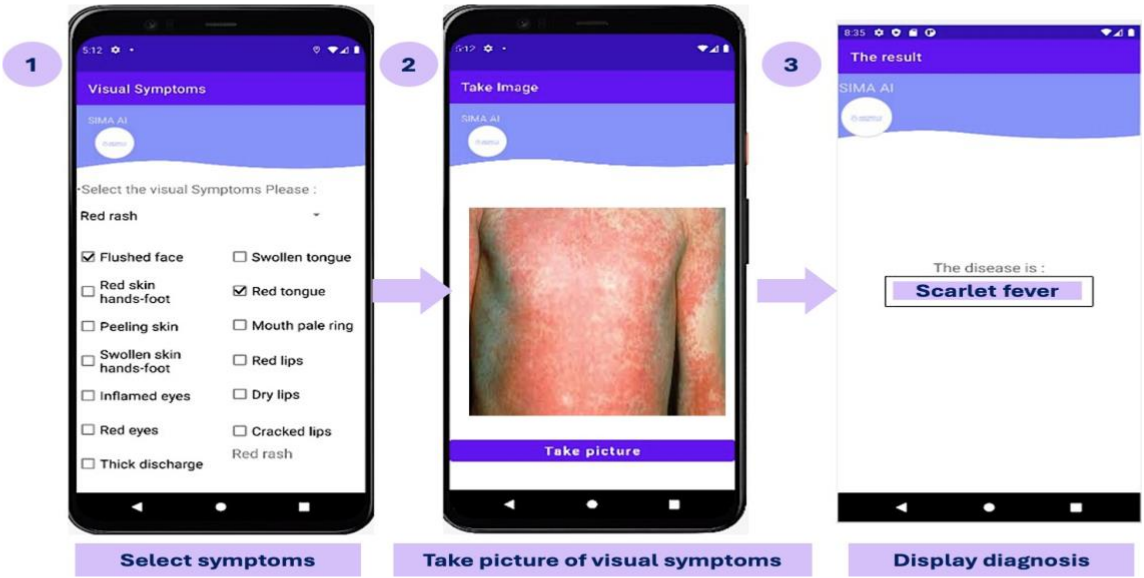


Figure 1. 1. Initial design of our application.

1.6 Conclusion

Pediatric diseases are critical and challenging due to the particular nature of a child's body and physiology. Misdiagnosis can have severe consequences, which highlights the importance prompt accurate diagnosis. This chapter provided statistics emphasizing the prevalence of misdiagnosis and its potential fatalities. In which we reviewed the general classification of pediatric diseases and discussed the symptoms of selected cases. In addition to recapping the knowledge gained from our short internship. This knowledge serves as the foundation to build a clean descriptive dataset that we use to create accurate computer vision based pediatric disease diagnostic system.

Chapter 2: Computer vision

2.1 Introduction

Computer vision stands as one of the key applications of Artificial Intelligence (AI). It fundamentally allows machine to smartly recognize the outer environment and interpret the images it captures. This field encompasses a vast area of research dedicated to developing various image processing algorithms. Among these methods, there are methods based on machine learning, and deep learning. This intricate process involves acquiring images or video data, meticulously analyzing it, and employing sophisticated processing techniques to extract meaningful information. Ultimately, the goal is to enable machines to not only see the world but also to interpret what it sees. In this chapter, we explain the computer vision techniques and their evolution through time, its functioning and usage in medical field. Then, we highlight its limits and how we could overcome them through employing multi-model learning and we explain the functioning of multi-model learning and its usage in medical field [13].

2.2 Computer vision evolution

Computer vision (CV) was defined as a subfield of AI that deals with automating the analysis of visual data, by building algorithms that can understand the content of images then extracts information from it [14] [15]. Authors of [16], defined the computer vision as a branch of computer science that studies how computers can see. In matter of fact, the scientist showed their interest about as early as in the 1950s. Frank Rosenblatt was the first to make the perceptron, which was considered as the initial step toward having computers that learn patterns like humans. After that David Marr introduced other CV techniques such as edge detection and segmentation. Even so, a long time was taken before arriving to the propose goal in the evolution of CV, until deep learning came along in the late 2000s.

Later, recognizing objects in images using computers became possible in 2010 because of convolutional neural networks (CNNs) evolution, which contain many techniques like ImageNet, and AlexNet that are characterized by their high accuracy. After that, more advanced computer programs like ResNet, VGGNet, and InceptionNet were developed, making computers even better at recognizing things in pictures [17]. The keystones of computer vision evolution through history are depicted in Figure. 2.1.

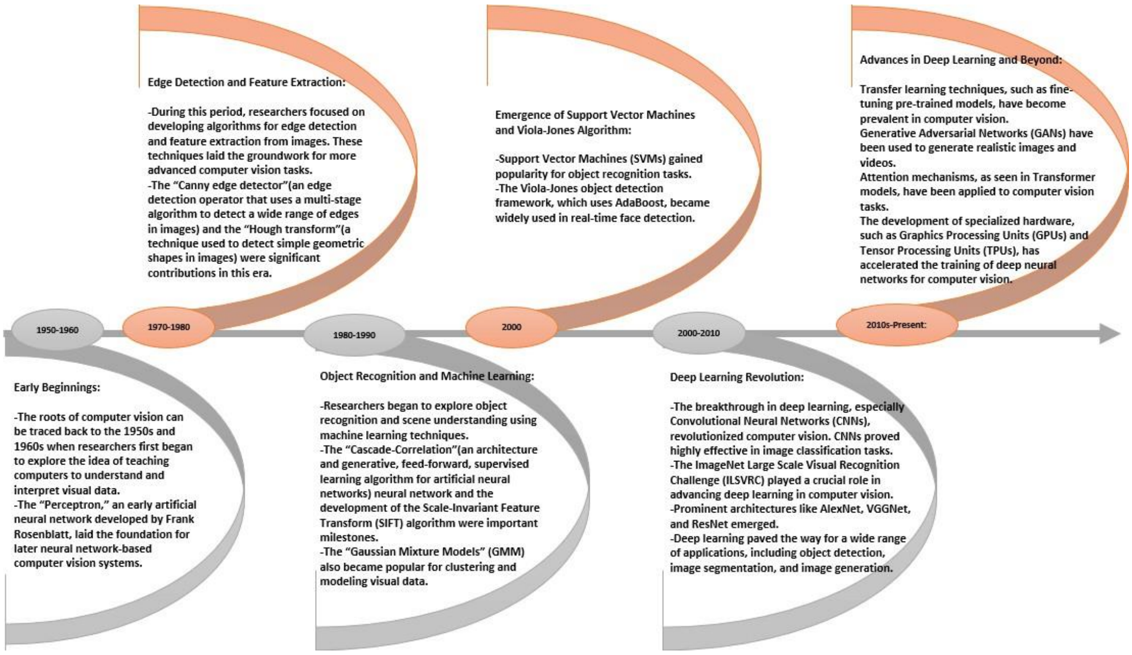


Figure 2. 1. Computer Vision History [18].

2.3 Computer Vision Methods

The type of information gained from an image through CV can vary from identification to space measurements in order to make computers interpret, describe, and respond to images and videos even in real-time. Computer vision has a lot of techniques, in the next subsections. We brief the most important ones.

2.3.1 Convolutional Neural Networks

Convolutional Neural Networks is a type of deep learning model. They consist of one or multiple convolution layers extract features from input by executing convolution ope- rations. Because each layer is composed of a collection of nonlinear functions of weighted sums at various coordinates of adjacent subsets of the outputs from the previous layer, the weights can be reused. Using CNN enables automatic extraction of features from a set of photos or videos in order to accomplish a certain goal, such as : image classification, face authentication, and image semantic segmentation [19] [20]. Figure. 2.2 shows the architecture of the CNN and the layers of it where the convolution layer aims to feature extraction, it followed by a pooling layer which reduces the in-plane dimensionality of the feature maps in order to introduce a translation invariance to small shifts and distortions, and decrease the number of subsequent learnable parameters, the sequence of convolution and pooling layers are repeated allowing the model to learn the features. It is followed by flatten phase aiming to transform the output feature maps of the final convolution or pooling layer into a one-dimensional (1D) array of numbers (or vector), and then a fully connected layer (also known as dense layers) to connect every input to every output by a learnable weight. Lastly, the softmax layer is used to do the classification in multi-class classification model [19] [21].

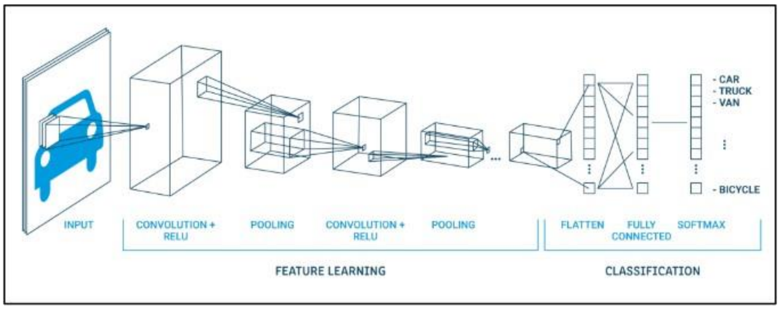


Figure 2. 2. Architecture of a CNN[22].

In the following subsections, we explain some of CNN architecture [21] :

- **AlexNet :** Inspired from LeNet, Krizhevky et al [21] [20], as the first CNN neu- ral network to win the ImageNet Challenge in 2012, which is designed to classify ImageNet data. It consists of eight weighted layers among which the first five convo- lution layers and three fully connected layers. Since it was designed for ImageNet data, so the last output layer classifies the input images into one of the thousand classes of the ImageNet dataset with the help of 1,000 units as shown in Figure. 2.3. Thus, AlexNet requires 61 million weights and 724 million MACs (multiply-add computation) to classify the image with a size of 227*227.

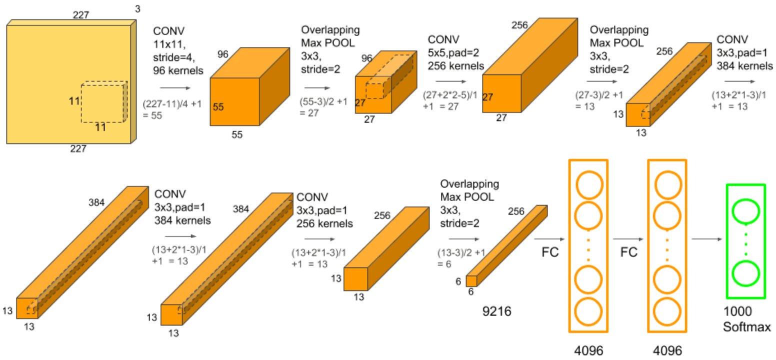


Figure 2. 3. Architecture of Alexnet[23].

- **VGGNet :** Simonyan and Zisserman released the VGGNet in 2014 as a new CNN architecture. They introduced a total 6 different CNN configurations, among them the VGGNet-16 and VGGNet-19 are the most successful ones. VGG-16 is trained to a deeper structure of 16 layers consisting of 13 convolution layers and three fully connected layers, requiring 138 million weights and 15.5G MACs to classify the image with a size of 224224 [21] [20]. The architecture of VGGNet-16 is shown in Figure. 2.4.

