

**Algerian Democratic and Popular Republic**

**Ministry of Higher Education and Scientific Research**

**Amar Telidjy University Laghouat**

**Faculty of Technology**



**Master's thesis**

**Sector: Electronic**

**Option: Embedded system**

**Presented by: Khamed Aissa**

**Theme**

**Skin Cancer Detection with Deep Learning**

**Board of Examiners**

<b>Nom et Prénom</b>	<b>Grade</b>	<b>Qualité</b>
REGGAB Mourad	MA.A	Chairperson
Zaitoni Abdelkader	MC.B	Examiner
BIRANE Abdelkader	MC.B	Supervisor

**2024/2025**

## Abstract

Skin cancer is a serious public health concern, and early detection is essential for effective treatment. This research explores the use of Deep Learning, particularly Convolutional Neural Networks (CNNs), in the automatic classification of skin lesions from medical images. The study includes an introduction to deep learning in medicine, a detailed explanation of CNN architecture, and a practical implementation using the Melanoma Cancer Image Dataset from Kaggle. Results show that CNNs can achieve high accuracy in detecting melanoma, making them promising tools for clinical support. Challenges such as data imbalance and variability in skin tone are also discussed.

## Résumé

Le cancer de la peau est un problème majeur de santé publique, et sa détection précoce est cruciale pour un traitement efficace. Ce travail de recherche examine l'utilisation de l'apprentissage profond (Deep Learning), notamment des réseaux de neurones convolutifs (CNN), pour la classification automatique des lésions cutanées à partir d'images médicales. L'étude comprend une introduction à l'apprentissage profond dans le domaine médical, une description détaillée de l'architecture des CNN, ainsi qu'une implémentation pratique utilisant le jeu de données Melanoma Cancer de Kaggle. Les résultats montrent que les CNN offrent une précision élevée dans la détection du mélanome, ce qui en fait des outils prometteurs pour l'aide au diagnostic. Les défis tels que le déséquilibre des données et la diversité des teints de peau sont également abordés.

## ملخص

يُعدّ سرطان الجلد من أخطر المشكلات الصحية، ويُعتبر الكشف المبكر عنه ضروريًا للعلاج الفعّال. تتناول هذه الدراسة استخدام تقنيات التعلم العميق، وخاصة الشبكات العصبية الالتفافية (CNN)، في تصنيف آلي لصور الآفات الجلدية المأخوذة من صور طبية. تشمل المقدمة حول دور التعلم العميق في المجال الطبي، وشرحًا مفصلاً لبنية CNN، وتطبيقًا عمليًا باستخدام قاعدة البيانات Melanoma Cancer Image Dataset من منصة Kaggle. أظهرت النتائج قدرة هذه الشبكات على تحقيق دقة عالية في اكتشاف سرطان الجلد، مما يجعلها أداة واعدة لدعم الأطباء. كما تم التطرق إلى تحديات مثل عدم توازن البيانات وتنوع لون البشرة.

# Table of Contents

Abstract.....	I
Table of Contents.....	II
Terminology.....	IV
Table of images.....	V
<b>General introduction .....</b>	<b>1</b>
<b>Chapter 1: Basics Of Deep Learning .....</b>	<b>3</b>
1.1 Introduction .....	3
1.2 artificial intelligence.....	4
1.2.1 Definition .....	4
1.2.2 Historical Milestones.....	4
1.2.3- Difference between AI, machine learning, and deep learning.....	5
1.3 Deep learning concept.....	6
1.3.1- definition.....	6
1.4 Types of neural networks.....	7
1.4.1 Convolutional neural networks (CNN).....	7
1.4.2 Recurrent neural network.....	8
1.4.3 Generative Adversarial Networks (GANs).....	10
1.5- Applications of deep learning in the medical field.....	10
1.5.1 Deep Learning for Medical Image Analysis and CAD.....	10
1.5.2- Disease diagnosis with deep learning.....	11
1.5.3-Challenges in Deep-Learning.....	12
1.6 cancer detection.....	13
1.6.1 Deep learning techniques in skin cancer.....	14
1.7 Communication Barrier between AI and Dermatologists.....	15
1.8 Conclusion.....	15
<b>Chapter 2: Convolution Neural NetworksCNN.....</b>	<b>16</b>
2.1-Introduction.....	17
2.2 Convolution neural network (CNN).....	17
2.2.1 CNN in image analyses .....	18
2.2.2-Difference between CNN and other algorithm.....	18
2.3-The Architecture of CNN.....	19
2.4 Mathematical Formulation.....	24
2.5-CNN for Image Classification.....	27

2.6 CNN in Medical Imaging.....	29
2.6.1 Medical image understanding.....	29
2.6.2 Imaging Modalities for Analytics and Diagnostics.....	29
2.6.3 CNN on medical image classification.....	30
2.6.4 Medical image classification.....	31
2.6.5 Medical image segmentation.....	31
2.7 Conclusion.....	32
<b>Chapter 3: CNN for Skin Cancer Detection.....</b>	<b>33</b>
3.1-Introduction .....	34
3.2 System design.....	34
3.3 Early Detection of Skin Cancer .....	34
3.4 Type of skin cancer .....	35
3.5 Development environment.....	36
3.6 Libraries used .....	37
3.7 Python in Skin Cancer Detection using Deep Learning.....	38
3.8 Melanoma Cancer Image Dataset of 10000 images on Kaggle.....	39
3.8.1 Database Content.....	39
3.8.2 The scientific value of the dataset.....	39
3.8.3 Database Challenges.....	39
3.8.4 Research Uses.....	40
3.9 Experimentation and results.....	40
3.9.1 Import the Libraries.....	40
3.9.2 Configuration Data set .....	41
3.9.3 Data preprocessing.....	41
3.9.4 Building a CNN neural network model.....	42
3.9.5 Assemble the model.....	43
3.9.6 Training model.....	43
3.9.7 Evaluation using classification.....	44
3.9.8 Performance curves.....	44
3.9.9 Performance Curve Analysis.....	45
3.10 Challenges and limitations in using convolutional neural networks (CNN) for skin cancer detection.....	46
3.11 conclusion .....	46
<b>General conclusion.....</b>	<b>48</b>
<b>The References .....</b>	<b>49</b>

## **Terminology**

AI : Artificial Intelligence

MI : Machine Learning

DL : Deep Learning

GPU : Graphics Processing Unit

CAD : Computer-Aided Design

CT : Computed Tomography

## Table of images

Fig.1.1 Artificial intelligence subfields.....	5
Fig.1.2 Human brain and the neural networks.....	6
Fig.1.3 Neural networks.....	7
Fig.1.4 RNN layers.....	9
Fig.1.5 Structure of a typical CNN.....	12
Fig.1.6 Overview of the nodule classification method.....	12
Fig.1.7 Type of skin tumor.....	14
Fig.1.8 CNN for skin cancer.....	15
Fig.2.1 Architecture of convolutional neural networks.....	20
Fig.2.2 Filtering operation.....	21
Fig.2.3 Pooling operation.....	22
Fig.2.4 Steps of CNN.....	23
Fig. 2.5 Example of 2-D convolution.....	25
Fig.2.6 Plan of CNN.....	26
Fig.2.7 Types of filters.....	28
Fig.2.8 After filtering.....	28
Fig.2.9 Imaging Modalities for Analytics and Diagnostics.....	30
Fig.3.1 System design.....	34
Fig 3.2 Import the Libraries .....	40
Fig 3.3 Configuration Data set .....	41
Fig 3.4 Data preprocessing part1.....	41
Fig 3.5 Data preprocessing part 2.....	42
Fig 3.6 Building a CNN neural network model .....	42
Fig 3.7 Assemble the model.....	43
Fig 3.8 Training model .....	43
Fig 3.9 Evaluation using classification .....	44
Fig 3.10 Performance curves .....	44
Fig 3.11 Performance Curve Analysis.....	45

---

# **Introduction**

---

## **General Introduction**

Skin cancer is one of the most common types of cancer worldwide, with its incidence steadily increasing over the past decades. Among its various forms, melanoma is considered the most aggressive and life-threatening if not detected and treated early. The key to improving patient outcomes lies in early diagnosis, which significantly increases the chances of successful treatment and survival.

Traditionally, skin cancer diagnosis has relied on clinical examination by dermatologists, followed by dermoscopic analysis and histopathological confirmation. While these methods are effective, they are time-consuming, dependent on the experience of medical professionals, and may not always be accessible in low-resource settings.

In recent years, the rapid advancement of artificial intelligence (AI), particularly deep learning, has opened new avenues in the field of medical image analysis. Convolutional Neural Networks (CNNs), a class of deep learning models, have shown exceptional performance in various image classification tasks, including skin lesion analysis. These models can learn complex visual patterns from large datasets and perform automatic feature extraction, making them suitable for detecting subtle differences between benign and malignant lesions.

This thesis explores the use of deep learning and CNN architectures for the detection of skin cancer through dermoscopic images. The work focuses on developing and evaluating models capable of classifying skin lesions into benign or malignant categories with high accuracy. By leveraging publicly available datasets and state-of-the-art neural network models, this study aims to contribute to the ongoing effort to develop accessible, accurate, and efficient tools for skin cancer diagnosis.

This work aimed to explore the application of deep learning—specifically Convolutional Neural Networks (CNNs)—in the field of skin cancer detection through medical image analysis. The research was structured into three chapters, each building upon the previous to form a complete pipeline from theoretical foundations to practical implementation.

Our document is organized as follows:

The chapter one introduced the concept of deep learning, its relationship to artificial intelligence and machine learning, and its transformative role in medicine. We examined how deep learning enhances diagnostic accuracy, particularly in medical imaging, and addressed its key benefits and challenges in clinical applications.

In the second chapter, we focused on CNNs as a core deep learning architecture for image analysis. We discussed the structure of CNNs—including convolutional, pooling, and fully connected layers—and their advantages in handling visual data such as dermoscopic images. The chapter also covered why CNNs are especially well-suited for detecting skin cancer from images.

The final chapter presented the practical implementation of a CNN-based model using a real dataset from Kaggle. It detailed the preprocessing steps, model design, training, and evaluation metrics. The results demonstrated that CNNs can achieve promising accuracy in classifying skin lesions, highlighting their potential as a supportive diagnostic tool for dermatologists.

---

# **Chapter I**

## **Basics of Deep Learning**

---

## **1.1 Introduction**

Artificial Intelligence (AI) has become a vital component in the evolution of modern medicine, offering new possibilities in diagnosis, treatment planning, and patient monitoring. One of the most advanced techniques within AI is Deep Learning (DL), a subset of machine learning that utilizes multilayered neural networks to automatically learn complex patterns in data. In the medical field, DL has shown exceptional performance in tasks that involve the analysis of medical images, such as radiographs, MRIs, CT scans, and dermoscopic images. By enabling machines to interpret visual data with high accuracy, deep learning is not only augmenting the capabilities of medical professionals but also helping to reduce diagnostic errors and improve healthcare outcomes. This chapter introduces the foundational concepts of deep learning and discusses its transformative role in medical imaging and disease [1].

## **1.2 Artificial intelligence**

### **1.2.1 Definition**

Artificial intelligence is a field of science that aims to create computers and machines capable of reasoning, learning, and acting in ways that would normally require human intelligence or involve data on a scale beyond what humans can analyze. AI is a broad field that covers a wide range of topics, including computer science, data analysis and statistics, hardware and software engineering, linguistics, neuroscience, and even philosophy and psychology. At the operational level for businesses, AI is a set of technologies based primarily on machine learning and deep learning, used for data analysis, predictions and forecasts, object categorization, natural language processing, recommendations, intelligent data retrieval, etc. [2]

### **1.2.2 Historical Milestones**

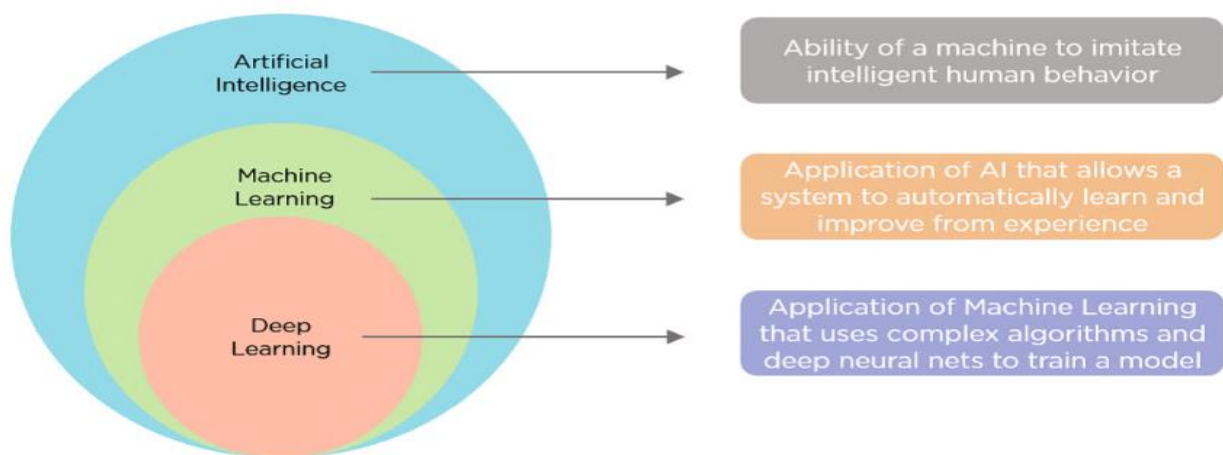
- *1950*: Alan Turing proposed the Turing Test to assess machine intelligence.
- *1956*: John McCarthy coined "Artificial Intelligence" and developed the Lisp language.
- *1967*: Frank Rosenblatt developed the Mark 1 Perceptron, an early neural network.
- *1980s*: Backpropagation enabled more practical training of neural networks.
- *1997*: IBM's Deep Blue defeated chess champion Garry Kasparov.
- *2011-2016*: AI made significant strides in games, language, and image classification (e.g., IBM Watson, AlphaGo).

