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

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of
the Requirements for the Degree of :

MASTER

In **Electronics**

Option: **Telecommunications**

Title:

Melanoma identification using convolutional neural networks (CNN)

Presented by:

- Boumali Khaled
- Bourtache Mohamed Islam

Supervisor:

Dr.Cherifi Dalila

Registration number :………/2019

Dedication

The five years in this institute have come to an end, i remember like yesterday when i entered the room where Mrs.Ourad, my old teacher of listening and speaking told us that once the door closed we have to forget all our shyness, to let go all the ideas that come into our minds. I dedicate this to her beacause she is the first person that i saw in INELEC , or should I say the right person to see first.

I dedicate this to Mrs.Cherifi, because we gave her so much trouble and in return she gave us so much hope and confidence in ourselves.

Mom, Dad, you don not need this dedication because you dedicated your life for us and I will do the same by God willing.

I dedicate this to the jury, you are the best teachers ever and for God's sake if you are reading this, go easy on us in the questions, a lot of people will be watching ...

Charles Buckowski wrote a book once and said « dedicated to nobody », loneliness is cruel this is why I am so glad that i have all my family who supports me and so many friends with me so I dedicate this to you all.

Finally ,they said choosing your partner for the final year project defence is harder than choosing your partner for the rest of your life , and by this I mean my wife , Oh God it rhymes ! We were on those stairs waiting for the final mark in the make up and I told you Khaled if we make it this year, we will work together ,destiny is beautiful we are finally here and I wanted to say that you are the best Khaled and I wish for us the best of luck !

Thank you.

Islam

Dedication

I dedicate this humble work to my parents because they were always there for me, for my sisters, my closest friends who represent a whole family and without forgetting my institition mates.

To Dr.Cherifi, her kindness and her support for all the students, she is the mother who is always there.

To the teachers of Inelec who did their best in growing our knowledge.

To everyone who gave the least support for moving on

And at last but not least, to the best partner ever with whom I learned a lot and with whom I passed my best two years at Inelec.

Thank you !

Khaled

Acknowledgement

We Thank God The Almighty

A special thanks to our supervisor Dr. Cherifi Dalila for her continuous help and immense suupport.

We also thank the staff of IGEE for providing a convenient environment without forgetting all the ones that contributed in this document.

Abstract

Melanoma is a malignant skin cancer with an increasing incidence. To reduce mortalitiy rates due to melanoma, early detection must be taken into consideration. One of the fastest and most useful ways to achieve early detection is to go through Deep learning, and more specificically CNN model whose ouput classifies whether the patient is suffereing from melanoma or not. For a better detection, the accuracy of the CNN output has to be high enough so that the patient gets a true result about their state. One of the the hyper-paramters of the CNN model, which leads to more accuracte results, is to add more hidden layers to our model ,at the same time, apply the data augmentation technique for a more perfromant model, and that is what we did in the first part of our project

In computer vision, transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve, we want to make a tool capable of detecting melanoma with a higher accuracy than the previous one.

In this project, we also focused nusing thepretrained models (VGG, ResNet, Inception and Xception) we then compared the results of our work with the existing works..

Contents

List of Figures

List of Tables

Introduction

Melanoma is the deadliest of skin cancers. Rates of diagnosis for the disease have increased dramatically over the past three decades, outpacing almost all other cancers. Today, it is one of the most common cancers found among young adults in the World. 96,480 people in the United States are expected to be diagnosed with new melanomas in 2019. Over 7,000 people in the United States are expected to die of melanoma in 2019— about 4,740 men and 2,490 women[1].

The rate of clinician's visual investigations gets at 65% of good detection at the very best. It is very difficult to distinguish some atypical lesions - which are benign - from melanoma because they have the same properties according to the well-known rules used by dermatologists.

Diagnostic instruments that allow early detection of melanoma are the key to improving survival rates and reducing the number of unnecessary biopsies, the associated morbidity and the costs of care.Advances in technology over the past two decades have enabled the development of new,sophisticated test methods which are currently undergoing laboratory and small-scale clinical testing.

Our purpose in this project is using Convolutional Neural Networks which work best for unstructured data, we will also depend on pre-trained models which will save us a lot of time.

Chapter 1

MELANOMA SKIN CANCER

1.1 Definition Of Melanoma :

Melanoma starts in skin cells called melanocytes. A cancerous (malignant) tumor is a group of cancer cells that can invade and destroy nearby tissue. A tumor can also spread (metastasize) to other parts of the body. Most pigment cells are found in the skin, but melanoma can also occur in the eyes (ocular melanoma) and other parts of the body, including, rarely, the intestines.

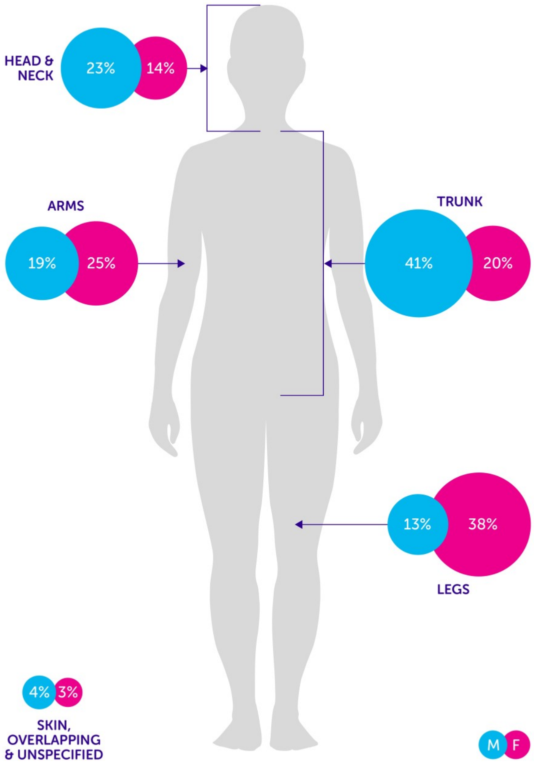


Figure 1.1: Repartition of Melanoma in the Human Body

The medical oncologist is the doctor who has special training in diagnosing and treating cancer in adults using chemotherapy , hormonal therapy and targeted therapy , he is the main health care provider for someone who has cancer .

Rich Joseph , a medical oncologist at Mayo Clinic in Jacksonville Florida explained to us that melanoma is distinct from other skin cancers and that is usually a poliferation of pigmented cells , it can also originate in different parts of mucosal tract including the head and neck as well as the gastrointestinal tract as well as on the bottoms of the palms and the soles which usually does not occur in other types of skin cancers.

In stage one melanoma has a cure rate of about 95% , in stage two it is a thicker melanoma which is still in the skin but has not spread to the lymph nodes or the body has still has a pretty high chance of cure of catching it early somewhere in the range of 78% to 80% , stage three melanoma means that it has escaped from the skin to the lymph nodes , stage three is devided in three stages : A , B and C depending on how many lymph nodes are involved and the cure rate drops all the way down as low as 20% if many lymph nodes are involved and as high as 60% to 70% if only one lymph is involved , a stage four melanoma is a melanoma that has left the lymph nodes and gone to two distant parts of the body and there the survival is much less the overal average median.

1.2 Warning Signs ABCDEs :

Moles, brown spots and growths on the skin are usually harmless — but not always. Anyone who has more than 100 moles is at greater risk for melanoma. The first signs can appear in one or more atypical moles. That’s why it’s so important to get to know our skin very well and to recognize any changes in the moles on our body. Look for the ABCDE signs of melanoma [1].

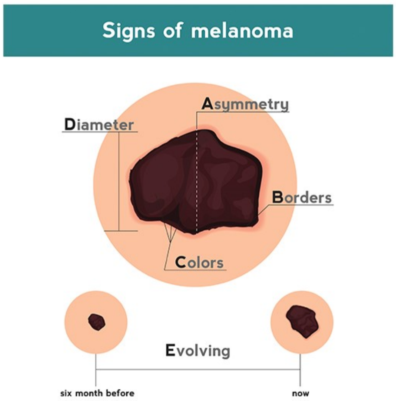


Figure 1.2: Warning signs of melanoma ABCDE [1]

1.2.1 Sign A (Assymetry) :

The benign mole, left, is not asymmetrical. If you draw a line through the middle, the two sides will match, meaning it is symmetrical. If you draw a line through the mole on the right, the two halves will not match, meaning it is asymmetrical, a warning sign for melanoma.

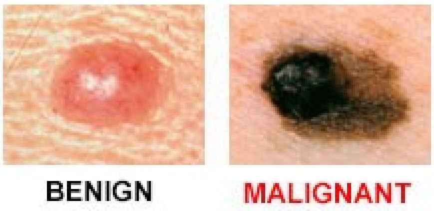


Figure 1.3: Sign A [1]

1.2.2 Sign B (Border) :

A benign mole has smooth, even borders, unlike melanomas. The borders of an early melanoma tend to be uneven. The edges may be scalloped or notched.

Figure 1.4: Sign B [1]

1.2.3 Sign C (Color) :

Most benign moles are all one color — often a single shade of brown. Having a variety of colors is another warning signal. A number of different shades of brown, tan or black could appear. A melanoma may also become red, white or blue.

Figure 1.5: Sign C [1]

1.2.4 Sign D (Diameter) :

Benign moles usually have a smaller diameter than malignant ones. Melanomas usually are larger in diameter than the eraser on your pencil tip ($\frac{1}{4}$ inch or 6mm), but they may sometimes be smaller when first detected.

Figure 1.6: Sign D [1]

1.2.5 Sign E (Evolving) :

In Common, benign moles look the same over time. Be on the alert when a mole starts to evolve or change in any way. When a mole is evolving, see a doctor. Any change — in size, shape, color, elevation, or another trait, or any new symptom such as bleeding, itching or crusting — points to danger.

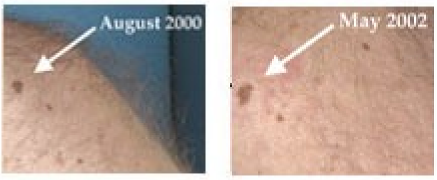


Figure 1.7: Sign E [1]

1.3 Causes Of Melanoma [2]:

1.3.1 Sun Exposure :

Both UVA and UVB rays are dangerous to the skin, and can induce skin cancer, including melanoma. Blistering sunburns in early childhood especially increase risk, but sunburns later in life and cumulative exposure also may be factors. People who live in locations that have more sunlight — like Florida, Hawaii, and Australia — develop more skin cancers, but some more northern locations with light-skinned populations also have a high number of skin cancers. Avoid using a tanning booth or tanning bed, since it increases your exposure to UV rays, raising your risk of developing melanoma and other skin cancers.

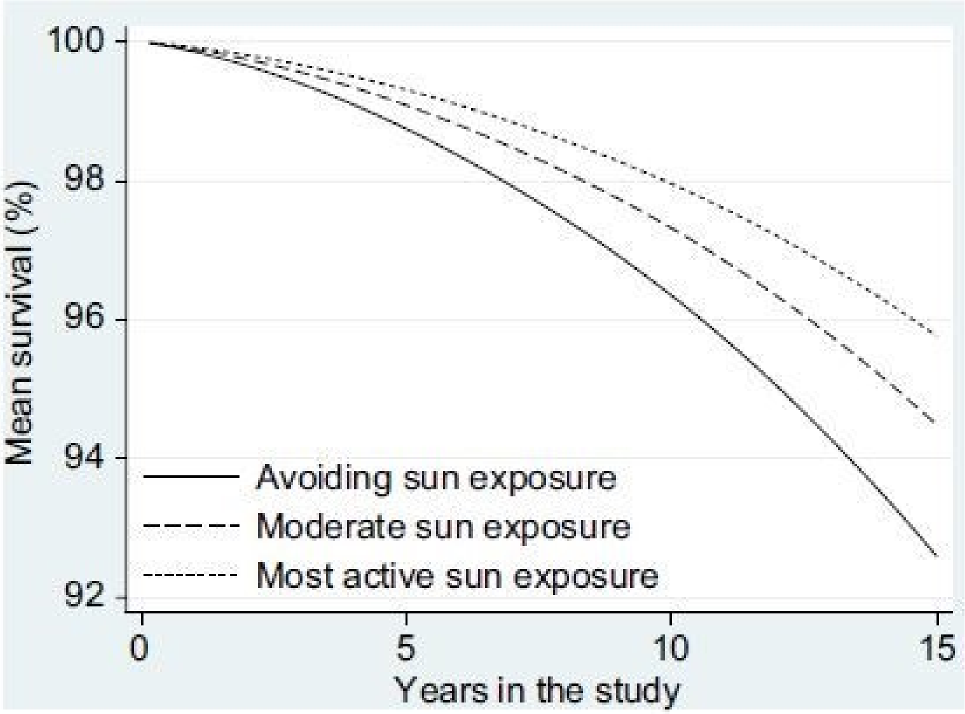


Figure 1.8: Adjusted all-cause survival plot of 30 000 women followed for sun exposure habits for 20 years

1.3.2 Moles :

There are two kinds of moles: normal moles — the small brown blemishes, growths, or "beauty marks" that appear in the first few decades of life in almost everyone — and atypical moles, also known as dysplastic nevi. Atypical moles can be precursors to melanoma and having them puts you at increased risk of developing the disease. But regardless of type, the more moles you have, the greater your risk. People with 50 or more moles are considered to be at increased risk of developing melanoma.

