Abstract

This dissertation deals with the analysis of medical images using deep learning, the medical images being analyzed are images of skin lesions. The goal of this work is to create a classifier, using deep neural networks, capable of distinguishing multiple kinds of skin lesions with a focus on cancerous lesions, which can be used as a valuable screening tool to be used as an early warning system.

We propose and benchmark different neural networks for this task, as well as an ensemble strategy that uses multiple architectures and trains them as a single neural network.

An important factor for this work is that the network should be able to classify not only images obtained by professionals in a clinical setting but also to classify images which are obtained via any RGB camera by the end-users, thus envisioning the possibility of implementing the proposed classification systems in portable devices. To test the potential of the proposed classifiers, multiple neural networks are trained using different techniques such as transfer learning, image pre-processing, and data augmentation, to test the benefits of using these techniques towards the performance of such a system.

This thesis was proposed and sponsored by the technology sector company, Glintt. The proposed solution achieved the following results in the classification of multiple kinds of skin lesions:

- Accuracy: 73.27% ± 1.40
- Sensitivity: 72.92% ± 0.89
- Specificity: 91.1% ± 0.46
- F1-score: 73.20% ± 1.41

Keywords: neural network; skin lesion; data augmentation; neural network architecture; dermoscopic image; convolutional neural network ; skin cancer; transfer learning.

Resumo

Esta dissertação lida com a análise de imagens médicas com recurso a *deep learning*, as imagens medicas a serem analisadas são imagens de lesões de pele. O objetivo deste trabalho é a criação de um classificador que, usando redes neuronais profundas, seja capaz de discriminar diferentes tipos de lesões de pele com um foco em lesões carcinogénicas, que possa ser usado como um sistema de aviso para daignostico precoce.

Para este fim, propõem-se e testam-se diferentes redes neuronais para esta tarefa e também é proposta uma estratégia de *ensemble* que usa múltiplas arquiteturas e treina-as como uma única rede neuronal.

Um fator importante deste trabalho é que esta rede deve ser capaz de classificar não só imagens obtidas por dermatologistas com ferramentas próprias mas também de imagens obtidas a partir de qualquer câmara RGB por utilizadores, assim possibilitando a implementação do proposto em dispositivos portáteis. Para testar o potencial dos classificadores propostos, múltiplas redes neuronais são treinadas usando diferentes técnicas tais como: *transfer learning*, processamento de imagem, e *data augmentation*, para que a utilidade das mesmas seja determinada quanto às melhorias que tragam à performance deste tipo de sistema.

Esta tese foi proposta e patrocinada pela empresa do sector tecnológico, Glintt. A solução proposta obteve os seguintes resultados na classificação de múltiplos tipos de lesões de pele:

- Accuracy: 73.27% ± 1.40
- Sensitivity: 72.92% ± 0.89
- Specificity: 91.1% ± 0.46
- F1-score: 73.20% ± 1.41

Keywords: neural network; skin lesion; data augmentation; neural network architecture; dermoscopic image; convolutional neural network ; skin cancer; transfer learning.

Contents

List of Tables

List of Figures

Acronyms

ANN	Artificial Neural Networks
BCC	Basal Cell Carcinoma
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DRN	Deep Residual Network
ELU	Exponential Linear Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LReLU	Leaky Rectified Exponential Linear Unit
KNN	K-Nearest Neighbour
ML	Machine Learning
MM	Melanoma
MSE	Mean Square Error
NFN	Number False Negatives
NFP	Number False Positives
NMC	Non-Melanoma carcinoma
NTN	Number True Negatives
NTP	Number True Positives
NV	Nevus, multiple kinds
PTCH1	Protein patched s 1
ReLU	Rectified Exponential Linear Unit
SCC	Squamous Cell Carcinoma
SK	Seborrheic Keratosis
SPP	Spatial Pyramid Pooling
SVM	Support Vector Machines

Chapter 1

Introduction

Motivation

With the life expectancy increase, and the aging population of Portugal, there has been an increase in the frequency of diseases observed, as such more visits to healthcare professionals are now happening. With this phenomenon, healthcare services and infrastructure have started to lose the ability to respond to demands [6]. To help with this problem, the company Glintt proposed the delivery of systems capable of reducing the strain placed upon said healthcare services. One such possible system is one capable of screening some suspicious skin lesions so that there can be a first indication of whether the skin lesion should be accompanied by a professional. Of particular interest are malignant skin lesions, as these can be difficult to identify, and their early identification can lead not only to better treatment options to the patient but also to possible complete cures of a disease which, if left unchecked, would prove fatal [52]. This would consist of a system that would take a picture of the suspect lesion, run it through a neural network trained for the classification of skin lesions and give said classification to the system user.

Objectives:

The main objective of this work is to create a system which is to be based on Deep Learning to be used for the screening of skin lesions from RGB pictures. Said system should be able to distinguish between multiple kinds of skin lesion, and be able to deliver an estimate of malignancy as an indicator that a visit to a health provider is necessary. This is not meant to be a diagnostic tool, but is meant to be an aid for healthcare practitioners, a screening system to relieve some of the burden placed on the healthcare system. Some of the questions to which this work aims to answer are:

- Is a system for the screening of skin lesions and their discrimination into different benign or malign classes a viable proposition?

- How does data augmentation and transfer learning affect the system's generalization ability?
- Is the performance of an ensemble of networks better than a single neural network when it comes to classification?

The basis of this work will be a set of neural networks, training them for this classification task. The neural networks to be used are to be VGG16 [69], Inception v3 and Inception-ResNet-v2 [75, 76], and Resnet50 [33].
The pre-processing of the images is expected to increase the quality and the precision of the features extracted from the images, therefore enabling the neural networks to create and extrapolate upon better feature spaces. Concerning data augmentation, it is expectable that this will have a positive effect upon the performance of the individual models, as it is hypothesized that this technique will improve the generalization capabilities of the neural networks.
The strategies to test these questions will be presented further down.

On Privacy Issues

In spite of the images used being of individual people, not all of them carry metadata that would allow for the identification of the subjects of the pictures, such metadata is discarded, and as nature of the pictures implies very little risk of subsequent identification of the persons whose lesions were documented. In spite of this reduced risk, pictures that do have the subjects face with discriminative capabilities have been removed from the dataset.

Chapter 2

Background

The Skin

The skin (Figure 2.1), or integumentary system, is an organ which is present in all vertebrates. In humans, it is the largest organ in terms of area. This organ serves as a barrier between the external environment and the internal environment of the body, as well as other functions such as temperature regulation, and serving as a medium for sensory input via touch.

Skin is not a simple structure, it is in fact composed of layers which are distinct from one another (see Figure 2.1). These layers include the epidermis, the outermost layer of skin, and the dermis, the biggest layer of skin. In line with the scope of this thesis, the main focus is on the epidermal layer, as it is the main source of skin specific carcinomas.

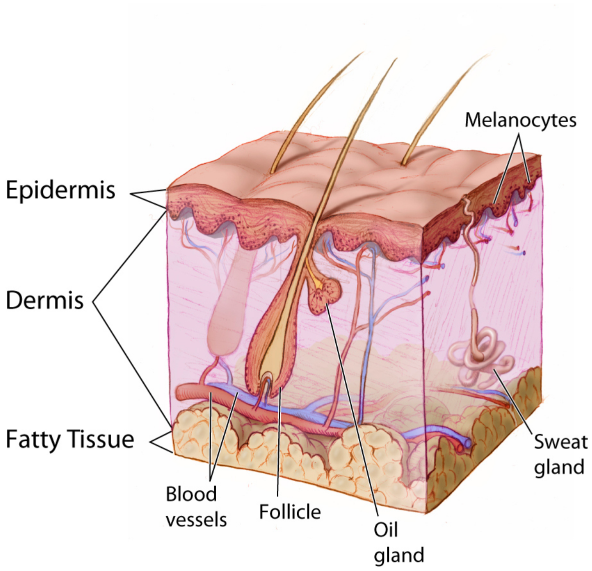


Figure 2.1: The skin's structure.
from https://visualsonline.cancer.gov/details.cfm?imageid=4604 image in public domain Author: Don Bliss

The epidermis itself is composed of multiple layers which again are distinct from one another and are further responsible for different characteristics of the skin. The layers, which can be observed in Figure 2.2, are [5, 48, 50]:

- Stratum Basale, also know as Stratum Germinativum, is the bottom-most layer of the epidermis, which is built of Basal cells, Merkel cells, and Melanocytes, whose branching structure extends into the next layer, in a single row structure. Upon cell division Basal cells can differentiate into Keratinocytes.
- Stratum Spinosum, so called due to the polyhedral shape of its composing cells, is responsible for creating a protective barrier for the lower layers, usually composed of multiple rows, usually between 8 and 10, of Keratinocytes which are formed in the Stratum Basel.
- Stratum Granulosum, so called due to the grainy appearance of its cells. It is responsible for the waterproofing of the skin as well as the creation of a set of proteins which help stopping the entry of foreign entities to the human body. It is composed by rows, usually between 3 and 5, of Keratinocytes. the Stratum Granulosum also serves as a separator between the metabolic active cells and the dead cells of the topmost layers of the skin.
- Stratum Lucidum is a layer that appears transparent, hence its name, and in contrast to other layers of the epidermis, which is not globally present. Such layers are present mainly in areas where the skin thickness is greater such as the sole of the feet and the palms of the hands. It is composed of Keratinocytes which have started to lose their internal cell organelles consisting, therefore, mostly of keratin fibers. This causes the cells to become flattened.
- Stratum Corneum, this is a layer composed of dead cells, usually between 15 and 30 layers thick. One of its effects is the growth inhibition of microbes as well as providing further waterproofing to the skin. The cells of this layer are periodically shed and replaced with cells from the Stratum Lucidum and the Stratum Granulosum.

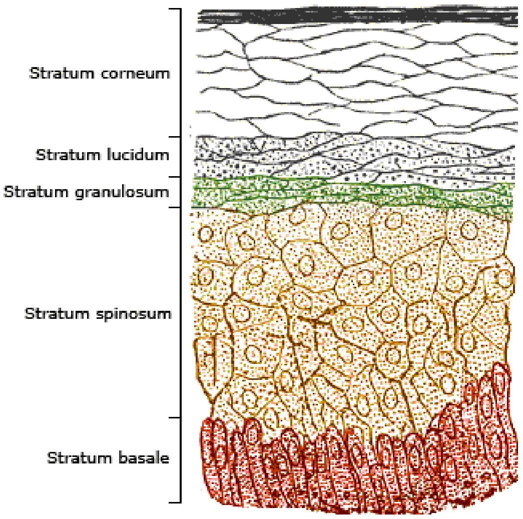
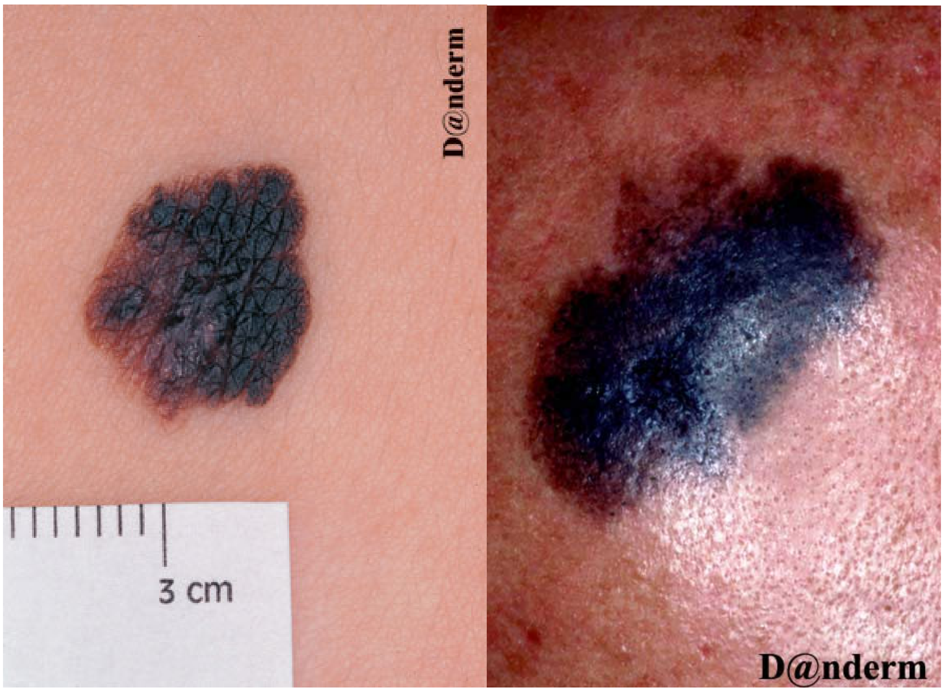


Figure 2.2: The epidermis structure.
from https://upload.wikimedia.org/wikipedia/commons/2/20/Skinlayers.png picture in public domain.

Skin lesions

Skin lesions are any abnormality or defect on the skin, including tumours. These can vary widely in multiple aspects such as colour, shape, texture amongst others and can present themselves in multiple forms, such as ulcers, moles, rashes, amongst others. When it comes to tumours, there are two types, benign and malignant, which will be addressed further down the document.
Whilst many people develop multiple skin lesions along their life, most are not worrisome as they are completely benign and not a cause of concern when analysed by a trained professional. Sometimes, however they can be either a type of lesion which is a known precursor to a malignant tumour on the skin.
In these cases, early detection is possible and tends to lead to curative treatments, when accompanied by a professional, as reported in Jonathan E. Mayer's paper [52], with up to a 49% reduction in mortality rate.



