

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES
Computer Science Department

Thesis

submitted in partial fulfilment of the requirements for the degree of

ACADEMIC MASTER

Field: **Mathematics and Computer Science**

Option: **Computer Science**

Specialty: **Distributed Systems and Artificial Intelligence**

ccPresented by :

- **Kamal bouredha**
- **Ikhlas hamaidi**

Title

Categorization of Digital Pathology Image using Deep Learning model

Examination Committee :

M.	Nawi Muhammad Anwar.....	MCA	Chair
M.	Lajdal Ibrahim.....	MAA	Examinator
M.	Guia Sana Sahar.....	MAA	Supervisor

Academic year : 2022-2023

THANKS

First and foremost, we thank **God Almighty** who guided us to his book and by which we were able to complete this humble work.

We also extend our sincere thanks and gratitude from the bottom of our hearts to

Guia Sana Sahar

who helped us, directed us, and treated us like a dear brothers, we wish to her happiness and success in various fields.

Abstract

The success of Artificial Intelligence (AI) in many tasks in computer vision expects that digital pathology applications can exploit AI potential. Histopathology primarily involves the examination of biological tissues at a microscopic level to observe the characteristics of both diseased cells and tissues.

The digital pathology toolkit's ability to automatically identify various tissue types within histological images is a crucial component. However, most existing studies focused on the binary classification of histological images (tumor/stroma), far from meeting the clinical demand.

In this thesis, we proposed a deep learning architecture for the multi-class classification of histological images. The multi-class texture challenge pertains to the task of categorizing several distinct tissue types when presented with a tissue sample, in this instance, comprising eight different types.

To find an optimal classification strategy, we adopted a Fully Convolutional Network (FCN) as a branch of convolution layers with downsampling and upsampling inside the network, followed by the fully connected layers, and then the softmax classifier.

Keywords :

Digital pathology images; histological images; multi-classification; deep learning.

ملخص

مع نجاح الذكاء الاصطناعي (AI) في العديد من المهام في مجال رؤية الحاسوب، ينتظر أن يستفيد علم الأمراض الرقمية من إمكانيات الذكاء الاصطناعي. علم الأمراض النسيجي معني أساساً بفحص الأنسجة البيولوجية لمراقبة مظهر الخلايا والأنسجة المصابة على مستويات مجهرية. التعرف التلقائي على أنواع الأنسجة المختلفة في الصور النسيجية ضروري جداً لمجموعة أدوات علم الأمراض الرقمية. ومع ذلك، تركز معظم الدراسات الحالية بشكل رئيسي على التصنيف الثنائي للصور النسيجية (ورم/أنسجة طبيعية)، وهو بعيد عن تلبية الاحتياجات السريرية. في هذه الرسالة، اقترحنا نموذج تعلم عميق للتصنيف متعدد الفئات للصور النسيجية؛ مشكلة النسيج متعدد الفئات تشير إلى مشكلة تجميع أنواع أنسجة متعددة معينة في ضوء عينة من الأنسجة (في هذه الحالة، ثمانية أنواع مختلفة).

للعثور على استراتيجية تصنيف مثالي، اعتمدنا شبكة FCN كفرع من الطبقات التلافيفية، تليها الطبقات المتصلة بالكامل ثم المصنف softmax.

الكلمات المفتاحية:

الصور الباثولوجية الرقمية، الصور النسيجية، متعدد التصنيف، تعلم عميق.

List of Figures

1.1	X-ray image	3
1.2	Ultrasound image [14]	4
1.3	MRI image [16]	4
1.4	Machine learning techniques include both unsupervised and supervised learning. [26]	9
1.5	An image explaining how the Clustering works [30]	10
1.6	Deep Neural networks [31]	10
1.7	Linear SVM[32]	11
2.1	KNN algorithms application example[32]	14
2.2	Fuzzy c-means algorithms example	16
2.3	SVMs application example	16
2.4	Typical biological neuron	18
2.5	Feedforward neuron network	19
2.6	Architecture of an RBF Network	19
2.7	Architecture of a SOM network	20
2.8	Architecture of a Recurrent Neural Network	21
3.1	VGG19 vs ResNet34	26
3.2	skip connections	27
3.3	Fully convolutional neural network architecture (FCN-8). [59]	28
3.4	ResNet-50 Architecture.	30
3.5	Model Architecture	31
4.1	Example OF Dataset with 8 classes	34
4.2	Python Logo[73]	36
4.3	Tensorflow Logo[74]	36
4.4	Scikit-Learn Logo[76]	37
4.5	Pandas Logo[77]	37
4.6	OpenCV Logo[78]	38
4.7	Matplotlib Logo[79]	38
4.8	Numpy Logo[80]	39
4.9	Colab Logo[81]	39
4.10	After perform augmentation	40

4.11	The model accuracy curve for both train and validation sets	48
4.12	The model loss curve for both train and validation sets	48
4.13	Area under the curve	49
4.14	History of precision	49
4.15	History of f1-score	50
4.16	The recap of all metric's history	50
4.17	evaluating the model	51
4.18	Prediction results	51

Contents

General Introduction

1 Cancer and digital diagnosis

1.1	Introduction	1
1.2	Cancer and digital diseases	1
1.2.1	overview	1
1.2.2	colorectal cancer	1
1.3	Medical imaging	2
1.3.1	Overview	2
1.3.2	Types of Medical Imaging	3
1.3.3	Digital pathology	4
1.4	Computer Aided Diagnosis	5
1.4.1	Overview	5
1.4.2	Artificial intelligence technique	6
1.4.3	Knowledge base systems	7
1.4.4	Machine learning technique	9
1.5	Conclusion	11

2 Image classification by deep learning **12**

2.1	introduction	13
2.2	Image classification	13
2.2.1	Definition	13
2.2.2	Principles of classification:	13
2.2.3	Medical image classification	16
2.3	Deep learning classification	17
2.3.1	Neural recent work	17
2.3.2	Different types of Neural Networks	18
2.4	Previous research	21
2.5	Conclusion	22

3 Proposed deep learning model **23**

3.1	Introduction	24
3.2	TRANSFER LEARNING	24

3.3	Residual Neural Network (ResNet)	25
3.3.1	Motivation	25
3.3.2	Residual Block	27
3.4	fully convolutional network FCN	27
3.5	Proposed Method	28
3.5.1	Overall Architecture	28
3.5.2	Dimensionality reduction	29
3.5.3	ResNet50	29
3.5.4	Fully Convolutional Network	30
3.5.5	Head Part	31
3.6	Conclusion	32

4 Implementation and experiment result 33

4.1	Introduction	34
4.2	Colonrectal cancer	34
4.3	Evaluation criteria	35
4.3.1	Area under the curve (AUC)	35
4.4	Software environment	35
4.4.1	Python	35
4.4.2	Tensorflow	36
4.4.3	Scikit-Learn	36
4.4.4	Pandas	37
4.4.5	OpenCV	37
4.4.6	Matplotlib	38
4.4.7	Numpy	38
4.5	Computing Resources	39
4.6	Implementation	39
4.6.1	Augmentation and Preprocessing	39
4.6.2	Implementation of the proposed method	40
4.7	Experiments and results	48
4.8	Conclusion	51

General Conclusion

General Introduction

Computer-aided detection or diagnosis (CAD) has emerged as a vibrant field of research in recent times. The real-world applications of CAD, particularly leveraging deep learning technology within artificial intelligence, have effectively addressed the challenge of manual data interpretation in traditional CAD systems.

Colorectal carcinoma is a significant contributor to cancer-related death in the United States. It ranks as the third most prevalent cancer in the United States for men, following prostate and lung/bronchus cancers, and for women, following breast and lung/bronchus cancers [1]. With a high frequency in many nations, colorectal cancer is a serious health hazard on a global scale. This form of cancer affects the colon or rectum and can have a significant negative effect on people's life. Automatically identifying various tissue types in histological pictures is crucial for the analysis of colorectal cancer samples in the field of digital pathology. The tumour/stroma ratio in histological samples has been calculated using texture analysis, a widely used approach. However, it should be noted that histological pictures frequently contain more than simply two tissue types, and prior research' efforts to solve the multi-class problem in the detection of colorectal cancer have been very restricted. [2]

We can improve the capabilities of digital pathology tools and further our understanding of this complicated illness by utilizing machine learning algorithms and AI artificial intelligence. [3]

Convolution neural network (CNN) is a state-of-the-art model for detection and diagnosis in deep learning, a subfield of machine learning where the computer automatically finds the characteristics needed for the job in raw input data. When computing resources are not accessible, the employment of pre-trained CNN and transfer learning mechanisms offers a potential alternative. Additionally, the use of the attention mechanism has led to favorable outcomes in terms of diagnosis and categorization.

For more effective classification and detection in the many histological pictures of colorectal cancer, this thesis combines a pre-trained CNN (ResNet-50) as the contracting path of the fully convolutional network (FCN) with the spatial atten-

tion module followed the head map composed of by the fully connected layers, and then the softmax classifier .

This thesis consists of 4 chapters:

- In **the first Chapter** we introduce some basic concepts of colorectal cancer, then We talked about Medical images and their types , and finally we review the field of computer aided diagnosis.
- In **the second Chapter** we will talk about deep learning, its definition, its applications and to the most famous algorithms, then we review colorectal cancer detection using deep learning techniques.
- In **the third Chapter** We will present the proposed deep learning model utilized in the project, as well as the technologies utilised.
- And in the last chapter we will talk about the dataset, the implementation of the proposed method and the experiment result.

Chapter 1

Cancer and digital diagnosis

1.1 Introduction

Globally, colorectal cancer is a serious health problem due to its high frequency in many nations. Successful treatment outcomes for colorectal cancer depend on early detection and a precise diagnosis. This emphasizes the requirement for more study and the investigation of novel strategies, such as the use of artificial intelligence and machine learning to enhance the precise identification and separation of various tissue types in colorectal cancer histology pictures. In this chapter, we define cancer disease and introduce some previous AI algorithms that have provided significant results in diagnosing and detecting cancer.

1.2 Cancer and digital diseases

There are many different types of cancer, all with different causes. It develops as a result of the body's aberrant cells growing and dividing uncontrollably.

1.2.1 overview

Cancers are conditions that cause abnormal cell development and can spread to other body areas. Their causes, symptoms, and treatments vary. Lung, colon and rectum, liver, stomach, and breast cancers are the most prevalent types of cancer, and they are also the main cause of death globally. Early cancer detection can increase survival, lower treatment costs, and lessen side effects. Testing healthy people as part of a screening plan aims to identify malignancies before they manifest any symptoms.[4]

1.2.2 colorectal cancer

The disease colorectal cancer has a significant global impact since it kills many people and has a large global impact. It is caused by aberrant colon or rectum cell proliferation and is regulated by genetic, environmental, and inherited factors. [5] The World Health Organization (WHO) has shown high colorectal cancer mortality rates, reiterating the disease's position as a primary cause of death. [6] Given its ubiquity, it is essential to raise awareness of this urgent health issue, adopt early detection measures, and enhance treatment options.

Definition

Colorectal cancer is a form of cancer that develops in the colon or rectum, which are organs in the digestive system. It is sometimes referred to as bowel cancer or

colon cancer. It often starts as a proliferation of abnormal cells, known as polyps, in the rectum or colon's inner lining. These polyps have the potential to grow into malignant tumors over time. These aberrant cells can invade surrounding tissues and potentially move to other regions of the body, a process known as metastasis, which is a hallmark of colorectal cancer. Changes in bowel habits, rectal bleeding, abdominal discomfort, unexplained weight loss, and exhaustion are typical signs of colorectal cancer. For successful treatment and better results, early diagnosis through screening tests and rapid medical intervention are crucial. [7]

Symptoms

Recent changes in bowel habits, stomach pain or discomfort, rectal bleeding, unexplained weight loss, and exhaustion are all common indicators of colon cancer. The more advanced the condition, the more severe these symptoms could be. Other symptoms include stomach distention, lethargy, lack of appetite, nausea, and vomiting. It is crucial to keep in mind that the particular symptoms encountered might change depending on the traits and stage of the malignancy. [8]

evaluation of colorectal cancer

A careful assessment and management of colorectal cancer are necessary since it is a serious health risk. An effective diagnosis and early detection are essential for successful treatment results. In order to find abnormalities or potential malignant growths, evaluation often entails a mix of screening procedures, including colonoscopies, fecal occult blood tests, and imaging scans. A thorough medical history is also taken, and a physical exam is performed to evaluate risk factors, symptoms, and general health. In order to identify colorectal cancer and determine its stage and aggressiveness, a biopsy and laboratory testing are essential. In order to uncover inherited variables that may contribute to the development of the disease, genetic testing may also be used as part of the evaluation procedure. A multidisciplinary approach combining healthcare specialists from different specialties enables a thorough evaluation and makes it easier to create a treatment plan that is suitable and catered to the needs of the specific patient. [9]

1.3 Medical imaging

1.3.1 Overview

By offering non-invasive or minimally invasive ways to see inside body structures and organs, medical imaging plays a crucial role in contemporary healthcare.

It includes many imaging techniques as X-rays, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and nuclear medicine methods. These imaging technologies enable medical personnel to identify and keep track of a variety of injuries, illnesses, and anomalies. Medical imaging enables early intervention and effective treatment planning by assisting in the diagnosis of cancers, fractures, infections, and other disorders. It offers thorough anatomical and functional details that aid in precise diagnosis, direct surgical operations, and track treatment outcomes. Medical imaging is always improving, offering better resolution, better imaging methods, and greater accessibility, all of which have a substantial positive impact on patient treatment and health outcomes.[10]

1.3.2 Types of Medical Imaging

We will explain the most important types of medical imaging available :

✓ **X-rays** : X-rays are a popular imaging method in medicine that employs electromagnetic radiation to create pictures of the body's interior organs. In order to see bones clearly and spot fractures, cancers, infections, and other anomalies, X-rays are frequently employed.

