Democratic and Popular Algerian Republic

Ministry of the Superior Teaching and the Scientific Research

Mohamed Khider University – Biskra

Faculty of Exact Sciences, Natural Sciences and Life

Department of Computer Science

Dissertation

Presented to obtain the academic master's degree in Computer Science

Option: SIOD

Healthcare using deep learning

Diabetic retinopathy detection using deep learning

By :

BERRA ERRAID FERHAT ABASS

Defended on Sept 2020, in front of the jury composed of:

Dr.		President	University M.K. Biskra
Dr.	**Dr.Meadi nadjib**	Advisor	University M.K. Biskra
Dr.		Examiner	University M.K. Biskra

2020

Acknowledgements

I would like to thank every one believed in me, helped me and pushed me forward to successed, i'm really greatfull to all of you. Special thanks to my supervisor Dr.Meadi nadjib, i will never forget your favors, you support me always with your advices and guidings, you taught me the fun of deep learning and you made me love it, because of you i'm presenting this prideful work.

Mom and dad may Allah bless you ! you might have no idea what i'm trying to do, what i'm studying and where i'm spending my time in, but you have been always proud and happy for me.

Dear brothers thanks for your support and encouragments me specially yassin, zakaria, larbi, ramzi chenafi, hdagha mouad, islam alouane, hiwani mehdi, dhia chettouh, dhia bouhamed, wail belarahmi, khebouri monjid and khelil hatem, you are the best of all and the best thing that ever happened to me.

Github, stackoverflow, Kaggle, Geoffery Hinton, Colab and every one contributed to make this work see the light, Thank you all !

BERRA ERRAID FERHAT ABASS

ABSTRACT

Diabetic retinopathy (DR) is a complication of diabetes that is caused by changes in the blood vessels of the retina. The symptoms can blur or distort the patient's vision and are a main cause of blindness.

Many of the complications of DR can be prevented by blood glucose control and timely treatment. Since the varieties and the complexities of DR, it is really difficult for DR detection in the time-consuming manual diagnosis.

This work is to attempt towards finding an automatic way to classify a given set of fundus images. We bring Alexnet a convolutional neural networks (CNNs) to DR detection, which includes 2 major difficult challenges: detection and classification .

Coupled with transfer learning and hyper-parameter tuning, we adopt Resnet, Efficientnet and analyze how well these models do with the DR image classification .

We employ publicly available Google platform called Colab for training these models.

This work proposes to applie the CNN architectures on 3 diabetec retinopathy datasets with diffrents class organization.

Key words: Healthcare, Health informatics, Machine learning, Deep leaning, Convolution neural networks, Transfer learning, Image processing, Diabetec retinopathy.

Résumé

La rétinopathie diabétique (RD) est une complication du diabète causée par des modifications des vaisseaux sanguins de la rétine. Les symptômes peuvent brouiller ou déformer la vision du patient et sont une des principales causes de cécité.

Bon nombre des complications de la RD peuvent être évitées par un contrôle glycémique et un traitement rapide. Étant donné les variétés et la complexité de la RD, il est vraiment difficile de détecter la RD dans le diagnostic manuel fastidieux.

Ce travail vise à tenter de trouver un moyen automatique de classer un ensemble des données d'images de fond d'œil. Nous apportons Alexnet qui est le réseaux des neurones convolutifs (CNN) pour la détection de RD, qui comprend 2 défis majeurs: la détection et la classification .

Couplé à l'apprentissage par transfert et le réglage des hyper-paramètres, nous adoptons Resnet et Efficientnet, L'analyse de l'efficacité de ces modèles avec la classification d'images DR.

Nous utilisons la plate-forme de Google qui est accessible au public. Ainsi qu'elle est appelée Colab pour l'entraînement des modèles.

Ce travail propose d'appliquer les architectures CNN sur 3 bases des données de RD avec des différentes organisations des classes.

Mots clés: soins de santé,Informatique de santé, Apprentissage automatique, Apprentissage en profondeur, Réseaux de neurones à convolution, Apprentissage par transfert, Traitement d'image, Rétinopathie diabétique.

ملخص

اعتلال الشبكية السكري هو أحد مضاعفات مرض السكري الذي ينتج عن تغيرات في الأوعية الدموية في شبكية العين. يمكن للأعراض أن تشوش أو تشوه رؤية المريض وهي سبب رئيسي للعمى.

يمكن منع العديد من مضاعفات اعتلال الشبكية السكري عن طريق التحكم في نسبة الجلوكوز في الدم والعلاج في الوقت المناسب. نظرًا لتنوع وتعقيدات اعتلال الشبكية السكري، فمن الصعب حقًا اكتشاف اعتلال الشبكية السكري في التشخيص اليدوي الذي يستغرق وقتًا طويلاً.

يهدف هذا العمل إلى محاولة إيجاد طريقة تلقائية لتصنيف قاعدة البيانات لصور قاع العين. نقدم Alexnet الشبكة العصبية الالتفافية لاكتشاف اعتلال الشبكية السكري، والذي يتضمن تحديين رئيسيين صعبين: الكشف والتصنيف.

بالاقتران مع نقل التعلم وضبط المعلمات الفائقة، نعتمد على Resnet و Efficientnet و نحلل مدى جودة أداء هذه النماذج مع تصنيف صور اعتلال الشبكية السكري.

نحن نستخدم منصة Google المتاحة للجمهور والتي تسمى Colab لتدريب هذه النماذج.

يقترح هذا العمل تطبيق بنى الشبكة العصبية الالتفافية على 3 قواعد بيانات خاصة باعتلال الشبكية السكري مع تنظيم مختلف للفئات.

الكلمات المفتاحية: الرعاية الصحية، المعلوماتية الصحية، التعلم الآلي، التعلم العميق، الشبكات العصبية الالتفافية، نقل التعلم، معالجة الصور، اعتلال الشبكية السكري.

Summary

List of Figures

List of Tables

General introduction

Health informatics is the logic of healthcare. It is the study of how clinical knowledge is created, shaped, shared and applied. It is the rational study of the way we think about healthcare, and the way that treatments are defined, selected and evolved. With such a pivotal role, the study of informatics is as fundamental to the practice of medicine and the delivery of healthcare in this century as anatomy or pathology was in the last. Health informatics is thus as much about computers as healthcare . Rather than drugs, X-ray machines or surgical instruments, the tools of informatics are more likely to be clinical guidelines, decision support systems, formal health languages, electronic records, or communication systems like social media. These tools, however, are only a means to an end, which is the delivery of the best possible healthcare. Health informatics is particularly focused on: diagnostics, therapy (medical intervention with as little strain on the patient as possible), therapy simulation, early-recognition and prevention, compensating physical handicaps, health consulting, health reporting, healthcare information systems, medical documentation and comprehensive documentation of medical knowledge and knowledge-based decision support.

The need of data analytics in health informatics for better decision making is a challenging domain for the past decade. This stimulates more interest of researchers for the design of data driven models on the basis of machine learning such as deep learning models for health informatics. Deep learning is an emerging machine learning technique with various applications for healthcare monitoring such as medical imaging, bioinformatics, pervasive sensing and public healthcare, etc. We choose in this work to focus on one of the famous medical imaging problem which is diabetic retinopathy detection using deep learning .

Problematic and motivation

Vision and eye health are one the most crucial organs in human body, it needs to be preserved to maintain life of the individuals. Eye diseases such as diabetic retinopathy is mainly caused due to the the damages of the retina, and since it is damaged, and diagnosed at late stages, there is almost no chance to reverse vision and cure it, which means that the patient will loose the vision ability partially, and may be entirely.

Diagnosing or predicting these pathologies at early stages can increase the chance of curing the patients and restore vision ability.

Since deep learning is nowadays used in different fields of medical imaging, compute raided diagnosis, image segmentation, image-guided therapy. This means that there are multiple areas in medicine, where deep learning methods can be applied and can help improve patient's healthcare.

Aim of the work

The aim of this work is to propose a classification models based on deep learning to automatically classify and identify the different diabetic retinopathy stages using retinal images , and to see deep learning performance.
The dissertation gives as well comparisons of different proposed approaches with previous works. In summary, the following main objectives can be identified:

- Evaluate different deep learning classification methods.
- Propose deep convolutional neural network architectures .
- Evaluate deep architectures.
- Comparisons the proposed approaches.
- Compare the obtained results.
- Comparisons with previous works.
- Conclusion on the use of deep learning in healthcare.

Dissertation structure

The Dissertation is organized as follows:

- **Chapter 1 Health informatics :** This chapter discusses healthcare and health informatics domains and technologies that helps in healthcare .
- **Chapter 2 Deep learning :** This chapter discusses the state of the art of machine learning and deep learning techniques beside our case study and the related works .
- **Chapter 3 System design :** This chapter describes the datasets and the system with all the phases .

- **Chapter 4 Implementation and results :** This chapter mentions the implementation tools, describes the code and discussions the result obtained .
- **Conclusion and future work :** Finally, this chapter summarizes the major findings and gives insights into future work.

Chapter 1

Health informatics

1.1 Introduction

Computer-based information and communication technologies continue to transform the delivery of health care and the conception and scientific understanding of the human body and the diseases that afflict it.
In fact, modern medicine has become almost inconceivable without the use of computers. While information technology has the potential to improve the quality and efficiency of patient care, it also raises important ethical and social issues. Medical informatics has evolved as rapidly as any science in history, paralleling and relying on extraordinary advances in information collection, storage, analysis and transmission.
Indeed, like sciences and technologies that precede it; biochemistry, microbiology, genetics and cell biology, pharmacology and so on; medical informatics is changing the standard of care.
We are too soon able to do things before we know whether we ought to or not. This is not a problem. It is rather a stage setting in which we have the opportunity to use the tools of applied and theoretical to track scientific change and eventually to guide it [1].
In addition, one of the rapidly growing areas of health care innovation lies in the advanced use of data science and machine learning .
The key opportunities here are:

- Health monitoring and diagnosis.
- Medical treatment and patient care.
- Pharmaceutical research and development.
- Clinic performance optimization [2].

The intersection of medicine and computing has so far generated an ensemble of interesting and important questions:

- What is exactly health informatics?

- What is the domains of health informatics ?

- What is the importance of health informatics?

- How does information technology change medical science ?

- How does (and will) medical decision-making change [1]?

We will find answers to those questions in this chapter by defining healthcare, health informatics, some domains of health informatics and finally the history of the intersection of computer science and healthcare .

1.2 Healthcare

After an intensive search for the meaning of healthcare, i found many meanings and definitions. Here are some basic definitions from different sources.

1.2.1 Definitions

In dictionary of Cambridge, healthcare is the set of services provided by a country or an organization for the treatment of the physically and the mentally ill [3].

In market business news, healthcare refers to the efforts that medical professionals make to restore our physical and mental well-being. The term also includes the provision of services to maintain emotional well-being. We call people and organizations that provide these services healthcare providers. We call the people either healthcare workers, healthcare providers, healthcare professionals or health professionals [4].

The definition of JIT (just in time) developed by nurses is " Giving the Right patient, the Right care, the Right way, while providing a Great Patient Experience ".

It includes providing the staff with the necessary and proper tools and supplies in the right locations, when they need them, and in the right amount with shortest nurse travel distances within efficient layouts. The goal is to create an efficient system with the same or better quality and safety than exists today. It does not do any good to create the most efficient or quickest system if we do not have the same or better patient and staff safety, ergonomics and quality [5].

1.2.2 Stages of healthcare

Every patient has required different care depending upon their health problem like some require normal care and some require extra special care.

So on the basis of patient condition healthcare divides into various stages. Following types of healthcare are explained below:

1.2.2.1 Primary healthcare

Primary Healthcare is essential healthcare made universally accessible to individuals and families in the community.

Primary healthcare addresses the main health problems in the community, providing promotive, preventive, curative and rehabilitative services accordingly. Since these services reflect and evolve from the economic conditions and social values of the country and its communities, they will vary by country and community, but will include at least:

promotion of proper nutrition and an adequate supply of safe water, basic sanitation, maternal and child care, including family planning, immunization against the major infectious diseases, prevention and control of locally endemic diseases, education concerning prevailing health problems and the methods of preventing and controlling them and appropriate treatment for common diseases [6].

1.2.2.2 Secondary healthcare

When your primary care provider refers you to a specialist, you are then in secondary care. Secondary care simply means you will be taken care of by someone who has more specific expertise in what is ailing you.

Specialists focus either on a specific system of the body or a specific disease or condition. For example, cardiologists focus on the heart and its pumping system. Endocrinologists focus on hormone systems and some specialize in diseases like diabetes or thyroid disease.

Secondary care is where most people end up when they have a medical condition to deal with that can’t be handled at the primary care level [7].

1.2.2.3 Tertiary healthcare

Specialized care that offers a service to those referred from secondary care for diagnosis or treatment, and which is not available in primary or secondary care. Tertiary care has become a common feature in certain specialties for rare conditions, where the diagnostic, treatment facilities are scarce, require scarce combinations of resources or which remain essentially the subject of research.

These facilities are commonly found in medical schools and teaching hospitals [8].

1.3 Health informatics

One of the popular degree programs within the field of Healthcare Management is the sub-field Health Informatics. **what is health informatics?**

1.3.1 Definitions

In medical publishing, Health Informatics is a term describing the acquiring, storing, retrieving and using of healthcare information to foster better collaboration among a patient’s various healthcare providers. Health Informatics plays a critical role in the push toward healthcare reform.

Health informatics is an evolving specialization that links information technology, communications and healthcare to improve the quality and safety of patient care.

It is a multidisciplinary field that uses health information technology (HIT) to improve health care via any combination of higher quality, higher efficiency and new opportunities. It is also called as health care informatics, healthcare informatics, medical informatics,

nursing informatics, clinical informatics, or biomedical informatics [9].
In healthcare management ,health informatics (also called Health Information Systems) uses information technology to organize and analyze health records to improve healthcare outcomes.
Health Informatics deals with the resources, devices and methods to utilize acquisition, storage, retrieval and use of information in health and medicine [10].
Tools include medical terminology, information and communication systems and computer technology. Healthcare informatics provides electronic access to medical records for patients, doctors, nurses, hospital administrators, insurance companies and heath information technicians. The Health Informatics field is rapidly growing and there are many online and on-campus certificate and degree programs in health informatics and health information management at both the undergraduate and graduate levels [11].

1.3.2 Domains of health informatics

The domain of health informatics is not well defined. It covers a wide range of research topics. Our objective is to give an overview about some famous domains in health informatics.

1.3.2.1 Bioinformatics

Bioinformatics is an interdisciplinary field that develops methods and software tools for understanding biological data. As an interdisciplinary field of science, bioinformatics combines biology, computer science, information engineering, mathematics and statistics to analyze and interpret biological data [12] .

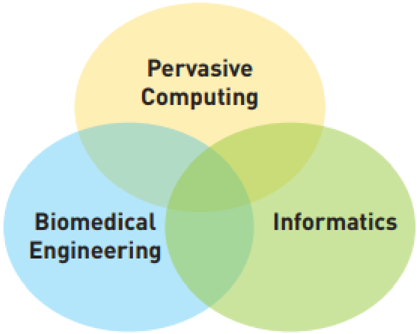


Figure 1.1: Three major fields of Pervasive Biomedical Informatics [12].

1.3.2.2 E-Health

The use of modern information and communication technologies to meet needs of citizens, patients, healthcare professionals, healthcare providers and policy makers [13].

In addition, E-Health is the cost-effective and secure use of information and communications technologies in support of health and health-related fields [14] .

- including health-care services.

- health surveillance.

- health literature.

- health education, knowledge and research.

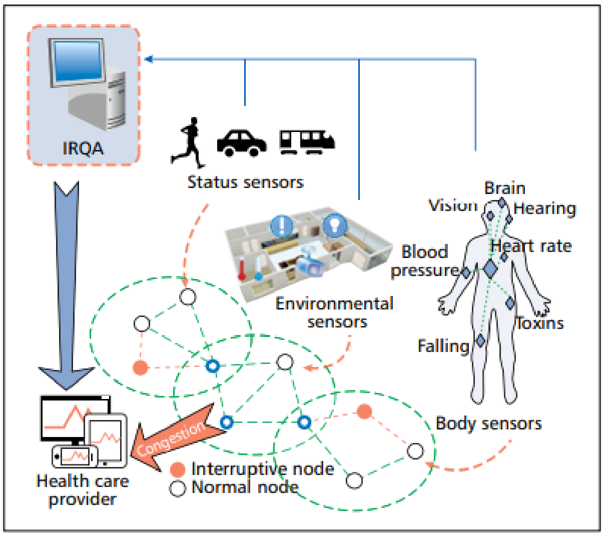


Figure 1.2: Example of E-health application:Problem statement of mobile big data network [15].

1.3.2.3 Health information exchange (HIE)

Health information exchange (HIE) is the mobilization of health care information electronically across organizations within a region, community or hospital system.
In practice the term HIE may also refer to the organization that facilitates the exchange. HIE systems facilitate the efforts of physicians and clinicians to meet high standards of patient care through electronic participation in a patient's continuity of care with multiple providers. Secondary healthcare provider benefits include reduced expenses associated with:

- the manual printing, scanning and faxing of documents, including paper and ink costs as well as the maintenance of associated office machinery.

- the physical mailing of patient charts and records, and phone communication to verify delivery of traditional communications, referrals and test results.

- the time and effort involved in recovering missing patient information, including any duplicate tests required to recover such information[16].

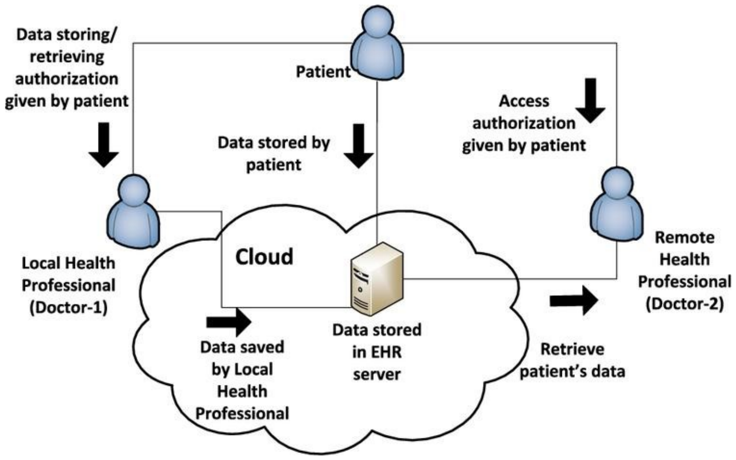


Figure 1.3: Health information exchange conception [16].

1.3.2.4 Health information management (HIM)

Health information is the data related to a person's medical history, including symptoms, diagnoses, procedures, and outcomes.

Health information records include patient histories, lab results, x-rays, clinical information and notes. A patient's health information can be viewed individually to see how a patient's health has changed, it can also be viewed as a part of a larger data set to understand how a population's health has changed and how medical interventions can change health outcomes.

Health information management (HIM) is the practice of acquiring, analyzing, and protecting digital and traditional medical information vital to providing quality patient care. It is a combination of business, science and information technology [17].

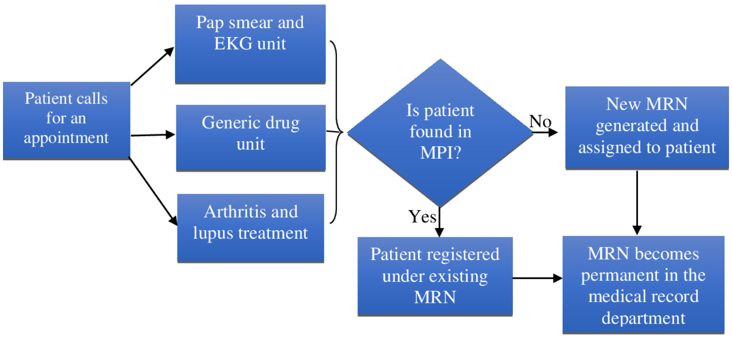


Figure 1.4: Example of Health information exchange conception [16].

1.3.2.5 International Classification of Diseases (ICD)

ICD is the foundation for the identification of health trends and statistics globally, and the international standard for reporting diseases and health conditions.

It is the diagnostic classification standard for all clinical and research purposes. ICD defines the universe of diseases, disorders, injuries and other related health conditions, listed in a comprehensive, hierarchical fashion that allows for:

- easy storage, retrieval and analysis of health information for evidenced-based decision-making.
- sharing and comparing health information between hospitals, regions, settings and countries.
- Data comparisons in the same location across different periods.

Uses include monitoring of the incidence and prevalence of diseases, observing reimbursements and resource allocation trends, and keeping track of safety and quality guidelines. They also include the counting of deaths as well as diseases, injuries, symptoms, reasons for encounter, factors that influence health status and external causes of disease[18].
Some version of IDC: ICD-6, ICD-7, ICD-8a, ICD-9, ICD-10 and ICD-11.

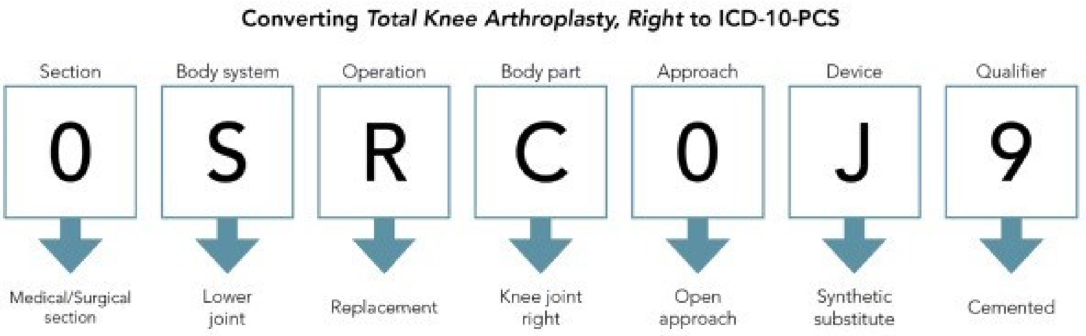


Figure 1.5: Example of classification of diseases ICD-10 [18].

1.3.2.6 Neuroinformatics

Is a research field concerned with the organization of neuroscience data by the application of computational models and analytical tools. These areas of research are important for the integration and analysis of increasingly large-volume, high dimensional and fine-grain experimental data.
Neuroinformaticians provide computational tools, mathematical models, and create interoperable databases for clinicians and research scientists. Neuroscience is a heterogeneous field, consisting of many and various sub-disciplines (e.g., cognitive psychology, behavioral neuroscience and behavioral genetics). In order for our understanding of the brain to continue to deepen, it is necessary that these sub-disciplines are able to share data and findings in a meaningful way, Neuroinformaticians facilitate this [19].