(a) Junction Naevus - benign (b) Melanoma - malign

Figure 2.3: Examples of Skin Lesions
from www.danderm-pdv.is.kkh.dk ©Danderm used with permission.

Another important factor shown by the pictures in Figure 2.3 is that, sometimes, determining the malignancy of a skin lesion is not immediate, as some of the features that dermatologists use to correctly identify, in this case, melanoma can be encountered also in non melanoma lesions e.g., colour. This is an example of reduced inter-class variability that affects the classification of skin lesion pictures.

Cancer

Cancer is a catch-all term for a tumour that is malign. These tumours can affect any organ of the human body and share a few characteristics, such as the way they reproduce and the way they spread, which does not apply to benign tumours. Cancer can be contracted by people of any age, however, it tends to affect people of more advanced age, and some risk increasing factors exist like, for instance, family-history, smoking, and the use of tanning beds [70].
The root cause of cancer is usually a mutation of the cells DNA, which causes them to be able to reproduce in uncontrolled way. This fact makes this kind of mutation dangerous as they create tissue with no way to control its growth, as it will increase in size and potentially affect other biological systems. These mutations can be caused by inheritance of genetic make-up of the patient as well as by environmental factors [14].
Another way that cancer can affect the human body is by the tendency of most forms of cancer to go through a process known as metastasis in which the cancer reaches the vascular system, or

the lymphatic system, and releases a special kind of cells known as stem cells. These stem cells can "seed" cancerous cells in other organs and systems than those which were initially affected, thus originating what are commonly referred to as secondary cancers [3, 28, 72].

Skin cancer

Skin cancer is, as the name indicates, a subset of the catchall term cancer that deals with the cancers which have an origin in the integumentary system. It is usually easier to detect by medical professionals as the organ affected is easily accessible in contrast with other organs or tissue, as, for instance, the liver, since it can often be detected even by the patient or by the medical professional with no need to resort to diagnostic tools.
There are multiple kinds of skin cancer, however, due to some of these kinds rarity we do not have access to the necessary volume of pictures for the training of the proposed system in this document. For this reason we will focus on the three most common kinds of cancer, namely Basal Cell Carcinoma (BCC), Squamous Cell Carcinoma (SCC), and Melanoma.

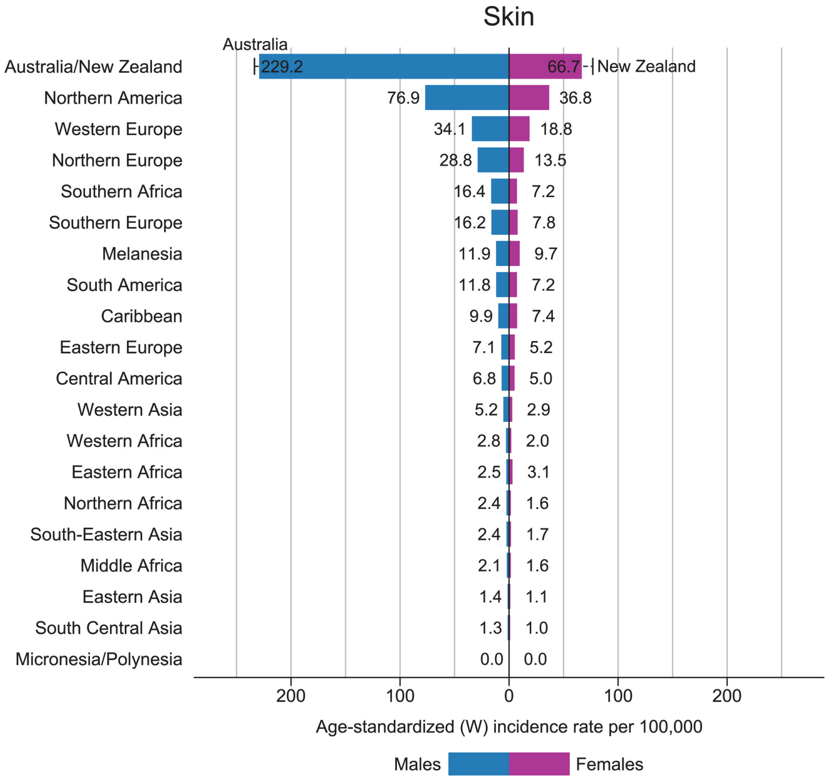


Figure 2.4: Skin cancer incidence statistics by world region.
from https://onlinelibrary.wiley.com/doi/full/10.3322/caac.21492 [8].

Figure 2.4 show the world wide incidence of this kind of cancer in the population, the values presented being age-standardized and gender specific.

Basal Cell Carcinoma

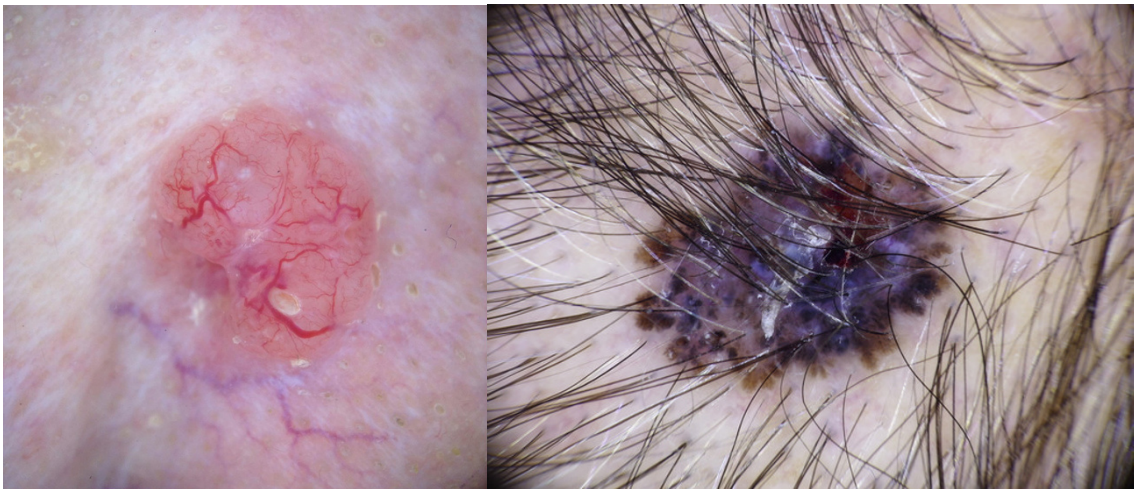
Basal Cell Carcinoma is the most common type of skin cancer (two examples are reported in Figure 2.5) [40, 46]. This type of skin lesion usually presents itself in areas of chronic solar exposure, such as the face, it is typically a slow growing and rarely metastasises. It is however very destructive to the tissues it affects, leading to the death of the patient.
First reported in Dublin around 1827 by Jacob, who coined it as a "rodent ulcer" [39], in the early 20th century, its histologic features were identified by Krompecher, as an epithelial carcinoma [60]. The factors that take part in the appearance of this kind of skin lesion are [60]:

- Ultraviolet radiation, both type A and B.
- Mutation of the Protein patched s 1 (PTCH1) gene, which is often inherited.
- X-ray radiations.
- Arsenic ingestion.
- Steroids use for imuno-suppression.
- Pre-existing conditions such as albinism.

The cells, which can develop this skin lesion are basal cells usually located in the interfollicular, the area of skin where hair grows, hair follicles, or sebaceous glands, as well as the cells in the Stratum Basale [60]. Upon spreading, it usually follows the path of least resistance, therefore it does not usually spread to bone or cartilage tissue. It will however, affect the tissues in between the aforementioned tissues [60], and can be pigmented. Clinical identification usually, follows a set of characteristics which are [60]:

- Primary Description:
  - Macule, a section of skin with altered colour.
  - Papule, a small and solid elevation of the skin.
  - Nodule, a small mass of rounded or irregular shape.
  - Plaque, a localized abnormal section of skin.
- Secondary Description:
  - Scaly, appears to have scales.
  - Crusted, has an hardened exterior i.e., a scab.
  - Lichenified, thick leather like skin.
  - Erosion, worn down section of skin.
  - Ulcerative, a open wound with destruction of tissue.
  - Smooth, continuous and even .
- Shape:
  - Annular, in the shape of a ring.
  - Round, a rounded shape.
  - Irregular, an irregular shape.
  - Serpiginous, a spreading lesion.
  - Diffuse, not concentrated.
  - Erythematous, abnormal redness of the skin.

- Location:
  - Specific.
  - Truncated.
  - Sun-exposed.
  - Previous Surgery site.
  - Others.
- Size.
- Fixation:
  - Fixed to deeper tissues.
  - Fixed to overlying skin.



(a) Clear Basal Cell Carcinoma

(b) Pigmented Basal Cell Carcinoma

Figure 2.5: Examples of Basal Cell Carcinomas
from www.isic-archive.com ©ISIC no restrictions.

Squamous Cell Carcinoma

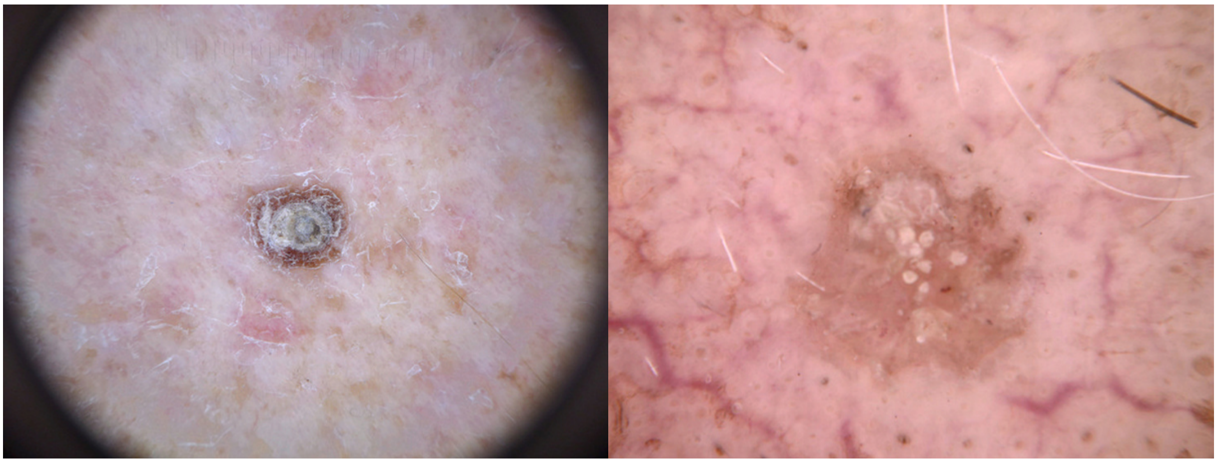
Squamous cell carcinoma (Figure 2.7) is the second most common kind of skin cancer [40, 46]. Unlike basal cell carcinomas, this kind of lesions can occur anywhere on the skin, but like basal cell carcinomas, they are most common on areas of the skin with a high incidence of solar exposure [40]. However they do have the ability to metastasise in a substantial minority of cases; said metastasis appear with some frequency into regional lymph nodes, and with less frequency, but still relevant, they can also metastasise into the following organs: lungs, liver, brain, bone, and into other parts of the skin [1].

This kind of skin lesion affects the cells in the Stratum Spinosum, and shares many causes with basal cell carcinomas. However these skin lesions have some additional causes [1]:

- Chronic injuries or diseases of the skin.
- Evolve from other skin lesions.
- Exposure to ionizing radiation.

Note that this type of carcinoma has a different name for another stage of its development should it not extend past the epidermis: it is classified as "Bowen’s Disease". Also some types of skin lesions, such as Actinic Keratoses (Figure 2.6), have been documented to evolve into SCC, which leads some dermatologists to treat these pre-cancerous skin lesions as SCC for preventive reasons [1].

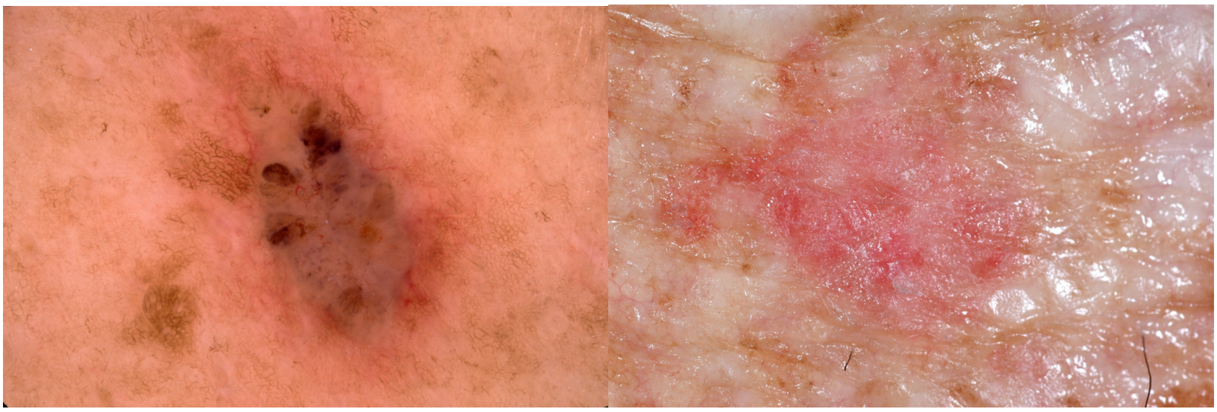
Squamous cell carcinomas, Figure 2.7, are usually diagnosed with subjection of the suspicious lesion to a "histology" of a biopsy of said lesion, as with all skin cancers for confirmation of the differential.



(a) Actinic Keratosis

(b) Another Acnitic Keratosis

Figure 2.6: Examples of Actinic Keratosis
from www.isic-archive.com ©ISIC no restrictions.