A little amount of ionizing radiation is used throughout the process to expose the body, and the resulting detailed pictures are then recorded on film or a digital detector to help with diagnosis and treatment planning.[11]The figure1.1 below shows an example of X-ray image

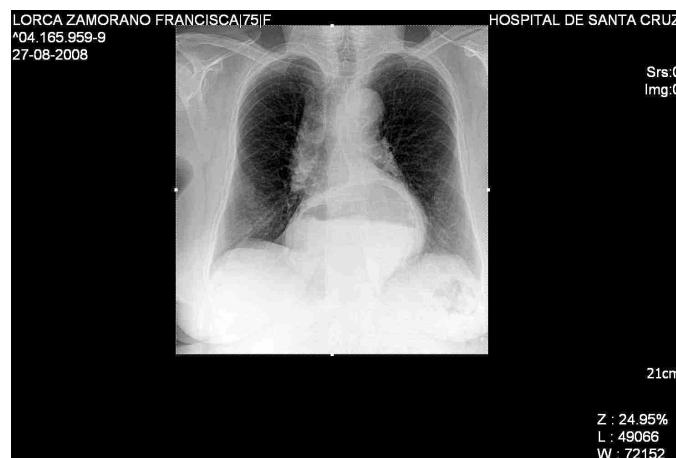


Figure 1.1: X-ray image
[12]

✓ **Medical ultrasound** High-frequency sound waves are used in medical ultrasonography, a non-invasive imaging method, to provide live pictures of inside organs and tissues. Medical ultrasonography is a technique that uses sound waves to create images that may be used to diagnose a variety of medical conditions. In

fields including obstetrics, cardiology, gastrointestinal, and musculoskeletal health, it helps in evaluating and diagnosing disorders. Medical ultrasonography plays a key role in directing medical treatments, evaluating therapy progress, and enhancing patient care due to its mobility, safety, and capacity to view soft tissues in real-time. [13]The figure1.2 below shows an example of Ultrasound image

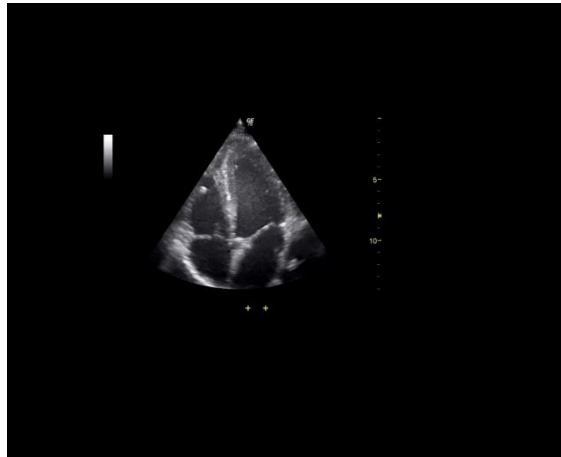


Figure 1.2: Ultrasound image [14]

✓ **Magnetic Resonance Imaging** A powerful magnetic field and radio waves are used to create comprehensive pictures of the body's interior structures during magnetic resonance imaging (MRI), a non-invasive medical imaging method that aids in the accurate and timely diagnosis of a variety of illnesses. [15]The figure1.3 below shows an example of MRI image image

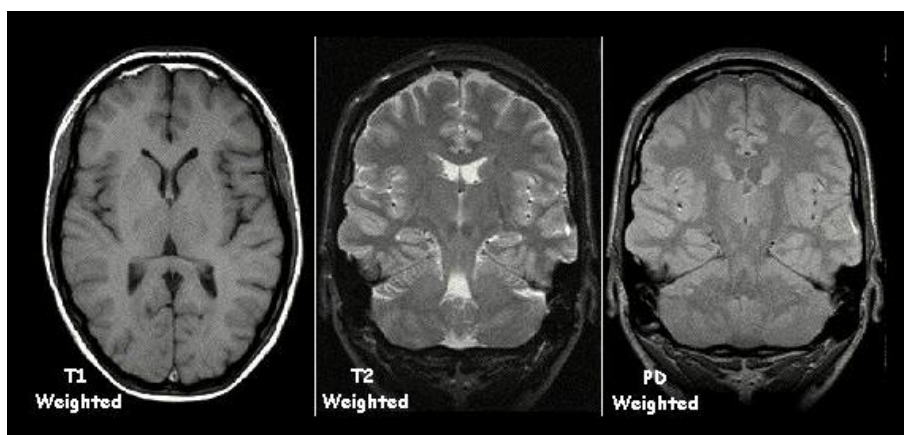


Figure 1.3: MRI image [16]

1.3.3 Digital pathology

High-resolution photographs of tissue samples may be captured, managed, and studied using digital technology in the discipline of digital pathology. Digital

pathology slides may be reviewed and interpreted by pathologists, enabling remote consultations, effective data archiving, and cutting-edge image processing algorithms for better pathology research and diagnosis. [17]

Image analysis

A crucial area of medical imaging is image analysis, which comprises the quantitative and qualitative analysis of pictures to extract valuable information and support diagnostic and research procedures. It includes several image analysis and interpretation methods and algorithms, including pattern recognition, segmentation, feature extraction, and classification. Healthcare workers can use image analysis to recognize and quantify anatomical features, find anomalies or lesions, monitor disease development, and assess therapy outcomes. The discipline is still developing thanks to the integration of artificial intelligence and machine learning, which offers better automation, accuracy, and efficiency in the interpretation of medical pictures.[18]

Artificial intelligence in medical imaging

Artificial intelligence (AI), specifically deep learning algorithms, has garnered significant interest for its remarkable capabilities in image identification tasks. These algorithms have the ability to automatically analyze and provide quantitative assessments of complex medical imaging. This advancement in AI technology has the potential to assist clinicians in making more precise diagnoses in reproductive imaging while reducing their workload. Currently, deep learning algorithms and classical machine learning algorithms are the two primary types of AI technologies that are often used in medical imaging. These developments in AI have enormous potential to transform the area of medical imaging and improve patient care. [19]

1.4 Computer Aided Diagnosis

1.4.1 Overview

A CAD system is a sophisticated computer system that helps in the identification and diagnosis of diseases. They serve as auxiliary tools for radiologists, utilizing algorithms and pattern recognition to analyze medical pictures. CAD systems improve accuracy, efficiency, and diagnostic results by bringing to light possible anomalies.[20]

1.4.2 Artificial intelligence technique

Medical imaging is being transformed by artificial intelligence (AI), which makes it possible for computer systems to imitate human intellect. AI is widely used in a variety of medical imaging modalities, such as nuclear medicine, ultrasound, and radiology. By automating some processes, its use in healthcare enables clinicians to diagnose patients with greater accuracy and lessens their effort. Medical personnel may improve patient care and results by utilizing AI's ability to increase the accuracy and efficiency of reproductive imaging diagnostics.[21]

Advantages of Artificial Intelligence

- AI is used in a variety of fields, including medicine, security, science, and so on.
- Helps in problem solving.
- Artificial intelligence provides data loss prevention.
- Provides a high success rate in business.
- Knowledge can be spread easily.
- AI can do any complex works.

Goals of Artificial intelligence

1. The goal is to develop knowledge base systems that are intelligent, capable of learning, capable of demonstration, capable of explanation, and capable of providing users with advice.
2. In order to incorporate human intelligence in computers, the goal is to create systems that display comprehension, reasoning, learning, and behavior that are similar to those of humans.

There are three important AI techniques

- **Search** :functions as a technique for locating solutions to issues with no clear answer. Additionally, it provides a foundation for integrating current direct approaches.
- **Knowledge application** : includes using the items' structural characteristics to solve complex issues.
- **Abstraction** : Allows important features and variations to be separated.

1.4.3 Knowledge base systems

Overview

Knowledge base systems, often referred to as KBS, are computerized systems that organize and store knowledge to aid in decision- and problem-solving processes. They are made to record and arrange knowledge in a systematic way so that users may access it as needed. A knowledge base system often consists of a database or repository where data is kept, as well as a collection of guidelines, formulas, and reasoning tools that make it easier to access and use knowledge. These systems help with problem-solving, decision-making, and rapid access to pertinent information in a variety of fields, including medical, engineering,... etc.

Applications of Knowledge base system

- These systems enable IT service management in information technology by providing a consolidated database of technical information, answers to issues, and best practices for effective IT operations and support.
- Knowledge base systems in customer service act as informational databases for answering frequently asked inquiries and offering self-service choices.
- They are used to give access to medical guidelines, recommendations for treatments, and assistance for diagnostic procedures.

Medical knowledge based systems

Knowledge base systems play a vital role in medical diagnosis by providing valuable support to healthcare professionals. In-depth medical knowledge, including symptoms, illnesses, treatment regimens, and research findings, is contained in these systems. Physicians can use knowledge base systems to retrieve pertinent information, compare patient symptoms and medical history, and obtain probable diagnosis or treatment recommendations when dealing with difficult situations. Healthcare practitioners may diagnose patients with more accuracy and efficiency by drawing on the large body of medical information that is housed within the system, improving patient outcomes. Furthermore, knowledge base systems can improve clinical decision-making, lower diagnostic mistakes, and promote lifelong learning in the medical industry.[22]

Studies on Medical knowledge based systems

1. The researcher **James Perry** and his colleagues presented a technical experiment for a knowledge base containing more than 2000 encoded diseases and 450 medicines with more than 8000 notes as semantic terms and using 117

disease cases. The results of the research showed that the suggestions made by the initial model are correct in 33% of the cases.[23]

2. The researcher **Roland Grad** and his colleagues presented a knowledge-based recommendation model that links patient electronic records with clinical information, and the results showed that the average accuracy is 98%. [24]

Advantages of knowledge based systems

- **Enhanced Decision-Making:** Knowledge-based systems provide users access to a database of structured knowledge, enabling them to make decisions that are more well-informed and precise. Improved results are the consequence of the organized knowledge base's assistance in problem-solving, diagnosis, and identifying relevant remedies.
- **Utilizing a knowledge base guarantees consistency** in the procedures used to make decisions and solve problems. The technology reduces human error and variability in results by adhering to specified rules and principles. Standardization and the constant use of best practices are encouraged by this.
- **Collaboration and information Sharing:** Knowledge-based platforms make it easier for people to share and disseminate information inside a company. Users have access to a common knowledge base where they may share their information and take use of the system's collective wisdom. This encourages cooperation, education, and ongoing development within the company.
- **Implementing a knowledge-based system simplifies operations,** cuts down on redundancy, and boosts efficiency. Organizations may save time and money by automating monotonous procedures, facilitating rapid access to pertinent information, and promoting self-service choices. Utilizing the knowledge base's stored expertise maximizes resource use and improves overall operational efficiency.

Limitations of knowledge based systems

- It bases its choice entirely on the knowledge base's information. Any discrepancy in the data might result in poor judgment.
- It unlike humans, is incapable of providing a one-size-fits-all solution to problems.
- The cost of keeping these systems in good working order is quite high.

1.4.4 Machine learning technique

Developing methods and models that allow computers to learn from data and make predictions or judgments based on that data without being explicitly programmed is the focus of the subfield of artificial intelligence known as machine learning. It entails using a lot of data to train a computer system so that it can recognize patterns, connections, and insights on its own. Depending on the type of data and the learning job at hand, machine learning algorithms can be divided into supervised learning, unsupervised learning, and reinforcement learning. By continually learning from fresh data, machine learning aims to provide computers the ability to gradually improve their performance or behavior without human involvement.[25]The diagram in the figure1.4 below shows machine learning techniques

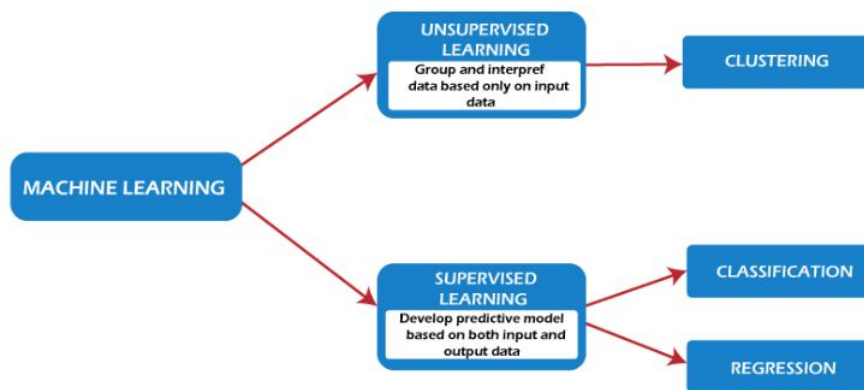


Figure 1.4: Machine learning techniques include both unsupervised and supervised learning. [26]

Now Let’s have a look at some of the most popular machine learning techniques:

1. **Regression** Regression is a machine learning approach that uses input information to forecast continuous numerical outputs. The link between independent variables and a dependent variable is what it seeks to determine. Regression models may be utilized for tasks like forecasting, price estimation, or trend prediction since they learn from previous data to create predictions.[27]
2. **Classification** A machine learning strategy called classification focuses on classifying data into specified groups or categories. It entails taking labeled data to train a model with known class labels, then using the trained model to classify new, ambiguous data points into one of the known classes. In processes like spam detection, picture recognition, sentiment analysis, and illness diagnosis, classification is frequently employed.[27]

3. **Clustering** is the process of dividing a population of data points into several groups such that data points within a group are more similar to one another and distinct from data points outside of that group.[28, 29]the Figure1.5 below explains how the Clustering works

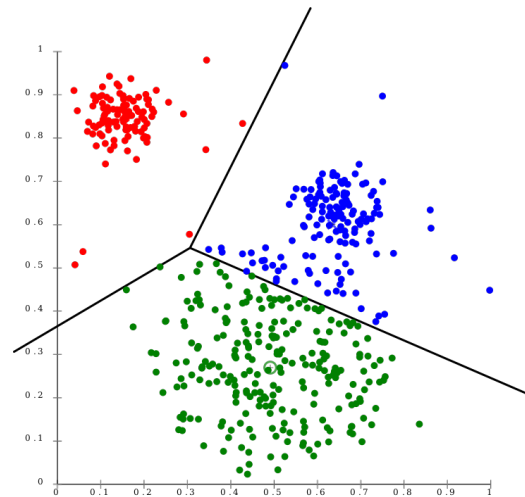


Figure 1.5: An image explaining how the Clustering works [30]

4. **Neural networks**, A family of machine learning models called neural networks, commonly referred to as artificial neural networks, are modeled after the structure and operation of biological brain networks. They are made up of linked neurons arranged in layers. In order to learn and model complicated patterns and correlations in the data, neural networks are used. They are excellent in deep learning applications, natural language processing, and picture and audio recognition.[28, 29]The figure1.6 below is an example of deep neural networks