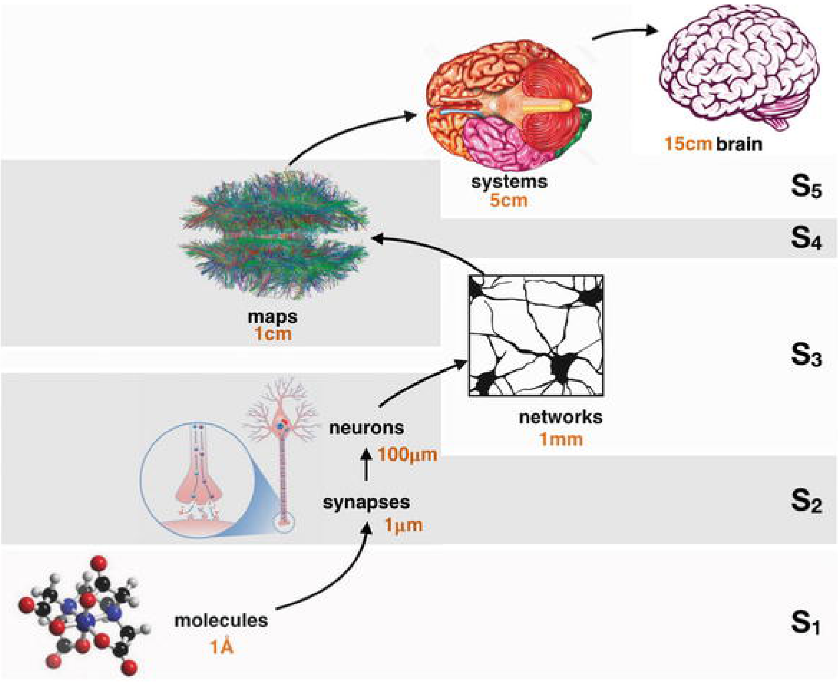


Figure 1.6: Data models in Neuroinformatics [19].

1.3.3 Importance of health informatics

Healthcare has been a constant topic of discussion in recent years and health informatics has played an important role in trying to ensure everyone is provided with the best possible quality of healthcare.

A big step was taken with the Affordable Care Act, which requires health care professionals to integrate healthcare with information technology by using electronic health records (EHR) and a combined healthcare infrastructure.

Health informatics can be found in every area of health care, including clinical, administrative and economic. Health informatics is important for many reasons.

- No more need to chart research just a click and you will found what u need.
- Make the communications between doctors and patients very easy and useful.
- Detecting of disease and fond the best traitement and cure.
- Gather quantities of data allowing a better analysis of the various diseases.
- Better follow-up of patients by doctors.
- Use of expert systems and machine learning for online monitoring patient health.
- Savings in material and financial resources relating to the movement of patients at various medical centers.
- Improvement of response times in emergency cases.

1.3.4 health informatics Technologies

Health informatics represents the intersection of information technology and design and delivery of health care services.

1.3.4.1 Robotics

Robotics is the design, engineering and use of robotic machines to perform partially or fully-automated physical cognitive functions.
One of the robotic machines on the market today is da Vinci, a robot-assisted and surgeon-controlled surgical system. The system uses minimally-invasive surgical instruments and 3D high-definition visuals to enhance accuracy and control.
Wearable robots are also poised to proliferate onto the market. These devices, such as robot protheses to replace missing limbs or robot exoskeletons to aid paralyzed people are used to rehabilitate or enhance a patient's physical capabilities [20].

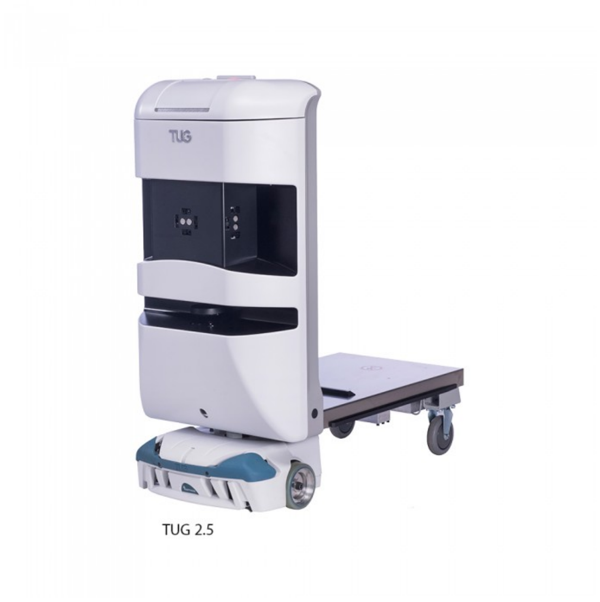


Figure 1.7: The TUG system transports a variety of hospital carts. It automates and tracks the delivery of goods in hospitals [21].

Robotics can also work in conjunction with health informatics to enable autonomous collection of data, which can provide accurate and continual data for health informatics pros across numerous health care areas, like surgery, drug delivery and cognitive rehabilitation and patient management.

1.3.4.2 3D Bioprinting

3D bioprinting is the printing of solid, three-dimensional objects from a digital file with an additive process utilizing different materials like liquid metals, polymers, ceramics,

epoxy resin or living cells.
Bioprinting living human tissue using stem cell bioink has the potential to reduce the shortage of organs for life-saving transplants. This is a work in progress: The company organovo made a human liver model out of 3D bioprinted liver tissues, which are being used for drug testing and development. Additionally, 3D-printed drugs can be tailored to specific patient needs, such as the FDA-approved Sprintam [22].

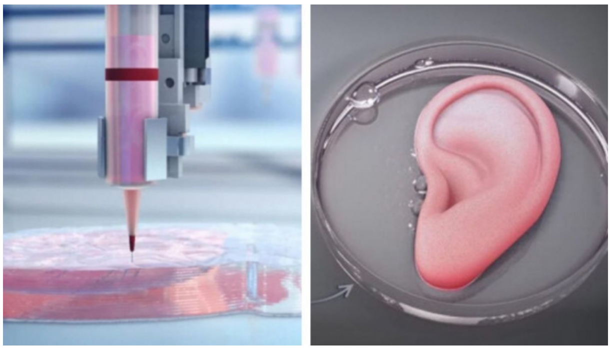


Figure 1.8: 3D-bioprinted ears [23].

1.3.4.3 Nanomedicine

Nanomedicine uses nanotechnology that is the study of molecular structure for developing precise devices to advance biomedical research and clinical practice.
Nanorobots injected into a patient's bloodstream can be used for drug delivery and disease monitoring. Therapeutic nanoparticles also have multiple applications in cancer patients, like improving circulation and delivering more sensitive diagnosis and imaging.

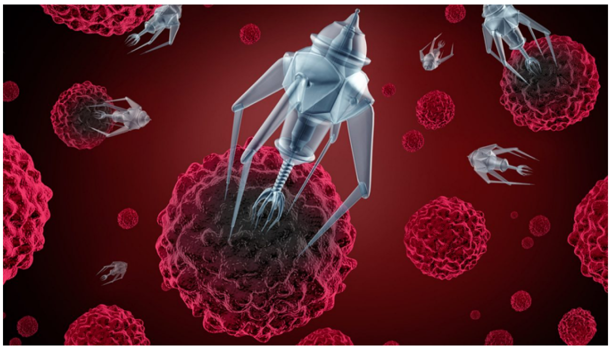


Figure 1.9: Healthcare Nanotechnology [24].

Nanomedicine informatics is necessary for integrating large, complex data sets at the nano level, implementing nanoparticles in living organisms and smart drug delivery. The National Cancer Institute's Biomedical informatics Grid (CIBIG) project established a collaborative research network for sharing cancer research data [25].

1.3.4.4 Cloud computing

Cloud computing enables the delivery and access to computing services and resources over the internet, including developmental platforms, databases, and analytics.
Cloud computing has diverse applications in the healthcare field. Its capacity for high volume data storage can result in lower costs for health care organizations. Real-time information sharing can also lead to better communication, especially in times of disaster. Additionally, remote patient monitoring and doctor interactions can produce better patient care. Finally, faster data sharing can lead to improved medical research.

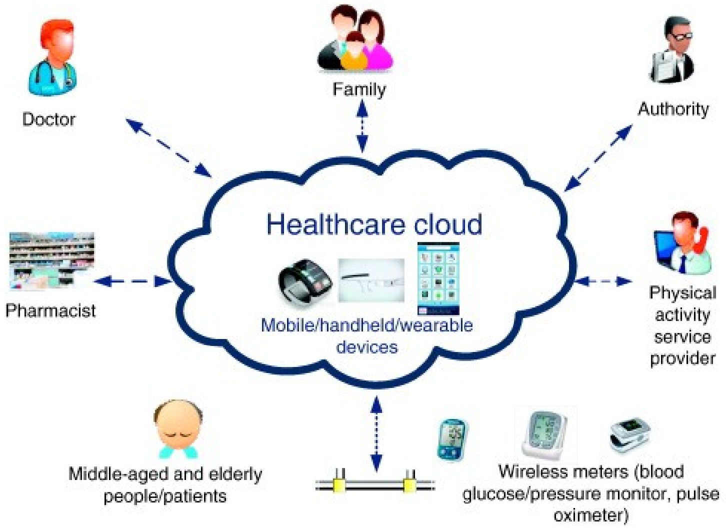

Doctor
Family
Authority
Healthcare cloud
Mobile/handheld/wearable devices
Pharmacist
Physical activity service provider
Middle-aged and elderly people/patients
Wireless meters (blood glucose/pressure monitor, pulse oximeter)

Figure 1.10: Cloud computing in the health care industry [26].

There's greater access to large amounts of patient health and medical research data than ever before. Health informatics is a necessary field for managing this data and making it available and useful for healthcare pros [27].

1.3.5 History of health informatics

Healthcare is an important science in life, so we must be more exact and sure of our result this is why doctors start using computer science in this domain, but how and when computer science start function in healthcare, the answer will be more clear in this Historical overview:

The field of health informatics, as we know it now, emerged when computer technology became sophisticated enough to manage large amounts of data. There were earlier experimentations in the field of dentistry, but it wasn't until the 1960's that health informatics began to standardize as a field of study.
According to HIMSS, Healthcare Information and Management Systems Society, the first standards for healthcare data reporting were established by the American Society for Testing and Materials (ASTM). These include standards for "laboratory message exchange,

properties for electronic health record systems, data content, and health information system security".

The first electronic medical record appeared soon after and new nomenclature for specific disciplines and services took shape. The subspecialty of bioinformatics was introduced in the late 1970's as the study of biological data, including DNA. Meanwhile, professional associations and government entities worked to further define the format for patient registration, orders, observation, discharge, insurance claims and financial transaction messages.

As health information technology continued to improve, so did requests for standardized data exchange protocols for disciplines like radiology and pharmacy. Today, imaging informatics and pharmacy informatics are two uniquely specialized areas of study for individuals pursuing a master's in health informatics.

Global standards were ultimately included as well, making the field of health informatics into the complex and still developing discipline that it is today. Perhaps the great irony of the field is that there is no standard when it comes to the technology behind the data. Hospital systems may have two or more different software systems in place.

Clinical information systems professionals have the challenge of achieving interoperability between various formats in order to ensure information availability across all areas of the healthcare enterprise.

However, in the final analysis, many health organizations have no choice but to opt for custom programming solutions to integrate dissimilar systems. There is tremendous opportunity for individuals who have the knowledge and training to continue evolving and improving the standards for both technology and patient care.

The future belongs to the creative and analytical informaticist who can serve as a liaison between the information technology team and clinicians. Understanding both sides of the health informatics equation is a must to engender collaboration, champion new initiatives, and move your career forward.

And now healthcare will use AI in diagnostics,treatments, operations and more in the future [28] .

1.4 Conclusion

This chapter covered heath informatics, healthcare, how informatics helped in healthcare, also definitions of some heath informatics technologies and in the end the history of health informatics.

Health informatics is a rapidly changing field that touches nearly every aspect of the modern healthcare organization.

The next chapter will introduce and discuss Deep learning and the chosen case to study .

Chapter 2

Deep learning

2.1 Introduction

Machine learning and deep learning have provided us with an exploration of a whole new research area. As more data and better computational power become available, they have been implemented in various fields.

The demand for artificial intelligence in the field of health informatics is also increasing and we can expect to see the potential benefits of artificial intelligence applications in healthcare.

Deep learning can help clinicians diagnose disease [29], identify cancer sites [30], identify drug effects for each patient [31], understand the relationship between genotypes and phenotypes [32] and predict infectious disease outbreaks with high accuracy [33].

In contrast to traditional models, its approach does not require domain-specific data preprocess, and it is expected that it will ultimately change human life a lot in the future.

Deep Learning is popular right now because it's easy and it works, but **what is it?**

First, we must understand the basis through which deep learning emerged.

2.2 Machine learning

Machine learning (ML) is a branch of artificial intelligence that systematically applies algorithms to synthesize the underlying relationships among data and information. For example, ML systems can be trained on automatic speech recognition systems (such as iPhone's Siri) to convert acoustic information in a sequence of speech data into semantic structure expressed in the form of a string of words. ML is already finding widespread uses in web search, credit scoring, stock market prediction, gene sequence analysis, behavior analysis, smart coupons, drug development, weather forecasting, big data analytics and many more applications.

ML will play a decisive role in the development of a host of user-centric innovations [34].

In 1959, Arthur Samuel described ML as the "field of study that gives computers the ability to learn without being explicitly programmed". He concluded that programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.
According to Tom M. Mitchell's definition of ML: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [34].

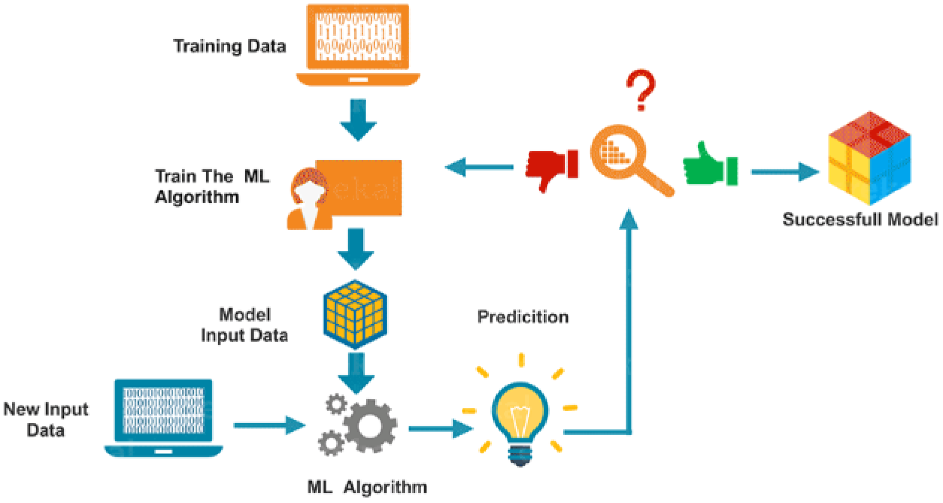


Figure 2.1: Machine learning design [35].

2.2.1 Type of learning

Machine learning algorithms are organized into taxonomy, based on the desired outcome of the algorithm. Common algorithm types include:
supervised learning, unsupervised learning and reinforcement learning.

2.2.1.1 Supervised learning

where the algorithm generates a function that maps inputs to desired outputs.
One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behavior of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.
Supervised learning techniques: Linear Regression, Logistic Regression, CART, Naïve Bayes, KNN and deep learning.

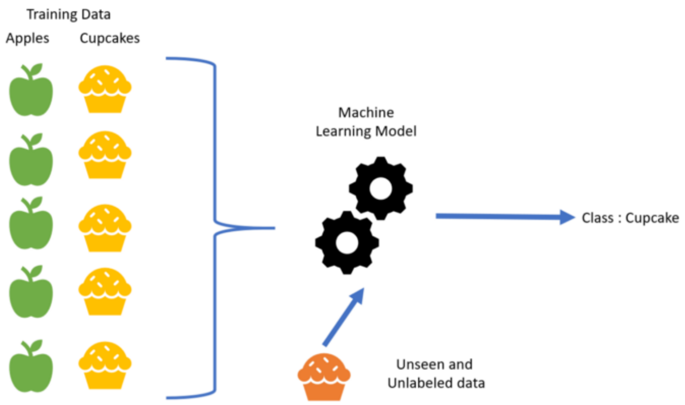


Figure 2.2: Supervised learning example [36].

For example, looking at the figure 2.2, we can tell there are two classes : apples and cupcakes. And we have a bunch of training data for them. Our algorithm learns the specifications of both the classes and now can predict the class of a new coming object.

2.2.1.2 Unsupervised learning

The model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it. What it cannot do is add labels to the cluster, like it cannot say this a group of apples or mangoes, but it will separate all the apples from mangoes [35].
Unsupervised learning techniques: Apriori, K-means, SVM and PCA .

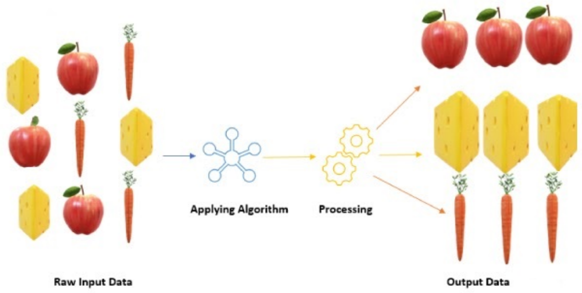


Figure 2.3: Unsupervised learning example [37].

For example, looking at the figure 2.3, we can say that if we are feeding apple, carrot and cheese as raw input data then our model will distinguish all three but it cannot tell whether a given cluster is of apple or not as it is unlabelled but any new data will automatically fit into the clusters that are formed .

2.2.1.3 Reinforcement learning

Reinforcement Learning is about taking suitable actions to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path to take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it, so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer and the reinforcement agent decides what to do in order to perform the given task. In the absence of training data set, it is bound to learn from its experience [38].

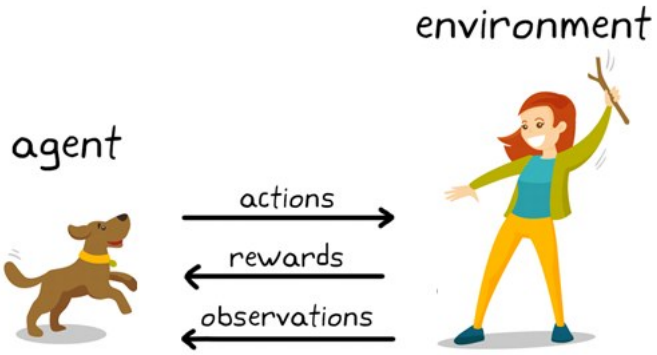


Figure 2.4: Reinforcement learning in dog training [39].

The goal of reinforcement learning in this example of the figure 2.4 is to train the dog (agent) to complete a task within an environment, which includes the surroundings of the dog as well as the trainer. First, the trainer issues a command or cue, which the dog observes (observation).

The dog then responds by taking an action. If the action is close to the desired behavior, the trainer will likely provide a reward, such as a food treat or a toy; otherwise, no reward or a negative reward will be provided.

At the beginning of training, the dog will likely take more random actions like rolling over when the command given is sit, as it is trying to associate specific observations with actions and rewards .

From the dog's perspective, the ideal case would be one in which it would respond correctly to every cue, so that it gets as many treats as possible.

So, the whole meaning of reinforcement learning training is to tune the dog's policy so that it learns the desired behaviors that will maximize some reward. After training is complete, the dog should be able to observe the owner and take the appropriate action [39].

2.3 Deep learning

Deep learning, as a new area of machine learning research, is a process which allows the computer to learn to perform tasks which are natural for the brain like image recognition. Currently, deep learning (DL) methods have had a profound impact on computer vision and image analysis applications, such as image classification, segmentation, image completion and so on. Deep learning focuses on a specific category of machine learning called Artificial Neural Networks which is inspired by functionality of the human brain. Modern deep learning provides a very powerful framework for supervised learning. By adding more layers and more units within a layer, a deep network can represent functions of increasing complexity.

Most tasks that consist of mapping an input vector to an output vector, and that are easy for a person to do rapidly, can be accomplished via deep learning, given sufficiently large models and sufficiently large datasets of labeled training examples [40].

2.3.1 Artificial Neural Networks

Artificial Neural networks have attracted considerable interest in recent years due to their ability to learn complicated maps from examples, an ability termed universal approximation.

Called networks because they are typically represented by composing together many different functions called neural because they are loosely inspired by neuroscience [40].

The learning algorithm must decide how to use those layers to produce the desired output, but the training data does not say what each individual layer should do.

Although artificial neurons were inspired by the biological processes scientists were able to observe in the brain back in the 50s, but artificial and biological neurons do differ in more ways than just the materials of their containers. So,we have to talk about the basic units of a neural network.

2.3.1.1 Biological neuron

The neuron is the basic building block of the brain and central nervous system. Neurons are specialized cells that transmit chemical and electrical signals. The brain is made up entirely of neurons and glial cells, which are non-neuronal cells that provide structure and support for the neurons. Nearly 86 billion neurons work together within the nervous system to communicate with the rest of the body. They are responsible for everything from consciousness and thought to pain and hunger [41].

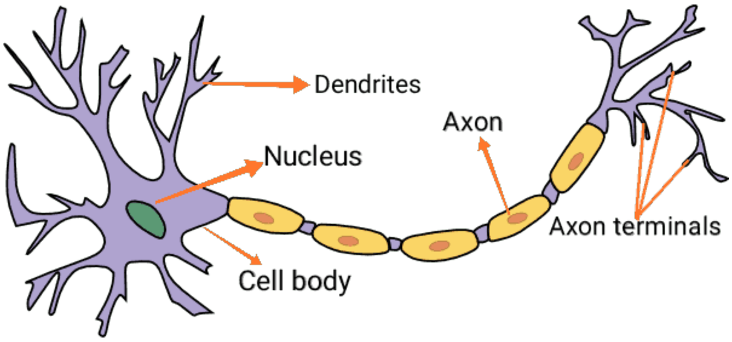


Figure 2.5: Biological neuron structure [42].

The above image shows the basic structural components of an average neuron, including the dendrite, cell body,axon and axon terminal.

1. **Dendrites** are branch-like structures extending away from the cell body, and their job is to receive messages from other neurons and allow those messages to travel to the cell body .

2. **cell body (or soma)** that contains a nucleus, smooth and rough endoplasmic reticulum, Golgi apparatus, mitochondria, and other cellular components.

3. **An axon**, at its most basic, is a tube-like structure that carries an electrical impulse from the cell body (or from another cell's dendrites) to the structures at opposite end of the neuron—axon terminals, which can then pass the impulse to another neuron. The cell body contains a specialized structure, the axon hillock, which serves as a junction between the cell body and the axon.

4. **The synapse** is the chemical junction between the axon terminals of one neuron and the dendrites of the next. It is a gap where specialized chemical interactions can occur, rather than an actual structure.

Function of a Neuron: The specialized structure and organization of neurons allows them to transmit signals in the form of electric impulses from the brain to the body and back. Individually, neurons can pass a signal all the way from their own dendrites to their own axon terminals; but at a higher level neurons are organized in long chains, allowing them to pass signals very quickly from one to the other. One neuron's axon will connect chemically to another neuron's dendrite at the synapse between them. Electrically charged chemicals flow from the first neuron's axon to the second neuron's dendrite, and that signal will then flow from the second neuron's dendrite, down its axon, across a synapse, into a third neuron's dendrites, and so on [43].