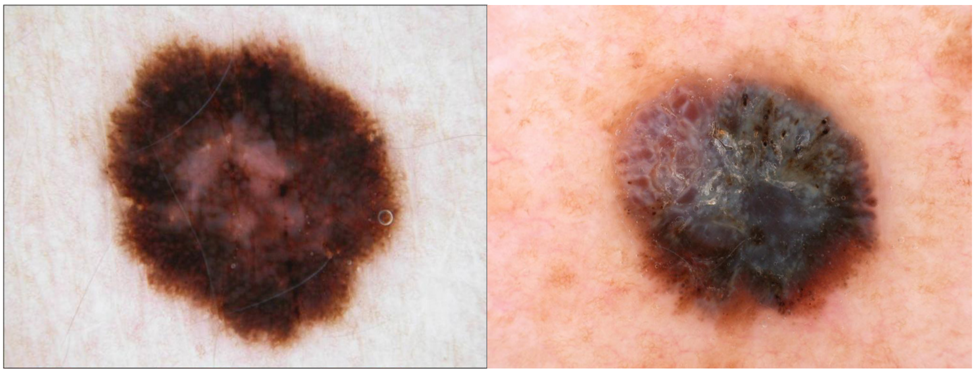
(a) Squamous Cell Carcinoma

(b) Another Squamous Cell Carcinoma

Figure 2.7: Examples of Squamous Cell Carcinoma
from www.isic-archive.com ©ISIC no restrictions.

Melanoma

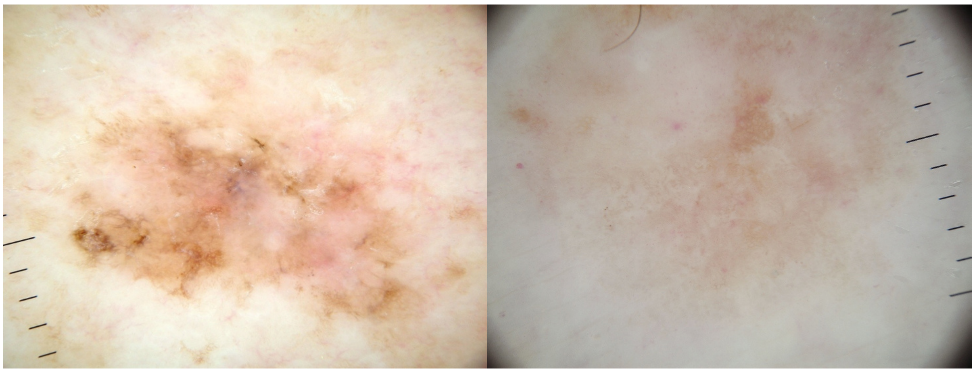
In spite of being, amongst the common kinds of skin cancer, the least common, melanoma (Figures 2.8 and 2.9) is the deadliest [8, 18, 40]. This kind of lesion affects the melanocytes present in the Stratum Basale. It can present itself in any area of the skin, but like both BCC and SCC it is mostly present in the areas of skin that are chronically exposed to solar radiation such as the face, hands, arms, and legs [47].
The risk factors for this kind of skin cancer are the same as mentioned in Section 2.1.3.1. However, for this kind of skin cancer, racial heritage and skin phenotype are especially relevant as they heavily affect the reactions to the solar radiation, especially ultraviolet radiation [47].



(a) A Melanoma (b) Another Melanoma

Figure 2.8: Examples of Melanoma
from www.isic-archive.com ©ISIC no restrictions.

In spite of melanomas affecting melanocytes, which are responsible for the creation of melanin that creates the skin colour observable in humans, they do not always present as heavily pigmented lesions, has can be seen in Figure 2.9.



(a) A Amelanotic Melanoma (b) Another Amelanotic Melanoma

Figure 2.9: Examples of Amelanotic Melanoma
from www.isic-archive.com ©ISIC no restrictions.

Clinical Classification

In the following are reported some rules that have been proposed and used by dermatologists for the classification of skin lesions.

The ABCD rule

Table 2.1: ABCD rule of dermoscopy [71, p. 20]

		Points	Weight Factor	Sub-score Range
Asymmetry	Complete Symmetry	0	1.3	0 - 2.6
	Asymmetry in one axis	1		
	Asymmetry in two axis	2		
Border	Eight segments, one point for abrupt cut-off of pigment	0-8	0.1	0 - 0.8
Colour	One point for each colour: White; Red; Light Brown; Dark Brown; Black; Blue;	1 - 6	0.5	0.5 - 3.0
Dermoscopic Structures	One point for every structure:Pigment Structure; Structureless Network; Dots; Globules; Branched Streaks;	1 - 5	0.5	0.5 - 2.5
Total score range:				1.0 - 8.9

First proposed by Nachbar, *et al.* [58] in 1984, the ABCD rule of dermoscopy, described in Table 2.1, aims at creating a grading system for skin lesions by evaluating features seen through a dermoscope, his is a special tool which dermatologists use as it magnifies the skin lesion and also helps to deal with the skin's natural reflective characteristics. This grading schematic has two thresholds for the classification of the lesions: if the score is $X \leq 4.75$ it can be considered benign. However, should the score be $X \geq 5.45$, it should be considered malignant. Any score in between these two should be considered suspicious and examined for its evolution.

The 7-point check list

Table 2.2: 7-point check-list method [71, p. 20]

Characteristic	Number of Points
Major	
Atypical Pigment Network	2
Blue-Whitish Veil	2
Atypical Vascular Pattern	2
Minor	
Irregular Streaks	1
Irregular Pigmentation	1
Irregular Dots/Globules	1
Regression Structures	1

This classification rule, described in Table 2.2, was first proposed by in 1998 by Argenziano [71] for the classification of pigmented lesions. It was the first to take into account the vascular characteristics of the lesion. The 7-point check-list distinguishes some features from the other by giving them a higher point value and calling them major features. After the points have been attributed to the lesion, if the sum of points of the lesion match $X > 3$ then the lesion is to be classified as malignant.

Menzies Method

Table 2.3: Menzies Method [71, p. 21]

Negative Features
Point and axial symmetry of pigmentation
Presence of a single colour
Positive Features
Blue-white veil
Multiple brown dots
Pseudopods
Radial streaming
Scar-like depigmentation
Peripheral black dots-globules
Multiple colours
Multiple blue/gray dots
Broadened network

This method, described in Table 2.3, first proposed by Menzies in [55], aims specifically at the classification of melanoma. For a lesion to be classified as a melanoma, both of the negative features must be absent and at least one of the positive features must be present.

3-point method

This method, although not as useful from a clinical standpoint as it is a simplified pattern analysis [71], it is, however targeted at non-experts as a screening technique. This method aims at the classification of malignancy vs non-malignancy of a skin lesion. It consists of an analysis of: (1) asymmetry;(2) atypical network, irregular pigmentation of the lesion, such as, colour variation and breaks in the lesion;(3) blue-white structures, whose colour can be described as a whitish blue; the presence of at least two of these features would indicate the lesion malignancy.

Artificial Intelligence

Artificial intelligence is a field of Computer Science the purpose of which is to attempt to understand intelligent entities. To achieve this, it is necessary to create such intelligent entities in the form of computer programs. For a computer program to be considered intelligent there is, at the very minimum, a requirement that it can, from a set of information, reason over said information to come to an actionable conclusion [25, 68].

These computer programs, and the programmers who create them, face multiple problems for example, how to represent information in a formal way such that the program can infer over it. One of the most famous and earliest application of artificial intelligence in a known problem with interesting results is the system known as "Deep Blue" [11], which was capable of playing chess at a very high level, defeating the then World Chess Champion Garry Kasparov. However the rules of chess are very well defined and extremely structured with a universe of only 64 positions and 32 entities that can move on the world, therefore even though it was a great achievement, it played to the strengths of artificial intelligence as this kind of knowledge is easy to represent and infer upon [25]. In this sense, it could be expected that computer system may be able to eventually defeat the best human players in such a controlled and well defined task.

Another good example of this kind of system is AlphaGo [13, 21], a system that plays the game Go. This example is interesting because of the game played which, when compared to chess, is significantly more computationally complex [10, 51] as well as more open ended; it can be hard to determine at most moments if the player is in a better position than the opponent. This algorithm beat the world champion Lee Sedol in 2016.

The ways knowledge is represented in artificial intelligence systems, which can also be called features, can be obtained in multiple manners such as, for example, using hand-crafted features or by allowing the system to create its own features.

Artificial intelligence deals with tasks that can be separated mainly into two distinct fields: classification and regression. Classification is the name given to the task of selecting a value for a given input that is selected from a discrete set of possible outputs, e.g., determining a whether it will be sunny or raining in the following day. Regression is the task of selecting a value from a given input when the possibility of outputs in a non-discrete range, e.g., the price of a house given its characteristics. Multiple approaches to the creation of this kind of systems exist, however, in this document we will focus on classification tasks, since the final objective is a classification problem.

Supervised Learning

Supervised learning is an approach used in machine learning where the learning is done in a supervised manner [68]. Using this approach the algorithm observes some input-output examples and learns to give the appropriate output based on a learned function which is applied on what is called a feature space, a n-dimensional representation of the data with $n = number\ features$, feedback is given to the machine immediately. This representation of data is immutable and its parameters are given by the programmer therefore it does not fit well into another type of learning called representation learning which shall be explained further down this document. Multiple algorithms that use this learning approach exist, some of the most relevant are as follows.

Support Vector Machines (SVM)

Support Vector Machines (SVM) [15] are a machine learning algorithm that, once given a set of labelled features, extrapolates what is called a hyperplane in the feature space that separates the samples of the dataset. It uses different labelled samples to maximize the margin of the hyperplane which is delimited by parallel lines that contain samples of different labels which are closest to the other labels, called support vectors. The hyperplane that separates one space into two subspaces is normally defined by $w * x - b = 0$, where w is a vector which is orthogonal to the hyperplane, and b is bias or deviation from the origin.

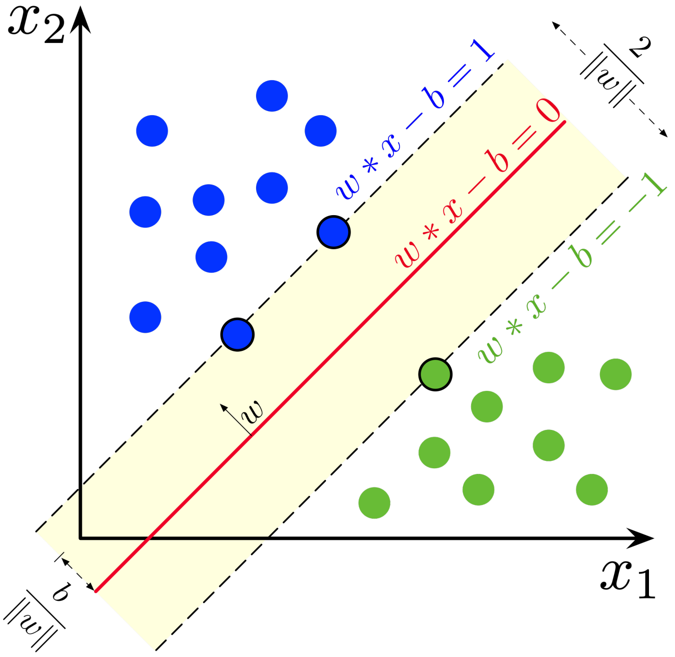


Figure 2.10: SVM Feature space hyperplane
from https://en.wikipedia.org/wiki/File:SVM_margin.png image by Larhmam released under CC-ASA-4.0

The hyperplane is the red line present in the Figure 2.10. In this example, a 2-dimensional feature space is considered; in this case a plane is simply a line and w is a two dimensional vectors of coefficients that determine the spatial position of the hyperplane [15].

K-Nearest Neighbour (KNN)

K-nearest neighbour (KNN) [27] is an artificial intelligence algorithm which exploits the similarity of same labelled features. When the system is asked to classify a new unlabelled sample, the algorithm simply represents the new sample in the feature space. It then finds the K nearest neighbouring points already present in said feature space. Afterwards the label with the most samples in the nearest, according to a given metric, neighbours is determined and that label is attributed to the unlabelled sample.

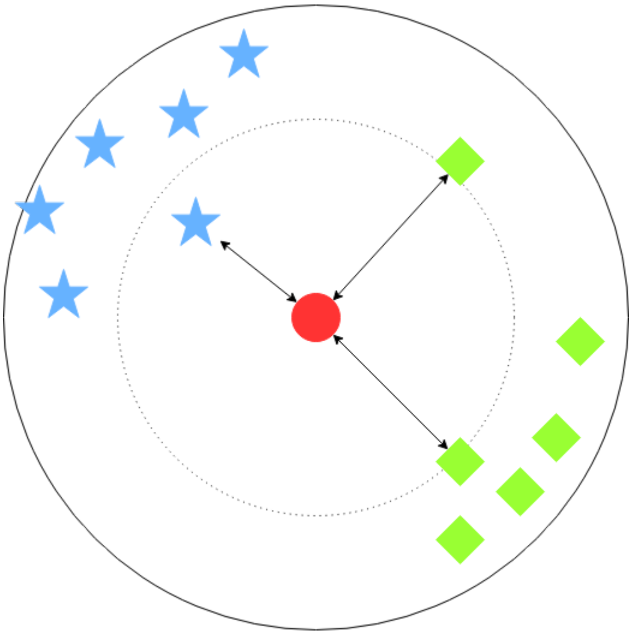


Figure 2.11: KNN Feature space classification diagram

Figure 2.11 shows a representation of how the 3-NN algorithm works of a 2-classes classification example. The point to be classified is in red, and the nearest neighbours are indicated by the arrows in between the point to be classified and the selected neighbours. In this case the classification would be the same as the green points as those are in a majority. Should the K selected be 12 for this dataset, the classification of said sample would have been attributed by a set of rules implemented by the programmer, for instance allow for multi-label, since the number of blue and green points inside the margin are the same.