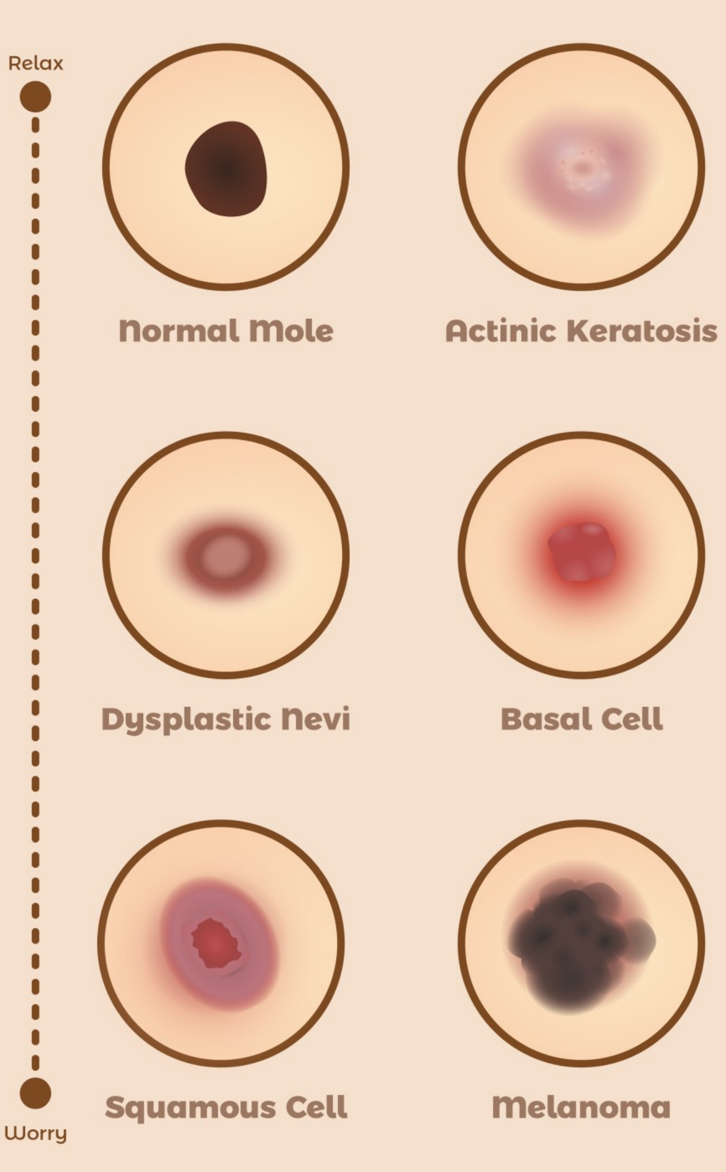


Figure 1.9: Different kinds of moles

1.3.3 Skin Type :

As with all skin cancers, people with fairer skin (who often have lighter hair and eye color as well) are at increased risk.

1.3.4 Weakened Immune System :

Compromised immune systems as the result of chemotherapy, an organ transplant, excessive sun exposure, and diseases such as HIV/AIDS or lymphoma can increase your risk of melanoma.

1.3.5 Heredity :

Heredity plays a major role in melanoma. About one in every 10 patients diagnosed with the disease has a family member with a history of melanoma. If mother, father, siblings or children have had a melanoma, we are then in a melanoma-prone family. Each person with a first-degree relative diagnosed with melanoma has a 50% greater chance of developing the disease than people who do not have a family history of the disease.

1.3.6 Family Syndrome :

When atypical moles are found in an individual belonging to a melanoma family, the condition is known as FAMMM, standing for Familial Atypical Multiple Mole Melanoma Syndrome. People with this syndrome are at the greatest risk of developing melanoma. In contrast, a research study found that those family members who did not have atypical moles were much less likely to develop melanoma.

1.3.7 Children : a Special case:

Children in melanoma-prone families need special care, because familial melanoma is likely to make its appearance early in life. Even though these cancers usually do not appear until after adolescence, they may arise in much younger children who have a family history of melanoma. Most physicians, therefore, advise parents to make a point of studying a child's skin frequently from infancy on. Physician examination in these families should start at the age of 10 and continue on a twice-a-year basis thereafter. Particular care should be taken at puberty and during adolescence when hormonal changes activate the moles.

1.4 Types Of Melanoma [3]:

The Four Basic Types Melanomas fall into four basic categories. Three of them begin in situ — meaning they occupy only the top layers of the skin — and sometimes become invasive; the fourth is invasive from the start. Invasive melanomas are more serious, as they have penetrated deeper into the skin and may have spread to other areas of the body.

1.4.1 Superficial spreading melanoma :

is by far the most common type, accounting for about 70 percent of all cases. This is the one most often seen in young people. As the name suggests, this melanoma grows along the top layer of the skin for some time before penetrating more deeply.The first sign is the appearance of a flat or slightly raised discolored patch that has irregular borders and is somewhat asymmetrical in form. The color varies, and you may see areas of tan, brown, black, red, blue or white. This type of melanoma can occur in a previously benign mole or arise as a new lesion. It can be found almost anywhere on the body but is most likely to occur on the trunk in men, the legs in women, and the upper back in both.Lentigo maligna is similar to the superficial spreading type, as it also remains close to the skin surface for a while, and usually appears as a flat or mildly elevated mottled tan, brown or dark brown discoloration. This type of in situ melanoma is found most often in the elderly, arising on chronically sun-exposed, damaged skin on the face, ears, arms and upper trunk.

1.4.2 Lentigo maligna

is the most common form of melanoma in Hawaii. When this cancer becomes invasive, it is referred to as lentigo maligna melanoma.

1.4.3 Acral lentiginous

melanoma also spreads superficially before penetrating more deeply. It is quite different from the others, though, as it usually appears as a black or brown discoloration under the nails or on the soles of the feet or palms of the hands. It is the most common melanoma in African-Americans and Asians, and the least common among Caucasians.

1.4.4 Nodular melanoma

is usually invasive at the time it is first diagnosed. The malignancy is recognized when it becomes a bump. It is usually black, but occasionally is blue, gray, white, brown, tan, red or skin tone.The most frequent locations are the trunk, legs and arms, mainly of elderly people, as well as the scalp in men.

This is the most aggressive of the melanomas, and is found in 10% to 15% of cases

Lentigo maligna **Superfitialy spread melanoma (SSM)**

Nodular melanoma **Acrolentiginous melanoma**

Figure 1.10: Different types of melanoma in the human body

1.5 The Stages Of Melanoma [4]:

Once the type of melanoma has been established, the next step is to classify the disease as to its degree of severity.Classifications for melanomas are called stages. The stage depends on the thickness, depth of penetration, and the degree to which the melanoma has spread. The staging is used to determine treatment. Early melanomas (Stages 0 and I) are localized; Stage 0 tumors are in situ, meaning they are noninvasive and have not penetrated below the outer layer of the skin (the epidermis). **Stage I** tumors have invaded below the epidermis into the skin's next layer (the dermis) but are small and have no other traits such as ulceration that put them at high risk of spreading (metastasizing) to nearby lymph nodes or beyond.**Stage II** tumors, though localized, are larger (generally 1 mm. thick or larger) and/or may have other traits such as ulceration that put them at high risk of spreading to the nearby lymph nodes or beyond. They are considered intermediate or "high-risk" melanomas.

More advanced melanomas (**Stages III and IV**) have metastasized to other parts of the body. There are also subdivisions within stages.

1.5.1 Early Melanoma

The thickness of the tumor, known as Breslow thickness or Breslow-depth, and the appearance of microscopic ulceration (meaning the epidermis on top of a major portion of the melanoma is not intact) continue to be the most important factors in staging primary melanoma tumors, while mitotic rate, the speed of cell division (how fast-growing the cancer cells are), has been dropped as a staging criterion for stage I tumors.

1.5.2 High Risk Melanoma

Intermediate, high-risk (**stage II**) tumors are 1.0 to 4.0 mm and/or are ulcerated. They are at high risk of spreading (metastasizing) to nearby lymph nodes or beyond, and patients are typically advised to undergo sentinel lymph node biopsy. Thick melanomas (still clinically **stage II**, but at very high risk of metastasizing) are greater than 4.0 mm, and patients are typically advised to undergo sentinel lymph node biopsy.

1.5.3 Advanced Melanoma

Stage III. By the time a melanoma advances to Stage III or beyond, an important change has occurred. Breslow thickness is by then irrelevant and is no longer included in staging, but the presence of microscopic ulceration continues to be used, as it has an important effect on the progression of the disease. At this point, the tumor has either spread to the nearby lymph nodes or to the skin between the primary tumor and those lymph nodes. (All tissues are bathed in lymph — a colorless, watery fluid consisting mainly of white blood cells — which drains into lymphatic vessels and lymph nodes throughout the body, potentially carrying cancer cells to distant organs.)

Stage IV. The melanoma has metastasized to lymph nodes distant from the primary tumor or to internal organs, most often the lung, followed in descending order of frequency by the liver, brain, bone and gastrointestinal tract. The two main factors in determining how advanced the melanoma is into Stage IV (the “M” category, for “metastases”) are the site of the distant metastases (nonvisceral, lung or any other visceral metastatic sites) and whether or not the serum lactate dehydrogenase (LDH) level is elevated. LDH, an enzyme found in your blood and almost every other cell of your body, turns sugar into energy, and the more you have in your blood or other body fluid, the more damage has been done to your body's tissues [5].

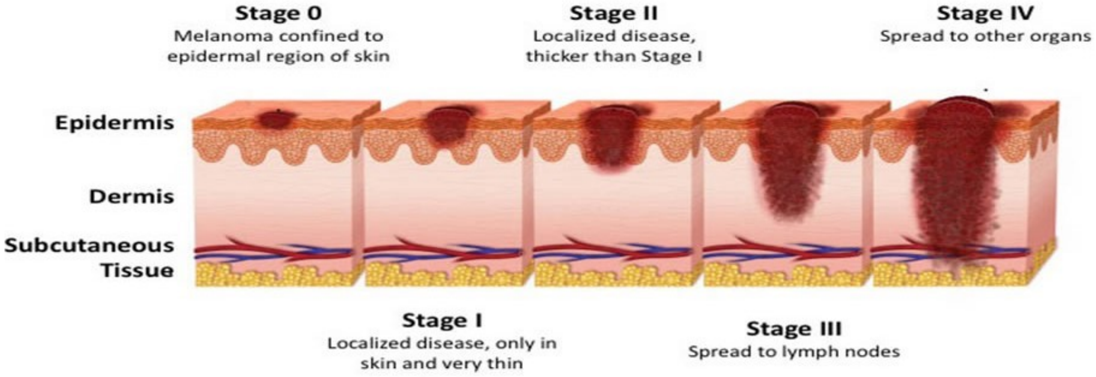


Figure 1.11: Stages Of Melanoma [5]

1.6 Treatments [6]:

1.6.1 For Stage I And Stage II Melanoma:

The first step in treatment is removal of the primary melanoma tumor, and the standard method of doing this is by surgical excision (cutting it out). Surgery has made great advances in the past decade, and much less tissue is removed than was customary in the past. Patients do just as well after the lesser surgery, which is easier to tolerate and produces a smaller scar.Surgical excision also called resection, and the borders of the entire area excised are known as the margins.

- **Office Surgery**
- **Setting the margins**
- **Mohs Micrographic Surgery**

1.6.2 For Stage III And Stage IV Melanoma:

When melanoma cells spread to the lymph nodes (stage III) or more distant parts of the body (stage IV), including organs, the disease is considered advanced, and additional therapy usually follows surgical removal of the original (primary) skin tumor. Several treatments are employed :

- **Immunotherapy**
- **Targeted Therapy**
- **Chemeotherapy**
- **Radiation Therapy**

1.7 Summary

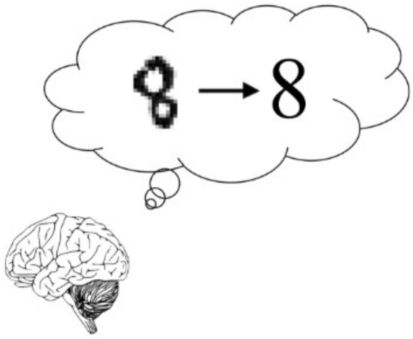
In this chapter, we defined the disease which we will be trying to identify in the third chapter, namely melanoma. We went through all details where we explained the causes, the warning signs, the different steps of melanoma, and the ways to treat it.

Chapter 2

The Theory Of Artificial Neural Network (ANN)

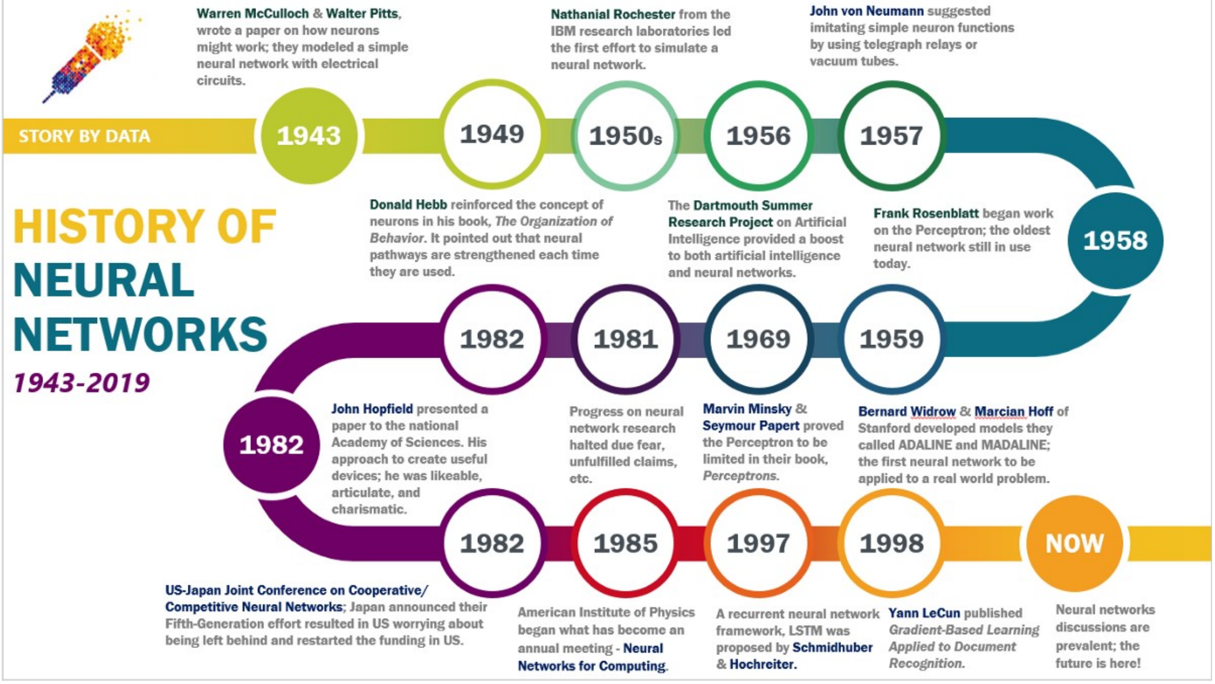
2.1 Introduction:

This is an eight, it is sloppily written and rendered at an extremely low resolution of (28 by 28 pix) but the brain has no trouble recognizing it as an eight, so how can the brain do this so effortlessely? So what if we are told to write a program that takes a grid of 28 by 28 pixels and outputs a single number between 0 and 10 telling us what he thins the digit is, so here comes the question, what is a neural network?