2.3.1.2 Artificial neuron

Artificial neuron is a basic building block of every artificial neural network. Its design and functionalities are derived from observation of a biological neuron that is basic building block of biological neural networks (systems) which includes the brain, spinal cord and peripheral ganglia.
Biological neuron and artificial neuron are Similar in design and functionalities . where the left side of a figure 2.6 represents a biological neuron with its soma, dendrites and axon and where the right side of a figure 2.6 represents an artificial neuron with its inputs, weights, transfer function, bias and outputs [44].

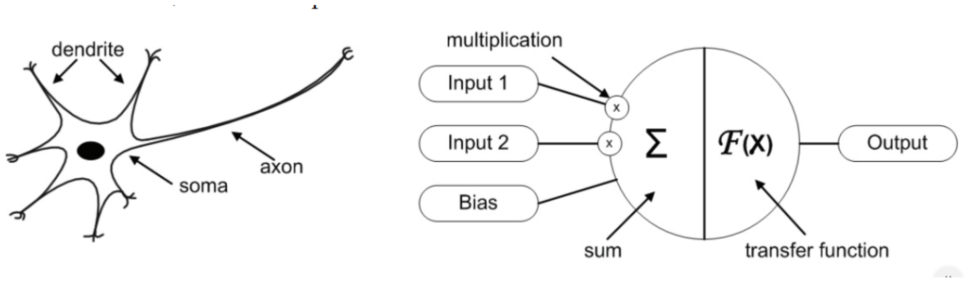


Figure 2.6: Biological and artificial neuron design [44].

In case of biological neuron the information comes into the neuron via dendrite, soma processes the information and passes it on via axon. In case of artificial neuron the information comes into the body of an artificial neuron via inputs that are weighted (each input can be individually multiplied with a weight).
The body of an artificial neuron then sums the weighted inputs, bias and "processes" the sum with a transfer function. At the end an artificial neuron passes the processed information via output(s). Benefit of artificial neuron model simplicity can be seen in its mathematical description below [44]:

$$y(k) = F(\sum_{i=0}^{m} w_i(k) * x_i(k) + b) \tag{2.1}$$

Where:

- x$_i(k)$is input value in discrete time k where i goes from 0 to m.

- w$_i(k)$is weight value in discrete time k where i goes from 0 to m.

- b is bias.

- f is a transfer function.

- y(k) is output value in discrete time k.

As seen from a model of an artificial neuron and its equation 2.1 the major unknown variable of our model is its transfer function. Transfer function defines the properties of artificial neuron and can be any mathematical function.
We choose it on the basis of problem that artificial neuron (artificial neural network) needs to solve and in most cases we choose it from the following set of functions: binary step function, linear function and non-linear function [44].
Binary step function:
A binary step function is a threshold-based activation function. If the input value is above or below a certain threshold, the neuron is activated and sends exactly the same signal to the next layer.

$$y = \begin{cases} 0 & \text{if } w_i x_i \geqslant \text{threshold} \\ 1 & \text{if } w_i x_i < \text{threshold} \end{cases} \tag{2.2}$$

The problem with a step function is that it does not allow multi-value output for example, it cannot support classifying the inputs into one of several categories [45].
Linear function:
A linear activation function takes the form: A = cx (2.3)
It takes the inputs, multiplied by the weights for each neuron, and creates an output signal proportional to the input. In one sense, a linear function is better than a step function because it allows multiple outputs, not just yes and no.

However, a linear activation function has two major problems:

- **Not possible to use back-propagation** (gradient descent) to train the model—the derivative of the function is a constant, and has no relation to the input, X. So it's not possible to go back and understand which weights in the input neurons can provide a better prediction.

- **All layers of the neural network collapse into one** with linear activation functions, no matter how many layers in the neural network, the last layer will be a linear function of the first layer (because a linear combination of linear functions is still a linear function). So a linear activation function turns the neural network into just one layer.

A neural network with a linear activation function is simply a linear regression model. It has limited power and ability to handle complexity varying parameters of input data [45].
Non-linear function:
Modern neural network models use non-linear activation functions. They allow the model to create complex mappings between the network's inputs and outputs, which are essential for learning and modeling complex data, such as images, video, audio, and data sets which are non-linear or have high dimension.

Almost any process imaginable can be represented as a functional computation in a neural network, provided that the activation function is non-linear.

Non-linear functions address the problems of a linear activation function:

- They allow back-propagation because they have a derivative function which is related to the inputs.

- They allow “stacking” of multiple layers of neurons to create a deep neural network. Multiple hidden layers of neurons are needed to learn complex data sets with high levels of accuracy [45].

The most famous non-linear activation functions are mentioned in the table 2.1 below.

Function	Formula	Advantages	Disadvantages
Sigmoid	$S(x) = \frac{1}{1 - e^{-x}}$	Smooth gradient. Output values bound. Clear predictions.	Vanishing gradient. Outputs not zero centered. Computationally expensive.
Hyberbolic Tangent	$Tanh(x) = \frac{sinh(x)}{cosh(x)}$	Zero centered.	Like the sigmoid function.
Softmax	$\delta(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$	Able to handle multiple classes . Useful for output neurons.	Does not support non linearly separable data. Does not support null rejection.
Relu	$y = max(x \cdot 0.1, x)$	Computationally efficient.	The dying Relu problem

Table 2.1: Some of non-linear activation functions [45].

2.3.1.3 Layers

In this classic artificial neural networks there are many types of layers used in the network, each type of layer is responsible for some computations.

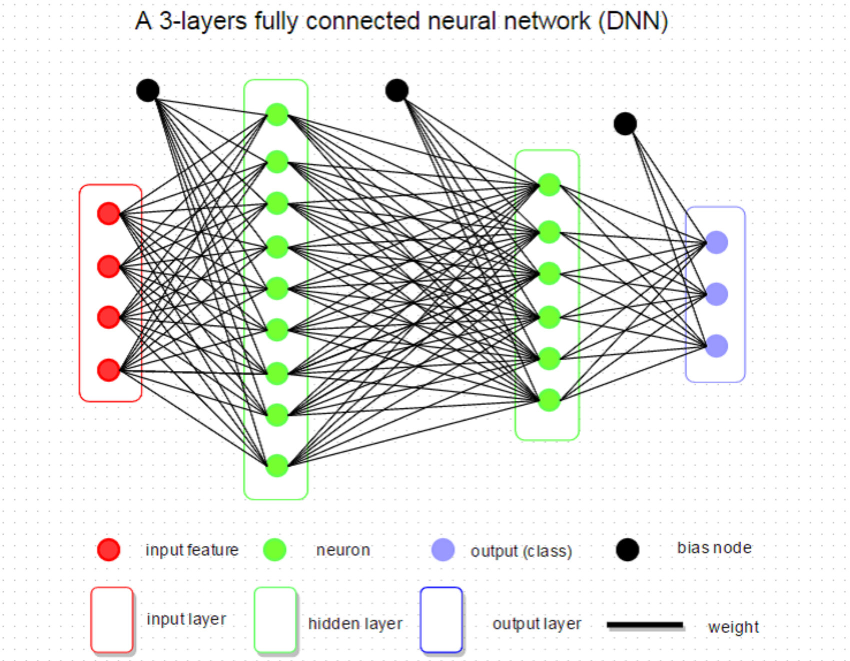


Figure 2.7: Artificial neural networks architecture [40].

- **Input layer:**Input layer is the first layer in the neural network, composed of input neurons and brings initial data to the hidden layers for further processing.
- **Hidden layer:** The layer or group of layers between the input and output layer. Deep learning is an optimization problem looks for the optimal solution of a very complex problem, many of these computations are made in the hidden layer(s).The choice of hidden layers depends on the complexity of the data, when using a less complex data it's recommended to use few hidden layers, using many hidden layers in a simple problem can lead to overfitting, and using simple architecture in complex problem leads to underfitting.
  Each hidden layer of the network is typically vector-valued. The dimensionality of these hidden layers determines the width of the model.
- **Output layer:** Output layer is the last layer in neural network which produces the outputs of the program, in classification tasks, the size of output layer is equal to number of classes [40].

2.3.2 Deep neural network architectures

There are many different neural network architectures and with time will be more architectures, so we choose some of popular and widely used neural network architectures that you should know in order to advance your knowledge about neural network architectures.

2.3.2.1 Convolutional Neural Networks

Convolutional neural networks or CNNs in short, are the popular choice of neural networks for different Computer Vision tasks such as image recognition. The name convolution is derived from a mathematical operation involving the convolution of different functions.
There are 4 primary layers or stages in designing a CNN:

- **Convolution:**The input signal is received at this layer.
- **Subsampling or Pooling :**Inputs received from the convolution layer are smoothened to reduce the sensitivity of the filters to noise or any other variation.
- **Activation:**This layer controls how the signal flows from one layer to the other, similar to the neurons in our brain.
- **Fully connected:**In this stage, all the layers of the network are connected with every neuron from a preceding layer to the neurons from the subsequent layer [46].

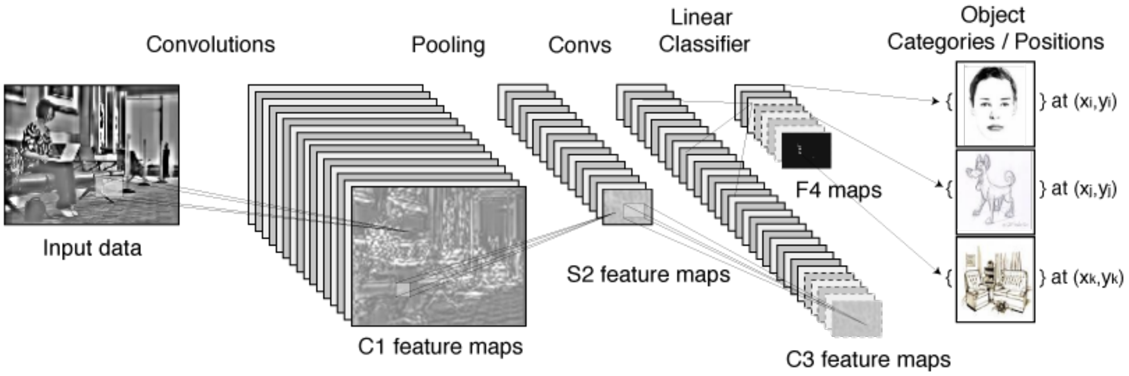


Figure 2.8: CNN architecture for visual recognition [47].

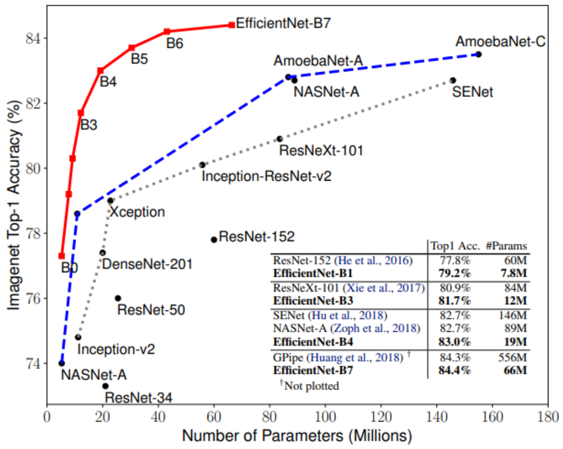
Advantages of CNN

- Very good for visual recognition.
- Once a segment within a particular sector of an image is learned, the CNN can recognize that segment present anywhere else in the image.

Disadvantages of CNN

- CNN is highly dependent on the size and quality of the training data.
- Highly susceptible to noise [48][49].

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future. Some of them: LeNet [50], AlexNet [51], ResNet [52], EfficientNet [53] and InceptionNet [54].



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 2.9: Imagenet Top-1 accuracy cnn models [53]

2.3.2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) have been very popular in areas where the sequence in which the information is presented is crucial. As a result, they find a lot applications in real-world domains such as natural language processing, speech synthesis and machine translation.
RNNs are called 'recurrent' mainly because a uniform task is performed for every single element of a sequence, with the output dependant on the previous computations as well. Think of these networks as having a memory, where every calculated information is captured, stored and utilized to calculate the final outcome [55].
Over the years, quite a few varieties of RNNs have been researched and developed:

- **Bidirectional RNN :** The output in this type of RNN depends not only on the past but also the future outcomes.
- **Deep RNN :**In this type of RNN, there are multiple layers present per step, allowing for a greater rate of learning and more accuracy.

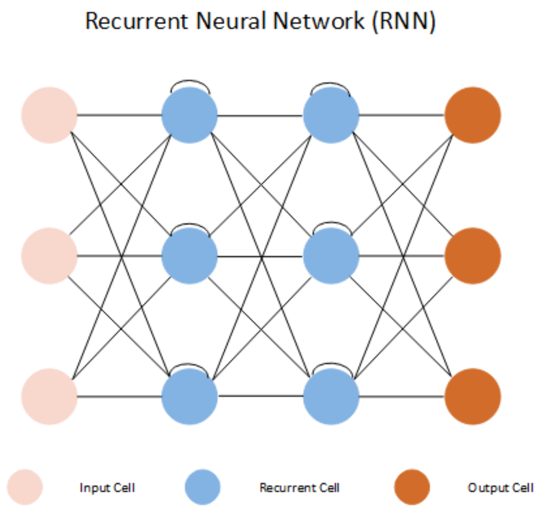


Figure 2.10: RNN architecture [56]

Advantages of RNN

- Unlike a traditional neural network, an RNN shares the same parameters across all steps. This greatly reduces the number of parameters that we need to learn.
- RNNs can be used along with CNNs to generate accurate descriptions for unlabeled images.

Disadvantages of RNN

- RNNs find it difficult to track long-term dependencies. This is especially true in case of long sentences and paragraphs having too many words in between the noun and the verb.
- RNNs cannot be stacked into very deep models. This is due to the activation function used in RNN models, making the gradient decay over multiple layers [57].

Architectures of RNNs: Identity-RNN, np-RNN, LSTM and GRU.

2.4 case study

Deep learning in healthcare remains a field bursting with possibility and remarkable innovation. for that ,we will applicate it on diabetic retinopathy detection .

2.4.1 Diabetes

When you hear the word “diabetes,” your first thought is likely about high blood sugar. Blood sugar is an often-underestimated component of your health. When it’s out of whack over a long period of time, it could develop into diabetes. Diabetes affects your body’s ability to produce or use insulin, a hormone that allows your body to turn glucose (sugar) into energy. Here’s what symptoms may occur to your body when diabetes takes effect [58].

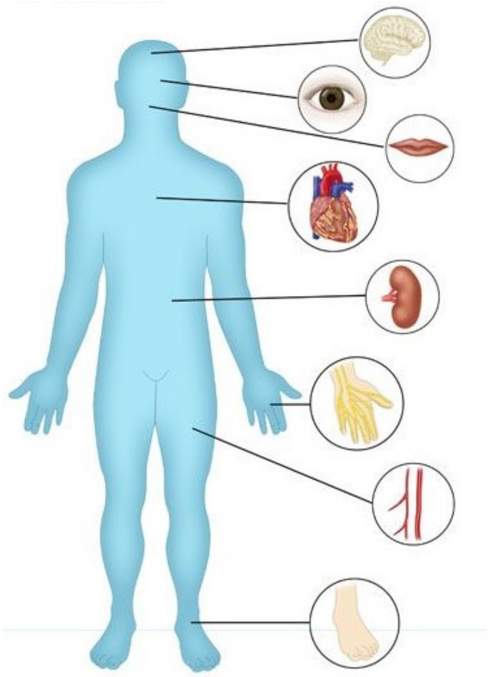


Figure 2.11: The effect of diabetes on the body [58].

- **The effect of diabetes on the heart :** Diabetes contributes to high blood pressure and is linked with high cholesterol which significantly increases the risk of heart attacks and cardiovascular disease.
- **The effect of diabetes on the storkes :** Similar to how diabetes affects the heart, high blood pressure and cholesterol raises the risk of strokes.
- **The effect of diabetes on the eyes :** As with all complications, this condition is brought on by a number of years of poorly controlled or uncontrolled diabetes. Diabetic retinopathy has a number of symptoms.
  Retinopathy is caused by blood vessels in the back of the eye (the retina) swelling and leaking. High blood pressure is also a contributing factor for diabetic retinopathy.
- **The effect of diabetes on the nerves :** The effects of diabetes on the nerves can be serious as the nerves are involved in so many of our bodily functions, from movement and digestion through to sex and reproduction.
- **The effect of diabetes on the skin :** Diabetes affect on the skin is usually a result of its affect on the nerves and circulation which can lead to dry skin, slow healing of cuts, burns and wounds, fungal and bacterial infections and loss of feeling in the foot [59].

After we saw some of the effect of diabetes on the body ,we choose to study diabetic retinopathy using informatics methods.

2.4.2 Diabetic retinopathy

Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

At first, diabetic retinopathy may cause no symptoms or only mild vision problems. Eventually, it can cause blindness.

The condition can develop in anyone who has type 1 or type 2 diabetes. The longer you have diabetes and the less controlled your blood sugar is, the more likely you are to develop this eye complication[60].

Symptoms You might not have symptoms in the early stages of diabetic retinopathy. As the condition progresses, diabetic retinopathy symptoms may include:

- Spots or dark strings floating in your vision (floaters)
- Blurred vision
- Fluctuating vision
- Impaired color vision

- Dark or empty areas in your vision

- Vision loss

Diabetic retinopathy usually affects both eyes.

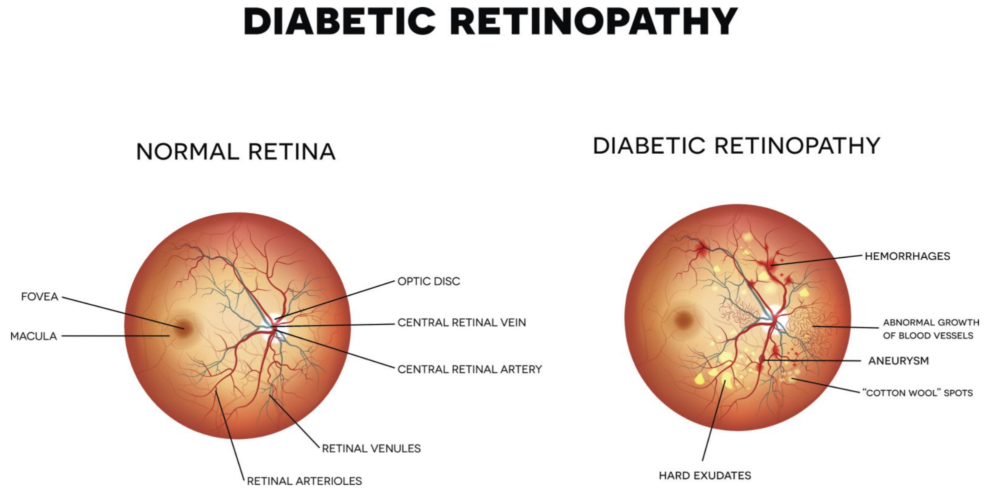


Figure 2.12: difference between normal retina and diabetic retinopathy [61].

2.4.2.1 Stages of diabetic retinopathy

1. **Mild nonproliferative retinopathy :** The first stage is also called background retinopathy. It means that there are tiny bulges in the tiny blood vessels in your retinas. The bulges are called microaneurysms. They may cause the vessels to leak small amounts of blood into your retinas.
   At this early stage, you probably don't have vision problems, so you may not need treatment [60].

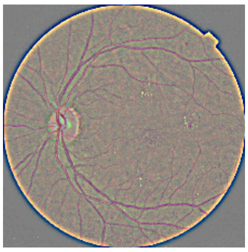


Figure 2.13: Mild nonproliferative retinopathy exemple.

2. **Moderate nonproliferative retinopathy :** The second stage is also called pre-proliferative retinopathy. At this stage, the blood vessels in your retinas swell. They may not carry blood as well as they used to. These things can cause physical changes to the retina .
   These changes can lead to diabetic macular edema . This happens when blood and other fluids build up in a part of your retina called the macula. The macula is key

for straight-ahead vision, like when you read or drive. When it swells, it can cause problems with this important part of your vision.

Half the people with diabetic retinopathy will get DME. It can happen at any stage, but it's more likely to happen as the condition advances [60].

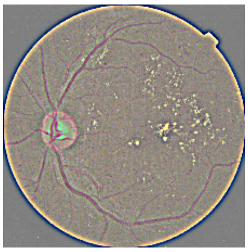


Figure 2.14: Moderate nonproliferative retinopathy exemple.

3. **Severe nonproliferative retinopathy** : This is also called proliferative retinopathy. In this stage, your blood vessels become even more blocked. This means even less blood goes to the retinas. Because of this, scar tissue forms. The lack of blood triggers a signal to your retinas to create new blood vessels.

   If the blood vessels close off completely, it's called macular ischemia. This can lead to blurry vision with dark spots some people describe as "floaters."

   If you reach this stage, there's a very high chance that you'll lose your vision [60].

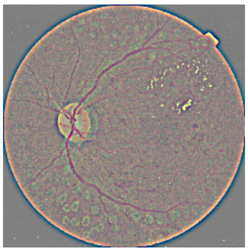


Figure 2.15: Severe nonproliferative retinopathy exemple.

4. **Proliferative diabetic retinopathy (PDR) :** In this advanced stage, new blood vessels grow in your retinas and into the gel-like fluid that fills your eyes. This growth is called neovascularization. These vessels are thin and weak. They often bleed. The bleeding can cause scar tissue.

   As the scar tissue gets smaller, it can pull the retina away from the back of your eye. This is called retinal detachment. It can lead to permanent loss of both straight-ahead and side vision [60].

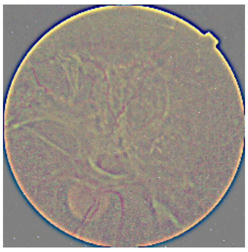


Figure 2.16: Proliferative diabetic retinopathy exemple.

2.4.2.2 Retinal imaging

Retinal imaging has developed rapidly during the last 160 years and is a now a mainstay of the clinical care and management of patients with retinal as well as systemic diseases. Fundus photography is widely used for population-based, large-scale detection of DR, glaucoma, and age-related macular degeneration. And with the drastic improvement in smartphone optical and sensory capabilities over the previous decade, using smartphones in fundus imaging is becoming a more powerful clinical tool, particularly in low-resource settings where high-end informatics infrastructure and imaging systems for teleophthalmology are not as widely available. Smartphone photography and videography has been used with increasing frequency since 2010, as first reported by Lord and colleagues. Currently, there are at least five known apparatus/software package solutions on the market: DigiSight Paxos Scope (San Francisco, USA), Peek Retina (London, UK), D-EYE (Padova, Italy), Remidio Fundus on Phone (Bangalore, India)], and Welch Allyn Panoptic with iExaminer (Skaneateles Falls, USA) [62].