- 2022-2024: Advancements in large language models (LLMs) and multimodal AI reshaped enterprise and medical AI applications [3].

### 1.2.3- Difference between AI, machine learning, and deep learning

As a whole, artificial intelligence contains many subfields, including:

- **Machine learning** automates analytical model building. It uses methods from neural networks, statistics, operations research and physics to find hidden insights in data without being explicitly programmed where to look or what to conclude.
- **A neural network** is a kind of machine learning inspired by the workings of the human brain. It's a computing system made up of interconnected units (like neurons) that processes information by responding to external inputs, relaying information between each unit. The process requires multiple passes at the data to find connections and derive meaning from undefined data.
- **Deep learning** uses huge neural networks with many layers of processing units, taking advantage of advances in computing power and improved training techniques to learn complex patterns in large amounts of data. Common applications include image and speech recognition.[4]

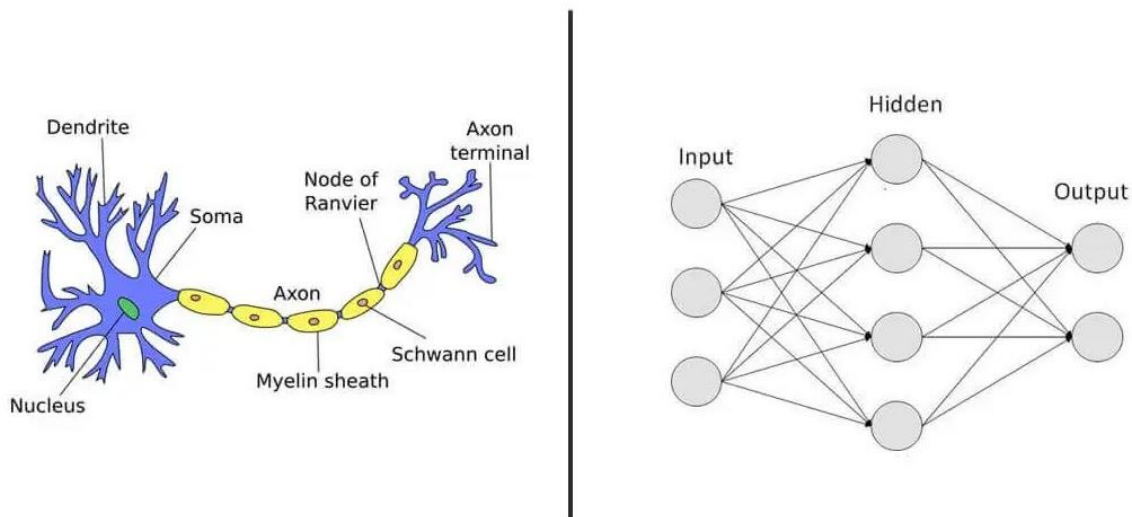


**Fig.1.1 Artificial intelligence subfields**

## 1.3 Deep learning concept

### 1.3.1 Definition

Deep learning is a subset of Machine Learning; Deep Learning imitates the functioning of the human brain using artificial neural networks. Deep learning is referred to as deep learning because it comes from using multiple layers in the network [5]



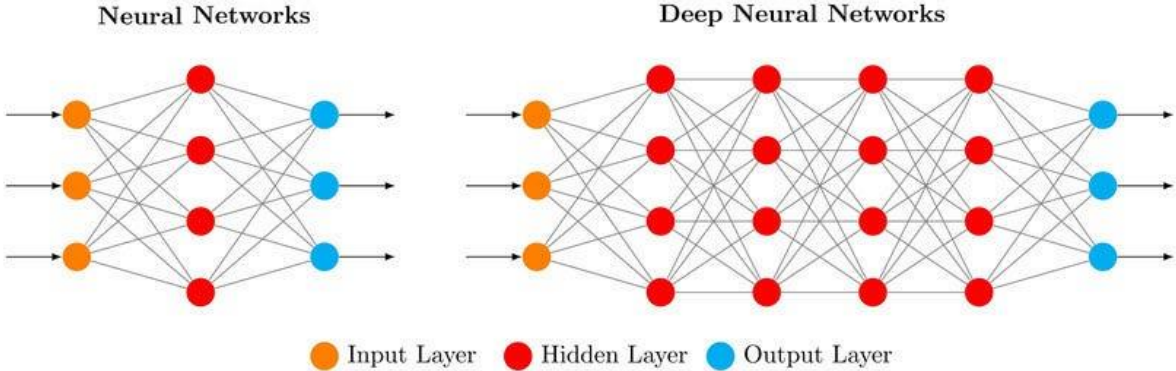
**Fig.1.2 Human brain and the neural networks**

Neural networks, or artificial neural networks, attempt to mimic the human brain through a combination of data inputs, weights, and biases, all acting like silicon neurons. These elements work together to accurately recognize, classify and describe objects in the data.

Deep neural networks are made up of multiple layers of interconnected nodes, each building on the previous one to refine and optimize prediction or categorization. This progression of calculations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data to be processed, and the output layer is where the final prediction or classification is performed.

Another process called backpropagation uses algorithms, such as the gradient descent algorithm, to calculate errors in the predictions, then adjusts the weights and biases of the function moving up through the layers to train the model. Together, forward propagation and backpropagation allow the neural network to make predictions and correct any errors. Over time, the algorithm becomes more precise.

Deep learning requires considerable computing power. High-performance graphics processing units (GPUs) are ideal because they can handle a large volume of calculations in multiple cores with generous available memory. Distributed cloud computing can also help. This level of computing power is necessary to train deep algorithms via deep learning. However, managing multiple GPUs on-premises can create a high demand on internal resources and be extremely expensive to scale. Regarding software requirements, most deep learning applications are coded with one of three learning frameworks: JAX, PyTorch, or TensorFlow.[6]



**Fig.1.3 Neural networks**

**1.4 Types of neural networks**

There are a lot of types of neural networks that have been developed by now. The Convolutional Neural network (CNN) and Recurrent neural network (RNN) can be considered as two of the most prominent types of neural networks among them, which form the basis for most pre-trained models in neural networks [9]

**1.4.1 Convolutional neural networks (CNN)**

Convolutional neural networks (CNNs) are a specific category of deep neural networks that employ the convolution operation in order to process the input data. Even though the main concept dates back to the 1990s and is greatly inspired by neuroscience (in particular by the organization of the visual cortex), its widespread use is due to a relatively recent success on the ImageNet Large Scale Visual Recognition Challenge of 2012. In contrast to the deep fully connected networks that have already been discussed, CNNs excel in processing data with a spatial or grid-like organization (e.g., time series, images, videos, etc.) while at the same time decreasing the number of trainable parameters due to their weight sharing properties [8]

A normal neural network consists of multiple layers called the input layer, output layer, and hidden layers. In each layer every node (neuron) is connected to all nodes (neurons) in the next layer with parameters called ‘weights’.

Neural networks consist of nodes called perceptrons that do necessary calculations and detect features of neural networks. These perceptrons try to reduce the final cost error by adjusting the weights parameters. Moreover, a perceptron can be considered as a neural network with a single layer.

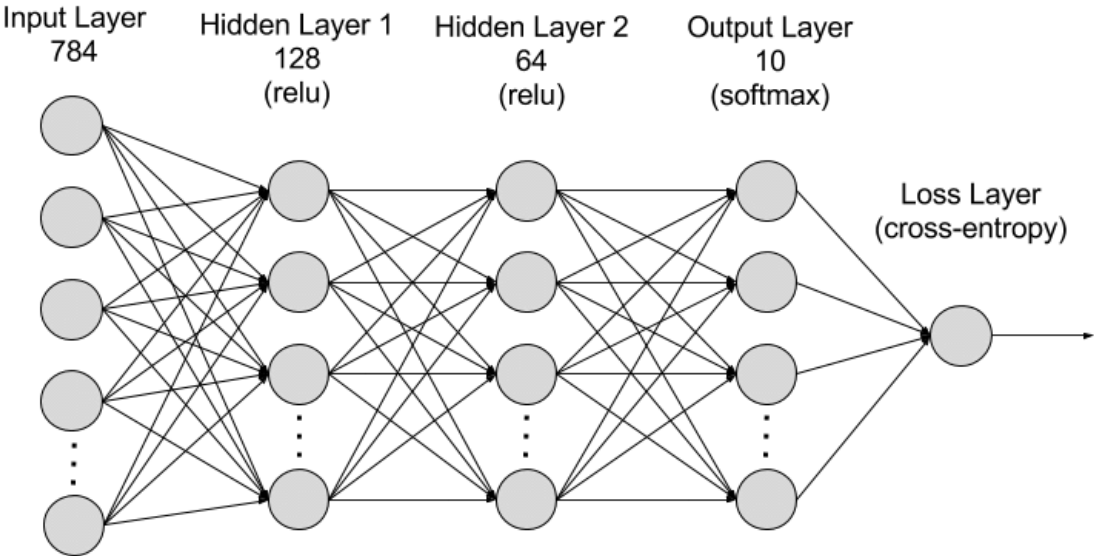
On the other hand, multilayer perceptrons are called deep neural networks. The perceptrons are activated when there is satisfiable input.

Now let’s move on to discuss the exact steps of a working neural network.

1. Initially, the dataset should be fed into the input layer which will then flow to the hidden layer.
2. The connections which exist between the two layers randomly assign weights to the input.
3. A bias is added to each input. Bias is a constant which is used in the model to fit best for the given data.
4. The weighted sum of all the inputs will be sent to a function that is used to decide the active status of a neuron by calculating the weighted sum and adding the bias. This function is called the activation function.
5. The nodes that are required to fire for feature extraction are decided based on the output value of the activation function.
6. The final output of the network is then compared to the required labeled data of our dataset to calculate the final cost error. The cost error is actually telling us how ‘bad’ our network is. Hence we want the error to be as smallest as we can.
7. The weights are adjusted through backpropagation, which reduces the error. This backpropagation process can be considered as the central mechanism that neural networks learn. It basically fine-tunes the weights of the deep neural network in order to reduce the cost value.

#### **1.4.2 Recurrent neural network**

A recurrent neural network (RNN) is a deep learning model that is trained to process and convert sequential data input into a specific sequential data output. Sequential data is data, such as words, sentences, or time series data, in which sequential components are interconnected based on complex semantics and syntax rules. An RNN is a software system made up of many interconnected components that mimic the way humans perform sequential data conversions, such as translating text from one language to another. RNNs are largely being replaced by transformer-based artificial intelligence (AI) and large language models (LLM), which are much more efficient in sequential data processing. How does a recurrent neural network work?



**Fig.1.4 RNN layers**

RNNs are made up of neurons: data processing nodes that work together to perform complex tasks. Neurons are organized into input, output and hidden layers. The input layer receives the information to be processed and the output layer provides the result. Data processing, analysis and prediction takes place in the hidden layer.

RNNs work by transmitting the sequential data they receive to the hidden layers step by step. However, they also have an automatic looping or recurring workflow: the hidden layer can remember and use previous inputs for future predictions in a short-term memory component. It uses the current input and stored memory to predict the next sequence.