Another version of this algorithm uses clustering to help with the classification. This variation creates one cluster per class and the closest cluster to the point to be classified will provide the point with its label, as well as being more resistant to outliers.

Representation learning

Representation learning is a subset of supervised learning with a defining characteristic; the dataset is given to the algorithm with little to none feature engineering, i.e., the act of pre-processing the dataset, so that the algorithm can exploit the prior knowledge the programmers might have on the subject. Of importance is the fact that in this case, the representation, i.e., the features, are automatically determined by the algorithm in the training phase. Some characteristics can be seen in the data [4]:

- Smoothness: The assumption that the function calculated will maintain the correlation between any two different points.
- Multiple explanatory factors: The label of a point varies according to multiple features of the dataset, helping it to achieve better generalization.

- Hierarchical organization of explanatory factors: Parameters have a hierarchy to them as it is possible to get more abstract parameter from simple parameters.
- Shared factors across tasks: Different tasks share some of the same representations, e.g. classifying Car vs Bicycle would share parameter to the classification of Truck vs Motorbike, this allows for a technique called transfer learning.
- Manifolds: dimensionality reduction should help to generalize the representation of knowledge, as the probabilities of items to classify would cluster better in reduced dimensions, manifolds can be described as a connected set of points that can be approximated well.
- Natural Clustering: There should be little overlapping of the classes, and the classes should be well separated.
- Temporal and spatial coherence: Consecutive or spatial proximity of observations tend to be associated with the same label.
- Sparsity: Given a set of features for an observation of a label not all of said features should be relevant to the classification of an observation.
- Simplicity of factor dependencies: Factors of the observations relate to each other in simple linear ways.

Should these characteristics be observed in the data, it should be possible to create a good data representation, allowing for the multiple algorithms to be applied.

Artificial Neural Networks

Artificial Neural Networks (ANN) are a subset of Machine Learning (ML) algorithms which operate in a way that mimics the way that neurons work. Some of the early approaches that attempted to achieve this are based on McCulloch & Pitts neural cells [53]. More modern neural networks built upon this model to create software that is capable of learning tasks.

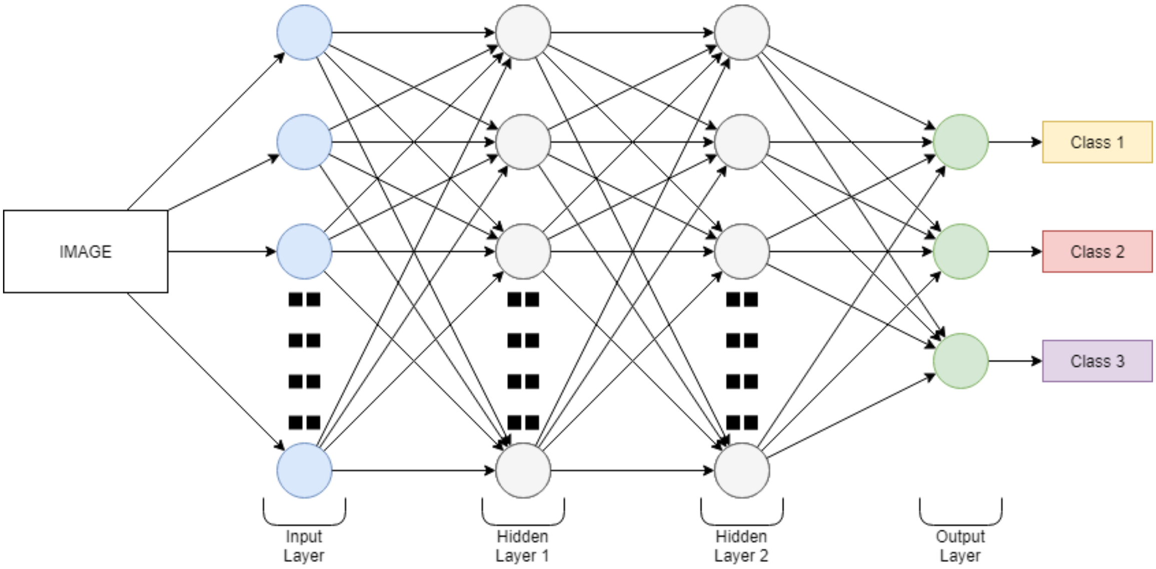


Figure 2.12: Neural Network classification architecture

In Figure 2.12 is reported a representation of a neural network architecture. This architecture takes an image and classifies it into one of three different classes. The architecture is formed by two hidden layers that are connected in a dense way. What follows is an explanation of how neural networks function [25, 61].

A **neural network** is a model which, being inspired by the biological brain, is composed of **layers** which are composed of a set of **neurons** which can be seen as individual nodes of the neural network. The neurons are capable of taking multiple inputs and generate an output. The output generated by a neuron is the result of the application of a function, called an **activation function**, to a weighted sum of all the inputs of the neuron plus a **bias**, a learned offset from the origin of the function. The number of layers, neurons and the types of layers of a neural network is called its **architecture**.

The layers of a neural network can come in multiple forms, some of which are present in all neural networks and some of which are specific of a particular kind of neural network. The layers that are present are the **input layer**, layer that receives the input, **hidden layers**, that do the actual work taking inputs only from other layers and pass their outputs only to other layers in the network, and finally the **output layer**, that take its input from other layers, but its output is sent as the prediction of the machine to a given observation.

The purpose of a neural network is to be able to **predict**, the act of labeling a previously unseen **observation**.

To achieve this purpose, the network needs to be **trained**. Training is the process through which a neural network learns to classify the new observations, creating and optimizing the **parameters**, variables the neural network learns.

This is possible because the data is composed of **features**: the neural networks uses these features to discover the **weights** and optimize their values according to their importance. Weights are coefficients of importance of a feature for the model.

A very important component of an ANN is what is called a **loss function**, which can also be

called a **cost function** or **error function**. This function, usually a non-convex function as such the minimum value found will usually be local minima, is what allows for the optimization of the learned weights. A fundamental characteristic of this function is to be derivable with respect to the parameters of the network, therefore allowing the system to compute the gradient with the process described bellow. It is also useful as a metric for the quality of the model.
Let L be the Mean Square Error (MSE) loss function expressed as:

$$L = \frac{1}{n}\sum_{\imath=1}^{n}(y_\imath - \hat{y}_\imath)^2, \tag{2.1}$$

with y being the ground truth and $\hat{y}$ being the predicted value, this loss is also know as L2 loss.

The optimization of the weights is done via a **gradient descent** algorithm. The **gradient** is a vector containing the partial derivatives respective to all the independent variables of the model function. The gradient descent uses the gradient to adjust the weights by the **learning rate**, a scalar selected by the programmer that is multiplied to the current gradient.
A gradient descent step is defined as:

$$w := w - \eta \bigtriangledown Q(w) \tag{2.2}$$

where w represents the set of network parameters weights, η the learning rate, and $Q(w)$ the Cost function.

The process of calculating this gradient over the whole neural network is called **back-propagation**. This back-propagation first calculates the output of each neuron in a **forward pass**, a pass that goes from the input layers towards the output layers. The partial derivative of the error for each parameter is then calculated and updated in a **backward pass** through the neural network.
The gradient optimization is prone to a few of problems, such as the **exploding gradient** problem, in which the gradient grows out of proportion destroying the learning already done by the network, and the **disapearing gradient** problem, which is the polar opposite of the exploding gradient, this means that the gradient grows so small that the computer becomes unable to represent it, thus making the neural network unable to learn further. Another problem that this algorithm can encounter are local minima where a gradient descent can be stuck and therefore not find the optimal value, finding a sub-optimal value.
To avoid local minima a technique called **momentum** can be used. This forces some oscillation around the found minimum, enabling the neural network to find a better minimum, hence avoiding local minima.
A weight update using the gradient with momentum can be expressed as:

$$v \to v' = \zeta v - \eta \bigtriangledown Q(w), \tag{2.3}$$

with v being the momentum factor, $\eta \bigtriangledown Q(w)$ as in Equation 2.2 and ζ an hyper-parameter. This is then used to update the weights in the following manner:

$$w \to w' = w + v' \tag{2.4}$$

Here the momentum can be seen as a bias added to the gradient descent.

The training of the neural network is not done all at once, it is done iteratively. Such iterations are called **epochs**: in each epoch, all observations of the training data will be processed by the neural network once; furthermore each epoch is divided into **batches**: each batch is a subset of the training data. Upon completing the analysis of each batch the back-propagation algorithm is then applied.
Even assuming that the learning process can determine the network weights that minimize the value of the loss function when applied to the training data, this does not immediately imply that the learned model will be able to produce reliable classification when applied to previously unseen data. To test the evolution of the performance of the network during the training process, the loss function is applied to a set of data, called validation data, that are not involved in the gradient descent procedure. Once the loss of the model does not improve further, the model has reached a state known as **convergence**. A few problems can become apparent once the neural network has finished training. These problems deal with how well the network can **generalize**, i.e. the ability of the model generated by the neural network to make predictions on previously unseen data. These problems are called **overfitting** and **underfitting**. Overfitting happens when the model is so closely correlated with the training data that it cannot predict anything outside of it. On the other hand, underfitting stems form the model's inability to capture the complexity of the training data. Some possible causes of this are insufficient training data, using the wrong features for training, problems with the architecture chosen for the neural network, or even **noise**, inaccuracies in the labels of the training data that disrupt the feature gathering, in the data.
Some methods to avoid these problems are for instance, **batch normalization**, normalization of the input or the output of the activation functions of hidden layers, and **regularization** which is a penalty applied to the loss function and whose **regularization rate**, a scalar applied to the loss function, serves as a hard-coded limitation for the minimization of the loss function.
Let L be the loss function defined in (2.1), and $\{\beta_j\}_1^p$ the set of weights/ parameters of the neural network A loss function including an ℓ_2 regularization term, also known as **weight decay** expressed as:

$$L' = L + \lambda \sum_{\jmath=1}^{p} \beta_\jmath^2 \tag{2.5}$$

with λ being the regularization rate.

Hyperparameters are some variables that the programmer can directly change, which affect the performance of the neural network, examples of these are:

- The number of epochs to run,
- The value of the different rates (e.g. learning rate, momentum rate, etc.),
- The the number of observations used in each batch, also called batch size.

A more complete glossary can be found in [26]. A deeper look into neural networks and their inner working will be undertaken in a forthcoming Chapter 3.

Deep learning

The term Deep Learning (DL) encompasses a set of ANNs, whose concept stems from the findings of Hubel and Wiesel [34, 35], as mentioned in [64], which describe the functioning of the visual processing of images by a cat's brain. This spurred interest by the computer sciences communities around the world, however, there was a huge limitation regarding the computational capacity of the computers that were to train and run these neural networks. With the leap in GPU technology, there was a realization that these could be used for this purpose as such after a stagnant period in the investigation of these networks the interest renewed [64]. These networks have multiple hidden layers and the number of layers can vary wildly, for instance, the known architecture VGG-16 [69] has 21 layers with 16 convolutional layers, whereas, the architecture known as ResNet [33] can have over 1000 layers. Of particular relevance for deep learning, networks are convolutional layers. These layers are specific to a type of neural networks which are called **CNNs!** (**CNNs!**) and that will be described in Chapter 3.

This approach tends to deliver better results at the cost of a more significant computational burden [61].

Deep Neural Network (DNN)

The concept of Deep Learning introduced in the previous chapter will be expanded upon here in a more detailed way.

The current increase in the use of deep neural networks for image classification was spurred in 2012 with the record-breaking performance of the neural network AlexNet [43] which beat SVM approaches in the famous ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[67] image classification competition with a top-5 error of $15,3\%$.

This showed that deep neural networks and more specifically convolutional neural networks are a worthwhile technique to pursue computer vision tasks such as image classification.

Convolutional Neural Networks

These neural networks are characterized by the presence of convolutional layers, which will be described in a later section. These neural networks have proven very effective in the tasks of image classification.

Normally split into two main sections, these are a feature extraction section and a classification section. The feature extraction area is where the convolutional layers exist, which can be associated with pooling layers, which will be described in the Section 2.4.2, and are responsible for extracting the features from the images, which are then sent into the classification section of the network. Usually, this section consists of a few dense layers with the last having a softmax activation, defined in Section 2.4.3, so that it outputs the probabilities of each of the classes in the task, with the highest probability amongst the classes being the selected classification prediction.

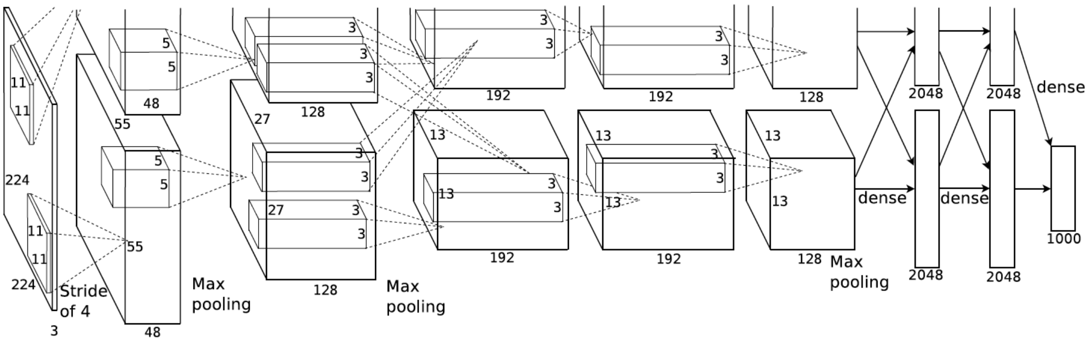


Figure 2.13: The Original picture of the architecture of the neural network AlexNet
from:[43]

Figure 2.13 shows the original architecture of AlexNet as seen in Krizhevsky's paper [43]. However, in Figure 2.14, a clearer overview of AlexNet's architecture can be observed. As can be seen in the original picture there are 2 different networks that interact amongst themselves, this is because the network was designed to use 2 GPUs in parallel.