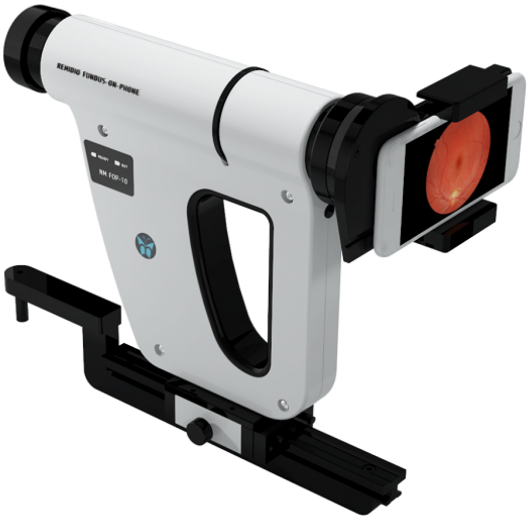


Figure 2.17: Remidio Fundus on Phone [63]

2.4.3 Related works

This subsection mentions the previous works dealing with classifying retina images using machine learning techniques.

2.4.3.1 Computer-based detection of diabetes retinopathy stages using SVM

U. R. Acharya et al . In this work, morphological image processing and support vector machine (**SVM**) techniques were used for the automatic diagnosis of eye health. In this study, 331 fundus images were analysed. Five groups were identified: normal retina, mild non-proliferative diabetic retinopathy, moderate non-proliferative diabetic retinopathy, severe non-proliferative diabetic retinopathy, and proliferative diabetic retinopathy. Four salient features blood vessels, microaneurysms, exudates, and haemorrhages were extracted from the raw images using image-processing techniques and feed to the SVM for classification. A sensitivity of more than 82 % and specificity of 86% was demonstrated for the system developed [64].

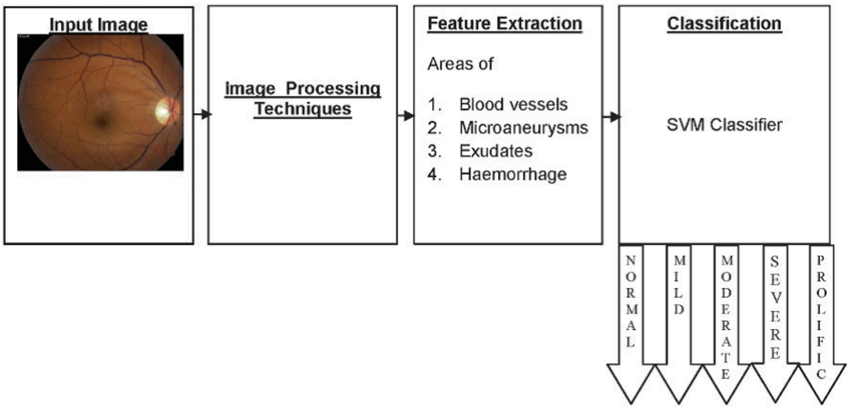


Figure 2.18: Proposed system for the detection and classification of different stages of DR [64].

2.4.3.2 Classification of diabetic and normal fundus images using new deep learning method

M. T. Esfahan et al. They used a known CNN, which is ResNet34 in their study to classify DR images of the Kaggle dataset into normal or DR image. ResNet34 is one available pretrained CNN architecture on ImageNet database. They applied a set of image preprocessing techniques to improve the quality of images. The image preprocessing included the Gaussian filter, weighted addition and image normalization. The image number was 35000 images and its size was (512,512) pixels. They reported an accuracy of 85 % and a sensitivity of 86 % [65].

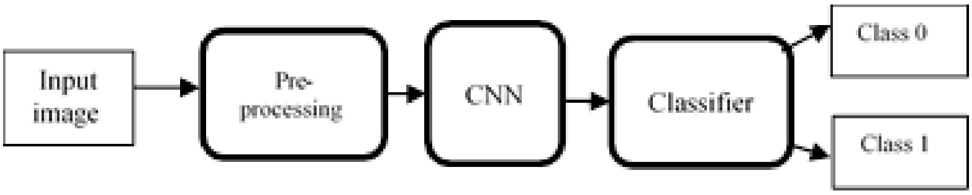


Figure 2.19: Block diagram of the proposed method (class 0= normal,class 1=DR) [65].

2.4.3.3 Decision tree CART algorithm for diabetic retinopathy classification

Aziza, Elaouaber Zineb, et al. Propose an automatic system for diabetic retinopathy detection from color fundus images. The proposed approach is based on the segmentation of blood vessels and extracts the geometric features, which are used in the early detection of diabetic retinopathy. The Hessian matrix, ISODATA algorithm and active contour are used for the segmentation of the blood vessels, they have used. Finally, they have applied the decision tree CART algorithm to classify images into normal (NO-DR) or DR. The proposed system was tested on the drive and Messidor datasets and achieved an average sensitivity, specificity and accuracy of 89%, 99% and 96%, respectively for the segmentation of retinal vessels and 91%, 100% and 93%, respectively for the classification of diabetic retinopathy [66].

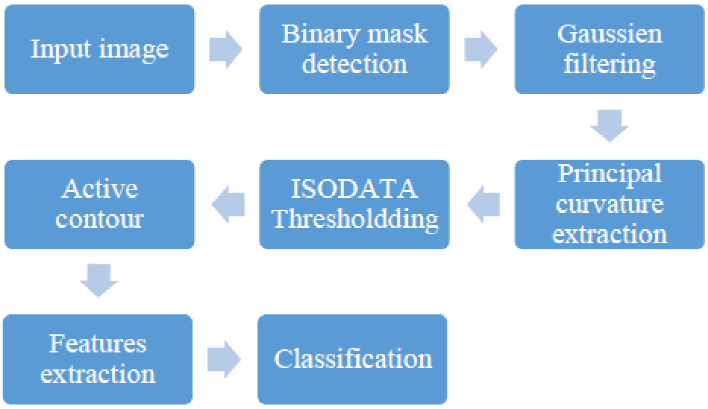


Figure 2.20: Flowchart of the proposed method for diabetic retinopathy detection from retinal image [66].

2.5 Conclusion

This chapter covered definitions of machine learning, deep learning, neural networks and diabetic retinopathy.
The objective of this chapter was to present deep learning architecture starting by neural network to the differences architectures of it, then give a general vision about our chosen topic which is diabetic retinopathy detection by given previous works on it associated with each method. .
Deep learning was the main focus of the chapter due to its importance and also because it is the main topic of this research.
The next chapter will introduce the design of the system, dataset and the proposed architectures will be mentioned as well.

Chapter 3

System design

3.1 Introduction

Diabetic retinopathy detection has attracted considerable attention in recent decades. Researchers have made significant breakthroughs in this area with the rapid development of deep learning algorithms and the most successful applications in this field.
far, they are using neural network-based methods. The convolutional neural networks have been very successful in various image recognition applications, including those of diabetic retinopathy detection.
In our project, we were interested in creating a 3 famous CNN models that uses image processing methods to get the best results for detection diabetic retinopathy .
To achieve our goal, we propose in this chapter alexnet, resnet and efficientnet architectures applied on 3 different structures of our original dataset .

3.2 Methodology

Generally, Our diabetic retinopathy detection system will be following a certain steps, as we represent the system in figure 3.1.

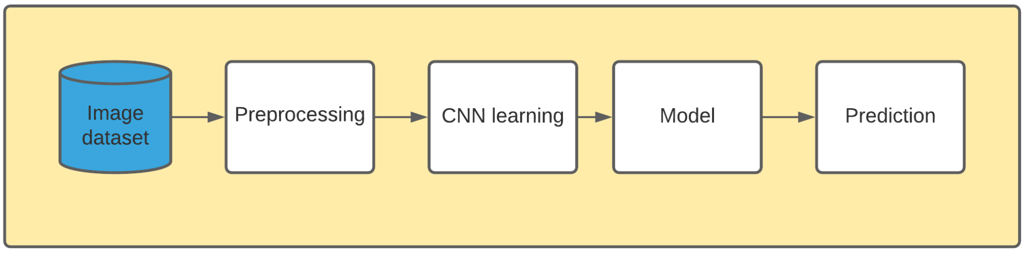


Figure 3.1: Block diagram of the proposed system.

As we noticed in the previous figure 3.1, 5 phases are part of system .

The system start by getting the images from the dataset, each image with its class after that we applicat preproccessing on the images and the dataset structure .
In the third phase we feed our CNN model with the images to learn and get a new model to pridict the class of new diabtect retinopathy images in the prediction phase .

3.2.1 Dataset description

We have used Kaggle images dataset from [67], the dataset is organized into 5 folders (No DR,Mild, Moderate,Severe,Proliferate DR) represent DR stages . There are 3,662 retina images (PNG) .
The original dataset is available at APTOS 2019 Blindness Detection [68]. These images are resized into 224x224 pixels so that they can be readily used with many deep learning models .
We create a new dataset by reorganized the dataset into 2 floders (No DR,DR) ,and another one reonrganized into 3 floders (No DR, NonProliferate,Proliferate DR).
Now we have 3 dataset to start applicat deep learning on each one.
But how we reorganized the new two datasets ?

3.2.1.1 Dataset structure

To make it clear ,we will describe our proposed structure of each dataset.
The first dataset: The first dataset is organized into 5 classes :

1. **No DR:**The patient retina is normal.
2. **Mild :**The patient has mild nonproliferative retinopathy.
3. **Moderate :**The patient has moderate nonproliferative retinopathy.
4. **Severe :**The patient has severe nonproliferative retinopathy.
5. **Proliferate DR:**The patient has proliferative retinopathy.

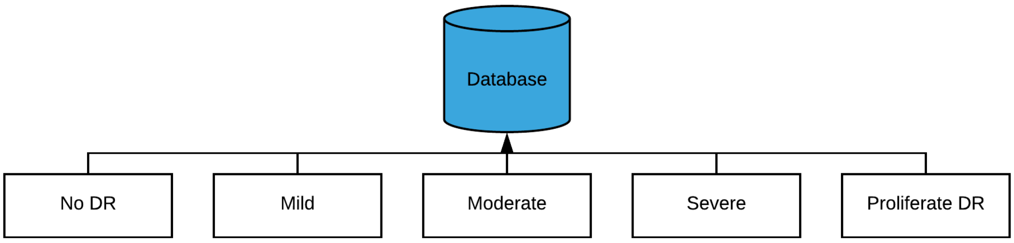


Figure 3.2: the first dataset splitting.

The second dataset: In this dataset ,we suppose in the mild stage the patient still have a normal retina because There is small areas of balloon-like swelling in the retina's tiny blood vessels. . and the 3 next stages the patient has diabetec retinopathy .
So we concatenate mild and no DR folders in one folder called no DR and we do the same to 3 other folders (Moderate,Severe,Proliferate DR) in the name DR for the new folder.

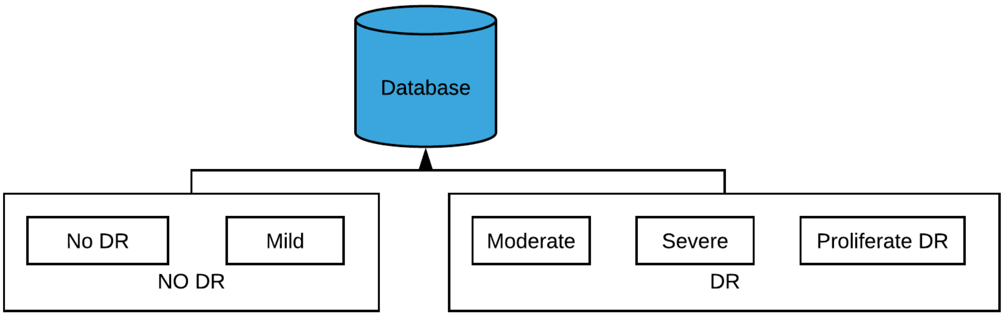


Figure 3.3: the second dataset splitting.

The third dataset: in this dataset ,we will have 3 classes :

1. **No DR:**The patient retina is normal.
2. **nonproliferative DR :**The patient has mild, moderate or severe .
3. **Proliferate DR:**The patient has proliferative retinopathy.

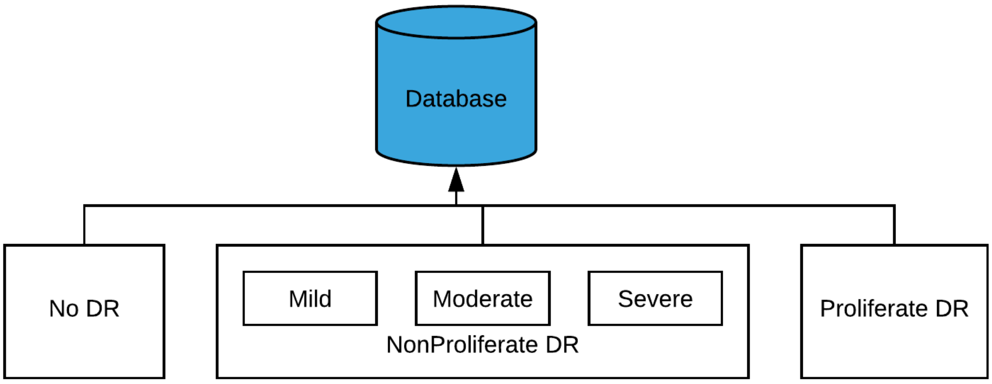


Figure 3.4: the third dataset splitting.

3.2.2 Preprocessing

Preprocessing phase is one crucial phase in image classification tasks, images needs to be looked at, the shape, the size, the noises and the pixels are preliminary things, and for

that we use gaussian filtering to do all this work.
Our image size is (244, 244, 3) the first two values refers to the width and the height of the image, and the third one refers to the image channels , meaning in this case that the images are in RGB(RED,GREEN,BLUE). As we noticed our dataset isn't that big, so we need to augment our dataset.

3.2.2.1 Gaussian filtering

Gaussian filtering is used to blur images and remove noise and detail.
Gaussian filters are ideal to start experimenting with filtering because their design can be controlled by manipulating just one variable the variance.
Gaussian filter function is defined in the equation 3.1 below:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.1}$$

Where μ= mean, with $\mu = 0$, σ^2 = variance, σ is the standard deviation of the gaussian function and (x,y) are the Cartesian coordinates of the image.

Marr and Hildreth suggested that by applying gaussian filters of different scales (σ) to an image; a set of images with different measures of smoothness can be acquired.
The 2D gaussian filter is mostly implemented as an image preprocessing step for image softening and noise reduction [69].

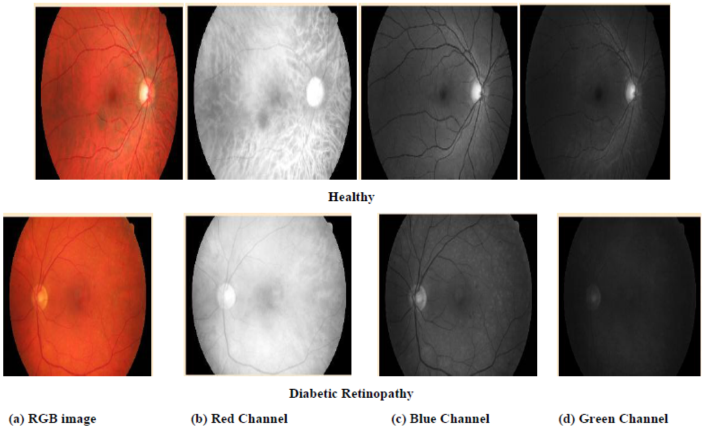


Figure 3.5: Colour Retinal Images [69].

The blood vessels in the fundus images are enhanced by applying Gaussian derivative filter as shown in figure 3.6 successively by varying sigma.

In order to obtain the enhanced blood vessels, the maximum frequency response is chosen from the responses of applied Gaussian filters.

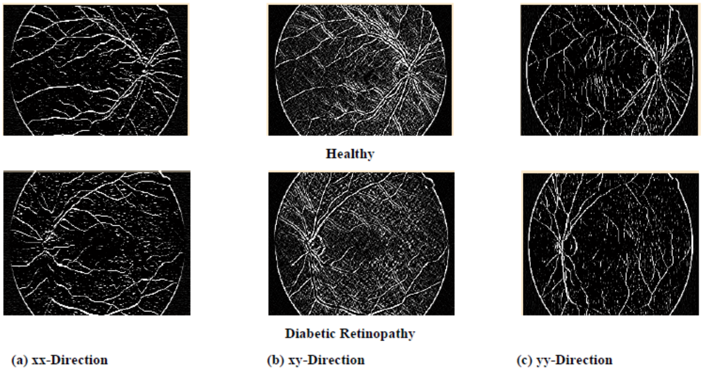


Figure 3.6: Derivative Filters [69].

3.2.2.2 Data augmentation

Data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.
We will focus on 3 main types of data augmentation techniques for image data, specifically:

- **Image flips :**An image flip means reversing the rows or columns of pixels in the case of a vertical or horizontal flip respectively.

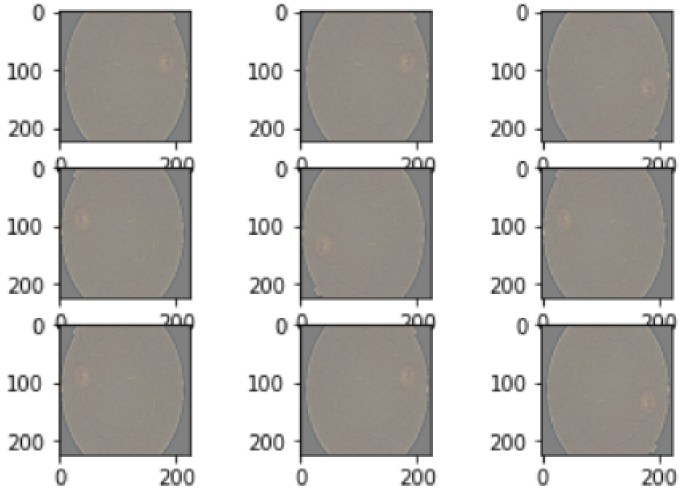


Figure 3.7: The flip augmentation.

- **Image rotations :**A rotation augmentation randomly rotates the image clockwise by a given number of degrees from 0 to 360.

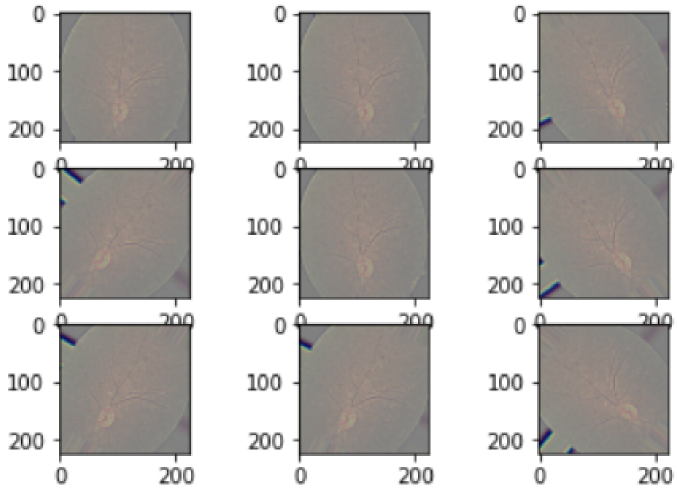


Figure 3.8: The rotations augmentation.

- **Image brightness :** The brightness of the image can be augmented by either randomly darkening images, brightening images, or both.

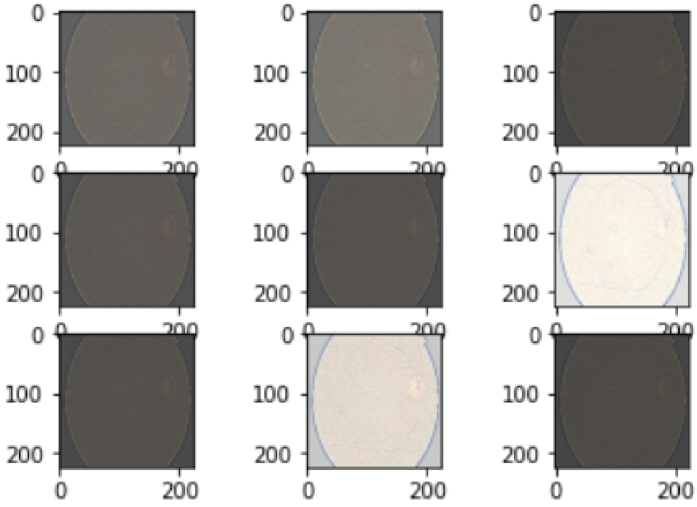


Figure 3.9: The brightness augmentation.

3.2.2.3 Data splitting

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset.
The second subset is not used to train the model, instead the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the validation dataset.
Our dataset is splitted into 2 part with only 20% images per class for validation and the rest for training.

3.2.3 Cnn learning

An overview of a convolutional neural network (CNN) training process. A CNN is composed of a stacking of several building blocks: convolution layers, pooling layers and fully connected layers.

A model's performance under particular kernels and weights is calculated with a loss function through forward propagation on a training dataset, and learnable parameters are updated according to the loss value through backpropagation with gradient descent optimization algorithm [70], as we will see in the figure 3.10.

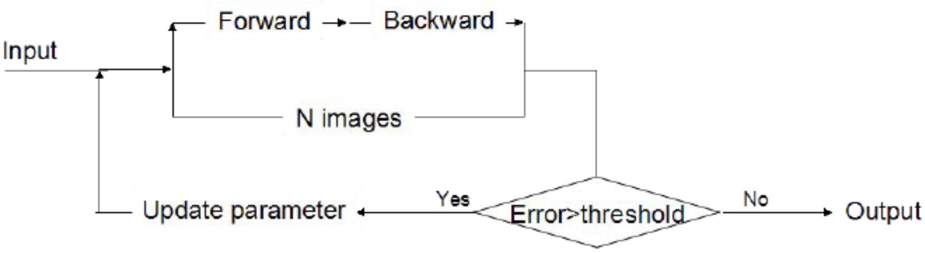


Figure 3.10: The serial algorithm of training [71].

3.2.3.1 Optimization in CNN learning

The choice of the optimizer in CNN learning model can improve the results of the models and reduces the timing from days to hours or minutes.

The objective of all optimizers is to reach the global minimum where the cost function attains the least possible value. The optimizer in the neural network is a parameter that could make the difference between algorithm converging or exploding.

There are a considerable number of optimizers, here's the most common : Gradient descent optimizer, Stochastic gradient descent, Mini batch gradient descent, RMSprop optimizer and Adam optimizer.

3.2.3.2 Model parameters

When training and fitting the model to a given dataset, some parameters needs to be specified, here are some of them mentioned briefly.

1. **The learning rate:** Is introduced as a constant (usually very small), in order to force the weight to get updated very smoothly and slowly (to avoid big steps and chaotic behaviour).

   If the error keeps going worse or oscillates wildly, reduce the learning rate.

   If the error is falling fairly consistently but slowly, increase the learning rate.

   Turn down the learning rate when the error stops decreasing.

2. **Epoch:** One epoch is when the entire dataset is passed forward and backward through the neural network only once. Epoch parameters refers to how many times the models aims to use the complete dataset for training.

3. **Batch:** While training, the dataset is divided to a number of batches/parts, big mini-batches are more computationally efficient.