For example, consider the sequence: The apple is red. You want the RNN to predict red when it receives the input sequence apple is. When the hidden layer processes the word apple, it stores a copy of it in its memory. Then, when she sees the word is, he recalls apple from his memory and understands the full sequence: apple is for context. It can then predict red for better

accuracy. This makes RNNs useful for speech recognition, machine translation, and other language modeling tasks. [10]

### **1.4.3 Generative Adversarial Networks (GANs)**

A generative adversarial network (GAN) is a deep learning architecture. It trains two neural networks to compete with each other to generate new, more authentic data from a given training dataset. For example, you can generate new images from an existing image database or original music from a song database. A GAN is called adversarial because it trains two different networks and pits them against each other. A network generates new data by taking a sample of input data and modifying it as much as possible. The other network tries to predict whether the generated data belongs to the original dataset. In other words, the prediction network determines whether the generated data is fake or real. The system generates newer, improved versions of fake data values until the prediction network can no longer distinguish the fakes from the original. [36]

## **1.5 Applications of deep learning in the medical field**

The emergence of artificial intelligence (AI) in medical imaging has heralded perhaps the greatest disruptive technology since the early days of Roentgen, Becquerel, and Curie, but certainly since the days of Hounsfield and Anger. In medical imaging, the artificial neural network (ANN) is the backbone of machine learning (ML) and deep learning (DL). An ANN is an analysis algorithm composed of layers of connected nodes. The inputs may be radiomic features that have been extracted from the image files or if using a convolutional neural network (CNN) may be the images themselves. AI in medical imaging ushers in an exciting era with reengineered and re-imagined clinical and research capabilities. An important driver of the emergence of AI in medical imaging has been the enhancement of visual recognition using AI in radiology to produce lower error rates than the human observer since 2015. Specific capabilities of ML in medical imaging include, without being limited to, detection and classification of lesions, automated image segmentation, data analysis, extraction of radiomic features, prioritizing reporting and study triage, and image reconstruction. An ANN can also be used in parallel to conventional statistical analysis to provide deeper insights into research data [6] [13]

### **1.5.1 Deep Learning for Medical Image Analysis and CAD**

CAD systems are developed with machine learning methods. A conventional machine learning approach to CAD in medical imaging used image analysis methods to recognize disease

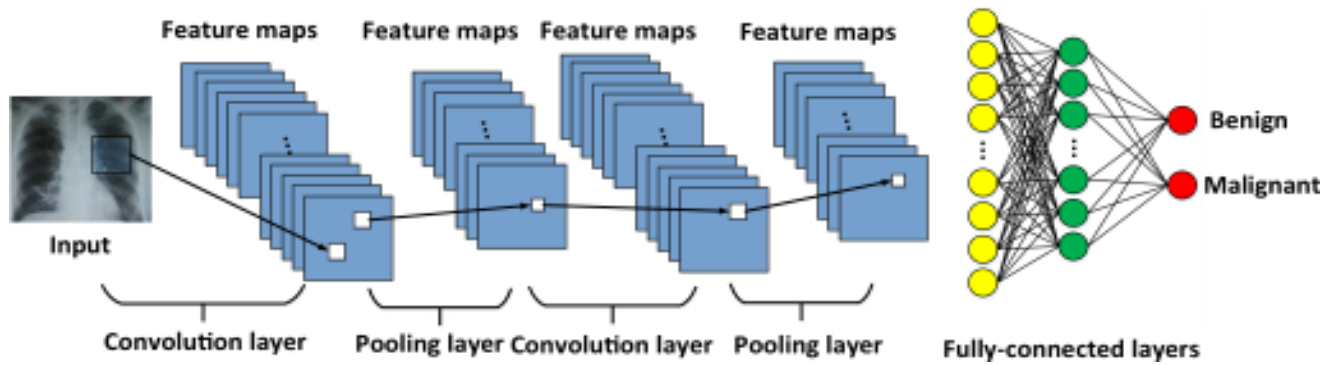
patterns and distinguish different classes of structures on images, e.g., normal or abnormal, malignant or benign. CAD developers design image processing and feature extraction techniques based on domain knowledge to represent the image characteristics that can distinguish the various states. The effectiveness of the feature descriptors often depends on the domain expertise of the CAD developers and the capability of the mathematical formulations or empirical image analysis techniques that are designed to translate the image characteristics to numerical values. The extracted features are then used as input predictor variables to a classifier, and a predictive model is formed by adjusting the weights of the various features based on the statistical properties of a set of training samples to estimate the probability that an image belongs to one of the states. Conventional machine learning approach has limitations in that the human developer may not be able to translate the complex disease patterns into a finite number of feature descriptors even if they have seen a large number of cases from the patient population. The hand-engineered features may also have difficulty to be robust against the large variations of normal and abnormal patterns. [14]

### **1.5.2- Disease diagnosis with deep learning**

The most commonly quoted applications of DL in medical imaging applications have been for object detection (e.g. location of a lesion), object segmentation (eg, lesion contouring), and object classification (e.g. malignant vs. benign lesion) . These operations are often performed in cascade and lend themselves to different medical applications. Object detection, for example, has been used extensively in computer aided diagnosis of mammograms to point the reader to potential tumors and likewise in lung and liver CT scans. Object segmentation has been utilized in automated radiation treatment planning for delineating tumors and organs as target and dose-sparing regions, respectively. Finally, the entire field of radiomics is an exercise in lesion classification (relying on prior lesion detection/segmentation).

These applications, to date, have been application specific, without general intelligence. Hence, AI for segmenting the spleen is poorly able to segment a liver without extensive training. [15]

#### **A. Disease diagnosis in chest X-ray image**

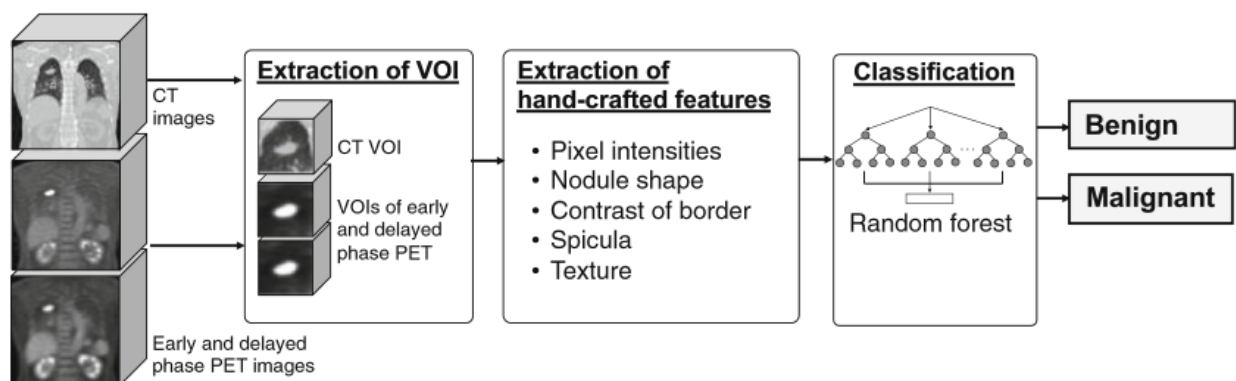


**Fig.1.5 Structure of a typical CNN**

The CNN employs alternating convolutional and pooling layers, and contains trainable filter banks per layer. Each individual filter in a filter bank is able to generate a feature map. This process is alternated and the CNN can learn increasingly more and more abstract features that will later be used by the fully connected layers to accomplish the classification task. [16]

### **B. Automated Malignancy Analysis of Lung Nodules in PET/CT Images Using Radiomic Features**

PET/CT examination is performed for detailed analysis of lung disease. In this combined technique, PET images provide functional information, while CT images render anatomical information, thereby facilitating a comprehensive analysis of the malignancy of nodules. [19]



**Fig.1.6 Overview of the nodule classification method**

### **1.5.3 Challenges in Deep Learning**

CAD or AI is expected to be useful decision support tools in medicine in the near future. Other than detection and characterization of abnormalities, applications such as pre-screening and triaging, cancer staging, treatment response assessment, recurrence monitoring, and prognosis

or survival prediction are being explored. Although no CAD systems with new AI techniques have been subjected to large scale clinical trials to date. [17]

Identifying medical domain knowledge is not an easy task. Firstly, the experiences of medical doctors are generally subjective and fuzzy. Not all medical doctors can give accurate and objective descriptions on what kinds of experiences they have leveraged to finish a given task. In addition, experiences of medical doctors can vary significantly or even contradictory to each other. Furthermore, medical doctors generally utilize many types of domain knowledge simultaneously. Finally, currently the medical domain knowledge is identified manually, and there is no existing work on the automatically and comprehensively identifying medical domain knowledge for a given area.

One solution to the automatic identifying medical knowledge is through text mining techniques on the guidelines, books, and medical reports related to different medical areas. Guidelines or books are more robust than individual experiences. Medical reports generally contain specific terms (usually adjectives) that describe the characteristics of tumors. These terms, containing important information to help doctors to make diagnosis, can potentially be beneficial for deep learning models. [18]

Besides the identification of medical domain knowledge, how to select appropriate knowledge to help image analysis tasks is also challenging. It should be noted that a common practice of medical doctors may not be able to help deep learning models because the domain knowledge might be learned by the deep learning model from training data. We believe that the knowledge that is not easily learned by a deep learning model can greatly help the model to improve its performance.

As a data-driven approach, the main challenge in the development of systems based on deep convolutional network architectures was and remains the availability and quality of data, particularly for medical imaging. In general image classification, sources of data can include social media and public online repositories. Further, labels are often well-defined and specialized training is unnecessary to identify objects in daily living. [20]

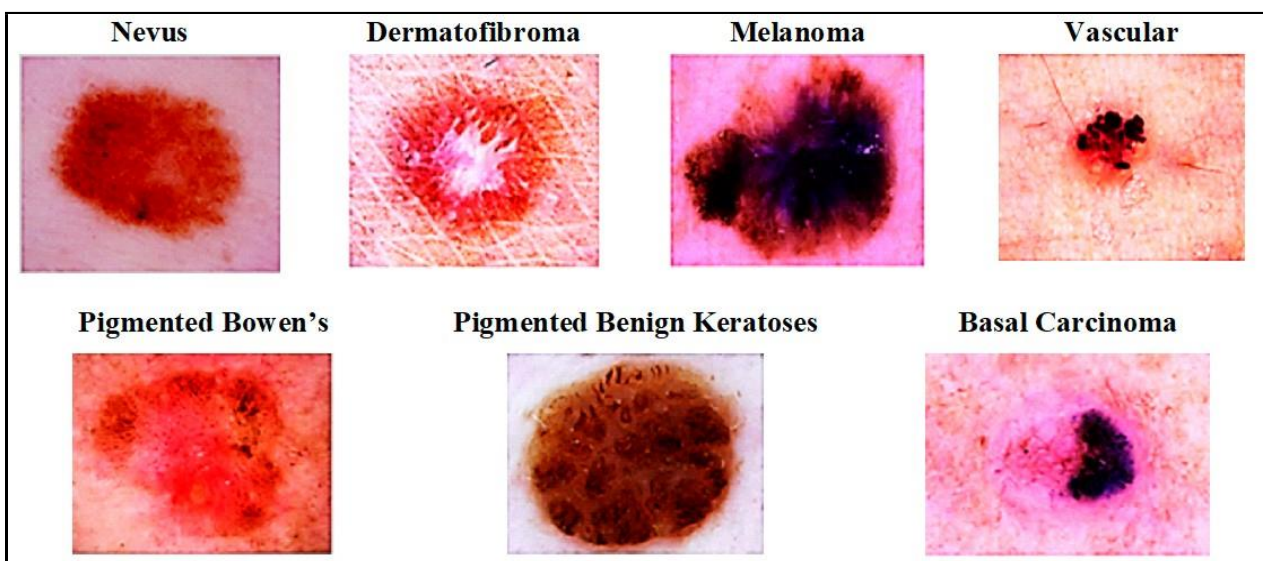
## **1.6 Skin cancer detection**

Skin cancer is the abnormal growth of skin cells. The cancerous growth may affect both the layers dermis and epidermis, but this review is concerned primarily with epidermal skin cancer; the two types of skin cancers that can arise from the epidermis are carcinomas and melanomas, depending on their cell type keratinocytes or melanocytes, respectively. It is a challenge to

estimate the incidence of skin cancer due to various reasons, such as the multiple sub-types of skin cancer. This poses as a problem between different lesion types is usually accompanied by the presence or the absence of certain dermoscopic features. [22]

### 1.6.1 Deep learning techniques in skin cancer

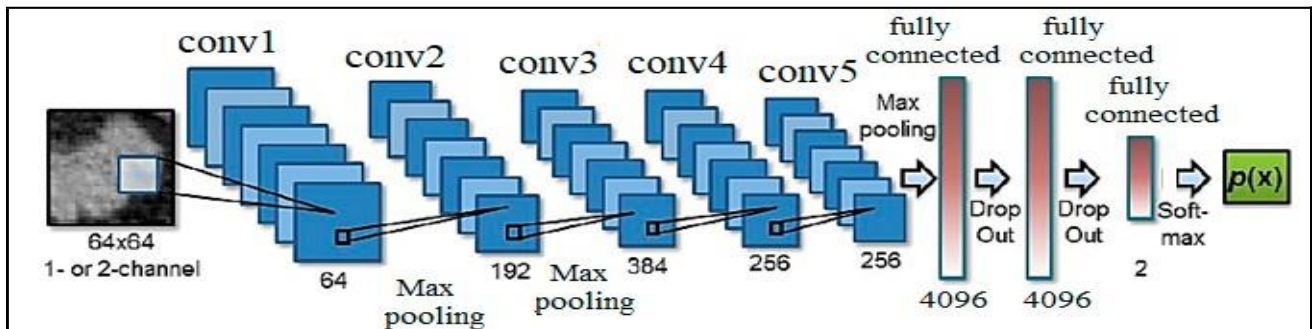
Deep neural networks play a significant role in skin cancer detection. They consist of a set of interconnected nodes. Their structure is similar to the human brain in terms of neuronal interconnectedness. Their nodes work cooperatively to solve particular problems. Neural networks are trained for certain tasks; subsequently, the networks work as experts in the domains in which they were trained. In our study, neural networks were trained to classify images and to distinguish between various types of skin cancer. Different types of skin lesion from International Skin Imaging Collaboration (ISIC) dataset are presented in Figure 2. We searched for different techniques of learning, such as ANN, CNN, KNN, and GAN for skin cancer detection systems. Research related to each of these deep neural networks is discussed in detail in this section. [23]



**Fig.1.7 Type of skin tumor**

A convolution neural network is an essential type of deep neural network, which is effectively being used in computer vision. It is used for classifying images, assembling a group of input images, and performing image recognition. CNN is a fantastic tool for collecting and learning global data as well as local data by gathering more straightforward features such as curves and edges to produce complex features such as shapes and corners. CNN's hidden layers consist of convolution layers, nonlinear pooling layers, and fully connected layers. CNN can contain multiple convolution layers that are followed by several fully connected layers. Three major

types of layers involved in making CNN are convolution layers, pooling layers, and full-connected layers . The basic architecture of a CNN is presented in Figure 8.



