People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research

University of Mohamed Khider – BISKRA

Faculty of Exact Sciences, Science of Nature and Life

Computer Science Department

Order Number : IA28/M2/2022

Thesis submitted in fulfilment of the requirements for obtaining Master degree(2nd Cycle) in Computer Science

Option : Artificial Intelligence

Smart Health Care System For Skin Cancer Diagnosis using CNN architecture

---

Written by : Ben Khalfallah Moufida

Defended in front of the jury composed of:

Merizig Abdelhak	MCB	President
Terrisa Sadek Labib	Professor	Supervisor
Kelfali Toufik	MAA	Examinator

Defended on : 27/06/2022

Abstract

Skin cancer is among the most common malignancies,its prevalence is expected to increase in the coming decade. Artificial intelligence is a promising solution to the problem of providing high-quality care to patients in areas where dermatologists are lacking.

The use of automated applications for accurate classification of skin lesions from digital pictures has made significant progress. The development and construction of a deep learning method for dermoscopic images categorization of the public dataset ISIC 2019 are discussed in this work.

We used the ResNet-18 architecture to build a deep learning model to reliably identify dermoscopic pictures of skin lesions into one of eight disease categories. Using our proprietary model, we reached a balanced validation accuracy of 74% besides to the training accuracy of 100%.

Résumé

Le cancer de la peau est parmi les tumeurs malignes les plus courantes, sa prévalence devrait augmenter au cours de la prochaine décennie. L'intelligence artificielle est une solution prometteuse au problème de fournir des soins de haute qualité aux patients dans les zones où les dermatologues font défaut.

L'utilisation d'applications automatisées pour la classification précise des lésions cutanées à partir d'images numériques a fait des progrès significatifs.Le développement et la construction d'une méthode d'apprentissage profond pour la catégorisation des images dermoscopiques de l'ensemble de données publiques ISIC 2019 sont discutés dans ce travail.

Nous avons utilisé l'architecture ResNet-18 pour construire un modèle d'apprentissage profond afin d'identifier de manière fiable les images dermoscopiques de lésions cutanées dans l'une des huit catégories de maladies.En utilisant notre modèle propriétaire, nous avons atteint une précision de validation équilibrée de 74% en plus de la précision d'entraînement de 100%.

Keywords

CAD : Computer Aided Diagnosis

ISIC 2019 :the International Skin Imaging Collaboration

DL : Deep Learning

ML : Machine Learning

TL : Transfer Learning

CNN : Convolutional neural network

ResNet : Residual Neural Network

AUC : Area under the ROC Curve

ROC :receiver operating characteristic

RNN :recurrent neural networks

lr : Learning Rate

RMSE : Root Mean Squared Error

RL Reinforcement learning

ReLU : Rectified Linear Unit

AI: artificial intelligence

GPU : Graphics processing unit

List of Figures

List of Tables

Contents

General Introduction

1 Background

Nowadays skin cancer has become a public health and financial problem ,it has been tackled by the dermatology field for many years using the same methodology [17]. When we consider that the number of cases diagnosed with skin cancer has increased dramatically over the previous 30 years [46].It becomes even more problematic when money is involved, considering that the public sector spends millions of dollars [8].The patient's specific analysis takes up a large portion of this time. The doctor examines the cancerous tumor and makes decisions based on what he or she finds. If any of these steps could be improved, the dermatology sector as a whole could be seeing a reduction in spending. Dermatology is among the most crucial disciplines of medicine, with chronic illnesses outnumbering hypertension, obesity, and cancer grouped. This is due to the fact that skin diseases are among the most humanity diseases, impacting individuals of different ages, genders, and cultures, impacting around 30 and 70 percent of the global population. It thus indicates at any specified moment, one out of every 3 persons would have an illnesses [6]. As a result, skin illnesses are a worldwide problem, ranking 18th in terms of global health burden [19].

Moreover, because dermatology has a long list of illnesses to address, medical imaging can prove to be a valuable resource. Furthermore , the arena has created its own terminology for defining these lesions. Nevertheless, textual characterisations have determinations, and a good image could successfully supersede many detailed phrases while being immune

to the message carrier's bias.

Furthermore, being aware of new or changing skin growths is recommended as a technique to detect early skin diseases [17]. Specialists still utilize the naked eye first, as well as procedures like ABCDE, which involves Scanning the tissue scope of focus for disparity, boundary variance, uniform shades, huge diameters, and changing skin spots over time. [6]. Medical image processing seems to be identical to naked-eye observation throughout this approach, so same methodologies and implications can be used. This lends credence to the notion that skin cancer can often be seen with the naked eye and medical photography.

In recent years, medical cancer diagnosis has become increasingly sophisticated, with doctors are focusing on generating premature and more precise diagnoses for rescuing patients' lives. These goals can be difficult to achieve, especially in the case of melanoma, the most lethal form of skin malignancies. Melanotic Carcinoma prevalence rate has risen dramatically in recent years as a result of Alters in the weather. Surprisingly, early diagnosis results in a 98% 5-year survival rate.

2 Problematic

Given the difficulties in identifying suspected areas, current project aims to develop a mechanism for categorizing suspected areas into one of 8 disorders of. The classifier seeks to rectify differentiate suspected areas by evaluating dermoscopy pictures using the stipulation of current purpose. Furthermore, this might be a beneficial tool for doctors and patients to utilize on everyday basis diagnosis.

Current research has been conducted with the goal for serving as a starting point to more advanced mechanisms to democratizing and distributing health-care access. A strong lesion classification mechanism could become a catalyst for the development of systems that put the power of early diagnosis and alarm in the hands of patients, even in remote areas where few clinicians can reach them. By the therapy of late-stage illnesses, these technologies have a potential to save many lives and save money.

The relevant research in this topic demonstrated the existance of numerous mechanisms have the ability to solve the current issue , however a significant disctinction proved between shallow and deep machine learning mechanisms.

3 Project Objective

Computer-aided diagnosis tools, which take into account the expert's knowledge, can provide a more objective analysis tool. Deep learning algorithms are an effective way to fill the lack among low-level data with expert interpretation. The ABCDE rule, which involves 4 specifications of a suspected area:boundary,asymmetry, color, differential structures, and changing patches of skin over time, is often used to make a dermatologist's decision. Lesions are able to be categorized into three classes according to a score also to a qualitative description of the lesion: melanoma, benign, or suggested follow-up.

Early automatic diagnostic studies ignored both the qualitative part of expert description and the ambiguity of the information itself. within the current research,a skin cancer diagnosis mechanism is offered according to a multi-input deep learning model. To deal with expert qualitative descriptions, a deep learning classifier is proposed. The model is then provided as a means of generating a resolution.

empirical validation has been performed to the dermoscopy pictures from the International Skin Imaging Collaboration (ISIC 2019) the public dataset . For the ISIC 2019, we get a 100% percent accuracy for dermoscopic images. A comparison of these results to existing methodologies reveals that they ensure higher accuracy rates.

Chapter I

Main Concepts

Introduction

In this chapter, we will outline the primary ideas that we will employ in our system, as well as the interactions between them, beginning with skin cancer and progressing through computer assisted diagnosis systems to deep learning components and aspects, as follows:

1 Skin Cancer

1.1 Definition

Skin carcinoma is the main health trouble worldwide.Skin carcinomas represent roughly one-third of all carcinoma cases.it is defined as the uncontrolled growth of abnormal cells in the epidermis, the skin's epithelial tissue, generated by unrepaired DNA damage that causes genetic changes. These genetic changes induce cells to proliferate and compose malicious tumors.[39].

1.2 Types of skin cancer

The tissue is built of various cells, that are continually changing. Flattening occurs as round basal cells under the surface elevate to supersede dead, flaking squamous cells on the superficial. Melanocytes tan the skin in the sun, while Merkel cells grant skin the capability to sense touch. Whether these cells are harmed, they could develop into skin cancer. Skin carcinomas are all vulnerable and thus should be handled as fast as possible. Nevertheless, melanoma types, medical interventions, and prognoses differ tremendously based on the kinds of cells involved. The following is a list of prevalent melanomas.[39][27]:

- Basal cell carcinoma
- Recurrent basal cell carcinoma
- Squamous cell carcinoma

- Melanoma
- Merkel cell carcinoma
- Kaposi sarcoma (KS)
- Actinic keratosis
- Lymphoma of the skin
- Keratoacanthoma

1.3 Definition of a skin lesion :

Melanoma or a skin legion is a portion of skin that has an abnormal evolution or appearance when combined with the surrounding skin. There are two types of skin lesions: **primary** and **secondary** [13].

- **Primary** Melanomas are abnormal tissue criteria that can be present at birth or grow over time.
- **Secondary** Primary skin lesions that have been agitated or juggled cause skin lesions. For example, if a mole is scuffed till it bleeds, eventually results in a skin lesion, an outer layer that becomes a secondary skin lesion.

Figure I.1: Examing a skin lesion

2 Computer Aided Diagnosis System

- **Computer-aided diagnosis (CAD)** is a method that has the ability to reduce the biases of classical histopathology image analysis. [36].

3 Artificial Intelligence (AI)

3.1 Definition

- **Artificial intelligence** or AI, is an emulation of smart human conduct. It's a machine or a mechanism intended to grasp its environment, understand its conduct, and make the necessary decisions.[11]

4 Machine Learning

4.1 Definition

- **Machine learning** is a subfield of artificial intelligence (AI) that concentrates on creating software that gains knowledge from data and experience and enhances its outcome or predictive accuracy over time without being explicitly trained to do so. [12].

4.2 Machine Learning Process :

ML is the process of training a piece of software, called a model, to make useful predictions from data. A ML model represents the mathematical relationship between the elements of data that an ML system uses to make predictions and passes on a five important steps [26] as shown in figure 1.2 below :

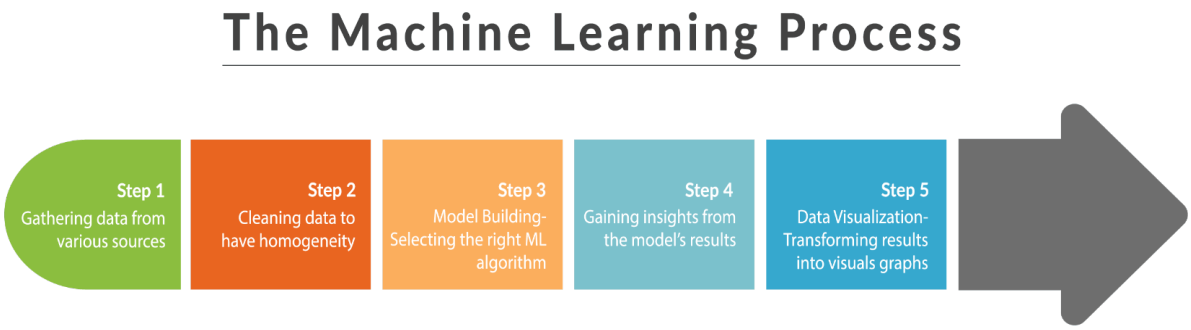

Figure I.2: Machine Learning Process[30]

4.3 Machine Learning Approaches

Machine learning has three main approaches (Supervised, Unsupervised, and Resinforcement learning) and each approach has sub-roots as shown in figure 1.3 below :

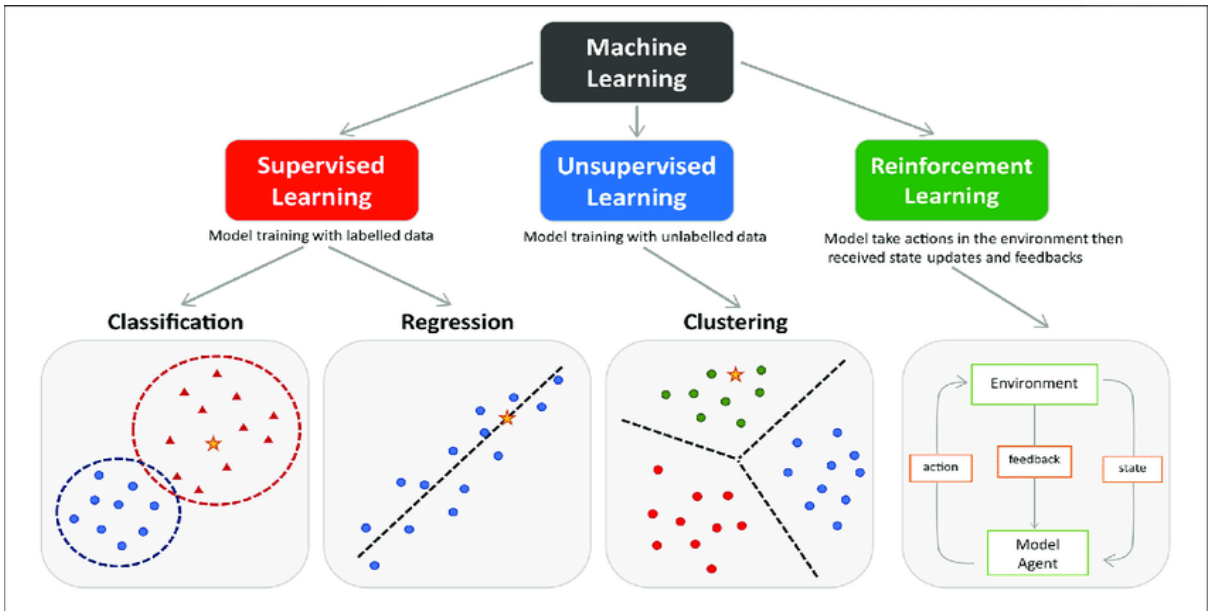

Figure I.3: Machine Learning Approaches[29]

5 Deep Learning

5.1 Definitions:

- **Deep learning** is a type of machine learning that allows machines to learn through experience, allows mechanisms to recognize objects and perform complex tasks with high precision but without human assistance, mimics the human brain, deals with algorithms inspired by the construction and functionalities of the brain, and 'learns' from large amounts of data [4] [5] [15].

5.2 Deep learning Evolution

The very first generation of artificial neural networks (ANNs) was made up of perceptrons in neural layers that were computationally confined [22].

The error rate was determined as well as backpropagated by the 2nd generation. Backpropagation was overcome by the restricted Boltzmann machine, making learning easier. Then, over time, other networks emerge. Figure 2.5 depicts deep models evolution alongside the classic model over time. [22].

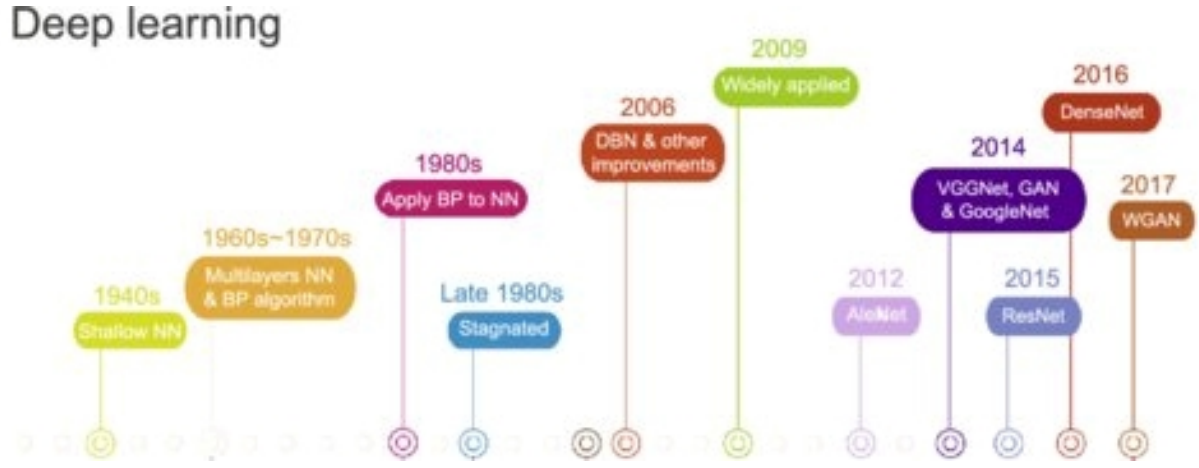

Figure I.4: Timeline of the development of deep learning [9]

Deep learning is now applied in a variety of uses, including Google's facial recognition and speech, Netflix and Amazon's recommendation engines, Apple's Siri fully automated

text and email responses, and virtual agents. [22].

5.3 Applications

Deep learning applications are employed in a variety of industries, ranging from automated driving to medical equipment, and it is frequently used to forecast rain, earthquakes, and tsunamis.Advertisers can use deep learning models to execute real-time bidding and targeted display advertising. Here are some of the most common deep learning applications [35].

- Voice Recognition System , Financial services
- Facial Recognition
- Healthcare , Research
- Law enforcement , Automated Driving
- Aerospace and Defense
- Industrial Automation , Electronics

5.4 The main components of a deep learning structure

Deep learning is a multi-layered neural network that is designed to gain knowledge from massive quantities of data in the same way that the human brain does. Deep learning methods generate calculations and make predictions multiple times within each layer of the neural network, progressively 'acquiring knowledge' and increasing the accuracy of the results over time. Deep learning, acts like the human brain, absorbs and processes information that enters the body through the five senses [18].

1. **Neural Networks**

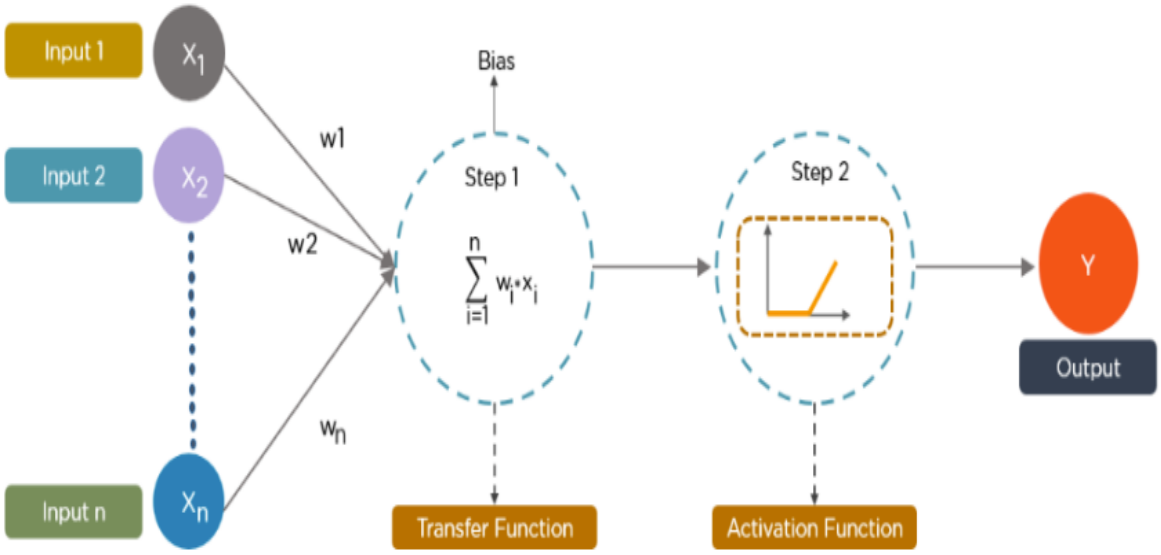

Figure I.5: Structure of an artificial neuron [7]

A perceptron (also known as an artificial neuron) is a linear formula. It requires one or more input data, which are multiplied by "weights" and added together. The results is then passed to a nonlinear function named an activation function, which is the neuron's outcome.[34].

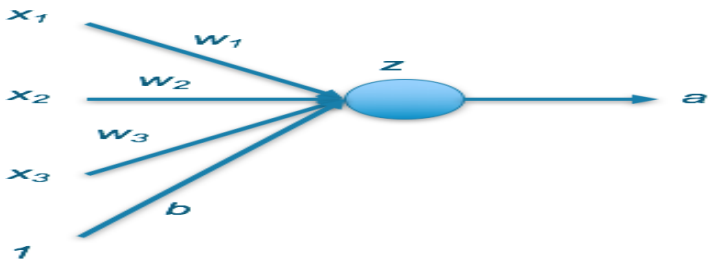

Figure I.6: Neuron's structure [33]

- **x** values are inputs, which can be both the core features or inputs out of a hidden layer . [33]
- a bias b that will aid in fitting the data [33]
- The neuron either passes the value a to all neurons in the next layer , or returns it as the final value.[33].

- A mathematical formula is used to begin the computation.

$$z = x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$

incorporating a non-linear activation function:

$$a = f(z)$$

[33].

In a neural network, 3layers of nodes are arranged beside one another:[7]

- The layer of inputs
- and the middle layers called The hidden layer(s)
- The layer of outputs

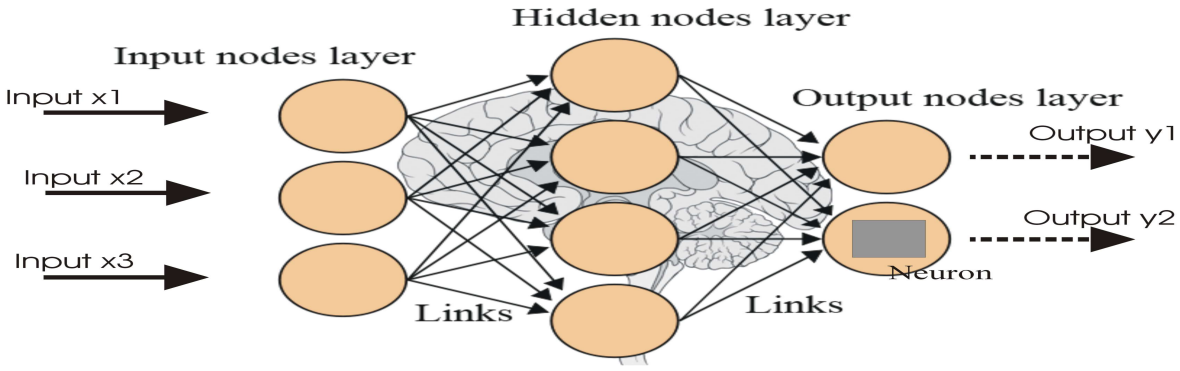

Figure I.7: Basic artificial neural networks structure [38]

Inside a neural network, every neuron focuses on the following tasks [7]:

- The product of each input and the weight of the channel over which it is passed is calculated.
- The weighted sum is calculated by adding the weighted products.
- The neuron's bias value is added to the weighted sum.