2.2 History and origins :

The idea of neural networks began unsurprisingly as a model of how neurons in the brain function, termed 'connectionism' and used connected circuits to simulate intelligent behaviour. In 1943, portrayed with a simple electrical circuit by neurophysiologist Warren McCulloch and mathematician Walter Pitts. Donald Hebb took the idea further in his book, The Organization of Behaviour (1949), proposing that neural pathways strengthen over each successive use, 'Threshold Logic' and 'Hebbian Learning' two major concepts both proposed in the 1940's. In 1950s, around this time, Frank Rosenblatt, proposed the idea of a Perceptron in 1958. Things started moving quickly for neural networks around this time and in 1959 at Stanford, Bernard Widrow and Marcian Hoff developed the first neural network successfully applied to a real world problem. These systems were named ADALINE and MADALINE after their use of Multiple ADAptive LINear Elements, in 1982 at the National Academy of Sciences Jon Hopfield presented his paper on what came to be known as Hopfield Net. In 1997 a recurrent neural network framework, Long Short-Term Memory (LSTM) was proposed by Schmidhuber & Hochreiter. In 1998, Yann LeCun published Gradient-Based Learning Applied to Document Recognition [7].



Warren McCulloch & **Walter Pitts**, wrote a paper on how neurons might work; they modeled a simple neural network with electrical circuits.

Donald Hebb reinforced the concept of neurons in his book, *The Organization of Behavior*. It pointed out that neural pathways are strengthened each time they are used.

Nathanial Rochester from the IBM research laboratories led the first effort to simulate a neural network.

The **Dartmouth Summer Research Project** on Artificial Intelligence provided a boost to both artificial intelligence and neural networks.

John von Neumann suggested imitating simple neuron functions by using telegraph relays or vacuum tubes.

Frank Rosenblatt began work on the Perceptron; the oldest neural network still in use today.

Bernard Widrow & **Marcian Hoff** of Stanford developed models they called ADALINE and MADALINE; the first neural network to be applied to a real world problem.

Marvin Minsky & **Seymour Papert** proved the Perceptron to be limited in their book, *Perceptrons*.

Progress on neural network research halted due fear, unfulfilled claims, etc.

John Hopfield presented a paper to the national Academy of Sciences. His approach to create useful devices; he was likeable, articulate, and charismatic.

US-Japan Joint Conference on Cooperative/ Competitive Neural Networks; Japan announced their Fifth-Generation effort resulted in US worrying about being left behind and restarted the funding in US.

American Institute of Physics began what has become an annual meeting - **Neural Networks for Computing**.

A recurrent neural network framework, LSTM was proposed by **Schmidhuber** & **Hochreiter**.

Yann LeCun published *Gradient-Based Learning Applied to Document Recognition*.

Neural networks discussions are prevalent; the future is here!

Figure 2.1: The history of artificial intelligence shown in a historical map [7]

2.3 Biological neural network Vs Artificial neural network :

Neural Networks attempt to bring computers a little closer to the brain's capabilities by imitating aspects of information in the brain in a highly simplified way. Information processing in biological neural networks is usually slow; this is because neurons need several milliseconds to react to a stimulus. In artificial neural networks information processing is very fast because electronic gates which they use to operate can achieve switching times of a few nanoseconds. The key idea behind neural network is that they can take in a lot of data, process it in parallel, and provide accurate output, much as the human brain does. For example, when you see an elephant, you know right away that it is an elephant. You don't have to stop and count legs or look at the shape and color. You process all of that data at the same time to know that you see an elephant. That's what an artificial neural network does for a computer system [8].

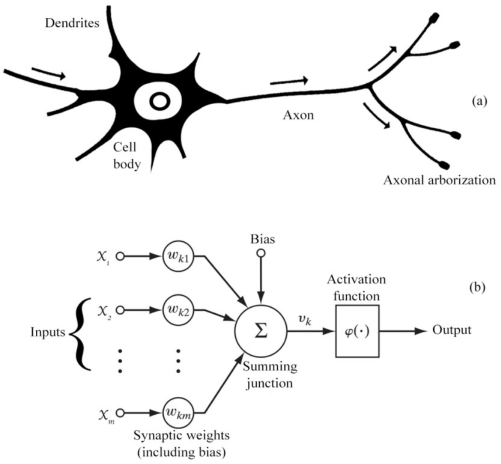


Figure 2.2: Similarity between biological NN and artificial NN [8]

Biological neural network	**Artificial neural network**
Have the following components : - Soma present - Dendrites present - Synapse present - Axon present	Have the following components : - Node present - Input present - Weight present-Output present
10^{11} neurons	$10^2 - 10^4$ neurons
Biological neural networks are constructed in a three dimensional way from microscopic components. These neurons seem capable of nearly unrestricted interconnections.	In the Artificial neural networks, this is not true. These are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers, which are then connected to one another.
Neural tissue involves random connectivity with no master blueprint.	Connectivity is precisely specified.
Information is stored at the synapses	Information is stored at the weights matrix.
Communications media in neural tissue involve stochastic.	They are deterministic and formally structured.

Table 2.1: Comparaison between the biological and artificial networks [8]

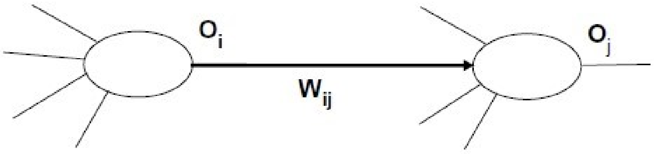


Figure 2.3: Basic perceptron architecture

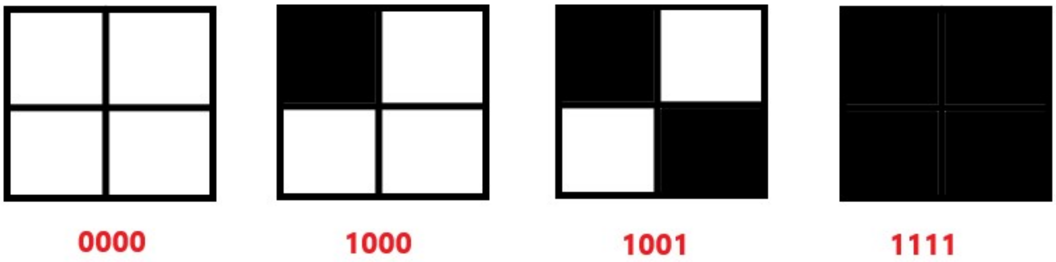


Figure 2.4: Sample four pixel images

2.4 Artificial Neural Networks :

2.4.1 The Perceptron :

Back to 1949, Donald Olding Hebb a Canadian neuropsychologist had influenced the domain of Artificial Intelligence and inspired the perceptron, his famous rule specified that the synaptic weight between two neurons is reinforced when the latter two are active simultaneously, mathematically it gives us this equation:

$$W_{ij}(t+1) = W_{ij}(t) + u \times O_i \times O_j where 0 \leq u \leq 1 \tag{2.1}$$

The perceptron use another learning rule which takes into account the expected result, it is considered that the perceptron is the first network of evolutionary artificial neurons that capable of learning, so In 1957 Franck Rosenblatt invented the perceptron, a machine capable of recognizing the alphabet with photoelectric cells as sensor.

To understand it we will use a simple example, a perceptron of two layers will recognize a four pixel square when the pixel is **white =0** and when the pixel is **black =1**, we will try to find the **(1001) square** which has two opposite black pixels, there are in all sixteen possible squares.

The entrance layer receives the information and transmits 1 or 0, one neuron will compose the exit layer and will be equipped with four synaptic weights, and an additional entrance which we will call er receives the input and transmits 1 or 0 - threshold entrance - with a synaptic coefficient of -1, the output is calculated as the weighted sum of inputs with synaptic coefficients.

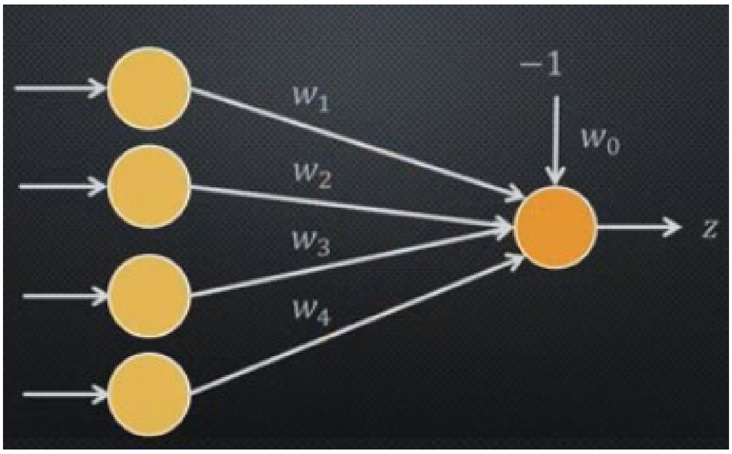


Figure 2.5: The preceptron used in the example

$$Y = \sum_{i=1}^{4} w_i x_i + w_0 x_0 And Z = f(y) while f(y) = \begin{cases} 1, if y \geq 0 \\ 0, if y < 0 \end{cases} \tag{2.2}$$

The learning process will consist of a series of steps designed to adjust the five synaptic coeffcients to obtain a 100% recognition rate.

$W_1 = W_2 = W_3 = W_4 = 0 And W_0 = 2$

$Wi(t+1) = Wi(t) + (d - z)Xi/u = 1$ Where d is the expected value.

$(X_0, X_1, X_2, X_3, X_4) = (-1, 1, 0, 0, 1)$

$Y = W_0X_0 + W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4$

$Y = 2 \times (-1) + 0 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 1 = -2$

$Z = f(y) = 0$ because (Y is negative) but the expected value $d = 1$.

So : $W_0(t+1) = W_0(t) + (d - z)X_0 = 2 + (1 - 0) \times (-1) = 2–1 = 1$ And So On …

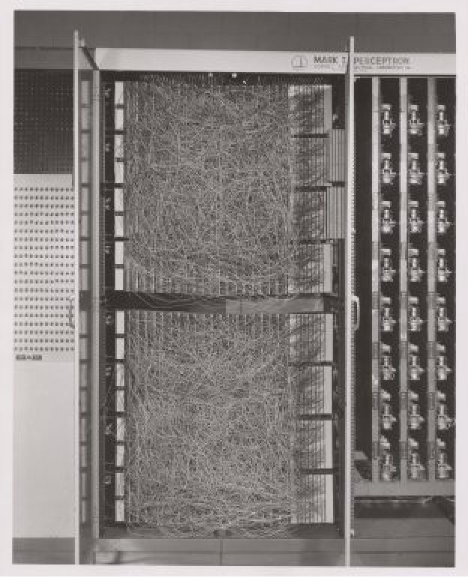


Figure 2.6: First Perceptron Created By Franck Rosenblatt In 1957 [9]

2.4.2 Multilayer Perceptron:

A multilayer perceptron (MLP) is composed of more than one perceptron. MLPs are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between, a number (arbitrary) of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

The artificial neural network paradigm has consistently demonstrated its effectiveness as a reliable nonlinear classification technique [9].

The multi-layer perceptron (MLP), a feed-forward supervised neural network often used for data classification, requires the desired class label for each sample in order that it may be compared to the actual output generated by the MLP. The learning strategy minimizes a global error function, $E = 0.5 \sum k(d_k - O_k)^2$ (where d_k and O_k are the respective components of the desired and actual outputs), for the set of input data. The MLP uses the back-propagation algorithm to pass this error back throughout the network in a localized, neuron-based, fashion . In general terms, MLP may be considered a non-linear regression system that performs a gradient descent search through the weight space, searching for (local) minima.

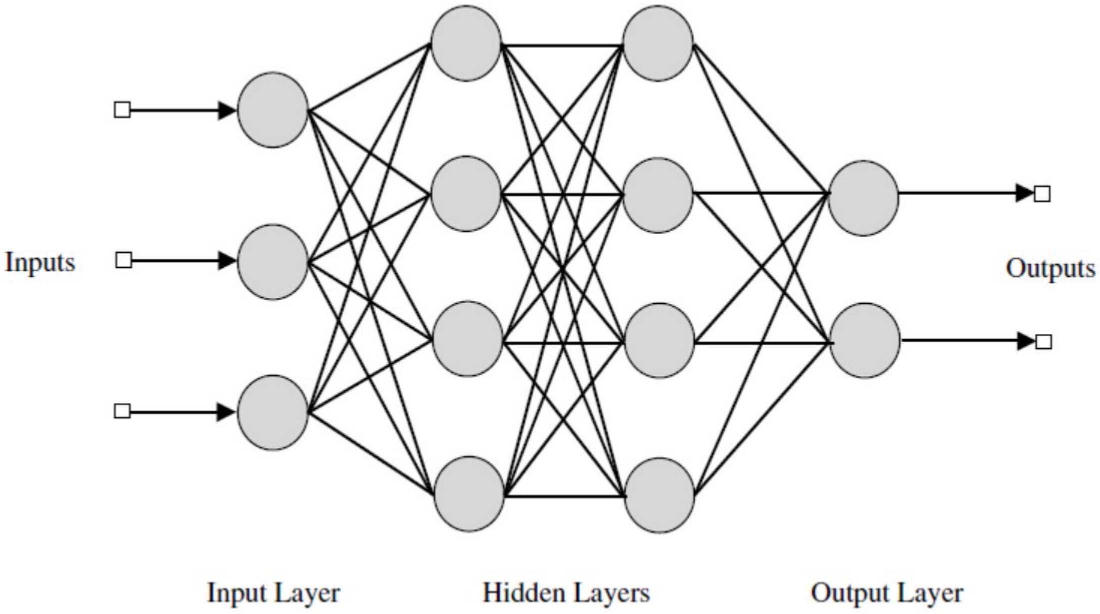


Figure 2.7: Multilayer Perceptron : Neural Network Model (Two hidden layers) [10]

2.4.3 Activation Functions [9]:

Neural network activation functions are a crucial component of deep learning. Activation functions determine the output of a deep learning model, its accuracy, and also the computational efficiency of training a model—which can make or break a large scale neural network. Activation functions also have a major effect on the neural network's ability to converge and the convergence speed, or in some cases, activation functions might prevent neural networks from converging in the first place.