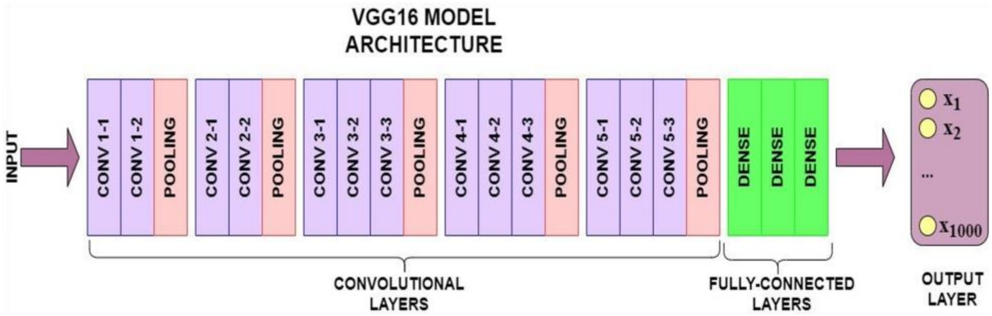


Figure 2. 4. The Architecture of VGGNet (VGGNet-16) [24].

- **GoogleNet :** Distinguished from the previously explained CNN models, the GoogleNet uses network branches instead of single line sequential architecture. The GoogleNet was proposed by Szegedy et al. [21] [20] in 2014. It is composed of 22 weighted layers. It uses Inception Module as the fundamental building block of the network that composed of different sized filters (as shown in Figure. 2.5).

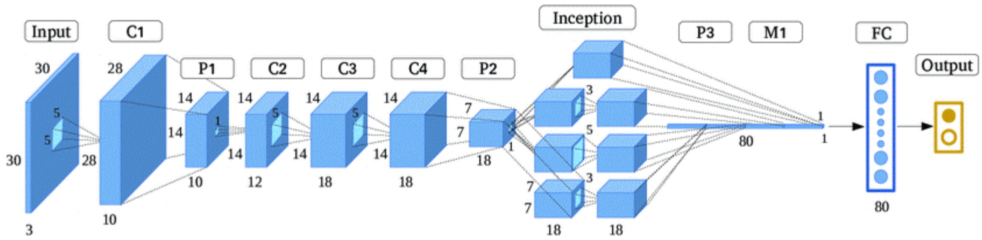


Figure 2. 5. The Architecture of GoogleNet[25].

- **ResNet :** Since a deep CNN model suffers from vanishing gradient problems. Microsoft, introduced in ResNet model the idea of identity skip connection to it. The ResNets architecture use residual mapping (H(x) = F(x) + x) instead of learning a direct mapping (H(x) = F(x)) and these blocks are called residual bocks. ResNet, the state-of-the-art effort, uses the shortcut structure to reach a human-level accuracy with a top-5 error rate below 5%. The complete ResNet architecture is consist of many residual bocks with 3*3 convolution layers, where, **F(x)** is a function that takes the input **x** and produces a learned representation. It typically consists of one or more 3x3 convolutional layers, followed by batch normalization and ReLU activation. The function **F(x)** is learned during training and is designed to capture the underlying patterns and features in the input data, **H(x)** represents the output of a residual block. Figure. 2.6 shows the architecture of ResNet. Figure. 2.7 illustrates the difference between the direct mapping and the residual mapping [20] [21]

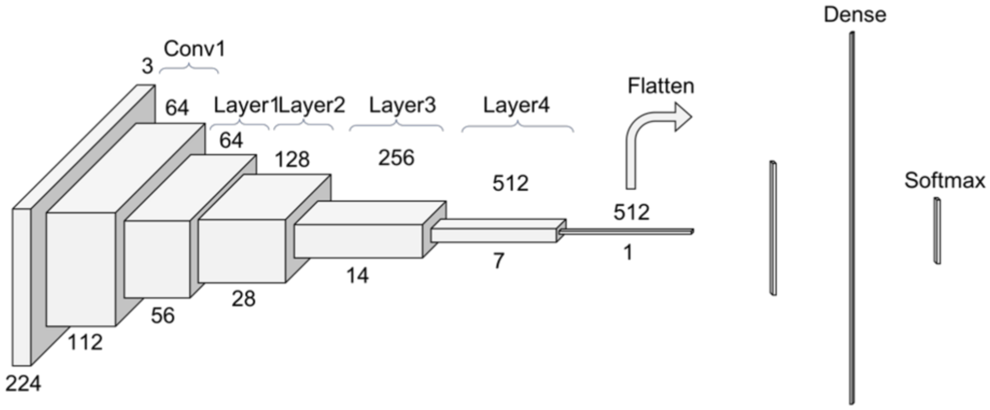


Figure 2. 6. The Architecture of ResNet **[26].**

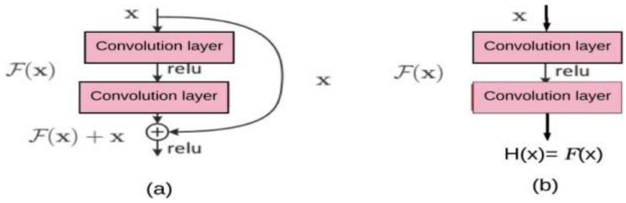


Figure 2. 7. (a) Mapping inside Residual block, (b) Simple direct mappings [21].

The authors propose several versions of ResNet with different depth, and they also used bottleneck layer for dimensionality reduction in each ResNet architecture that has depth more than 50. Although the ResNet (with 152 Layer) is 8 times deeper than VGGNets (22 layers), it has complexity lower than VGGNets (16/19) [21].

2.3.2 Recurrent neural network

Recurrent neural networks (RNN) were designed to tackle exactly the time-series problem of sequential input data, unlike of CNN that use to solve problems involving spatial data, such as images. RNNs achieves that by using feedback loops known as recurrent cells that can recall and detect patterns in those sequences. The recurrent cells enable the network to retain information over time. As such the input of RNN hidden layer consists of the current input and the previous samples. The recurrent cells update their internal states in response to the new input, enabling the RNN to identify relationships and patterns [20] [27], see Figure. 2.8 for illustration. The nodes in Figure 2.8 represent neurons or cells in the RNN. Each node processes the input data and maintains a hidden state, which allows the network to capture temporal dependencies in the data. The links represent the flow of information. In an RNN, the links allow the information to flow from one time step to the next, enabling the network to capture sequential relationships in the data. Loops represent the recurrent connections. These connections allow the information to flow from the current time step back to a previous time step, creating a feedback loop. This feedback loop enables the network to maintain a hidden state that captures information from previous time steps, allowing it to learn from sequential data.

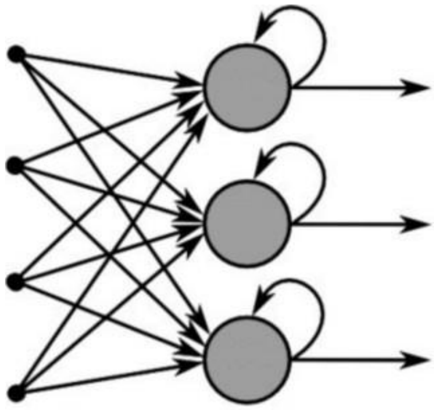


Figure 2. 8. Recurrent neural network [28].

- **Long Short-Term Memory (LSTM):** follows the read-write-and-forget principle which means that when predicting the output, the neural network reads and writes only the most useful part of input information, and it forgets the useless part. The memory block is the essential unit of an LSTM network. Each memory cell has at its core a recurrently self-connected linear unit called the Constant Error Carousel (CEC). By recirculating activation and error signals indefinitely, the CEC provides short-term memory storage for extended time periods. The input, forget, and output gate can be trained to learn, respectively, what information to store in the memory, how long to store it, and when to forget it out [29] [30] see Figure. 2.9 for illustration, where :
  - **c(t-1)** : This is the cell state from the previous time step. It represents the internal memory of the LSTM block.
  - **input gate** : This is a sigmoid activation function that determines how much of the new input should be allowed to update the cell state.
  - **forget gate** : This is another sigmoid activation function that determines how much of the previous cell state should be forgotten.
  - **input activation function (g)** : This is usually a tanh activation function that scales the new input to be between -1 and 1 before it is multiplied by the input gate.
  - **F(t)** : This is the output of the input activation function, which is then multiplied by the input gate to update the cell state.
  - **peephole connections** : These are connections from the cell state to the input, forget, and output gates. They allow the gates to have direct access to the cell state, which can improve performance.
  - **output gate** : This is a sigmoid activation function that determines how much of the current cell state should be output.
  - **output activation function (h)** : This is usually a tanh activation function that scales the output of the output gate to be between -1 and 1 before it is output as the hid- den state.

- **C(t)** : This is the updated cell state, which is a function of the previous cell state, the input gate, the forget gate, and the input activation function.
- **y(t-1)** : This is the hidden state from the previous time step. It is input to the LSTM block along with the new input.
- **Z :** This is the weight matrix that is multiplied by the concatenated input and hid- den state to produce the input, forget, and output gates.
- **+1** : This is a bias term that is added to the weighted input before it is passed through the activation functions.
- **σ :** This represents the sigmoid activation function.
- **Arrows** : The arrows in the figure represent the flow of data through the LSTM block. They indicate the connections between the nodes and the direction of the data flow.

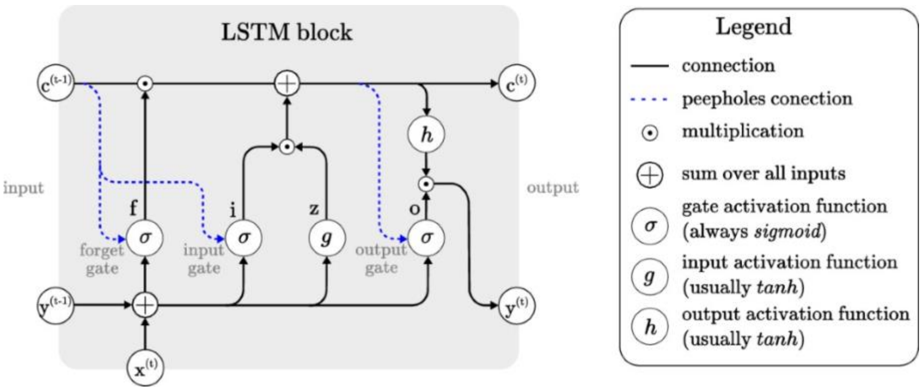


Figure 2. 9. LSTM block[31].

Table. 2.1 show more features details of the techniques that we had explained before :

Table 2. 1. Recapping table of the previous techniques.