- The final result is therefore passed through a function known as the activation function.

As a conclusion, data gives information in the form of inputs to each node. The node multiplies the inputs by random weights, computes them, and then adds a bias. Finally, activation functions are used to specify which neuron should be activated.

(a) **Activation Function**

- An activation function is a mechanism that a neuron uses to introduce characteristics into the network, determine whether the neuron should be activated, and identifies the output of that node given an input or set of inputs.[43].

(b) **Types of activation functions :**

- Non-linear function, Linear function, Binary Step function function [43].

i. **Linear Activation Functions:**

A linear activation function has the formula:

y=mx+c (m in the line equation represents W and c in neural nets represents b, so the equation can be changed to **y=Wx+b**).

It multiplies the inputs ($Xi's$) by the weights ($Wi's$) for each neuron to produce an output proportional to the input. To put it clearly, weighted sum input is proportional to output. [43].

ii. **Binary Step Activation Functions:**

The binary step activation function is also known as the "threshold activation function." It is a very basic function. The gradient (differential) of the binary step activation function is zero, that is a major issue in back-propagation for weight updation. Another issue of step activation functions is that they can only handle binary class problems. (However, with some modifications, we can use it for multi-class problems) [43].

iii. **Non-linear Activation Functions:**

To fire a neuron, most modern neural networks use the non-linear function as their activation function. Because they allow the model to create complex mappings between the network's inputs and outputs, they are necessary for learning and modeling complex data, such as images, video, audio, and non-linear or high-dimensional data sets. Differentials are possible in all non-linear functions.Nonlinear Activation Functions are classified primarily by their range or curves. **range or curves** [43] .

(c) **Loss functions**

definition :

The loss function is a technique for determining "how well your approach models your dataset." If your estimations are completely incorrect, your loss function will return a higher value. If they're pretty good, the number will be lower. Your loss function will let you know whether or not you're improving your model as you tune your algorithm. 'Loss' let us know how much the estimated value differs from the current value. [43].

Loss functions Types

i. **Regression Loss Function:** Regression models are used to estimate a continuous value, such as the price of a room , number of rooms, and size. The regression loss function is the name given to the loss function used in the regression problem.[43].

ii. **Multi-class Classification Loss Functions:** Multi-Class classification loss functions are used in predictive modeling challenges with multiple target parameters per class. It is simply an extension of the binary classification problem. [43].

(d) **Cost function :**

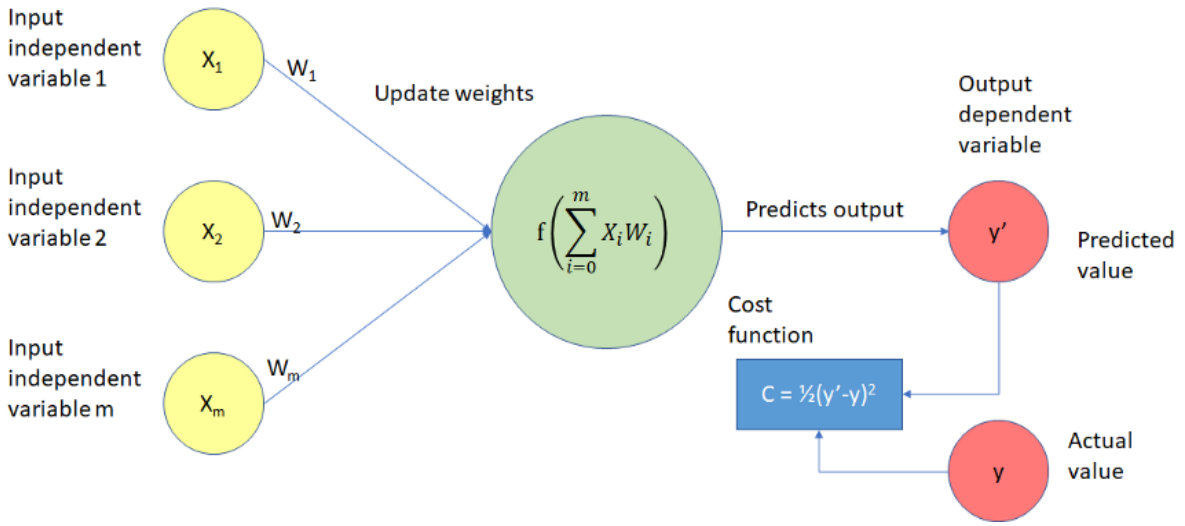

Figure I.8: Cost function calculations[43]

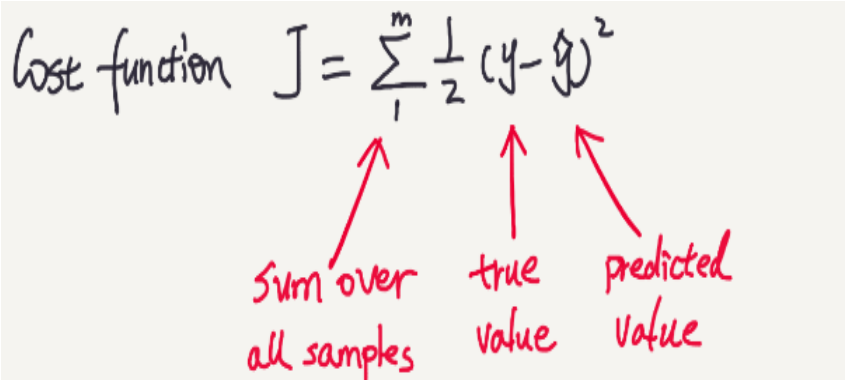

Figure I.9: Cost function formula [43]

For one training example, the loss error is calculated. If we have 'm' instances, the average loss function of the entire training set is referred to as the 'Cost function.' [43].

Cost function (J) = 1/m (Loss error sum for 'm' instances) [43]

The cost function graph against parameters (W and b) has the form of a cup up parabola to a only one minimum value known as the 'local optima.' [43] .

It is a function that evaluates the efficiency of a Trained Model against a set of data. The Cost Function quantifies the difference between predicted and

expected values and presents it as a single real number. Cost Function can be formed in a variety of ways based on problem. The primary objective of the Cost Function is to be either [43]:

- **Minimized** :The returned value is commonly referred to as a cost, loss, or error. The objective is to achieve model parameter values in which the Cost Function returns the smallest possible value [43].
- **Maximized** : The returned value is referred to as a reward. The objective is to achieve model parameter values with the absolute highest returned number. [43].
- Applying the following formula to a given model: $\hat{y} = \text{wx}$, while :
  $\hat{y}$: Estimatred value .
  w : A data vector that is used for prediction or training.
  x : weights [43].

5.5 Process of deep Learning Algorithms

Deep learning algorithms rely on artificial neural networks (ANNs), which simulate how the brain processes information. During whole training phase, algorithms use unknown elements in the input distribution to extract features, organize objects, and uncover important data patterns. This occurs at multiple layers, with methods used to develop the models, much like training robots for self-learning. Deep learning techniques employ a variety of methods. While no network is perfect, some algorithms are better suited to certain tasks than others. To choose the best, you must have a thorough understanding of all primary algorithms. [7].

6 Deep Learning Methods

The following section discusses most of the efficient strategies that can be employed to deep learning techniques to minimize training time and improve the model [21].

6.1 Backpropagation:

Backpropagation can be used to calculate the gradient of a function for each iteration of an optimization problem using a gradient-based method. [21].

6.2 Stochastic Gradient Descent:

The use of the convex function in gradient descent methods guarantees that the optimum solution minimum is found without becoming trapped in a local minimum. It may arrive at the optimum value in various ways and paths based on the function values and learning rate or step size.[21].

6.3 Learning Rate Decay :

Changing the learning rate improves the efficiency and shortens the training time of stochastic gradient descent methods. The most common method is to decrease the learning rate, that allows us to make large changes at the start and then gradually reduce the learning rate during the training process. This allows the weights fine tuning in the later stages. [21].

6.4 Dropout:

The dropout method can be used to address the overfitting problem in deep neural networks. Throughout the training, this approach is used by randomly dropping units and their connections. Dropout is a powerful regularization strategy to minimize overfitting and improving generalization error. Dropout improves supervised learning performance in computer vision, computational biology, document classification, and speech recognition. [21].

6.5 Max Pooling:

A filter is predefined in max pooling, and it is then adapted throughout the nonoverlapping subregions of the input, returning the maximum of the attributes encapsulated

in the window as the output. Max pooling can minimize dimensionality as well as the computational cost of learning multiple parameters. [21].

6.6 Batch Normalization:

Batch normalization minimises covariate shift, which speeds up deep neural networks. Whenever the weights are refreshed throughout the training, it normalizes the inputs to a layer for every mini-batch. Normalization improves learning by reducing training epochs. Normalizing the results of the previous activation layer can enhance the stability of a neural network. [21].

6.7 Skip-gram:

Skip-gram can be used to model word embedding methods. When two vocabulary words share a similar context in the skip-gram model, they are similar. For example, "cats are mammals" and "dogs are mammals" are both significant phrases with the same meaning of "are mammals." Skip-gram can be implemented by taking a context window with n terms, training the neural network by skipping one of the terms, and then using the model to estimate the skipped term. [21] .

6.8 Transfer Learning:

A model trained on a different task is used on another similar task in transfer learning. The knowledge gained while solving a specific issue can be converted to another network that will be trained on a similar problem. This allows for faster progress and improved performance while addressing the second issue. [21].

7 Deep Learning Frameworks

A deep learning framework aids in faster modeling a network without delving into the underlying techniques.Some deep learning frameworks are discussed below and are summarized in Table 2.2 [47].

- **Sonnet ,Keras ,TensorFlow**
- **MXNet ,Swift for TensorFlow**
- **Caffe,Gluon ,PyTorch**
- **Deeplearning4j ,ONNX ,Chainer**

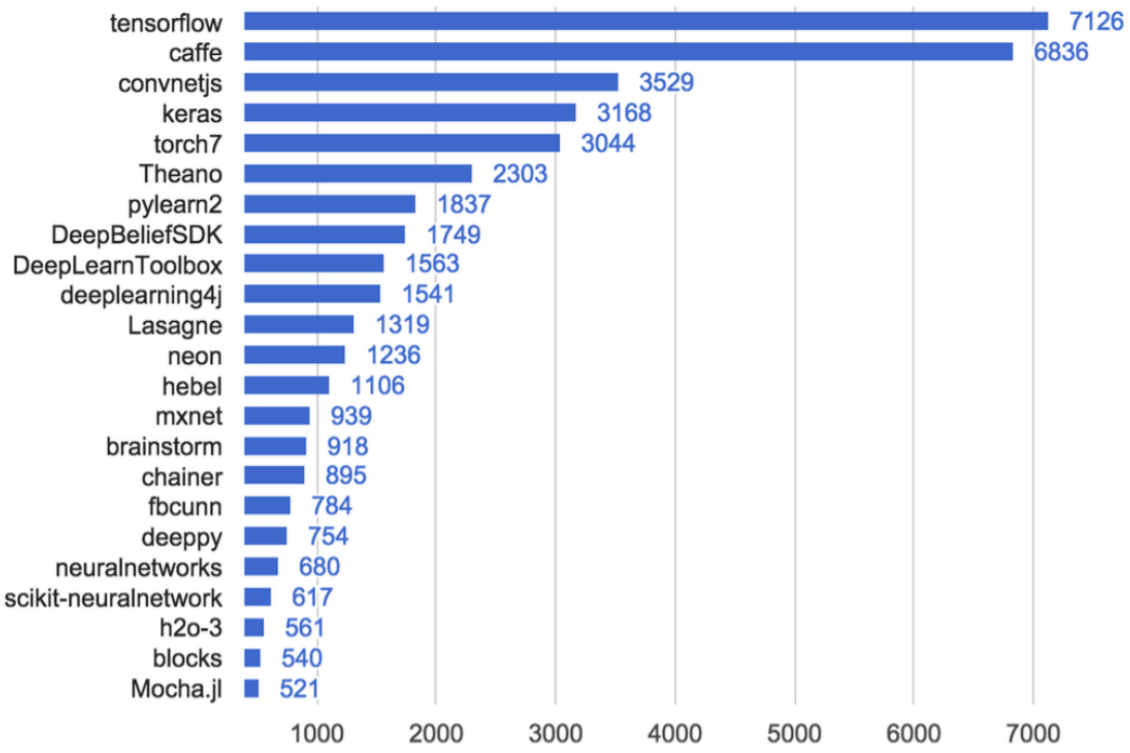

Figure I.10: Deep Learning Frameworks [23]

8 Types of Deep Learning Algorithms

Deep learning methods can work with almost all types of data and necessitate massive amounts of computing power and data to overcome complex problems. Let us now dig

frameworks	official launch year	Language	Use of CUDA	Transfer Learning
TensorFlow	2015	C++ and Python	Yes	Yes
Keras	2015	Python	Yes	Yes
PyTorch	2016	Python and C	Yes	Yes
Caffe	2013	C++	Yes	Yes
Deep learning 4j	2014	C++ and Java	Yes	Yes

Table I.1: Most used deep learning frameworks[21]

deeper into the deep learning algorithms. [7].

- ConvNets or Convolutional Neural Networks or (CNNs) [7]
- Long Short Term Memory Networksor (LSTMs) [7]
- Recurrent Neural Networks or (RNNs) [7]
- Generative Adversarial Networks or (GANs) [7]
- Radial Basis Function Networks or (RBFNs) [7]
- Multilayer Perceptrons or (MLPs) [7]
- Self Organizing Maps or (SOMs) [7]
- Deep Belief Networks or (DBNs) [7]
- Restricted Boltzmann or Machines(RBMs) [7]
- Autoencoders [7]

8.1 Convolutional Neural Networks (CNN's)

CNNs, also defined as ConvNets, are multi-layer neural networks that are primarily used for image processing and object detection. Yann LeCun created the first CNN, known as Le-Net, in 1988. It was used to recognize characters such as ZIP codes and numbers. [7].

CNNs are employed to detect anomalies, recognize satellite pictures, process medical images, forecast time series. [7].

The process of CNNs :

The ConvNets mechanism is built on various layer that work together to process and extract features from data, are listed as follows [7]:

- **The Kernel(Convolution Layer)**

  CNN has a convolution layer with many filters that accomplishes the convolutional transformations. [7].

- **ReLU (Rectified Linear Unit)**

  CNNs have a ReLU layer that carry out specific processes on elements. The result is a feature map that has been rectified. [7].

- **Pooling Layer**

  - The corrected feature map is then fed into a pooling layer. Pooling is a down sampling process that minimises the feature map's measurements. [7].
  - The pooling layer then flattens the resulting two-dimensional arrays from the pooled feature map to create a single, long, continuous linear vector. [7].

  Pooling is classified into two types, that are as follows:

  - **Max Pooling** :Max Pooling essentially gives the high value within the Kernel-covered picture. [42].
  - **Average Pooling** : The Kernel provides and returns the approximate value within the covered image via Average Pooling. [42].

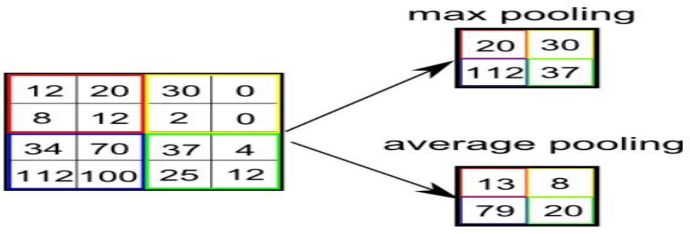

Figure I.11: Pooling Types [42]

- **Fully Connected Layer**

  When the flattened matrix from the pooling layer is provided as input, a fully connected layer forms, that classifies and identifies the images. [7].

Here's an example of a CNN-processed image.

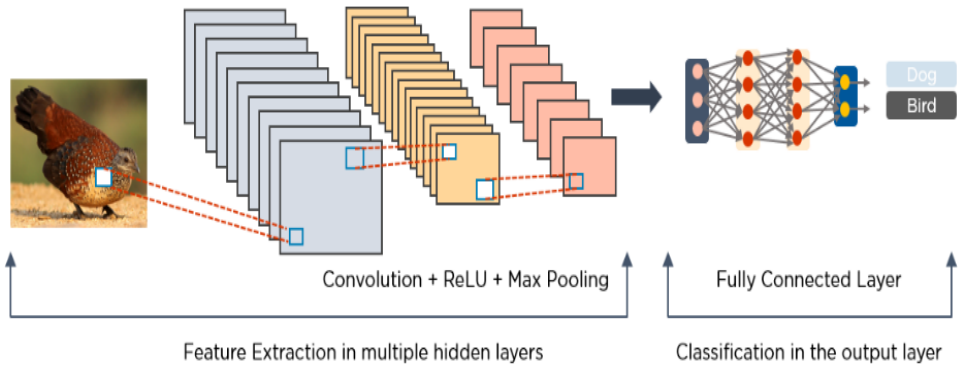

Figure I.12: Image processing via CNN [7]

Architecture Name	Main contribution	Error Rate	Reference
LeNet	First popular CNN architecture	[dist]MNIST:0.8 MNIST:0.95	(LeCun et al.1995)
AlexNet	Deeper and wider than the LeNet , Uses Relu , dropout and overlap Pooling , GPUs NVIDIA GTX 580	ImageNet :16.4	(Krizhevsky et al. 2012)
ZfNet	Visualization of intermediate layers	ImageNet: 11.7	(Zeiler and Fergus 2013)
VGG	Homogenous topology, Uses small size kernels	ImageNet:7.3	Simonyan and Zisseman 2015
GoogLeNet	Introduced block concept, Split transform and merge idea	ImageNet:6.7	Szegedy et al.2015
Inception-V3	Handles the problem of a representational bottleneck,replace large size filters with small filters	ImageNet:3.5 , Multi-Crop:3.58,Single-Crop:5.6	Szegedy et al.2016b
Highway Networks	Introduced an idea of Multi-path	CIFAR-10: 7.76	Srivastava et al.2015a

Table I.2: Image processing via CNN [2]

CNNs Architectures :

- **LeNet-5 Architecture (1998):**

  One of the most basic architectures is LeNet-5. It has two convolutional layers and three fully connected layers (hence the "5" neural network names are frequently obtained from the number of convolutional and fully connected layers. The average-pooling layer, as we now recognize it, was named a sub-sampling layer, and it had trainable weights (that is not the current practice when designing CNNs). This architecture has approximately 60,000 variables. [2].

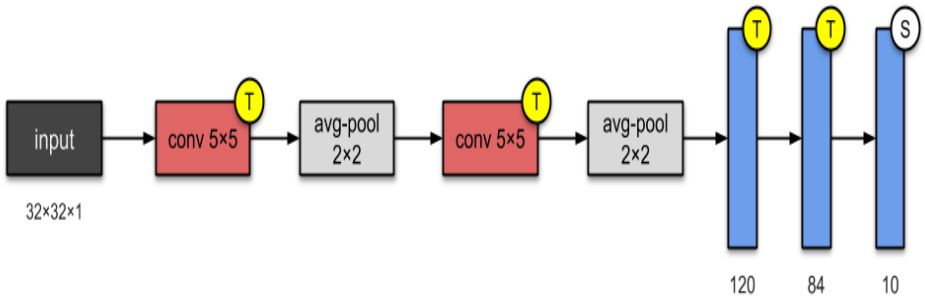

Figure I.13: LeNet-5 Architecture[2]

- **AlexNet(2012):**

  Alexnet is composed of five convolution layers that begin with an 11x11 kernel. It was the first to use max-pooling layers, ReLu activation functions, and dropout for the three massive linear layers. its system was used for classification tasks, with 1000 possible classes, which was insane at the time. It is now possible to implement it in 35 lines of PyTorch code [2]. It was the first convolutional model to be successfully trained on Imagenet, and it was quiet hard to implement such a model in CUDA at the time. Dropout is widely used in massive linear transformations to prevent overfitting. Prior to the release of auto-differentiation in 2015-2016, it took months to implement backpropagation on the GPU [2].

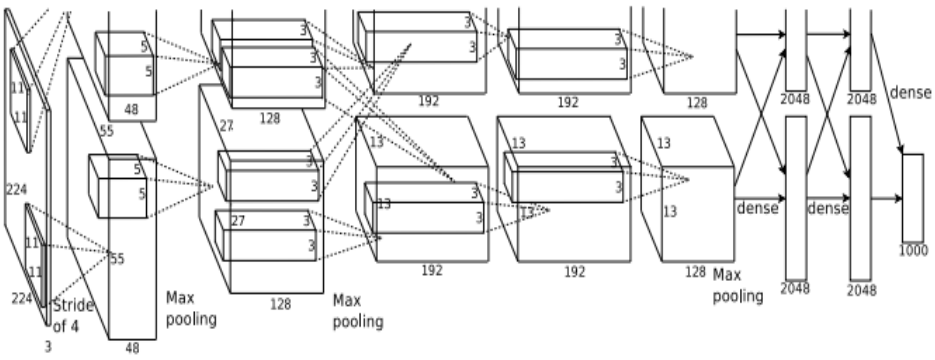

Figure I.14: AlexNet Architecture[2]

- **ZFNet(2013):**

  The champion of the ILSVRC 2013 was also a CNN, which became known as ZFNet. It obtained a top-5 error rate of 14.8%, which is now half of the previously discussed non-neural error rate. It was mostly accomplished by tinkering with AlexNet's hyperparameters whereas preserving the same architecture with additional Deep Learning components as discussed earlier in this dissertation. [37].