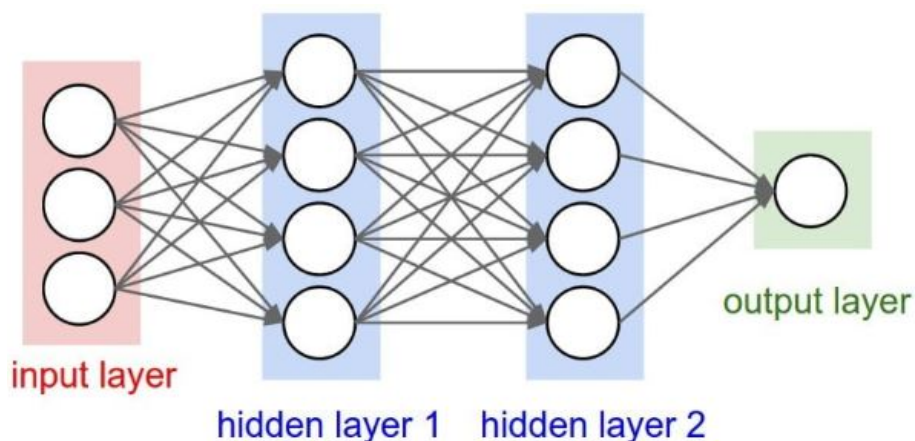


Figure 1.6: Deep Neural networks [31]

5. **Support Vector Machine** (SVM) is an old-school machine learning method that might still be useful for classifying massive volumes of data. is a supervised machine learning algorithm used for regression or classification. Although it is more frequently employed for classification, regression may also benefit greatly from its use. In essence, SVM identifies a hyper-plane that marks the division of different data types. In two dimensions, this hyper-plane is nothing more than a line.[28, 29]The figure1.7 below is an example of Linear SVM

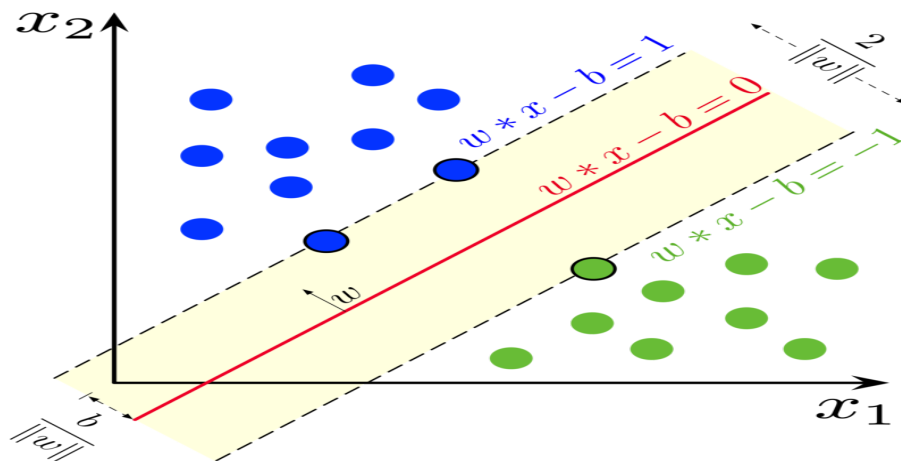


Figure 1.7: Linear SVM[32]

1.5 Conclusion

Finally, knowledge-based AI systems, like expert systems, have shown to be useful tools. They offer quick processing capabilities and lessen the demand for hard manual effort. These systems have several advantages, but obstacles like the high maintenance costs make it difficult for them to be widely adopted. Contrarily, machine learning has become the leading field of AI, bridging the divide between statistical impacts generated from data-driven learning theory and biological influences such as neural networks. The next chapter will discuss deep learning, which largely focuses on massive neural networks with specific capabilities for processing different kinds of input, such as pictures, sounds,..etc.

Chapter 2

Image classification by deep learning

The classification of images is a well-known issue in the domains of image processing, computer vision, and machine learning. We introduce the fundamental concepts of deep learning-based image classification in this chapter.

2.1 introduction

Due to its potential advantages, such as feature extraction and data classification issues, deep learning has attracted increasing attention in areas such as automatic speech recognition, computer vision, natural language processing, drug discovery toxicology, audio recognition, bioinformatics, and automatic driving of vehicles. The entire potential cost advantages for maintenance and renovation operations are increasing due to the dynamic nature of the research field and its many applications. The machine learning methods used in deep learning, which is a pervasive technology, help to model high-level abstract views of data by incorporating complicated structures into processing layers.

2.2 Image classification

Using a set of predetermined rules, image classification classifies and labels groups of pixels or vectors in an image. The categorization law may be created using one or more spectral properties or textural features. The classifications "supervised" and "unsupervised" are two broad ways. Creating a system that can automatically classify images is the goal of image classification. Thus, this system enables the execution of a task requiring skill, which can be costly to acquire for a person owing, in particular, to physical limits like focus, weariness, or the amount of time needed to process a huge volume of visual data. [33]

2.2.1 Definition

The act of categorizing a picture entails assigning it to a class, indicating its relationship to other images, and distributing it in accordance with previously defined categories. In general, a person can easily recognize a picture because throughout the course of his life, he has learned skills that enable him to adjust to differences brought on by various acquisition settings. For instance, he can distinguish an object in a variety of orientations that is partially obscured by another from a distance and under different lighting conditions. [33]

2.2.2 Principles of classification:

- **Supervised learning.** The objective of the supervised classification is mainly to define rules allowing to classify objects in classes from the qualitative or

quantitative variables characterizing these objects, all the data are labeled and the algorithms learn to predict the result of the input data.[34, 35]

There are the most important algorithms in supervised learning:

- **K-Nearest Neighbor** A supervised classification approach is the k-nearest neighbor algorithm. It uses a large number of labeled points to teach itself how to label new ones. In order to label a new point, it considers the labeled points that are its immediate neighbors and asks those neighbors to cast their votes. Therefore, the label for the new point is whichever label the majority of the neighbors use. Here, k is the number of neighbors that K-Nearest Neighbors examines. Because the goal is to attempt to categorize a point based on the recognized categorization of other points, it is supervised. Since KNN averages the k-nearest neighbors, it is often resistant to noisy data. The figure 2.1 below is an example of KNN

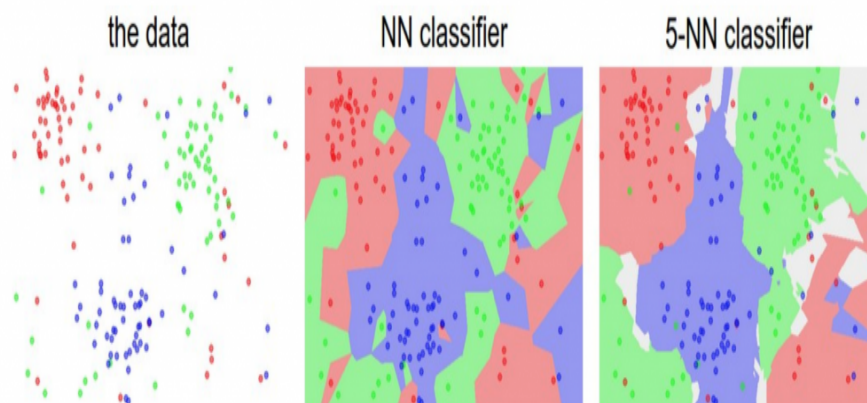


Figure 2.1: KNN algorithms application example[32]

- **Unsupervised learning**

- **K-means** Due to its ease of use, the k-means method is the most well-known unsupervised clustering technique. It divides an image's data into K clusters. K-means only generates a single level of clusters, in contrast to other so-called hierarchical approaches that produce a "cluster tree" structure to represent groups. The technique outputs a data partition where each cluster's items are as close to one another and as far away from each other's objects as is physically possible. The items and the centroid of each cluster within the partition serve as identifiers. The k-means method iteratively reduces the total distance between each item and the cluster centroid. Since the centroids' original positions affect the outcome, the procedure must be optimized by initially putting

the centroids as far apart as possible. Until the total can no longer drop, K-means rotates the objects in the cluster. If the right K value for the number of clusters is selected, the outcome is a set of compact and distinct clusters. The main steps of the k-means algorithm are:

1. Random choice of the initial position of K clusters.
 2. Re-Assign objects to a cluster according to a criterion of distance minimization (usually according to a measure of Euclidean distance).
 3. Once all the objects have been placed, recalculate the K centroids.
 4. Repeat steps 2 and 3 until no more reassignments are made. K-means is a simple algorithm that has been adapted to many problem domains[35].
- **Fuzzy c-means** A fuzzy unsupervised classification algorithm is fuzzy C-Means (FCM). He incorporates the idea of a fuzzy set into the creation of classes by using the C-means algorithm: each point in the dataset has a specific degree of membership in each cluster, and each cluster is identified by its centroid. It utilizes a criteria of reducing intra-class distances and maximizing interclass distances, like other unsupervised classification methods, but grants a specific level of membership to each class for each pixel. This approach creates the classes by an iterative process by minimizing an objective function, however it requires previous knowledge of the number of clusters. As a result, it enables the creation of a fuzzy partition of the picture by assigning a membership [36]degree (between 0 and 1) to each pixel for a certain class. The cluster with the highest degree of membership is the one with which a pixel is related. The Fuzzy C-means algorithm's primary phases are:
1. The arbitrary fixation of a membership matrix
 2. The computation of class centroids.
 3. The readjustment of the membership matrix according to the position of the centroids.
 4. Calculation of the minimization criterion and return to Step 2 if there is no convergence of criteria.
- **Support Vector Machine** Finding a hyperplane in an N-dimensional space (where N is the number of features) that clearly classifies the data points is the goal of the support vector machine method. There are several feasible hyperplanes that might be chosen to split the two groups of data points. The figure 2.2 below is an example of Support Vector Machine

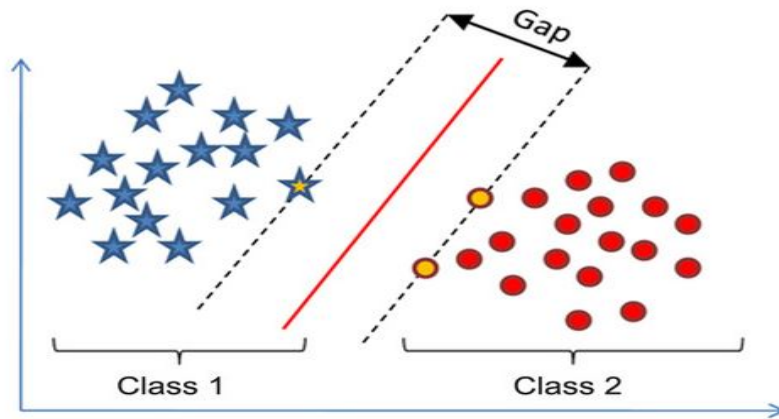


Figure 2.2: Fuzzy c-means algorithms example [37].

Our goal is to identify a plane with the biggest margin, or the greatest separation between data points for both classes. Maximizing the margin distance adds some support, increasing the confidence with which future data points may be categorised. [38]

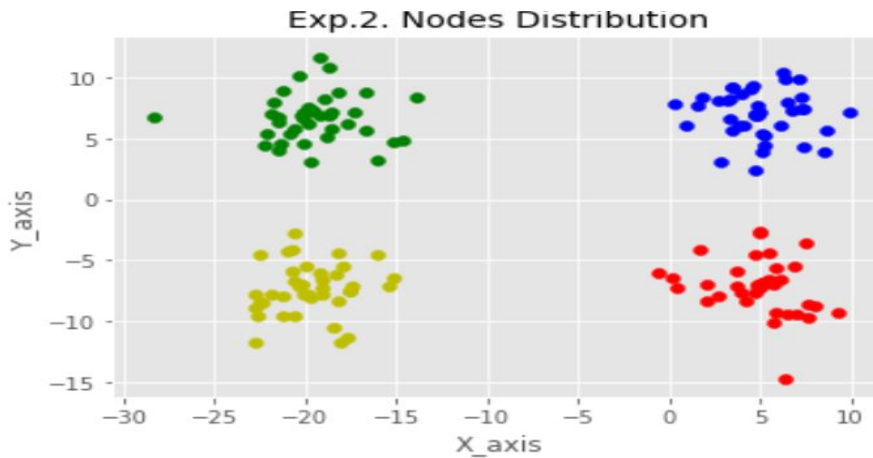


Figure 2.3: SVMs application example [39].

2.2.3 Medical image classification

Imaging modalities generate medical picture data. The challenge in this subject is to detect and comprehend which portions of the human body are afflicted by a given disease from the categorization of the extracted image in a similar pattern[40]. In order to recognize and characterize objects, we employ the chosen characteristics of picture representation that result from feature selection. Features and measurements may also be employed in the analysis of medical imaging to segment regions and extract relevant structures. The results are then inter-

preted using knowledge-based models and classification techniques. [41]. Feature and image classification techniques namely:

- **Statistical Classification Methods** These techniques fall into two categories: supervised and unsupervised procedures. K-means and fuzzy clustering are two examples of unsupervised techniques that group the data depending on how they are separated in the feature space. A supervised approach, on the other hand, requires training data, test data, and a class label in order to categorize the data. It also includes probabilistic techniques like the closest neighbor and Bayesian classifier. [41].
- **Rule-Based Systems:** The system utilizes a set of rules to analyze a feature vector stored in a database and trigger specific actions based on predefined conditions. The rules are composed of condition premises and actions determined by expert knowledge. [41]. This system uses a rule-based methodology, where many rules interact to direct the analytic process and update the database as required.
- **Neural Network Classifiers**

Artificial neural networks have been utilized for a variety of tasks, including feature classification, object identification, and picture interpretation. These networks include paradigms including back-propagation, radial basis functions, associative memory, and self-organizing feature maps. When dealing with uncertainty and imprecise data is crucial, fuzzy system-based techniques have occasionally been combined with neural networks to enhance performance [41, 42].

2.3 Deep learning classification

2.3.1 Neural recent work

In addition to serving as the fundamental unit of intellect, neurons also function as a multicellular organism's message-passing and control system. Neurons have a body, known as the soma, from which one or more processes, such as dendrites or axons, might branch out. The processes end in synapses and transport signals to and from the soma. There is only one cell in the neuron, just like each other cell in the body. The main distinction between neurons and the majority of other cells in the human body is that neurons do not renew. It possesses DNA code and is produced in a manner similar to that of other cells. [43]

2.3.2 Different types of Neural Networks

- **Feedforward Neural Network** This neural network is one of the simplest form of ANN, where the data or the input travels in one direction. The data passes through the input nodes and exit on the output confidence.