4. **Batch size:** Total number of training examples present in a single batch.

5. **loss functions:** Is a performance metric on how well the neural net manages to reach its goal of generating outputs as close as possible to the desired values loss, we calculate it with the equation 3.2 below:

$$loss = \sum [Desiredoutput - actualoutput]^2 \tag{3.2}$$

where the robustness of model increases along with the decrease of the value of loss function. The goal of the model is to minimize this loss.

3.2.4 Cnn Models

The propopsed deep learning approaches are given as experiments, meaning we propose an architecture then we enhance it by changing in the hyperparameters then evaluate the different results using the metrics.
We propose here 3 convolutional neural networks architectures :AlexNet, ResNet and EfficientNet.

3.2.4.1 Alexnet

Alexnet is the name of a convolutional neural network (CNN), designed by Alex Krizhevsky [72].
Alexnet, which employed an 8-layer CNN, won the ImageNet Large Scale Visual Recognition Challenge 2012 by a phenomenally large margin. This network showed, for the first time, that the features obtained by learning can transcend manually-designed features, breaking the previous paradigm in computer vision [51].

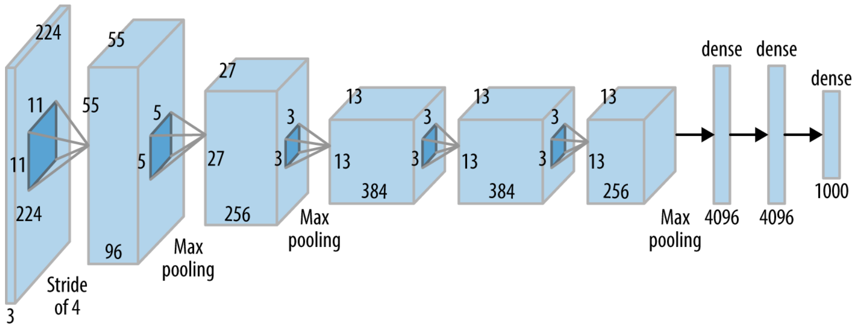


Figure 3.11: Alexnet architecture [73].

As we noticed in the figure 3.11 This alexnet contained eight layer, the first five were convolutional layers, some of them followed by max pooling layers and the last three were fully connected layers. It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid.

Advantages:

- The unique advantage of Alexnet is the directly image input to the classification model.
- The convolution layers can automatically extract the edges of the images, and fully connected layers learning these features.
- The complexity of visual patterns can be effective extracted by adding more conv layers [74].

Disadvantages:

- Alexnet is not deep enough compared to the later model such as Vggnet and Resnet.
- The use of large convolution filters (5*5) is not encouraged .
- Use normal distribution to initiate the weights in the neural networks cannot effective solve the problem of gradient vanishing.
- The performance is surpassed by more complex models such as ResNet [74].

3.2.4.2 Resnet

The residual neural network (Resnet) models, which are based on deep architectures that have shown good convergence behaviors and compelling accuracy, were developed by He et al [52].

Based on this, they won first place in the ILSVRC and common objects in context (COCO) classification challenge in 2015.

Resnet was built by several stacked residual units and developed with many different numbers of layers: 18, 34, 50, 101, 152 and 1202. However, the number of the operations can be varied depending on the different architectures [52]. For all of the above, the residual units are composed of convolutional, pooling, and layers. Resnet is similar to Vggnet [75], but Resnet is about eight times deeper than Vggnet [76].

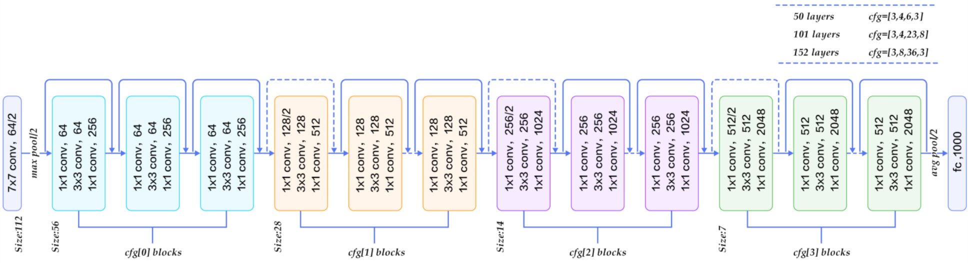


Figure 3.12: Resnet architecture [77].

We use Resnet 50 that contains 49 convolutional layers and a fully-connected layer at the end of the network.

Advantages:

- Accelerate the speed of training of the deep networks.
- Instead of widen the network, increasing depth of the network results in less extra parameters.
- Reducing the effect of Vanishing gradient problem.
- Obtaining higher accuracy in network performance .

Disadvantages:

- Optimization difficulty .
- Vanishing gradients, in some cases some neuron can die in the training [78].

3.2.4.3 Efficientnet

Efficientnets [53] are a family of neural network architectures released by Google in 2019 that have been designed by an optimization procedure that maximizes the accuracy for a given computational cost.
Efficientnets are recommended for classification tasks, since they beat many other networks (like DenseNet, Inception and Resnet) on the ImageNet benchmark, while running significantly faster.

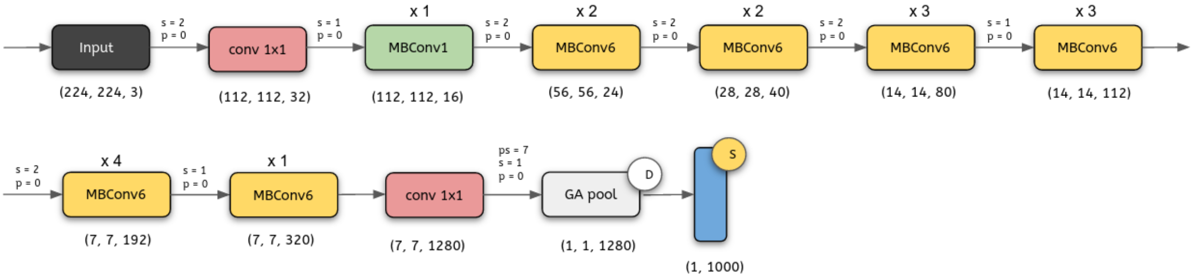


Figure 3.13: Efficientnet-B0 architecture [79].

The Efficientnet-B0 architecture wasn't developed by engineers but by the neural network itself.

They developed this model using a multi-objective neural architecture search that optimizes both accuracy and floating-point operations.
Taking B0 as a baseline model, the authors developed a full family of Efficientnets from B1 to B7 which achieved state of the art accuracy on ImageNet while being very efficient to its competitors [80].

Advantages:

- Enables the use of deep learning on mobile and other edge devices.
- Reduces compute cost, battery usage and also training and inference speeds.
- Higher accuracy than many known models.

Disadvantages:

- The network may prone to overfitting and learns a suboptimal solution.
- Negligible loss of classification accuracy [81].

3.2.5 Prediction

After the learning phase on the dataset and while the training no matter what a best model results have been achieved we save our model for image classification.
Then we use the model to predict the most likely class of the image .
Before the prediction we must consider the size, the shape and the channels of the images are the same with ours training samples or not, if not we must normalize them.
All this is summarized in the following figure 3.14.

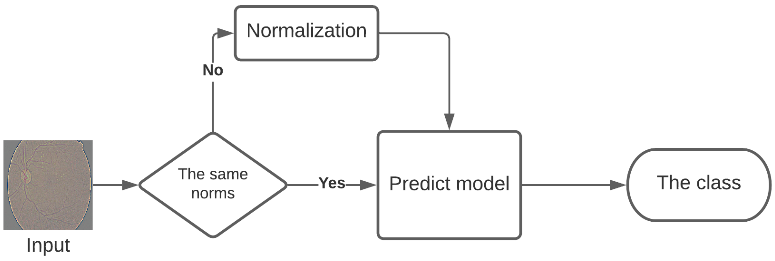


Figure 3.14: Block diagram of the prediction .

3.3 Conclusion

The objective of this chapter was to present the general and detailed design of our diabetic retinopathy classification system using deep learning .
According to the different dataset and CNN architectures we used, we could deduce in the next chapiter how to implement diabetic retinopathy detection architectures and which one is better for our datasets by discussion the experiments and results.

Chapter 4

Implementation and results

4.1 Introduction

After having detailed in the previous chapter as well as the phases of our system. The objective of this chapter is to implement the steps proposed to achieve the system designed . we start in the first part, by specifying the frameworks and tools used in development. Subsequently, we will explain all the experiments that we applied to the proposed methods and the results obtained. Finally, we are interested in presenting some parts of the developed code.

4.2 Implementation frameworks and tools

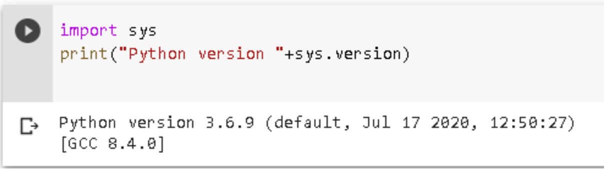
Deep learning frameworks are very powering and facilitating the artificial intelligence revolution since they provide pre implemented tools are templates for preprocessing, classification and evaluation. Without these tools it would be very hard for scientists to work on deep learning tasks.

This subsection reviews the different development frameworks and tools used in this thesis.

4.2.1 Python

Most used and famous programming language in datascience, is a high-level programming language, and its core design philosophy is all about code readability and a syntax which allows programmers to express concepts in a few lines of code. Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. It is used successfully in thousands of real-world business applications around the world, including many large and mission critical systems.

The python version used in this work is 3.6.9 [82].



```
import sys
print("Python version "+sys.version)
```

```
Python version 3.6.9 (default, Jul 17 2020, 12:50:27)
[GCC 8.4.0]
```

Figure 4.1: python version used in colab.

4.2.2 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. Matplotlib was originally written by John D. Hunter.
Everything in matplotlib is organized in a hierarchy. At the top of the hierarchy is the matplotlib "state-machine environment" which is provided by the matplotlib.pyplot module. At this level, simple functions are used to add plot elements (lines, images, text, etc.) to the axes in the figure.
Pyplot's state-machine environment behaves similarly to MATLAB and should be most familiar to users with MATLAB experience [83].

Figure 4.2: Exemples of matplotlib works[83].

4.2.3 Tensorflow

An open source machine learning library for research and production developed by Google brain team , with Python and C++ programming languages as backend, and known as the most used frameworks for deep learning tasks and with the highest activity in github. At a high level, TensorFlow is a Python library that allows users to express arbitrary computation as a graph of data flows [84].

4.2.4 Keras

Keras is an API designed for human beings, not machines . Keras follows best practices for reducing cognitive load: it offers consistent simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error. Keras is also a favorite among deep learning researchers, coming in 2nd in terms of mentions in scientific papers after tensorflow [85].

Keras uses as Backend tensorflow and Theano, we have used Keras with tensorflow backend in the implimentation part of this work.

4.2.5 Kaggle

Kaggle is an online community of data scientists and machine learners, owned by Google LLC. Kaggle got its start by offering machine learning competitions and now also offers a public data platform, a cloud-based workbench for data science, and short form AI education. Kaggle Kernels is a cloud computational environment that enables reproducible and collaborative analysis. Kernels supports scripts in R and Python, Jupyter Notebooks, and RMarkdown reports. Kaggle Kernels provide a free Cloud GPU NVidia K80 GPUs and 16 GB of Free RAM, the session is limited to 6 hours [86].
Our current work was inspired from a kaggle challenge announced lately in 2019 called APTOS 2019 blindness detection [68].

4.2.6 Google Colab

Google Colab is a free cloud service with free GPU and TPU 2 with a limit of 12 hours per session and 12 GB of RAM limits.
The execution of deep models with complicated architecture and huge data (as in our case) can lead to the use of extreme amount of hardware materials, Google Colabs provide a free GPU, currently they provide GPU with specs Tesla K80 , having 2496 CUDA cores, compute 3.7, 12GB (11.439GB Usable) GDDR5 VRAM. and a CPU of CPU: 1xsingle core hyper threaded i.e(1 core, 2 threads) Xeon Processors @2.3Ghz (No Turbo Boost) , 45MB Cache [87].

Why we need GPU in deep learning ?
Graphics processing unit, is a processor that is good at handling specialized computations.This is in contrast to a central processing unit (CPU), which is a processor that is good at handling general computations. CPUs are the processors that power most of the typical computations on our electronic devices. A GPU can be much faster at computing than a CPU. However, this is not always the case.
The speed of a GPU relative to a CPU depends on the type of computation being performed. The type of computation most suitable for a GPU is a computation that can be done in parallel. Parallel computing is a type of computation where by a particular computation is broken into independent smaller computations that can be carried out simultaneously.
The resulting computations are then recombined, or synchronized, to form the result of the original larger computation. The number of tasks that a larger task can be broken into depends on the number of cores contained on a particular piece of hardware. Cores are

the units that actually do the computation within a given processor, and CPUs typically have four, eight, or sixteen cores while GPUs have potentially thousands [88].

4.2.7 Transfer learning

Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [89]. transfer learning is the use of pretrained models in image classification tasks, has increased lately, due to the speed while training and the efficiency these models provides, this is due to the pretrained weights. in this work we will try some pretrained models to classify our images and see if these models will give better results.

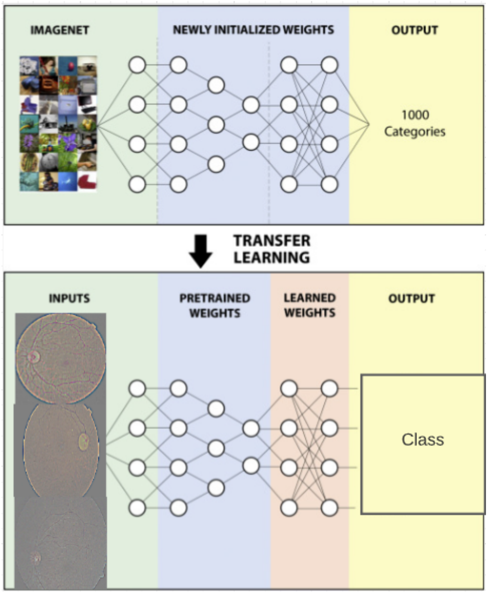


Figure 4.3: Transfer learning on our datasets.

4.2.8 Model evaluation metrics

This subsection discuses how we can evaluate our model results, what makes a model better than another and how we can predict that our model will give good results in the testing .
Many evaluation metrics to evaluate classification output quality.

4.2.8.1 Accuracy

Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives and false negatives, as it summarized in the equation 4.1.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.1}$$

A true positive or true negative is a data point that the algorithm correctly classified as true or false.
A false positive or false negative, on the other hand, is a data point that the algorithm incorrectly classified.

4.2.8.2 Recall

Named also Sensitivity, Recall is the ratio of correctly predicted positive observations to the all observations in actual class, in other way, recall measures the proportion of actual positives that are correctly identified.
the formula:

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

4.2.8.3 Precision

Out of all the classes, how much we predicted correctly. It should be high as possible.
the formula:

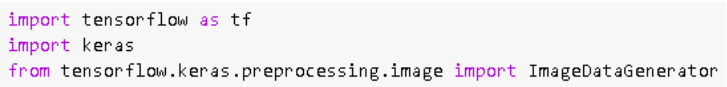
$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

4.3 Implementation phases

As we see in the global system design, the implementation has several phases .

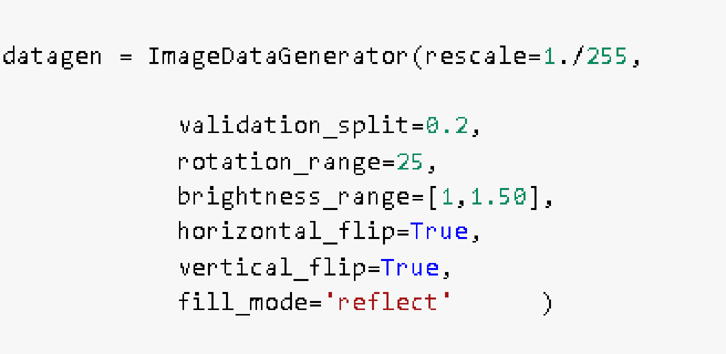
4.3.1 Loading and preprocessing the datasets

Datasets are available in multiple folders ,Each folder represent a class of images. We use the ImageDataGenerator method of the preprocessing sublibrary of keras library to do the preprocessing .
To load the data we use flow_from_directory from the same sublibary of ImageDataGenerator . before we use this two methods we must import tensorflow, keras, and preprocessing sublibrary.



```
import tensorflow as tf
import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Figure 4.4: Importing tensorflow, keras, and preprocessing sublibrary code.



```
datagen = ImageDataGenerator(rescale=1./255,

            validation_split=0.2,
            rotation_range=25,
            brightness_range=[1,1.50],
            horizontal_flip=True,
            vertical_flip=True,
            fill_mode='reflect'      )
```

Figure 4.5: ImageDataGenerator code.

Image data augmentation create transformed versions of images in the dataset that belong to the same class as the original image.
Arguments:

- **rescale:** rescaling factor. Defaults to None. If None or 0, no rescaling is applied, otherwise we multiply the data by the value provided (after applying all other transformations).

- **validation_split:** Float. Fraction of images reserved for validation (strictly between 0 and 1) and we will reserve for validation 20 % of dataset.

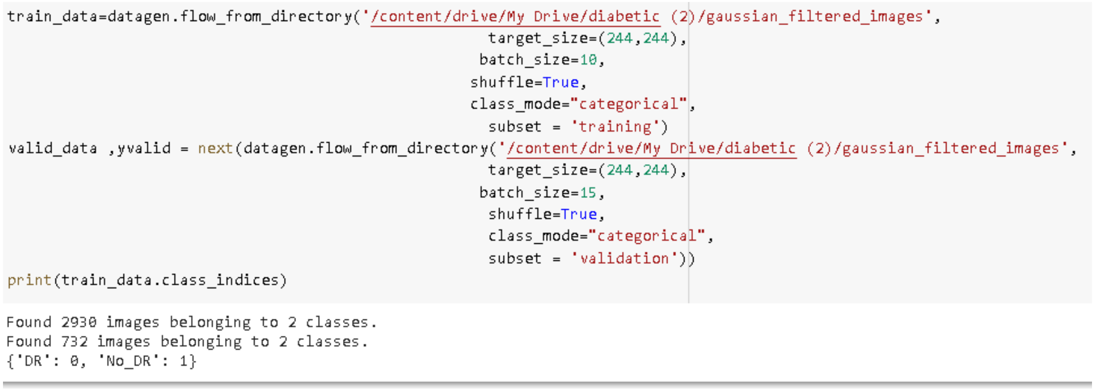
- **rotation_range:** Int. Degree range for random rotations.

- **brightness_range:** Tuple or list of two floats. Range for picking a brightness shift value from.

- **horizontal_flip:** Boolean. Randomly flip inputs horizontally.

- **vertical_flip:** Boolean. Randomly flip inputs vertically.

- **fill_mode:** One of ”constant”, ”nearest”, ”reflect” or ”wrap”. Default is ’nearest’. Points outside the boundaries of the input are filled according to the given mode: - ’constant’: kkkkkkkk—abcd—kkkkkkkk (cval=k) - ’nearest’: aaaaaaaa—abcd—dddddddd - ’reflect’: abcddcba—abcd—dcbaabcd - ’wrap’: abcdabcd—abcd—abcdabcd [90].



```
train_data=datagen.flow_from_directory('/content/drive/My Drive/diabetic (2)/gaussian_filtered_images',
                                                     target_size=(244,244),
                                                    batch_size=10,
                                                   shuffle=True,
                                                   class_mode="categorical",
                                                     subset = 'training')
valid_data ,yvalid = next(datagen.flow_from_directory('/content/drive/My Drive/diabetic (2)/gaussian_filtered_images',
                                                     target_size=(244,244),
                                                    batch_size=15,
                                                     shuffle=True,
                                                     class_mode="categorical",
                                                     subset = 'validation'))
print(train_data.class_indices)
```

```
Found 2930 images belonging to 2 classes.
Found 732 images belonging to 2 classes.
{'DR': 0, 'No_DR': 1}
```

Figure 4.6: flow_from_directory code.

Takes the path to a directory generates batches of augmented data.

Arguments:

- **directory:** string, path to the target directory. It should contain one subdirectory per class. And images inside each of the subdirectories directory tree will be included in the generator.

- **target_size:** Tuple of integers (height, width), defaults to (256, 256). The dimensions to which all images found will be resized.

- **class_mode:** One of "categorical", "binary", "sparse", "input" or None. We use "categorical". The type of label arrays that are returned will be 2D one-hot encoded labels.

- **batch_size:** Size of the batches of data (default: 32).

- **shuffle:** to shuffle the data we put True, and False sorts the data in alphanumeric order.

- **subset:** Subset of data ("training" or "validation") used only if validation_split is set in ImageDataGenerator.

After we call the function automatically print the number of images on the dataset and the subsets beside the classes .

We add the command print(train_data.class_indices) to show the classes and indices.

We do the same to the other datasets and we get this final structure as you will see in the table 4.1, the table 4.3 and the table 4.3 .

	Training	**Validation**	**Total**
No_DR	1 444	361	1805
Mild	296	74	370
Moderate	799	200	999
Severe	155	38	193
Proliferate_DR	236	59	295
Total	2930	732	3662

Table 4.1: the original Dataset structure.

	Training	**Validation**	**Total**
No_DR	1 740	435	2175
DR	1190	297	1487
Total	2930	732	3662

Table 4.2: The second Dataset structure.

	Training	**Validation**	**Total**
No_DR	1 740	435	2175
Nonproliferative	954	238	1192
Proliferate_DR	236	59	295
Total	2930	732	3662

Table 4.3: The third Dataset structure.

After Loading and preprocessing, the inputs data is ready for our CNN models.

4.3.2 Creating the models

In this section we will demonstrate how we applied deep learning to classify and identify the diabetic retinopathy, we use Alexnet8, Resnet50 and Efficientnet-B0 because they are the most famous good performance architectures when dealing with images, we done many experiments, and we changed the hyperparameters to obtained the best results.
Before we start the creation of any model, there are two type of models in keras.
Sequential API: It allows us to create models layer by layer in sequential order. But it does not allow us to create models that have multiple inputs or outputs.
It is best for simple stack of layers which have 1 input tensor and 1 output tensor.
Functional API : It provides more flexibility to define a model and add layers in keras. Functional API allows us to create models that have multiple input or output. It also allows us to share these layers.

In other words, we can make graphs of layers using Keras functional API [91].

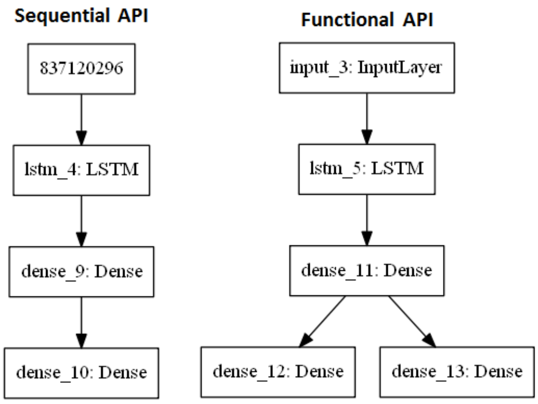
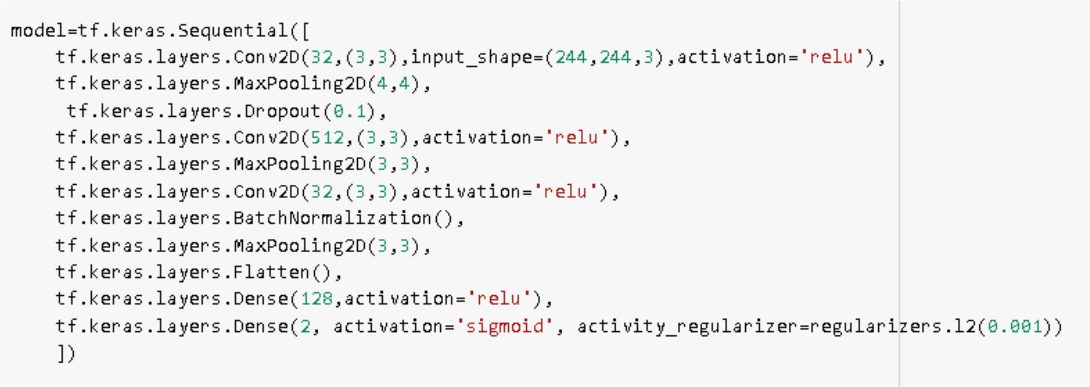


Figure 4.7: Keras sequential vs functional[92].

4.3.2.1 Alexnet8

the first CNN experiment architecture is AlexNet 8 ,composed of 3 Convolution layers, 3 Max pooling layers , one fully connected layer and one Dense layer.