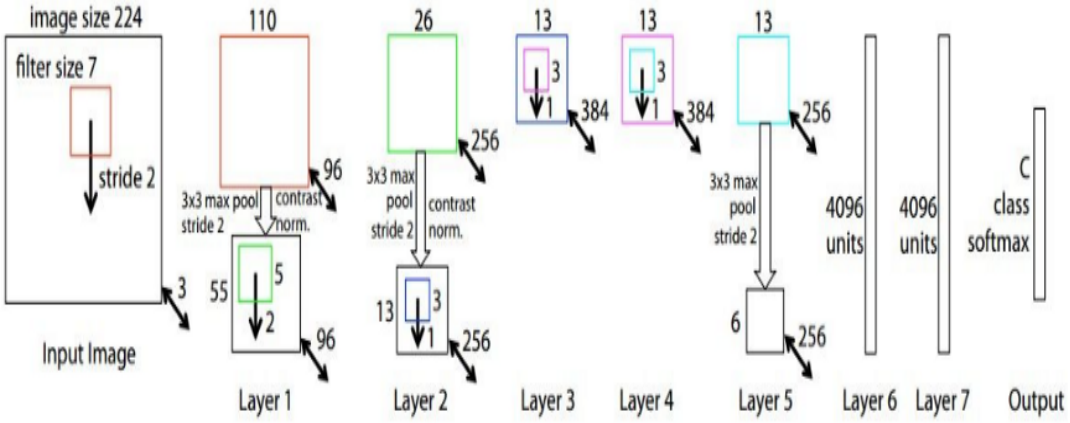

Figure I.15: ZFNet Architectur[37]

- **VGG (2014):**

  The structure was trained on 224 224 RGB images, and its core idea is that a stack of three 33 conv layers are equivalent to a single 7 layer. And possibly even better! Because they use three non-linear activations in between (rather than one), the function becomes more discriminative [2]. Furthermore, by using this design, the

number of variables is reduced [2]. It can be thought of as a regularisation of the 7*7 conv. filters, restricting them to a 3x3 non-linear decomposition. Lastly, it was with the 1st structure that normalization became a major problem. Nonetheless, pretrained VGGs are still used in Generative Adversarial Networks for feature matching loss, as well as neural style transfer and feature visualizations [2].

VGG-16:

In ImageNet, a dataset of over 14 million images classified into 1000 classes, the model has performed 92.7% top-5 test accuracy. Karen Simonyan and Andrew Zisserman of Oxford University's Visual Geometry Group Lab proposed it in 2014.[3].

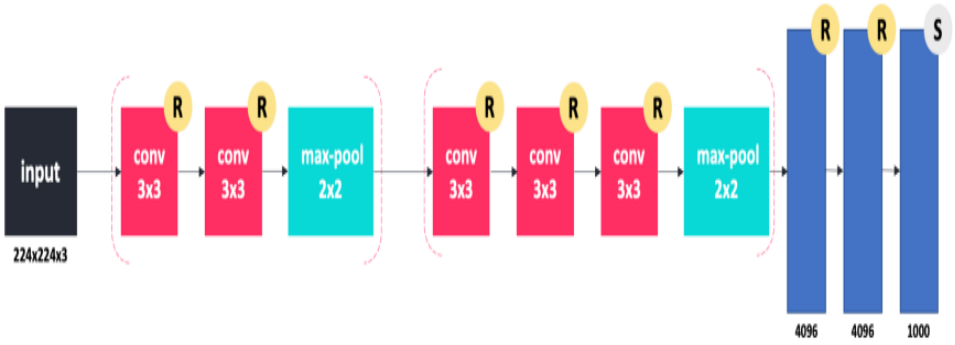

Figure I.16: VGG-16 architecture [3]

VGG-19:

VGG-19 is a 19-layer deep convolutional neural network that has the ability to classify pictures into 1000 object categories such as a keyboard, mouse, and many animals. With a precision of 92%, the model has been trained on over a million pictures from the Imagenet dataset. [3].

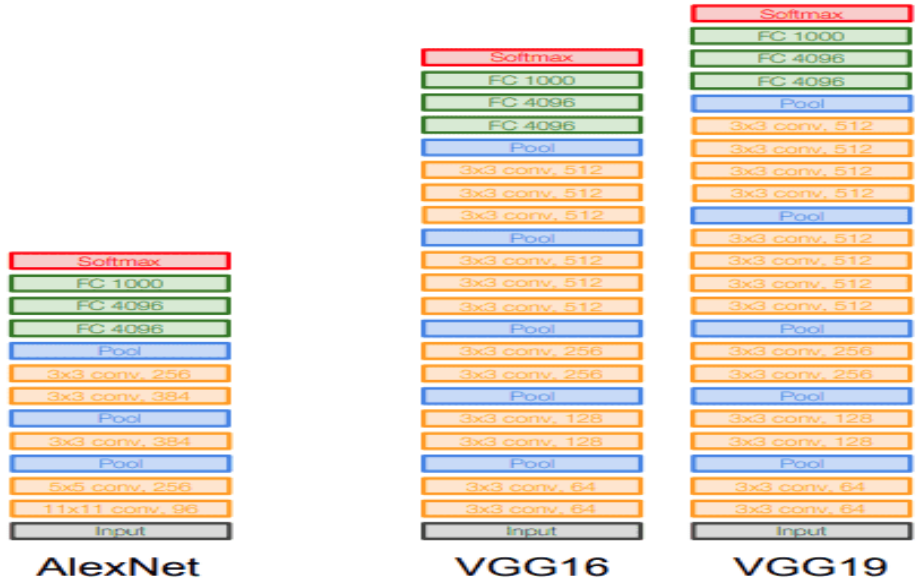

Figure I.17: VGG-16 , VGG-19 architectures comparison to AlexNet architecture[3]

- **VGGNet (2014) :**

  The community has dubbed the runner-up at the ILSVRC 2014 competition VGGNet, which was created by Simonyan and Zisserman. VGGNet has 16 convolutional layers and is appealing due to its very unified structure. Similar to AlexNet, only 3x3 convolutions are used, but there are many filters. It is currently the most popular option for extracting features from images [37].

Figure I.18: VGGNet architecture[37]

- **InceptionNet/GoogleNet (2014)**

  This architecture has increased the network's depth and width whereas preserving calculations fixed [2]. The inspiration came from the human visual system, which processes data at various scales before aggregating it locally. How can this be accomplished without causing a memory explosion? With 11 convolutions, the

response is! The ultimate objective is to reduce dimension by minimizing the output channels of each convolution block [2].

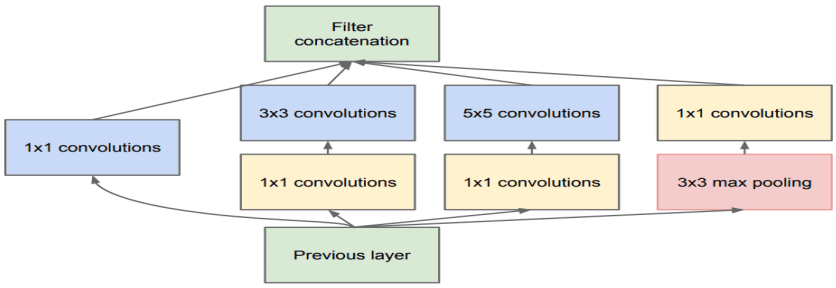

Figure I.19: InceptionNet using inception module[2]

- **Inception-v1 (2014):**

  This 22-layer structure with 5M variables is known as the Inception-v1, and it is widely utilised. The structure of an Inception module is the result of research into estimating sparse structures. Each module introduces three concepts.[2].

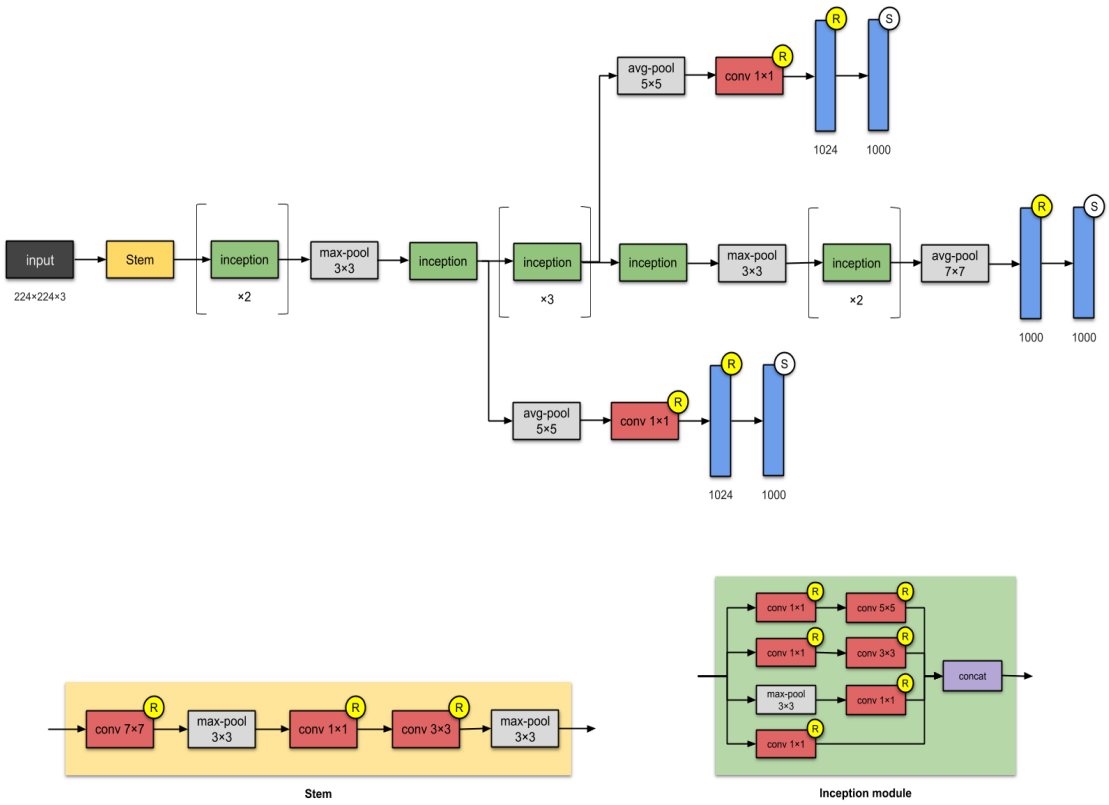

Figure I.20: Inception V1 Architecture[2]

- **Inception V2, V3 (2015)**

  The Inception model has been improved by the following guidelines below. [2]:

  - Factorize 5x5 and 7x7 convolutions (in InceptionV3) to two and three 3x3 sequential convolutions, respectively. This increases the computational speed. This is the same main idea as VGG.
  - Convolutions that can be separated spatially were used. Simply put, a 3x3 kernel is split into two smaller ones: a 1x3 kernel and a 3x1 kernel, which are then adapted sequentially.
  - The scope of the inception modules expanded (more feature maps).
  - They attempted to disperse the computational budget in a balanced manner across the network's depth and width.
  - They incorporated batch normalization.

  Later versions of the inception model are **InceptionV4** and **Inception-Resnet**.

- **ResNet: Deep Residual Learning for Image Recognition (2015)**

  Two tricks were used to address all of the previously mentioned issues, such as vanishing gradients.[2]:

  1. short skip connections and batch normalization

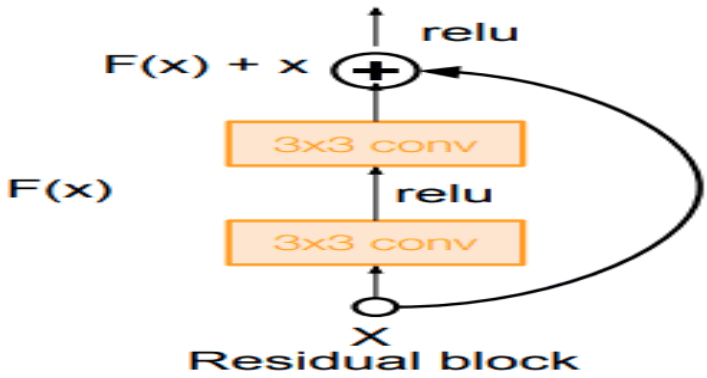

Figure I.21: Skip Connections[2]

The develloppers created deeper architectures ranging from 18 (Resnet-18) to 150

(Resnet-150) layers using that simple but effective block. [2].

FThey used 1x1 convs for the deepest models, as shown on the right:

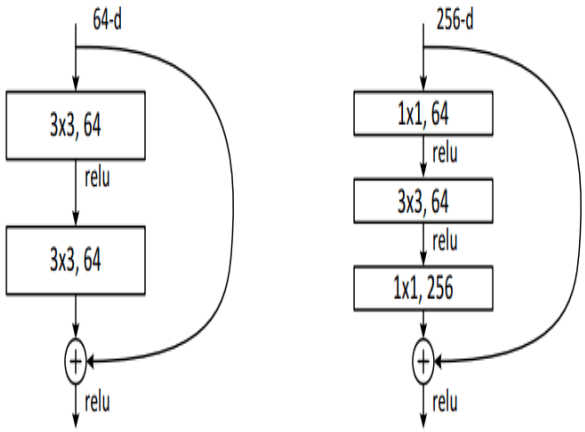

Figure I.22: Skip Connections for the deepest models[2]

- **ResNet-50 (2015):**

  ResNet was an early adopter of batch normalisation (the batch norm paper authored by Ioffe and Szegedy was submitted to ICML in 2015). ResNet-50 with 26M parameters is shown above.

  The basic building block for ResNets are the conv and identity blocks [2].

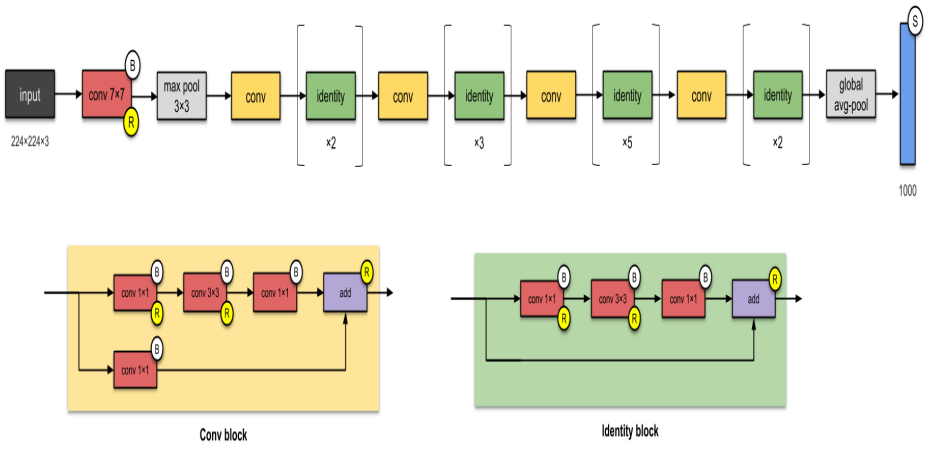

Figure I.23: ResNet-50 Architecture[2]

- **Xception (2016):**

  Xception is an Inception adaptation in which the Inception modules have been changed with depth - wise separable convolutions. It also has approximately the same number of characteristics as Inception-v1 (23M) [2].

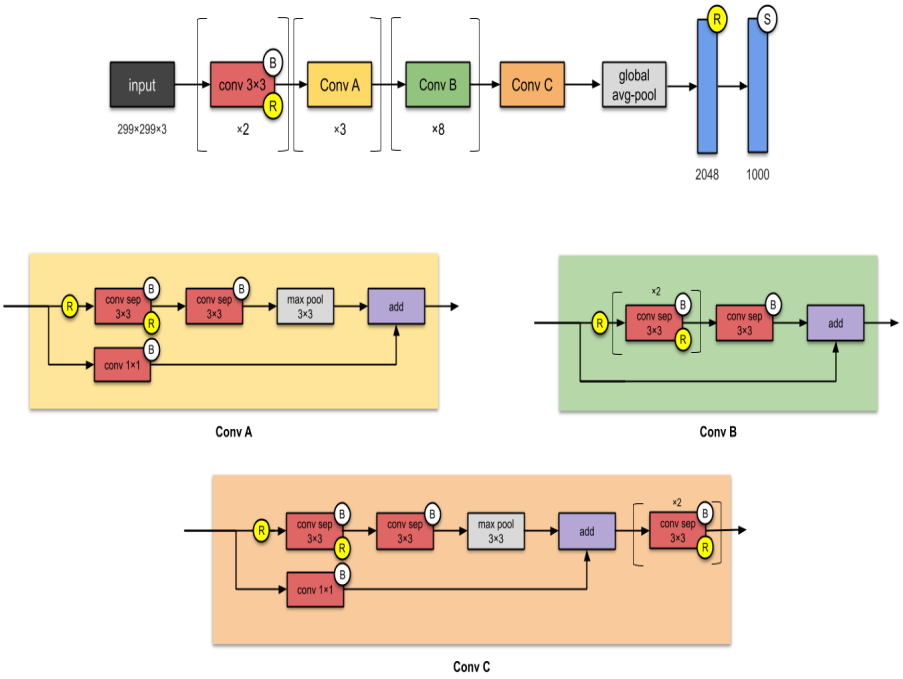

Figure I.24: Xception (2016) Architecture[2]

- **Squeeze Net :**

  It has 18 deep layers and has the ability to classify images into 1000 different object categories, such as keyboard, mouse, pencil, and various animals. SqueezeNet can be three times faster and 500 times tinier than AlexNet for the same precision. [2].

- **Shuffile Net :**

  It is a highly effective CNN architecture with 173 deep layers smartphone devices with 10–150 MFLOPs of computing power. On Image Net classification, it achieves lower top-1 error (absolute 7.8%) than the Mobile Net mechanism.[2].

- **ENet :**

  The ability to perform pixel-wise semantic segmentation in real-time is provided

by an efficient neural network. ENet is up to 18x faster than existing models, necessitates 75x fewer FLOPs, has 79x fewer parameters, and provides similar or better accuracy. In semantic segmentation, Enet is the fastest model. [2].

- **DenseNet: Densely Connected Convolutional Networks (2017)**

  A DenseNet is a category of convolutional neural network that uses dense connections among layers via Dense Blocks, which connect all layers (with matching feature-map sizes) straight. To maintain the feed-forward nature, evey layer receives additional inputs from all previous layers and sends its own feature-maps to all subsequent layers. [2].

  Thereby, the central idea is feature reuse, which results in extremely compact models. As a result, because there are no repetitive feature-maps, it requires fewer parameters than other CNNs. [2].

  The following is an example:

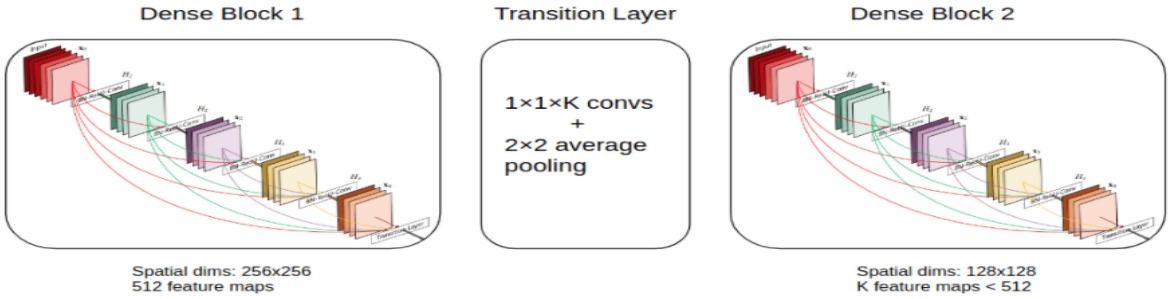

Figure I.25: Densenet Architecture[2]

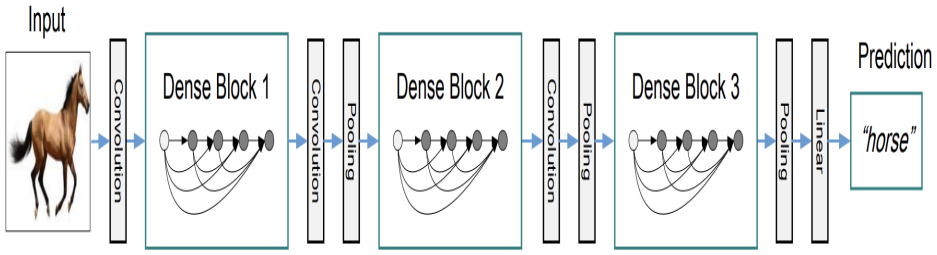

Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

Figure I.26: A Deep Densenet with three dense blocks[2]

- **Big Transfer (BiT):**

  Although many ResNet variants have been proposed, the most recent and well-known is BiT. Big Transfer (BiT) is a scalable ResNet-based image pre-training model [2].

  The selection of normalization layers makes the most significant contribution to the structure. To that end, the authors substituted group normalization (GN) and weight standardization for batch normalization (BN) (WS)[2].

- **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (2019) :**

  Engineering and scale are essential to EfficientNet. It demonstrates that by carefully designing your architecture, you can achieve excellent results with reasonable parameters.

  ResNet-152 is 7.6x smaller and 5.7x faster than EfficientNet-B1.

  The baseline structure, EfficientNet-B0, was discovered using neural architecture search to optimize both accuracy and FLOPS. [2].