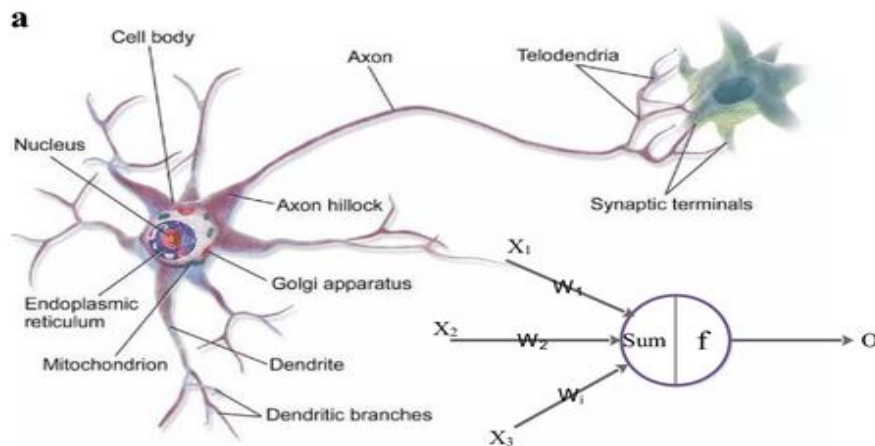


Figure 2.4: Typical biological neuron [42].

nodes. The hidden layers may or may not be present in this neural network. It has a front propagated wave and no back propagation, to put it simply, by applying a classifying activation function most of the time. In computer vision and speech recognition, where it is difficult to categorize the target classes, feed forward neural networks are used. These neural networks are simple to maintain and responsive to noisy data. [44]

- **Radial basis function Neural Network**

The theory of function approximation is where the concept of RBFNs originated. The Euclidean distance between the point being assessed and the center of each neuron is calculated, and the weight (effect) of each neuron is then determined by applying a radial basis function (RBF), also known as a kernel function, to the distance. The radius distance serves as the function's parameter, thus the term "radial basis function." In other words, RBFs are local receptors, and the output is dependent on the distance between the input and a certain stored value. Figure 6 shows a common radial basis function neural network design as an example.

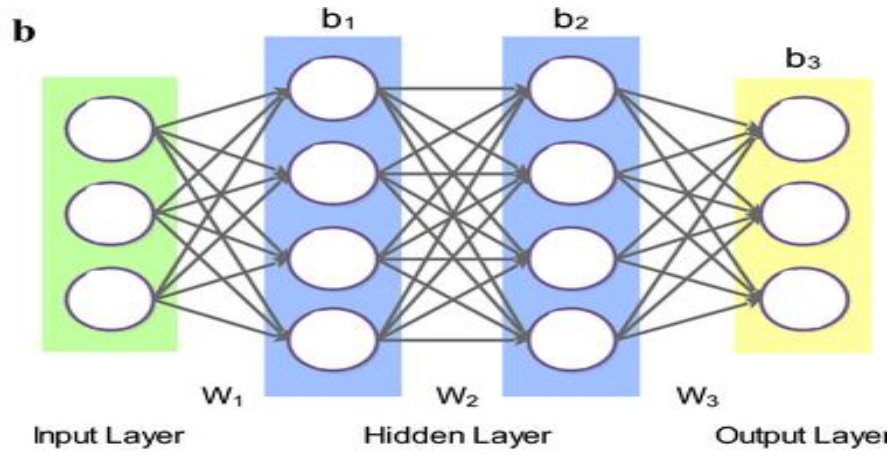


Figure 2.5: Feedforward neuron network [42].

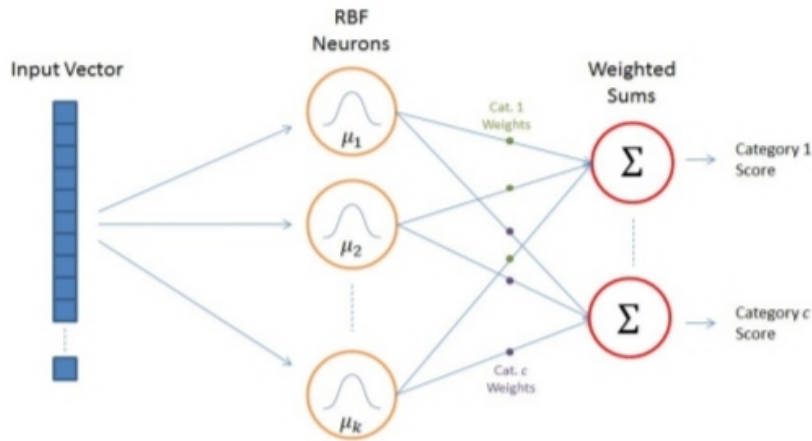


Figure 2.6: Architecture of an RBF Network [45].

- **Kohonen Self Organizing Neural Network**

Adapting to input patterns, the Kohonen Self-Organizing feature Map (SOM) is a neural network. Self-organization is a quality that is attained through competitive learning. The fundamental concept of competitive learning is that each cycle of learning is preceded by a competition. A winning processing element in the competition is chosen according to specified criteria. The standard is to reduce the Euclidean distance between the input and weight vectors. The weight vector of the winning processing element is then modified in accordance with the applied learning law. SOM varies from simple competitive learning in that it adapts the winning processing unit's neighbors as well, rather than just the winning processing element itself. Based on the usage of the neighborhood of the winning processing element, the self-organization feature of SOM [46].

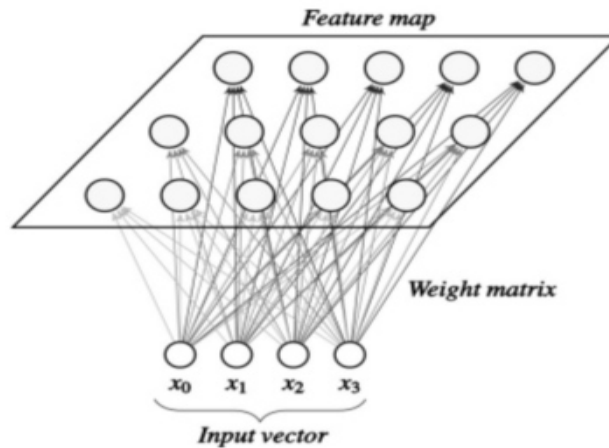


Figure 2.7: Architecture of a SOM network
[47]

- **Recurrent Neural Network (RNN)**

The result from the previous step is supplied as input to the next step in recurrent neural networks (RNNs), a form of neural network. Traditional neural networks have inputs and outputs that are independent of one another, but there is a requirement to remember the previous words in situations when it is necessary to anticipate the next word in a phrase. As a result, RNN was developed, which utilized a Hidden Layer to resolve this problem. The Hidden state, which retains some information about a sequence, is the primary and most significant characteristic of RNNs. RNNs have a memory that retains all data related to calculations. It executes the same action on all of the inputs or hidden layers to generate the output, using the same settings for each input. Unlike other neural networks, this one has less complicated parameters.[38] RNNs can be used in a lot of different places. For example, at language modelling and generating text, machine translation, speech recognition, generating image descriptions and video tagging

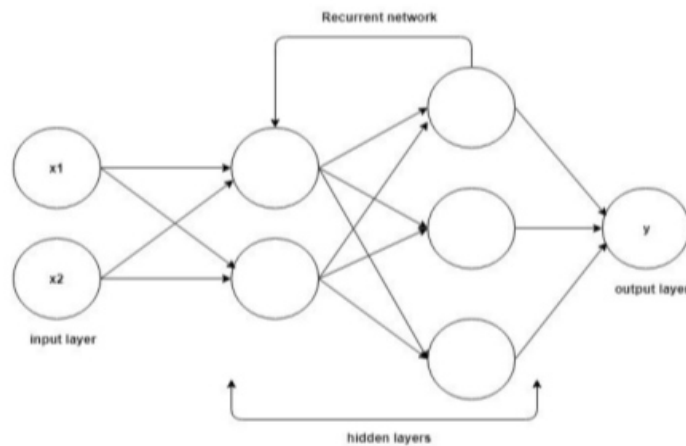


Figure 2.8: Architecture of a Recurrent Neural Network [48]

- **Convolutional Neural Network (CNN)** A particular kind of neural network for processing data with a grid-like architecture is the convolutional neural network (CNN). Data of the picture type may be compared to a 2D grid of pixels. Convolutional networks have achieved great success in real-world settings. Convolution is a special linear operation. Convolutional networks are simply networks of neurons that use convolution instead of matrix multiplication in at least one of their layers. They have wide applications in image and video recognition, recommendation systems and natural language processing.

2.4 Previous research

In previous studies by **Timo Gaiser**, **Jakob Nikolas Kather** and their colleagues on the classification of digital pathological images using deep learning: One of the most important steps in the development of digital pathology is the recognition of different tissue types within histological images. The challenge has most frequently been resolved via tissue analysis, particularly to determine the tumor to stroma ratio in tissue samples. Despite the fact that histological pictures frequently comprise more than two different kinds of tissue, multilayer categorization has received little attention in the literature. Notably, colon cancer, a highly metastatic class of malignancies, has very few reported outcomes of tissue aggregation in a multilayered instance. In this study, 5,000 histological images of human colorectal cancer tissues, including eight different forms of cancer, were shown. This dataset was used to assess how well different texture descriptors and classifiers perform in classification tasks. As a result, a classification strategy that outperformed conventional methods turned out to be the ideal. With an accuracy of 87.4% across eight classes, this advance has increased the cutting-edge accuracy

for distinguishing tumor stroma from 96.9% to 98.6% while also setting a new standard for differentiating multiple tissue classes.[49]

In a previous study by **Ashlin Ghosh** and **Srinath Jayachandran** on using deep learning to classify digital disease images: histopathology, or tissue microscopy, is an important diagnostic process for colorectal cancer. Pathologists often analyze the histology to determine the various tissue types, paying close attention to the tumor-to-stroma ratio. Deep transfer learning is used in this study to automatically classify tissue samples for colorectal cancer. To get better outcomes, this technique employs a one-cycle strategy for discriminating fine-tuning and structure-preserving color normalization. The study also offers visual justifications for the tissue categorization conclusions reached by the deep neural network. The research also investigates the use of SqueezeNet's memory-efficient architecture for deployment on resource-constrained devices, resulting in an impressive test accuracy of 96.2%. [50]

2.5 Conclusion

Deep learning attracts many scholars and has already made several medical contributions. The next chapter's topic is the proposed approach specification for deep learning model network classifying Colorectal cancer disease.

Chapter 3

Proposed deep learning model

3.1 Introduction

Many carefully planned designs have been developed in the field of picture classification, each one intended to be used in a certain order to enhance its effectiveness. To help these models retrieve spatial information from the features they encounter, many techniques have been developed. These techniques are designed to help the model concentrate its attention throughout the categorization process on the regions that are most important. We explore the specific approach we have used in this chapter, exploring its complexities and illuminating the key elements of the model's design. Through this investigation, we want to fully explain the guiding concepts at play and clarify the inner workings of this novel strategy.

3.2 TRANSFER LEARNING

What Is Transfer Learning ?

A model that has been trained for one task is repurposed for another task that is related to it using the machine learning technique known as transfer learning. The knowledge from the previously trained machine learning model is transferred to a different but closely related topic during the learning transfer technique. [51], For example :

- If you can ride a bike, you can easily learn to drive a car.
- You can pick up physics rapidly if you know maths.
- If you train the basic classifier to determine if a photo has a bag, you may use the training data to detect other objects like sunglasses.

What is the Process of Transfer Learning?

There are two ways that are often used:

1. Develop Model Approach
2. Pre-trained Model Approach

Since the Pre-trained Model Approach is so popular in the deep learning community, we will only focus on it in this context.

Pre-trained Model Approach

1. Chooses Source Model: A pre-trained source model is chosen from the available models. Many research institutions provide models based on enormous

and challenging datasets, which may be considered in the selection of candidate models. [52, 53]

2. Reusing a model: The pre-trained model can be used as the basis for a model for a second task that interests you. Depending on the modeling strategy used, this can include using the entire model or only some parts of it. [52, 53]
3. Tune Model: Depending on the input-output pair data available for the task at hand, the model may need to be modified or refined. [52, 53]

3.3 Residual Neural Network (ResNet)

Many computer vision applications are built on the classic neural network known as residual networks. This model triumphed in the 2015 ImageNet competition. ResNet changed the game by enabling us to train extremely deep neural networks with more than 150 layers.[54]

3.3.1 Motivation

Researchers have identified an intriguing pattern in convolutional neural networks (CNNs): the greater the depth of the network, the higher its potential for achieving strong performance because deeper models possess more parameters, enhancing their adaptability to various data distributions. However, there is a critical point at which increasing depth yields diminishing returns, resulting in a performance decline—a challenge known as the "vanishing gradient" problem. The vanishing gradient problem arises in extremely deep networks when gradients used in the backpropagation process become exceedingly small as they undergo multiple multiplicative operations during the chain rule calculations. Consequently, the updates to the network weights virtually halt, impeding the learning process. The VGG architecture is an illustrative example of this limitation, where increasing network depth negatively impacted its ability to generalize effectively. Residual Networks (ResNets) enable gradients to flow directly from later layers to earlier filters through skip connections. This architectural innovation effectively mitigates the vanishing gradient problem, facilitating the training of significantly deeper networks and ultimately enhancing their performance [55].The figure3.1 below shows the difference between two neural networks(VGG19 ,ResNet34)