Model	Characteristic	Architecture		Accuracy
AlexNet [32] [33]	Uses ReLU as an activation function, dropout regularization, and SGD Momentum as a learning algorithm.	**Input**	Images (227*227)	84.7% (on ImageNet dataset)
		Layers	13 layers (5 Convolutional, 3 Max pooling, 2 normalized, 2 fully connected, 1 softmax)	
VGGNet [34]	Uses ReLU as an activation function	**Input**	Images (224*224)	92.7% (on ImageNet dataset)
		Layers	14 layers (5 [3*3] Convolutional, 5 Max pooling, 3 fully connected, 1 softmax)	
GoogleNet [35] [36]	Uses ReLU as an activation function and uses auxiliary classifiers	**Input**	Images (224*224)	93.3% (on ImageNet dataset)
		Layers	22 layers (convolutional layers, pooling layers, Inception modules, fully connected layers, and softmax)	
ResNet [37]	Solve the issue of the vanishing/ exploding gradient, uses Batch Normalization at its core	**Input**	Images (224*224)	96.43%(on ImageNet dataset)
		Layers	ResNet50 uses a 34-layer plain network architecture	
LSTM [38] [39]	Capable of learning long term dependencies in sequential data, can recall previous long-term time-series data, has automatic control for retaining or discarding features	**Input**	Any of sequential nature	more than 94% (on Sarcasm Detection dataset)
		Layers	Input gate Forget gate Memory cell Output gate	

2.4 Computer vision applications in medical field

Computer vision involves tasks such as object detection, image classification, and segmentation. Medical imaging can greatly benefit from recent advances in image classification and object detection. Research studies have demonstrated promising results in complex medical diagnostics tasks such as dermatology, radiology, or pathology. Deep-learning systems could aid physicians by offering second opinions and flagging concerning areas in images [40] such as segmenting breast tumor images [41].

Its usage was particularly useful during COVID-19 pandemic for social distancing [42], hand hygiene detection [43] and COVID-19 contamination detection (did not defined the types of images that have used).

Also, authors [44] proved that computer vision can help to power solutions for the surgical operating room (OR). that provide real-time guidance for the doctors during surgical procedures.

In another side, authors [45] create Computer Vision System which include Home- based Physical Therapy, that is capable of customizing exercises, capturing exercise information, evaluating patient performance, providing therapeutic feedback to the patient and the therapist, checking the progress of the user over the course of the physical therapy, and supporting the patient. Moreover, computer vision can help the visually impaired as in [46] which incorporates an AI based voice interface.

2.5 Multimodal Learning (MMDL)

In medical field, doctors do not rely only on visually seen symptoms to diagnose a disease, they may need X-rays, MRI scans, blood analysis and other vital measurements to get a deeper understanding. As such, to do accurate diagnosis, models trained to understand each type of data are needed and combining them all together to do the disease identification is required, this concept is known as the multi-model learning [47]. In the following subsections, we explain the functioning of the multimodal learning and its application in medical field.

2.5.1 Multimodal Learning Functioning

Multimodal learning learns and improves performance through the use of multi data- sets that could be of different types. It is a machine learning subfield that aims to train AI models to process and find relationships between different types of data (modalities) typically, images, video, audio, and text [48].

Multimodal neural networks combine different types of unimodal neural networks. For example, an audiovisual model might have one network for videos (sequence of images) and another for audio. Each network processes its own data separately, this mechanism is known as the encoding. After encoding, the models are merged using various techniques, like concatenation, or attention. Data should be grouped as well. Once fused (both data and models), a final decision network takes this combined information and fused model to train it for the original task. To put it simply, multimodal architectures usually consist of three parts : Unimodal encoders that encode a single type of dataset ; a fusion network that combines the features extracted from each dataset input type, during the encoding phase ; and ; a classifier that accepts the fused data and makes predictions like shown in this Figure. 2.10 [48] [49].

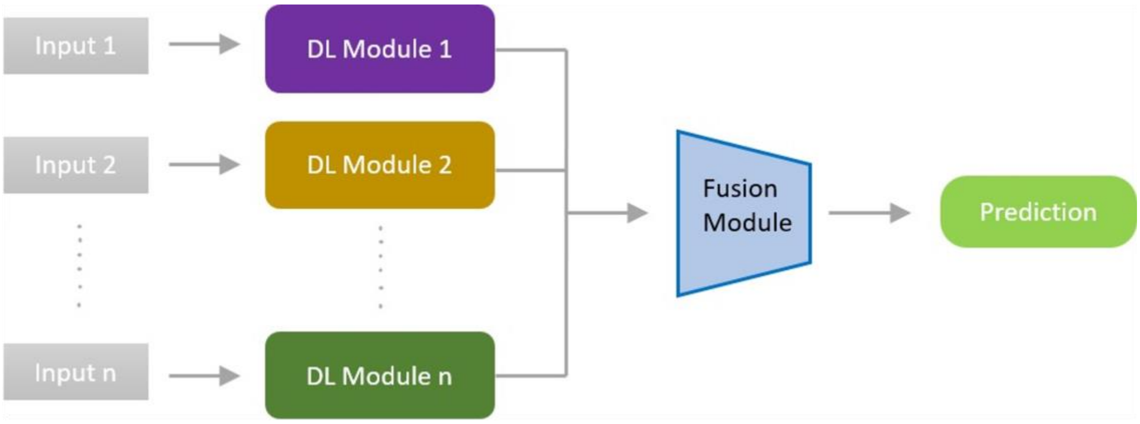


Figure 2. 10. Workflow of a typical multimodal. [48]

There are multiple steps to do the multi model learning, starting by the representation, translation, alignment, and fusion. We explain in the following subsections each of these methods :

- **Representation :** one of the challenges in multimodal learning is to represent and summarize data. The heterogeneity of multimodal data makes it challenging to construct a representation that considers the complementarity and redundancy of multiple types [48] [50].
- **Translation :** A second challenge addresses the translation of data and mapping data from one modality to another [50] [48].
- **Alignment :** A third challenge is to identify the direct relations between (sub)elements from two or more different modalities. Current research in multimodal learning aims to create modality-invariant representations. This means that when different modalities refer to a similar semantic concept, their representations must be similar/close together in a latent space [50] [48].
- **Fusion :** upon doing all of the above preparative steps related to having the correct representation of multi types of data, the last step is joining the models trained uniquely on each type of data into one model to perform a prediction task. Thus, finding the most optimal fusion technique is necessary. The fusion techniques include simple operations such as concatenation or weighted sum, and more sophisticated

attention mechanisms such as transformer networks, or attention-based recurrent neural networks [48] [50] .There are three types of fusion : early, intermediate/joint, and late. In early fusion, the inputs are combined before passed to the model. In the intermediate fusion, the outputs from one model become inputs for another. In the late fusion, each model is trained separately on unique data type and all of them are ensembled together and/or voting occurs between their inferences to decide which models result to consider. Figure. 2.11 explain the fusion types the early, the intermediate and the late [51].

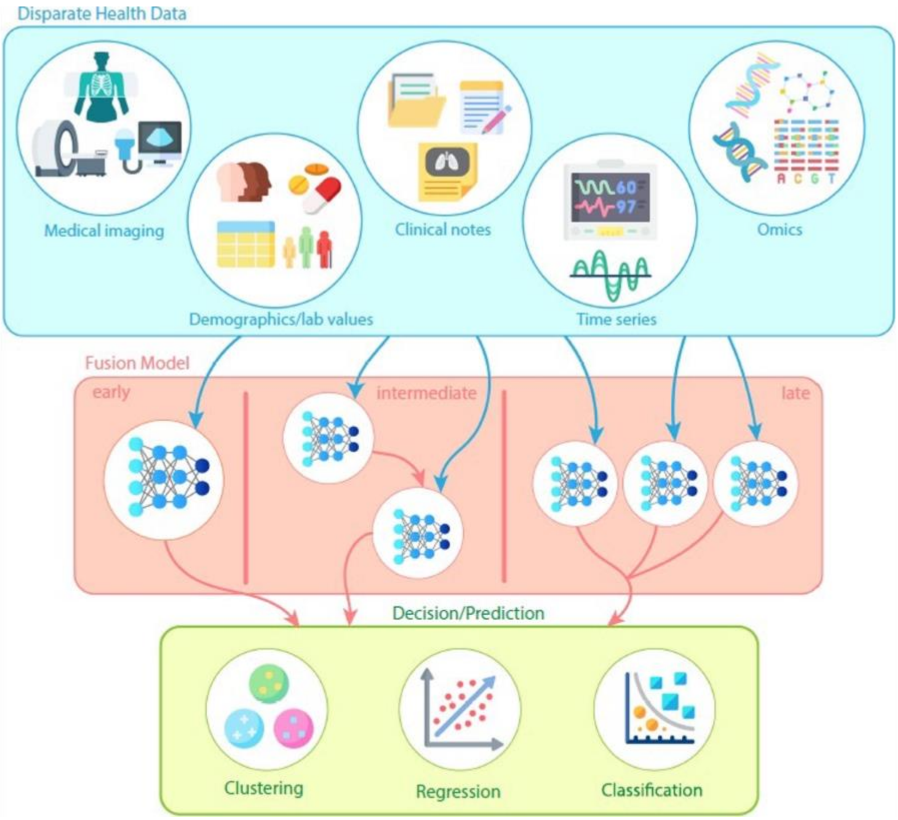


Figure 2. 11. Early, intermediate, and late fusion [51]

2.5.2 Multimodal Learning in Health Sciences and Medicine

Multimodal learning has been used in medical field to provide accurate diagnosis for complex hard to diagnose diseases. Authors [52] based multimodal disease risk prediction using patients structured data and text data, by using deep belief network (DBN) to fuse the features. Authors [53] collect their data from thousands of patients to do Alzheimer's research.

Authors of [54], [55] used multimodal learning to brain tumors diagnosis. They obtained images of magnetic resonance (MR) and computed tomography (CT) scanners, then they combined it using specific fusion technique which is "morphology filters and pyramids". Authors of [56] chose genomics, imaging and electronic health record (EHR) as datatype which they get it from 500,000 individuals to do various tasks as Alzheimer's research, with the same datatype as [53]. There are more uses of multimodal as fusing heterogeneous data (biomarkers) for Alzheimer's disease classification [57]. Authors [58] used for data type genomics, imaging that have collected from thousands of patients to do cancer research. Authors of [59] did various tasks among which diseases diagnostic with data type imaging, text and various modalities respectively which the dataset were more than millions. For multimodal assessment of Parkinson's disease, authors [60] used information from speech, handwriting, and gait by using recurrent convolutional neural network. Also, there is an unsupervised learning approach to identify novel signatures of health and disease from multimodal data [61], by using 1385 data features collected from diverse modalities, including metabolome, microbiome, genetics, and advanced imaging, from 1253 individuals that utilized a combination of unsupervised machine learning methods to identify multimodal biomarker signatures of health and disease risk. Authors of [62] used multimodal learning to pathogenic factors identification of brain imaging and gene in late mild cognitive impairment (LMCI). Resting functional magnetic resonance imaging (rfMRI) and gene data are obtained from 62 subjects (36 LMCI and 26 normal controls), then they combined it using Pearson correlation analysis technique where they obtained satisfied results and height accuracy. Table 2.2 resumes the state of art works by highlighting their aim, used technique and inputted dataset.