- **Noisy Student (2020)** :

  A semi-supervised iterative method was used. It significantly improved Efficient-performance Net's with 300M unlabeled images. The training scheme was dubbed "Noisy Student Training" by the author. It is made up of two neural networks known as the teacher and the student. [2],[44]:

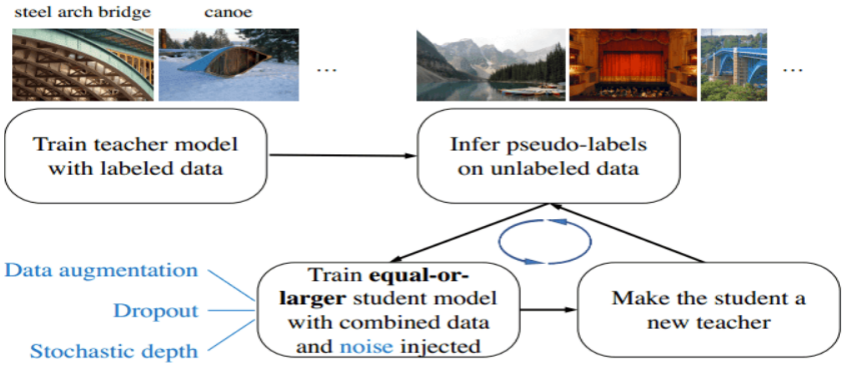

Figure I.27: Self-training with Noisy Student improves ImageNet classification[44]

- **Meta Pseudo-Labels (2021)** :

  a semi-supervised learning method that achieves a new top-1 accuracy of 90.2% on ImageNet, which is 1.6 percent higher than the previous state-of-the-art. Meta Pseudo Labels, like Pseudo Labels, includes a teacher network that generates pseudo labels on unlabeled data in order to teach a student network. In contrast to Pseudo Labels, where the teacher is fixed, the teacher in Meta Pseudo Labels is constantly adapted by feedback from the labeled dataset. As a result, the teacher creates more effective pseudo labels to teach the students. [2].

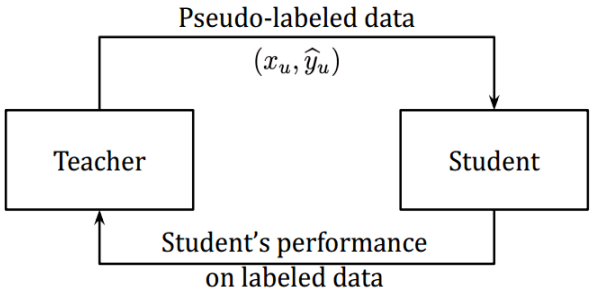

Figure I.28: Meta Pseudo Labels Process[2]

This table summarizes the architectures of the previously mentioned CNNs:

Model name	Number of parameters [Millions]	ImageNet Top 1 Accuracy	Year
AlexNet	60 M	63.3 %	2012
Inception V1	5 M	69.8%	2014
VGG 16	138 M	74.4 %	2014
VGG 19	144 M	74.5 %	2014
Inception V2	11.2 M	74.8 %	2015
ResNet-50	26 M	77.15 %	2015
ResNet-152	60 M	78.57 %	2015
Inception V3	27 M	78.8 %	2015
DenseNet-121	8 M	74.98 %	2016
DenseNet-264	22M	77.85 %	2016
BiT-L (ResNet)	928 M	87.54 %	2019
NoisyStudent EfficientNet-L2	480 M	88.4 %	2020
Meta Pseudo Labels	480 M	90.2 %	2021

Table I.3: Top 10 CNNs Architectures [41]

8.2 Long Short Term Memory Networks (LSTMs)

LSTMs are Recurrent Neural Network (RNN) categories that can learn and remember long-term dependencies. The default behavior is to recall past information for extended periods of time [7].

The process of LSTMs

- in the begining, they ignore irrelevant aspects of the previous state.
- Following that, they selectively update the cell-state values.
- Finally, the output of specific cell state components [7].

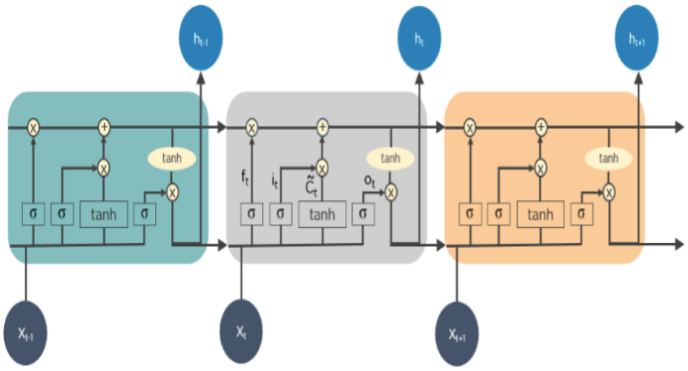

Figure I.29: How a LSTMs architecture operates[7]

8.3 Recurrent Neural Networks (RNNs)

RNNs have links that construct directed cycles, allowing the LSTM outputs to be served as inputs to the current phase [7].

An unfolded RNN appears as follows:

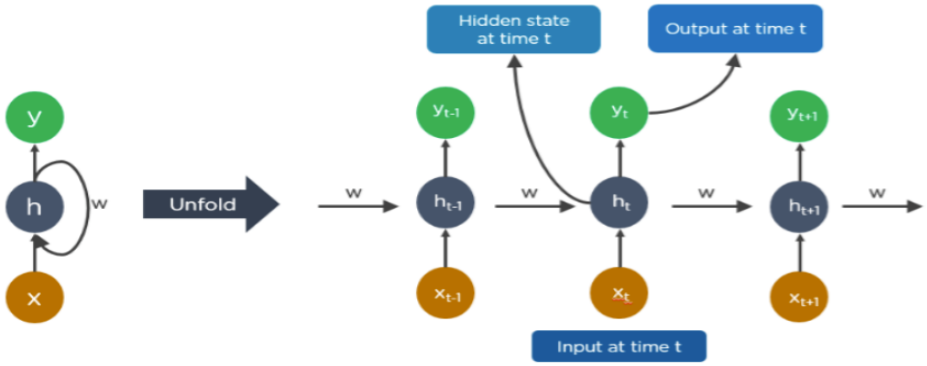

Figure I.30: RNNs Architecture[3]

The process of RNNs

- The time t-1 output feeds into the time t input.
- Similarly, at time t, the output feeds into the input at time t+1.
- RNNs can deal with inputs of any length.
- The computation takes into account historical data, and the model size does not grow in proportion to the input size. [7].

8.4 Generative Adversarial Networks (GANs)

GANs are deep learning generative techniques that generate new data instances that look similar to the training data. GANs are made up of two parts: a generator that learns to produce false data and a discriminator that learns from that false information [7]. GANs aid in the creation of realistic images and cartoon characters, as well as photographs of human faces and the rendering of 3D objects.

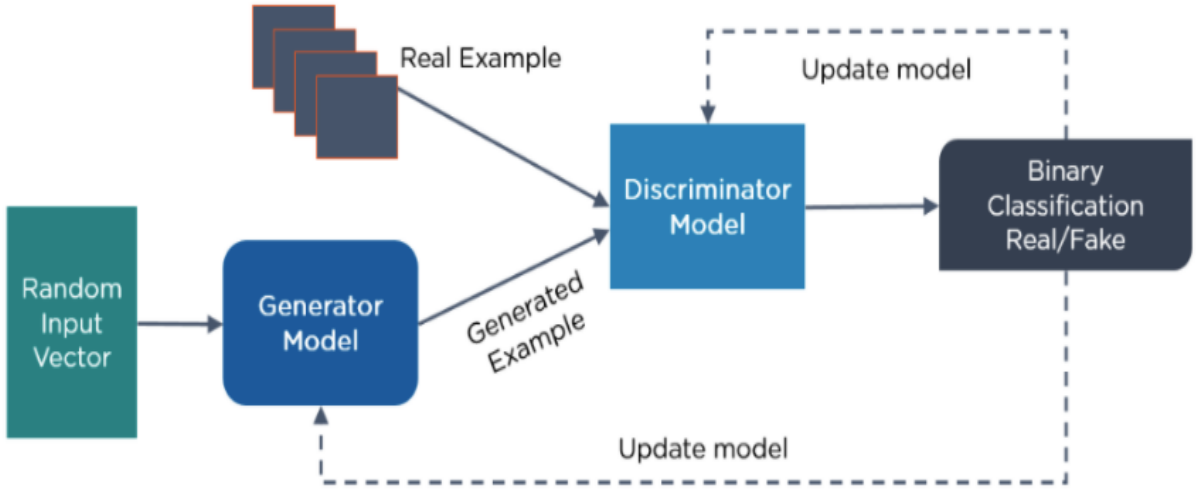

Figure I.31: Diagram of how GANs operate [7]

8.5 Radial Basis Function Networks (RBFNs)

RBFNs are feedforward neural networks with radial basis functions as activation functions. They are commonly used for classification, regression, and time-series prediction and have an input layer, a hidden layer, and an output layer. [7].

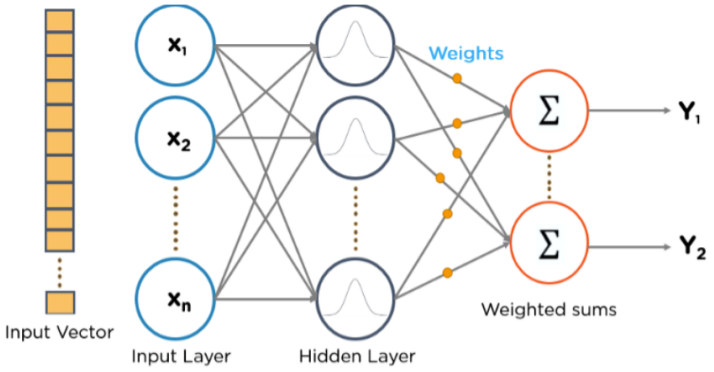

Figure I.32: RBFN architecture [?]

8.6 Multilayer Perceptrons (MLPs)

MLPs are a great way to get started learning about intelligent systems. MLPs are a type of feedforward neural network that has multiple layers of perceptrons with activation functions. MLPs are made up of a fully connected input and output layer. They have

the same number of input and output layers but may have multiple hidden layers and can be used to build speech, image, and machine translation software [7].

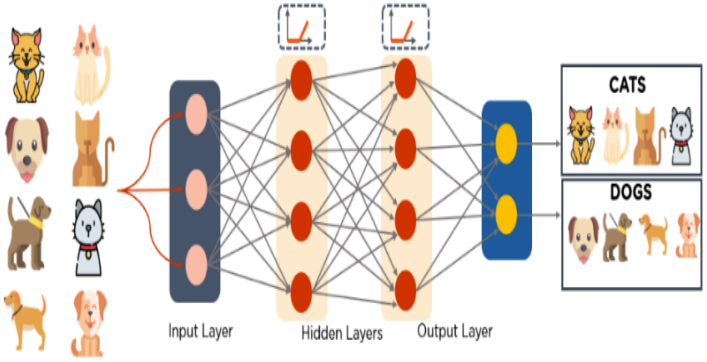

Figure I.33: Example of an MLP[7]

8.7 Self Organizing Maps (SOMs)

Professor Teuvo Kohonen created SOMs, that allow data visualization to reduce data dimensions using self-organizing artificial neural networks [7].

Data visualization attempts to address the issue of humans' inability to visualize high-dimensional data. SOMs are designed to assist users in comprehending this multidimensional data [7].

This is a representation of a multicolored input vector. This data is fed into a SOM, that converts it to 2D RGB values. then, it separates and classifies the various colors [7].

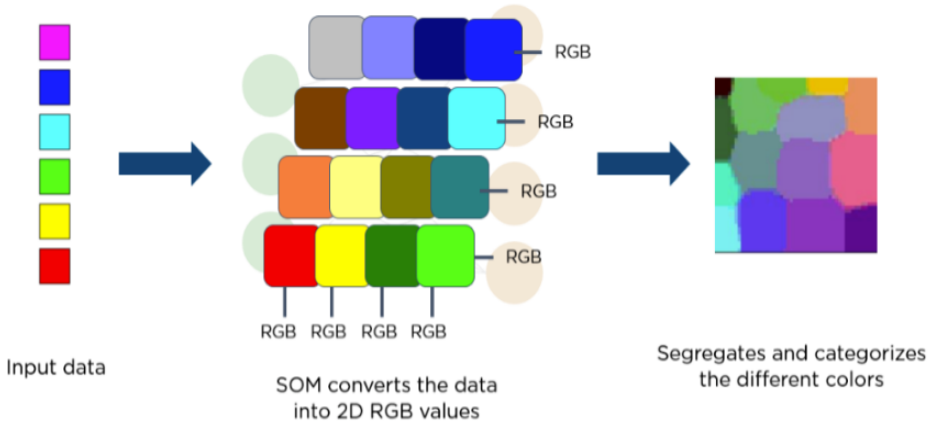

Figure I.34: Example of a SOMs processing[7]

8.8 Deep Belief Networks (DBNs)

DBNs are generative models made up of many layers of stochastic, latent variables. Latent variables have binary values and are often referred to as hidden units. DBNs are a stack of Boltzmann Machines with links between them, and each RBM layer communicates with both the previous and subsequent layers. DBNs are being used to recognize images, videos, and capture motion data. [7].

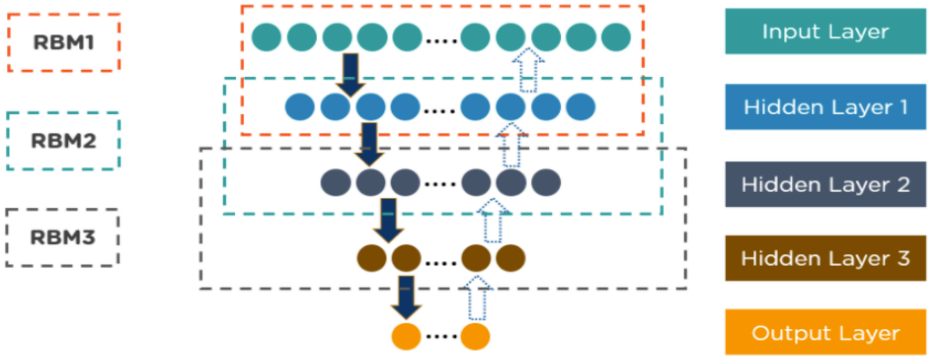

Figure I.35: Example of a DBNs Architecture[7]

8.9 Restricted Boltzmann Machines (RBMs)

RBMs are stochastic neural networks developed by Geoffrey Hinton that can learn from a probability distribution over a set of inputs. Deep learning algorithms such as this one are used for dimensionality reduction, classification, regression, collaborative filtering, feature learning, and topic modeling. RBMs are the building blocks of DBNs and are composed of two layers[7]:

- Visible units
- Hidden units

Every visible unit is linked to every hidden unit. RBMs have no output nodes and a bias unit that is connected to all of the visible and hidden units. [7].

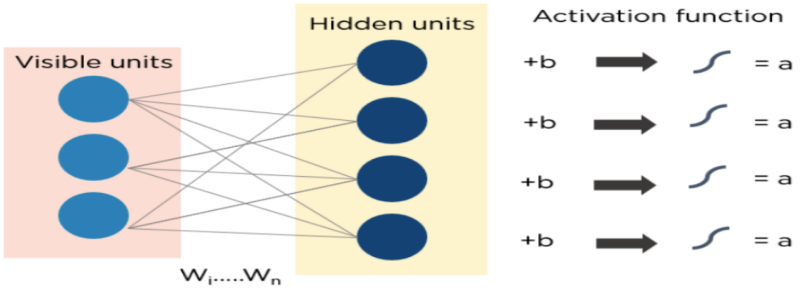

Figure I.36: Example of a RBMs Architecture and process[7]

8.10 Autoencoders

Autoencoders are a category of feedforward neural network with equivalent input and output. In the 1980s, Geoffrey Hinton created autoencoders to solve unsupervised learning issues. They are neural networks that have been trained to replicate data from the input layer to the output layer. Autoencoders are used in a variety of applications, including pharmaceutical discovery, popularity prediction, and image processing. [7].

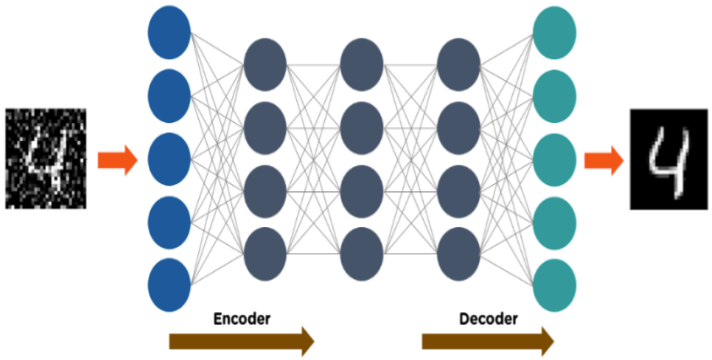

Figure I.37: Example of an Autoencoders Architecture[7]

Conclusion

Deep learning techniques have gained popularity over the last five years along a large variety of businesses.In this part, we've detailled about deep learning, its essential basics, applications, and popular types, as well as some important features of skin cancer.

Chapter II

Requirement analysis Process

Introduction

In this chapter, we'll go over the problem we'll be solving, the issues that it contains, and the strategies we'll utilize to solve it.

1 Problem description :

Our goal in this project is to develop a smart diagnosis system for the diagnosis of skin cancer, starting with the patient management system (clinical data with the necessary crud operations), then moving on to the diagnosis part, which uses a deep learning model for the necessary diagnosis results and takes as inputs dermoscopic images from the international dataset the ISIC 2019, which is a challenge in and of itself due to its difficulty of use for many reasons that we will detail in section (3.4).

2 Required Tools :

2.1 Required Programming Langages :

Python 3.9 :

Python 3.9.0 is the most recent major release of the Python programming language, and it includes numerous new features and optimizations. This is the first Python edition to use the 64-bit installer by default on Windows. The installer now explicitly prohibits installation on Windows 7. Python 3.9 is incompatible with this unsupported Windows version [40].

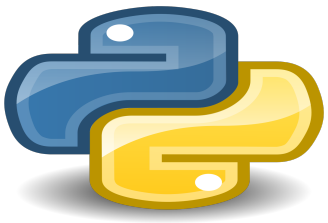

Figure II.1: Python 3.9 Logo [40]

PyQt5 :

PyQt is a group of Python extensions for The Qt Company's Qt application framework that runs on all Qt-supported platforms such as Windows, macOS, Linux, iOS, and Android [32]. PyQt is dual certified under the GNU GPL v3 and the Riverbank Commercial License on all supported platforms [32].

Figure II.2: PyQt5 Logo [32]

Why PyQt5 ?

- PyQt combines the Qt C++ cross-platform application framework and the Python interpreted language [32].
- Qt is much more than a graphical user interface toolkit. It includes network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, and a diverse set of GUI widgets [32].
- Qt classes use a type-safe but loosely coupled signal/slot mechanism for communicating between objects, making it simple to develop reusable software modules [32].
- a graphical user interface designer, is also included in PyQt that can produce both of Python and c++ code, It is also possible to add new Python GUI controls to Qt Designer [32].

SQLite3 :

SQLite is a C-language library that implements a SQL database engine that is small, fast, self-contained, high-reliability, and fully featured. SQLite is the world's most popular database engine. SQLite is developed into all mobile phones and most laptops, and it is included in a plethora of other applications that people use on a daily basis. SQLite file format is stable, cross-platform, and backwards compatible, and the developers promise to keep it that way until 2050 [41].

Figure II.3: SQLite Logo [41]

2.2 Required Softwares

Pycharm:

Figure II.4: Pycharm IDE logo [45]

Qt Designer:

Figure II.5: Qt Designer [20]

DBbrowser:

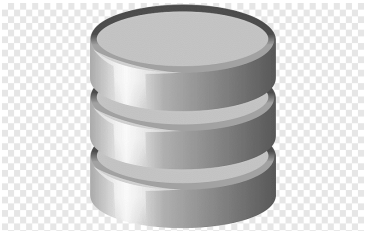

Figure II.6: DB browser [16]

Google Colab:

Google Colab, also known as Colaboratory, is a free cloud service provided by Google that is based on Jupyter Notebook and is intended for machine learning training and research. You can train machine learning models directly in the cloud using this platform. So, other than a browser, we don't need to install anything on our computer [24].

Figure II.7: Google Colab Logo [24]

3 Chosen Deep Learning Model :

A technique for the ISIC 2019 challenge dataset has been implemented utilizing transfer learning and the pre-trained deep neural network ResNet-18. Even with visual imbalances between classes, the suggested technique can accurately classify eight different types of lesions.

The primary goal of transfer learning (TL) is to quickly implement a model. Instead of creating a DNN (dense neural network) from scratch to resolve curent issues , the model will transmit the attributes it has learned from various datasets that have accomplished the same task. This is also referred to as knowledge transfer. **Accuracy**, **Confusion matrix**, **ROC**, **f1 score**, and **precision/recall** were **the performance measures** for the presented approaches. When the quantity of photos in all classes was balanced,

the performance of the suggested technique improved, overcoming the problem of image imbalance between classes.