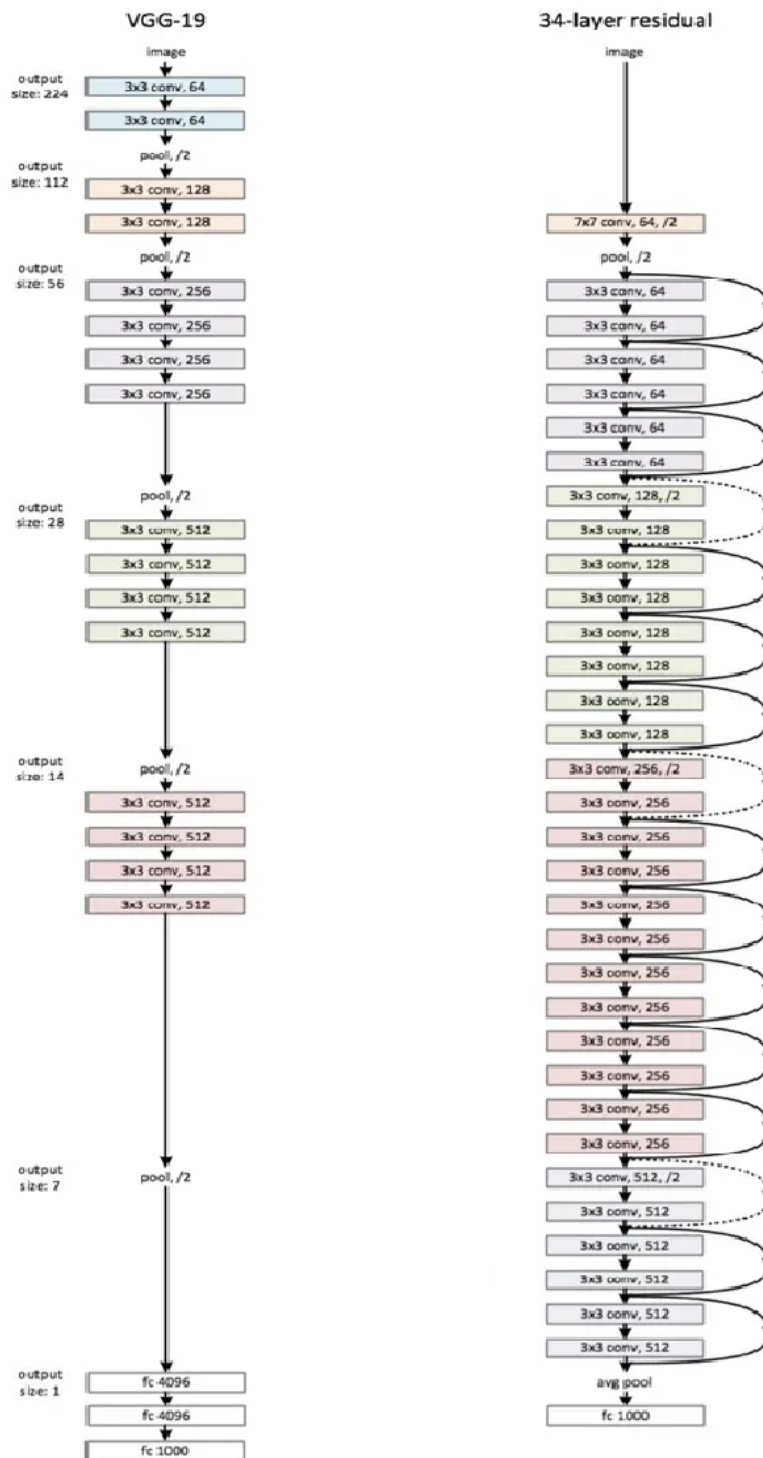


Figure 3.1: VGG19 vs ResNet34 [55]

3.3.2 Residual Block

To tackle the pervasive issue of vanishing or exploding gradients in deep neural networks, ResNet pioneers the concept of Residual Blocks. As the figure 3.2 below shows

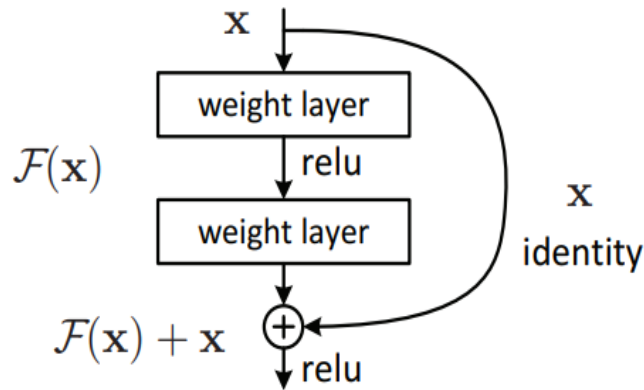


Figure 3.2: skip connections
[55]

These blocks implement a technique known as skip connections, which establish connections between the activations of one layer and those of subsequent layers while bypassing certain intervening layers. ResNet's essence lies in assembling these residual blocks, effectively constructing a neural network by stacking them together.

3.4 fully convolutional network FCN

A sort of neural network design called a Fully Convolutional Network (FCN) is mostly utilized for semantic segmentation [56, 57]. Fewer parameters and faster training are the results of FCNs' exclusive usage of locally linked layers, such as convolution, pooling, and upsampling, as opposed to dense layers [56]. FCNs can handle different picture sizes as long as all connections are local. The network is made up of an upsampling path for localization and a downsampling path for extracting and interpreting the context. To regain the fine-grained spatial information lost in the downsampling process, FCNs additionally use skip connections. The key features of an FCN include:

- **Parameter-rich:** Fully connected layers are parameter-rich because they pass on parameters without any kind of "spatial" reduction. [58]
- **Upsampling via Deconvolution:** FCNs use transposed convolutional layers to transform the height and width of intermediate feature maps back to those of the input image

- Fusing the Output: FCNs use a fusion operation to combine the results of multiple models to make the prediction more accurate

The first FCN was proposed in 2014 and has since become one of the most basic papers for semantic segmentation using FCN. The figure 3.3 below shows the architecture of FCN model.

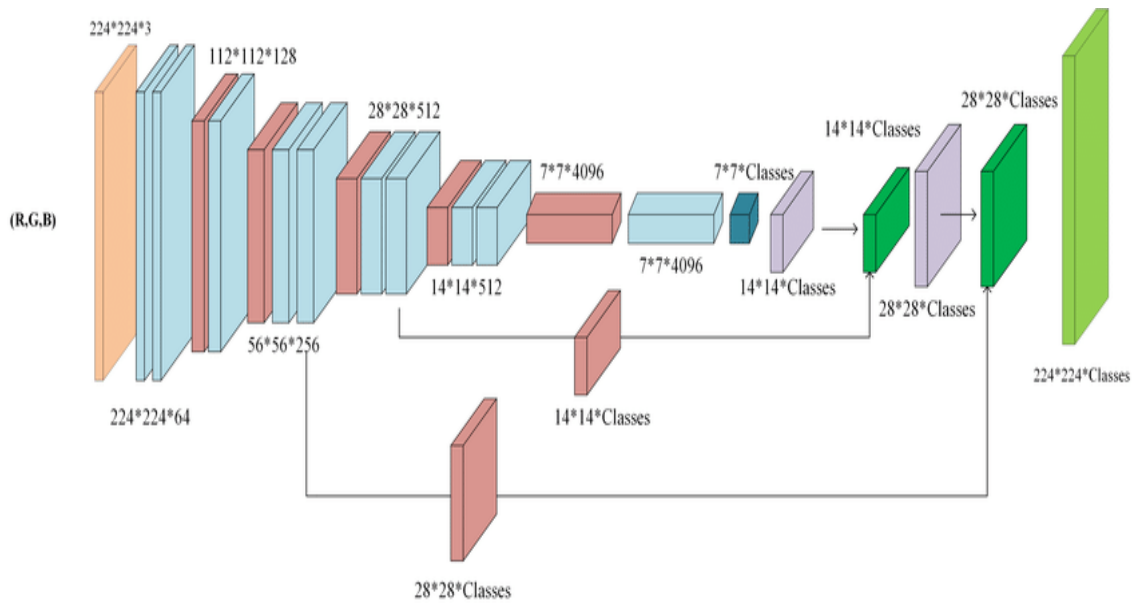


Figure 3.3: Fully convolutional neural network architecture (FCN-8). [59]

3.5 Proposed Method

Colorectal cancer is the third most frequent disease in the world, accounting for around 10% of all cancer occurrences and being the second biggest cause of cancer-related deaths. It mostly affects older people, with the majority of cases occurring in adults aged 50 and above[60]. Resnet and FCN networks are among the greatest solutions currently available because of their excellent performance in categorizing digital pathology pictures. [61, 62]

3.5.1 Overall Architecture

The proposed approach centers around a type of neural network known as Fully Convolutional Network (FCN), based on the Encoder-Decoder principle, followed by the design of a head map for image classification.

In the encoder path of the FCN designed for feature extraction, we use ResNet-50. ResNet-50 performs image encoding, significantly decreasing dimensionality and capturing crucial features within bottlenecks.

We also reduce the dimensionality of the original image to minimize the feature space's dimensions while preserving essential information, enhancing the deep learning model's speed and performance [63].

In the recap, we follow these steps :

First, we convert the input image from color to gray level using the color divergence principle[64] to reduce the dimensions of the feature space of the data without losing as much relevant information as possible.

Then, using the Imagenet dataset as the foundation, we built the encoding path of the FCN using ResNet-50. After that, before the decoding path of the FCN, we add the attention module. The decoding path consists of upsampling step performed by the transposed 2D convolution of the feature map.

Finally, we add the head map, the part responsible for classification in the network.

3.5.2 Dimensionality reduction

The RGB cube represents the image's three dimensions: red, green, and blue hues. The approach defined in [64] employs standard deviations to locate and present the potential color uniformity area related to the item. To achieve this, we determine the Euclidean distance matrix between each RGB color pixel in the original image and each of the RGB cube's eight vertices. We determine eight distances (between the pixel and a cube corner) for each pixel and then compute the standard deviation of these distances to get the singular value known as SDV for each pixel instead of three RGB values.[64]

3.5.3 ResNet50

Convolutional neural networks (CNNs) are used by the deep learning model ResNet50 to carry out image recognition tasks. The ResNet design, on which it is based, introduces the idea of residual blocks to deal with the issue of disappearing gradients and enable deeper networks. One max pooling layer, one average pooling layer, and 48 convolutional layers make up ResNet50's 50 layers. It can carry out 3.8 billion floating point operations per second (FLOPs) and has around 25.6 million parameters. Many computer vision applications, including image categorization, object identification, face recognition, and others, employ ResNet50.[65, 66, 67]

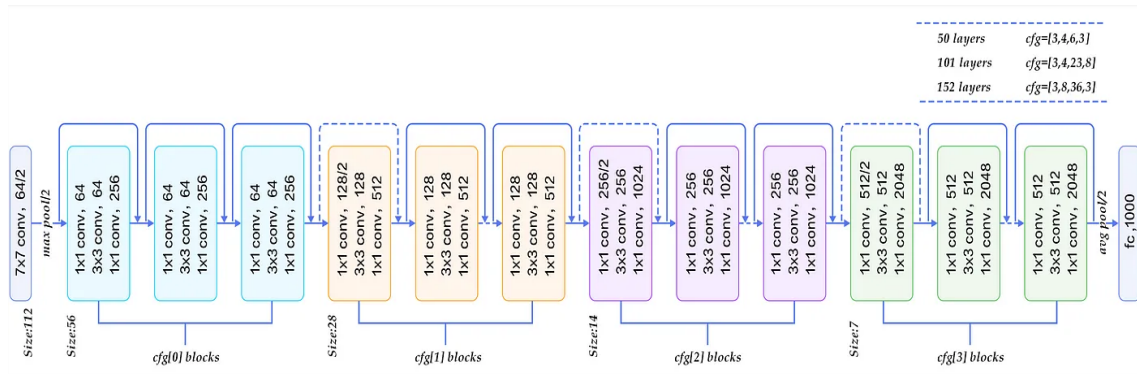


Figure 3.4: ResNet-50 Architecture. [59]

3.5.4 Fully Convolutional Network

Similar to U-Net, this architecture is designed for tasks like image segmentation. It captures information from pooled representations and gradually refines it through convolutional layers. The final output plays a role in the broader image-processing task.

The proposed FCN architecture consists of the encoding path defined by the ResNet-50 as a feature extractor. In the attention module in which we perform both the 2×2 max and average pooling to the output of the ResNet-50, we concatenate the result of max and average pooling; this attention mechanism is called spatial attention and defined in [68]. This step is followed by applying dropout regularization to prevent overfitting and enhance the deep neural network’s capacity for generalization.

The decoding path consists of a 2×2 upsampling step performed by the transposed 2D convolution of the output of the spatial attention module, then by a 2×2 convolution that reduces the number of feature channels to half. The 2×2 convolution is a common practice to control the dimensionality of the feature maps and make them more manageable.

We repeat these two steps (upsampling followed by channel reduction) iteratively until the feature maps reach the size of the original input image. This process is typical in the decoding path of many convolutional neural network architectures, especially in those designed for tasks like image segmentation or image generation, where the goal is to recover fine-grained details and spatial information from lower-resolution feature maps.

3.5.5 Head Part

To ensure accurate image categorization and mitigate potential issues, we carefully choose a specific set of layers. It's essential to note that the Head Part of the network takes on the responsibility of classification.

The Head Part encompasses fourteen layers, each with distinct characteristics:

1. A Flatten Layer, which processes the output from the Fully Convolutional Network (FCN).
2. A Dropout Layer with a rate of 0.5, employed for regularization.
3. A Dense Layer utilizing the Rectified Linear Unit (ReLU) activation function.
4. Another Dense Layer featuring 8 units and employing the softmax activation function.

These specific layers and their configurations are strategically chosen to perform the classification task effectively and address potential challenges.