```
model=tf.keras.Sequential([
    tf.keras.layers.Conv2D(32,(3,3),input_shape=(244,244,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(4,4),
     tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Conv2D(512,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(3,3),
    tf.keras.layers.Conv2D(32,(3,3),activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(3,3),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128,activation='relu'),
    tf.keras.layers.Dense(2, activation='sigmoid', activity_regularizer=regularizers.l2(0.001))
    ])
```

Figure 4.8: Alexnet implementation code.

we use sequential API to implement Alexnet 8 by calling the function tf.keras.Sequential, then we add the layers one by one until we get to the output layer called dense layer.

Layers:

- **Conv2D:** 2D convolution layer ,32 is number of filters,(3,3) the kernel of filters,input_shape the shape of the input image and activation is the activation function we use.

- **MaxPooling2D:** Max pooling operation for 2D spatial data, (4, 4) will take the max value over a 4x4 pooling window in every step.

- **Dropout:** Dropout layer randomly sets input units to 0, 0.1 mean 10 % of the input units sets 0.

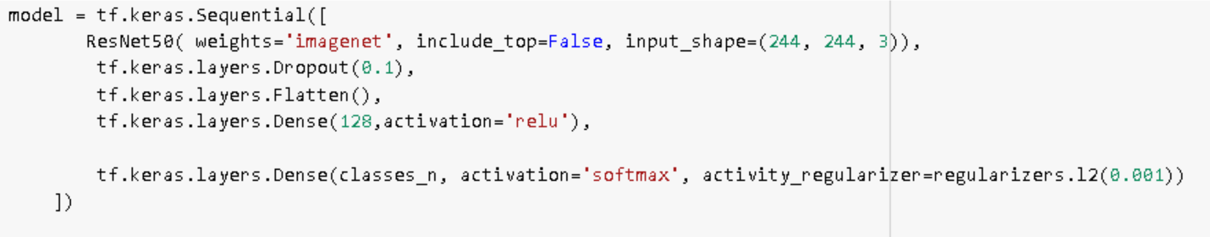
- **BatchNormalization:**Normalize and scale inputs or activations.

- **Flatten:** Reshapes the tensor to have a shape that is equal to the number of elements contained in the tensor.

- **Dense:** Is a Layer in which each input neuron is connected to the output neuron, like a Simple neural net, the parameters 128 is the dimensionnality of the output, activation is the activation function we use and activity_regularizer to add weights .

4.3.2.2 Use pretrained model

We tuned ResNet50 model by removing the top three layers, and changed the ouput layer to have neurons corresponding to our classes. We made this changing because the dense layers are responsible for classification not learning features, so we keep the conv and the pooling layers and we drop the rest.



```
model = tf.keras.Sequential([
       ResNet50( weights='imagenet', include_top=False, input_shape=(244, 244, 3)),
        tf.keras.layers.Dropout(0.1),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128,activation='relu'),

        tf.keras.layers.Dense(classes_n, activation='softmax', activity_regularizer=regularizers.l2(0.001))
    ])
```

Figure 4.9: Resnet50 implementation code.

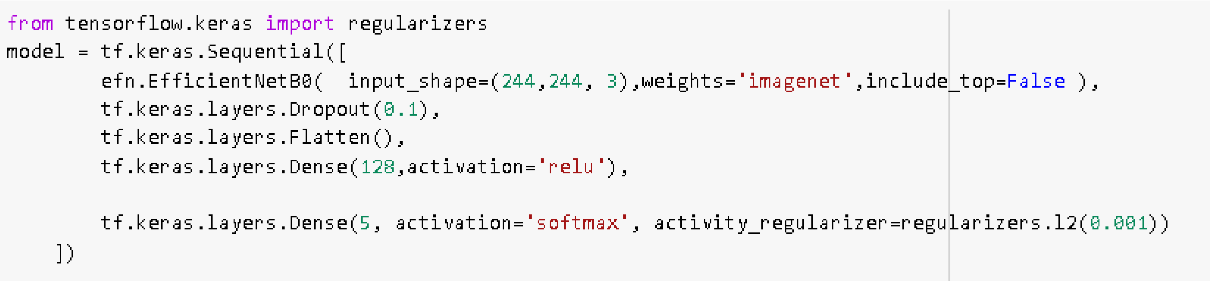
ResNet50 arguments:

- **input_shape:** Is the shape of the input images.

- **include_top:** whether to include the fully-connected layer at the top of the network.

- **weights:** None (random initialization), 'imagenet' (pre-training on ImageNet), or the path to the weights file to be loaded.

For the other layer we explained them in the previous subsection 4.3.2.1. we do the same with Efficientnet-B0 as we will see in the next figure 4.10.



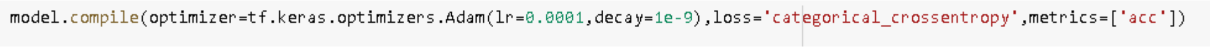
```
from tensorflow.keras import regularizers
model = tf.keras.Sequential([
        efn.EfficientNetB0(  input_shape=(244,244, 3),weights='imagenet',include_top=False ),
        tf.keras.layers.Dropout(0.1),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128,activation='relu'),

        tf.keras.layers.Dense(5, activation='softmax', activity_regularizer=regularizers.l2(0.001))
    ])
```

Figure 4.10: EfficientNetB0 implementation code.

4.3.3 Training the models

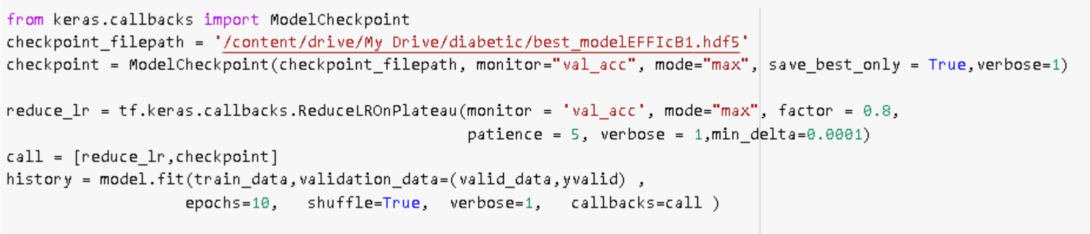
The model is compiled with categorical cross entropy loss function and the adam optimizer is used. The 'accuracy' metric is used to evaluate the model.
Training the model using a Google colab's GPU speeds up the training process,We set the number of epochs to 10, and use a regularization method Reduce learning rate .

In our case, when a metric has stopped improving. Models often benefit from reducing the learning rate by a factor of learning stagnates.
Model checkpoint is used to save the model. The monitor parameter allow us to set a metric which we want to keep an eye on. In our case, we only save the model when the validation accuracy is the max. We save the best model .
Note : All our experiments have been compiled with these parameters.



```
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.0001,decay=1e-9),loss='categorical_crossentropy',metrics=['acc'])
```

Figure 4.11: Compiling code.



```
from keras.callbacks import ModelCheckpoint
checkpoint_filepath = '/content/drive/My Drive/diabetic/best_modelEFFIcB1.hdf5'
checkpoint = ModelCheckpoint(checkpoint_filepath, monitor="val_acc", mode="max", save_best_only = True,verbose=1)

reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor = 'val_acc', mode="max", factor = 0.8,
                                                 patience = 5, verbose = 1,min_delta=0.0001)
call = [reduce_lr,checkpoint]
history = model.fit(train_data,validation_data=(valid_data,yvalid) ,
                   epochs=10,   shuffle=True,   verbose=1,   callbacks=call )
```

Figure 4.12: Training code.

We use fit function to train the models.
Arguments of fit:

- **train_data :** List of the training data.
- **validation_data :** List of samples used to evaluate the loss and metrics for any model after any epoch has ended.
- **Epochs :** An integer and number of epochs we want to train our model for.
- **Verbose :** Specifies verbosity mode, 1 to show progress bar and details per epoch .
- **Shuffle :** Use the data randomly.

The figures and values of the training of the models and datasets will be discussed in the results section.
We use variable 'history' to save the training epochs and details of work to plot it later using matplotlib ,we will see the plotting in the results section.

4.4 Experiments and results

This section is the essence of our work, different experiments of different proposed architectures and approaches are detailed and explained. Comparisons between the experiments are illustrated in tables and graphs.
We have conducted many experiments on our datasets, and we analyzed each obtained result from each experiment by changing the deep architecture, the hyperparameters, tuned pretrained models and modified versions of these models.
We compare them based on the obtained accuracy and validation accuracy. For each dataset we applicat 3 CNN models to get the experiments.

4.4.1 First dataset

This is the original dataset with 5 classes (NO_DR, Mild, Moderate, severe and Proliferate_DR).
We have trained three CNN models , and we compared the results obtained of each model based on the accuracy, validation accuracy and training time.

4.4.1.1 Alexnet8

We use alexnet8 for the first experiment after 2 minutes of training using google colabs GPU, we obtained these results .

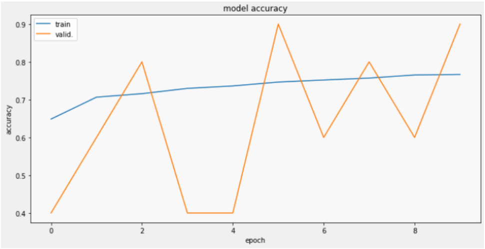


Figure 4.13: The model accuracy obtained by the alexnet8

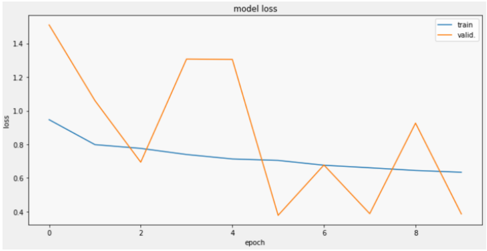
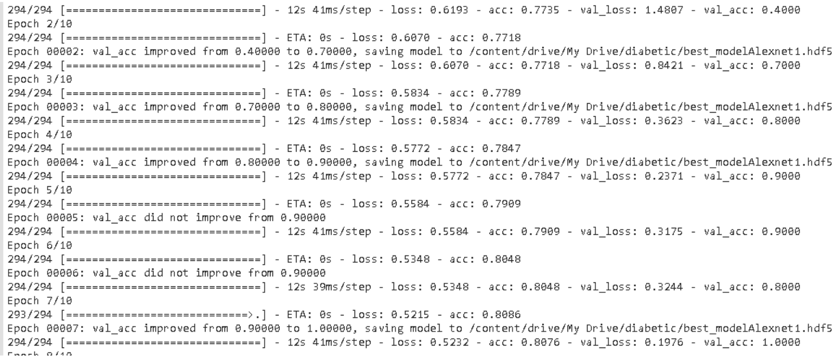


Figure 4.14: The model loss obtained by the alexnet8

The blue line is showing how well the model is learning by each epoch(full iteration on the dataset) and the orange line is the learning curve calculated from a hold-out validation dataset that gives an idea of how well the model is generalizing, in this case the model is doing fine since the training accuracy is 80,08% and the validation accuracy 100%.
As we said before our model will be saved with the best validation accuracy, this figure 4.15 will show which one the model is saved.



```
294/294 [==============================] - 12s 41ms/step - loss: 0.6193 - acc: 0.7735 - val_loss: 1.4807 - val_acc: 0.4000
Epoch 2/10
294/294 [==============================] - ETA: 0s - loss: 0.6070 - acc: 0.7718
Epoch 00002: val_acc improved from 0.40000 to 0.70000, saving model to /content/drive/My Drive/diabetic/best_modelAlexnet1.hdf5
294/294 [==============================] - 12s 41ms/step - loss: 0.6070 - acc: 0.7718 - val_loss: 0.8421 - val_acc: 0.7000
Epoch 3/10
294/294 [==============================] - ETA: 0s - loss: 0.5834 - acc: 0.7789
Epoch 00003: val_acc improved from 0.70000 to 0.80000, saving model to /content/drive/My Drive/diabetic/best_modelAlexnet1.hdf5
294/294 [==============================] - 12s 41ms/step - loss: 0.5834 - acc: 0.7789 - val_loss: 0.3623 - val_acc: 0.8000
Epoch 4/10
294/294 [==============================] - ETA: 0s - loss: 0.5772 - acc: 0.7847
Epoch 00004: val_acc improved from 0.80000 to 0.90000, saving model to /content/drive/My Drive/diabetic/best_modelAlexnet1.hdf5
294/294 [==============================] - 12s 41ms/step - loss: 0.5772 - acc: 0.7847 - val_loss: 0.2371 - val_acc: 0.9000
Epoch 5/10
294/294 [==============================] - ETA: 0s - loss: 0.5584 - acc: 0.7909
Epoch 00005: val_acc did not improve from 0.90000
294/294 [==============================] - 12s 41ms/step - loss: 0.5584 - acc: 0.7909 - val_loss: 0.3175 - val_acc: 0.9000
Epoch 6/10
294/294 [==============================] - ETA: 0s - loss: 0.5348 - acc: 0.8048
Epoch 00006: val_acc did not improve from 0.90000
294/294 [==============================] - 12s 39ms/step - loss: 0.5348 - acc: 0.8048 - val_loss: 0.3244 - val_acc: 0.8000
Epoch 7/10
293/294 [============================>.] - ETA: 0s - loss: 0.5215 - acc: 0.8086
Epoch 00007: val_acc improved from 0.90000 to 1.00000, saving model to /content/drive/My Drive/diabetic/best_modelAlexnet1.hdf5
294/294 [==============================] - 12s 41ms/step - loss: 0.5232 - acc: 0.8076 - val_loss: 0.1976 - val_acc: 1.0000
```

Figure 4.15: Training process for Alexnet8.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 12 seconds, the model is validated at the end of each epoch.

4.4.1.2 Resnet50

We use Resnet50 for the second experiment after 4.66 minutes of training using google colabs GPU, we obtained these results .

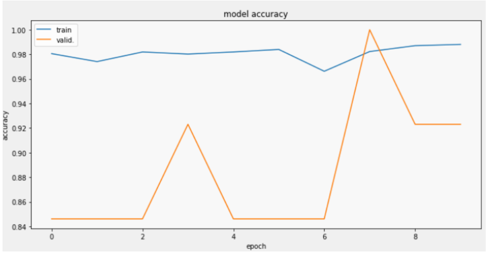


Figure 4.16: The model accuracy obtained by the Resnet50

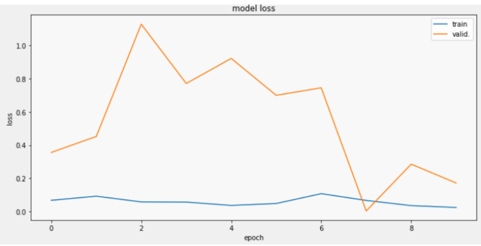
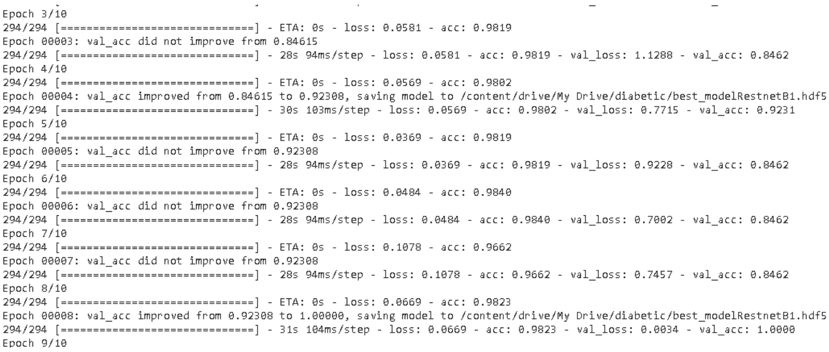


Figure 4.17: The model loss obtained by the Resnet50

In this case Resnet50 is better then alexnet8 since the training accuracy is 98,23% and the validation accuracy 100%.
This figure 4.18 will show when the best model is saved.



```
Epoch 3/10
294/294 [==============================] - ETA: 0s - loss: 0.0581 - acc: 0.9819
Epoch 00003: val_acc did not improve from 0.84615
294/294 [==============================] - 28s 94ms/step - loss: 0.0581 - acc: 0.9819 - val_loss: 1.1288 - val_acc: 0.8462
Epoch 4/10
294/294 [==============================] - ETA: 0s - loss: 0.0569 - acc: 0.9802
Epoch 00004: val_acc improved from 0.84615 to 0.92308, saving model to /content/drive/My Drive/diabetic/best_modelRestnetB1.hdf5
294/294 [==============================] - 30s 103ms/step - loss: 0.0569 - acc: 0.9802 - val_loss: 0.7715 - val_acc: 0.9231
Epoch 5/10
294/294 [==============================] - ETA: 0s - loss: 0.0369 - acc: 0.9819
Epoch 00005: val_acc did not improve from 0.92308
294/294 [==============================] - 28s 94ms/step - loss: 0.0369 - acc: 0.9819 - val_loss: 0.9228 - val_acc: 0.8462
Epoch 6/10
294/294 [==============================] - ETA: 0s - loss: 0.0484 - acc: 0.9840
Epoch 00006: val_acc did not improve from 0.92308
294/294 [==============================] - 28s 94ms/step - loss: 0.0484 - acc: 0.9840 - val_loss: 0.7002 - val_acc: 0.8462
Epoch 7/10
294/294 [==============================] - ETA: 0s - loss: 0.1078 - acc: 0.9662
Epoch 00007: val_acc did not improve from 0.92308
294/294 [==============================] - 28s 94ms/step - loss: 0.1078 - acc: 0.9662 - val_loss: 0.7457 - val_acc: 0.8462
Epoch 8/10
294/294 [==============================] - ETA: 0s - loss: 0.0669 - acc: 0.9823
Epoch 00008: val_acc improved from 0.92308 to 1.00000, saving model to /content/drive/My Drive/diabetic/best_modelRestnetB1.hdf5
294/294 [==============================] - 31s 104ms/step - loss: 0.0669 - acc: 0.9823 - val_loss: 0.0034 - val_acc: 1.0000
Epoch 9/10
```

Figure 4.18: Training process for Resnet50.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 28 seconds, the model is validated at the end of each epoch.

4.4.1.3 Efficentnet-B0

We use Efficentnet-B0 for the second experiment after 18,83 minutes of training using google colabs GPU, we obtained these results .

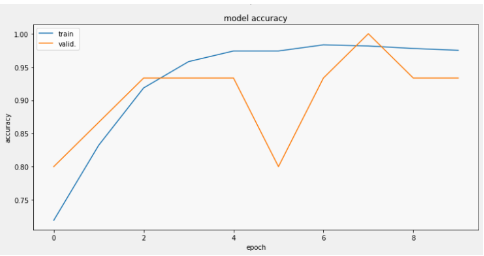


Figure 4.19: The model accuracy obtained by the Efficentnet-B0

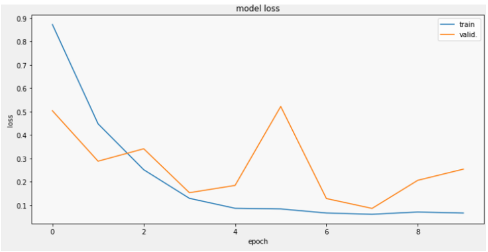
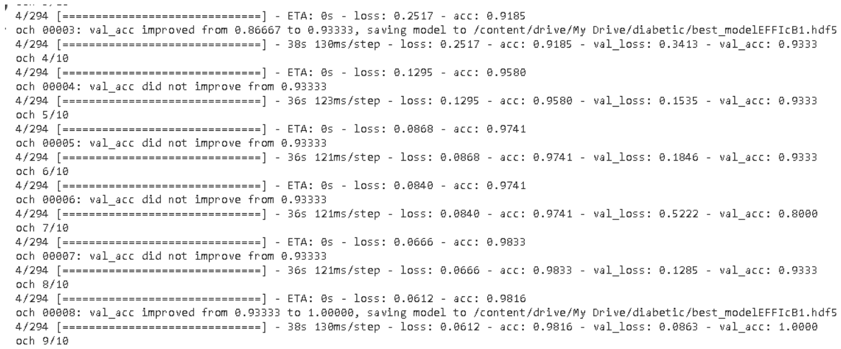


Figure 4.20: The model loss obtained by the Efficentnet-B0

In this case the Efficentnet-B0 is greater then Alexnet8 but not then Resnet50 since the training accuracy is 98,16% and the validation accuracy 100%.
This figure 4.21 will show when the best model is saved.