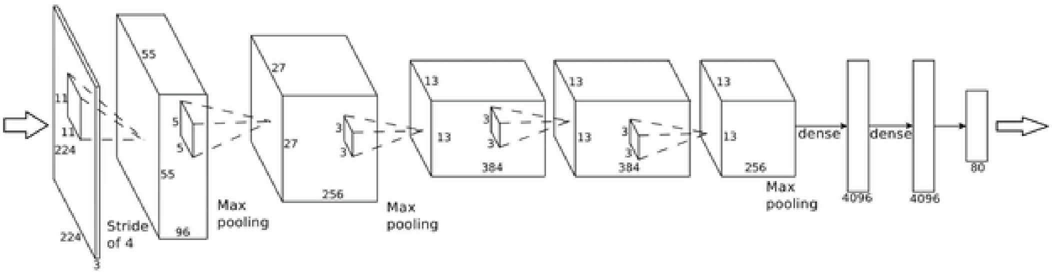


Figure 2.14: The architecture of the neural network AlexNet
released under CC4.0 adapted from:[62]

As can be seen by comparing Figures 2.13 and 2.14 the latter there does not have a separation for the utilization of 2 GPUs but instead it doubles the size of the convolutional layers and the dense layers, with the exception of the last layer, as the first was used for the ImageNet challenge and therefore needed to classify 1000 classes, the second one was used to classify amongst 80 different classes.

This architecture has 5 convolutional layers with a maxpool operation after the convolutional layers 1, 2, 5, followed by 3 fully connected dense layers, it uses filters with dimensions 5×5 and 3×3.

It learns over 60 million parameters, which can lead to some overfitting problems. To mitigate this, a technique called drop-out, which will be described later, is implemented with the first 2

convolutional layers. AlexNet requires that the images have a fixed size 224 × 224.
Whilst most **CNNs!** require a fixed image size, there is a kind of convolutional neural network which is invariant to the size of the input. Implementing a special kind of pooling called Spatial Pyramid Pooling (SPP) [31]. This technique consists of the application of a pooling layer preceding the fully connected layers. It allows for a range of image sizes and resolutions to be accepted as input by extracting parts of the images instead of using the full image provided. This kind of pooling applies to any CNN.
Convolutional neural networks have been selected for use in the system described in this document as they show the best results in image classification tasks [29]. Furthermore, an ensemble of multiple different **CNNs!** will be used as these have been proven to improve results over a single network approach as reported by Miclut [56].

Types of layers

As mentioned in the previous Section 2.3, it is possible to classify the layers of a neural network in a finer grain than what was already described. Here are presented a few examples of said different layers.

Dropout The drop-out layer (Figure 2.15) is used as a method for the reduction of overfitting of a neural network, at the risk of forcing the neural network to underfit.
This type of layer "drops", which is to say stops some of the neurons from connecting to the next layer, usually with a probability that is hard-coded by the programmer when he is defining the architecture of the neural network. The higher this probability is, the less likely it is that the network will over-fit, but the more likely it will under-fit. This happens only during training when the neural network is asked for a prediction all nodes are used.

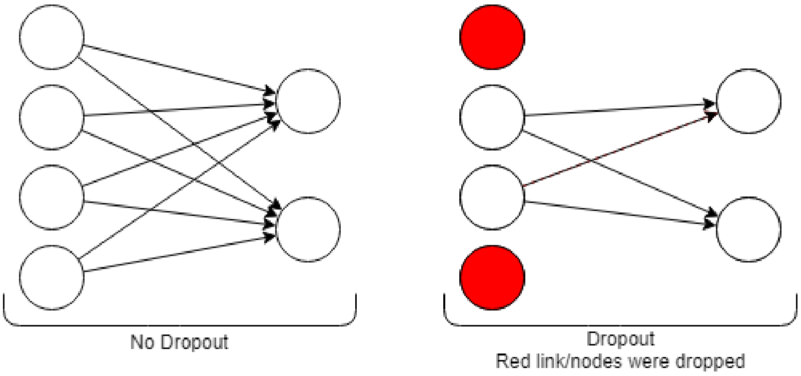


Figure 2.15: Example of a Dropout layer
own work.

Convolutional The convolutional layer is the defining feature of a Convolutional Neural Network (CNN). The way these layers work is by applying a **convolution** (Figure 2.16) to the data. A convolution is a mathematical operation which is performed in two steps:

1. element-wise multiplication of the **convolutional filter** and a slice of the input matrix, and
2. summing all the obtained values in the resulting product matrix.

A convolutional filter is one of the required parts for a convolution. This filter is usually smaller than the input matrix.

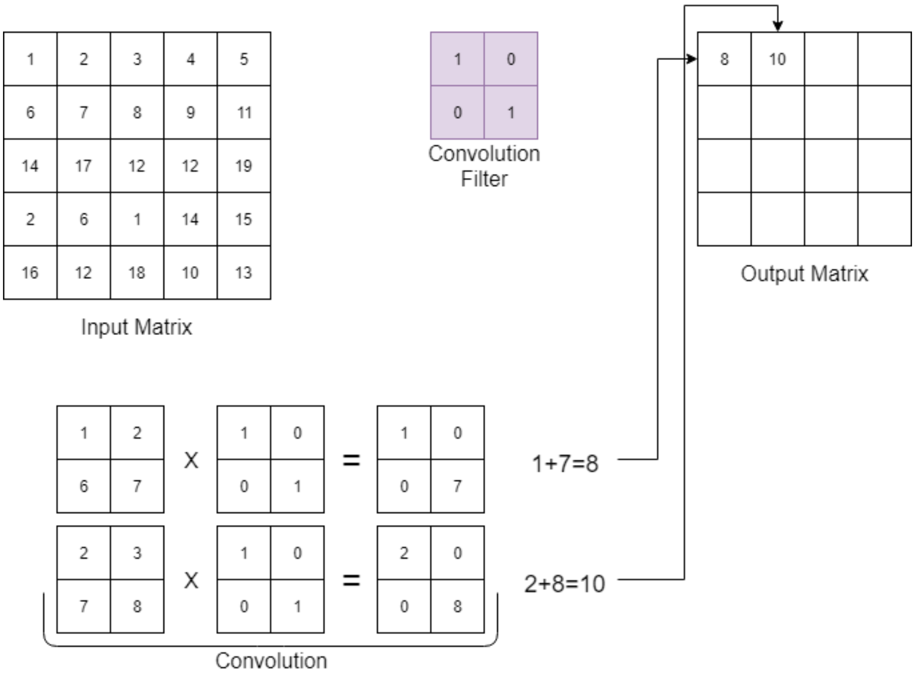


Figure 2.16: Example of a convolution operation
own work

The convolution is defined in mathematics as:

$$y[n,m] = \sum_j \sum_i x[i,j] \cdot h[n-i, m-j], \tag{2.6}$$

with y being the result matrix, x being the convolutional filter, and with h being the input matrix, however most implementations of this operation for DL do not follow this formula exactly, opting to do a correlation which changes the $h[n-i, m-j]$ into $h[n+i, m+j]$. This difference correlates only to a flipping of the filter weights since the filters are learned by the neural network. Another difference in this operation is that there is a **stride** added, which can be taken as a shift of the filter over the input matrix meaning that less multiplications are performed. This changes the factor $h[n+i, m+j]$ into $h[n \times s_x + i, m \times s_y + j]$, for a stride (s_x, s_y) with s_x, s_y different from 1.

In the example reported in Figure 2.16 the input matrix is 5 × 5 and the convolutional filter is 2 × 2 thus making the output matrix take a shape of 4 × 4. The filter itself is, in this case, the 2 × 2 identity matrix. The input matrix is split into multiple slices with the same dimensions as the convolutional filter. The filter is then multiplied to each of these slices and a sum of the resulting values is calculated. Said sum is then stored in the output matrix. An important note is that this operation is performed in a by channel basis which is to say, in a RGB picture the

R, G, B values of each pixel are separated into their own matrices and then the convolution is performed on each of them, for this each layer receives and outputs multiple 2-dimensional feature maps.

Pooling The pooling layer is usually used after a convolutional layer to downsample the output of the convolutional layers.
There are two types of commonly used pooling operations (Figure 2.17), which are:

- Max-pooling is a type of pooling which returns a matrix where each value is determined by the function

  $$o[i, j] = max_{n=1,\ldots,N;m=1,\ldots,M} f[(i-1) * N + n, (j-1) * M + m], \tag{2.7}$$

  where N and M represent the size of the slice of input considered for each max operation.

- Average-pooling is a type of pooling which returns a matrix where each value is determined by the function

  $$o[i, j] = avg_{n=1,\ldots,N;m=1,\ldots,M} f[(i-1) * N + n, (j-1) * M + m], \tag{2.8}$$

  where N and M represent the size of the slice of input considered for each averaging operation.

These layers usually allow the network to have some translational invariance.

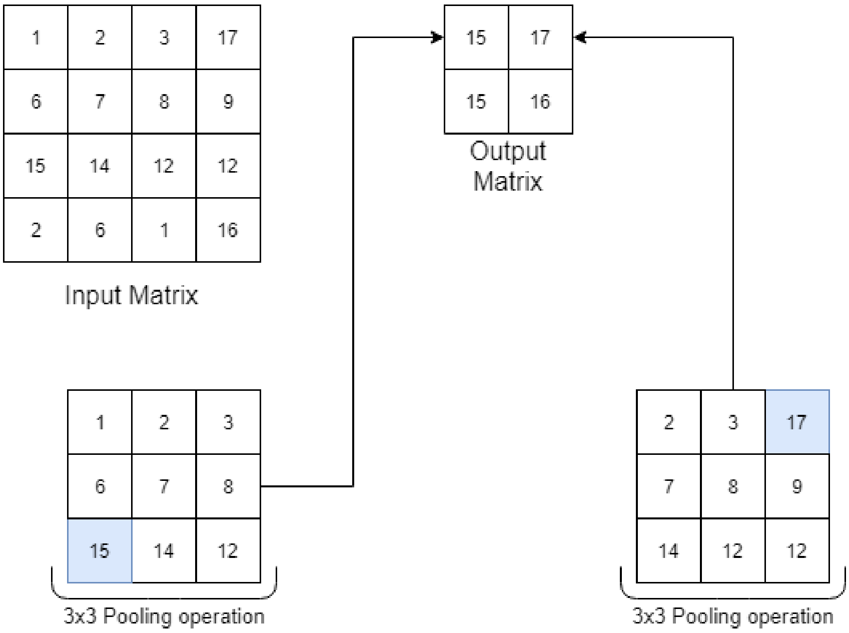


Figure 2.17: Example of a max-pooling operation
own work

In this Figure 2.17 a pooling operation is shown. The input matrix is split into multiple slices and the maximal value of each of the slices is selected into the output matrix. In this example, the slices are 3×3 and the input matrix is 5×5 which leads to 4 different slices therefore, the output matrix has 4 values.

Activation functions

The choice of which activation function is to be used in a layer has an impact upon the training and therefore the usefulness of the network for the task. A neuron outputs the result of the weighted sum of its inputs on the activation function. Here are some of the most important activation functions used in neural networks.

Hyperbolic Tangent This function is used for its non linearity and also for being a simple and well known function.

Function: $f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$

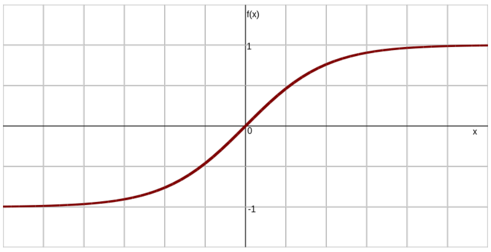


Figure 2.18: Tanh function
from https://en.wikipedia.org/wiki/File:Activation_tanh.svg
Released under CC4.0 author: Laughsinthestocks.

Sigmoid This function is commonly used in neural networks to introduce non-linearity in the model, and its generalization function for more than two cases is described in Section 2.4.3.

Function: $f(x) = \frac{1}{1+e^{-x}}$

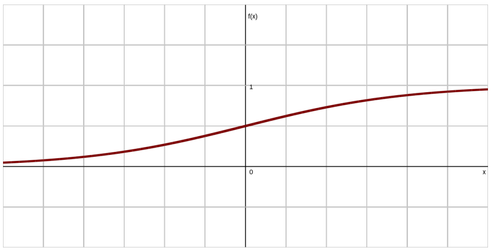


Figure 2.19: Sigmoid function
from https://en.wikipedia.org/wiki/File:Activation_logistic.svg
Released under CC4.0 author: Laughsinthestocks.

Rectified Exponential Linear Unit (ReLU) [24, 64] This function allows for an easier creation of sparse representations, which is to say it reduces the amount of neurons that are activated and thus improving computational times and avoid the disappearing gradient problem. However, the fixing of a minimum output of 0 may hurt the back-propagation algorithms as it will ignore negative features, being functionally the same as a non activating neuron.