**Fig.1.8 CNN for skin cancer**

### 1.7 Communication Barrier between AI and Dermatologists

Giant strides in the field of Artificial Intelligence for skin cancer diagnosis mean it may have established its place in the scene for years to come, but it has still not been able to breach the communication barrier that exists between dermatologists and AI. It is suggested that dermatologists must embrace the potential shown by AI applications in various fields, such as clinical and research situations. The preconceived notion surrounding the use of AI in the cancer diagnosis domain that the introduction of technology may eventually downsize the workforce has set the wrong precedent and has brought about apprehensions about adopting AI for the same. It must be understood that, while AI has been ever improving and returns a higher accuracy with respect to diagnoses, clinicians are undoubtedly more skillful in identifying mimics, as well as patterns that have not been made available to the models through the training datasets. It's reiterated that the role of dermatologists is not limited to detecting and identifying skin lesions, but also to extract valuable information and inferences from their observations. At present, the latter is not quite fine-tuned and is still in the rudimentary stages in machines. [26]

### 1.8 Conclusion

Deep learning has reshaped AI applications in medicine, particularly in image-based diagnostics. With architectures like CNNs, RNNs, and GANs, AI systems now aid in disease classification, treatment planning, and prognostics. Despite advancements, challenges remain, chief among them being data limitations, domain integration, and clinician acceptance. Continued research and ethical consideration will be key to fully integrating AI into healthcare.

---

Chapter II  
Convolution Neural Networks  
CNN

---

## **2.1-Introduction**

Deep convolutional neural networks (CNNs) have significantly transformed numerous fields, including the field of medical image analysis, becoming the state-of-the-art algorithms for tasks such as disease detection, organ segmentation, and image enhancement. Their ability to automatically learn hierarchical features from medical imaging data has enabled breakthroughs in diagnostic accuracy and patient care. CNNs typically consist of convolutional layers that detect spatial hierarchies in images, pooling layers that reduce dimensionality while preserving critical features, and fully connected layers that synthesize these features into predictions [8,9]. This advanced architecture allows CNNs to tackle complex image recognition tasks that are challenging for traditional machine learning (ML) approaches [10].

Medical image analysis requires robust algorithms capable of extracting subtle patterns from high-dimensional and often noisy datasets, a challenge that CNNs are uniquely equipped to address. By leveraging their ability to learn intricate features directly from imaging modalities such as X-rays, computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET), CNNs have enabled advancements in automated diagnostics, tumor detection, and precision medicine. These networks are increasingly used in clinical workflows to augment radiologists' expertise, improve accuracy, and reduce interpretation times.

Despite their success in medical applications, CNNs are not without limitations. Challenges such as the lack of large annotated medical datasets, model interpretability, and ethical concerns remain significant barriers to widespread adoption in clinical practice. Overcoming these issues requires continued innovation in architecture design, data augmentation techniques, and explainable AI frameworks [16]. Furthermore, the rapid evolution of CNN architectures, such as U-Net for segmentation tasks and DenseNet for feature reuse, highlights the need for a comprehensive review focused on their medical applications. [28]

## **2.2 Convolution neural network (CNN)**

-A convolutional neural network is a type of deep learning algorithm that is most often applied to analyze and learn visual features from large amounts of data. While primarily used for image-related AI applications, CNNs can be used for other AI tasks, including natural language processing and in recommendation engines. [29]

### **2.2.1 CNN in image analyses**

While CNNs are designed to solve problems with visual imagery, they also have many applications outside of image recognition and analysis, including image classification, natural language processing, drug discovery, and health risk assessments [30]

tasks of CNNs in the field of medical imaging mainly include disease classification and grading, localization and detection of pathological targets, organ region segmentation, and image denoising, enhancement, and fusion to assist clinicians in efficient and accurate image processing. [31]

It automatically extracts and selects features and classifies them, which provides radiologists with quicker and more accurate findings in real time. Actually, as an important branch of deep learning, CNN is increasingly used in various computer vision problems, especially in medical imaging domain. [32]

### **2.2.2 Difference between CNN and other algorithms**

#### **A. Difference between CNN and RNN**

Two categories of algorithms that have propelled the field of AI forward are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Compare how CNNs and RNNs work to understand their strengths and weaknesses, including where they can complement each other.

The main differences between CNNs and RNNs include the following:

- CNNs are commonly used to solve problems involving spatial data, such as images. RNNs are better suited to analyzing temporal and sequential data, such as text or videos.
- CNNs and RNNs have different architectures. CNNs are feedforward neural networks that use filters and pooling layers, whereas RNNs feed results back into the network.
- In CNNs, the size of the input and the resulting output are fixed. A CNN receives images of fixed size and outputs a predicted class label for each image along with a confidence level. In RNNs, the size of the input and the resulting output can vary.
- Common use cases for CNNs include facial recognition, medical analysis and image classification. Common use cases for RNNs include machine translation, natural language processing, sentiment analysis and speech analysis. [33]

#### **B. Difference between CNN and GAN**

Convolutional neural networks (CNNs) and generative adversarial networks (GANs) are examples of neural networks, a type of deep learning algorithm modeled after how the human brain works.

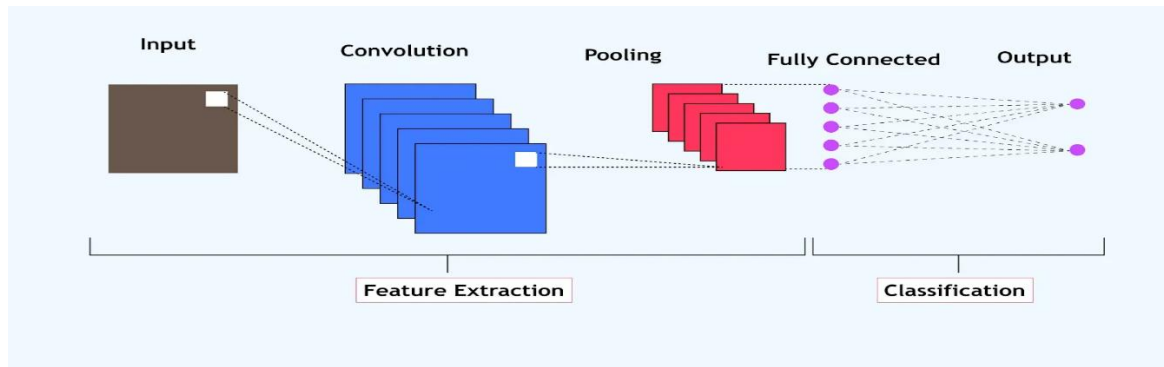
- CNNs, one of the oldest and most popular of the deep learning models, were introduced in the 1980s and are often used in visual recognition tasks. GANs are relatively newer. Introduced in 2014, GANs were one of the first deep learning models used for generative AI. CNNs are sometimes used within GANs to generate and discern visual and audio content.
  - "GANs are essentially pairs of CNNs hooked together in an 'adversarial' way, so the difference is one of approach to output or insight creation, albeit there exists an inherent underlying similarity," said computer pioneer John Blankenbaker, retired principal data scientist at global management consulting firm SSA. "How they answer a given question, however, is slightly different."
  - For example, CNNs might try to determine if a picture contains a cat - a recognition task - while GANs will try to make a picture of a cat, a generation task. In both cases, the networks are building up a representation of what makes a picture of a cat distinctive.
- [34]

### **2.3 Architecture of CNN**

Convolutional Neural Networks (CNNs) are a type of deep learning model used for image recognition, processing, and classification. With basic CNN architecture, you can automatically and efficiently extract features from input data. But what is CNN in machine learning?

CNNs are a key technique in machine learning and deep learning, specializing in processing grid-like data such as images. Unlike traditional models like decision trees or SVMs, CNNs use filters to detect patterns like edges or shapes automatically.

This efficient feature extraction allows CNNs to handle complex visual data, making them ideal for tasks like image classification over other algorithms. [38]



**Fig.2.1 Architecture of convolutional neural networks**

The basic functionality of the example CNN above can be broken down into four key areas.

1. Input Layer
2. The convolutional layer
3. The pooling layer
4. The fully-connected

#### **A. The convolutional layer**

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

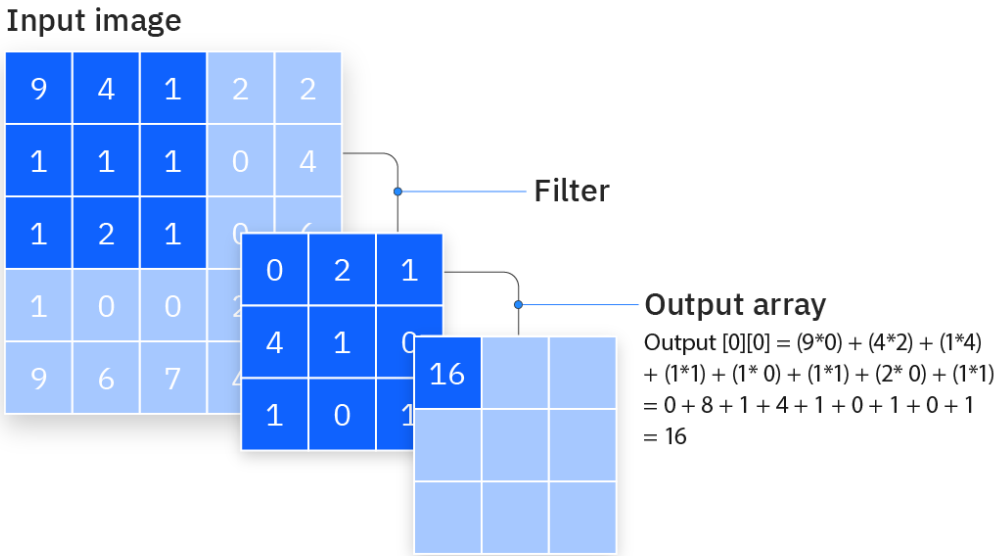
The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map or a convolved feature.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters such as the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three

hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

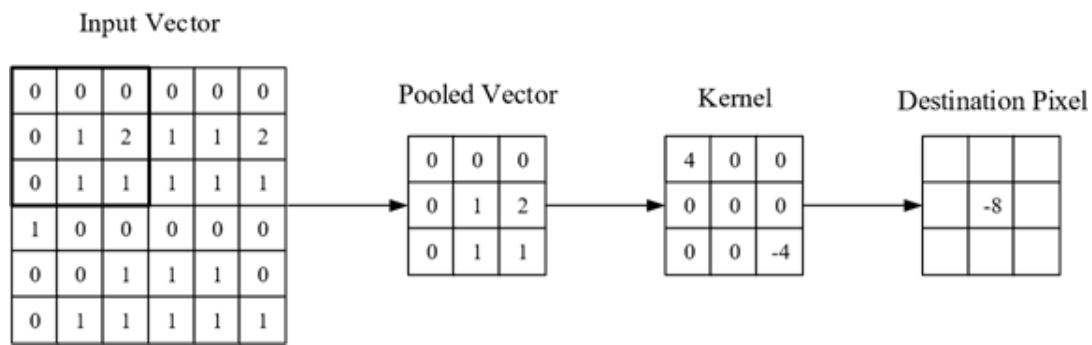
1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.
2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:
  - Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
  - Same padding: This padding ensures that the output layer has the same size as the input layer.
  - Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model. [40]



**Fig.2.2 Filtering operation**

## B. The pooling layer



**Fig.2.3 Pooling operation**

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model.

The pooling layer operates over each activation map in the input, and scales its dimensionality using the “MAX” function. In most CNNs, these come in the form of max-pooling layers with kernels of a dimensionality of  $2 \times 2$  applied with a stride of 2 along the spatial dimensions of the input. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size.

Due to the destructive nature of the pooling layer, there are only two generally observed methods of max-pooling. Usually, the stride and filters of the pooling layers are both set to  $2 \times 2$ , which will allow the layer to extend through the entirety of the spatial dimensionality of the input. Furthermore overlapping pooling may be utilised, where the stride is set to 2 with a kernel size set to 3. Due to the destructive nature of pooling, having a kernel size above 3 will usually greatly decrease the performance of the model.

It is also important to understand that beyond max-pooling, CNN architectures may contain general-pooling. General pooling layers are comprised of pooling neurons that are able to perform a multitude of common operations including L1/L2-normalisation, and average pooling. [41]

## C. The fully-connected

The fully connected layer plays a critical role in the final stages of a CNN, where it is responsible for classifying images based on the features extracted in the previous layers. The term fully connected means that each neuron in one layer is connected to each neuron in the subsequent layer.