```
4/294 [==============================] - ETA: 0s - loss: 0.2517 - acc: 0.9185
och 00003: val_acc improved from 0.86667 to 0.93333, saving model to /content/drive/My Drive/diabetic/best_modelEFFIcB1.hdf5
4/294 [==============================] - 38s 130ms/step - loss: 0.2517 - acc: 0.9185 - val_loss: 0.3413 - val_acc: 0.9333
och 4/10
4/294 [==============================] - ETA: 0s - loss: 0.1295 - acc: 0.9580
och 00004: val_acc did not improve from 0.93333
4/294 [==============================] - 36s 123ms/step - loss: 0.1295 - acc: 0.9580 - val_loss: 0.1535 - val_acc: 0.9333
och 5/10
4/294 [==============================] - ETA: 0s - loss: 0.0868 - acc: 0.9741
och 00005: val_acc did not improve from 0.93333
4/294 [==============================] - 36s 121ms/step - loss: 0.0868 - acc: 0.9741 - val_loss: 0.1846 - val_acc: 0.9333
och 6/10
4/294 [==============================] - ETA: 0s - loss: 0.0840 - acc: 0.9741
och 00006: val_acc did not improve from 0.93333
4/294 [==============================] - 36s 121ms/step - loss: 0.0840 - acc: 0.9741 - val_loss: 0.5222 - val_acc: 0.8000
och 7/10
4/294 [==============================] - ETA: 0s - loss: 0.0666 - acc: 0.9833
och 00007: val_acc did not improve from 0.93333
4/294 [==============================] - 36s 121ms/step - loss: 0.0666 - acc: 0.9833 - val_loss: 0.1285 - val_acc: 0.9333
och 8/10
4/294 [==============================] - ETA: 0s - loss: 0.0612 - acc: 0.9816
och 00008: val_acc improved from 0.93333 to 1.00000, saving model to /content/drive/My Drive/diabetic/best_modelEFFIcB1.hdf5
4/294 [==============================] - 38s 130ms/step - loss: 0.0612 - acc: 0.9816 - val_loss: 0.0863 - val_acc: 1.0000
och 9/10
```

Figure 4.21: Training process for Efficentnet-B0.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 36 seconds or 38 seconds, the model is validated at the end of each epoch.

4.4.1.4 Comparisons of our experiments

After we seen the three experiments, we conclude that the best model for the first dataset is resnet50 .
This table 4.4 show the comparisons of our experiments.

	Accuracy	**Validation Accuracy**	**Time**
Alexnet8	80.08%	100%	2 min
Resnet50	**98.23%**	**100%**	**4.66 min**
Efficentnet-B0	98.16%	100%	18.83 min

Table 4.4: Comparisons of our experiments on the first dataset.

4.4.2 Second dataset

The second dataset has 2 classes (NO_DR and DR).
We have trained the same CNN models, and we compared the results obtained of each model based on the accuracy, validation accuracy and training time, same we did with the first one .

4.4.2.1 Alexnet8

We use alexnet for the first experiment after 1,83 minutes of training using google colabs GPU, we obtained these results .

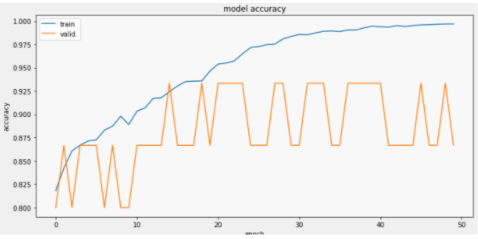


Figure 4.22: The model accuracy obtained by the alexnet8

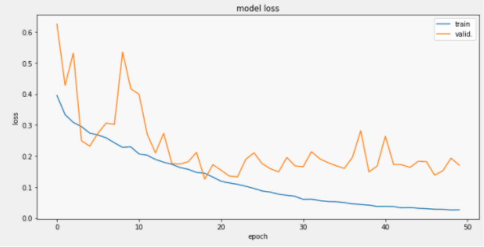
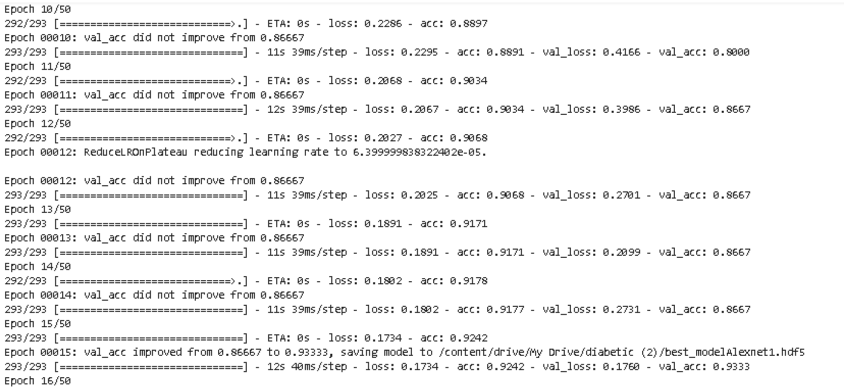


Figure 4.23: The model loss obtained by the alexnet8

In this case the model is doing good job since the training accuracy is 92,42% and the validation accuracy 93,33%.
This figure 4.24 will show when the best model is saved.



```
Epoch 10/50
292/293 [============================>.] - ETA: 0s - loss: 0.2286 - acc: 0.8897
Epoch 00010: val_acc did not improve from 0.86667
293/293 [==============================] - 11s 39ms/step - loss: 0.2295 - acc: 0.8891 - val_loss: 0.4166 - val_acc: 0.8000
Epoch 11/50
292/293 [============================>.] - ETA: 0s - loss: 0.2068 - acc: 0.9034
Epoch 00011: val_acc did not improve from 0.86667
293/293 [==============================] - 12s 39ms/step - loss: 0.2067 - acc: 0.9034 - val_loss: 0.3986 - val_acc: 0.8667
Epoch 12/50
292/293 [============================>.] - ETA: 0s - loss: 0.2027 - acc: 0.9068
Epoch 00012: ReduceLROnPlateau reducing learning rate to 6.399999838322402e-05.

Epoch 00012: val_acc did not improve from 0.86667
293/293 [==============================] - 11s 39ms/step - loss: 0.2025 - acc: 0.9068 - val_loss: 0.2701 - val_acc: 0.8667
Epoch 13/50
293/293 [==============================] - ETA: 0s - loss: 0.1891 - acc: 0.9171
Epoch 00013: val_acc did not improve from 0.86667
293/293 [==============================] - 11s 39ms/step - loss: 0.1891 - acc: 0.9171 - val_loss: 0.2099 - val_acc: 0.8667
Epoch 14/50
292/293 [============================>.] - ETA: 0s - loss: 0.1802 - acc: 0.9178
Epoch 00014: val_acc did not improve from 0.86667
293/293 [==============================] - 11s 39ms/step - loss: 0.1802 - acc: 0.9177 - val_loss: 0.2731 - val_acc: 0.8667
Epoch 15/50
293/293 [==============================] - ETA: 0s - loss: 0.1734 - acc: 0.9242
Epoch 00015: val_acc improved from 0.86667 to 0.93333, saving model to /content/drive/My Drive/diabetic (2)/best_modelAlexnet1.hdf5
293/293 [==============================] - 12s 40ms/step - loss: 0.1734 - acc: 0.9242 - val_loss: 0.1760 - val_acc: 0.9333
Epoch 16/50
```

Figure 4.24: Training process for Alexnet8.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 11 seconds, the model is validated at the end of each epoch.

4.4.2.2 Resnet50

we use Resnet50 for the second experiment after 4,66 minutes of training using google colabs GPU, we obtained these results .

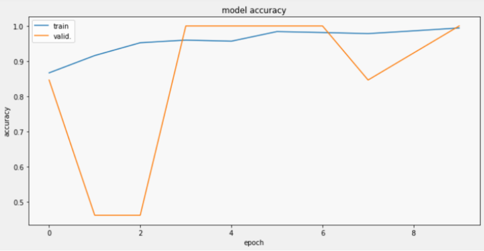


Figure 4.25: The model accuracy obtained by the Resnet50

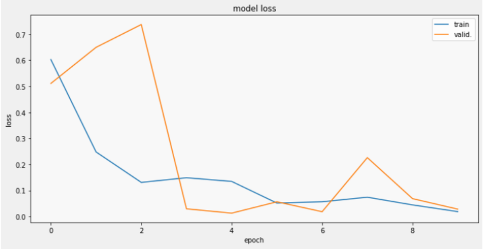
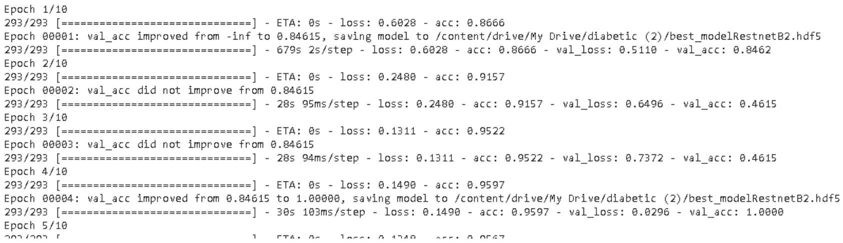


Figure 4.26: The model loss obtained by the Resnet50

In this case the Resnet50 is greater then alexnet8 since the training accuracy is 95,97% and the validation accuracy 100%.
This figure 4.27 will show when the best model is saved.



```
Epoch 1/10
293/293 [==============================] - ETA: 0s - loss: 0.6028 - acc: 0.8666
Epoch 00001: val_acc improved from -inf to 0.84615, saving model to /content/drive/My Drive/diabetic (2)/best_modelRestnetB2.hdf5
293/293 [==============================] - 679s 2s/step - loss: 0.6028 - acc: 0.8666 - val_loss: 0.5110 - val_acc: 0.8462
Epoch 2/10
293/293 [==============================] - ETA: 0s - loss: 0.2480 - acc: 0.9157
Epoch 00002: val_acc did not improve from 0.84615
293/293 [==============================] - 28s 95ms/step - loss: 0.2480 - acc: 0.9157 - val_loss: 0.6496 - val_acc: 0.4615
Epoch 3/10
293/293 [==============================] - ETA: 0s - loss: 0.1311 - acc: 0.9522
Epoch 00003: val_acc did not improve from 0.84615
293/293 [==============================] - 28s 94ms/step - loss: 0.1311 - acc: 0.9522 - val_loss: 0.7372 - val_acc: 0.4615
Epoch 4/10
293/293 [==============================] - ETA: 0s - loss: 0.1490 - acc: 0.9597
Epoch 00004: val_acc improved from 0.84615 to 1.00000, saving model to /content/drive/My Drive/diabetic (2)/best_modelRestnetB2.hdf5
293/293 [==============================] - 30s 103ms/step - loss: 0.1490 - acc: 0.9597 - val_loss: 0.0296 - val_acc: 1.0000
Epoch 5/10
293/293 [==============================] - ETA: 0s - loss: 0.1248 - acc: 0.9567
```

Figure 4.27: Training process for Resnet50.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 28 seconds, the model is validated at the end of each epoch.

4.4.2.3 Efficentnet-B0

We use Efficentnet-B0 for the second experiment after 18,83 minutes of training using google colabs GPU, we obtained these results .

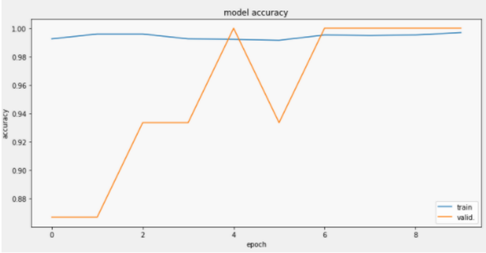


Figure 4.28: The model accuracy obtained by the Efficentnet-B0

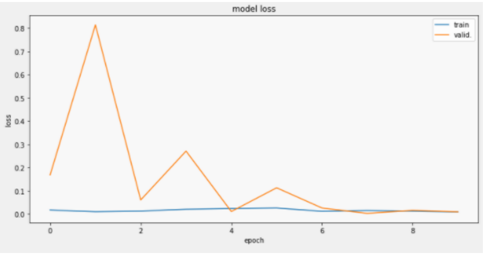
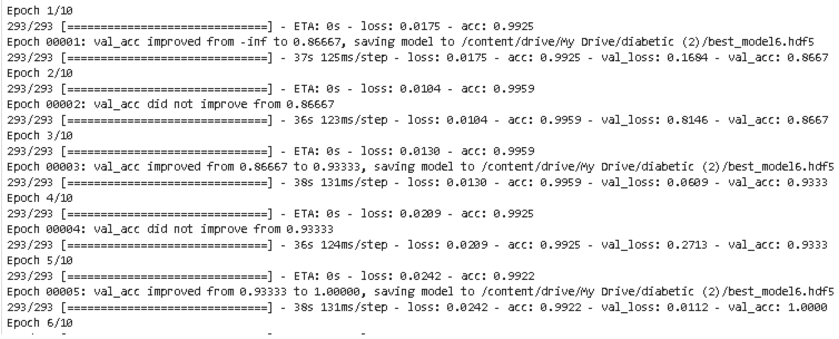


Figure 4.29: The model loss obtained by the Efficentnet-B0

In this case Efficentnet-B0 is greater then Alexnet8 and Resnet50 since the training accuracy is 99,22% and the validation accuracy 100%.
This figure 4.30 will show when the best model is saved.



```
Epoch 1/10
293/293 [==============================] - ETA: 0s - loss: 0.0175 - acc: 0.9925
Epoch 00001: val_acc improved from -inf to 0.86667, saving model to /content/drive/My Drive/diabetic (2)/best_model6.hdf5
293/293 [==============================] - 37s 125ms/step - loss: 0.0175 - acc: 0.9925 - val_loss: 0.1684 - val_acc: 0.8667
Epoch 2/10
293/293 [==============================] - ETA: 0s - loss: 0.0104 - acc: 0.9959
Epoch 00002: val_acc did not improve from 0.86667
293/293 [==============================] - 36s 123ms/step - loss: 0.0104 - acc: 0.9959 - val_loss: 0.8146 - val_acc: 0.8667
Epoch 3/10
293/293 [==============================] - ETA: 0s - loss: 0.0130 - acc: 0.9959
Epoch 00003: val_acc improved from 0.86667 to 0.93333, saving model to /content/drive/My Drive/diabetic (2)/best_model6.hdf5
293/293 [==============================] - 38s 131ms/step - loss: 0.0130 - acc: 0.9959 - val_loss: 0.0609 - val_acc: 0.9333
Epoch 4/10
293/293 [==============================] - ETA: 0s - loss: 0.0209 - acc: 0.9925
Epoch 00004: val_acc did not improve from 0.93333
293/293 [==============================] - 36s 124ms/step - loss: 0.0209 - acc: 0.9925 - val_loss: 0.2713 - val_acc: 0.9333
Epoch 5/10
293/293 [==============================] - ETA: 0s - loss: 0.0242 - acc: 0.9922
Epoch 00005: val_acc improved from 0.93333 to 1.00000, saving model to /content/drive/My Drive/diabetic (2)/best_model6.hdf5
293/293 [==============================] - 38s 131ms/step - loss: 0.0242 - acc: 0.9922 - val_loss: 0.0112 - val_acc: 1.0000
Epoch 6/10
```

Figure 4.30: Training process for Efficentnet-B0.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 36 seconds or 38 seconds, the model is validated at the end of each epoch.

4.4.2.4 Comparisons of our experiments

After we seen the three experiments,we conclude that the best model for the second dataset is Efficentnet-B0 .
This table 4.5 show the comparisons of our experiments.

	Accuracy	Validation Accuracy	Time
Alexnet8	92,42%	93,33%	1,83 min
Resnet50	95,97%	100%	4.66 min
Efficentnet-B0	**99,22%**	**100%**	**18.83 min**

Table 4.5: Comparisons of our experiments on the second dataset.

4.4.3 Third dataset

the third dataset has 3 classes (NO_DR, Nonproliferative and Proliferate_DR).
We have trained the same CNN models, and we compared the results obtained of each model based on the accuracy, validation accuracy and training time ,same we did with the first and the second datasets. Alexnet8 We use alexnet for the first experiment after 2 minutes of training using google colabs GPU , we obtained these results .

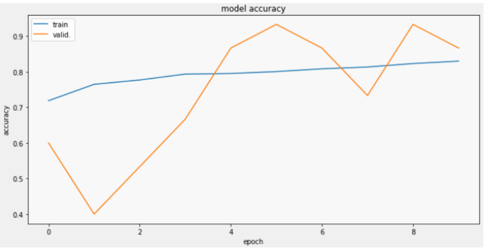


Figure 4.31: The model accuracy obtained by the alexnet8

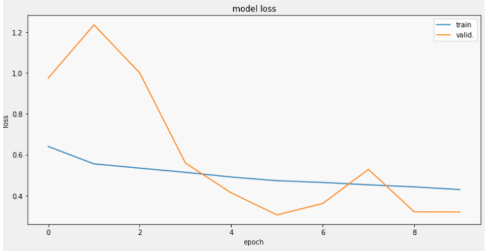
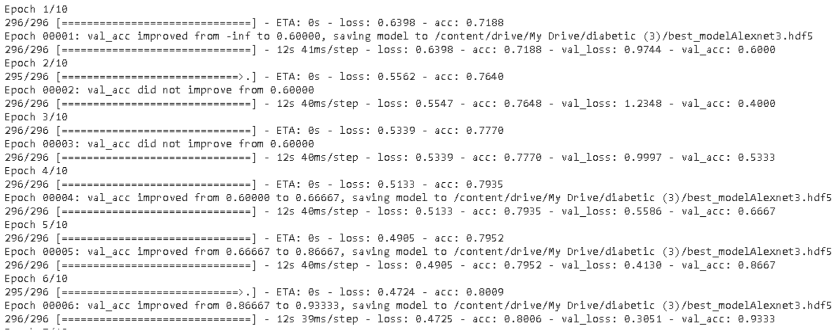


Figure 4.32: The model loss obtained by the alexnet8

In this case the model is doing fine since the training accuracy is 80,06% and the validation accuracy 93,33%.
This figure 4.33 will show when the best model is saved.



```
Epoch 1/10
296/296 [==============================] - ETA: 0s - loss: 0.6398 - acc: 0.7188
Epoch 00001: val_acc improved from -inf to 0.60000, saving model to /content/drive/My Drive/diabetic (3)/best_modelAlexnet3.hdf5
296/296 [==============================] - 12s 41ms/step - loss: 0.6398 - acc: 0.7188 - val_loss: 0.9744 - val_acc: 0.6000
Epoch 2/10
295/296 [============================>.] - ETA: 0s - loss: 0.5562 - acc: 0.7640
Epoch 00002: val_acc did not improve from 0.60000
296/296 [==============================] - 12s 40ms/step - loss: 0.5547 - acc: 0.7648 - val_loss: 1.2348 - val_acc: 0.4000
Epoch 3/10
296/296 [==============================] - ETA: 0s - loss: 0.5339 - acc: 0.7770
Epoch 00003: val_acc did not improve from 0.60000
296/296 [==============================] - 12s 40ms/step - loss: 0.5339 - acc: 0.7770 - val_loss: 0.9997 - val_acc: 0.5333
Epoch 4/10
296/296 [==============================] - ETA: 0s - loss: 0.5133 - acc: 0.7935
Epoch 00004: val_acc improved from 0.60000 to 0.66667, saving model to /content/drive/My Drive/diabetic (3)/best_modelAlexnet3.hdf5
296/296 [==============================] - 12s 40ms/step - loss: 0.5133 - acc: 0.7935 - val_loss: 0.5586 - val_acc: 0.6667
Epoch 5/10
296/296 [==============================] - ETA: 0s - loss: 0.4905 - acc: 0.7952
Epoch 00005: val_acc improved from 0.66667 to 0.86667, saving model to /content/drive/My Drive/diabetic (3)/best_modelAlexnet3.hdf5
296/296 [==============================] - 12s 40ms/step - loss: 0.4905 - acc: 0.7952 - val_loss: 0.4130 - val_acc: 0.8667
Epoch 6/10
295/296 [============================>.] - ETA: 0s - loss: 0.4724 - acc: 0.8009
Epoch 00006: val_acc improved from 0.86667 to 0.93333, saving model to /content/drive/My Drive/diabetic (3)/best_modelAlexnet3.hdf5
296/296 [==============================] - 12s 39ms/step - loss: 0.4725 - acc: 0.8006 - val_loss: 0.3051 - val_acc: 0.9333
```

Figure 4.33: Training process for Alexnet8.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 12 seconds, the model is validated at the end of each epoch.

4.4.3.1 Resnet50

we use Resnet50 for the second experiment after 5,16 minutes of training using google colabs GPU , we obtained these results .

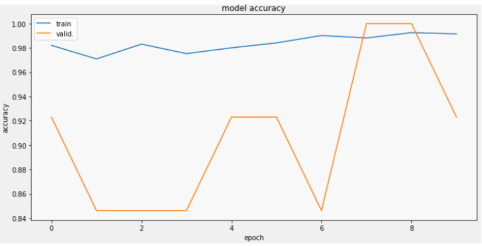


Figure 4.34: The model accuracy obtained by the Resnet50

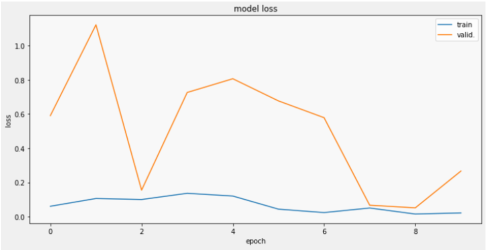
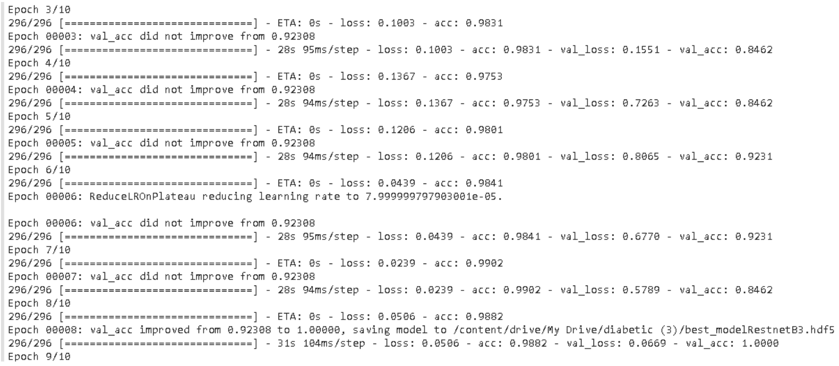


Figure 4.35: The model loss obtained by the Resnet50

In this case the Resnet50 is greater then alexnet8 since the training accuracy is 98,82% and the validation accuracy 100%.
This figure 4.36 will show when the best model is saved.



```
Epoch 3/10
296/296 [==============================] - ETA: 0s - loss: 0.1003 - acc: 0.9831
Epoch 00003: val_acc did not improve from 0.92308
296/296 [==============================] - 28s 95ms/step - loss: 0.1003 - acc: 0.9831 - val_loss: 0.1551 - val_acc: 0.8462
Epoch 4/10
296/296 [==============================] - ETA: 0s - loss: 0.1367 - acc: 0.9753
Epoch 00004: val_acc did not improve from 0.92308
296/296 [==============================] - 28s 94ms/step - loss: 0.1367 - acc: 0.9753 - val_loss: 0.7263 - val_acc: 0.8462
Epoch 5/10
296/296 [==============================] - ETA: 0s - loss: 0.1206 - acc: 0.9801
Epoch 00005: val_acc did not improve from 0.92308
296/296 [==============================] - 28s 94ms/step - loss: 0.1206 - acc: 0.9801 - val_loss: 0.8065 - val_acc: 0.9231
Epoch 6/10
296/296 [==============================] - ETA: 0s - loss: 0.0439 - acc: 0.9841
Epoch 00006: ReduceLROnPlateau reducing learning rate to 7.999999797903001e-05.