Table 2. 2. Recap of related state of art

Reference	Year	Datatype and source	AI Technique used	Task
[52]	2000	structured data and text data	Deep belief network (DBN)	Disease risk prediction
[53]	2008	Imaging, Genomics, EHR	Deep Learning	Alzheimer's research
[54]	2012	Various modalities	Deep Learning	Various tasks such as diseases classification and cancer research
[55]	2014	MRI, computed tomography (CT)	Deep Learning	Brain Tumors diagnosis
[56]	2014	Genomics, Imaging, EHR	Deep Learning	Various tasks such as Alzheimer's research
[57]	2015	All biomarkers	Deep Learning	Alzheimer's Disease Classification
[58]	2015	Genomics, Imaging	Deep Learning	Cancer research
[59]	2016	Imaging, Text	Deep Learning	Various tasks such as diseases diagnostic
[60]	2019	Speech, handwriting, and gait	RNN	Assessment of Parkinson's Disease
[61]	2020	Metabolome, Microbiome, Genetics, and advanced imaging	Unsupervised machine learning	Identify multimodal biomarker signatures of health and disease risk
[62]	2021	MRI, gene data	Deep Learning	Pathogenic Factors Identification of Brain Imaging and Gene in Late Mild Cognitive Impairment

2.6 Conclusion

In this chapter, we explained the evolution of computer vision in the era of artificial intelligence. Also, we concentrated on its application in medical field. Moreover, we described the functioning of ResNet50 model which we are using to identify diseases from pictures and LSTM which is used to identify diseases from textual symptoms. Lastly, we explained the multimodal learning which is the core of our application, which in our case combines the ResNET50 and LSTM to have a model that can identify pediatric diseases from images and textual symptoms. In the next chapter, we illustrate the technical and programmatical details of creating the backend of our application which is the phase related to training the models.

Chapter 3: Multimodal training

3.1 Introduction

Making mistakes is a human trait, however, some mistakes are not allowed mainly those that endanger human lives, such as making medical mistake or malpractice. The reasons for such errors may be related to lack of knowledge, keen sight, and precision. Especially, with diseases having partial common symptoms being it internal or external. While expert doctors may easily be able to distinguish these diseases and diagnose them accurately and promptly, the case may not be the same for junior and/or nonspecialized doctors. Children's case is more sensitive as a misdiagnosis case may lead to severe consequences in the future, from developing chronical diseases, to permanent disabilities and to death in dire cases. As such, we thought of making a pediatric diagnosis application that uses natural language processing and computer vision to classify the diseases from textual and visual symptoms. For the sake of this, we selected five pediatric diseases that have partial common symptoms which are: scarlet fever, Kawasaki, measles, chickenpox, and roseola. The symptoms of these diseases were previously explained in chapter 1, Table. 2.1. For the multimodal training we used the models which we explained in chapter 2. We used ResNet50 for image classification, and LSTM for text classification. Then, we combined between these two models to obtain the final multi-trained model. The details of each step are given in the following sections. In the next chapter, we explain how the model is used from our developed mobile application.

3.2 Contribution explanation

To be able to make our proof of concept (prototype), we had to first search about the selected diseases visual and textual symptoms. Then, we had to talk with doctors to understand the medical terminology and confirm our findings. Subsequently, we selected the models, trained them separately on each type of dataset, the fused them in one model. Figure. 0.1 define how our multimodal work in pseudo code. Figure. 3.2 explains the building process of the backend of our application mainly related to the training and inference parts.

```python
def train_image_model(images):
  # Train the image model using the images dataset.
  image_model = train_model(images)
  return image_model

def train_text_model(texts):
  # Train the text model using the texts dataset.
  text_model = train_model(texts)
  return text_model

def fusion_module(image_model_output, text_model_output):
  # Fuse the outputs of the image and text models.
  fused_output = fuse(image_model_output, text_model_output)
  return fused_output

def predict(fused_output):
  # Make a prediction based on the fused output.
  prediction = make_prediction(fused_output)
  return prediction

# Load the images and texts datasets.
images = load_images()
texts = load_texts()

# Train the image and text models.
image_model = train_image_model(images)
text_model = train_text_model(texts)

# Get the outputs of the image and text models.
image_model_output = image_model.predict(images)
text_model_output = text_model.predict(texts)

# Fuse the outputs of the image and text models.
fused_output = fusion_module(image_model_output, text_model_output)

# Make a prediction based on the fused output.
prediction = predict(fused_output)

# Print the prediction.
print(prediction)
```

Figure 3. 1. Multimodal Pseudo Code

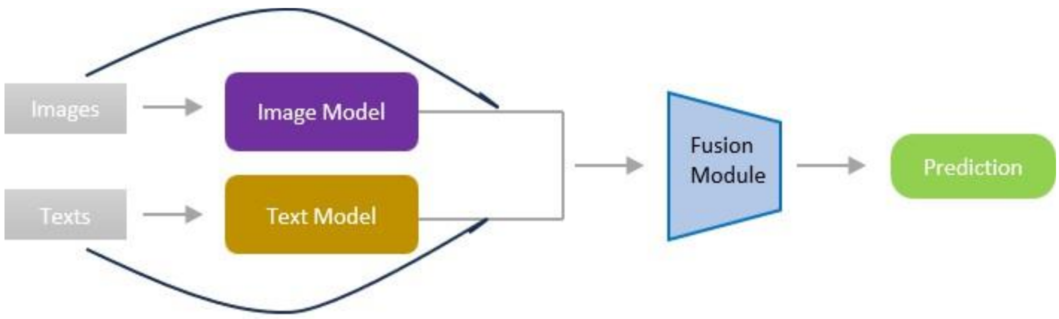


Figure 3. 2. The backend of our application training.

Here are more details about the using of our project, Figure. 3.3 represent the use case diagram and Figure. 3.4 the class diagram:

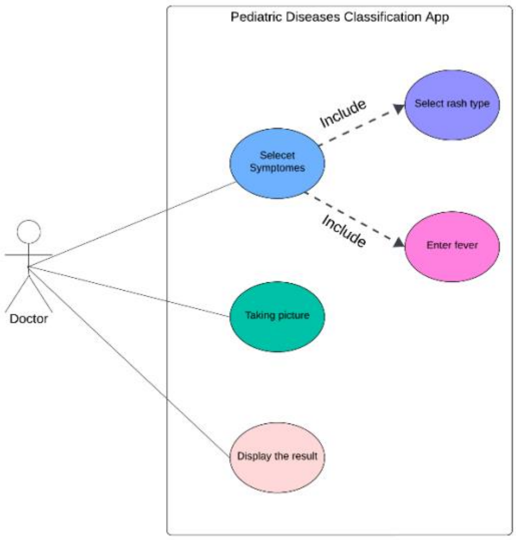


Figure 3. 3. Use case diagram

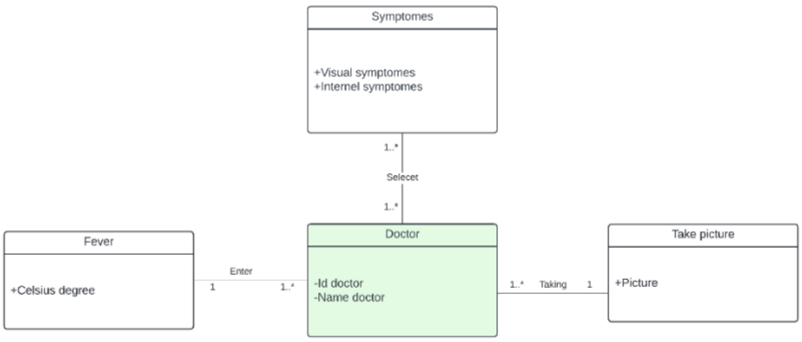


Figure 3. 4. Class diagram

3.2.1 Training the images model

Prior to deciding which model to use for the image classification, we first read the literature to find the commonly used models for medical image classification. From reading related literature such as ([52] [58] [53] [60] [61] [63] [64]), we found that VGG16, ResNet50 were mostly used. To decide which one to use in our case, we conducted two experiences in which we used VGG16 and ResNet50. In both experiences, we trained the model on our collected dataset, VGG16 gave us bad validation accuracy (around 20%) for 10 epochs of training. On the contrary, ResNet50 gave good results (more than 90%) for 30 epochs of training. As such, we decided to continue using Resnet50 for the medical image classification. In matter of fact, for ResNet50, we used transfer learning instead on training the model from scratch. We did so to obtain accurate results faster. First, we imported the model which was trained on ImageNet dataset from TensorFlow, then, we froze the model and did not include its top layer we added a flattening layer, then two dense layers one using ReLu activation function and the other uses softmax activation function to do the classification. Noting that we did the training in google Colab after uploading our dataset to google drive and linking it to it. The pictures dataset were collected from google image results for Measles and Chickenpox diseases from Roboflow [65] [66] [67] and from Kaggle [68]. After collecting the data, it was sorted and filtered to avoid having noisy dataset that would lead to wrong results, then, the data augmentation techniques such as: rotation and vertical flip were used to increase the size of the dataset, by the end of this phase we had 5000 images in the dataset with an average of 1000 images per class. Table. 3.1 resumes the dataset description per class. Table. 3.2 resumes the hyper parameters that we adjusted in our model

Classes		**scarlet fever**	**Kawasaki**	**measles,**	**chickenpox**	**roseola**
Original dataset	**Train**	24	32	216	196	24
	Valid	1	2	8	10	1
	Test	2	8	3	4	1
Augmented dataset	**Train**	969	956	986	969	963
	Valid	34	33	34	34	33
	Test	4	8	3	4	2

Table 3. 1. Images dataset

Image size	180*180
Batch size	64
Epochs	30 epochs
Learning rate	0.0001
Number of classes	5
Loss function	sparse categorical cross entropy
Optimizer	Adam

Table 3. 2. Our model hyper parameters

3.2.2 Training the text model

For the textual symptoms, we searched them on the internet and talked to doctors during our internship in order to understand them well. However, we found no ready to use public dataset. As such, we had to self-construct our dataset for each disease. For that, we started first by listing the diseases symptoms. Created multiple scenarios for each diseases, for example, if we consider the fever symptom, for Kawasaki disease varies from 39 to 40 C°, we created multiple inputs where we varied only the fever value (38.0, 38.1, 38.2, ..,41 C°) and fixed the others. After doing this processing for all the 5 diseases, we had constructed the initial dataset which was small in size (60 inputs). Subsequently, we augmented the textual constructed dataset by applying word swapping technique. Lastly, we checked the augmented dataset, cleaned it and passed it to the model. The final dataset had 4703 lines, each line contains a diseases symptom description. Table. 3.3 resumes the description of the textual dataset. To train the textual dataset, we used LSTM model because it excels in processing sequential data. Before passing the dataset to the model, it had to be preprocessed to eliminate the useless characters. Next, we changed some hyper parameters of the LSTM model and then we trained it. We resumed the hyper parameters in Table. 3.4 . Similarly to ResNet50 case, the training was conducted in Colab and the dataset was passed to it from google drive as CSV file.

Overally speaking, the trained LSTM gave high training and validation accuracy.

Classes	Chickenpox	Kawasaki	Measles	Roseola	Scarlet fever
Train	896	923	910	875	863
Valid	47	49	48	46	45
Test	10	10	10	10	10

Table 3. 3. Texts dataset

MAX_SEQUENCE_LENGTH	50
MAX_NB_WORDS	5000
Batch size	128
Epochs	100 epochs
Learning rate	0.001
Number of classes	5
Loss function	sparse categorical cross entropy
Optimizer	Adam
Dataset size	4703

Table 3. 4. Our model hyper parameters

We then compile the model as we did with image model with specifying the number of epochs, batch size, X_trian as the column of texts and Y_trian as the column of categories and gave 20 as percentage from train data to validation data. After, the model gave good results and height accuracy, then we had converted it as a h5 file, and save it in Drive. To use it in the fusion with image model to get our proposed multimodal.