Function: $f(x) = \begin{cases} 0 : \text{for } x < 0 \\ x : \text{for } x \geq 0 \end{cases}$

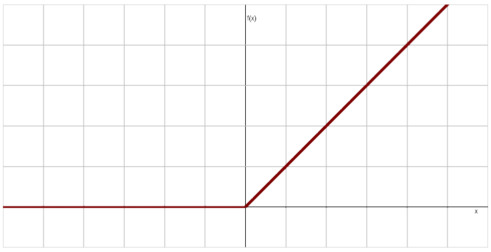


Figure 2.20: ReLU function
from https://en.wikipedia.org/wiki/File:Activation_rectified_linear.svg
Released under CC1.0 author: Laughsinthestocks.

Exponential Linear Unit (ELU) [64] Because ELU allows for negative values, this makes the mean of the activations closer to zero, which allows for faster training and convergence than that allowed by ReLU.

Function: $f(x) = \begin{cases} \alpha * (e^x - 1) & : \text{for } x \leq 0 \\ x & : \text{for } x_0 \end{cases}$ With α being a scalar.

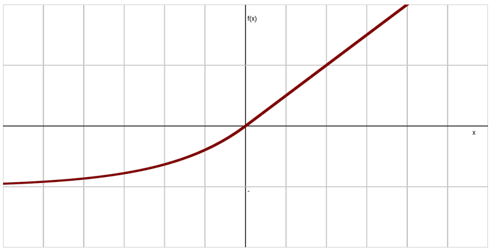


Figure 2.21: Elu function
from https://en.wikipedia.org/wiki/File:Activation_elu.svg
Released under CC4.0 author: Laughsinthestocks.

Leaky Rectified Exponential Linear Unit (LReLU) [64] This function works in a very similar way to the ReLU function, however, it leaks, which is to say that some signal is allowed should the input be negative. It avoids a problem of the ReLU which is the leading of some neurons to never Activate [64].

Function: $f(x) = \begin{cases} 0.01x : \text{for } x < 0 \\ x \quad\;\; : \text{for } x \geq 0 \end{cases}$

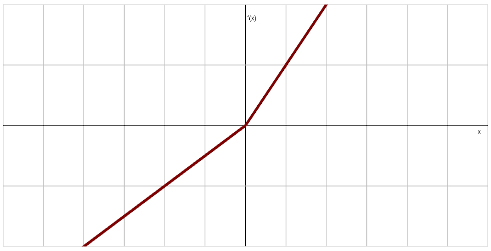


Figure 2.22: LReLU function
from https://en.wikipedia.org/wiki/File:Activation_prelu.svg
Released under CC1.0 author: Laughsinthestocks.

Softmax This function is a generalization of the logistic function that flattens a J-dimensional vector $\vec{x}$ of real values to a J-dimensional vector $\vec{y} = f(\vec{x})$ of real values with $J \geq 2$, which each entry y_j in the interval $[0, 1]$ and such that $\sum_{j=1}^{J} y_j = 1$

Function: $f(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}$ with $1 \leq i \leq J$,
where J is the number of possible classes. The purpose of this activation is to give a probability for each class so as to provide a prediction of classification.

Example of how it works Lets take a neuron that takes three inputs $(x1, x2, x3)$ with a bias denoted as b which is set as 0, and used the ReLU activation function.

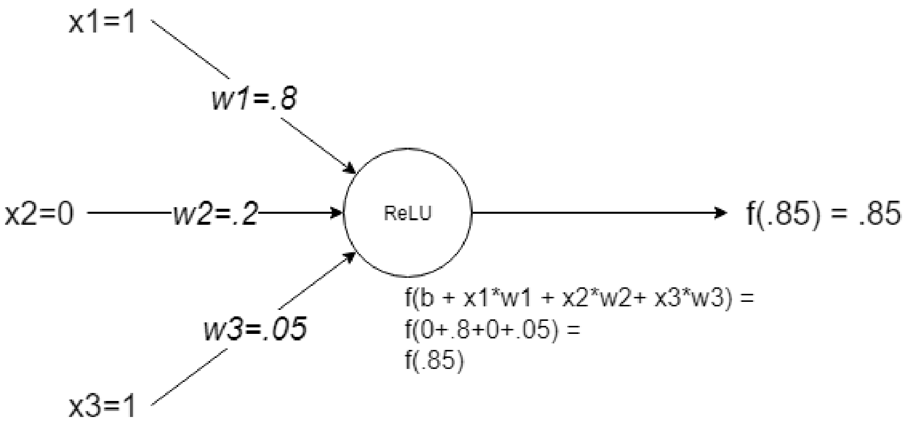


Figure 2.23: Example of the way neurons work with b=0
own work.

In Figure 2.23 it can be seen how a neuron with the ReLU activation function works with the given the inputs, and bias set to 0. This activates the neuron with a output of .85. Let us see what would happen if we set the bias to -1.

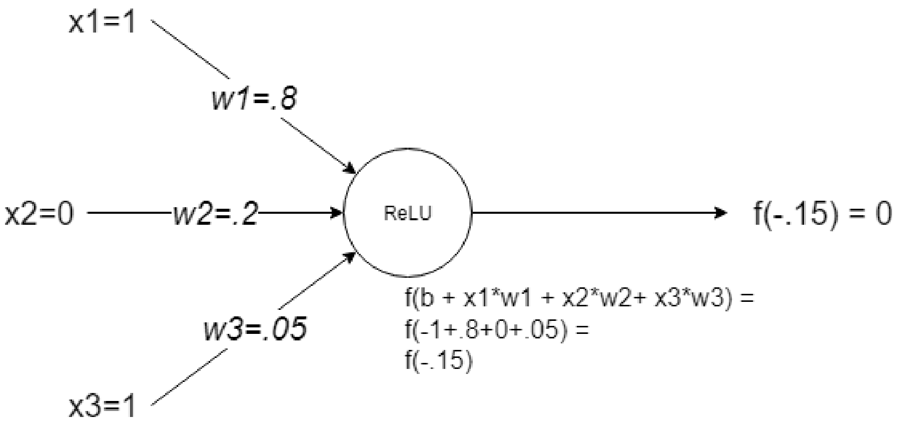


Figure 2.24: Example of the way neurons work with b=-1
own work.

In Figure 2.24 the impact of a different bias can be seen. The inputs are exactly the same as the ones for Figure 2.23. Due to the bias being -1 the neuron will not activate, which is to say the output is 0. Now for the impact of a different activation function, lets use the hyperbolic tangent function.

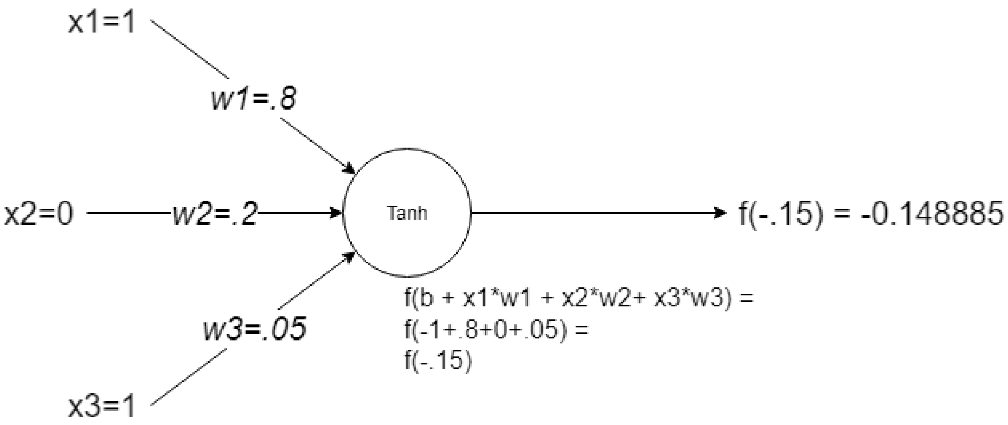


Figure 2.25: Example of the way neurons work with b=-1 and using tanh activation
own work.

In Figure 2.25 it can be seen that under the same conditions the different activation functions can, and most likely will, give different results and therefore have an impact upon the model generated by the neural network.

Invariance

Invariance is the ability that a neural network shows in generalizing to a characteristic of the input, for instance, in the case of images, rotational, translational, and scale invariance.
This means that an image containing a dog, should the network be translational invariant, it would be of no relevance where on the image the dog is. Should the network be rotational invariant then the dog could be on its side and the classification would not be problematic.
These characteristics, when acquired by the neural networks, make for more robust and general models for a given classification task.
For example **CNNs!** are known to acquiring a certain level of invariance due to the way the feature maps generated by the different filters and pooling, as they also reduce the dimensionality of the feature maps. For instance, if a Max-pool attributes a value of X to a point of the output matrix it means that this value of X could be present at any point of the slice of the input matrix. This allows for small variations in the shapes of the images.

Transfer learning

Transfer learning is a very important technique that can be used for the training of neural networks. This is the act of using the weights found by a neural network trained for a different task, to initialize that same neural network for a new task, for instance, using a model that can discriminate images of dogs and cats, being used for the discrimination of horses and zebras. This carries the assumption that the weights learned for the first task will be useful for the second, as well as the assumption that the semantic features learned for the first task will be useful for other tasks. This is the reason why the ImageNet dataset is normally chosen for transfer learning, due to the number of different classes and images that are present. To this end, the architecture of the first network is imported and the weights are loaded. However the output layers are replaced, and the neural network is retrained for the new task, normally using reduced learning rates so it becomes fine-tuned for the new task. Normally the number of epochs to achieve a point of convergence is reduced when compared to the training without transfer learning, therefore reducing computational costs.

Chapter 3

State of the Art

In this chapter is a discussion on the state of the art that was analyzed under the purview of this work, how it was collected, the analysis, as well as the conclusion pertaining to the research direction that this work followed.

Sources

Some books are being used for this analysis of the state of the art in this document, namely:

- Color atlas of melanocytic lesions of the skin [71]
- Human anatomy & physiology [48]
- Anatomy and Physiology [50]
- Anatomy and physiology [5]
- The cancer book: A guide to understanding the causes, prevention, and treatment of cancer [14]
- Deep Learning [25]
- Artificial intelligence: a modern approach [68]

These were found via searches on google and google scholar as pertaining to knowledge that is normally considered common in the field and therefore not usually subject to an active investigation by the communities of each respective field. These were obtained either by their free distribution online, or by what was considered, by the author of this document, as sufficient information via Google's book preview.

Some of the articles used for this state of the art were not selected via a structured query to a database but rather as the ones which introduced certain architectures or concepts which are required or pertinent to the scope of this work.

Table 3.1: Search terms and results

Query	Engine	Number Papers
(("skin lesion" OR "skin cancer") AND "image classification" AND "dermoscopy")	IEEEXplore Scopus	65 37
(("skin lesion" OR "skin cancer") AND "image classification" AND "dermoscopy" AND (deeplearning OR "deep learning"))	IEEEXplore Scopus	7 12
("isic archive" AND "skin cancer" AND "image classification" AND dermoscopy)	IEEEXplore Scopus	0 1
(("skin cancer" OR "skin lesion") AND "neural network" AND "image classification")	IEEEXplore Scopus	40 86
(("skin cancer" OR "skin lesion") AND "neural network" AND ("deep learning" OR "deeplearning"))	IEEEXplore Scopus	16 168

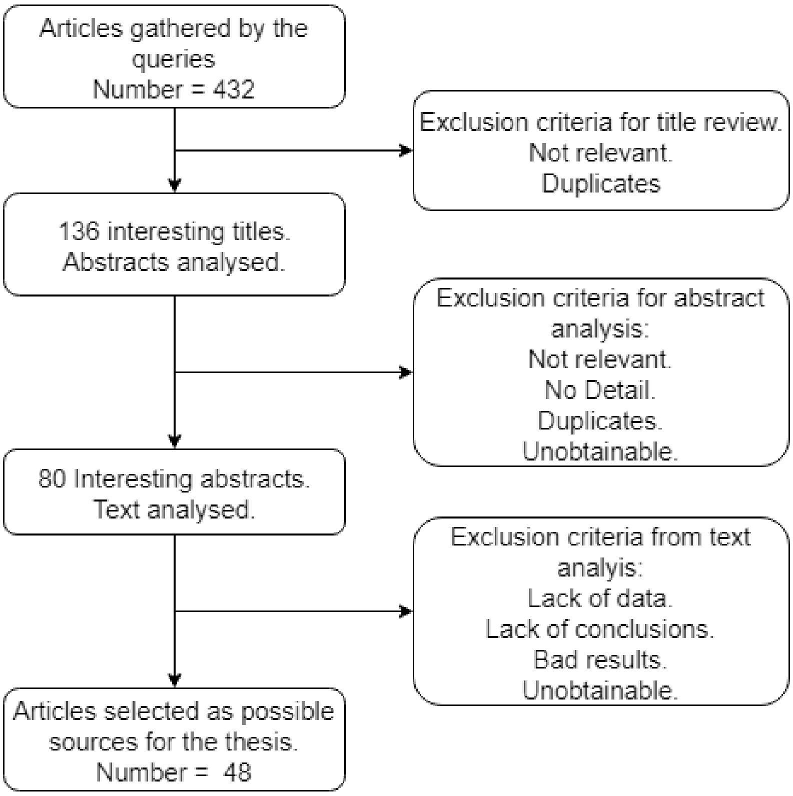

Articles gathered by the queries
Number = 432
Exclusion criteria for title review.
Not relevant.
Duplicates
136 interesting titles.
Abstracts analysed.
Exclusion criteria for abstract analysis:
Not relevant.
No Detail.
Duplicates.
Unobtainable.
80 Interesting abstracts.
Text analysed.
Exclusion criteria from text analyis:
Lack of data.
Lack of conclusions.
Bad results.
Unobtainable.
Articles selected as possible sources for the thesis.
Number = 48

Figure 3.1: Source Selection

All searches were performed on the 3^{rd} of January 2019 with the following limiters when possible:

- Date between 2010 to 2019 - to ensure recent data.
- OpenAccess - to ensure the documents would be accessible.