The activation function takes into account the interaction effects in different parameters and does a transformation after which it gets to decide which neuron passes forward the value into the next layer.

It doesn't matter how many hidden layer we attach in neural net, all layers will behave same way because the composition of two linear function is a linear function itself. Neuron can not learn with just a linear function attached to it.

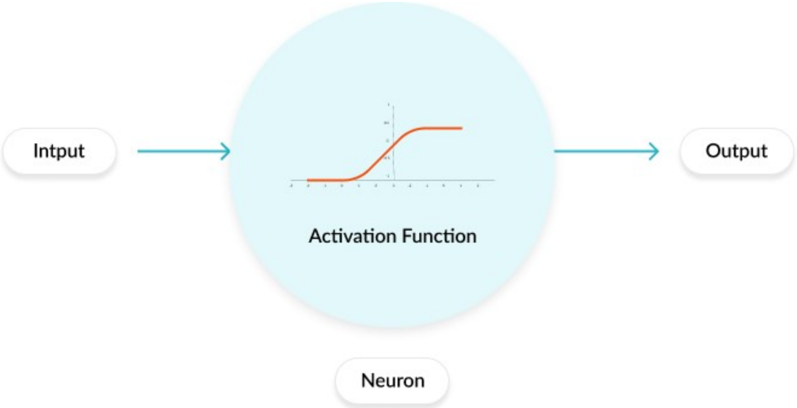


Figure 2.8: Illustration of an activation function in the Neural Network [11]

A - Binary Step Function:

It is a threshold-based activation function. If the input value is greater or less than a certain threshold, the activation of the neuron is on and it sends exactly the same signal to the next layer. The only disadvantage is that it does not allow multi-value outputs : it cannot support classifying the inputs into one of several categories [10].

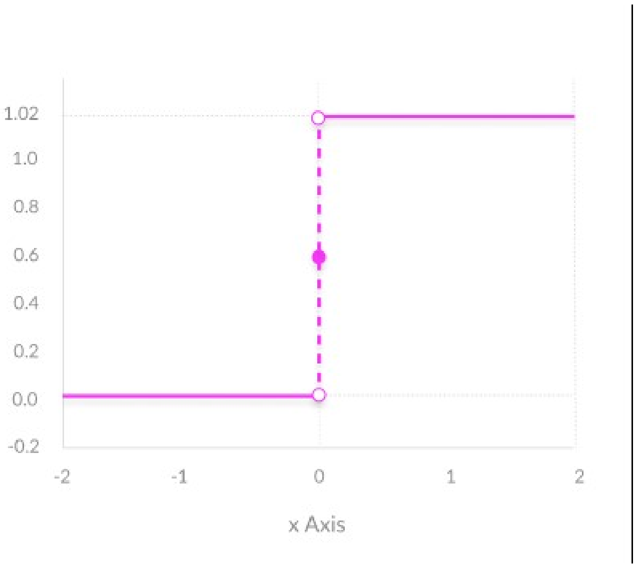


Figure 2.9: Graph of the Binary Step Function [11]

B - Linear Function:

The form of a linear activation function is : $A = cx$. Inputs, multiplied by the weights for each neuron, are taken to create an output signal proportional to the input. This is why, a linear function is better than the step function because it allows multiple outputs, not just yes and no or "1 and 0".

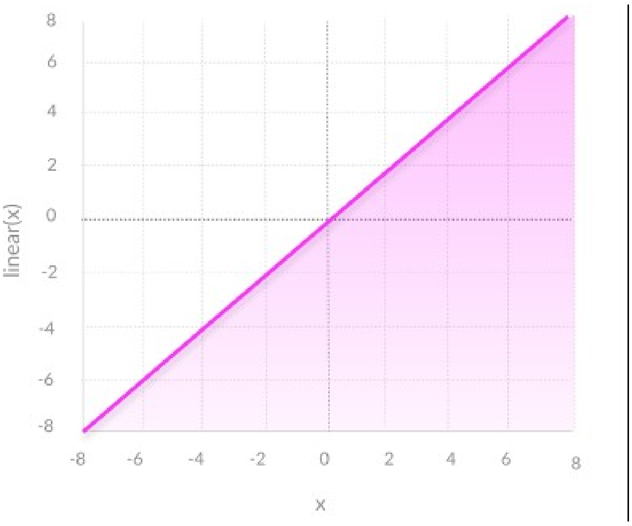


Figure 2.10: Graph of the Linear Function [11].

C - Non-Linear Functions:

Nowdays, non-linear activation functions are the most used functions by neural network models. The Non-Linear Function allows to create complex mappings between the network's (inputs and outputs), which are essential for learning and modeling complex data, such as images, video, audio which are non-linear or have high dimensionality. Almost every process that you can imagine is represented as a functional computation in a neural network, provided that the activation function is non-linear.

1-SIGMOID / LOGISTIC

It has Smooth gradient, preventing "jumps" in output values which bound between 0 and 1, normalizing the output of each neuron. Its clear predictions **For X above 2 or below -2** , tends to bring the prediction to the edge of the curve, very close to 1 or 0. This enables clear predictions. But for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction. It is also computationally expensive.

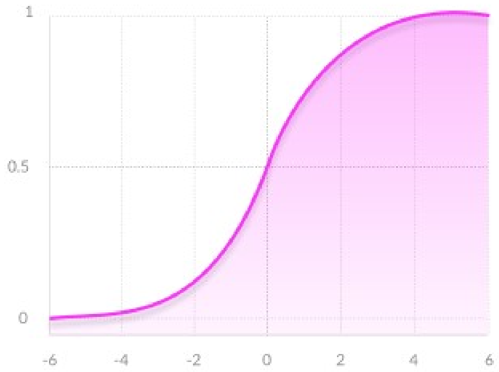


Figure 2.11: Graph of Sigmoid Function [11].

2-TANH / HYPERBOLIC TANGENT

It has the *zero centered* , making it easier to model inputs that have strongly negative, neutral, and strongly positive values.But it is very similar to the sigmoid function this is why it doesn't help too much.

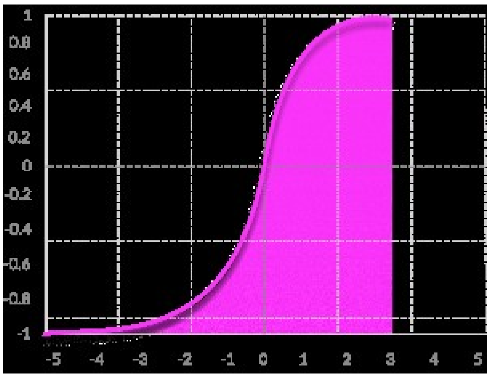


Figure 2.12: Graph of Hyperbolic Tangent Function [11].

3-RELU (RECTIFIED LINEAR UNIT)

The ReLU is computationally efficient because it allows the network to converge very quickly.It looks like a linear function but it's not , because we are talking about Non-Linear functions, remember ? It also has a derivative function which allows for backpropagation which we will talk about later on. But when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.

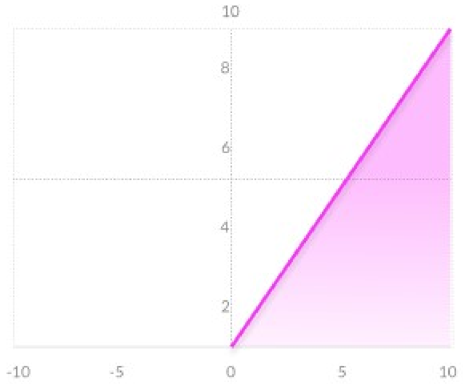


Figure 2.13: Graph of the ReLU function [11].

4-SWISH

The reaserchers at Google discovered a new function called Swish that performs better than ReLU with a similar level of computational efficiency. In experiments on ImageNet with identical models running ReLU and Swish, the new function achieved a higher accuracy with 0.6-0.9% difference.

$$\alpha(x) = \frac{x}{1+e^{-x}}[11]$$

Figure 2.14: Graph of the Swish Function [11].

5-SOFTMAX

The Softmax is a function able to handle multiple, it normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class. It is known for being used only for the output layer, for neural networks that need to classify inputs into multiple categories.

2.5 Training a Neural Network:

the neuron count has risen over the years to express more complex models. Layers also have evolved from each layer being fully connected in multilayer networks to locally connected patches of neurons between layers.There are four fundamental network architectures [12]:

- **U**nsupervised Pre-trained Networks
- **C**onvolutional Neural Networks
- **R**ecurrent Neural Networks
- **R**ecursive Neural Networks.

2.5.1 Back-Propagation:

It is simply the partial derivative of a function (Gradient), it is difficult to solve the neuron weights of a multi-layer network analytically. The back-propagation algorithm provides a simple and effective solution to solving the weights iteratively.The classical version uses gradient descent as optimization method.

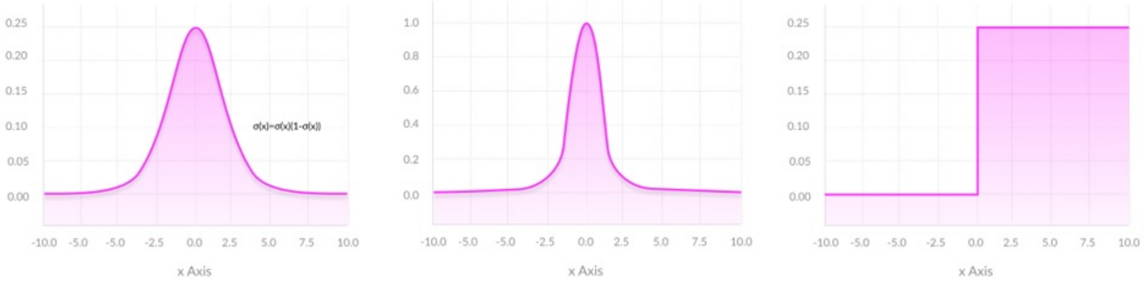


Figure 2.15: Graphs of Sigmoid, Tanh and ReLU Derivatives used for Back-propagation Method.

2.5.2 Stochastic Gradient Descent

The word ‘stochastic‘ means a system or a process that is linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. SGD uses only a single sample to perform each iteration which solves the problem of the Gradient Descent. The sample is randomly shuffled and selected for performing the iteration.

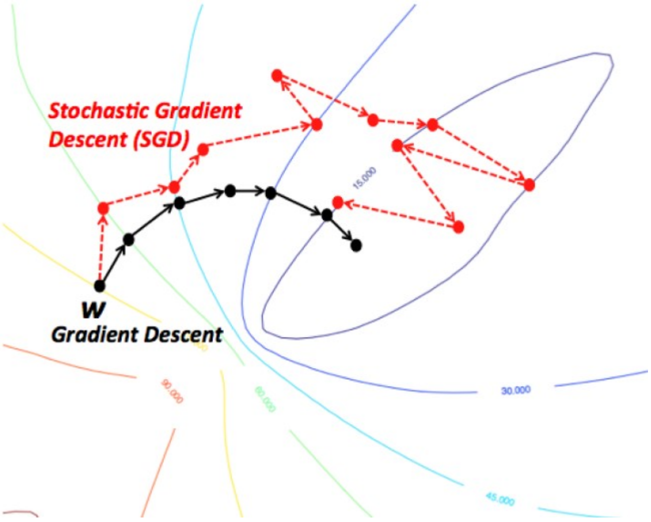
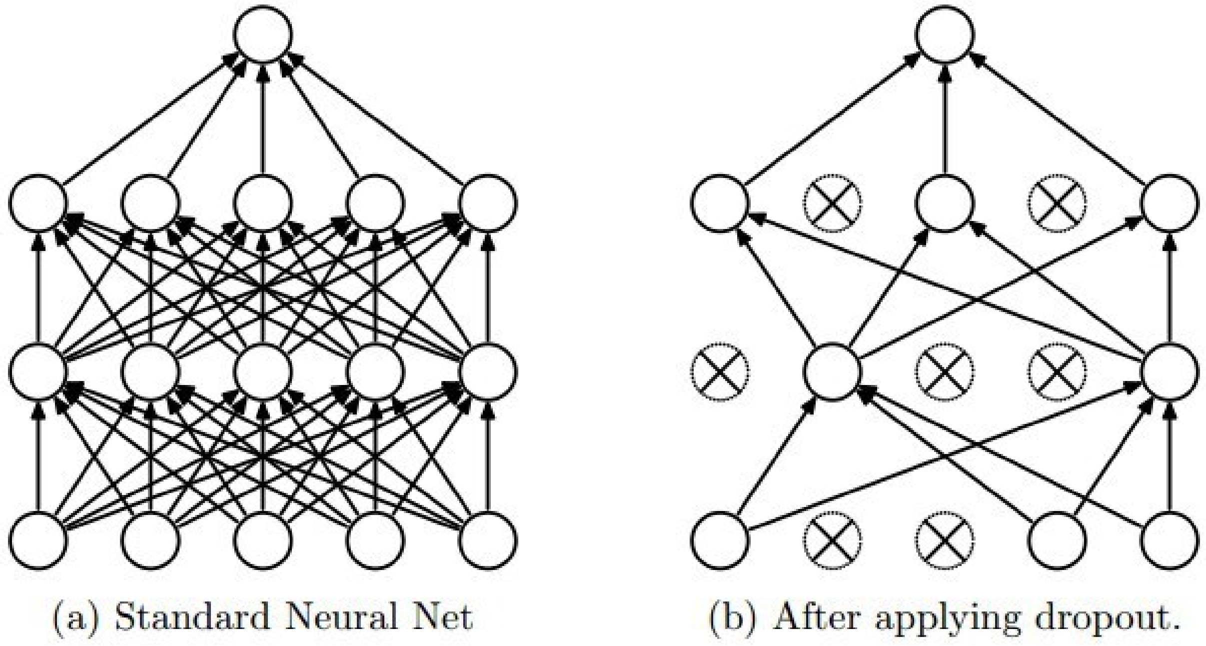


Figure 2.16: a Graph that shows how SGD solves the problem of GD [13]

2.5.3 Dropout

The key idea is to randomly drop units (along with their connections) from the neural network during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. This prevents units from co-adapting too much.