3.1 Employed Model's Architecture :

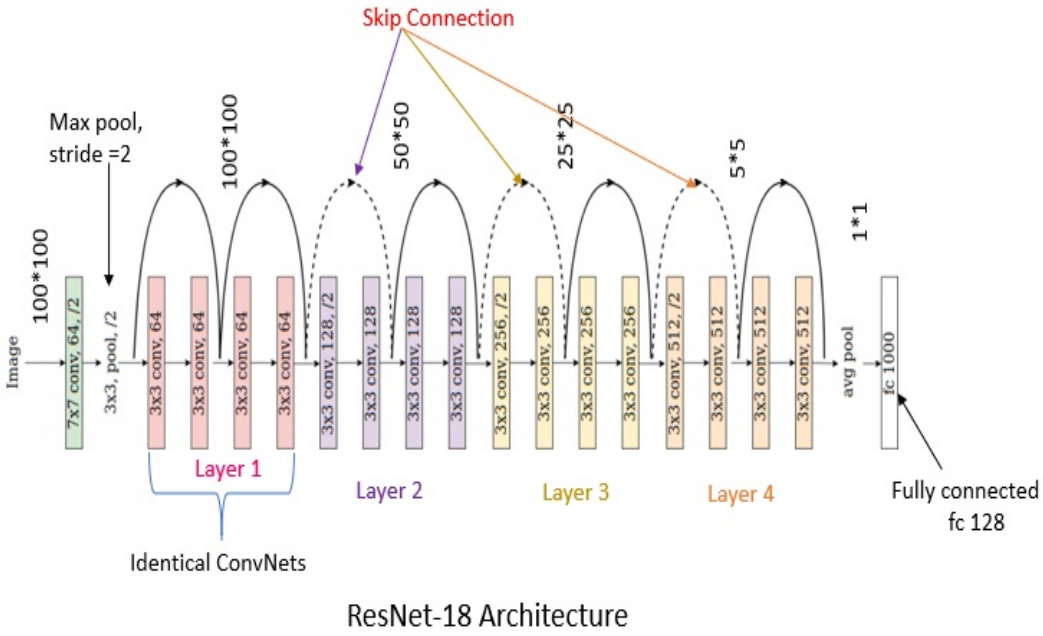

Figure II.8: The structure of ResNet-18

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure II.9: The structure of ResNet-18.

4 Employed Dataset :

4.1 ISIC 2019 :

The ISIC 2019 challenge dataset was used to train, test, and evaluate the proposed model, which includes [14]:

- For training, a diverse dataset of 25 000 images from eight classes was provided. The final test set includes an additional, unidentified class.
- The ISIC 2019 dataset includes images from the HAM10000, the BCN_20000, and the MSK Dataset.
- HAM10000 contains 10000 images with a size of 600450; this dataset was used in the older ISIC 2018 challenge by ViDIR Group, Department of Dermatology, Medical University of Vienna. [10]
- Whereas the BCN_20000 includes 19424 pictures of 1024x1024 pixels in size.[26]
- There is an additional unknown class in the test set that was not present in the training dataset.
- It contains images of the classes melanoma (MEL), melanocytic nevus (NV), basal cell carcinoma (BCC), actinic keratosis (AK), benign keratosis (BKL), dermatofibroma (DF), vascular lesion (VASC) and squamous cell carcinoma (SCC).
- The main traning dataset contains 25 331 dermoscopic images.

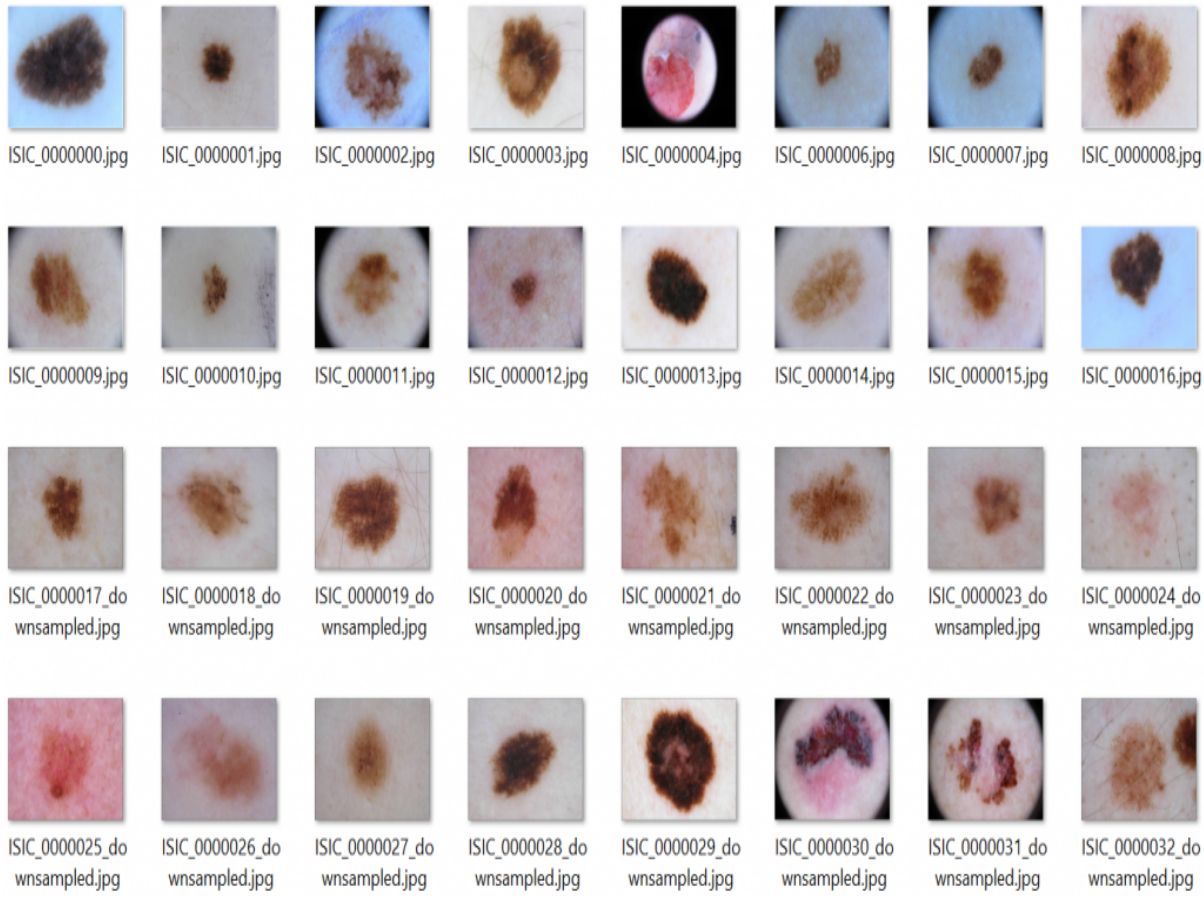

Figure II.10: ISIC 2019 Dataset [28]

4.2 Dataset issues

- The ISIC 2019 came with intense class imbalance issue [1].
- the Multiple resolutions for images; this is due to the variety of sources (HAM1000, 96 BCN 20000 MSK dataset) [1].
- Missing metadata [1] .

4.3 Dataset's solutions for the previous mentionned problems :

- In this work the class imbalance solved using data augmentation of the original dataset images.
- The multiples images resolutions solved using different cropping strategies.

5 Use Case Diagram :

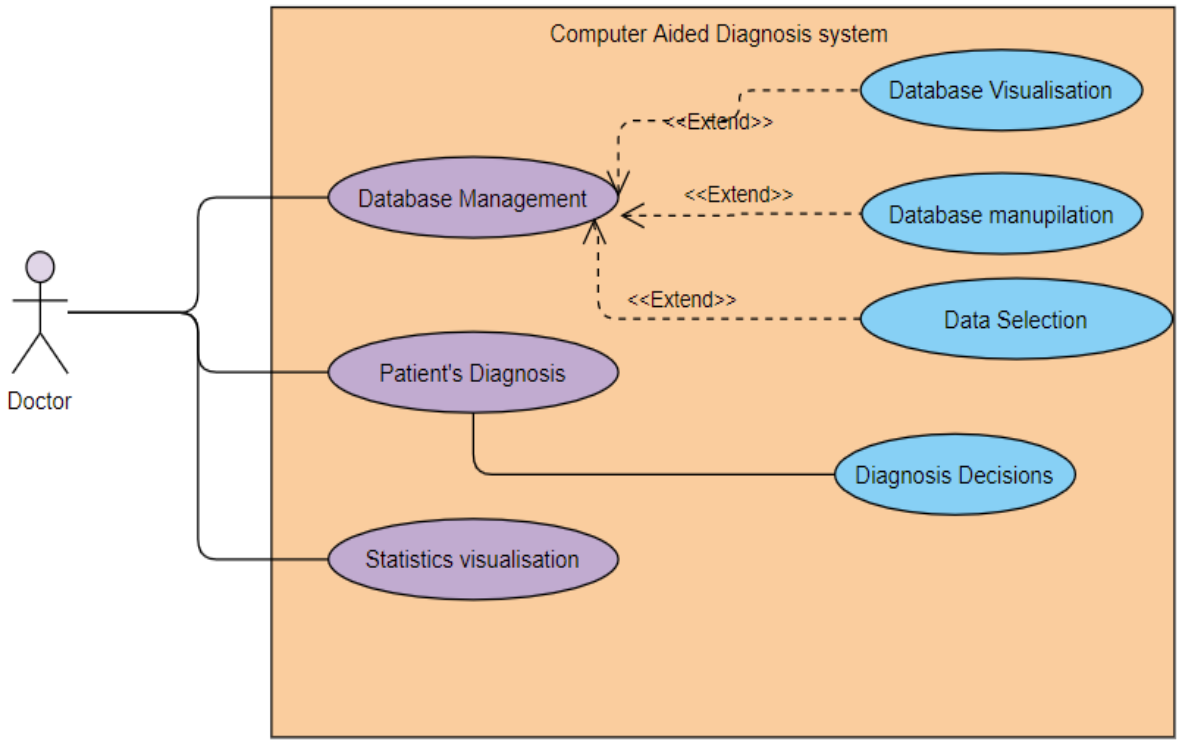

Figure II.11: Use Case Diagram

Conclusion

In this part ,We've seen the problem we're going to solve, the proper offered solutions for each assignment, and the necessary tools while The practical side of these tasks will be covered in chapter4.

Chapter III

Static & Dynamic Design

Introduction

In this chapter, we'll present the main patterns that define our software's primary entities, their functionalities, and their interactions through static and dynamic design, as follows:

1 Static Design

In the static design, we'll show the relational data model, which is composed of four classes: **Patient**,**Dermatologist**, **Lesion**, and **Exam**, as well as their attributes, as shown in the figure below:

1.1 Relational data model :

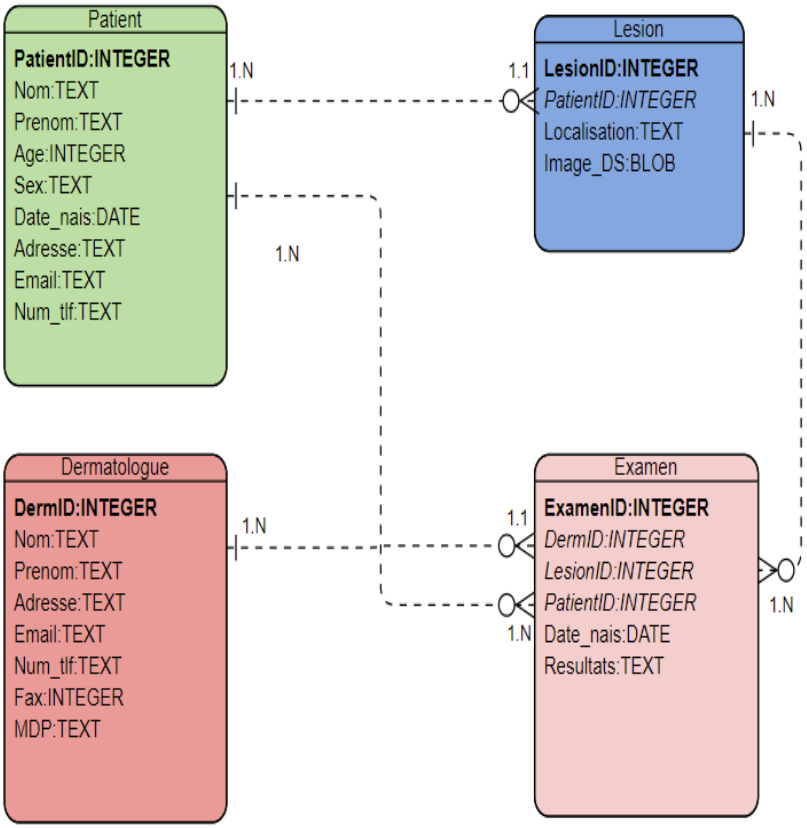

Figure III.1: The relational data model

2 Dynamic Design

In the dynamic design we are going to illustrate the sequence diagram , navigation model , and finally the solution model as follows :

2.1 Sequence diagram :

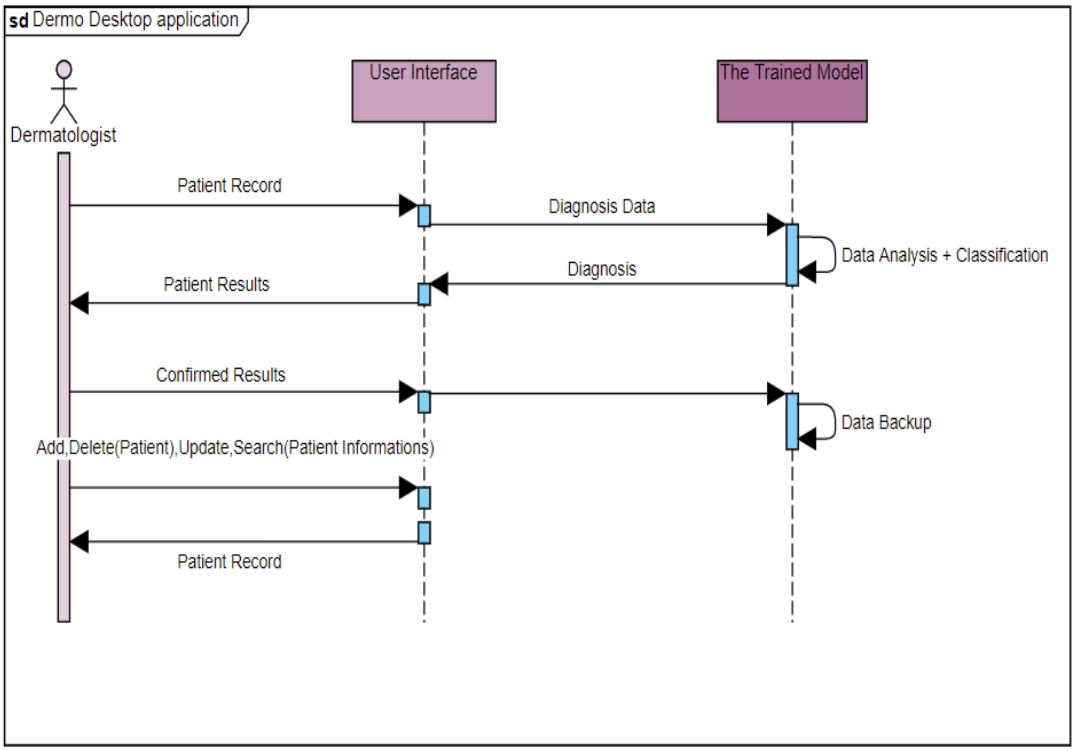

Figure III.2: Sequence Diagram

2.2 Navigation model :

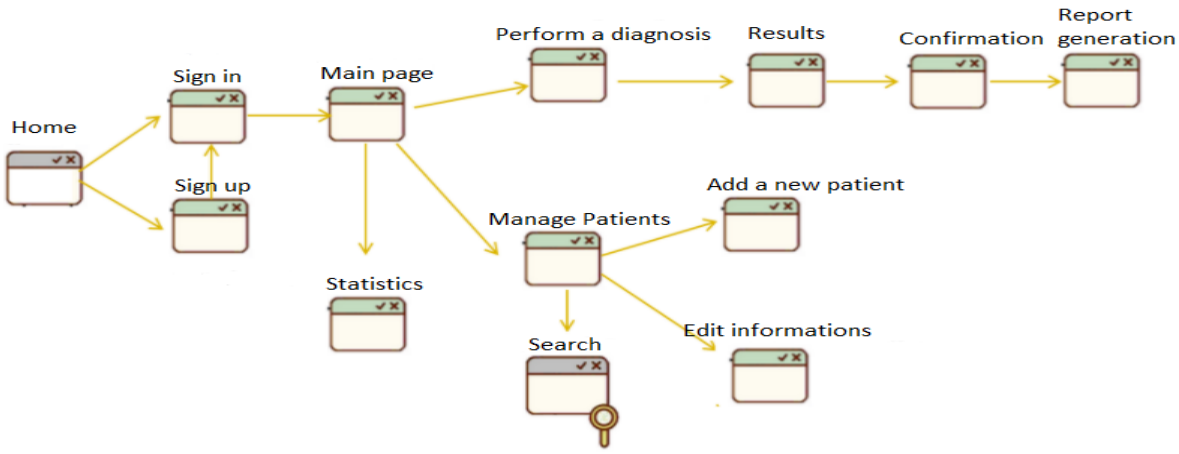

Figure III.3: Navigation Model

2.3 The proposed method :

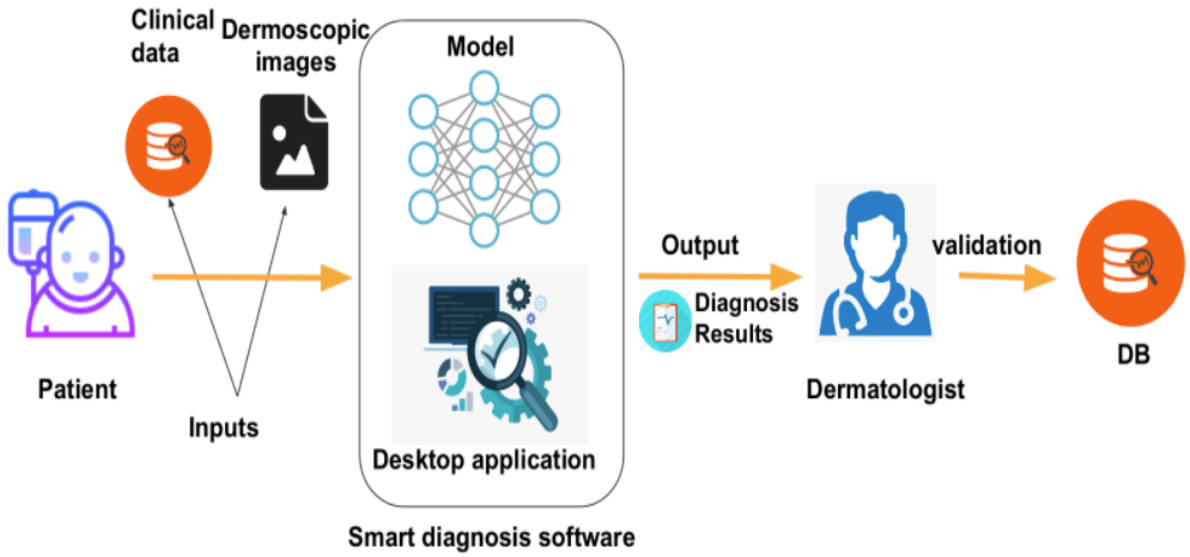

Figure III.4: Synoptic Schema

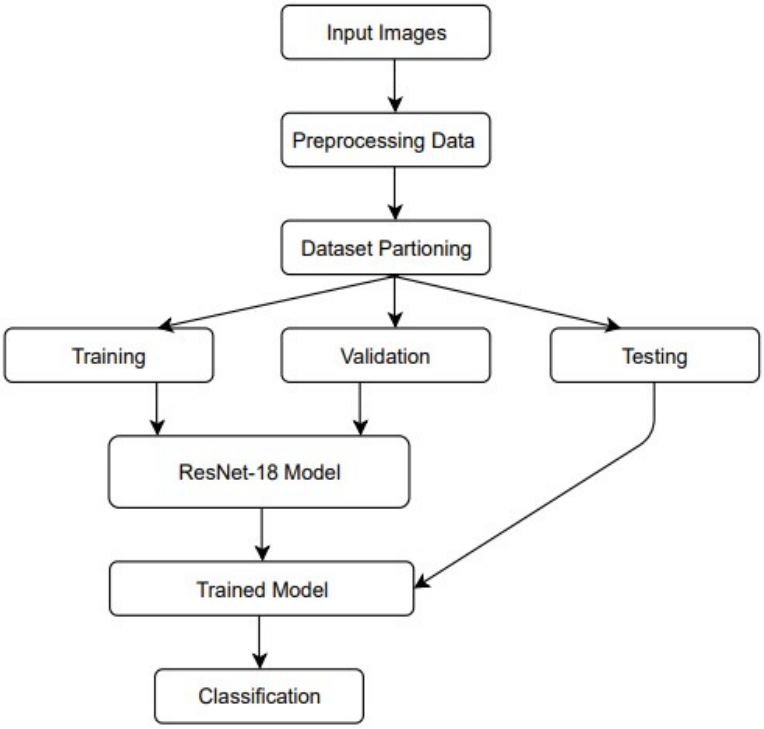

Figure III.5: The proposed DL model flowchart

Conclusion

In this part Through diagrams, we illustrated our system's major functionalities .

Chapter IV

Software illustration & Deep learning model implementation

Introduction

Within this chapter we will illustrate the main user interfaces in our desktop application besides to the implementation of our ResNet-18 deep learning model.

1 Software implementation :

In this part we are going to see the principal user interface it contains three main tabs as follows:

1.1 Patient management system

In this tab we can add , edit , delete , and search patient informations.