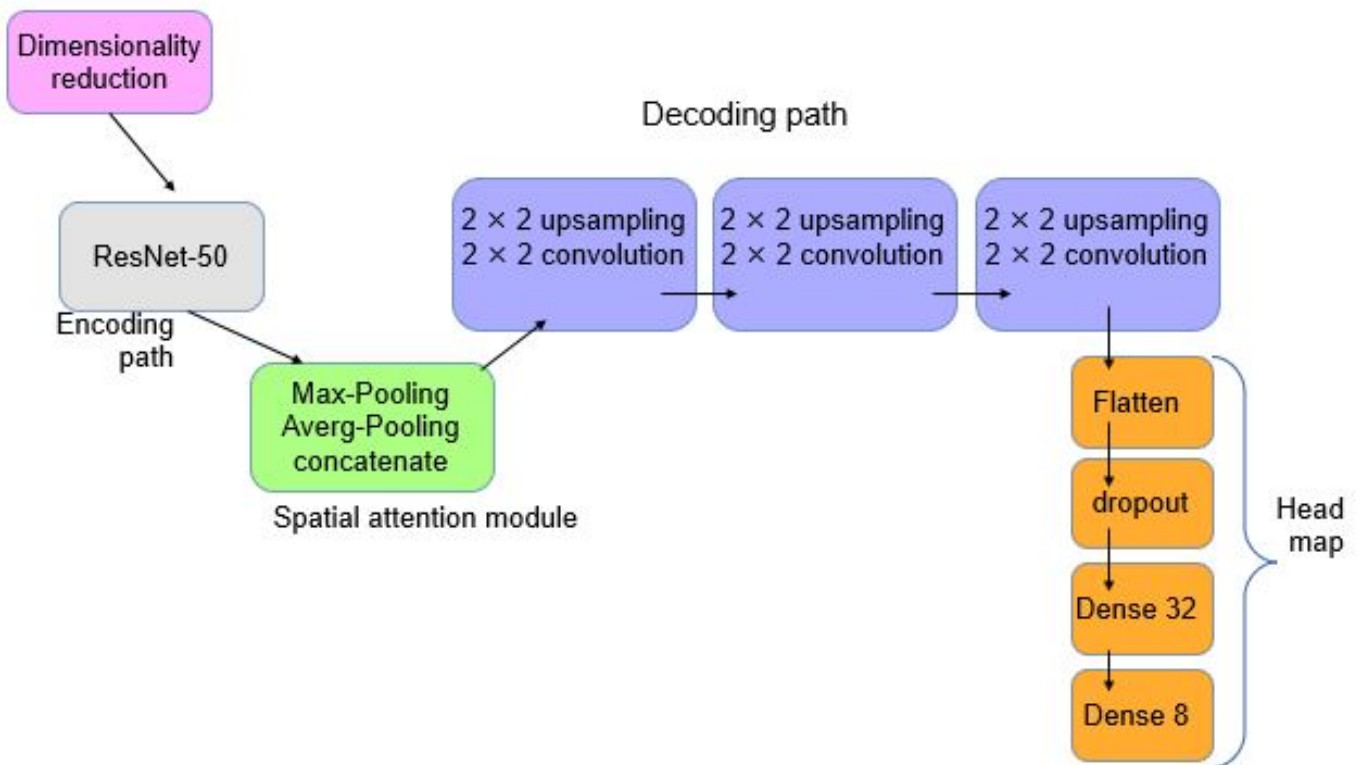


Figure 3.5: Model Architecture

3.6 Conclusion

In this current chapter, we described the model's architecture and explained the role of each of its parts. In the next chapter, we will detail the implementation of the proposed approach.

Chapter 4

Implementation and experiment result

4.1 Introduction

After thoroughly describing the issue we're trying to solve and discussing the theoretical ideas and approach used in the precedent chapter, we'll turn our attention to the practical side of the subject. We will discuss significant findings and our own experiences in this area.

4.2 Colonrectal cancer

The dataset contains two common publicly available collections which are[69]

- Kather texture 5000 images dataset [70]
- Kather texture 10 images dataset[70]

The total number of images we obtained from the two groups is 5010 images, which we divided into three groups as follows

	Colonrectal cancer	Total	Percentage
Test	500	10	10%
Train	4010	4010	80%
Val	500	1000	10%
Total	5010	5010	100%

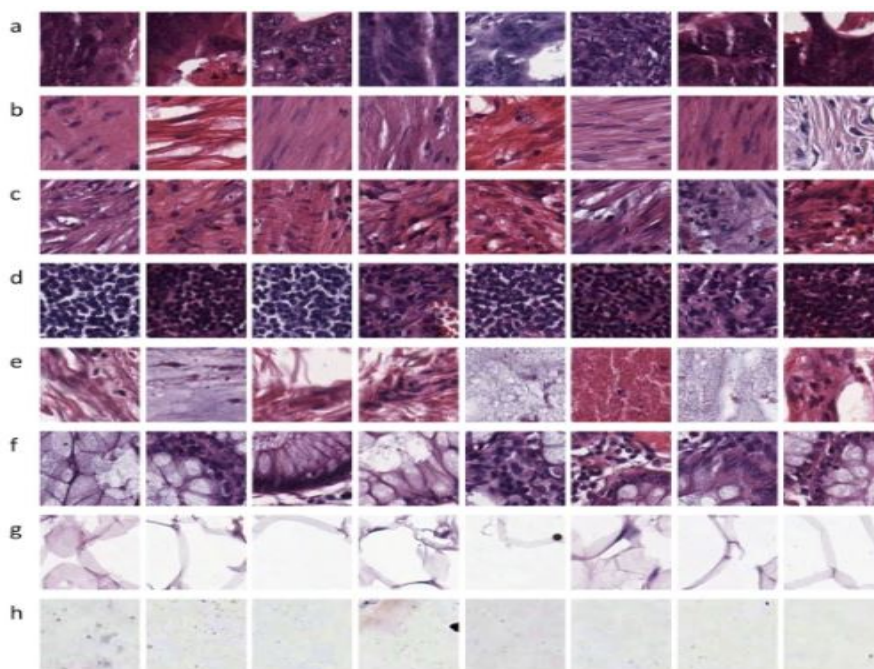


Figure 4.1: Example OF Dataset with 8 classes [49]

4.3 Evaluation criteria

We have relied on some well-known metrics to evaluate the performance of our model and they are (accuracy, precision, recall, f1-score, AUC)[71].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - score = \frac{2}{precision^{-1} + recall^{-1}} \quad (4.4)$$

4.3.1 Area under the curve (AUC)

The chance that positive instances will be accurately identified from those that are wrongly classified as negative is represented by the area under the curve, and the greater the area under the curve, the better.[71].

$$Area = P(X_1 > X_0) \quad (4.5)$$

where X_1 and X_0 are the score for the positive and negative instance respectively[71].

4.4 Software environment

Although there are many alternative programming languages and tools available in this industry, we have decided to utilize Python along with certain well known libraries and frameworks.

4.4.1 Python

Python is a dynamically semantic, object-oriented, interpreted programming language. Python's straightforward syntax emphasizes readability and thus lowers the cost of programming due to its high-level built-in data structures, dynamic typing, and dynamic binding. It is also well suited for use as a scripting or glue language to connect existing components together. maintenance[72].

Python's support for modules and packages promotes program modularity and code reuse. The Python interpreter and its substantial standard library are freely distributable and accessible in source or binary form for all major platforms.[72].

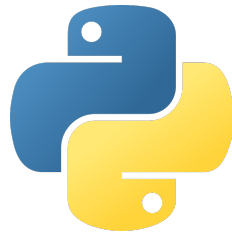


Figure 4.2: Python Logo[73]

4.4.2 Tensorflow

TensorFlow is an end-to-end open source machine learning platform. It offers a vast, adaptable ecosystem of tools, libraries, and community resources that enables academics to advance the field's state-of-the-art and developers to quickly create and deploy ML-powered applications.[74].



Figure 4.3: Tensorflow Logo[74]

4.4.3 Scikit-Learn

A Python package called Scikit-learn integrates a variety of cutting-edge machine learning methods for medium-scale supervised and unsupervised applications. This package focuses on utilizing a general-purpose high-level language to make machine learning accessible to non-specialists.[75].



Figure 4.4: Scikit-Learn Logo[76]

It has minimum dependencies and is given under the simplified BSD license, enabling its usage in both academic and commercial settings. Emphasis is placed on usability, performance, documentation, and API consistency.[75].

4.4.4 Pandas

Built on top of the Python programming language, pandas is an open source data analysis and manipulation tool that is quick, strong, versatile, and simple to use.[77].



Figure 4.5: Pandas Logo[77]

4.4.5 OpenCV

Being a BSD-licensed product, OpenCV makes it simple for businesses to use and modify the code. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. It was created to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. [78].

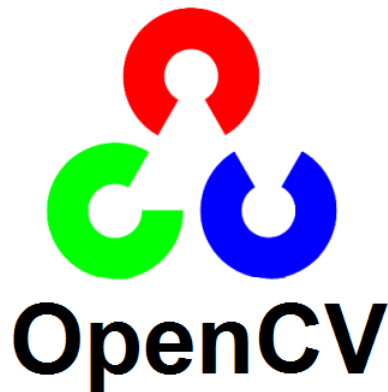


Figure 4.6: OpenCV Logo[78]

4.4.6 Matplotlib

For building static, animated, and interactive visualizations in Python, check out Matplotlib. Matplotlib makes difficult things possible and simple things easy.[79].



Figure 4.7: Matplotlib Logo[79]

4.4.7 Numpy

It is a Python library that offers a multidimensional array object, various derived objects (like masked arrays and matrices), and a variety of routines for quick operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more. NumPy is the fundamental package for scientific computing in Python.[80].



Figure 4.8: Numpy Logo[80]

4.5 Computing Resources

Anyone may write and execute arbitrary Python code through the web using Google Research’s Colaboratory, or ”Colab” for short. Data analysis, training, and machine learning are three areas where Colab excels.[81].



Figure 4.9: Colab Logo[81]

Technically speaking, Colab is a hosted Jupyter notebook service that allows free access to computer resources like GPUs without the need for installation. [81].

4.6 Implementation

The following configurations are used with the suggested method to ensure proper model training: Adam Optimizer, Learning Rate of 1e-3, Batch Size of 64, Binary Crossentropy Loss Function, Cross Validation with 8 folds, Categorical Accuracy Metric, and finally val-acc and val-loss callback Thresholds to stop training at specific thresholds and save weights.

4.6.1 Augmentation and Preprocessing

There are several ways to work with the image data generator class, including (flow from directory, flow from dataframe, flow), as well as the Image Data Generator class allows us to train the model on a large number of images by loading them directly from the hard disk, and this is advantageous because it makes the

model train more effectively and avoids the problem of overfitting. The number of training pictures may be increased by using conditional generative adversarial networks (GANs) to create modified versions of our initial photos. [82].

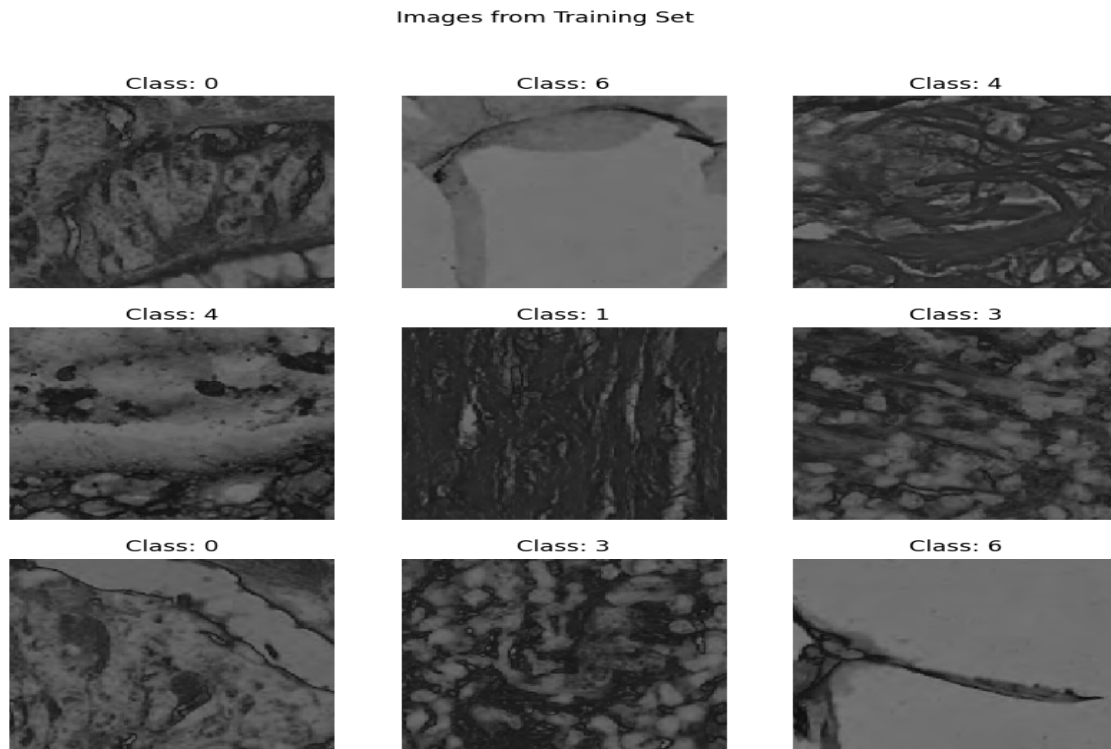


Figure 4.10: After perform augmentation

4.6.2 Implementation of the proposed method

Code Listing 4.1: Import necessary packages

```

from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, Conv2DTranspose, concatenate
from tensorflow.keras.applications import ResNet50
from tensorflow.keras import layers
from tensorflow.keras.layers import InputLayer, BatchNormalization, Dropout,
    Flatten, Dense, Activation, MaxPool2D,
    Conv2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
import matplotlib.pyplot as plt
from sklearn import metrics
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import skimage.io
import os
import tqdm
import glob
from tensorflow.keras.optimizers import Adam
from keras.models import Sequential, load_model
    
```

First we import the necessary packages required for the code Listing 4.1.

Code Listing 4.2: mount drive to google colab

```
from google.colab import drive
drive.mount('/content/drive/MyDrive/ttt')
```

The dataset is saved in a Google Drive account, we link Google Drive storage to Google Colab environment.4.2.

Code Listing 4.3: Dimensionality reduction

```
#fonction pour le calcul de la moyenne d'un vecteur
def moyenne(tableau):
    return sum(tableau, 0.0) / len(tableau)

#fonction pour le calcul de la variance d'un vecteur
def variance(tableau):
    m=moyenne(tableau)
    return moyenne([(x-m)**2 for x in tableau])

#fonction pour le calcul de l'ecart type d'un vecteur
def ecartype(tableau):
    return variance(tableau)**0.5
def sdvt_im_org(img):
    #img = (img / 127.0) - 1.0
    w, h = img.shape[:2]
    imge = np.zeros((w , h, 9), dtype='f')
    for i in range (w) :
        for j in range (h) :
            x = img[i][j]
            imge[i][j][0] = sqrt ((1+x[0])**2 + (1+x[1])**2 + (1+x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (0,0,0) du cube
                                                                    englobant l'image
            imge[i][j][1] = sqrt ((1 - x[0])**2 + (1+x[1])**2 + (1+x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (255,0,0)
            imge[i][j][2] = sqrt ((1+x[0])**2 + (1 - x[1])**2 + (1+x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (0,255,0)
            imge[i][j][3] = sqrt ((1+x[0])**2 + (1+x[1])**2 + (1 - x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (0,0,255)
            imge[i][j][4] = sqrt ((1 - x[0])**2 + (1 - x[1])**2 + (1+x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (255,255,0)
            imge[i][j][5] = sqrt ((1 - x[0])**2 + (1+x[1])**2 + (1 - x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (255,0,255)
            imge[i][j][6] = sqrt ((1 + x[0])**2 + (1 - x[1])**2 + (1 - x[2])**2 ) #
                                                                    distance entre pixel et sommet de
                                                                    coordonnee (0,255,255)
```

```
imge[i][j][7] = sqrt ((1 - x[0])**2 + (1 - x[1])**2 + (1 - x[2])**2 ) #  
                    distance entre pixel et sommet de  
                    coordonnee (255,255,255)  
imge[i][j][8] = ecartype([imge[i][j][k] for k in range (8)]) #ecart type  
                    des distances  
  
im = imge[:, :, 8]  
  
return (im)
```

We created a function to reduce the dimensions of the images while maintaining their quality. Code Listing 4.3.