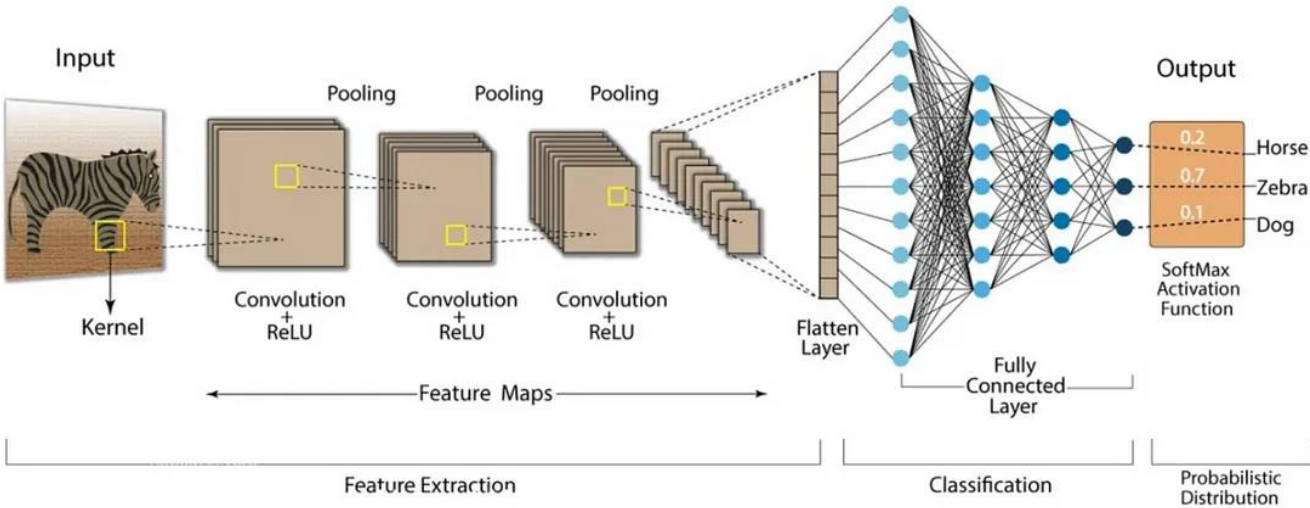
The fully connected layer integrates the various features extracted in the previous convolutional and pooling layers and maps them to specific classes or outcomes. Each input from the previous layer connects to each activation unit in the fully connected layer, enabling the CNN to simultaneously consider all features when making a final classification decision.

Not all layers in a CNN are fully connected. Because fully connected layers have many parameters, applying this approach throughout the entire network creates unnecessary density, increases the risk of overfitting and makes the network expensive to train in terms of memory and compute. Limiting the number of fully connected layers balances computational efficiency and generalization ability with the capability to learn complex patterns. [42]

**D. Output Layer**

The Output Layer is the last layer of the CNN, where the final decision is made. This layer usually consists of neurons equal to the number of possible classes (for example, in a binary classification problem, there are two neurons).

- **Activation Function:** The output layer typically uses an activation function like softmax for multi-class classification problems or sigmoid for binary classification. These functions convert the output values into probabilities.
- **Purpose:** The output layer is where the CNN makes its prediction, based on the features learned throughout the network. [43]



**Fig.2.4 Steps of CNN**

## 2.4 Mathematical Formulation

### A. The Convolution Operation

Convolutional networks (LeCun, 1989), also known as convolutional neural networks, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. [44]

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n)$$

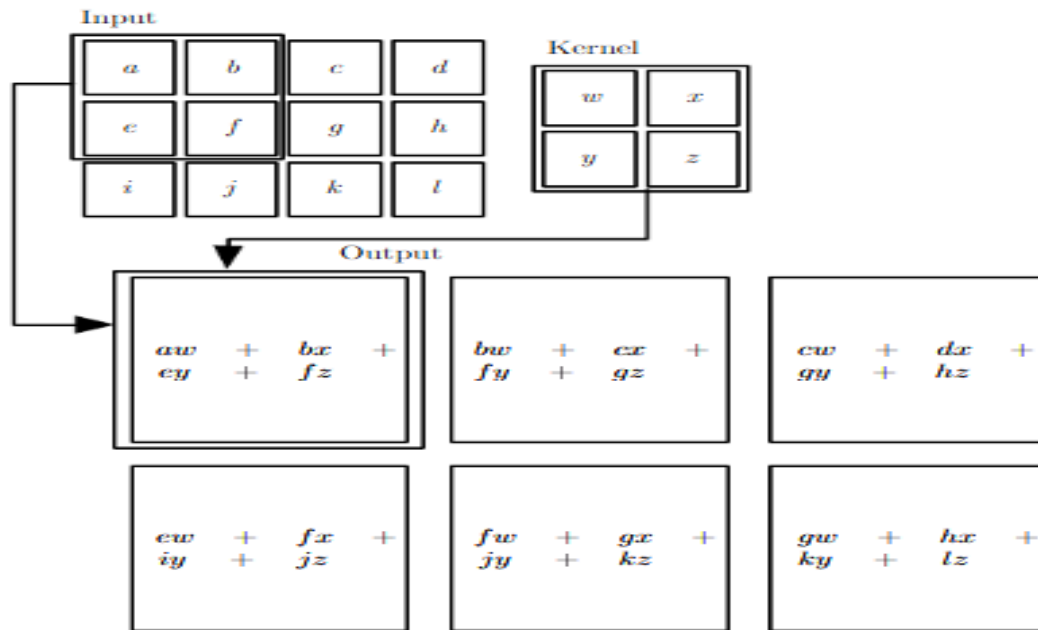
(2.1)

Where:

- $I$  : the input image (2D matrix)
- $K$  : the convolution kernel (filter)
- $S(i, j)$  : the output (feature map) at position  $(i, j)$
- $m, n$ : indices over the kernel size

### B. Motivation of The Convolution Operation

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant



**Fig. 2 .5 Example of 2-D convolution**

An example of 2-D convolution without kernel flipping. We restrict the output to only positions where the kernel lies entirely within the image, called “valid” convolution in some contexts. We draw boxes with arrows to indicate how the upper-left element of the output tensor is formed by applying the kernel to the corresponding upper-left region of the input tensor.

### C. Pooling layer

Feature motifs, which result as an output of convolution operation, can occur at different locations in the image. Once features are extracted, its exact location becomes less important as long as its approximate position relative to others is preserved. Pooling or down-sampling is an interesting local operation. It sums up similar information in the neighborhood of the receptive field and outputs the dominant response within this local region . [45]

$$s(i, j) = \max_{(m, n) \in R(i, j)} (I(m, n)) \quad (2.2)$$

Where:

- $S(i, j)$ : output value at position  $(i, j)$
- $I(m, n)$ : values from the input feature map in region  $R$
- $R(i, j)$ : the  $k \times k$  pooling window centered at  $(i, j)$  .

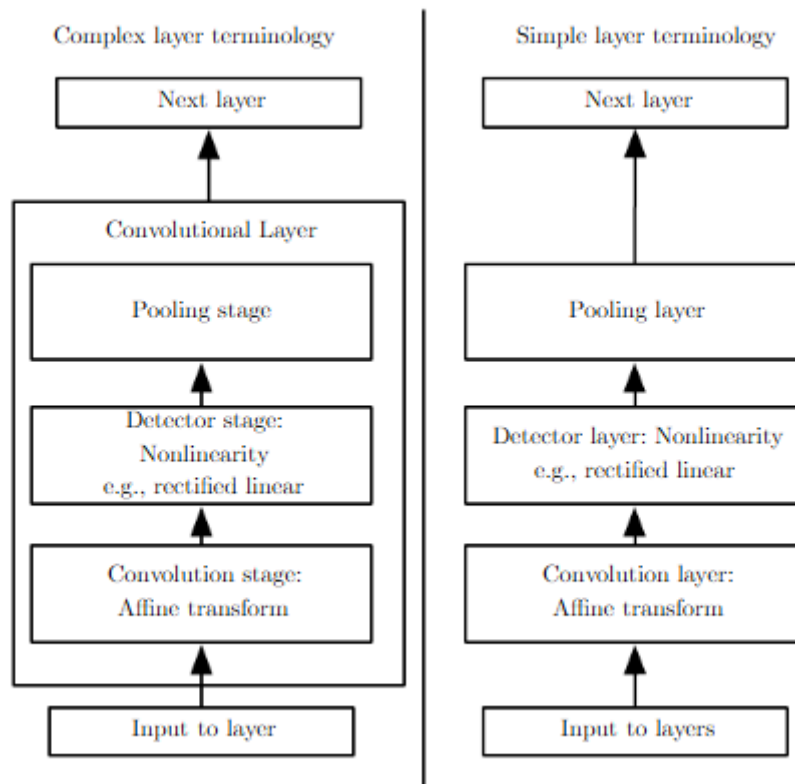
Average Pooling Equation :

$$s(i, j) = \frac{1}{K} \sum_{(m,n) \in R(i,j)} I(m, n) \quad (2.3)$$

**Note:**

- Pooling is non-parametric (no learning involved).
- It is usually applied with a stride  $> 1$  (e.g., stride = 2) to reduce spatial dimensions.

typical layer of a convolutional network consists of three stages In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a nonlinear activation function, such as the rectified linear activation function. This stage is sometimes called the detector stage. In the third stage, we use a pooling function to modify the output of the layer further. A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling(Zhouand Chellappa, 1988) operation reports the maximum output within a rectangular neighborhood. Other popular pooling functions include the average of a rectangular neighborhood, theL2norm of a rectangular neighborhood, or a weighted averagebased on the distance from the central pixel [46]



**Fig.2.6 Plan of CNN**

## 2.5 CNN for Image Classification

It works by breaking the image into small pieces and looking for patterns. These patterns help it recognize things in the image, like edges or shapes. CNNs are used in many applications, like facial recognition, self-driving cars, and medical image analysis. [47]

It has already been confirmed that a convolutional neural network (CNN or ConvNet) can be used in image classification. However, image classification with deep convolutional neural networks hasn't been addressed exactly.

As an example, focusing on image classification with convolutional neural networks allows users to input examples of images with vehicles in them and not overwhelm the neural network. In a standard neural network, the larger the image the more time it will take to process what we're hoping to extrapolate from the data. Image classification using convolutional neural networks makes this faster as each layer is only looking for a specific set of data and then passing the data along to the next layer.

While a standard neural network is still going to be able to process the image dataset, it will take longer and may not create the desired results in a timely manner. Image classification with deep convoluted neural networks, though, will be able to handle more images in a shorter timeframe to be able to quickly identify the types of vehicles being shown in an image and classify them appropriately.

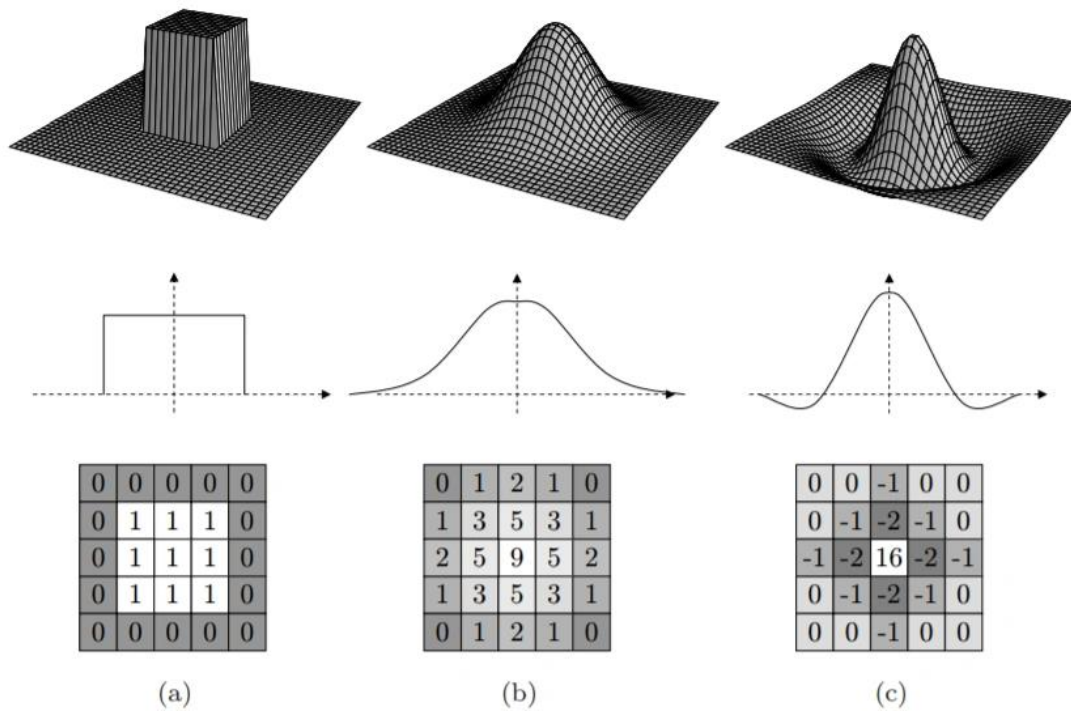
The applications for image classification with deep convolutional neural networks are endless once a model is properly trained, so let's touch on what the best learning process is for image classification. [48]

Image classification using CNN involves the extraction of features from the image to observe some patterns in the dataset. Using an ANN for the purpose of image classification would end up being very costly in terms of computation since the trainable parameters become extremely large.