3.3 Set up (environment)

For the backend part we used python programming language and for the development environment setup, we had two options : the first is to do the training locally within the computer, in which we need to install and set up Anaconda, then install the keras, tensorflow and numpy libraries. Lastly, download, train, evaluate and refine the models. The second option is to use google Colab, install the needed libraries to train, evaluate and refine the models. Table. 3.5. resumes the hardware characteristics of our computer and those offered by Colab. After comparing the hardware characteristics offered by Colab and those of my computer, I decided to carry on the training using Colab. My choice was motivated by multiple other reasons apart from the hardware characteristic part which out bested mine. Colab allowed an ease exchange and illustration of code with my supervisor, facilitated the backup task in case of any mishaps, and allowed me to work on other aspects of the projects such as writing the manuscript or working on the frontend while waiting for the training to end in Colab. Table. 3.6 recapitulate the backend environment setup.

Colab	**RAM**	13 GB
	GPU (core/frequency)	NVIDIA Tesla T4 GPU / 14.63 GB
	TPU (core/frequency)	8 cores with high-bandwidth memory
My computer	**RAM**	8 GB
	CPU (core/frequency)	4 cores /1.00GHz
	GPU (core/frequency)	UHD Graphics / 4GB

Table 3. 5. The hardware characteristics

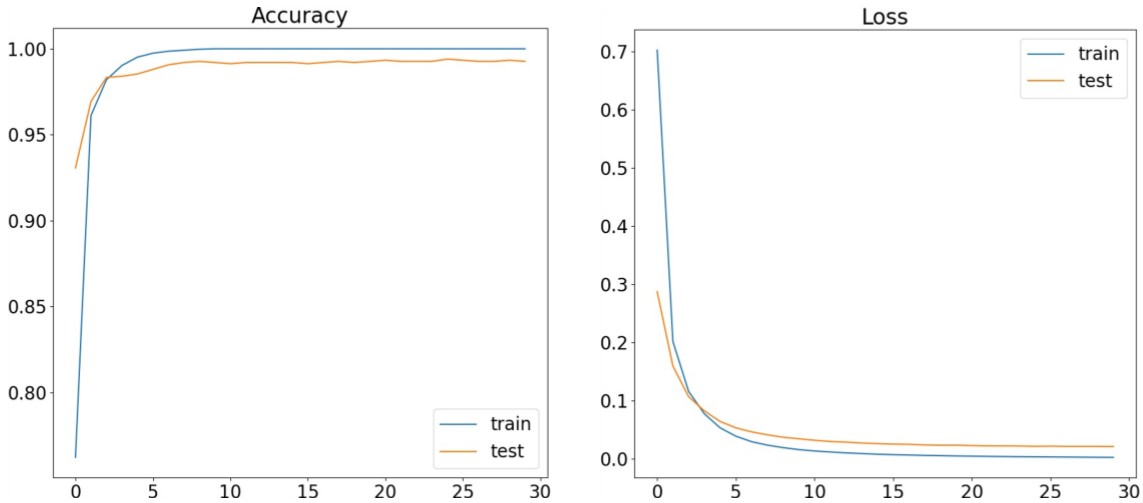
Component		Description			
Dataset	Images	Source	Google image, Roboflow [65] [66] [67] and Kaggle [68]		
		Size	5000	Training	3500
				Validation	1500
	Texts	Source	Google [3] [4] [5] [6] [7] [8] [9] [10] [11] [12].		
		Size	4703	Training	3292
				Validation	1411
Training Environment		Google Colab Python Language TensorFlow Libraries			
Models		Image classification		**ResNet50**	
		Text classification		**LSTM**	
		Multi model learning		**Late Fusion**	
Exported format		TFlite			

Table 3. 6. Backend Environment Setup

3.3.1 Models Results

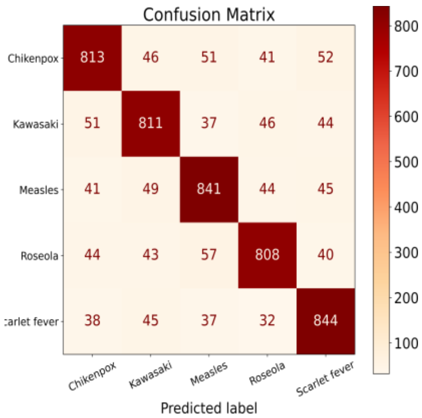
In this section, we illustrate and discuss the obtained results when training each of the three models, the ResNet50, the LSTM and the fused model.

- **ResNet50:** upon training ResNet50 for 30 epochs over 4 hours in Colab, we obtained a low validation loss value (0.0206) and high validation accuracy (99,27%). Further- more, upon checking the overall graphs of loss and accuracy for both training and validation, we noticed no signs of overfitting or underfitting. Moreover, we checked confusion matrix that illustrates the false positive and false negative cases, which although existed in our case, they were low in comparison to correctly predicted values. Figure. 3.5a, Figure. 3.5b, Figure. 3.6a, Figure. 3.6b represent: the accuracy graph, loss graph, the confusion matrix and the main classification metrics respectively.



(a) The accuracy graph. **(b)** The loss graph.

Figure 3. 5. Training and Validation history

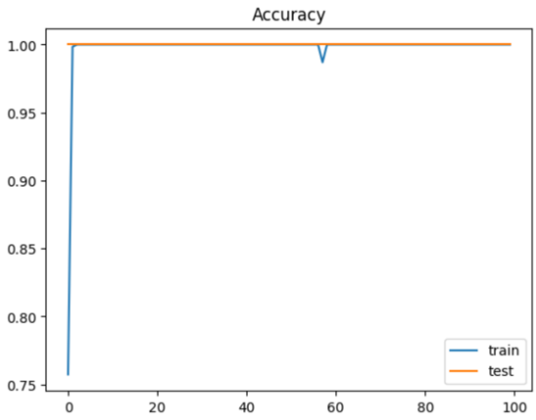


	precision	recall	f1-score	support
0	0.82	0.81	0.82	1003
1	0.82	0.82	0.82	989
2	0.82	0.82	0.82	1020
3	0.83	0.81	0.82	992
4	0.82	0.85	0.84	996
accuracy			0.82	5000
macro avg	0.82	0.82	0.82	5000
weighted avg	0.82	0.82	0.82	5000

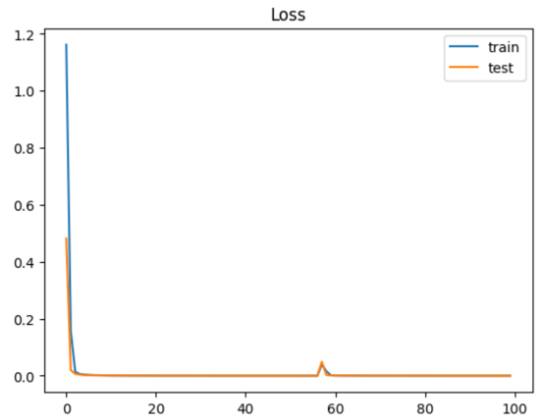
(a) Confusion matrix of images model. **(b)** The main classification metrics of images model.

Figure 3. 6. Images Model Training Results

- **LSTM :** upon training for 100 epochs for over 1 hour, the validation loss nullified and the validation accuracy reached 100%. Furthermore, upon checking the overall graphs of loss and accuracy for both training and validation, we noticed no signs of overfitting or underfitting. Moreover, we checked confusion matrix that illustrates the false positive and false negative cases, which although existed in our case, they were low in comparison to correctly predicted values. Figure. 3.7a, Figure. 3.7b, Figure. 3.8a, Figure. 3.8b represent the accuracy graph, loss graph, the confusion matrix and the main classification metrics respectively.

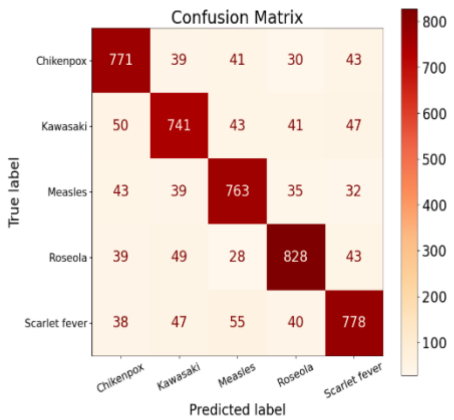


(a) The accuracy graph.



(b) The loss graph.

Figure 3. 7. Training and Validation history



(a) Confusion matrix of text model.

	precision	recall	f1-score	support
0	0.82	0.83	0.83	924
1	0.81	0.80	0.81	922
2	0.82	0.84	0.83	912
3	0.85	0.84	0.84	987
4	0.83	0.81	0.82	958
accuracy			0.83	4703
macro avg	0.82	0.83	0.82	4703
weighted avg	0.83	0.83	0.83	4703

(b) The main classification metrics of text model.

Figure 3. 8. Text Model Training Results

Let us now define the relative terminology of the main classification metrics :

- **The numbers** in the main classification metrics 0,1,2,3 and 4 refers to the diseases Chickenpox, Kawasaki, Measles, Roseola and Scarlet fever respectively and support refers to the number of the images or lines of each disease.
- **Accuracy :** defines, in general, the correctness of a model, it is calculated as illustrated in equation (1) [69] :

$$\frac{(TP + TN)}{\text{total number of predictions}} \qquad (1)$$

Where : **TP** stands for true positive, it is the number of cases in which the disease was classified correctly. **TN** stands for true negative, it is the number of cases in which the application did not classify the disease where it is in actual true (in actual it is not the disease).

- **Precision :** is a measure of how accurate a models positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model. It is calculated as illustrated in equation (2) [69]:

$$\frac{TP}{\mathrm{TP+FP}} \qquad (2)$$

Where, **FP** (False positives) The number of cases in which the application classified the disease where it is another one (mistake in classification).

- **The recall :** it measures the models ability to detect Positive samples. The higher the recall, the more positive samples detected. It is calculated as illustrated in equation (3) [69]:

$$\frac{TP}{\mathrm{TP+FN}} \qquad (3)$$

Where, **FN** stands for false negatives, it is the number of cases in which the application failed to classify the disease and classifies it as another one.

- **F1-score :** is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall. It is calculated as illustrated in equation (4) [69]:

$$\frac{2*(Precision*Recall)}{Precision+Recall} \qquad (4)$$

- **Specificity** : the number of samples predicted correctly to be in the negative class out of all the samples in the dataset that actually belong to the negative class. It is calculated as illustrated in equation (5) [69]:

$$\frac{TN}{FP+TN} \qquad (5)$$

3.4 AI multimodal training

In this section, we depict the multimodal training which was conducted by fusing the two previously explained trained unimodal.

3.4.1 The proposed multimodal method

Before deciding on the method to use for the multimodal training we skimmed through the literature, then we found the paper [51], in which its authors read over 128 articles published between 2011-2021 in scientific databases like: PubMed, Google Scholar, and IEEEXplore. They then concluded that the most common health areas utilizing multimodal methods were neurology and oncology. Notably, they found that there was an improvement in predictive performance when using data fusion. However, as we talked in chapter 2 about the fusion types, and as well as in our case we have two different model with two different types of inputs that is why we selected "Late Fusion".