Epoch 00006: val_acc did not improve from 0.92308
296/296 [==============================] - 28s 95ms/step - loss: 0.0439 - acc: 0.9841 - val_loss: 0.6770 - val_acc: 0.9231
Epoch 7/10
296/296 [==============================] - ETA: 0s - loss: 0.0239 - acc: 0.9902
Epoch 00007: val_acc did not improve from 0.92308
296/296 [==============================] - 28s 94ms/step - loss: 0.0239 - acc: 0.9902 - val_loss: 0.5789 - val_acc: 0.8462
Epoch 8/10
296/296 [==============================] - ETA: 0s - loss: 0.0506 - acc: 0.9882
Epoch 00008: val_acc improved from 0.92308 to 1.00000, saving model to /content/drive/My Drive/diabetic (3)/best_modelRestnetB3.hdf5
296/296 [==============================] - 31s 104ms/step - loss: 0.0506 - acc: 0.9882 - val_loss: 0.0669 - val_acc: 1.0000
Epoch 9/10
```

Figure 4.36: Training process for Resnet50.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 31 seconds, the model is validated at the end of each epoch.

4.4.3.2 Efficentnet-B0

We use Efficentnet-B0 for the second experiment after 6,16 minutes of training using google colabs GPU , we obtained these results .

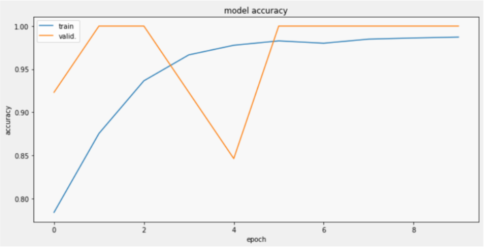


Figure 4.37: The model accuracy obtained by the Efficentnet-B0

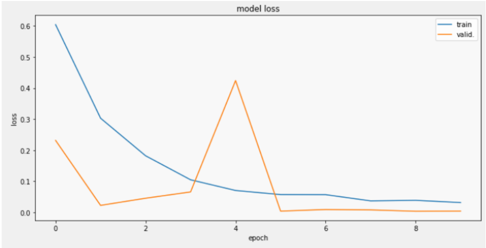
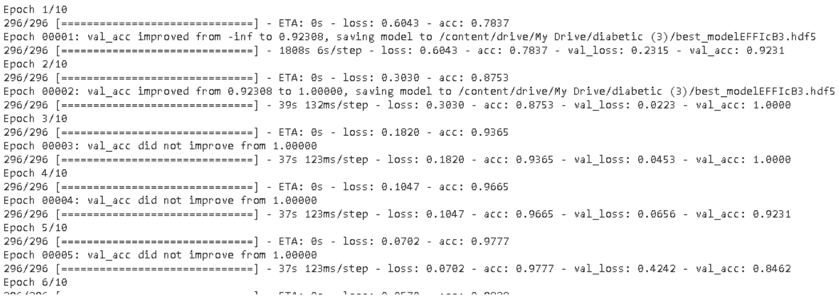


Figure 4.38: The model loss obtained by the Efficentnet-B0

In this case Efficentnet-B0 is greater then Alexnet8 but not Resnet50 since the training accuracy is 93,65% and the validation accuracy 100%.
This figure 4.39 will show when the best model is saved.



```
Epoch 1/10
296/296 [==============================] - ETA: 0s - loss: 0.6043 - acc: 0.7837
Epoch 00001: val_acc improved from -inf to 0.92308, saving model to /content/drive/My Drive/diabetic (3)/best_modelEFFIcB3.hdf5
296/296 [==============================] - 1808s 6s/step - loss: 0.6043 - acc: 0.7837 - val_loss: 0.2315 - val_acc: 0.9231
Epoch 2/10
296/296 [==============================] - ETA: 0s - loss: 0.3030 - acc: 0.8753
Epoch 00002: val_acc improved from 0.92308 to 1.00000, saving model to /content/drive/My Drive/diabetic (3)/best_modelEFFIcB3.hdf5
296/296 [==============================] - 39s 132ms/step - loss: 0.3030 - acc: 0.8753 - val_loss: 0.0223 - val_acc: 1.0000
Epoch 3/10
296/296 [==============================] - ETA: 0s - loss: 0.1820 - acc: 0.9365
Epoch 00003: val_acc did not improve from 1.00000
296/296 [==============================] - 37s 123ms/step - loss: 0.1820 - acc: 0.9365 - val_loss: 0.0453 - val_acc: 1.0000
Epoch 4/10
296/296 [==============================] - ETA: 0s - loss: 0.1047 - acc: 0.9665
Epoch 00004: val_acc did not improve from 1.00000
296/296 [==============================] - 37s 123ms/step - loss: 0.1047 - acc: 0.9665 - val_loss: 0.0656 - val_acc: 0.9231
Epoch 5/10
296/296 [==============================] - ETA: 0s - loss: 0.0702 - acc: 0.9777
Epoch 00005: val_acc did not improve from 1.00000
296/296 [==============================] - 37s 123ms/step - loss: 0.0702 - acc: 0.9777 - val_loss: 0.4242 - val_acc: 0.8462
Epoch 6/10
```

Figure 4.39: Training process for Efficentnet-B0.

The training process shows the progression in the learning and validation phases, an entire epoch is done in 37 seconds or 38 seconds, the model is validated at the end of each epoch.

4.4.3.3 Comparisons of our experiments

After we seen the three experiments, we conclude that the best model for the third dataset is Resnet50 .
This table 4.6 show the comparisons of our experiments.

	Accuracy	**Validation Accuracy**	**Time**
Alexnet8	80,06%	93,33%	1,83 min
Resnet50	**98,82%**	**100%**	**5,16 min**
Efficentnet-B0	93,65%	100%	6,16 min

Table 4.6: Comparisons of our experiments on the third dataset .

4.4.4 Comparisons our work and the previous works

In this section we will compare our different obtained results from our proposed deep convolutional networks with the previous works.

	Accuracy	**Dataset organization**
Alexnet8	92,42%	2 classes
Alexnet8	80,06%	3 classes
Alexnet8	80,08%	5 classes
Resnet50	95,97%	2 classes
Resnet50	**98,82%**	**3 classes**
Resnet50	**98,23%**	**5 classes**
Efficentnet-B0	**99,22%**	**2 classes**
Efficentnet-B0	93,65%	3 classes
Efficentnet-B0	98,16%	5 classes
Decision tree	96%	2 classes
Resnet34	85%	2 classes
SVM	86%	5 classes

Table 4.7: Accuracies comparisons of our experiments and previous works .

we conclude for the table 4.7 that convolutional neural nets give better results in term of accuracy and efficiency compared to classical Machine learning algorithms. CNN gives good results and Efficientnet-B0 outperforms all the results in the datasets with 2 classes but for the other 2 datasets Resnet50 is the best one .

Also we noticed that more classes we have, less accuracy we get.

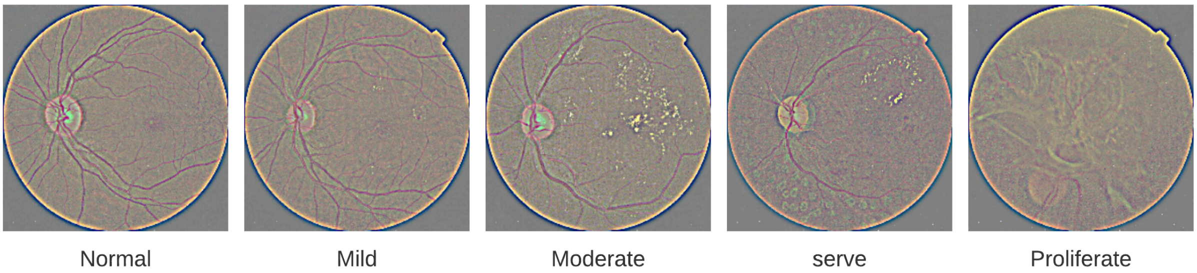


Figure 4.40: Fundus images of Normal retina and DR stages .

Among the difficulties we encountered in this study is that it is difficult to distinguish between cases of the retina, as we see in the figure 4.40 the convergent cases are similar and with many images from different eyes it becomes more difficult.

The detection of the patient has DR or not, it is much easier than the classification of retina in 5 classes.

4.5 Conclusion

This chapter covered the major findings of the thesis, the implementation and the experiments were detailed and explaied, the results were illustrated as figures, tables commented and comparisons about the obtained results.
And the implementation so easy with python and pretrained models.
We have also seen in this chapter that CNN models doesn't give the same results with different classes organization for the same dataset.

Conclusion and future work

This work represents case study and aims to develop and propose a classification models based on deep learning to automatically detect and diagnose retinal images better than the diagnosis of experts radiology and also previous published papers. To achieve research objectives we conducted many experiments using deep convolutional neural networks methods. As turned out, our proposed CNN models outperformed all the previous works.

The results has shown that the use of convolutional neural networks can give very interesting results in image classification and can be used to asses doctors in medical diagnose and opens up to a new, simple and effective method for early diabetic retinopathy detection. We have also seen that the use of some pretrained models can enhance the results in time and model effectiveness. Our approaches shows good results in term of classification performance, yet we didn't trained our models for long epochs due to the hardware limitations, and we haven't test our models in different datasets to see if they can perform as well as they did in this dataset.

For future, we would study further many related problems and test our models in different datasets to diagnose more retinal diseases. We also aim to integrate our model directly into an fundus scanning device to asses the ophthalmologists in making decisions, also into phones with remidio fundus on phone camera, it will help the patient to diagnose his retina .

Bibliography

[1] Georg Marckmann and Kenneth Goodman. “Introduction: Ethics of information technology in health care”. In: (2006).

[2] Kyoungyoung Jee and Gang-Hoon Kim. “Potentiality of big data in the medical sector: focus on how to reshape the healthcare system”. In: *Healthcare informatics research* 19.2 (2013), pp. 79–85.

[3] *healthcare*. URL: `https://dictionary.cambridge.org/dictionary/engalish/healthcare`.

[4] *healthcare*. URL: `http://www.businessdictionary.com/definition/health-care.html`.

[5] Charles Protzman, George Mayzell, and Joyce Kerpchar. *Leveraging lean in healthcare: Transforming your enterprise into a high quality patient care delivery system.* Productivity Press, 2018.

[6] World Health Organization et al. *Primary health care: A joint report.* World Health Organization, 1978.

[7] Fiona M Burns et al. “Missed opportunities for earlier HIV diagnosis within primary and secondary healthcare settings in the UK”. In: *Aids* 22.1 (2008), pp. 115–122.

[8] *tertiary-health-care*. URL: `https://pallipedia.org/tertiary-health-care/`.

[9] *medical publishing*. URL: `http://www.imedpub.com/scholarly/health-informatics-journals-articles-ppts-list.php`.

[10] Reinhold Haux and Petra Knaup. “Recommendations of the International Medical Informatics Association (IMIA) on education in health and medical informatics”. In: *Methods of Information in Medicine* 39.03 (2000), pp. 267–277.

[11] *healthcare management*. URL: `https://www.healthcare-management-degree.net/faq/what-exactly-is-health-informatics/`.

[12] Partha Pratim Ray and Poulami Majumder. “An Introduction to Pervasive Biomedical Informatics”. In: ().

[13] *the E-Health*. URL: http://europa.eu/.

[14] seventh report Committee A. "World Health Organization , Ninth plenary meeting". In: (2005).

[15] Kun Wang et al. "Mobile big data fault-tolerant processing for ehealth networks". In: *IEEE Network* 30.1 (2016), pp. 36–42.

[16] US Department of Health, Human Services, et al. "State Health Information Exchange Cooperative Agreement Program". In: *US Printing Office. HealthIT* (2011).

[17] *Health information management*. URL: https://www.ahima.org/careers/healthinfo.

[18] *International Classification of Diseases*. URL: https://www.who.int/classifications/icd/en/.

[19] Sally Adee. "Reverse engineering the brain". In: *IEEE Spectrum* 45.6 (2008), pp. 51–53.

[20] Maurits Butter et al. "Robotics for healthcare". In: (2008).

[21] Maurits Butter et al. *Robotics for healthcare: Final report*. English. 2008.

[22] Sean V Murphy and Anthony Atala. "3D bioprinting of tissues and organs". In: *Nature biotechnology* 32.8 (2014), pp. 773–785.

[23] Donovan Alexander. *The Science Fiction World of 3D Printed Organs*. Apr. 2020. URL: https://interestingengineering.com/the-science-fiction-world-of-3d-printed-organs.

[24] Donovan Alexander. *The Science Fiction World of 3D Printed Organs*. Apr. 2020. URL: https://interestingengineering.com/the-science-fiction-world-of-3d-printed-organs.

[25] Betty YS Kim, James T Rutka, and Warren CW Chan. "Nanomedicine". In: *New England Journal of Medicine* 363.25 (2010), pp. 2434–2443.

[26] Alexander Raif. *Cloud computing in the health care industry*. Nov. 2018. URL: https://medium.com/@alex.raif86/cloud-computing-in-the-health-care-industry-744580a07868.

[27] *5 Emerging Technologies and Their Impact on Health Informatics - Health Informatics Online Masters: Nursing Medical Degrees*. Dec. 2018. URL: https://healthinformatics.uic.edu/blog/5-emerging-technologies-and-their-impact-on-health-informatics/.

[28] Evelyn JS Hovenga, Michael R Kidd, and Sebastian Garde. *Health informatics: an overview*. Vol. 151. Ios Press, 2010.

[29] Eric J Topol. *Deep learning detects impending organ injury in the clinic*. 2019.

[30] Ida Arvidsson et al. "Generalization of prostate cancer classification for multiple sites using deep learning". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 191–194.

[31] Brandon Fan, Weiguo Fan, Carly Smith, et al. "Adverse drug event detection and extraction from open data: A deep learning approach". In: *Information Processing & Management* 57.1 (2020), p. 102131.

[32] Marylyn D Ritchie et al. "Methods of integrating data to uncover genotype–phenotype interactions". In: *Nature Reviews Genetics* 16.2 (2015), pp. 85–97.

[33] Sangwon Chae, Sungjun Kwon, and Donghyun Lee. "Predicting infectious disease using deep learning and big data". In: *International journal of environmental research and public health* 15.8 (2018), p. 1596.

[34] Mariette Awad and Rahul Khanna. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Apress, 2015.

[35] *What is Machine Learning?: Types of Machine Learning*. May 2019. URL: `https://www.edureka.co/blog/what-is-machine-learning/`.

[36] Canburak Tümer. *Machine Learning Basics with Examples - Part 2 Supervised Learning*. Aug. 2018. URL: `https://medium.com/@canburaktumer/machine-learning-basics-with-examples-part-2-supervised-learning-e2b740ff014c`.

[37] URL: `https://www.analyticssteps.com/blogs/introduction-machine-learning-supervised-and-unsupervised-learning`.

[38] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge, 1998.

[39] *Three Things to Know About Reinforcement Learning*. URL: `https://www.kdnuggets.com/2019/10/mathworks-reinforcement-learning.html`.

[40] Jeff Heaton. *Ian goodfellow, yoshua bengio, and aaron courville: Deep learning*. 2018.

[41] Harvey Lodish. *Overview of Neuron Structure and Function*. Jan. 1970. URL: `https://www.ncbi.nlm.nih.gov/books/NBK21535/`.

[42] Kite. *Introduction to Artificial Neural Networks in Python*. Aug. 2019. URL: `https://roioverload.com/biology-inspires-the-artificial-neural-network/`.

[43] Boundless. *Boundless Psychology*. URL: `https://courses.lumenlearning.com/boundless-psychology/chapter/neurons/`.

[44] Andrej Krenker, Janez Bešter, and Andrej Kos. "Introduction to the artificial neural networks". In: *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech* (2011), pp. 1–18.

[45] *7 Types of Activation Functions in Neural Networks: How to Choose?* URL: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/.

[46] Amey Varangaonkar - et al. *Top Deep Learning Architectures.* Sept. 2018. URL: https://hub.packtpub.com/top-5-deep-learning-architectures/.

[47] Athanasios Voulodimos et al. "Deep learning for computer vision: A brief review". In: *Computational intelligence and neuroscience* 2018 (2018).

[48] Manjunath Jogin et al. "Feature extraction using Convolution Neural Networks (CNN) and Deep Learning". In: *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).* IEEE. 2018, pp. 2319–2323.

[49] Ping-Rong Chen et al. "DSNet: An efficient CNN for road scene segmentation". In: *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).* IEEE. 2019, pp. 424–432.

[50] Yann LeCun et al. "LeNet-5, convolutional neural networks". In: *URL: http://yann. lecun. com/exdb/lenet* 20.5 (2015), p. 14.

[51] *7.1. Deep Convolutional Neural Networks (AlexNet)¶ Colab [mxnet] Open the notebook in Colab Colab [pytorch] Open the notebook in Colab Colab [tensorflow] Open the notebook in Colab.* URL: https://d2l.ai/chapter_convolutional-modern/alexnet.html#architecture.

[52] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[53] Mingxing Tan and Quoc V Le. "Efficientnet: Improving accuracy and efficiency through automl and model scaling". In: *arXiv preprint arXiv:1905.11946* (2019).

[54] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 2818–2826.

[55] Danilo Mandic and Jonathon Chambers. *Recurrent neural networks for prediction: learning algorithms, architectures and stability.* Wiley, 2001.

[56] URL: https://www.edrawsoft.com/neural-network-examples.html.

[57] Hongyang Xu et al. "Sentiment Analysis of Chinese Version Using SVM & RNN". In: *Proceedings of the 6th International Conference on Information Engineering.* 2017, pp. 1–5.

[58] Ann Pietrangelo Cherney and Kristeen. *The Effects of Diabetes on Your Body.* June 2040. URL: https://www.healthline.com/health/diabetes/effects-on-body#1.

[59] Written by Editor Updated on 15th January 2019 and Editor. *Knowing how diabetes affects your body can help you look after your body and prevent diabetic complications from developing.* Jan. 2020. URL: `https://www.diabetes.co.uk/how-does-diabetes-affect-the-body.html`.

[60] *Diabetic retinopathy.* May 2018. URL: `https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611#:~:text=Diabetic%20retinopathy%20(die-uh-,or%20only%20mild%20vision%20problems.`.

[61] Harvard Health Publishing. *Retinopathy.* URL: `https://www.health.harvard.edu/a_to_z/retinopathy-a-to-z`.

[62] Michael Abràmoff and Christine N. Kay. "Chapter 6 - Image Processing". In: *Retina (Fifth Edition).* Ed. by Stephen J. Ryan et al. Fifth Edition. London: W.B. Saunders, 2013, pp. 151–176. ISBN: 978-1-4557-0737-9. DOI: `https://doi.org/10.1016/B978-1-4557-0737-9.00006-0`. URL: `http://www.sciencedirect.com/science/article/pii/B9781455707379000060`.

[63] *Remidio Fundus.* URL: `https://www.remidio.com/`.

[64] Udyavara R Acharya et al. "Computer-based detection of diabetes retinopathy stages using digital fundus images". In: *Proceedings of the institution of mechanical engineers, part H: journal of engineering in medicine* 223.5 (2009), pp. 545–553.

[65] Mehdi Torabian ESFAHANI, Mahsa GHADERI, and Raheleh KAFIYEH. "Classification of diabetic and normal fundus images using new deep learning method". In: *Leonardo Electron J Pract Technol* 17.32 (2018), pp. 233–248.

[66] Elaouaber Zineb Aziza et al. "Decision tree CART algorithm for diabetic retinopathy classification". In: *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA).* IEEE. 2019, pp. 1–5.

[67] Sovit Ranjan Rath. *Diabetic Retinopathy 224x224 Gaussian Filtered.* Feb. 2020. URL: `https://www.kaggle.com/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered`.

[68] *APTOS 2019 Blindness Detection.* URL: `https://www.kaggle.com/c/aptos2019-blindness-detection/data`.

[69] SA Jameel and AR Mohamed Shanavas. "Implementation of Improved Gaussian Filter Algorithm for Retinal Fundus Images". In: *International Journal of Computer Applications* 132.8 (2015), pp. 0975–8887.

[70] Rikiya Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9.4 (2018), pp. 611–629.

[71] Tianyi Liu et al. "Implementation of training convolutional neural networks". In: *arXiv preprint arXiv:1506.01195* (2015).

[72] *AlexNet.* July 2020. URL: https://en.wikipedia.org/wiki/AlexNet#cite_note-quartz-1.

[73] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. "A Review of Convolutional Neural Networks". In: *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE).* IEEE. 2020, pp. 1–5.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems.* 2012, pp. 1097–1105.

[75] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[76] Edna Chebet Too et al. "A comparative study of fine-tuning deep learning models for plant disease identification". In: *Computers and Electronics in Agriculture* 161 (2019), pp. 272–279.

[77] Aditya Chatterjee. *Evolution of CNN Architectures: LeNet, AlexNet, ZFNet, GoogleNet, VGG and ResNet.* Jan. 2019. URL: https://iq.opengenus.org/evolution-of-cnn-architectures/.

[78] Baki Er. *Microsoft Presents : Deep Residual Networks.* Aug. 2016. URL: https://medium.com/@bakiiii/microsoft-presents-deep-residual-networks-d0ebd3fe5887.

[79] *EfficientNet: Summary and Implementation.* URL: https://hackmd.io/@bouteille/HkH1jUArI#2-Evaluating.

[80] Anand Borad. *Image Classification with EfficientNet: Better performance with computational efficiency.* Dec. 2019. URL: https://medium.com/analytics-vidhya/image-classification-with-efficientnet-better-performance-with-computational-efficiency-f480fdb00ac6.

[81] Xiaojie Jin et al. "WSNet: Learning Compact and Efficient Networks with Weight Sampling". In: (2018).

[82] *Python core team.* URL: https://www.python.org/.

[83] *matplotlib documentation.* URL: https://matplotlib.org/faq/usage_faq.html#coding-styles.

[84] *tensorflow.* URL: https://www.tensorflow.org/.

[85] Keras Team. *keras.* URL: https://keras.io/.

[86] *Your Machine Learning and Data Science Community.* URL: https://www.kaggle.com/.

[87] *Google colab.* URL: https://colab.research.google.com/.

[88] *CUDA Explained - Why Deep Learning uses GPUs.* URL: https://deeplizard.com/learn/video/6stDhEA0wFQ.

[89] Jeremy West, Dan Ventura, and Sean Warnick. "Spring research presentation: A theoretical foundation for inductive transfer". In: *Brigham Young University, College of Physical and Mathematical Sciences* 1.08 (2007).

[90] Keras Team. *Keras documentation: Image data preprocessing.* URL: https://keras.io/api/preprocessing/image/#imagedatagenerator-class.

[91] Navin Kumar Manaswi. "Understanding and working with Keras". In: *Deep Learning with Applications Using Python.* Springer, 2018, pp. 31–43.

[92] Mahdi Hanifi. *Sequential API vs Functional API model in Keras.* Dec. 2019. URL: https://medium.com/@hanify/sequential-api-vs-functional-api-model-in-keras-266823d7cd5e.