For example, if we have a 50 X 50 image of a cat, and we want to train our traditional ANN on that image to classify it into a dog or a cat the trainable parameters become –

$(50*50) * 100$  image pixels multiplied by hidden layer + 100 bias + 2 \* 100 output neurons + 2 bias = 2,50,302

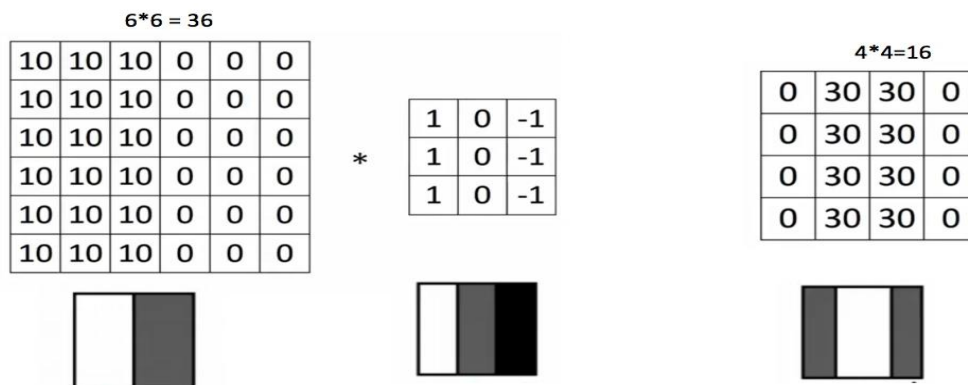
We use filters when using CNN for image classification. Filters exist of many different types according to their purpose.



**Fig.2.7 Types of filters**

Filters help us exploit the spatial locality of a particular image by enforcing a local connectivity pattern between neurons. Additionally, these filters enable the model to capture intricate details and spatial relationships within the image.

Convolution basically means a pointwise multiplication of two functions to produce a third function. Here one function is our image pixels matrix and another is our filter. We slide the filter over the image and get the dot product of the two matrices. The resulting matrix is called an “Activation Map” or “Feature Map”. [49]



**Fig.2.8 After filtering**

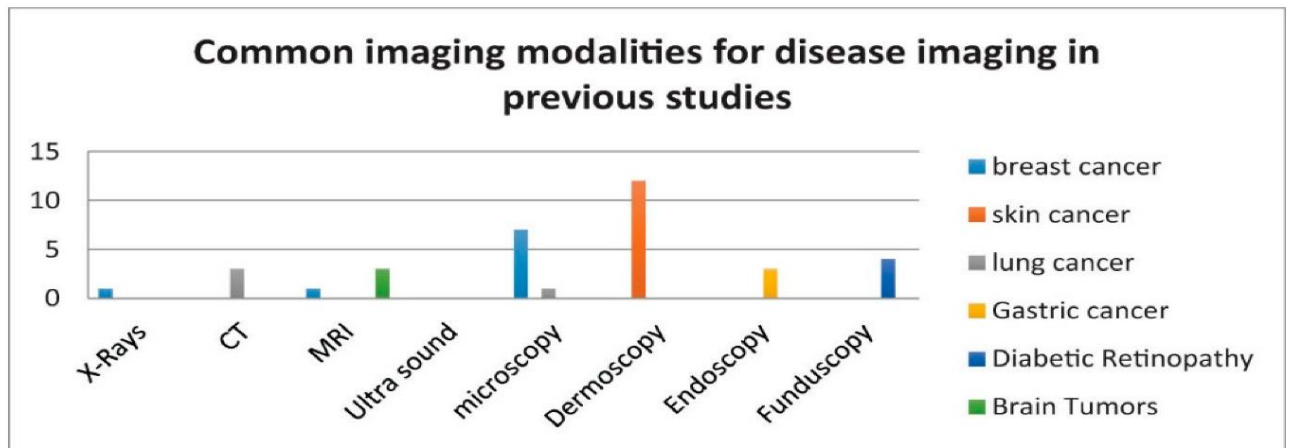
## **2.6 CNN in Medical Imaging**

### **2.6.1 Medical image understanding**

Medical imaging is necessary for the visualization of internal organs for the detection of abnormalities in their anatomy or functioning. Medical image capturing devices, such as X-ray, CT, MRI, PET and ultrasound scanners capture the anatomy or functioning of the internal organs and present them as images or videos. The images and videos must be understood for the accurate detection of anomalies or the diagnosis of functional abnormalities. If an abnormality is detected, then its exact location, size and shape must be determined. These tasks are traditionally performed by the trained physicians based on their judgment and experience. Intelligent healthcare systems aim to perform these tasks using intelligent medical image understanding. Medical image classification, segmentation, detection and localization are the important tasks in medical image understanding pooling [50]

### **2.6.2 Imaging Modalities for Analytics and Diagnostics**

To create an image, several methods are used. Examples of these measurements are radiofrequency signal capacity in an MRI, sound pressure for ultrasounds, and radiation absorption in X-ray imaging. In a digital image, one measurement is used to determine each image point, while in multi-channel images, several measurements are gathered [18]. To create diagnostic images, a wide variety of imaging modalities are employed, such as computed tomography (CT), X-ray, magnetic and functional resonance imaging (MRI and fMRI) [19], and positron emission tomography (PET) scans [4,5]. Common DL applications using medical imaging include image classification [11], segmentation, synthesis, and regression [12,20]. Figure 2.9 depicts various imaging modalities used [21]. In the initial phase of using more precise imaging methods to halt the spread of disease, medical imaging techniques are crucial as an aid to early diagnosis in the treatment or eradication of many medical disorders. [51]



**Fig.2.9 Imaging Modalities for Analytics and Diagnostics**

### 2.6.3 CNN on medical image classification

With the different CNN-based deep neural networks developed and achieved a significant result on ImageNet Challenger, which is the most significant image classification and segmentation challenge in the image analyzing field [27]. The CNN-based deep neural system is widely used in the medical classification task. CNN is an excellent feature extractor, therefore utilizing it to classify medical images can avoid complicated and expensive feature engineering. Qing et al. [28] presented a customized CNN with shallow ConvLayer to classify image patches of lung disease. The authors also found that the system can be generalized to other medical image datasets. Moreover, in other research, it also found that CNN based system can be trained from big chest X-ray (CXR) film dataset and state-of-art with high accuracy and sensitivity results on their dataset, like Stanford Normal Radiology Diagnostic Dataset containing more than 400,000 CXR and a new CXR database (ChestX-ray8), which consist of 108,948 frontal-view CXR [29]. Moreover, using limited data makes it hard to train an adequate model. Therefore the transfer learning of CNN is wildly used in medical image classification tasks. Kermany et al. [3] use InceptionV3 with ImageNet trained weight and transfer learning on a medical image dataset containing 108,312 optical coherence tomography (OCT) images. They got an average accuracy of 96.6%, with a sensitivity of 97.8% and a specificity of 97.4%. The authors also compared the results with six human experts. Most of the experts got high sensitivity but low specificity, while the CNN-based system got high values on both sensitivity and specificity. Moreover, on the average weight error measure, the CNN-based system exceeds two human experts. The authors also verified their system on a small pneumonia dataset, including about five thousand images, and achieved an average accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1%. This system finally may help in accelerating diagnosis and referral

of patients and therefore introduce early treatment, resulting in an increased cure rate. Moreover, Vianna [30] also studied how to utilize transfer learning to build an X-ray image classification system that is the critical component of a computer-aided-diagnosis system. The authors found a fine-tuned transfer learning system with data augmentation effectively alleviate overfitting problem and yield a better result than two other models: training from scratch and a transfer learning model with only a retrained last classification layer.[52]

#### 2.6.4 Medical image classification

Medical image classification involves determining and assigning labels to medical images from a fixed set. The task involves the extraction of features from the image, and assigning labels using the extracted features. Let  $I$  denote an image made of pixels and  $c_1, c_2, \dots, c_r$  denote the labels. For each pixel  $x$ , a feature vector  $\zeta$ , consisting of values  $f^{\{f_0\}}(x_i)$  is extracted from the neighborhood  $N(x)$  using (1), where  $x_i \in N^{\{f_0\}}(x)$  for  $i=0, 1, \dots, k$ .

$$\zeta = (f(x_0), f(x_1), \dots, f(x_k)) \quad (2.4)$$

A label from the list of labels  $c_1, c_2, \dots, c_r$  is assigned to the image based on  $\zeta$ .

#### 2.6.5 Medical image segmentation

Medical image segmentation helps in image understanding, feature extraction and recognition, and quantitative assessment of lesions or other abnormalities. It provides valuable information for the analysis of pathologies, and subsequently helps in diagnosis and treatment planning. The objective of segmentation is to divide an image into regions that have strong correlations. Segmentation involves dividing the image  $I$  into a finite set of regions  $R_1, R_1, \dots, R_S$  as expressed in .

$$S_{i,j,k} = \frac{e^{z_{i,j,k}}}{\sum_{c=1}^K e^{z_{i,j,c}}} \quad (2.5)$$

$z_{i,j,k}$  the raw output (logit) for pixel  $(i,j)$  and class  $k$ ,

$S_{i,j,k}$  is the predicted probability that pixel  $(i,j)$  belongs to class  $k$

Loss Function (Cross-Entropy Loss):

$$\tau = - \sum_{l=1}^H \sum_{j=1}^W \sum_{k=1}^K y_{i,j,k} \log (S_{i,j,k}) \quad (2.6)$$

Where:

- $y_{i,j,k} \in \{0,1\}$  is the ground truth one-hot encoded label for pixel  $(i,j)$ .
- $S_{i,j,k}$  is the predicted probability.

## 2.7 Conclusion

CNNs have become indispensable tools in medical image analysis, enabling accurate classification, detection, and segmentation. Their success, however, depends on data availability, interpretability, and clinician adoption. Strategies like transfer learning and GANs help mitigate data scarcity, while explainable AI and interdisciplinary training programs (e.g., the UK Topol Fellowship) facilitate integration into clinical workflows [36], [37], [54]. As CNN technology advances, its role in diagnostics and treatment planning is set to expand significantly.

---

# **Chapter 3**

## **CNN for Skin Cancer Detection**

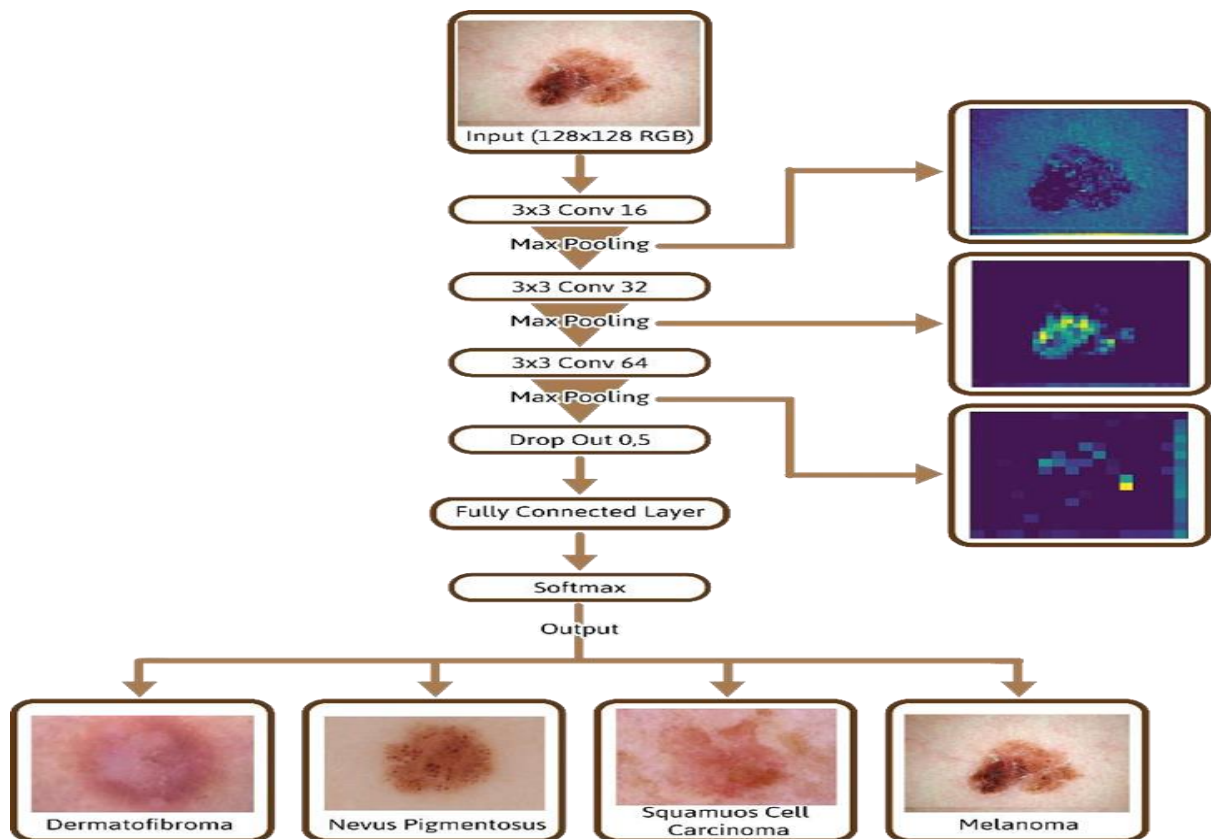
---

### 3.1 Introduction

Convolutional neural networks (CNNs) have demonstrated remarkable capability in detecting skin cancer. By analyzing and learning from vast datasets of dermatological images, CNNs can effectively identify patterns and features indicative of various skin conditions, including melanoma and other types of skin cancer.