(a) Standard Neural Net

(b) After applying dropout.

Figure 2.17: A neural network with dropout during training [14]

2.5.4 Max Pooling

Max pooling is done by applying a max filter to usually non-overlapping subregions of the initial representation. The object is to down-sample an input representation, reducing its dimensionality.

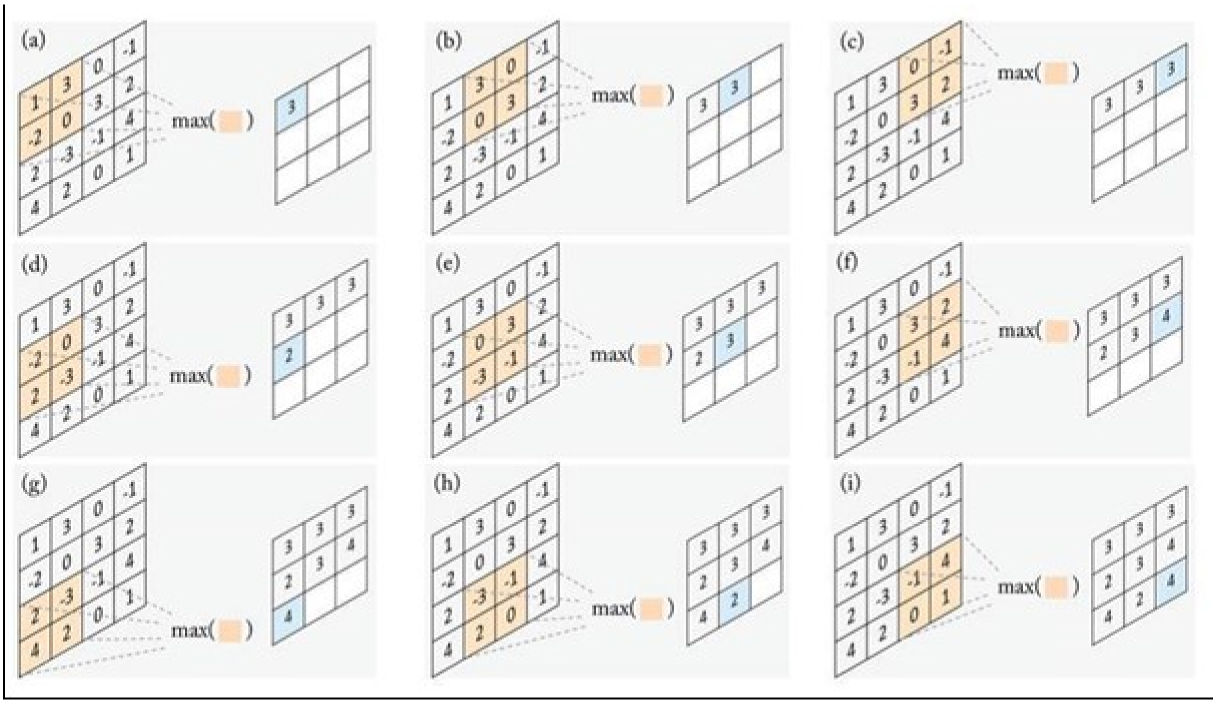


Figure 2.18: The operation of max-pooling layer [14]

2.5.5 Batch Normalization

Batch normalization is a method we can use to normalize the inputs of each layer, in order to fight the internal covariate shift problem. Also, batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers.

Step1: Calculate the mean and variance of the layers input :

$$ub = \frac{1}{m}\sum_{i=1}^{m} XiBatchMean \tag{2.3}$$

$$\sigma^2 b = \frac{1}{m}\sum_{i=1}^{m}(Xi - ub)^2 BatchVariance \tag{2.4}$$

Step 2 : Normalize the layer inputs using the previously calculated batch statistics.

$$\overline{Xi} = \frac{Xi - ub}{\sqrt{\sigma^2 b + \epsilon}} \tag{2.5}$$

Step 3 : Scale and shift in order to obtain the output of the layer.

$$Yi = \gamma\overline{Xi} + \beta \tag{2.6}$$

(γ and β are learned during training)

2.5.6 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

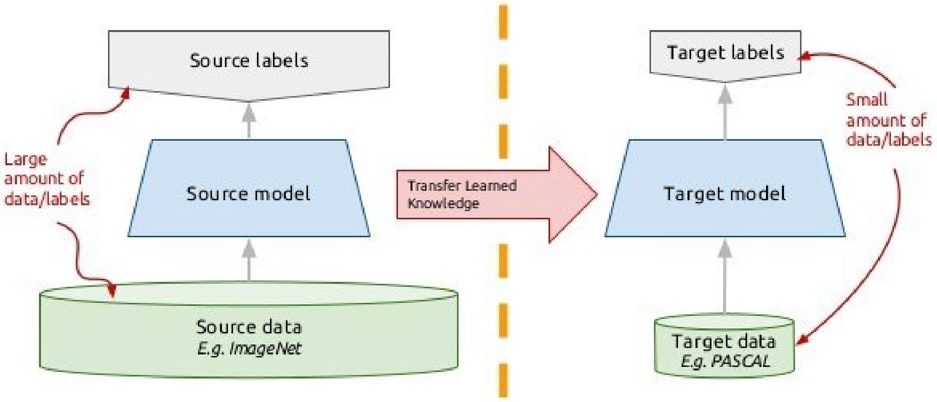


Figure 2.19: The Idea of Transfer Learning [15]

2.6 Convolutional Neural Networks (CNN):

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other [16].

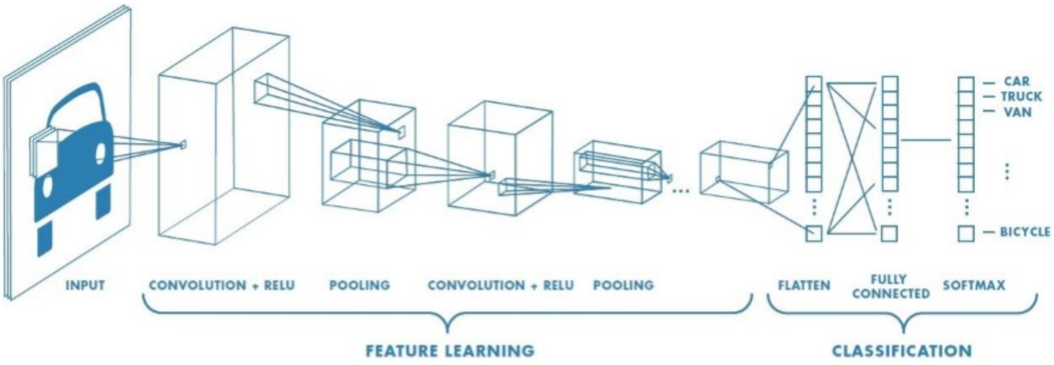


Figure 2.20: Architecture of Convolutional Neural Network in Car Recognition [16]

2.6.1 components of a convolutional neural network::

Input Image:

In the figure, we have an RGB image which has been separated by its three color planes. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

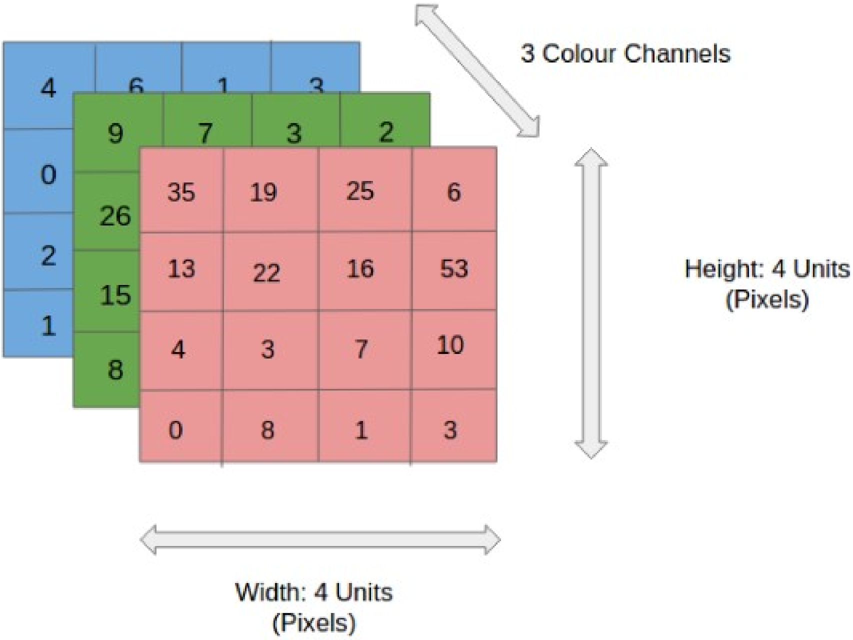


Figure 2.21: 4x4x3 RGB Image [16]

Convolution Layer -The Kernel:

Will take images from the preceding layers and convolve with them the specified number of filters to create images called output feature maps. The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer.

Pooling Layer :

imilar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. There are two types of Pooling: -**M**ax Pooling **AND A**verage Pooling.

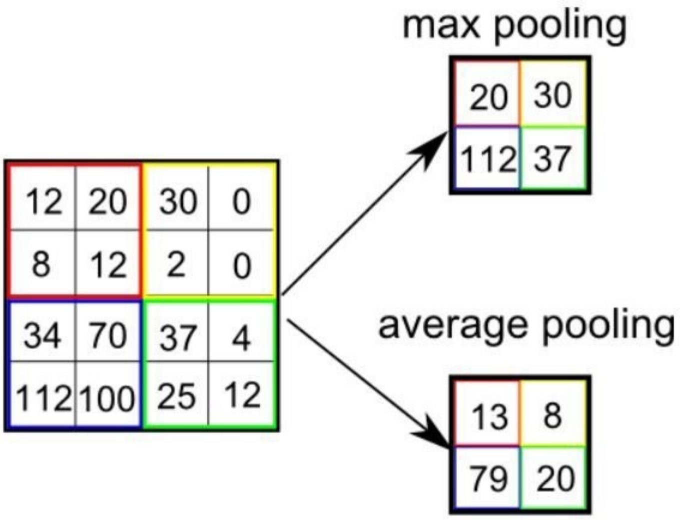


Figure 2.22: 4x4x3 Explanation of Max and Average Pooling [16]

2.6.2 Classification - Fully Connected Layer :

In a typical CNN, full-connected layers are usually placed toward the end of the architecture. Adding a Fully-Connected layer is a cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer.

Softmax :

A special kind of activation layer, usually at the end of FC layer outputs, it can be viewed as a fancy normalizer which produces a discrete probability distribution vector and is very convenient when combined with cross-entropy loss.

2.7 Pre-Trained Models:

Pre-trained models are Neural Network models trained on large benchmark datasets like ImageNet. The Deep Learning community has greatly benefitted from these open-source models and this is also one of the major reasons for rapid advances in Computer Vision research. Due to time restrictions or computational restraints, it's not always possible to build a model from scratch which is why pretrained models exist. In Our experimental part we will focus on six pre-trained models :

2.7.1 Inception v3 :

InceptionV3 is one of the state-of-the-art convolutional neural networks from Google. It's an advanced version of GoogLeNet that won the ImageNet ILSVRC-2014 competition with its out-of-the-box convolutional neural network architecture.

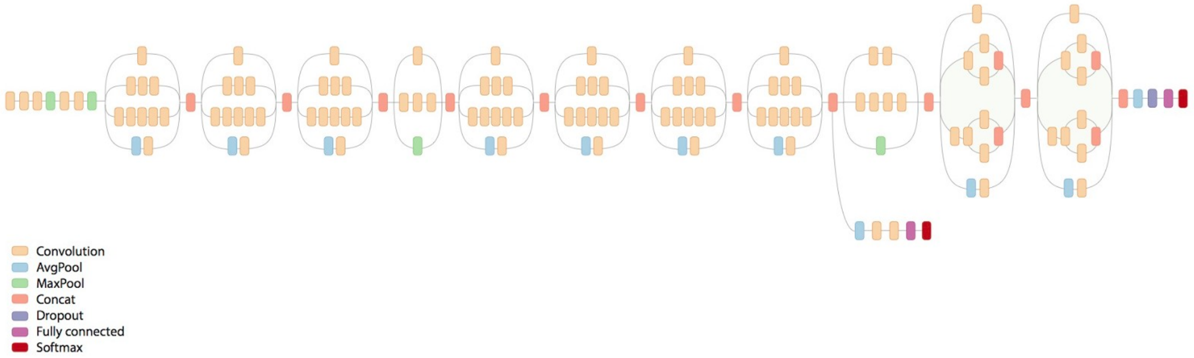


Figure 2.23: The Architecture of Inception V3 [17]

2.7.2 Xception :

This architecture slightly outperforms Inception V3 on the ImageNet dataset and since the Xception architecture has the same number of parameters as Inception V3, the performance gains are not due to increased capacity but rather to a more efficient use of model parameters.

2.7.3 VGG16 :

VGG16 is a convolutional neural network model (16 layers deep) proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet.