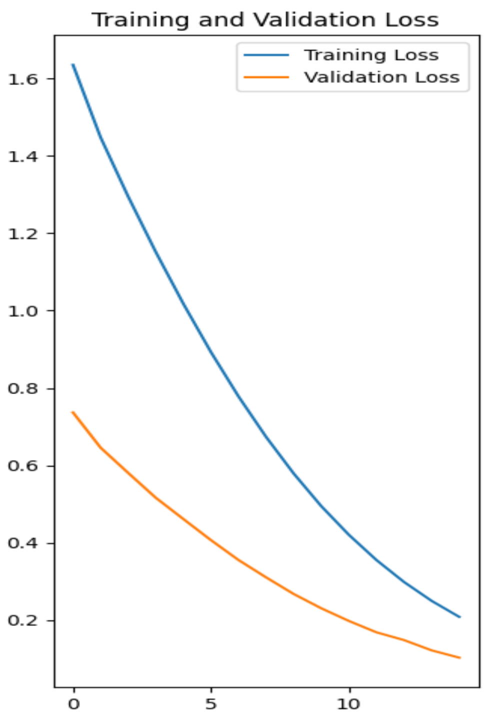
3.4.2 Fusion method application

In the fusing phase, we used the same datasets used to train the unimodels and we processed them and prepared them the same way. Then, we loaded our models, combined their features and added classification layer. Then we trained this model for 15 epochs, Table. 3.7 resumes the hyper parameters of the fused model.

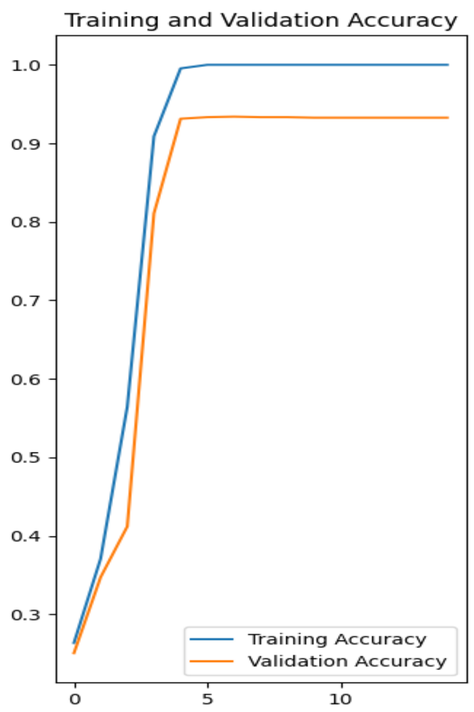
Image size	180*180
Text line size	50
Batch size	32
Epochs	15 epochs
Learning rate	0.001
Number of classes	5
Loss function	sparse categorical cross entropy
Optimizer	Adam

Table 3. 7. Our model hyper parameters

Our multimodal gave a good result, we obtained a low validation loss value (<1) and high validation accuracy (93%). Furthermore, upon checking the overall graphs of loss and accuracy for training, we did not notice signs of overfitting or underfitting. Moreover, we checked confusion matrix that illustrates the false positive and false negative cases, which although existed in our case, they were low in comparison to correctly predicted values. Figure. 3.9a, Figure. 3.9b, Figure. 3.10a, Figure. 3.10b represent: the accuracy graph, loss graph, the confusion matrix and the main classification metrics respectively:

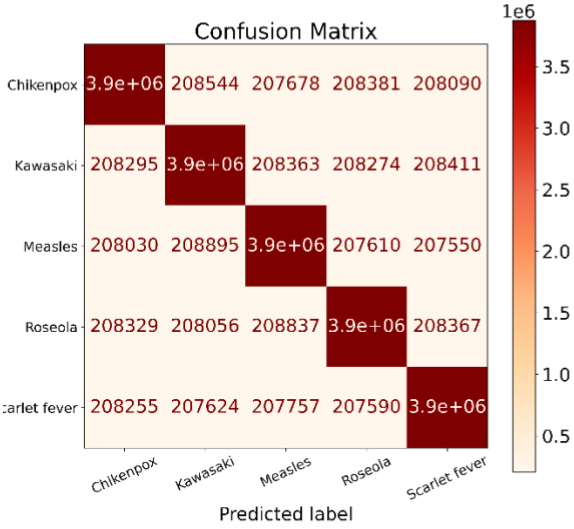


(a) Loss History.



(b) Accuracy History.

Figure 3. 9. Multimodal Training History



(a) Confusion matrix of Multimodal.

	precision	recall	f1-score	support
0	0.82	0.82	0.82	4701744
1	0.82	0.82	0.82	4706329
2	0.82	0.82	0.82	4701106
3	0.82	0.82	0.82	4702957
4	0.82	0.82	0.82	4702864
accuracy			0.82	23515000
macro avg	0.82	0.82	0.82	23515000
weighted avg	0.82	0.82	0.82	23515000

(b) The main classification metrics of Multimodal.

Figure 3. 10. Multimodal Training Results

3.5 Exporting the model

In this section, we explain how we used and exported our multimodal that we trained previously, as we clarified in the chapter 1, our application frontend is mobile app, that runs on Android systems. To be able to use the model from our mobile application which we explain its building process in the next chapter, we need to convert the model to TFLite (TensorFlow Lite) format. To convert our model to tflite file, we executed this instruction "tf.lite.TFLiteConverter.from_keras_model", after that we downloaded the model to our computer to be used in our Android studio project.

Chapter 4: Frontend development

4.1 Amelioration of Initial Design

In this section, we explain the frontend of our application and its design. As we discussed previously in chapter 1, that during my internship in the hospital, I had the chance to observe the medical software DEM interface, which I inspired from to make my application design. Also, I surveyed doctors' opinion of my first app design version as in the Figure. 2.10 (Chapter 1). They agreed with my suggestion to make the textual symptoms as checkboxes, writing the fever in text field and putting the rash types as a drop-down list. They also appreciated the background color that I selected for the application and found it comfortable for the eyes. Similarly, they were satisfied with the symptoms being arranged on pages, where the first page contained visual symptoms, followed by the page including the temperature measurement and pains. Trailed by the page including internal symptoms, and then the page that uses the camera to capture the visual symptoms. The final page is for displaying the diagnosed disease. Figure. 4.1 depicts the final design of our mobile application. Noting that the "check mark " in textual symptoms pages saves the results when moving between pages, "save data" button is to save all textual symptoms and move to the page that captures the visual symptoms using the camera. "See the results" button uses the fused model and displays the diagnosed disease.

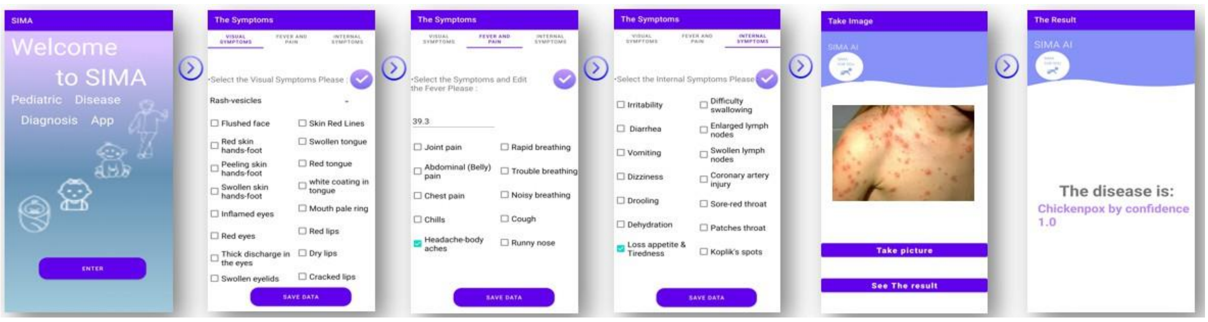


Figure 4. 1. Final design of our application.

4.2 Creating the Frontend

Creating the frontend as a mobile application was the doctor's preference as it would facilitate its usage to taking pictures of patient's visual symptoms. For this we chose to be our application as an android application as the majority of smartphone users use android system. Also, although the backend of our application was purely developed in python, the frontend is developed using java, we also used XML for designing the pages layouts. Firstly, for the main page displayed in Figure. 4.2. There is no login information, as currently we aimed to make an offline application that runs without internet. We opted for this choice considering that this application is particularly useful for hospitals in rural areas where the specialist pediatric doctors may not be available, our application may help a generalist doctor or trainee to do a correct diagnosis. The second reason behind our choice is that medical field is quite delicate with high security and privacy requirements, doctors would highly object using our application if it is to record their medical interventions per their accounts in our server, also, the patients or their legal gradians may not agree that their medical record and pictures are saved in third party server out of the hospital. Respecting these requirements, our current application keep no records neither about the patient nor the doctor, even the picture taken during the diagnosis are volatile and are deleted as soon as the diagnosis ends. As for the main view design, it is inspired from the topic, which includes the stages of a child's life as the application is for pediatric diseases that happens to children aged between 6 months to 15 years. Secondly, we put the symptoms in one activity on multiple tabs and allowed moving between them, if the doctor forget checking a symptom or mistaking checks irrelated symptom, see Figure. 4.3 for illustration.

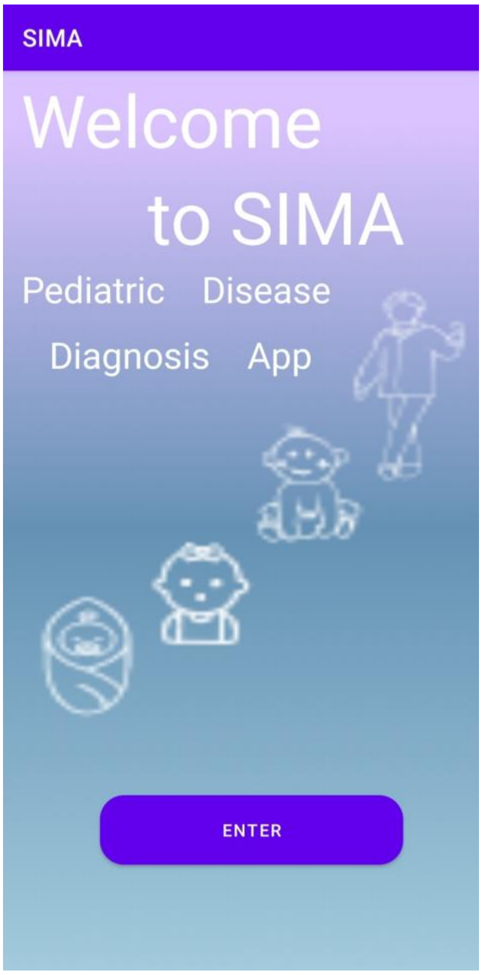


Figure 4. 2. Enter Page.

Figure 4. 3. Switching between symptoms tabs.

4.3 Using the Model

After finished the design of our application depicted in Figure. 4.1. We imported the tflite file model that we downloaded from Colab to our project in Android Studio. Since our model has two types of inputs, we had to process each type of data similarly to what we did in Colab as we detailed in the Chapter 3 before passing it to the fused model.