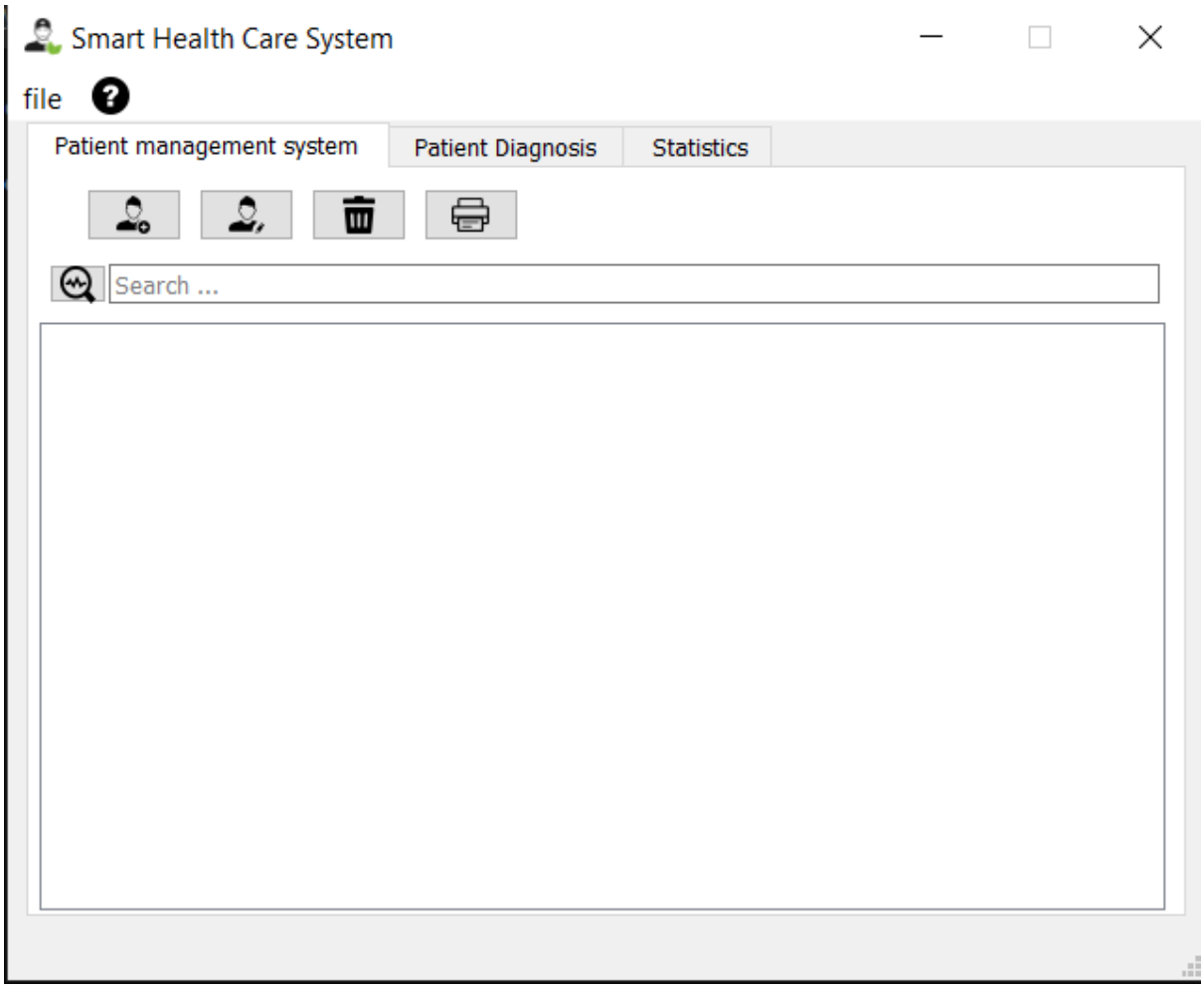

Figure IV.1: Patient Management System UI

1.2 Patient Dignosis

In this tab we can upload a skin image with its characteristics then apply diagnosis on it

.

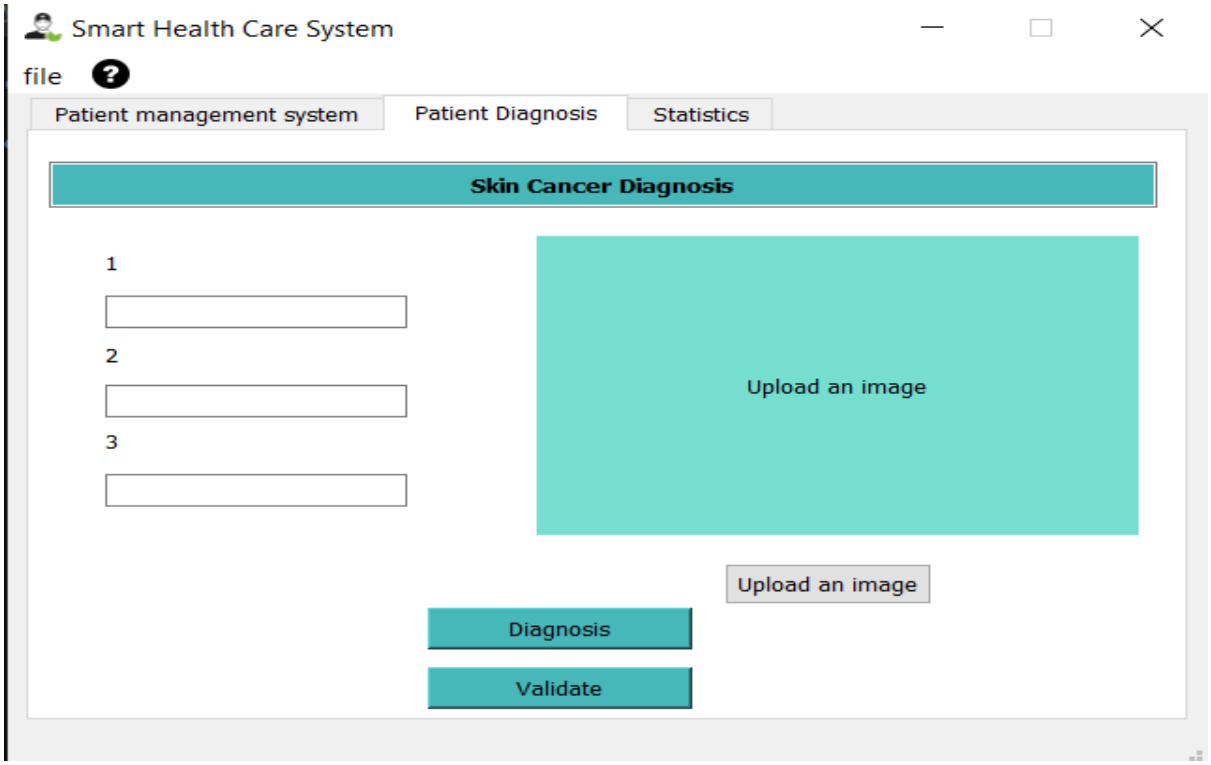

Figure IV.2: Patient Diagnosis

1.3 Statistics

This tab gonna include statistics about existing skin cancer diseases percentage in the application

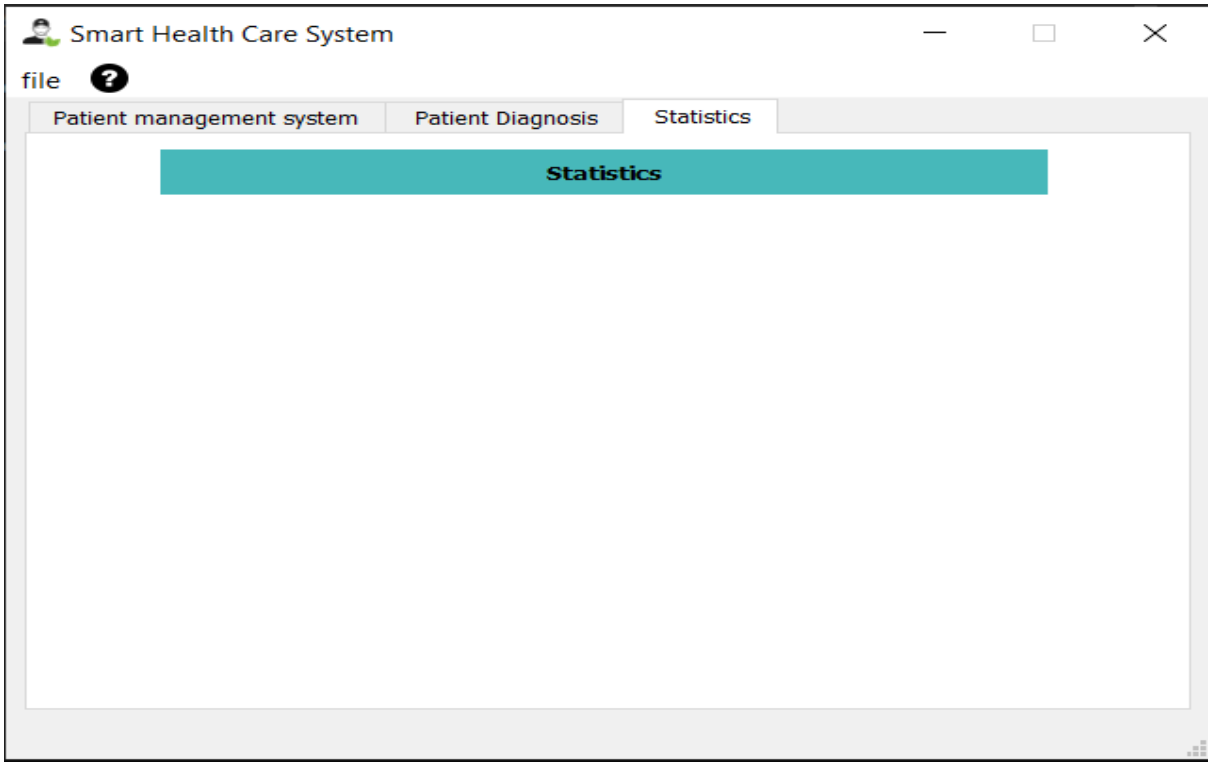

Figure IV.3: Statistics

2 ResNet-18 implementation :

In this section we have two main parts to implement, first one **Data preparation** second one **Data training and validation**.

2.1 Data preparation(Data normalization, Data augmentations, Data balancing, Use of metadata, Image Normalization):

Data Preparation

Preprocessing is used for all ISIC-2019 input images to achieve higher consistency in classification results and enhanced features. For this reason, the CNN approach necessitates a massive amount of repetitive training, as well as a large-scale image dataset to avoid over-fitting.

Image Resizing

The original ISIC dataset includes images in (1022 pixels × 767 pixels) dimensions. The dataset has been rescaled to 224 x 224. that will significantly improve quality of the model while also speeding up the processing process

Dataset	ISIC Challenge 2019
Type	Dermoscopic
Image size	1022 pixels × 767 pixels
Number of images	25,333
Image type	JPEG (RGB)
Class labels	0: Melanoma 1: Melanocytic Nevus 2: Basal Cell Carcinoma 3: Actinic Keratosis 4: Benign Keratosis 5: Dermatofibroma 6: Vascular Lesion 7: Squamous Cell Carcinoma

Table IV.1: Dataset information for ISIC 2019 [25]

Image Normalization

Image Normalization is a method that is used to normalize the pixel values of a picture so that they have a similar distribution. It is advantageous to normalize images before feeding them into the neural network because this speeds up the approach to the global minima at the error surface while performing gradient descent. In some ways, it aids the network's convergence. Furthermore, as all pixel values are scaled, the computations become significantly less intensive for the machine to perform [25].

Use of metadata

Although each single lesion could show up multiple times in the training datasets presented by ISIC2019 (up to 15 pictures for the same lesion), there is indeed a real risk of inflated validation results if the validation set and the training set intersect on the level of lesions. As a result, we had to use metadata to ensure that there is no lesion-level

overlap between these sets.

Data balancing

Because the data is highly imbalanced, we use the over/up-sampling data balancing method for training, which involves multiplying the data with various ratios on various classes.

Data augmentation

Different data augmentation techniques have been adopted to the training set in this section. The computational cost was reduced by using smaller pixel values within the same range, which was achieved through scale transformation. As a result of the parameter value, the value of each pixel ranged from 0 to 1. Because the rotation transformation was used to rotate the images to a specific angle, 15 was used to rotate the images. Using the width shift range transformation, images can be shifted arbitrarily to the right or left. The width shift parameter was set to (0.1) . The height shift range parameter was set to 0.1 to move the training images vertically.

Shear transformation is a method whereby one axis of a picture is fixed and the other axis is stretched to a specific angle known as a shear angle; in this case, a 0.15 shear angle was used. To perform the random zoom transformation, the zoom range argument was used; a value greater than 0.1 indicates that the images were magnified. As a result, the image was magnified using a zoom range of 0.1. Flip was used to rotate the image horizontally. The brightness transformation, with 0.0 representing no brightness and 1.0 representing maximum brightness, was used.

As a consequence, a zoom range of 0.1–1.0 was employed. The channel values are randomly shifted by a random value selected from a specific range in channel shift transformation; as a result, the 15 channel shift range was used, and the fill mode was the closest, as shown in Table .

After applying the mentioned methods in previous paragraphs in data preparation phase(images resizing ,data sampling , use of meta-data ..) we made some analysis about the amount

Transformations	Setting
Rotation transformation	15°
Zoom transformation	0.1
Horizontal flip	True
Shear transformation	0.15
channel shift	15

Table IV.2: Image augmentation specifications

of pitures in each class before the augmentation phase we found the following results as shown in the pictures below:

Output :

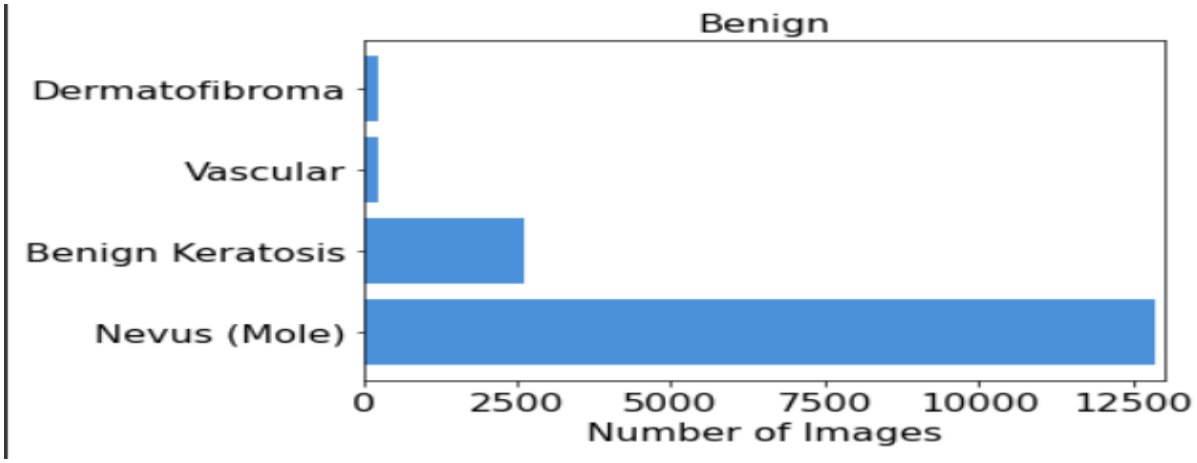

Figure IV.4: Number of images in the classes of benign before augmentation

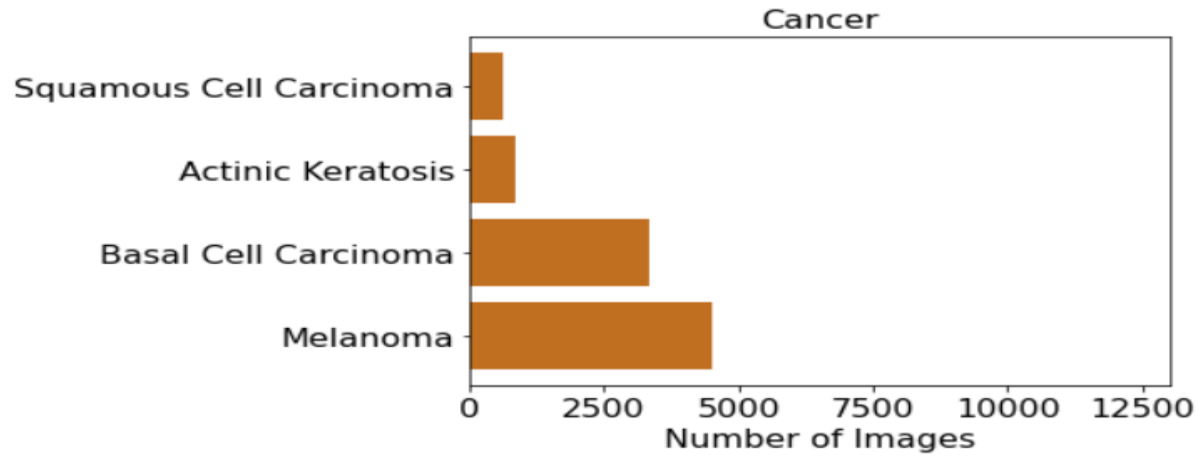

Figure IV.5: Number of images in the classes of cancer

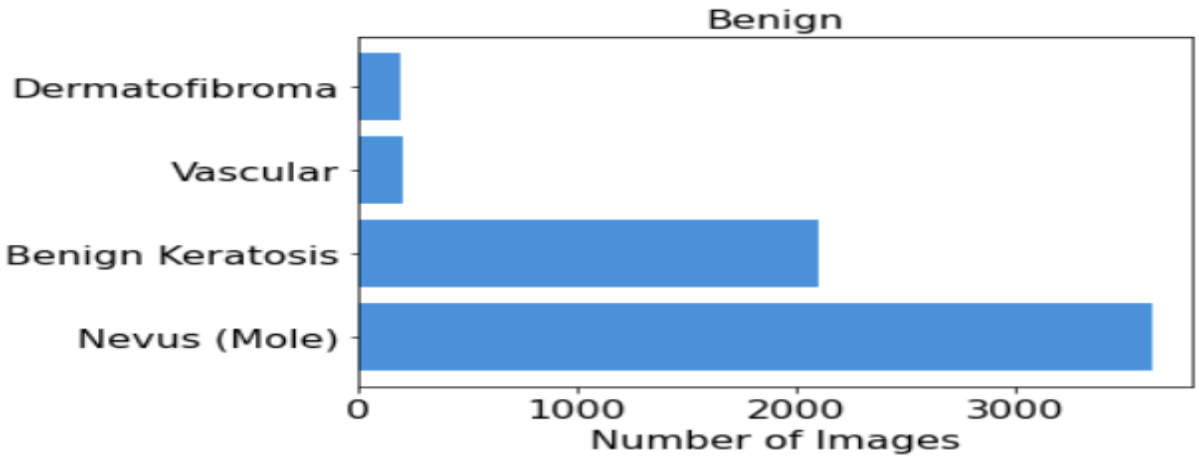

Figure IV.6: number of images per class in training data after undersampling for benign classes

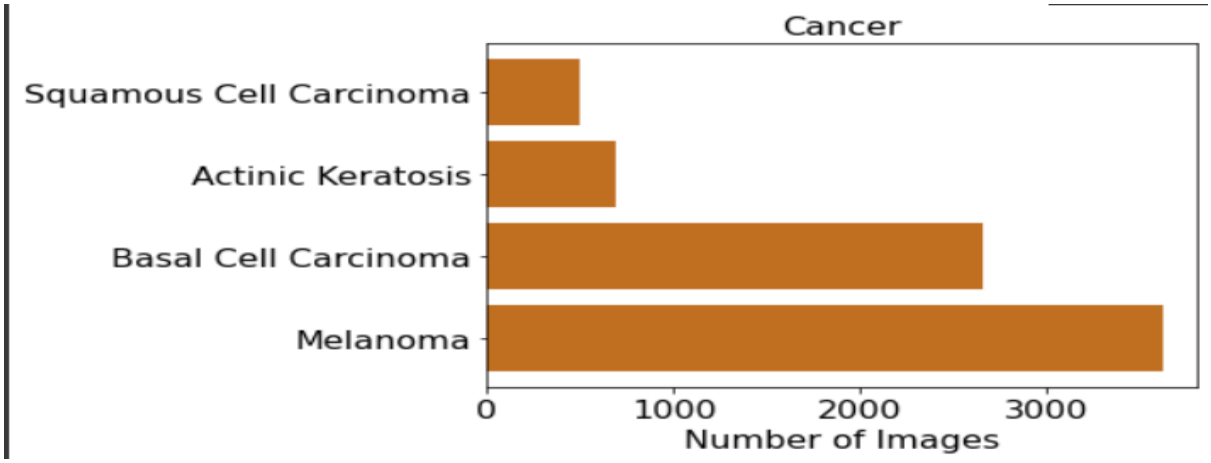

Figure IV.7: number of images per class in training data after undersampling for cancer classes

So here after launching the appropriate lines of code that concern the augmentation phase we got 8 balanced classes as shows in pictures below :

```
nv 3617
Need around 1 augmentations per image to balance to 3617
bkl 2099
Need around 1 augmentations per image to balance to 3617
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:32: TqdmDeprecat
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
100% 2099/2099 [21:24<00:00, 393.33it/s]
bkl data points =  3617  -- skipping more augmentations...
vasc 202
Need around 17 augmentations per image to balance to 3617
100% 202/202 [24:41<00:00, 6.19s/it]
vasc data points =  3619  -- skipping more augmentations...
df 191
Need around 18 augmentations per image to balance to 3617
100% 191/191 [25:08<00:00, 7.40s/it]
```