2.7.4 VGG19 :

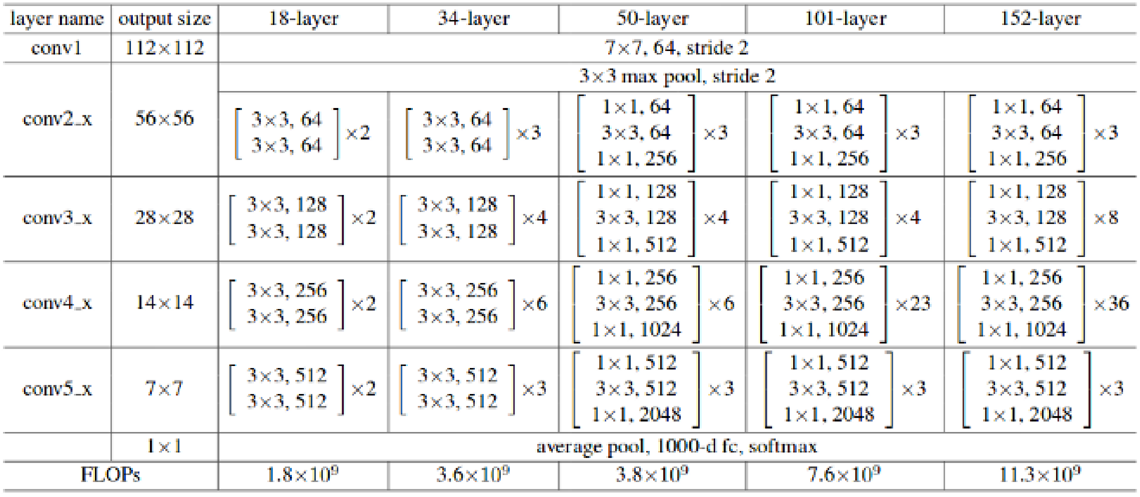
VGG-19 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 19 layers deep and can classify images into 1000 object categories. It was used by the VGG team in the ILSVRC-2014 competition.

2.7.5 ResNet :

ResNet is a short name for Residual Network. On the ImageNet dataset researchers evaluate residual nets with a depth of up to 152 layers—8x deeper than VGG nets but still having lower complexity. The Network won 1st place in the ILSVRC 2015 classification competition with top-5 error rate of 3.57

ResNet-50 is 50 layers deep and can classify images into 1000 object categories, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

ResNet-101 is 101 layers deep and can classify images into 1000 object categories, replacing VGG-16 layers in Faster R-CNN with ResNet-101. They observed a relative improvements of 28



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 2.2: ResNet Architecture [17]

2.8 Summary

In this chapter, we discussed the theory of the ANN. We did the comparison between ANN and biological neurons to understand how it was inspired from the human brain. We presented the perceptron in its simple and complex forms, we also explained several activation functions and the different ways to train a network before getting around to CNNs and pre-trained models.

Chapter 3

Experiments and Results

3.1 Introduction:

In this chapter, we will focus on the implementation of the melanoma identification using convolutional neural networks (Deep learning). The implementation process passes through the training, the validation and the testing phases using a dataset that contains melanoma and non-melanoma images. We interpreted this whole process into a Python code followed by its results and described in details in the following sections. In addition to that, we will work on changing some parameters and pre-trained models to see their effect on the performance of our CNN.

3.2 Evaluation metrics:

When evaluating clinical tests in medicine, there are mainly four common evaluative metrics, which are sensitivity, specificity, accuracy and loss function. Sensitivity measures the proportion of actual positives that are correctly identified (disease detected) while specificity measures the proportion of actual negatives that are correctly identified (disease not detected)[18]. Accuracy measures the proportion of the true diagnostic that is correctly identified, i.e. it shows the extent at which the test is good while the loss function evaluates the difference between the output and the target variable given an input and a target. These metrics are calculated according to equations (3.1), (3.2), (3.3), (3.4) and (3.5).

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}[18] \tag{3.1}$$

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive}[18] \tag{3.2}$$

$$Accuracy = \frac{TrueNegative + TruePositive}{TrueNegative + TruePositive + FalsePositive + FalseNegative}[18] \tag{3.3}$$

The Loss function falls under two major function measures:

3.2.1 Mean Squared Error (MSE):

MSE is a measure of the average of the squares between the actual observations and those predicted as described by (3.4).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2[18] \tag{3.4}$$

In the case of neural networks the predicted values ŷi are the outputs of the final layer and the true values yi the output of the modelled function, N being the number of training samples.

3.2.2 Cross Entropy:

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. The cross-entropy loss can be calculated by Eq 3.5

$$Crsso\_Entropy = -\frac{1}{N}\sum_{i=1}^{N} y \log \hat{y} - (1-y)\log(1-\hat{y})[18] \tag{3.5}$$

3.3 Tools:

The algorithm of our code was written in Python language with Keras library using Python software version 3.6.0 and Pycharm software as IDE 2019.1.1 edition (Integrated development environment).

3.3.1 Python:

Python is an open source interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python 3.7.3 is the newest major release of the Python language, and it contains many new features and optimizations [19].

Figure 3.1: Python logo

3.3.2 Keras:

Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano (Keras, 2017). It was developed with the purpose of enabling fast experimentation. Providing results with the least possible delay, Keras constitutes a key for good research. It is designed on the following properties: modularity, so many functions, graphs and neural layers to create new models when combined together, minimalism, that it is important each module to be short and simple and easy extensibility, the ability to add new models [20].

3.3.3 PyCharm:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains . It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda . PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license [21].

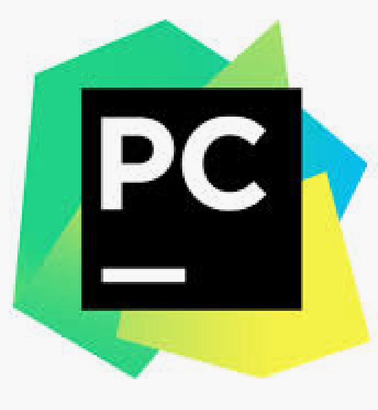


Figure 3.2: PyCharm

3.4 Hyper-parameters:

Hyperparameters are varaibles that we need to set before applying a learning algorithm to a dataset. The challenge with hyperparameters is that there are no magic number that works everywhere. The best numbers depend on each task and each dataset. Hyperparameters can be divided into 2 categories:

1. Optimizer hyperparameters
2. Model Specific hyperparameters

3.4.1 Optimizer Hyper parameters:

They are related more to the optimization and training process

1.Learning rate: designs for the amount that the weights are updated during training. If our learning rate is too small than optimal value then it would take a much longer time (hundreds or thousands) of epochs to reach the ideal state Or, on the other hand, if our learning rate is too large than optimal value then it would overshoot the ideal state and our algorithm might not converge

Figure 3.3: Illustration of error curves (Error vs. Weights) [22]

2.Batch size: defines the number of samples (training examples) that propagate through the network in one iteration. The batch size can be one of three options:

a. **batch mode:** where the batch size is equal to the total dataset thus making the iteration and epoch values equivalent

b. **mini-batch mode:** where the batch size is greater than one but less than the total dataset size. Usually, a number that can be divided into the total dataset size.

c. **stochastic mode:** where the batch size is equal to one. Therefore the gradient and the neural network parameters are updated after each sample.

3.Number of epochs: A single epoch is a single step in training a neural network; in other words when a neural network is trained on every training samples only in one pass we say that one epoch is finished.

3.4.2 Model specific Hyper-parameters:

1. Number of hidden units: is the number of hidden neurons that should be between the size of the input layer and the size of the output layer. The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer. The number of hidden neurons should be less than twice the size of the input layer.

2. **Number of convolutional (hidden) layers:** is the number of the layers in between input layers and output layers, where artificial neurons take in a set of weighted inputs and produce an output through an activation function.

3.5 Experiment and results:

3.5.1 Part 1: CNN Model based on the number of layers and data augmentation:

Experiment:

We built a CNN model using Keras in an algorithm using Python language for the training of the dataset to predict later if the given image represents a melanoma or a non-melanoma.with a fixed learning rate equal to 0.01 and batch size of 200 because these two values of hyper-parameters gave the best results in the work that was done in 2018 in which the number of layers was fixed to 4. In our first case, we used only one hidden layer and applied the data augmentation. Then, we added more layers to see the effect of the number of layers on the performance of our model while we kept applying the data augmentation.

Data structure:

Before we start building the model, we had needed to download all the required training and test dataset, we downloaded both the folders named “ test_set” and “training_set” into our working directory, it took us a while as there are more than 10,000 images in both folders, which is the training data as well as the test dataset. The folder “**training_set**” contains two sub folders **melanoma** and **non-melanoma**, each holding 1780 images of the respective category. Second, the folder “**test_set**” contains two sub folders **melanoma** and **non-melanoma**, each holding 5341 images of respective category.

Convolutional Neural Network is process that always involves four main steps:

Step - 1 : Convolution

Step - 2 : Pooling

Step - 3 : Flattening

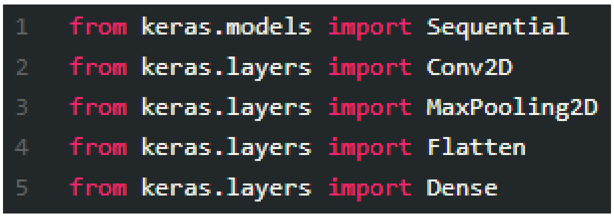
Step - 4 : Full connection

We went through each of the above operations while coding our neural network.

Coding part:

Experiment 1: CNN with one convolutional layer:

First, we imported all the needed Keras libraries and packages which we used later to build our CNN, we confirmed that every package is installed perfectly, there are two ways of using Keras: using Tensorflow backend and by using Theano backend. We coded using Tensorflow backend.



```
1   from keras.models import Sequential
2   from keras.layers import Conv2D
3   from keras.layers import MaxPooling2D
4   from keras.layers import Flatten
5   from keras.layers import Dense
```

In line 1, we imported Sequential from Keras.models, to set our neural network model as a sequential network.

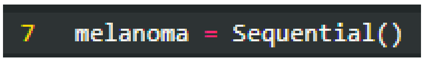
In line 2, we imported Conv2D from Keras.layers, this is to perform the convolution operation i.e. the first step of a CNN, on the training images. Since we are working on images here, which are basically 2 Dimensional arrays, we are using Convolution 2-D.

In line 3, we imported MaxPooling2D from Keras.layers, which is used for pooling operation, that is the second step in the process of building a CNN. For building this particular neural network, we used a Maxpooling function, there exist different types of pooling operations like Min Pooling, Mean Pooling, etc. Here in MaxPooling, we need the maximum value pixel from the respective region of interest.

In line 4, we imported Flatten from Keras.layers, which is used for Flattening. Flattening is the process of converting all the resultant 2-dimensional arrays into a single long continuous linear vector.

And finally, in line 5, we imported Dense from Keras.layers, which is used to perform the full connection of the neural network, which is the step 4 in the process of building a CNN.

Then, we created an object of the sequential class below:

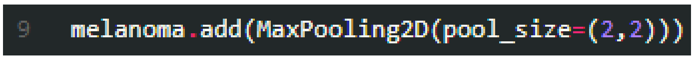


```
7   melanoma = Sequential()
```

Later, we coded the Convolution step as below:

```
melanoma.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
```

We took the object which already has an idea of how our neural network is going to be(Sequential), then we added a convolution layer by using the “Conv2D” function. The Conv2D function is taking 4 arguments, the first is the number of filters i.e. 32 here, the second argument is the shape each filter which is 3x3 here, the third is the input shape and the type of image(RGB or Black and White) of each image which is a 64x64 resolution and “3” stands for RGB, which is a color image, the fourth argument is the activation function we wanted to use, here ‘relu’that stands for a rectifier function. Then, we performed pooling operation on the resultant feature maps that we got after the convolution operation was done on an image. The primary aim of a pooling operation is to reduce the size of the images as much as possible. The key thing to understand here is that we tried to reduce the total number of nodes for the upcoming layers.



```
melanoma.add(MaxPooling2D(pool_size=(2,2)))
```

We started by taking our classifier object and adding the pooling layer. We took a 2x2 matrix to have a minimum pixel loss and get a precise region where the feature are located. Here, We just reduced the complexity of the model without reducing its performance.

▷ **Adding a convolutional layer always requires adding a pooling layer.**

Next, we implemented Dropout by randomly selecting nodes to be dropped out with a given probability of 0.2 (20%) to reduce over fitting.

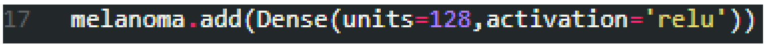
```
melanoma.add(Dropout(0.2))
```

Afterwards, we converted all the pooled images into a continuous vector through Flattening. Flattening here is taking the 2-D array, i.e. pooled image pixels and converting them to a one-dimensional single vector.

```
melanoma.add(Flatten())
```

We have used flatten function to perform flattening, we had no need to add any special parameters, Keras would understand that the “classifier” object is already holding pooled image pixels and they need to be flattened.

In the next step, we created a fully connected layer, and to this layer we connected the set of nodes we got after the flattening step, these nodes act as an input layer to these fully-connected layers. As this layer will be present between the input layer and output layer, we can refer to it a hidden layer.



```
melanoma.add(Dense(units=128,activation='relu'))
```

Dense is the function to add a fully connected layer, ‘units’ is where we define the number of nodes that should be present in this hidden layer, these units value will be always between the number of input nodes and the output nodes but choosing the most optimal number of nodes can be achieved only through experimental tries. However, it is a common practice to use a power of 2 and the activation function will be a rectifier function.

Next, we set our output layer, which should contain only one node, as it is binary classification. This single gives us a binary output of either a melanoma or a non-melanoma.

```
melanoma.add(Dense(units=1,activation='sigmoid'))
```

We can observe that the final layer contains only one node, and we used a sigmoid activation function for the final layer because its range is (0, 1) and it can represent the probability of binary class.

After we had completed building our CNN model, we compiled it as it is shown below:

```
melanoma.compile(optimizer='adam(lr=0.01p)', loss='binary_crossentropy',metrics=['accuracy'])
```

From above:

- Optimizer parameter is to choose the stochastic gradient descent algorithm. Here, we used Adam optimizer with a fixed learning rate of 0.01. Adam [23] is an adaptive learning rate optimization algorithm that has been designed specifically for training deep neural networks by computing individual learning rates for different parameters.
- Loss parameter is to choose the loss function.
- Finally, the metrics parameter is to choose the performance metric (The accuracy).