First, we saved the selected symptoms in the symptom's activity as string, then we took a picture in image activity and processing the captured image by putting it in bitmap then changing its width and height as (180,180). Second, we processed the text string by tokenizing it. Finally, when the two inputs are ready, we passed them as inputs of our multimodal, to get an output as one of the five diseases that we defined, with its confidence. The disease classification displayed is the one that the application is most sure of, see Figure. 4.4 for illustration of project flow diagram.

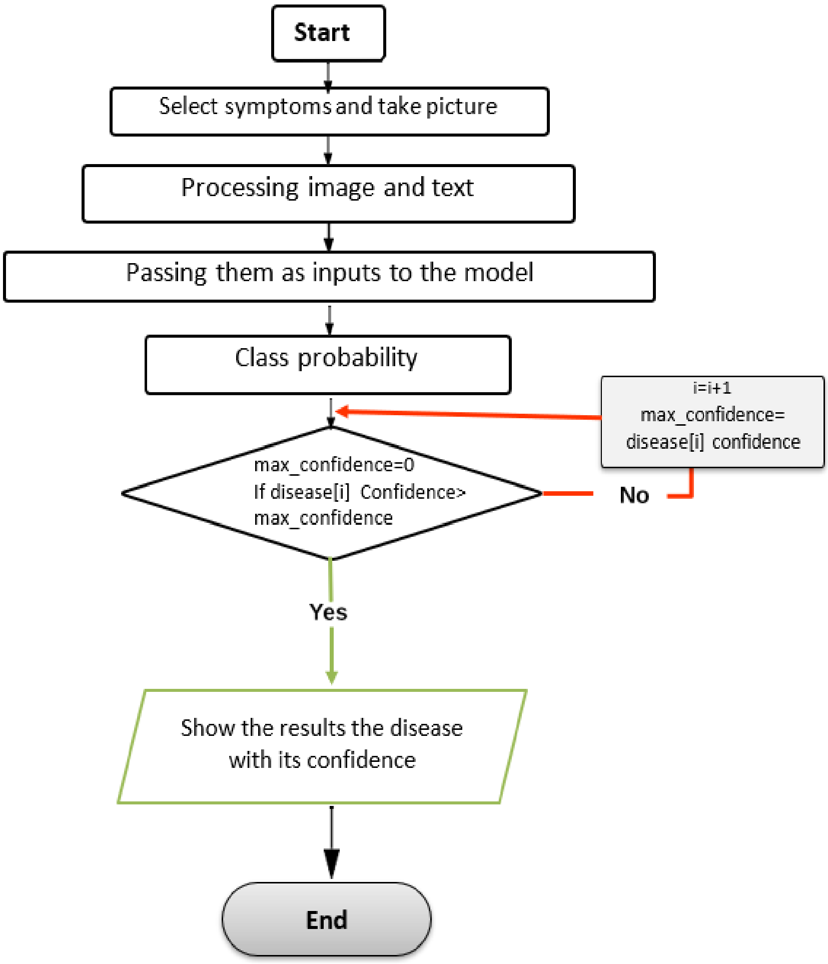


Figure 4. 4. Project flow diagram.

4.4 Evaluation in term of accuracy and speed

Although we evaluated the backend prior to importing it into the frontend. We re-evaluated the results obtained by the mobile application that used the converted TFlite to confirm that the accuracy is the same and to measure the speed of the application. We remind the reader of the results of our multimodal resumed in Chapter 3. The fused model gave 93% validation accuracy. Although high, the model still had some confusion when exposed to new pictures with diseases having similar visual symptoms such as red rushes. When we tried the model from the mobile application, we obtained similar results. We compared the inference time as well when using the model in Colab and from the mobile app. Table. 4.1 resumes the comparison results:

Table 4. 1. Colab Vs Android

Environment	Accuracy	Speed
Model in Colab	1 in training and 0.8 in testing	488ms
Model in Android	approximately the same as in Colab	Slower than Colab, around 2 second

4.5 Conclusion

In conclusion, we improved the initial design of our mobile application for pediatric diseases based on feedback from doctors. The frontend of the application we chose was an android application. We respected the privacy requirements of the medical field, by not saving patient's record data. Moreover, we put the symptoms as one activity on multiple tabs, allowing the doctor to move between them. After importing our model into the project in Android Studio, we processed its inputs similarly to what was done in Colab training. As a result, it shows the disease classification.

General Conclusion

In today's conditions, misdiagnosis cases and their fatalities are increasing, it is important to classify the disease correctly using images and symptoms, in the most ac- curate, automatic and fast way. Disease classification and diagnostic are important to prevent misdiagnosis and fatalities and so that the doctors can take necessary action. Children in particular cannot explain carefully how they feel or where is there pain. Thus, it is necessary to classify pediatric diseases carefully by taking pictures and knowing the symptoms for early diagnostic. Our application is an android mobile application that uses AI, to classify pediatric diseases from their textual and visual symptoms. Our application achieves that through the use of multimodal learning which is a complex learning combining multi uni-models trained each on specific type of dataset. In our case, it combines the ResNet50 model trained to classify diseases from their visual symptoms (images) and LSTM model trained on textual symptoms for the same. The aim and advantages of our application is :

- Helping the doctors to take decision for pediatric diagnostic more accurately, timely and efficiently.
- Easy to use, from smart phones.
- Runs offline making it practical for everyone.

Multimodal learning is challenging task, we resume the keys challenges that we had to solve to successfully accomplish our task :

- Non heterogeneous dataset (images and texts) which required different processing and different classification models using.
- Unavailability of medical datasets especially textual dataset.

For future work, we intend to explore the use of other neural network architectures for the classification in attempt to further improve the accuracy in classify medical images and text. We also intend to add different types of diseases to the dataset to enrich it further. Also, adding more types of symptoms such blood tests, MRI scans, X-Rays ... etc. Also, we intend to make our app available in multi-languages by adding language package. Furthermore, to make the application universal, we want to present the temperature in both Celsius degree and Fahrenheit degree.

The Auto Evaluation Grid

Task/objective	State	Details and remarks
☒: Achieved. ☐: Not achieved		
An overall study on pediatric misdiagnosis.	☒	Non-exhaustive listing
Reviewing some techniques in computer vision and image classification	☒	CNN-based architectures
Studying techniques used for natural language processing and textual classification	☒	LSTM
Understanding multimodal learning techniques and usage	☒	
Implementing transfer learning of ResNet50 for image classification	☒	
Implementing LSTM for text classification	☒	
Implementing the multimodal learning through late fusion	☒	
Models'performance evaluation and refining	☒	
Exporting the fused model to TFlite	☒	
Implementing mobile application using java in android studio	☒	
Using the fused model from the mobile application	☒	
Building and exporting the application to run on phones	☒	
Adding more pediatric diseases	☐	Not added in the current version due to time limit, and dataset unavailability, to be added in its amelioration.
Adding real-time inference from phone camera directly	☐	Not added in the current version due to time limit, to be added in its amelioration.
Adding language package to make the application available in multiple languages	☐	Not added in the current version due to time limit, to be added in its amelioration.
Inputting the temperature in Fahrenheit degree	☐	Not added in the current version due to time limit, to be added in its amelioration.
Adding more symptom types such as blood test results, X-rays, MRI scans, etc.	☐	Not added in the current version due to time limit, to be added in its amelioration.

Bibliography

[1] Errors of diagnosis in pediatric practice : A multi-site survey. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2921702/. 2010, Jun 21.

[2] R. Jakob. Disease classification. https://www.sciencedirect.com/topics/ immunology-and-microbiology/disease-classification. 2008.

[3] Kawasaki disease. https://www.cdc.gov/kawasaki/about.html. 2020, May 29.

[4] What is kawasaki disease? *American Heart Association*, Vol(Num) :1–2, 2023.

[5] Scarlet fever. https://www.mayoclinic.org/diseases-conditions/ scarlet-fever/symptoms-causes/syc-20377406. 2022, Jun 07.

[6] Diana Abu-Dayeh. Scarlet fever. *An-Najah National University*, Vol(Num) :1–2, 2014.

[7] Measles. https://www.mayoclinic.org/diseases-conditions/measles/ symptoms-causes/syc-20374857. 2022, May 11.

[8] Saudi MOH. Measles. , Vol(Num) :1–2, More than 10 years.

[9] Chickenpox. https://www.mayoclinic.org/diseases-conditions/chickenpox/ symptoms-causes/syc-20351282. 2023, April 19.

[10] Adriana Lopez, MHS, Theresa Harrington, MD,MPHTM, and Mona Marin, MD. Varicella. *Centers for Disease Control and Prevention*, Vol(Num) :1–2, –.

[11] Roseola. https://kidshealth.org/en/parents/roseola.html#:~:text=What%20Are%20the%20Signs%20%26%20Symptoms,(glands)%20in%20the%20neck. May 2023.

[12] St Georges Hospital. Roseola. *St George's University Hospitals NHS Foundation Trust*, Vol(Num) :1–2, April 2022.

[13] Seeing the future : An introduction to computer vision in ai. https://neurosys.com/blog/introduction-to-computer-vision-in-ai#:~:text=Computer%20vision%20is%20a%20whole,from%20the%20world%20around%20them. 2024,January 10.

[14] RANJAY KRISHNA. Computer vision : Found at ions and applications. *STANFORD UNIVERSITY*, Vol(Num) :17, December 2017.

[15] Rosalie Waelen. The power of computer vision. a critical analysis. *Simon Stevin Series in the Ethics of Technology*, Vol(Num) :15–16, 2023.

[16] Yasunari Matsuzaka and Ryu Yashiro. Ai-based computer vision techniques and expert systems. *AI MDIP*, Vol(Num) :1, 04,2023.

[17] EMB and Team. Expand my business https://blog.emb.global/ computer-vision-and-machine-learning/. 2024, March 27.

[18] Ambika. What is computer vision ? (history, applications, challenges). https://medium.com/@ambika199820/ what-is-computer-vision-history-applications-challenges-13f5759b48a5. 2023, Sep 18.

[19] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks : an overview and application in radiology. *SpringerOpen*, Vol(Num) :1–2, April 2018.

[20] Gaudenz Boesch. Deep neural network : The 3 popular types (mlp, cnn and rnn). https://viso.ai/deep-learning/ deep-neural-network-three-popular-types/. 2024, January 10.

[21] Anirudha Ghosh, Farhana Sultana, A. Sufian, and Amlan Chakrabarti. Fundamental concepts of convolutional neural network. *ResearchGate*, Vol(Num) :26–35, January 2020.

[22] Aaryan Gupta. Evolution of convolutional neural net- work architectures. https://medium.com/the-pen-point/ evolution-of-convolutional-neural-network-architectures-6b90d067e403. 2020 , May 17.

[23] Abhijeet Pujara. Concept of alexnet :- convolutional neu- ral network. https://medium.com/analytics-vidhya/ concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30. 2020, Nov 3.

[24] NIKHIL TOMAR. Vgg16 unet implementation in tensorflow. https:// idiotdeveloper.com/vgg16-unet-implementation-in-tensorflow/. 2023 , FE- BRUARY 6.

[25] Zhiling Guo, Qi Chen, Guangming Wu, Yongwei Xu, Ryosuke Shibasaki, and Xiaowei Shao . Village building identication based on ensembleconvolutional neural networks. *sensors*, Vol(17) :10–13, 2017.