Figure IV.8: Execute augmentation on training data for benign classes

```
mel 3617
Need around 1 augmentations per image to balance to 3617
bcc 2658
Need around 1 augmentations per image to balance to 3617
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:32: TqdmDeprecati
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
100% 2658/2658 [05:05<00:00, 856.33it/s]
bcc data points =  3617  -- skipping more augmentations...
akiec 693
Need around 5 augmentations per image to balance to 3617
100% 693/693 [13:12<00:00, 19.47it/s]
akiec data points =  3618  -- skipping more augmentations...
scc 502
Need around 7 augmentations per image to balance to 3617
100% 502/502 [21:21<00:00, 3.66s/it]
scc data points =  3617  -- skipping more augmentations...
```

Figure IV.9: Execute augmentation on training data for cancer classes

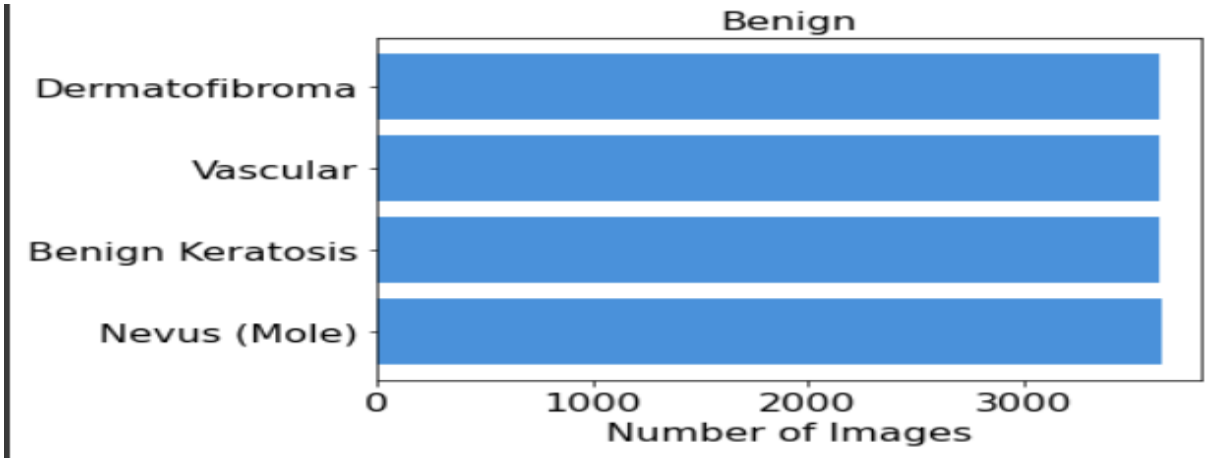

Figure IV.10: number of images per class in training data after augmentation

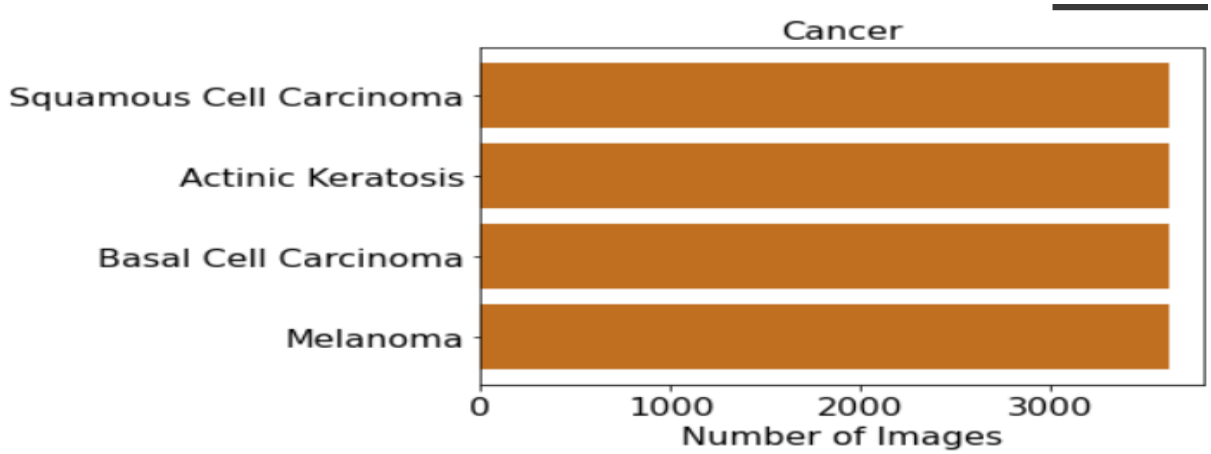

Figure IV.11: number of images per class in training data after augmentation

classes Labels	before augmentation	after augmentation
Dermatofibroma	191	3617
Vascular	202	3617
Benign Keratosis	2099	3617
Nevus(Mole)	3617	3617
Squamous Cell Carcinoma	502	3617
Actinic Keratosis	693	3617
Basal Cell Carcinoma	2658	3617
Melanoma	3617	3617

Table IV.3: Number of pictures per classe before and after augmentation

Output:

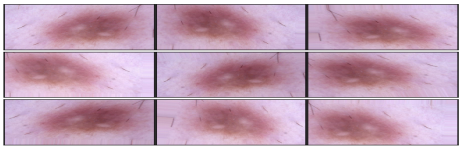

Figure IV.12: Exemple of Augmentation

2.2 Model Training and Evaluation

In this phase we are going to train our model using the specifications mentionned in the table (IV.4) .First we need to reload our metadata files and Pytorch DataSets , then importing the pretrained ResNet-18 with ImageNet weights to adapt it in our model to proceed in splitting our dataset for training and validation then loading the training phase.

Parameters	Values
Architecture used	ResNet-18
Type of Transfer	None
Train Layers	3/4
Learning Algorithm	Adam
Activation Function	Relu , Sigmoid , LogSoftmax
Loss Function	Binary Cross Entrophy
Batch Size	64
Epochs	50
Learning Rate	0.00005

Table IV.4: DL Model Hyper Parameters

The results after splitting the dataset into two parts first for training and the second for validation as fllows(Table IV.5) :

Output : (28951, 3391)

Dataset	Training	Validation
ISIC 2019	28951	3391

Table IV.5: Summary of the ISIC 2019 dataset after splitting

Loading the Pre-trained CNN Model then Adapt it for Custom Dataset

The Pytorch API calls a pre-trained model of ResNet18 by using models.resnet18(pretrained=True),

the function from TorchVision's model library. **Output :**

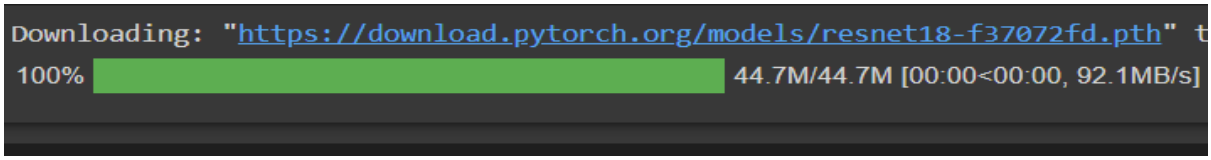

Figure IV.13: Importing pretained ResNet-18 with ImageNet weights

After we trained our model we got the following results :

Output :

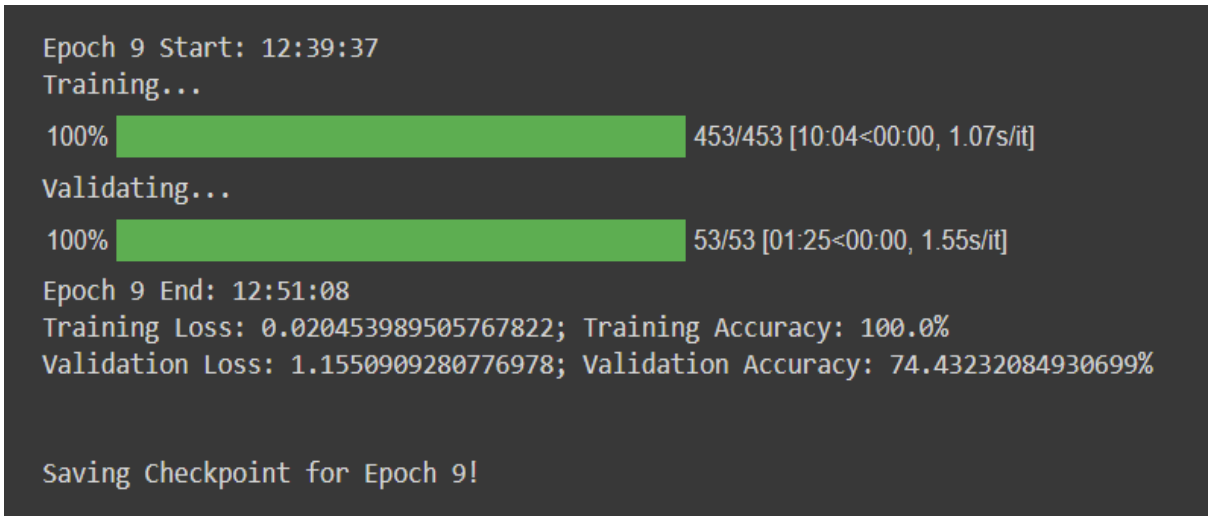

Figure IV.14: Model training Epoch 9 .

2.3 Validation and Metrics

we apply the following metrics to validate the efficiency of our model.

Classification Accuracy

The classification accuracy is calculated by dividing the number of correct predictions by the total number of accurate predictions. [31].

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} [31]$$

Where :

TN : The true negative [31] & TP : The true positive [31]

FN : The false negative [31] & FP : The false positive [31]

Precision

the Numerous examples show that classification accuracy is not always a reliable indicator of overall model performance. One of these situations is when the distribution of classes is skewed. If we assume that all samples are of the highest quality, we will obtain a high accuracy rate, which is illogical. Precision, on the other hand, indicates that inconsistency can be found when using the same instrument repeatedly, such as when measuring the same part classification accuracy is measured as the percentage of correct predictions to the total number of accurate predictions. Precision is one of such measures, which is characterized as [31]:

$$Precision = \frac{TP}{(TP + FP)}[31]$$

Recall

Another important estimator is recall, which is characterized as dividing input samples into classes that the system successfully predicts. The recall is calculated as[31]:

$$Recall = \frac{TP}{(TP + FN)}[31]$$

For example, if a skin lesion image is labeled with melanoma and the model predicts it to be melanoma, this is considered a true positive case. False negative occurs when an image is labeled with melanoma but is classified as any of the other six classes[31]. A false positive case occurs when the classification model indicates that a skin lesion image has melanoma but it actually belongs to one of the other six diseases. If the classifier suggests that a non-melanoma skin lesion image is non-melanoma, this is a case of true negative[31].

F1-SCORE

The f1 score is a well-known statistic that combines precision and recall into a single number. The f1 score is calculated as [**?**]:

$$f1SCORE = \frac{2 * (Precision * Recall)}{(Precision + Recall)} [31]$$

Confusion Matrix

We used the Confusion Matrix to see how well our model performs in each class.

AUC Score and ROC Curve

The receiver operating characteristic (ROC) is a probability curve, and the area under curves (AUC) shows the level of separability. The ROC curve is a graph that illustrates the relationship among specificity (false positive rate) and sensitivity (true positive rate) [31].

Outputs:

We plot the Confusion Matrix to see how well our model performs in each class so we got the following results :

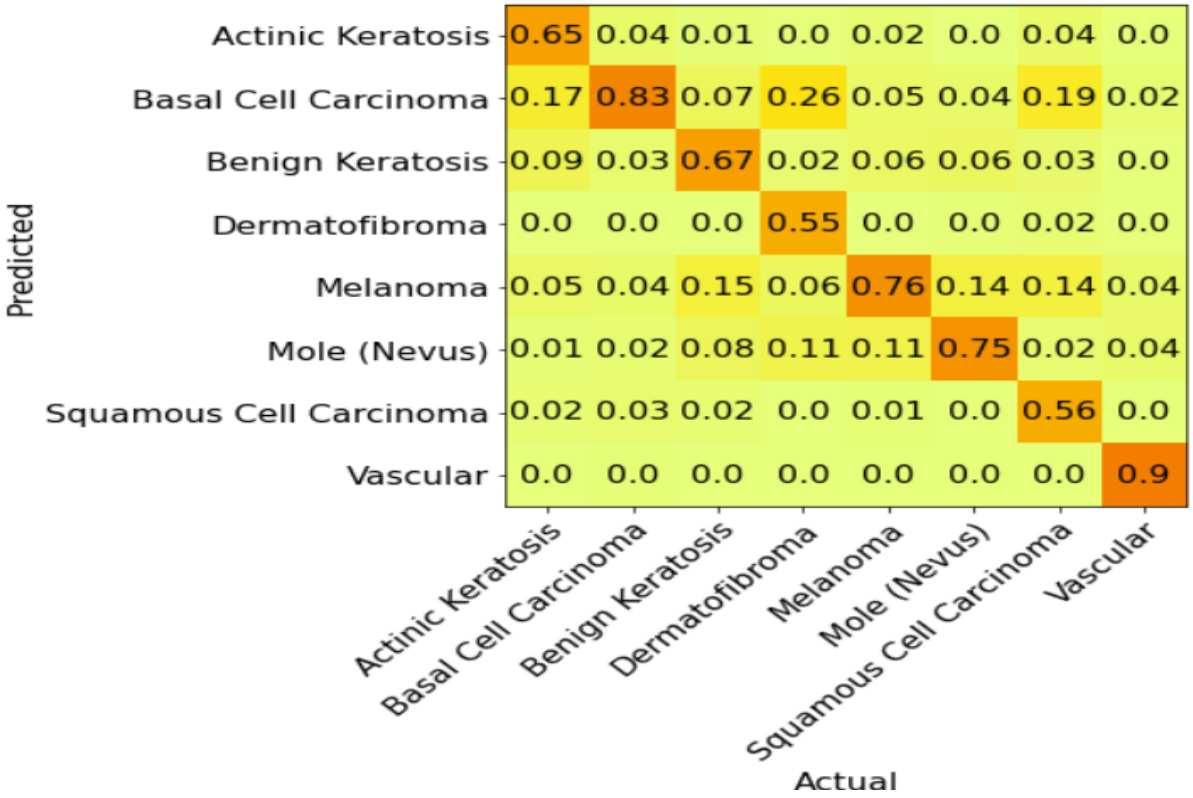

Figure IV.15: Confusion Matrix Validation Metric

The confusion matrix reveals that the method performing well in classifying the various types of skin lesions in the validation data. Let's compute some metrics to represent the fact that some classes perform better than others.

ROC curves for each class:

Output :