Following that, we fit our CNN to the image dataset that we downloaded. Nevertheless, before we did that, we pre-processed the images to prevent over-fitting. Over fitting is when we get a great training accuracy and a very poor test accuracy due to over fitting of nodes from one layer to another. So before we fit our images to the neural network, we had performed some “image augmentations” on them, which is basically synthesising the training data. We used Keras.preprocessing library for doing the synthesising part as well as to prepare the training set as well as the test set of images that are present in a properly structured directories, where the directory’s name is taken as the label of all the images present in it. For example: All the images inside the ‘Melanom’ named folder will be considered as ‘Melanoma’ by Keras. The second and the third line below are used for “data augmentation” :

- **class_mode:** one of ”categorical”, ”binary”, ”sparse”, ”input”, ”other” or None. Like in our case, class_mode=’binary’ means 1D numpy array of binary labels.
- **Shear_range**: Float. Shear Intensity (Shear angle in counter-clockwise direction in degrees)
- **zoom_range:** Float or [lower, upper]. Range for random zoom.
- **horizontal_flip:** Boolean. Randomly flip inputs horizontally
- **target_size:** tuple of integers (height, width). The dimensions to which all images found will be resized.
- **batch_size:** size of the batches of data. Here, it was set equal to 200, i.e. the model weights will be updated after each batch of 200 samples.
- **class_mode:** one of ”categorical”, ”binary”, ”sparse”, ”input”, ”other” or None. Like in our case, class_mode=’binary’ means 1D numpy array of binary labels.

Therefore, in general, what we tried to do above is creating synthetic data out of the same images by performing different type of operations on these images like flipping, rotating, blurring, etc. (Data augmentation).

Thereupon, fit the data to our model:

In the above code, ‘steps_per_epoch’[1] And ‘epochs’, A single epoch is a single step in training a neural network; in other words when a neural network is trained on every

---

[1] Steps_per_epoch holds the number of training images, i.e. the number of images the training_set folder contains.

training samples only in one pass we say that one epoch is finished. So training process should consist more than one epoch. In this case we have defined 25 epochs validation_data are the data used to help detect over-fitting and to assist in hyper-parameter search. In our case, we made the validation dataset equivalent to the test dataset, which is used to measure the performance of the model since we do not have the validation data; that means that the test dataset played two roles. Validation_steps: holds the number of testing (and validating) images in the test_set folder.

Here after, we made the predictions from our trained model and calculated the metrics (Accuracy, specificity and sensitivity) that we need for our results study:

"**Numpy**" is an open source Python package for scientific and mathematical computing of data and it supports large, multidimensional arrays and matrices. "**Glob**" finds all the pathnames matching a specified pattern. The test_image holds the image that needs to be tested on the CNN and 'image.load_img' is used for importing and loading the images from our folder "dataset". Once we got the test image, we prepared the image to be sent into the model by converting its resolution to 64x64 as the model only except that resolution. Then we used predict () method on our classifier object to get the prediction. As the prediction is in binary form in order to receive either a 1 or 0, which represent a Non-melanoma or a melanoma respectively. tn stands for true negative, fp stands for false positive, tp stands for true positive and fn stands for false negative. After that, we calculated the metrics (Accuracy, sensitivity and specificity) using the equations written in the "**Evaluation metrics**" section as below:

The table below shows the obtained results after working with only one convolutional layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	58.80%	56.16%	65.88%	51.64%
With data augmentation	64.94%	50.02%	69.25%	57.53%

Table 3.1: Training results for 1 hidden layer

Experiment 2: CNN with 2 convolutional layers:

The code is the same we just add another layer in addition to the previous layer with same features. Here, we increased the probability of dropout because we have more layers.

The table below shows the results after the addition of the second layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	66.04%	48.71%	72.70%	60.45%
With data augmentation	73.17%	41.86%	79.03%	68.26%

Table 3.2: Training results for 2 hidden layers.

We observe here that accuracy got higher than it was with 2 convolutional layers, at the same time the loss decreased.

Experiment 3: CNN with 3 convolutional layers:

Still everything is the same, we just added a fourth convolutional layer and we increased the probability of the dropout layer to prevent any more over fitting.

The table below shows the results obtained after the addition of the third layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	79.05%	35.47%	81.55%	72.58%
With data augmentation	87.08%	29.90%	89.25%	79.77%

Table 3.3: Training results for 1 hidden layer

Experiment 4: CNN with 4 convolutional layers:

We added a fifth convolutional layer. We set the dropout probability to 0.35.

The table below shows the results obtained after adding the fourth layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	88.13%	26.52%	91.33%	86.39%
With data augmentation	91.84%	23.05%	94.65%	89.48%

Table 3.4: Training results for 4 hidden layers.

Again, we see that the accuracy increased and loss decreased.

Experiment 5: CNN with 5 convolutional layers:

We added a sixth layer. The dropout was set to a probability of 0.4.

The table below shows the results after the addition of the fifth layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	91.75%	23.21%	92.95%	88.84%
With data augmentation	95.09%	19.97%	95.53%	92.61%

Table 3.5: Training results for 5 hidden layers.

Experiment 6: CNN with 6 convolutional layers:

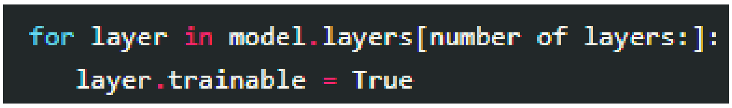
The table below shows the results obtained after the addition of the sixth layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	91.98%	21.98%	93.19%	89.05%
With data augmentation	95.66%	19.30%	96.43%	94.48%

Table 3.6: Training results for 6 hidden layers.

Experiment 7: CNN with 7 convolutional layers:

▷ **We can indicate the number of layers to avoid repetition using this algorithmic technique:**



```
for layer in model.layers[number of layers:]:
    layer.trainable = True
```

The table below shows the results obtained after the addition of the seventh layer :

	Accuracy	**Loss**	**Specificity**	**Sensitivity**
Without data augmentation	92.36%	21.64%	94.25%	89.99%
With data augmentation	96.51%	18.45%	96.91%	94.80%

Table 3.7: Training results for 7 hidden layers.

Graphical Analysis:

The following figures represent the obtained results by varying the number of hidden layers:

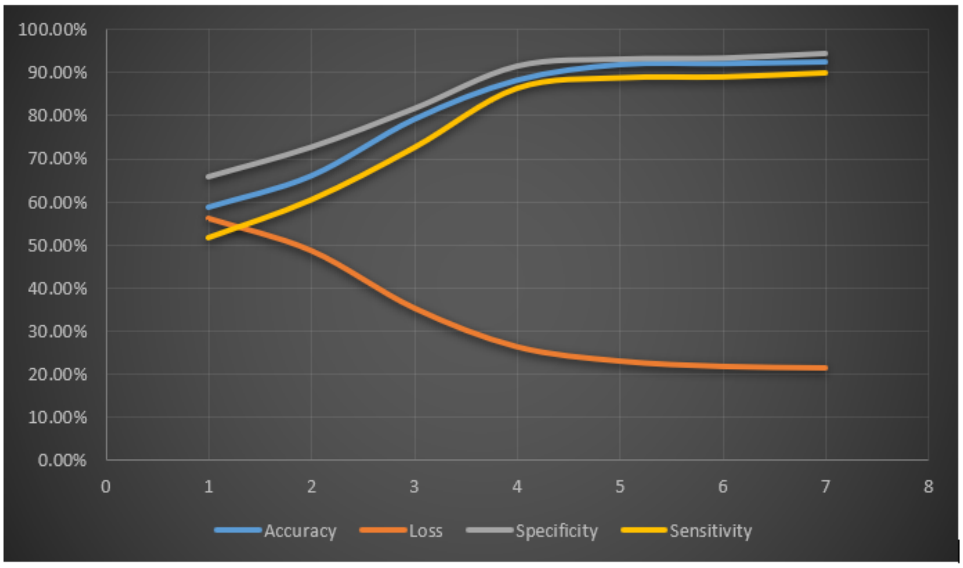


Figure 3.4: Graphical training results for different number of hidden layers without data augmentation

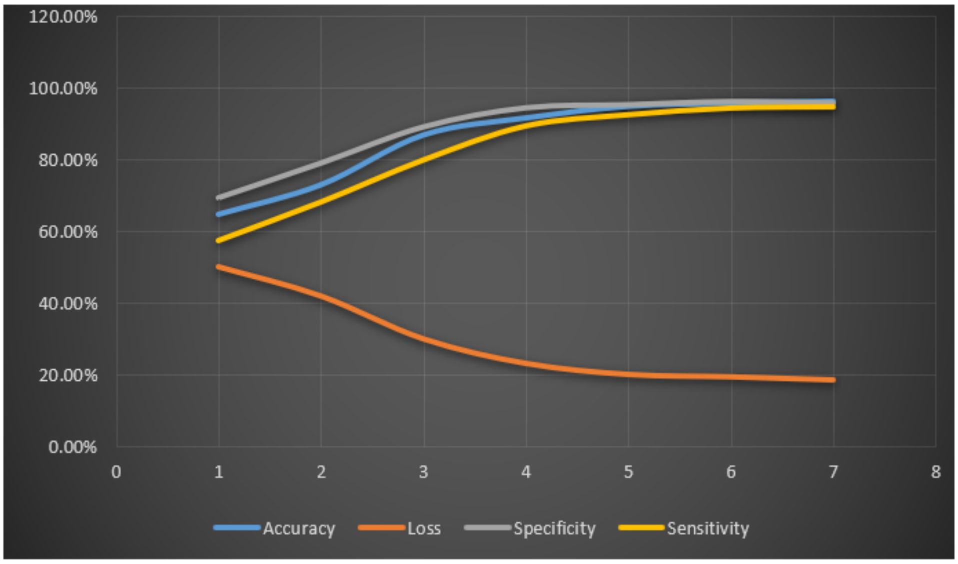


Figure 3.5: Graphical training results for different number of hidden layers with data augmentation

The figure 3.6 summarizes the whole work that we did above. It shows that the convolution layers followed by the pooling layers are responsible of the feature extraction from the data and the fully connected layer is responsible of the classification of the images of the data.

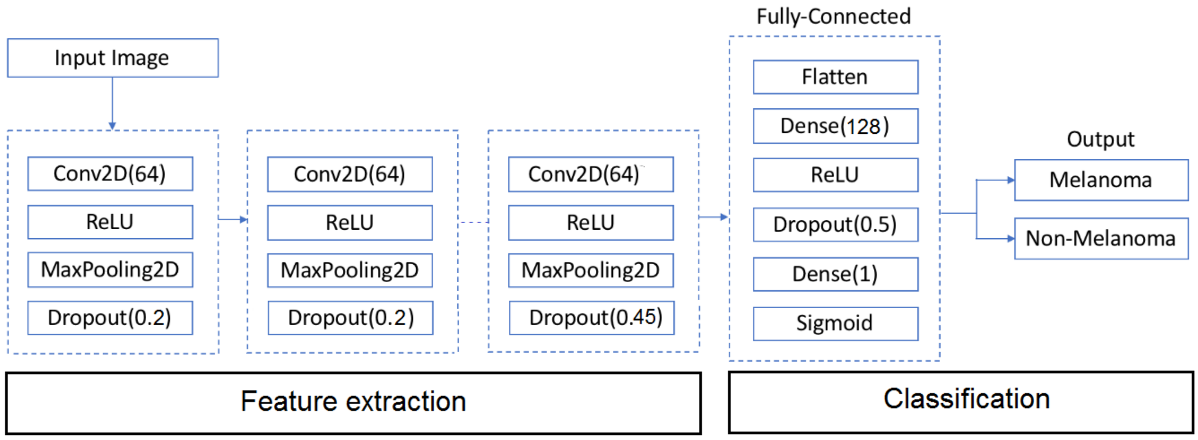


Figure 3.6: Block diagram of the CNN model

Discussion:

Convolutional neural network (CNN) was implemented for the classification of images of the dataset into melanoma and non-melanoma using a specific algorithm with a Python language and Keras library. The work that was done in 2018 was implemented using 4 convolutional layers with a varying learning rate and batch size till the most accurate results were reached at 0.01 and 200 respectively (91.41%). In the work that was done in 2019, the batch size and the learning rate were fixed to 200 and 0.01 respectively since they gave the best results in the work 2018, and we started building our network starting from one convolutional layer to study its effect on the CNN model; the accuracy, the specificity and the sensitivity increased while the lost decreased, this happened gradually. At the fourth layer, the accuracy was found 88.13%, which means it is less than the one that was found with 4 convolutional layers in the work 2018, that is due to some reasons like: the over fitting even though we included the dropout operation in our algorithm to reduce over fitting not to completely omit it, the dataset is not the same that was used in both works, that means that the extracted features are not the same, consequently, that creates a contrast and nuance in the performance of the model, even the learning time can be a reason of that(we ignore the time the learning took in the work 2018; it just can be a possibility). However, once we added a fifth convolutional layer to our CNN model, we got a higher accuracy rate of 92.05%. We carried on adding more layers until the seventh layer because the accuracy kept increasing with a slight percentage that can be negligible (the graphical training results Figure3.4 and Fgiure3.5 Confirm this) because of the fixed rates of batch size and learning rate but we obtained satisfying results regarding the performance of our model that reached an accuracy of 93.36% (Table3.7). So we could notice that number of hidden layers as a hyper-parameter plays an important role in affecting the CNN model without forgetting the fact that data augmentation so much helped us reach higher accurate results, from the tables (from table 3.1 to table 3.7) we can observe that each time we add a new layer and apply the data augmentation, the accuracy rate increases.