[26] Pablo Ruiz. Understanding and visualizing resnets. https://towardsdatascience. com/understanding-and-visualizing-resnets-442284831be8. 2018 , Oct 8.

[27] Lev Craig. What is machine learning and how does it work ? *SpringerOpen*, Vol(Num), Aug 2023.

[28] Pablo Ruiz. Introduction to recurrent neural network. https://www. geeksforgeeks.org/introduction-to-recurrent-neural-network/. 2023, Dec 04.

[29] Michael I. Jordan. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, Vol(Num) :117, 2002.

[30] Introduction to recurrent neural network. https://www.geeksforgeeks.org/ introduction-to-recurrent-neural-network/. 2023, Dec 04.

[31] Greg Van Houdt, Carlos Mosquera, and Gonzalo Napoles. A review on the long short-term memory model. *Springer Nature*, Vol(53) :59295955, 2020.

[32] Siddhesh Bangar. Alexnet architecture explained. https://medium.com/ @siddheshb008/alexnet-architecture-explained-b6240c528bd5. 2022, Jun 24.

[33] Difference between alexnet and googlenet. https://www.geeksforgeeks.org/ difference-between-alexnet-and-googlenet/. 2023, Jul 31.

[34] Siddhesh Bangar. Vgg-net architecture explained. https://medium.com/ @siddheshb008/vgg-net-architecture-explained-71179310050f. 2022, Jun 28.

[35] Understanding googlenet model cnn architecture. https://www.geeksforgeeks. org/understanding-googlenet-model-cnn-architecture/. 2021, Nov 18.

[36] Dewansh Singh. Dloa (part-16)-googlenet cnn. https://medium.com/ai-genesis/ dlao-part-16-googlenet-cnn-bc9bc941c510#:~:text=Key%20Features%20of% 20GoogLeNet%20CNN&text=GoogleNet%20model%20has%20nine%20identical,

inception%20units%20or%20inception%20modules.&text=There%20are%20Convolutional%20Kernels%20of,end%20of%20the%20inception%20unit. 2023, May 16.

[37] Siddhesh Bangar. Resnet architecture explained. https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d. 2022, Jul 5.

[38] Anishnama. Understanding lstm : Architecture, pros and cons, and implementation. https://medium.com/@anishnama20/understanding-lstm-architecture-pros-and-cons-and-implementation-3e0cca194094. 2023, Apr 28.

[39] Risto Miikkulainen and Babak Hodjat. Intelligent nanotechnology. *SienceDirect*, Vol(Num) :307–329, 2023.

[40] Gaudenz Boesch. Top 19 applications of deep learning and computer vision in health-care. https://viso.ai/applications/computer-vision-in-healthcare/. 2024.

[41] Alotaibi Mohammed, Aljouie Abdulrhman, Alluhaidan Najd, Qureshi Wasem, Almatar Hessa, Alduhayan Reema, Alsomaie Barrak , and Almazroa Ahmed. Breast cancer classification based on convolutional neural network. *Heliyon*, Vol(Num) :2– 13, 2023.

[42] Zeeshan Hameed B. M, Patil, Vathsala, Shetty, Dasharathraj, Naik, Nithesh, Nagaraj, Nikhil, and Sharma, Disha. Use of artificial intelligence-based computer vision system to practice social distancing in hospitals to prevent transmission of covid-19. *Indian Journal of Community Medicine*, Vol(3) :379–380, 2020.

[43] Amit Singh, Albert Haque, Alexandre Alahi, Serena Yeung, Michelle Guo, Jill R Glassman, William Beninati Terry Platchek, Li Fei-Fei, and Arnold Milstein. Automatic detection of hand hygiene using computer vision technology. *Naional Library of Medicine*, Vol(8) :13161320, 2020.

[44] François Chadebecq, Francisco Vasconcelos, Evangelos Mazomenos, and Danail Stoyanov. Computer vision in the surgical operating room. *Naional Library of Medicine*, Vol(36) :456462, 2020.

[45] Yiwen Gu, Shreya Pandit, Elham Saraee, Timothy Nordahl, Terry Ellis, and Margrit Betke. Home-based physical therapy with an interactive computer vision system. *IEEE*, Vol(Num) :1–3, 2019.

[46] Jagadish K. Mahendran, Daniel T. Barry, Anita K. Nivedha, and Suchendra M. Bhandarkar. Computer vision-based assistance system for the visually impaired using mobile edge artificial intelligence. *IEEE*, Vol(Num) :1–3, 2021.

[47] Pei Xiangdong, Zuo Ke , Li Yuan , and Pang Zhengbin. A review of the application of multimodal deep learning in medicine : Bibliometrics and future directions. *International Journal of Computational Intelligence Systems*, Vol(Num) :1–5, 2023.

[48] Konstantinos Poulinakis. Multimodal deep learning : Definition, examples, applications. https://www.v7labs.com/blog/multimodal-deep-learning-guide. 2022, December 15.

[49] George Lawton. multimodal ai. https://www.techtarget.com/searchenterpriseai/definition/multimodal-AI. May 2023.

[50] Tadas Baltrusaitisand Chaitanya Ahuja and Louis-Philippe Morency. Multimodal machine learning : A survey and taxonomy. *TRANSACTIONS OF PATTERN ANA- LYSIS AND MACHINE INTELLIGENCE*, Vol(Num) :1–2, 2018.

[51] Adrienne Kline, Hanyin Wang, Yikuan Li, Saya Dennis, Meghan Hutch, Zhenxing Xu, Fei Wang, Feixiong Cheng, and Yuan Luo. Multimodal machine learning in precision health : A scoping review. *digital medicine*, Vol(171) :1–3, 2022.

[52] A.L. Goldberger, et al. Stanley, PhysioBank, Phys ioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000 (June 13)) e215–e220, http: //dx.doi.org/10.1161/01.CIR.101.23.e215, Circulation Electronic Pages: http: //circ.ahajournals.org/content/101/23/e215.full PMID:1085218.

[53] C.R. Jack, Jr., M.A. Bernstein, N.C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P.J. Britson, J. L. Whitwell, C. Ward, et al., The alzheimer's disease neuroimaging initiative (ADNI): MRI methods, J. Magn. Reson. Imaging: Official J. Int. Soc. Magn. Reson. Med. 27 (4) (2008) 685–691.

[54] D. Demner-Fushman, S. Antani, M. Simpson, G.R. Thoma, Design and develop ment of a multimodal biomedical information retrieval system, J. Comput. Sci. Eng. 6 (2) (2012) 168–177.

[55] A. P. Jamesaand B. V. Dasarathyb. Medical image fusion : A survey of the state of the art. *Information Fusion Consultant*, Vol(Num) :10–11, 2014.

[56] N.E. Allen, C. Sudlow, T. Peakman, R. Collins, U. biobank, UK biobank data: come and get it, Sci. Transl. Med. 6 (224) (2014) 224ed4.

[57] Parvathy Sudhir Pillai and Tze-Yun Leong. Fusing heterogeneous data for alzheimer's disease classification. *Naional Library of Medicine*, Vol(Num) :1–5, 2015.

[58] K. Tomczak, P. Czerwińska, M. Wiznerowicz, Review the cancer genome atlas (TCGA): An immeasurable source of knowledge, Contemp. Oncol./Współczesna Onkologia 2015 (1) (2015) 68–77.

[59] A. García Seco de Herrera, R. Schaer, S. Bromuri, H. Müller, Overview of the ImageCLEF 2016 medical task, in: Working Notes of CLEF 2016 (Cross Language Evaluation Forum), 2016.

[60] Juan Camilo Vasquez-Correa, Tomas Arias-Vergara, J R Orozco-Arroyave, Bjorn Eskofie, Jochen Klucken, and Elmar Noth. Multimodal assessment of parkinson's disease : A deep learning approach. *Naional Library of Medicine*, Vol(23) :1–4, 2019.

[61] Ilan Shomorony et al. An unsupervised learning approach to identify novel signatures of health and disease from multimodal data. *Naional Library of Medicine*, Vol(7) :10–12, 2020.

[62] Pei Xiangdong, Zuo Ke , Li Yuan , and Pang Zhengbin. Pathogenic factors identification of brain imaging and gene in late mild cognitive impairment. *Naional Library of Medicine*, Vol(13) :1–5, 2021.

[63] Yixue Hao , Mohd Usama , Jun Yang , M. Shamim Hossain, and Ahmed Ghoneim. Recurrent convolutional neural network based multimodal disease risk prediction.*SienceDirect*, Vol(92) :76–83, 2019.

[64] Buket BAYRAM, Bahadr KULAVUZ, Berkay ERTURUL, Bulent BAYRAM, Tolga BAKIRMAN, Tuna ÇAKAR, and Metehan DOAN. Classification of skin lesion images with deep learning approaches. *ResearchGate*, Vol(10) :244–290, 2022.

[65] Zhixiang Zhao, Che-Ming Wu , Shuping Zhang, Fanping He , Fangfen Liu , Ben Wang, Yingxue Huang , Wei Shi, Dan Jian, Hongfu Xie, Chao-Yuan Yeh, and Ji Li. A novel

convolutional neural network for the diagnosis and classification of rosacea : Usability study. *Naional Library of Medicine*, Vol(9), 2021.

[66] Arslan Akram, Javed Rashid, Muhammad Arfan Jaffar, Muhammad Faheem, and Riaz ul Amin. Segmentation and classification of skin lesions using hybrid deep learning method in the internet of medical things. *Naional Library of Medicine*, Vol(29), 2023.

[67] Kiran Likhar and Sonali Ridhorkar. Enhancing skin cancer detection : A comparative analysis of models with vgg-16, vgg-19, and inception v3. *IJISAE*, Vol(12), 2024.

[68] Adam M. Ibrahim, Mohammed Elbasheir, Somia Badawi, Ashraf Mohammed, and Amir F. Mohammed Alalmin. Skin cancer classification using transfer learning by vgg16 architecture (case study on kaggle dataset). *Scientific Research*, Vol(15), 2023.

[69] Yashwant Ingle and Nuzhat Faiz Shaikh. Deep learning for skin cancer classification : A comparative study of cnn and vgg16 on ham10000 dataset. *SSRN*, Vol(Num), 2024.

[70] Alavikunhu Panthakkan, S.M. Anzar , Sangeetha Jamal, and Wathiq Mansoor a . Computers in biology and medicine. *SienceDirect*, Vol(150), 2022.

[71] Digital Twin. Measles dataset. https://universe.roboflow.com/digital-twin/ measles, sep 2022. visited on 2024-05-20.

[72] gizem. measles dataset. https://universe.roboflow.com/gizem-3hati/ measles-5vcbl, dec 2023. visited on 2024-05-20.

[73] data. chickenpox dataset. https://universe.roboflow.com/data-od5oq/ chickenpox-1ixkx, apr 2023. visited on 2024-05-20.

[74] SACHIN KUMAR. Monkeypox images dataset (monkey-pox+chickenpox+measles+normal preprocessed skin images dataset). https://www.kaggle.com/datasets/sachinkumar413/monkeypox-images-dataset, 2022.

[75] Confusion matrix in machine learning. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/. 2024, May 13.