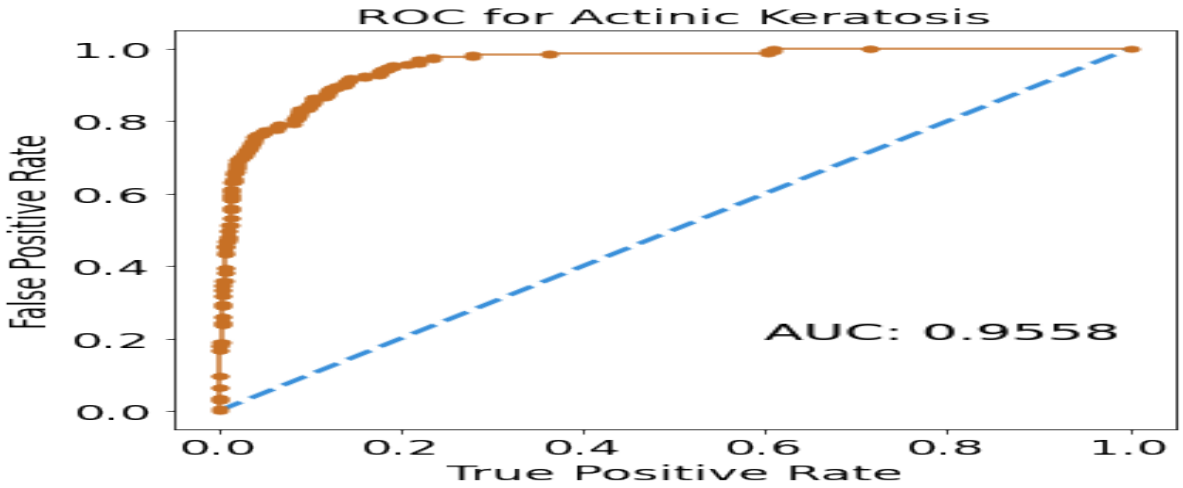

Figure IV.16: ROC for Actinic Keratosis

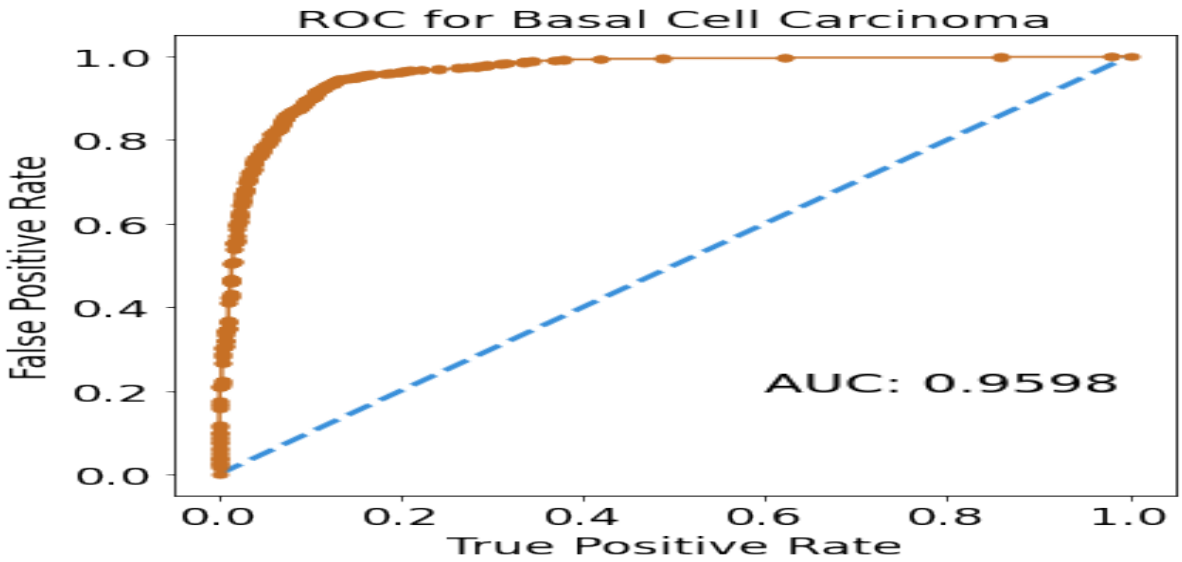

Figure IV.17: ROC for Basal Cell Carcinoma

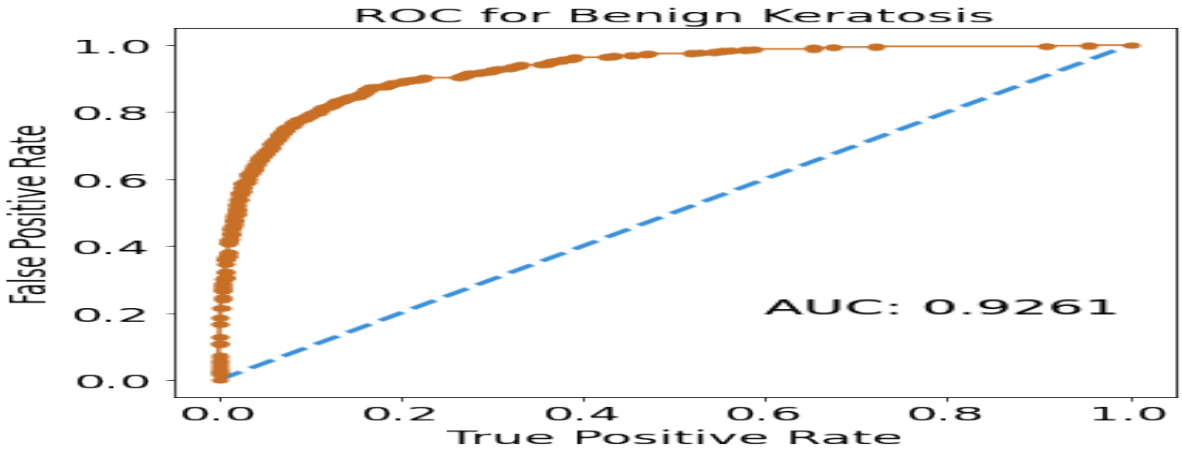

Figure IV.18: ROC for Benign Keratosis

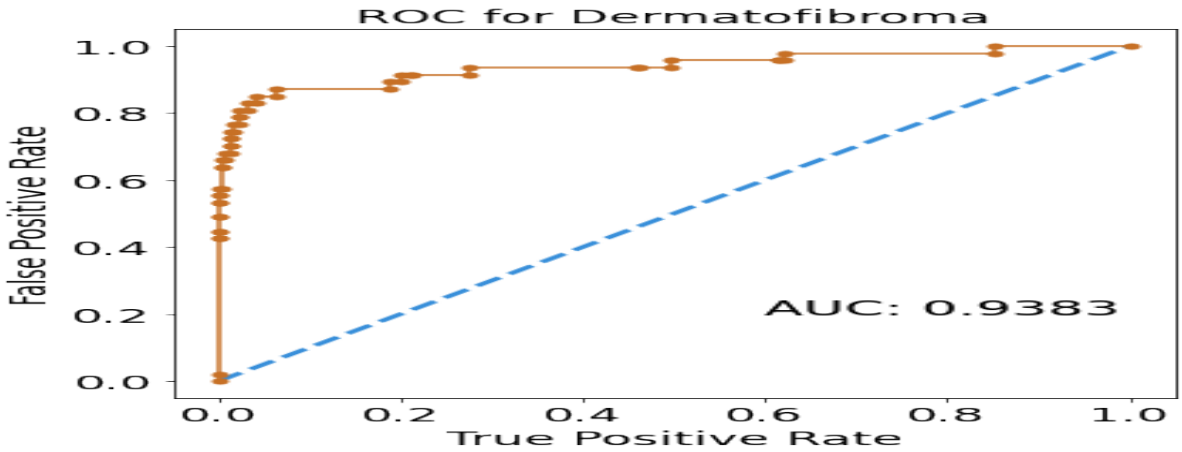

Figure IV.19: ROC for Dermatofebroma

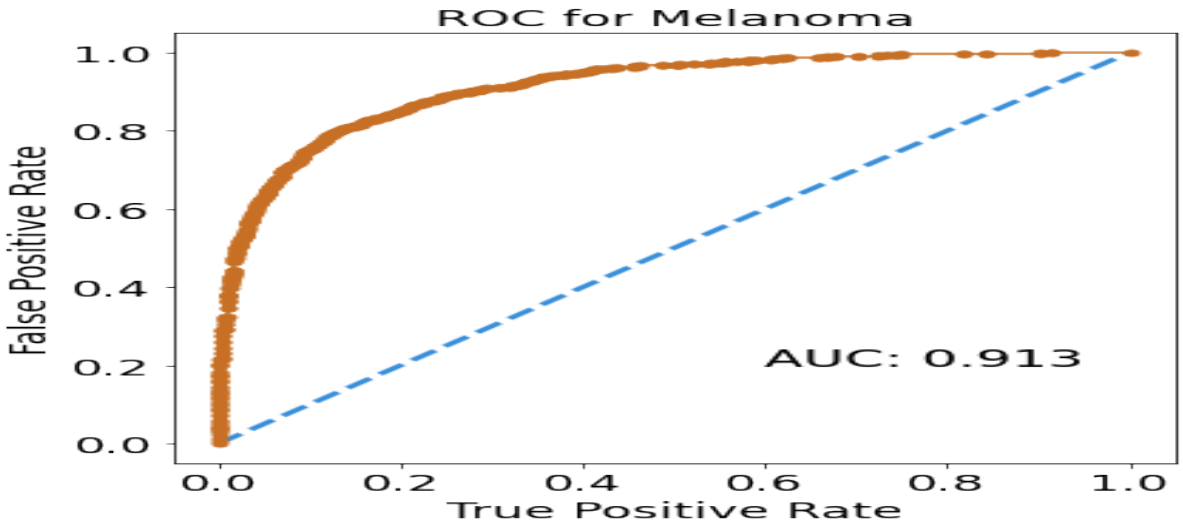

Figure IV.20: ROC for Melanoma

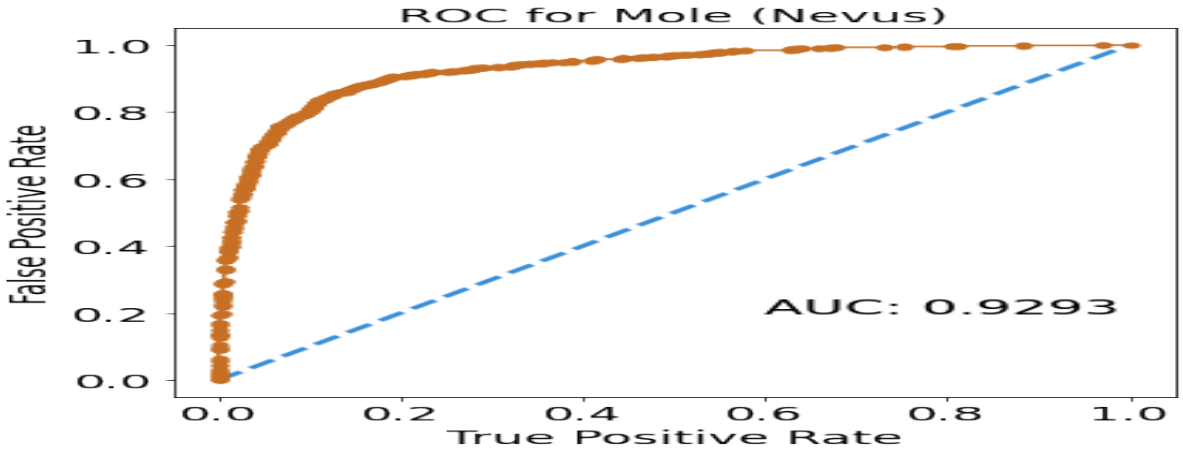

Figure IV.21: ROC for Mole(Nevus

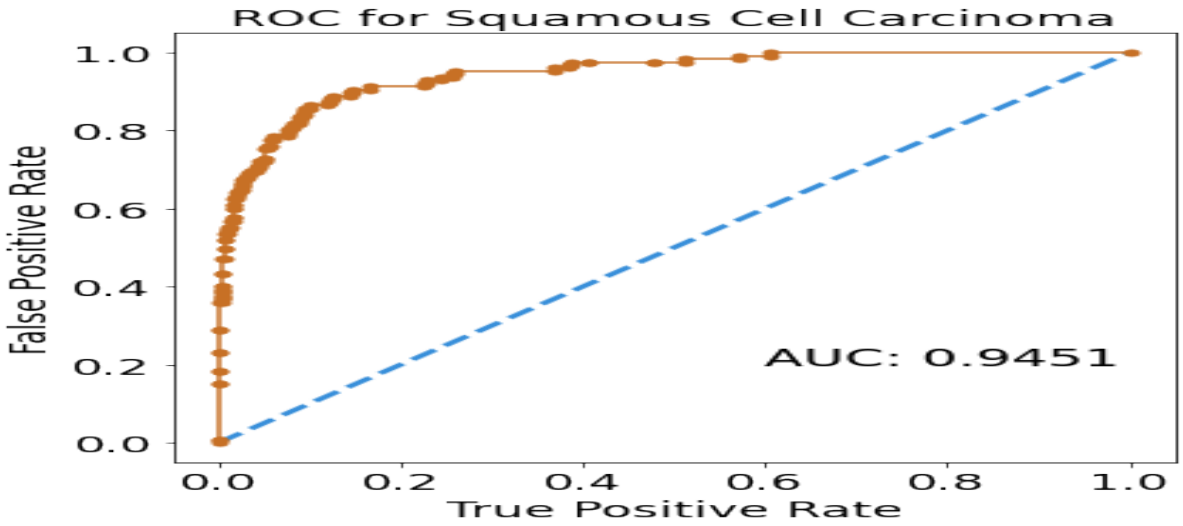

Figure IV.22: ROC for Squamous Cell Carcinoma

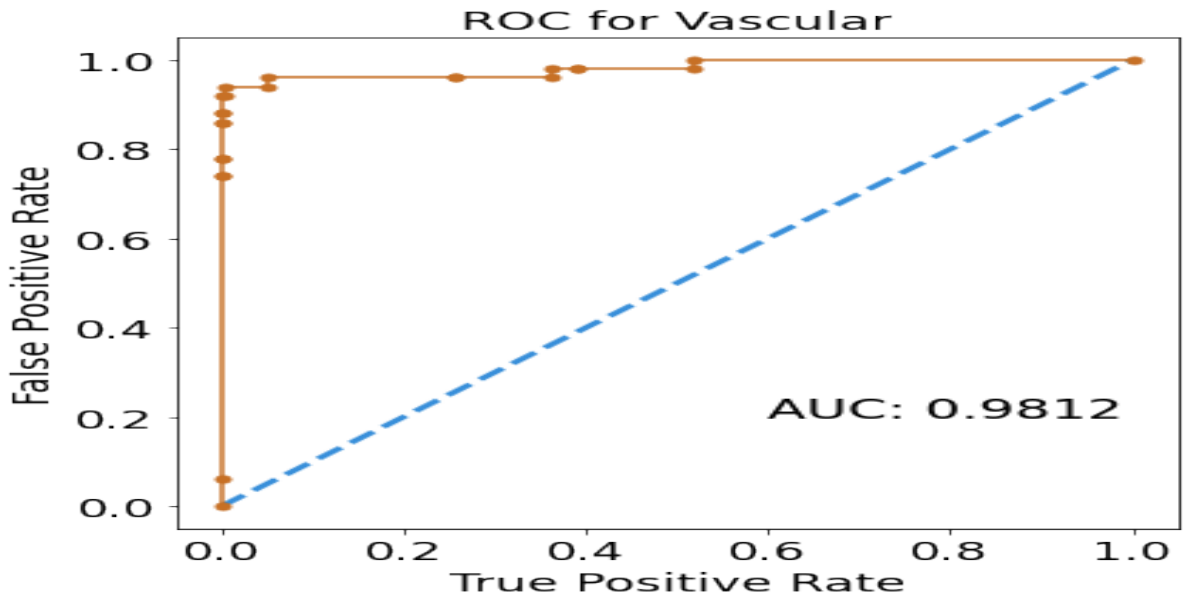

Figure IV.23: ROC for Vascular

Precision/Recall Curves for Each Class

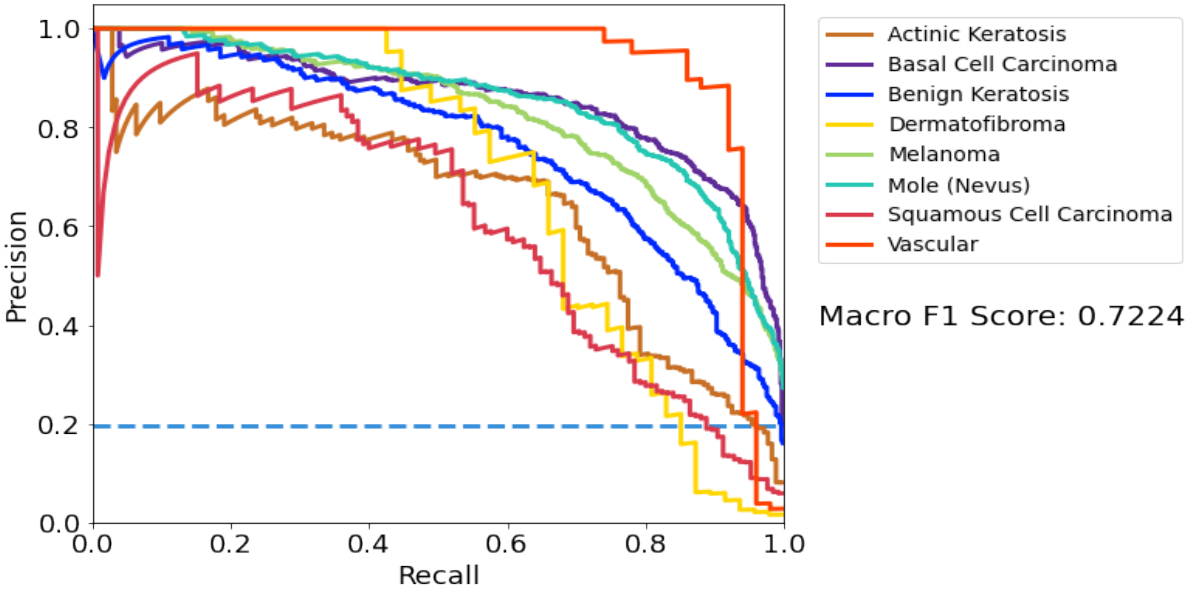

Figure IV.24: Precision and recall curve

The model has precision and recall of roughly 69.9% across all classes (usually) based on the macro F1 Score. Let's see what the baseline macro F1 is that we'd expect from random chance.

Output : 0.13537791847105424

So, if we were to randomly assign our class predictions, taking into account the total number of images from each class in our validation split, we'd expect an F1 score of 12.8% overall. Fair to say our model is performing well over baseline!

Conclusion

In this chapter Through Code portions of the proposed deep learning model we explained the necessary steps that we used for its implemntation besides to the solution methods for the dataset problems(unbalanced classes , multi-images resolutions and missing data) .

General Conclusion

General Conclusion

within the current study, ResNet-18 pretrained DL model has been developped for detecting cancer and benign skin lesions, that can be applied to any malicious lesion. The proposed mechanism is used to determine whether a disease is malignant or benign using images from the ISIC2019 challenge dataset of skin cancer. To increase the dataset size and improve ResNet- 18 accuracy, data augmentation strategies have been used. This structure works well with a training accuracy of 100% a validation accuracy of 74,43%.

The proposed structure was developped to provide excellent validation accuracy out of needing for model training from scratch to improve the effectiveness of the model. After collecting a huge amount of pictures with high-dimensions, we conclude that it is more appropriate and produces more accurate and efficient results than traditional biopsy methods. Patient information plays a crucial role in the diagnosis process.

this research will be conducted on a set of skin cancer photos for patient population, in the future we are willing to add more features to this system such as statistics to improve our system's functionalities.

Bibliography

[1] Zeina Al Masryb Yazid Bourezanec Noureddine Zerhounib Abir Belaalaa, Labib Sadek Terrissaa. Towards improved skin lesions classification using automatic hyperparameters selection and transferlearning. *Journal of Artificial Intelligence in Medicine*, 28(1):05, 2020.

[2] Nikolas Adaloglou. Best deep cnn architectures and their principles: from alexnet to efficientnet. *AI Summer*, 2021.

[3] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.

[4] Bishwaranjan Bhattacharjee, Scott Boag, Chandani Doshi, Parijat Dube, Ben Herta, Vatche Ishakian, KR Jayaram, Rania Khalaf, Avesh Krishna, Yu Bo Li, et al. Ibm deep learning service. *IBM Journal of Research and Development*, 61(4/5):10–1, 2017.

[5] Bishwaranjan Bhattacharjee, Scott Boag, Chandani Doshi, Parijat Dube, Ben Herta, Vatche Ishakian, KR Jayaram, Rania Khalaf, Avesh Krishna, Yu Bo Li, et al. Ibm deep learning service. *IBM Journal of Research and Development*, 61(4/5):10–1, 2017.

[6] David R Bickers, Henry W Lim, David Margolis, Martin A Weinstock, Clifford Goodman, Eric Faulkner, Ciara Gould, Eric Gemmen, and Tim Dall. The burden

of skin diseases: 2004: A joint project of the american academy of dermatology association and the society for investigative dermatology. *Journal of the American Academy of Dermatology*, 55(3):490–500, 2006.

[7] Avijeet Biswal. Top 10 deep learning algorithms you should know in 2021. *Simplilearn. Simplilearn. com/tutorials/deep-learning-tutorial/deep-learning-algorithm*, 2021.

[8] Titus Josef Brinker, Achim Hekler, Jochen Sven Utikal, Niels Grabe, Dirk Schadendorf, Joachim Klode, Carola Berking, Theresa Steeb, Alexander H Enk, and Christof Von Kalle. Skin cancer classification using convolutional neural networks: systematic review. *Journal of medical Internet research*, 20(10):e11936, 2018.

[9] Chensi Cao, Feng Liu, Hai Tan, Deshou Song, Wenjie Shu, Weizhong Li, Yiming Zhou, Xiaochen Bo, and Zhi Xie. Deep learning and its applications in biomedicine. *Genomics, proteomics & bioinformatics*, 16(1):17–32, 2018.

[10] Bill Cassidy, Connah Kendrick, Andrzej Brodzicki, Joanna Jaworek-Korjakowska, and Moi Hoon Yap. Analysis of the isic image datasets: usage, benchmarks and recommendations. *Medical Image Analysis*, 75:102305, 2022.

[11] IBM Cloud Education. Machine learning [internet] 2020.

[12] IBM Cloud Education. Machine learning [internet] 2020.

[13] Kavita Gandhi, Khaled Ezzedine, Kathryn P Anastassopoulos, Reema Patel, Vanja Sikirica, Shoshana R Daniel, Lynne Napatalung, Yuji Yamaguchi, Rebecca Baik, and Amit G Pandya. Prevalence of vitiligo among adults in the united states. *JAMA dermatology*, 158(1):43–50, 2022.

[14] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. *MethodsX*, 7:100864, 2020.

[15] K Goyal. Deep learning vs neural networks: Difference between deep learning and neural networks, 2019.

[16] Jian Bin Guo, Lukas Leung, and John Magee. Academic aide—free online math question database for academic improvement. In *2015 IEEE Integrated STEM Education Conference*, pages 195–200. IEEE, 2015.

[17] Michael R Hamblin, Pinar Avci, and Gaurav K Gupta. *Imaging in dermatology*. Academic Press, 2016.

[18] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[19] Roderick J Hay, Nicole E Johns, Hywel C Williams, Ian W Bolliger, Robert P Dellavalle, David J Margolis, Robin Marks, Luigi Naldi, Martin A Weinstock, Sarah K Wulf, et al. The global burden of skin disease in 2010: an analysis of the prevalence and impact of skin conditions. *Journal of Investigative Dermatology*, 134(6):1527–1534, 2014.

[20] Miikka Lukumies. Qt framework in modern embedded user interface development. 2019.

[21] Amitha Mathew, P Amudha, and S Sivakumari. Deep learning techniques: an overview. In *International conference on advanced machine learning technologies and applications*, pages 599–608. Springer, 2020.

[22] Amitha Mathew, P Amudha, and S Sivakumari. Deep learning techniques: an overview. In *International Conference on Advanced machine learning technologies and applications*, pages 599–608. Springer, 2020.

[23] Andreas Mayr, Benjamin Lutz, Michael Weigelt, Tobias Gläßel, Dominik Kißkalt, Michael Masuch, Andreas Riedel, and Jörg Franke. Evaluation of machine learning for quality monitoring of laser welding using the example of the contacting of hairpin

windings. In *2018 8th International Electric Drives Production Conference (EDPC)*, pages 1–7. IEEE, 2018.

[24] Henri Michel. Google colab: Le guide ultime.", 2019.

[25] Hardik Nahata and Satya P Singh. Deep learning solutions for skin cancer detection and diagnosis. In *Machine Learning with Health Care Perspective*, pages 159–182. Springer, 2020.

[26] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons. b*, 4:51–62, 2017.

[27] Jean-Phillip Okhovat, Derek Beaulieu, Hensin Tsao, Allan C Halpern, Dominique S Michaud, Shimon Shaykevich, and Alan C Geller. The first 30 years of the american academy of dermatology skin cancer screening program: 1985-2014. *Journal of the American Academy of Dermatology*, 79(5):884–891, 2018.

[28] Andre GC Pacheco and Renato A Krohling. An attention-based mechanism to combine images and metadata in deep learning models applied to skin cancer classification. *IEEE journal of biomedical and health informatics*, 25(9):3554–3563, 2021.

[29] Junjie Peng, Elizabeth C Jury, Pierre Dönnes, and Coziana Ciurtin. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: applications and challenges. *Frontiers in Pharmacology*, page 2667, 2021.

[30] Rahul. Artificial intelligence demystified.

[31] Javed Rashid, Maryam Ishfaq, Ghulam Ali, Muhammad R Saeed, Mubasher Hussain, Tamim Alkhalifah, Fahad Alturise, and Noor Samand. Skin cancer disease detection using transfer learning technique. *Applied Sciences*, 12(11):5714, 2022.

[32] LTD Riverbank Computing. Pyqt, 2020.

[33] Stacey Ronaghan. Deep learning: Overview of neurons and activation functions. *Medium. com, Jul*, 2018.

[34] Stacey Ronaghan. Deep learning: Overview of neurons and activation functions. *Medium. com, Jul*, 2018.

[35] Neha Sharma, Reecha Sharma, and Neeru Jindal. Machine learning and deep learning applications-a vision. *Global Transitions Proceedings*, 2(1):24–28, 2021.

[36] Ryohei Takahashi and Yuya Kajikawa. Computer-aided diagnosis: A survey with bibliometric analysis. *International journal of medical informatics*, 101:58–67, 2017.

[37] M Tejaswini, P Pranuthi, S Ravichand, and T Anuradha. Land cover change detection using convolution neural network. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 791–794. IEEE, 2019.

[38] Victor Amoako Temeng, Yao Yevenyo Ziggah, and Clement Kweku Arthur. A novel artificial intelligent model for predicting air overpressure using brain inspired emotional neural network. *International Journal of Mining Science and Technology*, 30(5):683–689, 2020.

[39] Manuel Valdebran, Kruti Parikh, Deborah S Sarnoff, and Dirk M Elston. Irregular pigmented lesion on the genital area. *Indian Dermatology Online Journal*, 7(2):117, 2016.

[40] Guido Van Rossum et al. Python programming language. In *USENIX annual technical conference*, volume 41, pages 1–36. Santa Clara, CA, 2007.

[41] Marianne Winslett and Vanessa Braganholo. Richard hipp speaks out on sqlite. *ACM SIGMOD Record*, 48(2):39–46, 2019.

[42] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495, 2017.

[43] Mansheng Xiao, Yuezhong Wu, Guocai Zuo, Shuangnan Fan, Huijun Yu, Zeeshan Azmat Shaikh, and Zhiqiang Wen. Addressing overfitting problem in deep

learning-based solutions for next generation data-driven networks. *Wireless Communications and Mobile Computing*, 2021, 2021.

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[45] Frank F Xu, Bogdan Vasilescu, and Graham Neubig. In-ide code generation from natural language: Promise and challenges. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(2):1–47, 2022.

[46] Rui Yin, Tanupriya Agrawal, Usman Khan, Gaurav K Gupta, Vikrant Rai, Ying-Ying Huang, and Michael R Hamblin. Antimicrobial photodynamic inactivation in nanomedicine: small light strides against bad bugs. *Nanomedicine*, 10(15):2379–2404, 2015.

[47] Kechi Zhang, Wenhan Wang, Huangzhao Zhang, Ge Li, and Zhi Jin. Learning to represent programs with heterogeneous graphs. In *2022 IEEE/ACM 30th International Conference on Program Comprehension (ICPC)*, pages 378–389. IEEE, 2022.