These networks excel in recognizing irregularities in moles or lesions, leveraging their ability to extract hierarchical features from images and classify them based on learned patterns. Their accuracy in detecting skin cancer has shown promising results comparable to or even surpassing that of trained dermatologists, showcasing their potential as a valuable tool in early diagnosis and improving healthcare outcomes for patients. [48]

### 3.2 System design



**Fig.3.1 System design**




### 3.3 Early Detection of Skin Cancer

The world's most common cancer is a relentless disease that strikes one in five people by age 70. The good news is that most cases are curable if they are diagnosed and treated early enough. But in order to stop skin cancer, we have to spot it on time.

Skin cancer is the cancer you can see. Unlike cancers that develop inside the body, skin cancers form on the outside and are usually visible. That's why skin exams, both at home and with a dermatologist, are especially vital.

Early detection saves lives. Learning what to look for on your skin gives you the power to detect cancer early when it's easiest to cure, before it can become dangerous, disfiguring, or deadly. [50]

### 3.4 Type of skin cancer

Skin Cancer Type [60]	Appearance / Development	Description	Metastasis
<b>Basal cell carcinoma (BCC)</b> 	It usually appears as a raised, pearly bump or a flat, scaly patch. Usually occurs in <b>sun-exposed areas</b> of the skin.	The <b>most common type</b> of skin cancer.	Unlikely to spread when treated early.
<b>Squamous cell carcinoma (SCC)</b> 	It usually appears as a firm, red nodule or a scaly patch. Usually occurs in <b>sun-exposed areas</b> of the skin.	<b>Less common</b> than basal cell carcinoma	Unlikely to spread when treated early.
<b>Melanoma</b> 	It usually appears as a mole that changes in size, shape, or colour. Can <b>develop anywhere</b> in a person's body. <b>In men:</b> face or the trunk. <b>In women:</b> the lower region or the legs	The <b>most dangerous and uncommon type</b> of skin cancer.	<b>Highly aggressive</b> and can <b>spread</b> to other parts of the body <b>if not treated early</b>

### **3.5 Development environment**

#### **a. Anaconda**

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages.

The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages, and update them.[56]

#### **b. Jupyter**

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality. [57]

#### **c. Kaggle**

is the world's largest data science and machine learning community. It offers tools and resources to help data scientists and machine learning practitioners explore, analyze and share data, and build models in a collaborative environment.

One of the key resources Kaggle offers is public datasets. For instance, Kaggle has hosted challenges and shared datasets such as the “*SIIM-ISIC Melanoma Classification*” competition, which provides thousands of dermoscopic images to train machine learning models to classify skin lesions and detect melanoma.

These datasets and collaborative tools make Kaggle a vital platform for researchers and students working on **skin cancer detection using deep learning**, allowing them to test and improve convolutional neural network (CNN) models in a reproducible environment.[58]

### 3.6 Libraries used

- a. **TensorFlow:** can be used to develop models for various tasks, including natural language processing, image recognition, handwriting recognition, and different computational-based simulations such as partial differential equations. [59]
- b. **Numpy:** NumPy is the fundamental package for scientific computing with Python. It contains, among other things, a powerful N-dimensional array object and sophisticated broadcasting functions, NumPy enables efficient numerical computation in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. [60]
- c. **Scikit-learn:** also known as sklearn, is an open-source, machine learning and data modeling library for Python. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python libraries, NumPy and SciPy.
- d. Scikit-learn was first released in 2010, and it has since gained a prominent place in the Python machine learning ecosystem. It implements numerous data modeling and machine learning algorithms, and provides consistent Python APIs. It supports a standardized and concise model interface across models. For example, Scikit-learn makes use of a simple fit/predict workflow model for its classification algorithms.[61]
- e. **Matplotlib:** is a Python library used to create a wide variety of charts and plots, including line graphs, bar charts, scatter plots, histograms, and more. It is widely used in data science and machine learning to visualize data and model performance.[62]
- f. **OS:** The OS module in Python is a part of the standard library of the programming language. When imported, it lets the user interact with the native OS Python is currently running on. In simple terms, it provides an easy way for the user to interact with several os functions that come in handy in day to day programming. [63]

### **3.7 Python in Skin Cancer Detection using Deep Learning**

Python is the most widely used programming language in the field of deep learning and medical image analysis due to its simplicity, rich libraries, and strong community support. In the task of skin cancer detection, Python plays a central role across all stages of model development and deployment.

#### **Key Roles of Python :**

##### **1. Data Handling & Preprocessing:**

- Libraries: pandas, numpy, opencv, PIL
- Tasks: loading dermoscopic images, resizing, normalization, augmentation

##### **2. Model Building (CNN):**

- Libraries: TensorFlow, Keras, PyTorch
- Define convolutional layers, activation functions, pooling, and fully connected layers

##### **3. Training & Evaluation:**

- Tools: sklearn, matplotlib, seaborn
- Visualizing accuracy/loss curves, confusion matrices, ROC curves

##### **4. Dataset Management:**

- Loading datasets like Melanoma Cancer Image Dataset from Kaggle using kaggle API or tensorflow\_datasets

##### **5. Model Deployment (Optional):**

- Tools: Flask, Streamlit, ONNX
- Build web apps or deploy the model in clinical tools

#### **Python characteristics:**

- Easy syntax for fast prototyping
- Open-source and widely adopted in academia and industry
- Powerful visualization and debugging tools
- Compatible with GPU acceleration (via CUDA & TensorFlow/PyTorch)

### **3.8 Melanoma Cancer Image Dataset of 10000 images on Kaggle**

The Melanoma Skin Cancer Dataset is one of the most prominent open-source databases available on the Kaggle platform, dedicated to research and development in the field of automated skin cancer detection using computer vision and deep learning techniques. This collection provides a rich resource of encoded and documented images of different skin lesions, making it highly suitable for training classification models based on convolutional neural networks (CNNs).

#### **3.8.1 Database Content**

**Number of Images:** The collection contains approximately 10,000 high-resolution skin images of suspicious skin lesions.

**Category Types:** The data is divided into two main categories:

- **Benign (non-cancerous):** These are non-cancerous lesions.
- **Malignant (malignant):** These are conditions that show signs of skin cancer (such as melanoma).

**File Organization:** Images are organized into separate folders by category (benign or malignant), making them easier to use in classification models.

**Image Type:** Images are captured using dermoscopic imaging, a common technique for diagnosing skin cancer.

**Metadata (if available):** Some versions of the collection contain metadata files that include additional information such as the patient's age, gender, and location of the lesion on the body.

#### **3.8.2 The scientific value of the dataset:**

- A wide variety of skin types and skin lesions enhances the ability of trained models to generalize when tested on new data.
- The availability of a large number of images enables the training of deep models while maintaining generalization ability and reducing the risk of overfitting.
- The availability of both malignant and benign cases enables the use of binary classification techniques.

#### **3.8.3 Database Challenges**

The dataset contains a larger number of benign images than malignant images, which can bias the model toward the more benign class.

This requires the use of techniques such as resampling or weighted loss functions during training.

Variation in Lighting and Image Quality:

Some images vary in lighting or focus, requiring preprocessing techniques such as lighting uniformity or noise removal.

Skin Tone Variability:

Significant variation in skin tone can impact model performance if it is not trained to adequately represent all skin types, especially darker skin, which may be underrepresented.

Examples of preprocessing methods:

1. Resizing images to standardize the input.
2. Contrast enhancement to highlight skin lesion features.
3. Normalization of pixel values between 0 and 1.
4. Augmentation (such as rotation and reflection) to increase data size and improve generalization.

### 3.8.4 Research Uses

This suite has been used in several studies, Kaggle projects, and academic research, and several CNN-based models have achieved accuracy exceeding 85% in case classification. Modern techniques such as ResNet, VGG, and MobileNet have been employed with encouraging results in early melanoma detection. [52-53-54]

## 3.9 Experimentation and results

### 3.9.1 Import the Libraries

```
[1]: import os
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers, models
from sklearn.metrics import classification_report
```

**Fig 3.2 Import the Libraries**

We import the necessary libraries for work:

tensorflow/keras: To build a CNN model.

matplotlib: to plot accuracy and loss curves.

sklearn.metrics: To get the model accuracy report (Classification Report).

os, numpy: to work with files and arrays.

### 3.9.2 Configuration Data set

```
▶ base_dir = '/kaggle/input/melanoma-cancer-dataset'  
  train_dir = os.path.join(base_dir, 'train')  
  test_dir = os.path.join(base_dir, 'test')
```

**Fig 3.3 Configuration Data set**

Locate the images within the project folder.

train\_dir contains training images.

test\_dir contains the evaluation images.

### 3.9.3 Data preprocessing

```
# 3. إعداد البيانات  
img_size = (128, 128)  
batch_size = 16  
  
train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=15,  
                                   zoom_range=0.1, horizontal_flip=True)  
test_datagen = ImageDataGenerator(rescale=1./255)
```

**Fig 3.4 Data preprocessing**

ImageDataGenerator Used to load images and convert them into a trainable form.

rescale=1./255: Normalize images (from 0-255 to 0-1).

rotation, zoom, horizontal\_flip: Data augmentation techniques to strengthen the model and avoid over-learning.

We use different settings for training and testing.

```

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='binary'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False
)

```

**Fig 3.5 Data preprocessing**

### 3.9.4 Building a CNN neural network model

```

# 4. بناء نموذج مبسط CNN
model = models.Sequential([
    layers.Conv2D(16, (3, 3), activation='relu', input_shape=(128, 128, 3)),
    layers.MaxPooling2D(2, 2),
    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(1, activation='sigmoid')
])

```

**Fig 3.6 Building a CNN neural network model**

The model consists of:

3 Conv2D layers capture visual patterns in images.

3 MaxPooling2D layers to reduce dimensions and increase efficiency.

Flatten layer to convert the output to a vector.

Dense layer for classification, with Dropout to avoid over-learning.

A final Sigmoid layer to classify the image into one category (benign or malignant).

$$z = X * W + b \quad (3.1)$$

X is the input,

W is the filter (kernel),

b is the bias, and \* denotes convolution.

Then:  $A = \text{ReLU}(Z)$

### 3.9.5 Assemble the model

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

**Fig 3.7 Assemble the model**

Adam: An efficient algorithm for updating weights.

Binary Crossentropy: A loss function suitable for a binary classification problem.

Accuracy: A performance metric used to track model accuracy.

### 3.9.6 Training model

```
[12]: # 6. التدريب
      history = model.fit(
          train_generator,
          epochs=3,
          validation_data=test_generator
      )

/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
Epoch 1/3
743/743 ————— 181s 240ms/step - accuracy: 0.7326 - loss: 0.5161 - val_accuracy: 0.8070 - val_loss: 0.4083
Epoch 2/3
743/743 ————— 128s 172ms/step - accuracy: 0.8338 - loss: 0.3934 - val_accuracy: 0.8835 - val_loss: 0.3237
Epoch 3/3
743/743 ————— 129s 174ms/step - accuracy: 0.8429 - loss: 0.3645 - val_accuracy: 0.7905 - val_loss: 0.4164
```

**Fig 3.8 Training model**

The model was trained on 3 epochs of training data (Epochs), with performance verified on test data.

Used in training (backpropagation):

$$w_{new} = w_{old} - \eta * \partial L / \partial w \quad (3.2)$$

Where:

-  $\eta$ : learning rate

-  $\partial L / \partial W$ : gradient of the loss with respect to the weights

### 3.9.7 Evaluation using classification

```
[13]: # 7. التقييم السريع على دفعة واحدة فقط
X_sample, y_sample = next(test_generator)
Y_pred = model.predict(X_sample)
y_pred = (Y_pred > 0.5).astype(int)

print("تقرير تصنيف لدفعة واحدة فقط:")
print(classification_report(y_sample, y_pred, target_names=["Benign", "Malignant"]))
```

```
1/1 |-----| 0s 176ms/step
تقرير تصنيف لدفعة واحدة فقط:

```

	precision	recall	f1-score	support
Benign	1.00	0.94	0.97	16
Malignant	0.00	0.00	0.00	0
accuracy			0.94	16
macro avg	0.50	0.47	0.48	16
weighted avg	1.00	0.94	0.97	16

**Fig 3.9 Evaluation using classification**

After training, a batch of test data was used to generate model predictions, and then performance indices were calculated

### 3.9.8 Performance curves

```
# 8. رسم منحنيات الأداء
plt.figure(figsize=(12, 5))

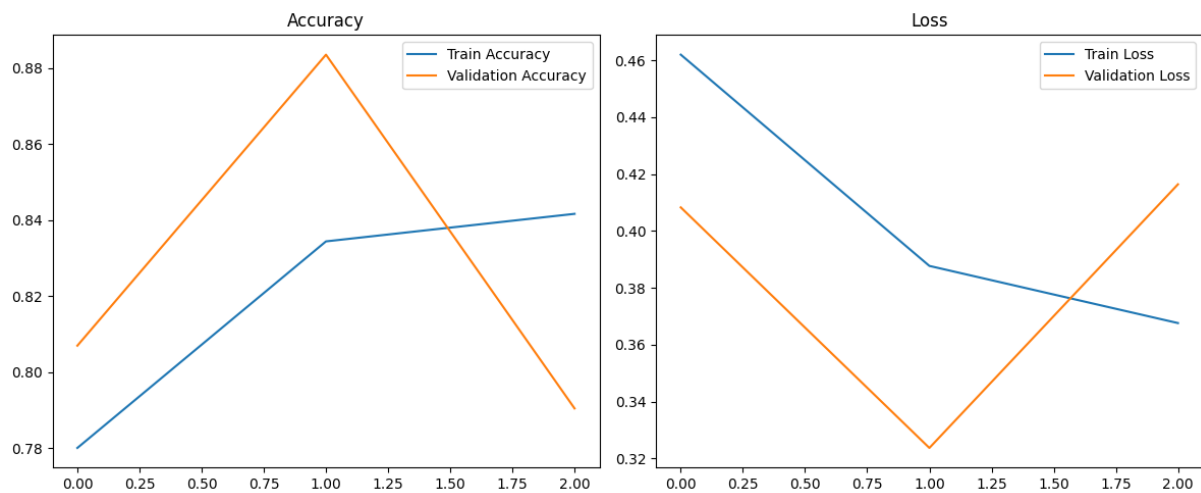
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.title('Accuracy')

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.title('Loss')

plt.tight_layout()
plt.show()
```

**Fig 3.10 Performance curves**

### 3.9.9 Performance Curve Analysis



**Fig 3.11 Performance Curve Analysis**

#### A. Accuracy Curve Analysis

- The x-axis represents the number of training epochs (0 to 2).
- The y-axis shows the accuracy on both training and validation data.