3.5.2 Part2: Transfer learning:

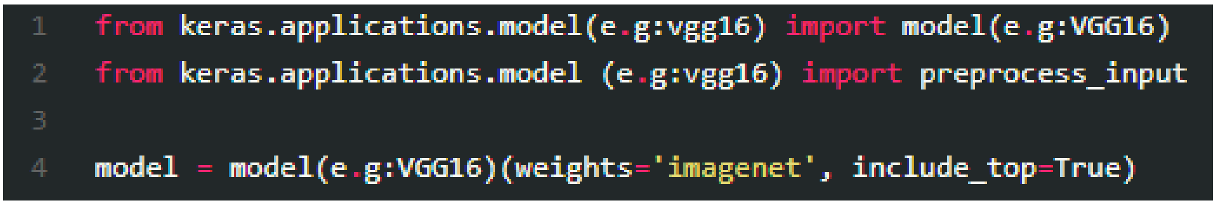
After we had worked on the hyper-parameters of our model, we moved to improving the performance using some pre-trained models whose weights are already pre-trained on ImageNet.

ImageNet:

The ImageNet project is a large visual database designed for use in visual object recognition software research. More than 14 million images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories] with a typical category, such as "balloon" or "strawberry", consisting of several hundred images. The database of annotations of third-party image URLs is freely available directly from ImageNet, though the actual images are not owned by ImageNet.[Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes. The challenge uses a "trimmed" list of one thousand non-overlapping classes.[24]

Experiment:

We went through some pre-trained models; based on their characteristics each mentioned in Chapter 2, we followed the same algorithm we had implemented with the hyper-parameters, i.e. we wrote the Python codes as we did with the previous classification. In addition to that, this is what we did to distinguish between every model using Keras:



```
from keras.applications.model(e.g:vgg16) import model(e.g:VGG16)
from keras.applications.model (e.g:vgg16) import preprocess_input

model = model(e.g:VGG16)(weights='imagenet', include_top=True)
```

- The first two lines are used to import the pre-trained model
- “weights=’imagenet’”is used to import the weights concerning this model from ImageNet databes.
- “Include_top=True” means that the final dense layer (fully connected layer) is kept because dense layers are the ones which are capable of interpreting the found patterns to classify the images into their appropriate categories.

After using the pre-trained models with Python language, we found the following results:

Model	**Accuracy**	**Loss**	**Specificity**	**Sensitivity**
VGG16	71.38%	43.58%	79.98%	62.68%
VGG19	73.60%	41.36%	78.47%	64.69%
Resnet50	77.93%	37.03%	82.94%	79.91%
ResNet101	79.49%	35.47%	83.56%	82.54%
InceptionV3	82.95%	32.01%	87.25%	84.85%
Xception	89.09%	25.87%	93.66%	86.43%

Table 3.8: Training results using some pre-trained models

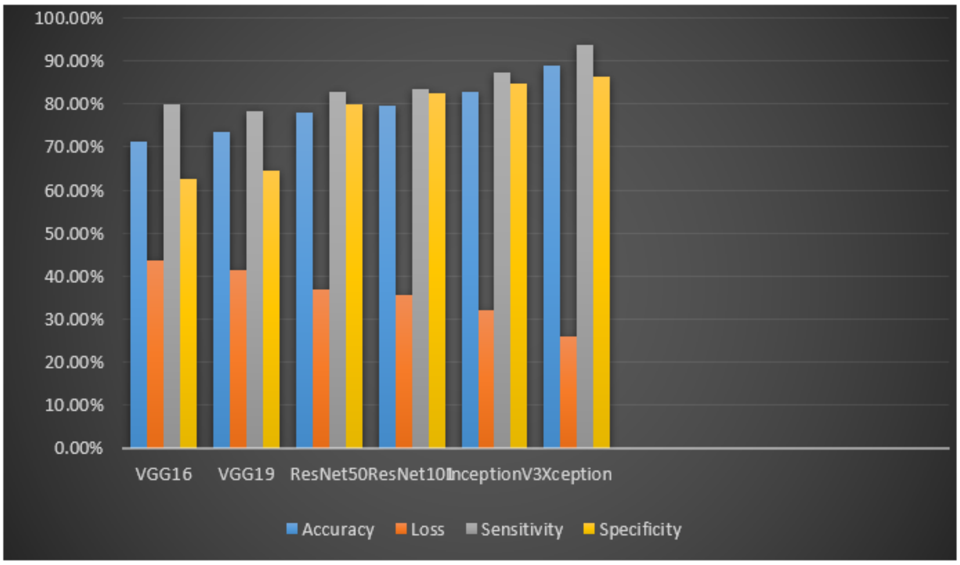


Figure 3.7: Graphical results using some pre-trained models

Discussion:

We re-studied the performance of our CNN model using Transfer learning that involves some pre-trained models, which are VGG16, VGG19, ResNet50, ResNet101, InceptionV3 and Xception. After going through each of them using Python and Keras library, their features in the algorithms and obviously their pre-trained weights imported from ImageNet even though it does not include of melanomas and non-melanomas but many features are extracted from the other images that assist in the classification of images, we found out that Xception is the most powerful and accurate model (Table 3.8 ,Figure 3.5 and Figure 3.7) because it replaces the standard Inception modules with depthwise separable convolutions [2]. Hence, this kind of convolutions does less multiplications, which gives the network the ability to process more in a shorter amount of time.

The next table compares sensitivity and specificity rates of clinicians' visual investigations with the work that was done in 2018 and our work.

---

[2] Depthwise separable convolutions work with kernels that cannot be factored into two smaller kernels.They are so named because they deal not just with the spatial dimensions, but with the depth dimension — the number of channels — as well.

Authors	**Methods**	**Sensitivity**	**Specificity**
Argenziano Et al. (2006) [25]	**Dermoscopy** [3]	79.2%	71.8%
Menzies Et al. (2018) [26]	**Dermoscopy**	53.1%	89.0%
Louifi A., Soulami A. and Dr.Cherifi D. (2018) [27]	**CNN model using 4 layers, batch size=200, learning rate=0.01**	87.75%	92.27%
Louifi A., Soulami A. and Dr.Cherifi D. (2018) [27]	**CNN model using InceptionV3**	72.08%	83.7%
Proposed work (2019)	**CNN model using 5 layers, batch size=200, learning rate=0.01**	88.84%	92.95%
Proposed work (2019)	**CNN model using 6 layers, batch size=200, learning rate=0.01**	89.05%	93.19%
Proposed work (2019)	**CNN model using 7 layers, batch size=200, learning rate=0.01**	89.99%	94.25%
Proposed work (2019)	**CNN model using Xception**	86.43	93.66

Table 3.9: Comparison of the specificity and sensitivity of this work in different experiments with clinicians and work 2018

We see that we got better results using layers whose number is more than 4 and using a more powerful and accurate pre-trained model. The non-experienced dermatologists can use the present work as shown globally on table 3.9 as an additional tool to verify the evolution and the stages of the skin lesion.

---

[3] Dermoscopy is a noninvasive method that allows the in vivo evaluation of colors and microstructures of the epidermis, the dermoepidermal junction, and the papillary dermis not visible to the naked eye

3.6 Future work:

In order to improve the performance of our model (accuracy, specificity and sensitivity), there is a research topic which suggests joining and combining the pre-trained models like the ones we used in our work or even another additional ones joined to them and takes their outputs as inputs to max pooling layer whose outputs become inputs to the combiner model which then predicts the final output of the initial dataset, this technique is called “Super deep learning”.

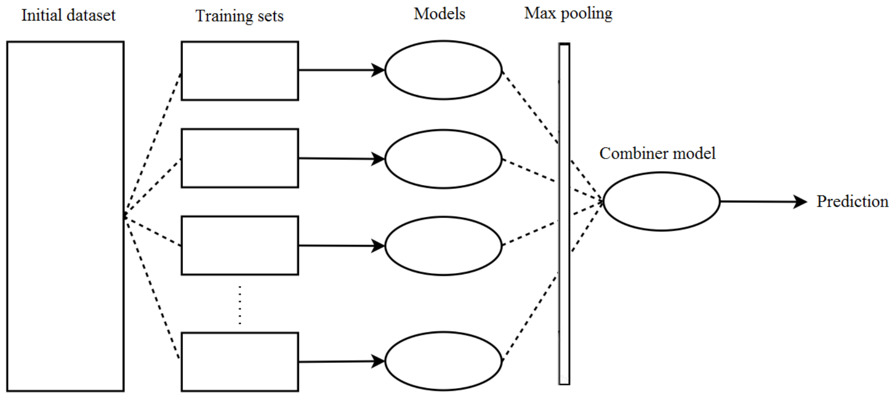


Figure 3.8: Super deep learner architecture.

3.7 Summary:

In the experimental part, we built our CNN model starting from one hidden layer to seven hidden layers using a fixed batch size and learning rate to study the effect of number of layers on the model and found out that this hyper-parameter contributes to improving the performance the more we add layers, at the same time we applied data augmentation technique at each step which helped us getting better results. Then, we used several pre-trained models and obtained their training results and Xception model was the most powerful one among them all.

Conclusion

This project had a purpose of classifying images of melanomas and non-melanomas using convolutional neural network (CNN) model so that it can be put into practice with dermatologists in order to distinguish between benign melanoma and malignant melanoma, thus the dermatologists gain more time in prediction and preserves the healthy safety of patients. CNN is an art of machine learning that is based on binary classification whose accuracy determines the state of the performance of the model with some pre-processing. The hyper-parameter" number of hidden (convolutional) layers" has a huge effect on the performance of the model. The more we increase the number of layers, the more we get a better accuracy, it achieved an accuracy of 92.36%, specificity of 94.25%, sensitivity of 89.99% and a loss of 21.64% while with only one convolutional layer it achieved an accuracy of 58.08%, specificity of 65.88% sensitivity of 51.64% and a loss of 56.16%, these results confirm what was said earlier. Furthermore, the date augmentation is a tuning that extremely take part in improving the performance, this technique was applied at each time a new layer was added, and in fact all the metrics achieved better rates knowing that the batch size was fixed at 200 and the learning rate was fixed at 0.01, these two rates gave the best results in the project that was done in 2018. Transfer learning is another method for improving the performance using some pre-trained models but in this work our the aim was not really to improve the previously found results, in fact the aim was to study deeply every used pre-trained model and see its results. As it was mentioned earlier, pre-trained models have already trained weights on ImageNet however this latter does not include melanoma and non-melanoma images, which means that we did not have the same feature on which we can build model to improve the performance. In general, we compared between the results of the pre-trained models and we found that Xception was the most performing one; it achieved an accuracy of 92.36%, specificity of 94.25%, sensitivity of 89.99% and a loss of 21.64%. Such reliable results allow for self-examination for first diagnosis from individual users.

Bibliography

[1] MA Weinstock-Jama,2000 https://www.skincancer.org/skin-cancer-information/melanoma

[2] Oliveria S, Saraiya M, Geller A, Heneghan M, Jorgensen C (2006). *"Sun exposure and risk of melanoma"*

[3] TL Diepgen, V Mahler - British Journal of Dermatology, 2002http://projects.ncsu.edu/project/bio183de/Black/cellcycle/cellcycle_reading/skin_cancer.php

[4] SM Swetter-The Surgical clinics of North America, 2003 https://www.skincancer.org/skin-cancer-information/melanoma/types-of-melanoma.

[5] V Madan, JT Lear, RM Szeimies - The lancet, 2010 https://www.cancer.net/cancer-types/melanoma/stages

[6] Balch C, Buzaid A, Soong S, Atkins M, Cascinelli N, Coit D, Fleming I, Gershenwald J, Houghton A, Kirkwood J, McMasters K, Mihm M, Morton D, Reintgen D, Ross M, Sober A, Thompson J, Thompson J (2001). *"Final version of the American Joint Committee on Cancer staging system for cutaneous melanoma"*

[7] KL Moozar, CS Wong, J Couture - Canadian journal of surgery, 2003 https://www.curemelanoma.org/patient-eng/melanoma-treatment/

[8] A must-read tutorial when you are starting your journey with Deep Learning . Magdalena Ługowska's blog

[9] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.

[10] SK Pal, S Mitra - IEEE Transactions on neural networks, 1992 - ieeexplore.ieee.org

[11] M Leshno, VY Lin, A Pinkus, S Schocken - Neural networks, 1993 – Elsevier

[12] Akshay Chandra Lagandula. **Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works**

[13] L Bottou - Proceedings of COMPSTAT'2010, 2010 – Springer

[14] N Srivastava, RR Salakhutdinov -Advances in Neural Information Processing Systems 26 (NIPS 2013)

[15] A Gulli, S Pal - 2017 https://engmrk.com/kerasapplication-pre-trained-model/

[16] Y Kim - arXiv preprint arXiv:1408.5882, 2014 https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[17] A Gulli, S Pal - 2017 https://engmrk.com/kerasapplication-pre-trained-model/

[18] Anna Gummeson.*Prostate Cancer Classification using Convolutional Neural Networks,2015.*

[19] Python developers, Python platform, 2015, https://www.python.org/downloads/release/python-360/

[20] Keras, what is Keras ?,2016,https://keras.io/

[21] Pycharm developers, Jetbrains platform, 2017, https://www.jetbrains.com/pycharm/download

[22] J Bergstra, Y Bengio - Journal of Machine Learning Research, 2012, https://towardsdatascience.com/hyperparameters-in-deep-learning-927f7b2084dd, "Hyperparameters in Deep Learning", February 9th 2019, Samarth Agrawal

[23] Diederik P. Kingma and Jimmy Lei Ba. Adam . *A method for stochastic optimization.*

[24] "New computer vision challenge wants to teach robots to see in 3D". New Scientist. 7 April 2017. Retrieved 3 February 2018.

[25] Argenziano et.al, Dermoscopy of pigmented skin lesions: results of a consensus meeting via the Internet, 2006, https://www.ncbi.nlm.nih.gov/pubmed/12734496

[26] Menzies Method. (2018, January 17). dermoscopedia. Retrieved 22:55, July 6, 2019 from https://dermoscopedia.org/w/index.php?title=Menzies$_{M}ethodoldid = 9988$.

[27] Louifi Akram, Soulami Amer, Dr. Cherifi D. "Melanoma identification using CNN", Master thesis, IGEE UMBB, Boumerdes 2018