Code Listing 4.4: image data generator

```

num_classes = 8
input_shape = (224, 224, 3)

train_datagen = ImageDataGenerator(rescale = (1./127)-1,
                                   validation_split = 0.2,

                                   rotation_range=5,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   #zoom_range=0.2,
                                   horizontal_flip=True,
                                   vertical_flip=True,
                                   fill_mode='nearest')

valid_datagen = ImageDataGenerator(rescale = (1./127)-1,
                                   validation_split = 0.2)

test_datagen = ImageDataGenerator(rescale = (1./127)-1,
                                   validation_split = 0.2)
train_dataset = train_datagen.flow_from_directory(directory = '/content/drive/
MyDrive/ttt/train_sdv',
                                                target_size = (224,224),
                                                class_mode = 'categorical',
                                                color_mode = 'rgb',
                                                batch_size = 32)

valid_dataset = valid_datagen.flow_from_directory(directory = '/content/drive/
MyDrive/ttt/val_sdv',
                                                target_size = (224,224),
                                                class_mode = 'categorical',
                                                color_mode = 'rgb',
                                                batch_size = 32)

test_dataset = test_datagen.flow_from_directory(directory = '/content/drive/
MyDrive/ttt/test_sdv',
                                                target_size = (224,224),
                                                class_mode = 'categorical',
                                                color_mode = 'rgb',
                                                batch_size = 32)

```

This function return objects of image data generator for the three sets (training, validation, testing), where the changes to the images on the validation set are the same as training set images, but the test images are only normalized Code Listing4.4.

Code Listing 4.5: Train the model

```

def fcn_with_resnet(input_size=(224, 224, 3), num_classes=8):
    # Load ResNet pre-trained on ImageNet
    resnet_backbone = ResNet50(input_shape=input_size, include_top=False,
                               weights='imagenet')

    resnet_backbone.trainable = False # Freeze the ResNet weights

    inputs = Input(input_size)

    # ResNet backbone
    resnet_features = resnet_backbone(inputs)

    # UNet-style decoder
    up6 = Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')(
        resnet_features)

    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(up6)
    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv6)
    sdvt1=sdvf_tensor(conv6)

    #Attention module
    pool = MaxPooling2D(pool_size = (2, 2), strides=1, padding = 'same')(sdvt1)
    avrg = AveragePooling2D(pool_size=(2, 2), strides=1, padding = 'same')(
        sdvt1)

    conc = concatenate([pool, avrg], axis = -1)
    drop5 = Dropout(0.5)(conc)

    up7 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(drop5)
    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(up7)
    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv7)
    sdvt2 = sdvf_tensor(conv7)

    up8 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(sdvt2)
    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(up8)
    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv8)
    sdvt3=sdvf_tensor(conv8)

    up9 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')(sdvt3)
    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(up9)
    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv9)
    sdvt4=sdvf_tensor(conv9)

    # Flatten and use a Dense layer as the output
    flat = Flatten()(sdvt4)
    head = (BatchNormalization())(flat)
    head1 = (Dense(128, kernel_initializer='he_uniform'))(head)
    head2 = (BatchNormalization())(head1)
    head3 = (Activation('relu'))(head2)
    head4 = (Dropout(0.5))(head3)
    head5 = (Dense(64, kernel_initializer='he_uniform'))(head4)
    head6 = (BatchNormalization())(head5)
    head7 = (Activation('relu'))(head6)
    head8 = (Dropout(0.5))(head7)
    head9 = (Dense(32, kernel_initializer='he_uniform'))(head8)
    head10 = (BatchNormalization())(head9)
    head11 = (Activation('relu'))(head10)

    output = Dense(num_classes, activation='softmax')(head11)

```

```
    return Model(inputs=inputs, outputs=output)

model = fcn_with_resnet(input_size = (224, 224, 3))

def f1_score(y_true, y_pred): #taken from old keras source code
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    recall = true_positives / (possible_positives + K.epsilon())
    f1_val = 2*(precision*recall)/(precision+recall+K.epsilon())
    return f1_val
METRICS = [
    tf.keras.metrics.BinaryAccuracy(name='accuracy'),
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall'),
    tf.keras.metrics.AUC(name='auc'),
    f1_score,
]

mcp = ModelCheckpoint('model_resnet50_fcn.h5')
model.compile(optimizer=tf.keras.optimizers.Adam(0.0001), loss='
                categorical_crossentropy', metrics=
                METRICS)

%time
history=model.fit(train_dataset, validation_data=valid_dataset, epochs = 30,
                 verbose = 1, callbacks=[mcp])

# Assuming you have a test dataset called 'test_dataset'
test_results = model.evaluate(test_dataset)
print("Test Loss:", test_results[0])
print("Test Accuracy:", test_results[1])
print("Test Precision:", test_results[2])
print("Test Recall:", test_results[3])
print("Test AUC:", test_results[4])
print("Test F1 Score:", test_results[5])
```

In this part of the source code, we create the architecture of the model and compile it, train it, and then evaluate.4.5.

Code Listing 4.6: Plot Accuracy & Loss

```

def Train_Val_Plot(acc, val_acc, loss, val_loss, auc, val_auc, precision,
                  val_precision, f1, val_f1):

    fig, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(1, 5, figsize= (20, 5))
    fig.suptitle(" MODEL'S METRICS VISUALIZATION ")

    ax1.plot(range(1, len(acc) + 1), acc)
    ax1.plot(range(1, len(val_acc) + 1), val_acc)
    ax1.set_title('History of Accuracy')
    ax1.set_xlabel('Epochs')
    ax1.set_ylabel('Accuracy')
    ax1.legend(['training', 'validation'])

    ax2.plot(range(1, len(loss) + 1), loss)
    ax2.plot(range(1, len(val_loss) + 1), val_loss)
    ax2.set_title('History of Loss')
    ax2.set_xlabel('Epochs')
    ax2.set_ylabel('Loss')
    ax2.legend(['training', 'validation'])

    ax3.plot(range(1, len(auc) + 1), auc)
    ax3.plot(range(1, len(val_auc) + 1), val_auc)
    ax3.set_title('History of AUC')
    ax3.set_xlabel('Epochs')
    ax3.set_ylabel('AUC')
    ax3.legend(['training', 'validation'])

    ax4.plot(range(1, len(precision) + 1), precision)
    ax4.plot(range(1, len(val_precision) + 1), val_precision)
    ax4.set_title('History of Precision')
    ax4.set_xlabel('Epochs')
    ax4.set_ylabel('Precision')
    ax4.legend(['training', 'validation'])

    ax5.plot(range(1, len(f1) + 1), f1)
    ax5.plot(range(1, len(val_f1) + 1), val_f1)
    ax5.set_title('History of F1-score')
    ax5.set_xlabel('Epochs')
    ax5.set_ylabel('F1 score')
    ax5.legend(['training', 'validation'])

    plt.show()

Train_Val_Plot(history.history['accuracy'], history.history['val_accuracy'],
               history.history['loss'], history.history['val_loss'],
               history.history['auc'], history.history['val_auc'],
               history.history['precision'], history.history['val_precision'],
               history.history['f1_score'], history.history['val_f1_score']
               )

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('val_loss')

```

```
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

Here we plot the accuracy and loss curves and after that save them. Code Listing.4.6.

Code Listing 4.7: Show test images after prediction

```
test_datagen = ImageDataGenerator(rescale=1.0/255)

test_generator = test_datagen.flow_from_directory('/content/drive/MyDrive/ttt/
                                                test_sdv',
                                                shuffle=False,
                                                batch_size=32,
                                                target_size=(224, 224),
                                                class_mode='categorical')

predictions = model.predict(test_generator)

# Convert dict_keys to a list to access elements by index
class_labels = list(test_generator.class_indices.keys())

fig, ax = plt.subplots(nrows=2, ncols=8, figsize=(12, 10))
idx = 0

for i in range(2):
    for j in range(8):
        predicted_class_idx = np.argmax(predictions[idx])
        predicted_label = class_labels[predicted_class_idx]
        ax[i, j].set_title(f"{predicted_label}")
        ax[i, j].imshow(test_generator[0][0][idx])
        ax[i, j].axis("off")
        idx += 1