Observations:

- Validation accuracy starts relatively high ( $\sim 0.81$ ), peaks in the second epoch ( $\sim 0.88$ ), and then drops in the third ( $\sim 0.79$ ).
- Training accuracy steadily increases from  $\sim 0.78$  to  $\sim 0.84$ .

Interpretation:

- This pattern indicates a potential beginning of overfitting. The model learns the training data well but starts to lose generalization ability on unseen validation data.

#### B. Loss Curve Analysis

- The x-axis is the number of epochs.
- The y-axis shows the loss value (lower is better).

Observations:

- Training loss decreases consistently ( $\sim 0.47$  to  $\sim 0.35$ ).

- Validation loss reaches its minimum ( $\sim 0.32$ ) in the second epoch, then increases ( $\sim 0.43$ ) in the third.

Interpretation:

- The initial drop shows effective learning.

- The later rise suggests overfitting, where the model performs worse on validation data despite improving on training data.

### **3.10 Challenges and limitations of using CNN for skin cancer detection**

Despite significant progress in the application of deep learning techniques, particularly convolutional neural networks (CNNs), to skin cancer diagnosis, there are still many challenges and limitations that affect the accuracy and reliability of these models.

1. The need for large and diverse data: CNN models require large amounts of accurately labeled data to train effectively. However, obtaining comprehensive datasets that represent a wide variety of skin lesion types, skin colors, and lighting conditions remains a significant challenge. A lack of such data can lead to models that are unable to generalize well to new cases or cases not represented in the training data.
2. Difficulty distinguishing between some skin lesions: Some skin lesions, such as benign moles and melanoma, may look very similar visually, making it difficult for models to accurately distinguish between them. This similarity can lead to high rates of false positives or false negatives, impacting the model's effectiveness in clinical diagnosis.
3. Database-induced bias: Datasets used to train CNN models often contain biases resulting from data collection methods or an unbalanced representation of certain classes. For example, certain skin tones or lesion types may be overrepresented, leading to uneven model performance when applied to different populations.
4. The Need for Medical Intervention for Final Review: Although CNN models can provide valuable diagnostic assistance, they cannot replace the clinical expertise of physicians. Review and evaluation by specialists remain essential to ensure accurate diagnosis and appropriate treatment decisions, especially in complex or unclear cases.

These challenges highlight the importance of developing more complex and flexible models, as well as improving the quality and diversity of datasets used for training, to ensure greater effectiveness and reliability of AI applications in skin cancer diagnosis. [55]

### **3.11 Conclusion**

Convolutional Neural Networks (CNNs) have proven to be highly effective tools for skin cancer detection, especially when dealing with large-scale dermoscopic image datasets. Their ability to automatically learn hierarchical features from raw image data allows them to distinguish between malignant and benign skin lesions with impressive accuracy.

By leveraging powerful architectures and preprocessing techniques, CNNs minimize the need for manual feature engineering and enable consistent, scalable diagnosis support systems. When trained properly, CNNs can assist dermatologists in early melanoma detection, potentially saving lives through timely intervention.

However, challenges remain, such as the need for diverse and balanced datasets, interpretability of model decisions, and integration into real-world clinical workflows. Continued research and refinement are essential to improve generalization and ensure ethical, safe application in healthcare settings.

## **General conclusion**

In this work, we review the application of artificial intelligence techniques, particularly convolutional neural networks (CNNs), in the detection of skin cancer, one of the most common and dangerous types of cancer. We begin by providing an overview of deep learning and its growing role in medical applications. We then move on to an in-depth explanation of the structure and workings of CNNs, focusing on their properties that make them effective in medical image processing.

Real data of skin lesion images was analyzed using well-known datasets such as the Melanoma Cancer Image Dataset, and a CNN model was built to classify these images according to their likelihood of being affected by skin cancer. Training and evaluation results demonstrated that the model was able to achieve acceptable performance in terms of accuracy, sensitivity, and specificity, reflecting the great potential of this type of model in contributing to the early detection of skin cancer, thereby improving treatment and survival rates.

However, several challenges remain, such as the need for large and diverse data, dealing with biases in databases, and difficulties in distinguishing some similar skin lesions. Furthermore, these models still require medical supervision to ensure accurate and reliable diagnosis.

In conclusion, the use of CNNs in medicine, particularly in skin cancer diagnosis, represents a promising step toward a more accurate and effective future in healthcare. Further research and development are needed to improve these models and make them more comprehensive and reliable in various clinical contexts.

## References

- [1] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annu. Rev. Biomed. Eng.*, vol. 19, pp. 221–248, 2017, doi: 10.1146/annurev-bioeng-071516-044442.
- [2] Google Cloud, "What is artificial intelligence?," [Online]. Available: <https://cloud.google.com/learn/what-is-artificial-intelligence>
- [3] IBM, "Artificial Intelligence," [Online]. Available: <https://www.ibm.com/think/topics/artificial-intelligence>
- [4] SAS, "Artificial intelligence, machine learning, deep learning and beyond," [Online]. Available: [https://www.sas.com/en\\_hk/insights/articles/big-data/artificial-intelligence-machine-learning-deep-learning-and-beyond.html](https://www.sas.com/en_hk/insights/articles/big-data/artificial-intelligence-machine-learning-deep-learning-and-beyond.html)
- [5] Oracle, "Définition du deep learning," [Online]. Available: <https://www.oracle.com/fr/artificial-intelligence/definition-deep-learning/>
- [6] IBM, "Deep Learning," [Online]. Available: <https://www.ibm.com/fr-fr/think/topics/deep-learning>
- [7] DataScientest, "Deep neural network: What is it and how is it working?," [Online]. Available: <https://datascientest.com/en/deep-neural-network-what-is-it-and-how-is-it-working>
- [8] M. Vakalopoulou, S. Christodoulidis, N. Burgos, O. Colliot, and V. Lepetit, "Deep learning: basics and convolutional neural networks (CNN)."
- [9] MizeTech, "How does a neural network work?," [Online]. Available: <https://mize.tech/blog/how-does-a-neural-network-work-implementation-and-5-examples/>
- [10] Amazon Web Services, "What is recurrent neural network?," [Online]. Available: <https://aws.amazon.com/what-is/recurrent-neural-network/>
- [11] IBM, "Recurrent neural networks," [Online]. Available: <https://www.ibm.com/think/topics/recurrent-neural-networks>
- [12] How.dev, "What are the types of RNN?," [Online]. Available: <https://how.dev/answers/what-are-the-types-of-rnn>
- [13] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021.
- [14] G. Lee and H. Fujita, Eds., *Deep Learning in Medical Image Analysis: Challenges and Applications*, Springer, 2020, p. 5.
- [15] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 8.
- [16] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 5.
- [17] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 7.

- [18] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 18.
- [19] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, pp. 84–85.
- [20] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 162.
- [21] Q. Dou et al., *Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging*, Springer, 2021, p. 19.
- [22] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "AI-Powered Diagnosis of Skin Cancer: A Contemporary Review, Open Challenges and Future Research Directions," *arXiv preprint*, arXiv:1911.11872, 2019.
- [23] M. Dildar et al., "Skin Cancer Detection: A Review Using Deep Learning Techniques," *Sensors*, vol. 20, no. 22, 2020.
- [24] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "AI-Powered Diagnosis of Skin Cancer: A Contemporary Review, Open Challenges and Future Research Directions," *arXiv preprint*, arXiv:1911.11872, 2019.p. 27.
- [25] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "AI-Powered Diagnosis of Skin Cancer: A Contemporary Review, Open Challenges and Future Research Directions," *arXiv preprint*, arXiv:1911.11872, 2019. p. 32.
- [26] MDPI, "Deep Learning for Medical Imaging," *Information*, vol. 16, no. 3, 2024. [Online]. Available: <https://www.mdpi.com/2078-2489/16/3/195>
- [27] Intel, "Convolutional neural networks," [Online]. Available: <https://www.intel.com/.../convolutional-neural-networks.html>
- [28] ARM, "Convolutional neural networks explained," [Online]. Available: <https://www.arm.com/glossary/convolutional-neural-network>
- [29] National Library of Medicine, "The tasks of CNNs in medical imaging," [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11074758/>
- [30] ScienceDirect, "CNNs in medical imaging," [Online]. Available: <https://www.sciencedirect.com/.../S0010482524015920>
- [31] TechTarget, "CNN vs RNN," [Online]. Available: <https://www.techtarget.com/.../CNN-vs-RNN>
- [32] TechTarget, "CNN vs GAN," [Online]. Available: <https://www.techtarget.com/.../CNN-vs-GAN>
- [32] Kaggle, "General discussion," [Online]. Available: <https://www.kaggle.com/discussions/general/381944>
- [33] AWS, "What is GAN?," [Online]. Available: <https://aws.amazon.com/what-is/gan/>

- [34] Blent.AI, "GAN deep learning," [Online]. Available: <https://blent.ai/blog/a/gan-deep-learning>
- [35] UpGrad, "Basic CNN architecture," [Online]. Available: <https://www.upgrad.com/blog/basic-cnn-architecture/>
- [36] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," 2015, pp. 4–5.
- [37] IBM, "Convolutional neural networks," [Online]. Available: <https://www.ibm.com/think/topics/convolutional-neural-networks>
- [38] O'Shea and Nash, An Introduction to Convolutional Neural Networks. op. cit., p. 8.
- [39] TechTarget, "Definition of CNN," [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>
- [40] R. Ravjot, "Decoding CNNs," *Medium*, [Online]. Available: <https://ravjot03.medium.com/.../1a8806cbf536>
- [41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, p. 326.
- [42] A survey of the recent architectures of deep convolutional neural networks p. 8.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, pp. 335–336.
- [44] Analytics Vidhya, "Learn image classification with CNN," [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/02/...cnn...>
- [45] MDPI, "Sustainability and diagnostics," *Sustainability*, vol. 15, no. 7, 2023. [Online]. Available: <https://www.mdpi.com/2071-1050/15/7/5930>
- [46] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *J. Ambient Intell. Humaniz. Comput.*, 2020.
- [47] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: A survey," *J. Big Data*, vol. 8, 2021.
- [48] E. Topol, "Preparing the healthcare workforce to deliver the digital future," *Postgrad. Med. J.*, vol. 99, no. 1178, pp. 1287–1293, 2023.
- [48] Labellerr, "Detecting skin cancer using CNNs," [Online]. Available: <https://www.labellerr.com/blog/detecting-skin-cancer-using-convolutional-neural-networks...>
- [49] A. Shah et al., "A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN)," *Elsevier*, 2023.
- [50] Skin Cancer Foundation, "Early detection of skin cancer," [Online]. Available: <https://www.skincancer.org/early-detection/>
- [51] Island Hospital, "Skin cancer diagnosis," [Online]. Available: <https://islandhospital.com/skin-cancer/>

- [52] Kaggle, "Melanoma Skin Cancer Dataset of 10,000 Images," [Online]. Available: <https://www.kaggle.com/>
- [53] A. Hekler et al., "Pathologist-level classification of histopathological melanoma images with deep neural networks," *Eur. J. Cancer*, vol. 115, pp. 79–83, 2019.
- [54] N. C. F. Codella et al., "Skin lesion analysis toward melanoma detection," *Nature*, vol. 563, no. 7732, pp. 378–382, 2018.
- [55] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "AI-based image classification for skin cancer diagnosis," *arXiv preprint*, arXiv:1911.11872, 2019.
- [56] Domino, "Anaconda," [Online]. Available: <https://domino.ai/data-science-dictionary/anaconda>
- [57] Project Jupyter, "Jupyter," [Online]. Available: <https://jupyter.org/>
- [58] TensorFlow, "Official site," [Online]. Available: <https://www.tensorflow.org/?hl=fr>
- [59] NumPy, "Documentation index," [Online]. Available: <https://numpy.org/devdocs/index.html>
- [60] Domino, "Scikit-learn (sklearn)," [Online]. Available: <https://domino.ai/data-science-dictionary/sklearn>
- [61] Matplotlib, "Official site," [Online]. Available: <https://matplotlib.org/>
- [62] Kaggle, "About," [Online]. Available: <https://www.kaggle.com/about>
- [63] Edureka, "OS module in Python," [Online]. Available: <https://www.edureka.co/blog/os-module-in-python>