Papers relating to submissions for the ISIC-Archive challenges [37] were also taken into consideration.

Analysis

In this section, the related work will be analyzed, for fairness and as close to an apple to apple comparison when comparing the reported results only papers that were submitted to the ISIC-2017 challenge will be considered. This is deemed to be fair as the dataset and target are the same across these papers, only varying on some of the methods used for the feature extraction and classification of the images. On the other hand, due to the nature of the dataset which contains only dermoscopic images, no real conclusions can be taken for clinical images. To address this a further paper will be analyzed, as it is considered as a landmark paper, and is the paper that best approaches the proposed system in terms of trained classifications.

Table 3.2: Unrestricted papers analysis:
(A) Feature extraction method; (B) Data augmentation;
(C) Image pre-processing; (D) Image segmentation;

(A)	All papers
CNN	26
Deep Residual Network (DRN)	1
Ensemble	1
Manual (programatically)	24

(B)	All papers	CNN papers
YES	27	22
NO	27	4

(C)	All papers	CNN papers
YES	39	17
NO	15	9

(D)	All papers	CNN papers
YES	31	9
NO	23	17

Due to the observable in Table 3.2, the trends across all papers in the field tend towards the use of Convolutional Neural Networks (CNN) over other methods, towards the use of image pre-processing, and towards image segmentation. Once we consider the CNN based systems however the trends are slightly different. In this case the trends lean towards the use of Data augmentation and image pre-processing as well as not using segmentation. The only metric that will be used for the analysis of the state of the art is the Area Under Curve (AUC) as it is the only metric present in all the papers.

When analyzing the related work with the aforementioned restrictions we are reduced to 4 papers.

Table 3.3: Restricted papers analysis

Author of Paper	Pre-Processing	Augmented	Segmented	Classification
Adrian Galdran et.al [22]	No	Yes	No	Ensemble
Fábio Perez et.al [63]	No	Yes	Yes	Softmax
Balazs Harangi [30]	Yes	Yes	No	Softmax
Yuexiang Li and Linlin Shen [44]	Yes	Yes	Yes	Ensemble

These papers, as mentioned before, all used CNNs to extract the features of the images. The task they are attempting to solve is the three-way classification of Melanomas, Soborrheic Keratosis, and Nevi, using the ISIC-2017 challenge data-set.
All papers, with the exception of the last, use transfer learning with their particular neural networks using Imagenet weights for this, achieving an AUC of 87%.

The paper by Adrian Galdran *et al.* [22] does not report using image pre-processing, although they probably resized the images for input in the ResNet [32] architecture used. This approach used data augmentation with the following techniques: 1. Rotations, 2. Scaling of the image, 3. Lighting correction, 4. White balance, 5. Gamma correction, 6. Color manipulation, and 7. Non linear transforms. Segmentation was performed via a U-Net [66]. The classification itself is done by taking the maximal value from the output layer of the ResNet neural network, achieving an AUC of 88%.

In the paper by Fábio Perez *et al.* [63] multiple CNNs are tested. The best performing neural network proved to be an Inception v4 [76] and, as for pre-processing, this reports resizing the images for input. There is a report of the usage of data augnmentation using the following methods: 1. Rotations, 2. Flips of the image, 3. Lighting correction, 4. Gamma correction, 5. Color manipulation, 6. Random erasure of section of the images, 7. Noise was added to the images, 8. and Non linear transforms . In this work no segmentation was used. Again like the previous paper classification was done by taking the max value of the output layer, achieving an AUC of 93%.

In Balazs Harangi's paper [30] an ensemble of CNNs was used. In this ensemble the following architectures were used: 1. GoogLenet [74], 2. ResNet [32], 3. Alexnet [43], 4. VGG19 [69]. Again in this paper, no mention of image pre-processing was present but, like Adrian Galdran *et al.* paper [22], the resizing of the input images was most likely done.
As for data augmentation, this was very basic using only rotations and image flipping, and no segmentation was used. As for the classification itself, this was done in the following manner: each one of the neural networks sends their confidence level, the result of the output layers, to

a function which, for each class, calculate the sum of each of the classes. The class with the maximal value is taken as the final classification.

In the last paper, Li and Shen's [44], a novel approach for this task is adopted. The paper reports a framework created by the authors for the classification of skin lesions, this framework is called Lesion Indexing Network (LIN). As for image pre-processing, the paper reports the cropping and resizing of the images. The reported data augmentation was, like in the previous papers, done in a very simple manner with the usage of rotations and flips.
The segmentation in this paper was done using a fully convolutional residual Network (FCRN), which is a modified ResNet first proposed in the author's paper [45]. For feature extraction, while being a CNN, the architecture used is not reported being referred to as Lesion Feature Network (LFN). The proposed framework is composed of two FCRNs for the creation of both the segmentation and, as the authors call it, a "coarse" classification prediction. This is then passed to another element called the lesion index calculation unit (LICU). This LICU is used to refine the previous "coarse" classification via the calculation of a distance heat-map. The "coarse" classification is obtained from the class-wise sum of the outputs from the FCRNs. The LICU system is defined as:
Let $v_i(x, y)$ be the value of (x, y) in ith coarse map, the normalized probability for each of the possible classifications (p_i) can be deduced by:

$$p_i(x, y) = \frac{v_i(x, y) - min_i(v_i(x, y))}{\sum_{i=1}^{3}(v_i(x, y) - min_i(v_i(x, y)))}, \ i \in 1, 2, 3, \tag{3.1}$$

with i being the possible classification. This work achieved an AUC of 91%.

This analysis can lead us to accept that the data augmentation and image processing steps are in fact important for this classification task. In addition, an ensemble of CNNs appears to be a good method for the classification of skin lesions, as it achieves the best AUC which is an indicator of the diagnostic ability of the model.

Chapter 4

Design

This chapter contains a description of how the work put forth was designed and of relevant development decisions that were made, as well as of integration of the current state-of-the-art in the proposed task.

Data

Arguably the most important factor in this work is the decision of what kind of data to use, its sources, and finally how reliable the labels for the data are. Since no data was collected in hospitals and clinics, the source of the data would be from publicly available online repositories. Due to the characteristics of the available data and the project's intentions, both clinical and dermoscopic images are used. Dermoscopic images, due to their characteristics, tend towards providing a better image of the relevant lesion features, whilst clinical images provide a better relationship between the potentially diseased skin area and normal skin, as well as being more readily available for the target users.

From the multiple repositories found online both clinical and dermoscopic images are present. These repositories are designed to serve as teaching aids for aspiring dermatologists, as such, the labels, which are given by medical experts in the area, are trusted. As on the quantity of data and its sources, this information is aggregated in Table 4.1. This data, however, was not all used, as it would be incompatible with the time for this project the creation of a system capable of classifying the hundreds of different possible diagnoses as well as, due to the rarity of some of these diagnoses. Therefore we reduced the number of observations in some of these classes. As such, the decision was made to focus on the carcinogenic lesions and some of the most prominent classes that appear in the differential diagnosis that a dermatologist would consider given the visual analysis of the lesion.

These images have some characteristics that must be taken into consideration whether or not they would be of interest for the proposed system. For one, the images from Derm101 [7] have a watermark that is superimposed on the image and over the lesion itself. Something similar happens with the images from DermNetNZ [17], but these watermarks rarely are on the lesion itself. In spite of the presence of watermarks in the DermNetNZ subsection of the images, these were used if the watermark did not overlap the lesion and in similar numbers across all classes to help reduce the effect these would have on the classification.

Table 4.1: Data repositories information

Repsository	Number of images	Annotated	Segmentation Anotations
ISIC-Archive [36]	>23000	Yes	Yes
Derm101 [7]	>23000	Yes	No
DermNetNZ [17]	>20000	Yes	No
Atlas Dermatologico [16]	>9000	Yes	No
Danderm [59]	>3000	Yes	No
Hellenic Derm Atlas [2]	>2500	Yes	No
derm.cs.sfu.ca [41]	>1000	Yes	No
Skin Cancer 909 [65]	>500	Yes	No
PH2 [54]	200	Yes	Yes
MED-NODE [23]	170	Yes	No
Skin Cancer UK [12]	11	Yes	No

To this effect the images used were taken, with the copyright owner's permission, from the following sources via scrapping or direct download when possible:

- ISIC-Archive [36]
- DermNetNZ [17]
- Atlas Dermatologico [16]
- Danderm [59]
- Hellenic Derm Atlas [2]

Whilst additional data could potentially be useful in the classification efforts said information is not present for all the images. Said data could be, as is the case of the ISIC-Archive, pertaining to the patients age, gender, and lesion location. Since this information is not present for all the images, as well for the aforementioned security concerns, it has been ignored in this work.

Classes

As previously mentioned, this work focuses on a small number of possible diagnoses. The singled out possible diagnosis are those which could be considered the most dangerous, namely basal cell carcinoma, squamous cell carcinoma, and melanoma, as well as lesions that must be ruled out when a dermatologist is performing a diagnosis, namely, melanocytic nevus and seborrheic keratosis. These were selected taking into account a second factor, this being the abundance of good quality images. Taking into account the possible diagnosis, i.e., Merkel cell carcinoma, another type of malignant skin lesion which is so rare that a very reduced amount of samples was found, would be problematic as it would require either a one-shot-learning approach to the training or a very aggressive data-augmentation protocol that would most likely provide very little generalization in the class. Efforts were made to keep the number of images balanced from each source so that the impact of the watermarks could be minimized. The distribution of the different classes used, across the different repositories is as described in Table 4.2.

Table 4.2: Distribution percentages for the different classes used in the repositories

Nevus, multiple kinds (NV)	:80.72%
Melanoma (MM)	: 9.43%
Basal Cell Carcinoma (BCC)	: 2.55%
Seborrheic Keratosis (SK)	: 1.82%
Squamous Cell Carcinoma (SCC)	: 0.98%

Pre-Processing

Multiple pre-processing techniques were experimented with. From the multiple techniques available, selected from the state-of-the-art, from which multiple protocols were created. These protocols received the names pre-naive (Table 4.3), pre-contrast (Table 4.5), and pre-histogram (Table 4.4). The selection criteria for the pre-processing method to use was a combination of the range of the resulting accuracies as well as if the neural network would tend towards under fitting on the generated images.

Table 4.3: Pre-naive pipeline explanation

Technique	Comments
Aspect Ratio fixer	Makes sure the aspect ratio is 1×1 so that it can then be resized, this happens by cutting out the larger axis from both directions by an equal amount of pixel rows.
Resize	Resizes the images to 300×300 to help save on storage space, uses a bilinear interpolation.
Gaussian de-noising	Helps ensure all images have minimal signal noise.
Color balance	Color balancing algorithm that uses histogram normalization, source from [38]. Saturation percentile set to 1.
Contrast correction	This contrast correction uses the LAB color space to apply CLAHE technique.
Gamma adjustment	Adjusts the gamma of the image using a gamma differential of .6
Brightness and contrast adjustment	Uses a polynomial function to adjust the brightness and the contrast of the images, uses an alpha of 1.8 and a beta of 70, follows the function: $I(p) * alpha + beta$ with I being the value for the pixel inputted, operates on all channels.
Color balance	As the previous one. Saturation percentile set to 1.
Contrast correction	As the previous one.

Whilst the images produced by the pipeline described on Table 4.5 appear to visually normalize the lesions and enhance the features of the lesions, the accuracy of this pipeline varies wildly when the images are used in a neural network, and has a tendency to underfit the model. Therefore this pipeline was discarded.

Table 4.4: Pre-histogram pipeline explanation

Technique	Comments
Gaussian de-noising	Helps ensure all images have minimal signal noise.
Aspect Ratio fixer	Makes sure the aspect ratio is 1 × 1 so that it can then be resized, this happens by cutting out the larger axis from both directions by an equal amount of pixel rows.
Resize	Resizes the images to 300 × 300 to help save on storage space, uses a bilinear interpolation.
Contrast correction	This contrast correction uses the LAB color space to apply CLAHE technique.
Histogram equalization	This does histogram equalization on each of the channels.

Whilst the images produced by the pipeline described on Table 4.4 appear to visually not change the images much, the accuracy of this pipeline did not vary as wildly as the previous pipeline, (table 4.3), when the images are used in a neural network. This pipeline was discarded for reasons explained in the discussion of the next pipeline.

Table 4.5: Pre-contrast pipeline explanation

Technique	Comments
Gaussian de-noising	Helps ensure all images have minimal signal noise.
Aspect Ratio	Makes sure the aspect ratio is 1 × 1 so that it can then be resized, this happens by cutting out the larger axis from both directions by an equal amount of pixel rows.
Resize	Resizes the images to 300 × 300 to help save on storage space, uses a bilinear interpolation.
Contrast correction	This contrast correction uses the LAB color space to apply CLAHE technique.

Whilst the images produced by the pipeline described on Table 4.5 appear to visually not change the images much, as well as being very similar to the previous pipeline (Table 4.4). The accuracy of this pipeline did not vary as wildly as the previous pipeline, (Table 4.4), when the images are used. This pipeline was selected over the previous as the this pipeline had very similar accuracies and a smaller difference between the lowest and highest accuracy.