plt.tight_layout()
plt.suptitle("Test Dataset Predictions", fontsize=20)
plt.show()
```

Finally, We make predictions using test images and display the results Code Listing4.7.

4.7 Experiments and results

The model performed well after being trained across a number of epochs, as can be seen in the figure 4.11 below.

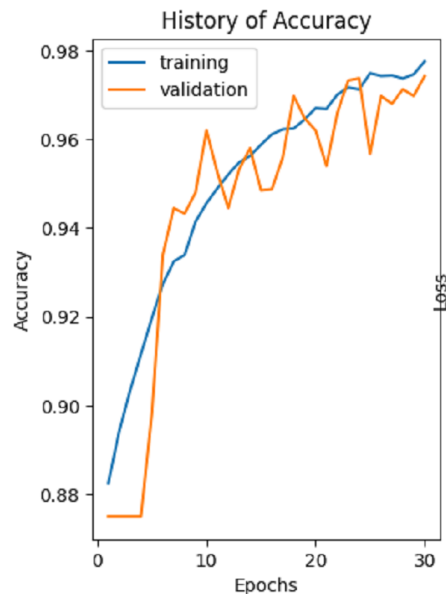


Figure 4.11: The model accuracy curve for both train and validation sets

As you can see in the above figure 4.11, the accuracy curve of both the training and validation sets are increasing well and at the end they reach excellent accuracy "0.9776 and 0.9743 respectively".

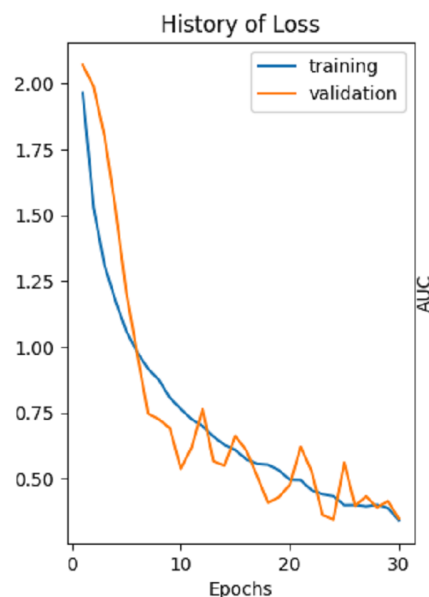


Figure 4.12: The model loss curve for both train and validation sets

Also in the above figure 4.12, the loss function curve of both the training and

validation sets are decreasing well and at the end they reach "0.3427 and 0.3489 respectively".

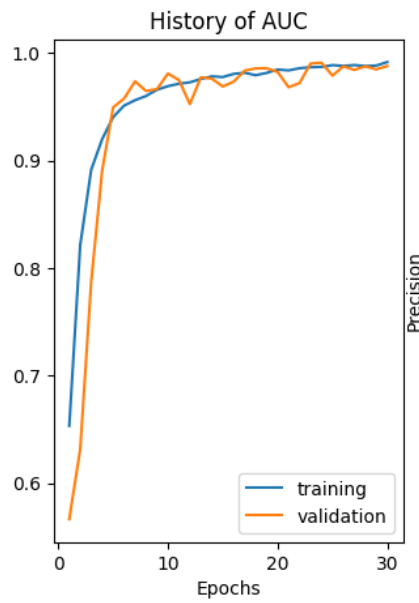


Figure 4.13: Area under the curve

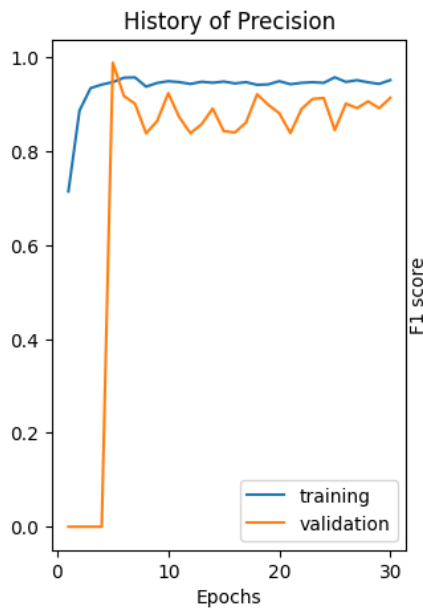


Figure 4.14: History of precision

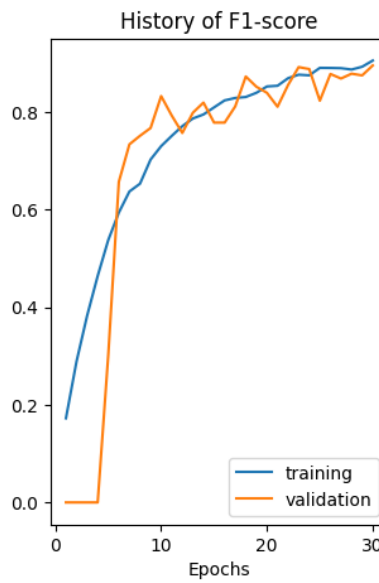


Figure 4.15: History of f1-score

Figures (4.13, 4.14 and 4.15) illustrate the history of Area under the curve AUC, precision and F1-score respectively.

Below (figure 4.16) is a recap of all used metrics.

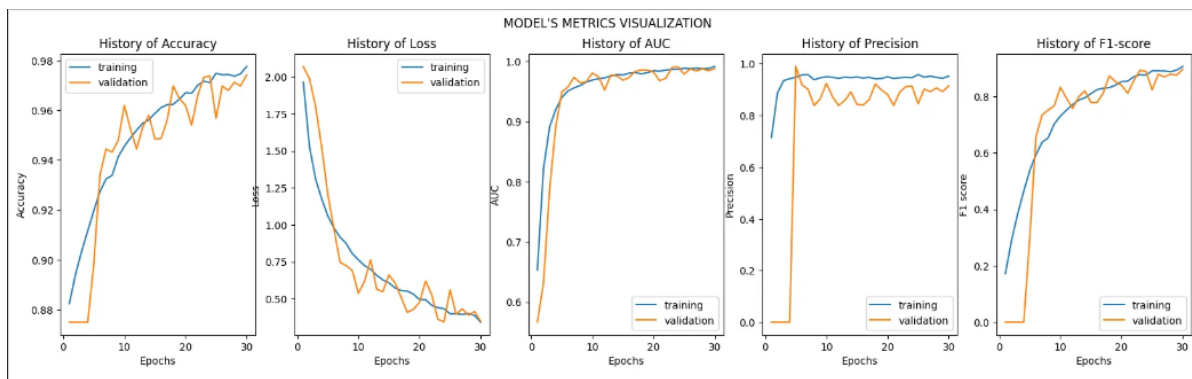


Figure 4.16: The recap of all metric's history

Figure 4.17 represents the evaluation score of all used metrics applied to the test set.

Test Loss: 0.3894927501678467

Test Accuracy: 0.9734252095222473

Test Precision: 0.9065040946006775

Test Recall: 0.8779527544975281

Test AUC: 0.9858623147010803

Test F1 Score: 0.8921592831611633

Figure 4.17: evaluating the model

We achieved good results for all metrics (accuracy, loss, AUC, precision, and f1-score) after testing the model on the test set, as shown in the above figures. This indicates that the model can properly discriminate between classes. 4.18.

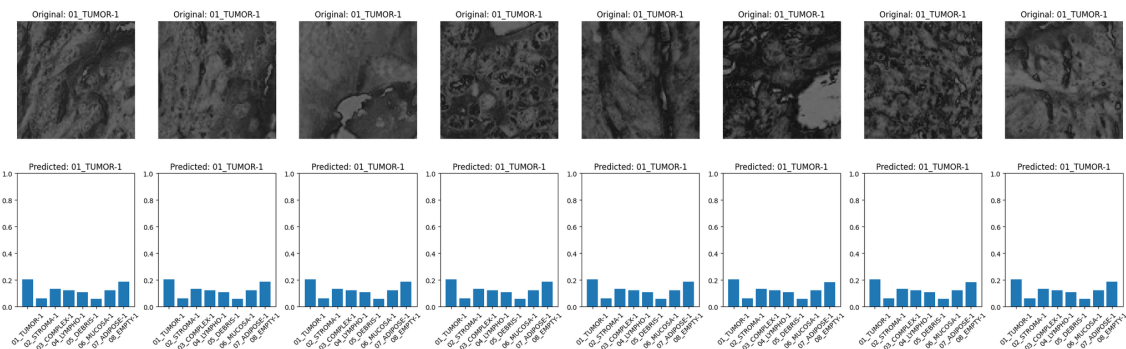


Figure 4.18: Prediction results

4.8 Conclusion

In this chapter, we discussed the application side and described the tools that were used to execute the suggested model. Finally, we came to the conclusion that it is possible to get successful outcomes by combining different strategies.

General Conclusion

The dataset and the model's architecture are the two key aspects that significantly affect the quality of the results. Various popular architectures (resnet, fcn, vgg19, etc.) were developed in a specific order to address problems that were not well-solved by earlier designs.

In order to create a model with a solid structure that can solve a classification problem, we created a model architecture that combines different strategies a type of neural network called Fully Convolutional Network (FCN), based on the Encoder-Decoder principle, followed by the design of a head map for image classification. The classification accuracy curve for colorectal cancer detection achieved impressive results, with accuracy scores of 0.9776 for the training set and 0.9743 for the validation set.

When applied to digital pathology images, the suggested method's performance can serve as as a highly valuable and efficient diagnostic tool for computer-aided diagnosis.

Bibliography

- [1] Matthew Fleming et al. “Colorectal carcinoma: Pathologic aspects”. In: *Journal of gastrointestinal oncology* 3.3 (2012), p. 153.
- [2] Mikala Egeblad, Elizabeth S Nakasone, and Zena Werb. “Tumors as organs: complex tissues that interface with the entire organism”. In: *Developmental cell* 18.6 (2010), pp. 884–901.
- [3] Mingchao Du et al. “Application of deep learning algorithm in feature mining and rapid identification of colorectal image”. In: *IEEE Access* 8 (2020), pp. 128830–128844.
- [4] In: (). URL: <https://www.pfizer.com/disease-and-conditions/colon-cancer>.
- [5] Fatima A Hagggar and Robin P Boushey. “Colorectal cancer epidemiology: incidence, mortality, survival, and risk factors”. In: *Clinics in colon and rectal surgery* 22.04 (2009), pp. 191–197.
- [6] Melissa M Center et al. “Worldwide variations in colorectal cancer”. In: *CA: a cancer journal for clinicians* 59.6 (2009), pp. 366–378.
- [7] RS Houlston. “What we could do now: molecular pathology of colorectal cancer”. In: *Molecular Pathology* 54.4 (2001), p. 206.
- [8] Mitchell S Cappell. “Pathophysiology, clinical presentation, and management of colon cancer”. In: *Gastroenterology Clinics of North America* 37.1 (2008), pp. 1–24.
- [9] Wei Chen and Wendy L Frankel. “A practical guide to biomarkers for the evaluation of colorectal cancer”. In: *Modern Pathology* 32 (2019), pp. 1–15.
- [10] John C Waterton. “Medical imaging: overview and the importance of contrast”. In: *Biomedical Imaging* (2011), pp. 1–20.
- [11] Otha W Linton. “Medical applications of X rays”. In: *Beam Line* 25.2 (1995), pp. 25–34.
- [12] URL: https://commons.wikimedia.org/wiki/File:Radiograf%C3%ADa_pulmones_Francisca_Lorca.jpg.

- [13] Paul G Newman and Grace S Rozycki. “The history of ultrasound”. In: *Surgical clinics of north America* 78.2 (1998), pp. 179–195.
- [14] URL: https://en.wikipedia.org/wiki/Medical_ultrasound#/media/File:Ultrasound_of_human_heart_apical_4-chamber_view.gif.
- [15] Tal Geva. “Magnetic resonance imaging: historical perspective”. In: *Journal of cardiovascular magnetic resonance* 8.4 (2006), pp. 573–580.
- [16] URL: https://en.wikipedia.org/wiki/Magnetic_resonance_imaging#/media/File:T1t2PD.jpg.
- [17] Anant Madabhushi and Metin N Gurcan. “History of the SPIE Medical Imaging Digital Pathology Conference”. In: *Journal of Medical Imaging* 9.S1 (2022), pp. 012203–012203.
- [18] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.
- [19] John C Gore. *Artificial intelligence in medical imaging*. 2020.
- [20] Maryellen L Giger and Kenji Suzuki. “Computer-aided diagnosis”. In: *Biomedical information technology*. Elsevier, 2008, pp. 359–XXII.
- [21] Rezaul Begg. “Artificial intelligence techniques in medicine and health care”. In: *Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2008, pp. 1750–1757.
- [22] Nathanaël Lemonnier et al. “Traditional knowledge-based medicine: a review of history, principles, and relevance in the present context of P4 systems medicine”. In: *Progress in preventive medicine* 2.7 (2017), e0011.
- [23] “medical knowledge base”. In: (). URL: <https://bmcmeginformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0804-1#citeas>.
- [24] Manuel Gil et al. “Towards a knowledge-based recommender system for linking electronic patient records with continuing medical education information at the point of care”. In: *IEEE Access* 7 (2019), pp. 15955–15966.
- [25] Issam El Naqa and Martin J Murphy. *What is machine learning?* Springer, 2015.
- [26] In: (). URL: <https://www.javatpoint.com/machine-learning-techniques>.
- [27] C Castillo-Botón et al. “Machine learning regression and classification methods for fog events prediction”. In: *Atmospheric Research* 272 (2022), p. 106157.
- [28] Giuseppe Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

- [29] Daniele Ucci, Leonardo Aniello, and Roberto Baldoni. “Survey of machine learning techniques for malware analysis”. In: *Computers & Security* 81 (2019), pp. 123–147.
- [30] “Clustering in Machine Learning”. In: (). URL: https://en.wikipedia.org/wiki/Cluster_analysis#/media/File:KMeans-Gaussian-data.svg.
- [31] In: (). URL: <https://www.bmc.com/blogs/deep-neural-network/>.
- [32] In: (). URL: <https://heartbeat.comet.ml/understand-the-fundamentals-of-the-k-nearest-neighbors-knn-algorithm-533dc0c2f45a>.
- [33] Chesner Desir. “Classification automatique d’images, application à l’imagerie du poumon profond”. PhD thesis. Université de Rouen, 2013.
- [34] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.
- [35] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [36] Weina Wang et al. “The global fuzzy c-means clustering algorithm”. In: *2006 6th World Congress on Intelligent Control and Automation*. Vol. 1. IEEE. 2006, pp. 3604–3607.
- [37] In: (). URL: <https://matlab1.com/support-vector-machine-2/>.
- [38] Rohith Gandhi. “Support vector machine—introduction to machine learning algorithms”. In: *Towards Data Science* 7.06 (2018).
- [39] In: (). URL: <https://matlab1.com/support-vector-machine-2/>.
- [40] Denise Aberle et al. “A primer on imaging anatomy and physiology”. In: *Medical imaging informatics* (2010), pp. 15–90.
- [41] Mohd Fauzi Bin Othman, Norarmalina Abdullah, and Nur Aizudin Bin Ahmad Rusli. “An overview of MRI brain classification using FPGA implementation”. In: *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*. IEEE. 2010, pp. 623–628.
- [42] In: (). URL: https://www.researchgate.net/figure/a-Biological-neuron-structure-and-neuron-model-b-Deep-neural-network-structure_fig1_334637084.
- [43] Jiuxiang Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern Recognition* 77 (2018), pp. 354–377. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2017.10.013>. URL: <https://www.sciencedirect.com/science/article/pii/S0031320317304120>.

- [44] Chesner Desir. “Classification Automatique d’Images, Application à l’Imagerie du Poumon Profond”. Theses. Université de Rouen, July 2013. URL: <https://theses.hal.science/tel-00879356>.
- [45] In: (). URL: <https://www.scirp.org/journal/paperinformation.aspx?paperid=96342>.
- [46] Klaus Hechenbichler and Klaus Schliep. “Weighted k-nearest-neighbor techniques and ordinal classification”. In: (2004).
- [47] In: (). URL: <https://www.kdg.be/en/research-and-expertise/sustainable-chemistry/sussol-sustainable-solvents>.
- [48] In: (). URL: <https://medium.com/hackernoon/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860>.
- [49] Jakob Nikolas Kather et al. “Multi-class texture analysis in colorectal cancer histology”. In: *Scientific reports* 6.1 (2016), p. 27988.
- [50] Srinath Jayachandran and Ashlin Ghosh. “Deep transfer learning for texture classification in colorectal cancer histology”. In: *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. Springer. 2020, pp. 173–186.
- [51] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. “What is being transferred in transfer learning?”. In: *Advances in neural information processing systems* 33 (2020), pp. 512–523.
- [52] Nour Eldeen M Khalifa et al. “Deep transfer learning models for medical diabetic retinopathy detection”. In: *Acta Informatica Medica* 27.5 (2019), p. 327.
- [53] In: (). URL: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>.
- [54] In: (). URL: <https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33>.
- [55] 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.
- [56] In: (). URL: https://d2l.ai/chapter_computer-vision/fcn.html.
- [57] In: (). URL: <https://paperswithcode.com/method/fcn1>.
- [58] In: (). URL: <https://ai.stackexchange.com/questions/21810/what-is-a-fully-convolution-network>.
- [59] Sankaranarayanan Piramanayagam et al. “Supervised classification of multi-sensor remotely sensed images using a deep learning framework”. In: *Remote sensing* 10.9 (2018), p. 1429.

- [60] URL: <https://www.who.int/news-room/fact-sheets/detail/colorectal-cancer>.
- [61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].
- [62] Mangalam Sankupellay and Dmitry Konovalov. “Bird call recognition using deep convolutional neural network, ResNet-50”. In: *Proc. Acoustics*. Vol. 7. 2018. 2018, pp. 1–8.
- [63] Mostafa Ibrahim and Kevin Bryson. “Generalized Categorisation of Digital Pathology Whole Image Slides using Unsupervised Learning”. In: *arXiv preprint arXiv:2012.13955* (2020).
- [64] Sana Sahar Guia et al. “A Salient Object Detection Technique Based on Color Divergence”. In: *Proceedings of the 4th International Conference on Future Networks and Distributed Systems*. 2020, pp. 1–6.
- [65] In: (). URL: <https://datagen.tech/guides/computer-vision/resnet-50/>.
- [66] In: (). URL: https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/ResNet50.
- [67] In: (). URL: <https://iq.opengenus.org/resnet50-architecture/>.
- [68] Sanghyun Woo et al. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.
- [69] Tawsifur Rahman et al. “Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization”. In: *IEEE Access*. 8 (), pp. 191586–191601.
- [70] Jakob Nikolas Kather et al. *Collection of textures in colorectal cancer histology*. Zenodo, May 2016. DOI: 10.5281/zenodo.53169. URL: <https://doi.org/10.5281/zenodo.53169>.
- [71] Dr. Guangliang Chen. “Evaluation Criteria (for binary classification)”. In: (), pp. 06–19. URL: <https://www.sjsu.edu/faculty/guangliang.chen/Math251/lec6eval-criteria.pdf>.
- [72] “What is Python? Executive Summary”. In: (). URL: <https://www.python.org/doc/essays/blurb/>.
- [73] “Python Logo”. In: (). URL: <https://www.python.org/>.
- [74] “About Tensorflow”. In: (). URL: <https://www.tensorflow.org/>.
- [75] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

- [76] “About scikit-learn”. In: (). URL: <https://scikit-learn.org/>.
- [77] “About Pandas”. In: (). URL: <https://pandas.pydata.org/>.
- [78] “About OpenCV”. In: (). URL: <https://opencv.org/about/>.
- [79] “About Matplotlib”. In: (). URL: <https://matplotlib.org/>.
- [80] “What is NumPy?” In: (). URL: <https://numpy.org/doc/stable/user/whatisnumpy.html>.
- [81] “Frequently Asked Questions”. In: (). URL: <https://research.google.com/colaboratory/faq.html>.
- [82] Shanqing Gu, Manisha Pednekar, and Robert Slater. “Improve image classification using data augmentation and neural networks”. In: *SMU Data Science Review* 2.2 (2019), p. 1.