Data-Augmentation

Data augmentation was performed as a method to attempt to give the neural network some properties, such as translational invariance, rotational invariance, and on an early stage quality invariance. Quality invariance can be defined as a method of adding noise to an image, normally Gaussian noise, in an attempt to ensure that the neural network would be able to handle these kinds of images. However, due to the de-noising step of the pre-processing, this was removed from the data preparation as these types of noises, which are additive in nature, can be diminished greatly in the pre-processing stage. Another factor that affected the decision to drop this in the data-augmentation stage is the amount of images this would add, which for the time-frame available could make the training process last too long. The quality variation on the images taken, different cameras with different RGB sensors, means that a measure of quality invariance is should be present in the trained models.

Table 4.6: Data augmentation pipeline

Method	#Images generated	Comments
Rotate	4	Rotates in 90º, 180º, 270º, and 360º
Mirror the image	5	Mirrors each of the previous images
Shift the image	10	Shifts each image in both axis in a random amount of up to 75 pixels in any direction

This process generates 19 augmented examples from each image, labeled as the originating image, as described in Table 4.6. These methods were extracted from the state of the art analysis performed, and applied to each subset of the data used.

Segmentation

Finding the optimal approach for leseion segmentation is a problem too vast to be properly addressed in this work. For this reason, the analysis of lesion segmentation algorithms was not considered in this work. The problem stemmed from the fact that when segmentation is used in the state of the art, it is applied to a simpler problem, the segmentation of pigmented lesions, normally melanoma vs nevus. However, due to the less focused approach in this work as to the type of lesion, choosing to attempt to classify other possible diagnosis such as basal-cell carcinoma and squamous-cell carcinoma, the segmentation is not a trivial problem to be solved, as the lesions can be very similar in characteristics to healthy skin. This leads to the need of a previous classification of the lesion before the segmentation, which defeats the purpose of this work.

Selected CNNs

Here we present a description of the individual neural networks selected from the state of the art analysis. When possible, a full description of the the layers contained in each of these networks will be given. However, due to the depth of some of the selected networks, some will be explained in a more general manner. The models used were taken from Keras.applications package as it provides an easy way to use and obtain weights from their submissions to ImageNet [67] competition for transfer learning.

VGG-16

Table 4.7: Details for the VGG-16 network

Number of Parameters	Number of Trained Parameters	Number of Layers	Number of Convolution Layers	Number of Pooling layers	Pooling used
134,281,029	134,281,029	24	13	5	Max Pooling

This neural network was first proposed by Simonyan, Karen and Zisserman, Andrew in 2014 [69]. This neural network is composed of five blocks of convolution layers in the following manner:

1. (3 ,3) Convolution layer
2. (3, 3) Convolution layer
3. (2, 2) Max Pooling layer with strides (2, 2)

This layout is exactly the composition of the first two blocks, whilst the other three contain an extra convolution layer before the pooling layer. As for the number of filters, these are the numbers. Block 1: 64 filters; Block 2: 128 filters; Block 3: 256 filters; Block 4: 512 filters; Block 5: 512 filters.

VGG uses ReLu activations on all of its layers, as input RGB images with a resolution of 224 by 224 pixels. The convolutional blocks work to extract features from the input however, for classification, a three layered dense net is used, with the output layer using a softmax activation for the output generation. This dense net has the following structure:

1. Flatten the 2D features extracted
2. Dense layer with 4096 nodes
3. Dense layer with 4096 nodes
4. Dense layer with x nodes, x is number of classes in the classification task

This network achieved a top-5 error of 6.8% in the ImageNet dataset [67]. The following figure (Figure 4.1), contains a graphical visualization of this neural network's architecture.

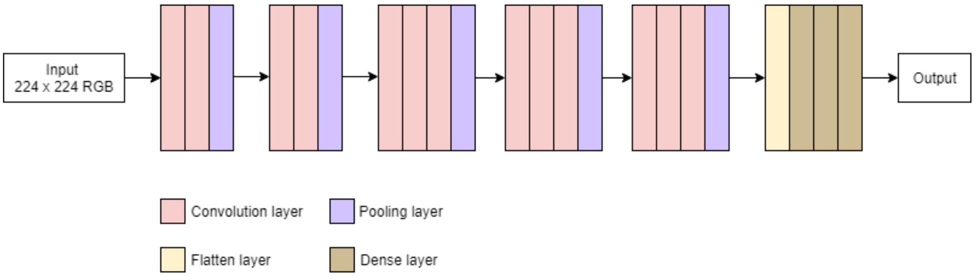


Figure 4.1: VGG-16 Architecture

Inception

Table 4.8: Details for the Inception V3 network

Number of Parameters	Number of Trained Parameters	Number of Layers	Number of Convolution Layers	Number of Pooling layers	Pooling used
21,802,784	21,768,352	312	94	14	Max, Avg Pooling

The inception neural networks are developed by Google. First known as GoogLeNet this was described by Christian Szegedy in [74], where a new kind of module was introduced, the inception module. The inception module consists of a separation of the input into multiple branches which are then concatenated, Figure 4.2.

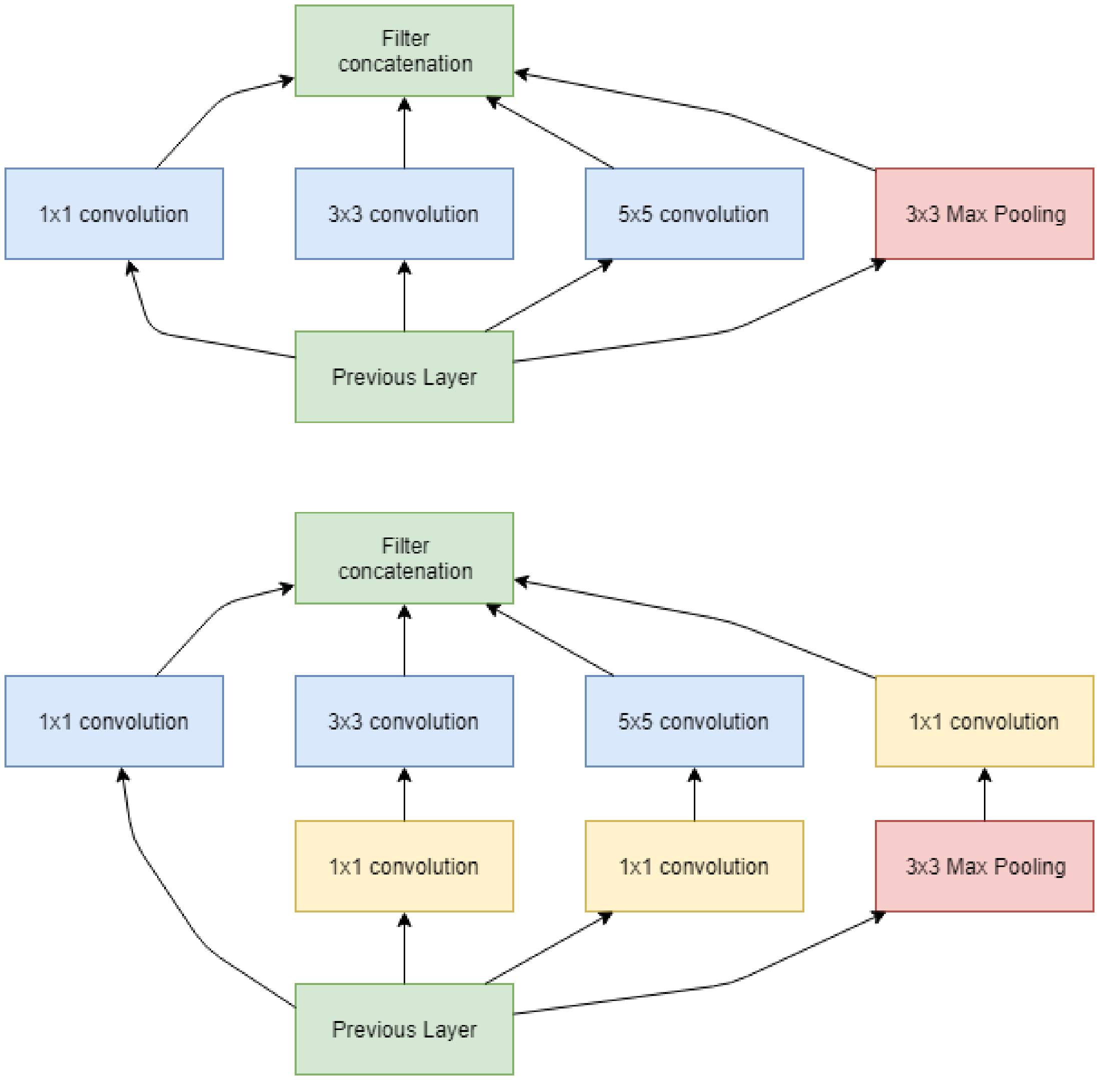


Figure 4.2: Description of the Inception module. Top: a naive implementation of the proposed module. Bottom: a visualization of the same principle with dimensionality reduction.[1]

Later, an alteration was proposed to these modules to improve upon their efficiency [75] by factorizing the filter matrices, with small impact to the performance of the network, reporting that the employment of this factorization does not perform well in early layers but is effective on medium feature maps defined as feature maps of shape $M \times M$ with M ranging between 12 and 20 [75]. This alteration consists in making the convolution process itself more efficient. This helps with the computational complexity of the algorithm which in turn allows for more expedient completion times. This obviously allows for a shorter experimentation period and more time for result gathering/analysing for fixed time projects. This technique can potentially be extended to other networks. This process works due to the properties of matrix factorization and matrix multiplication.

Moreover the inception block, as described in the bottom of Figure 4.2, was created with a further motivation of allowing the convolutional layers to operate at different scales concatenating the result of these scale differences to be consumed by the following sections of the network. This allows for both a capability to extract broad and detailed features from the image.

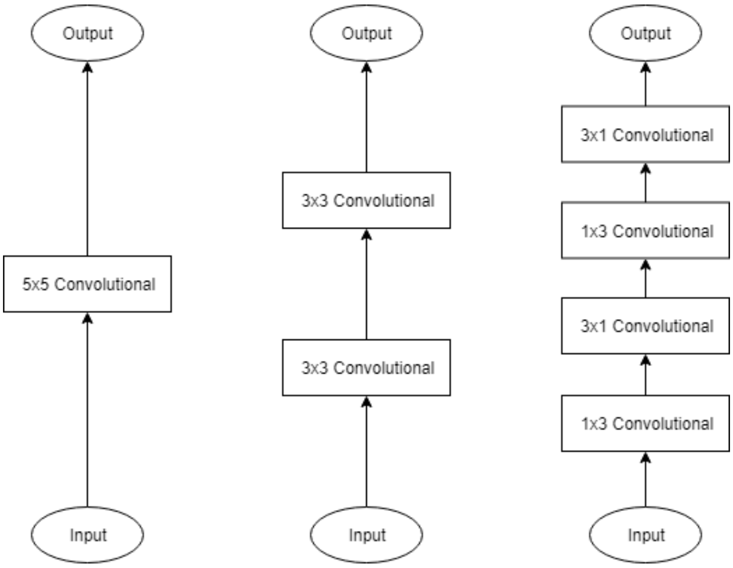
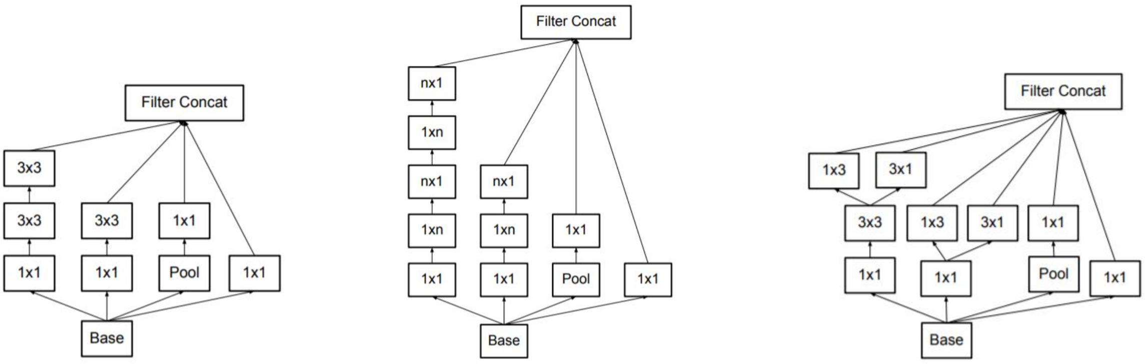


Figure 4.3: Equivalent convolutional operations

This lead to the creation of different building blocks used in the architecture implemented in the inception v3 neural network, shown in Figure 4.4.



(a) Substitutes a 5×5 with two 3×3 convolutions

(b) Full $n \times n$ factorization used with n=7 for 17×17 grids

(c) Expanded filter bank outputs for promoting higher dimentional representations

Figure 4.4: Different inception blocks as suggested in the paper [75]
Taken from the paper

The network uses ReLu activation and the complete Inception v3 architecture is described in the following table as per the Szegedy's specification [75]:

Table 4.9: Inception v3 neural network architecture

Type of layer	Kernel size/block used	Stride
Convolutional	3×3	2
Convolutional	3×3	1
Convolutional	3×3	1
Pooling	3×3	2
Convolutional	3×3	1
Convolutional	3×3	2
Convolutional	3×3	1
3 Inception blocks	Block type (a)	-
5 Inception blocks	Block type (b)	-
2 Inception blocks	Block type (c)	-
Pooling	8×8	-
Dense	Logits	-
Softmax	Classifier	-

This neural Network achieved a Top-5 Error of 4.2% in the ImageNet dataset [67] placing amongst top positions for the networks tested in this dataset.

ResNet

Table 4.10: Details for the ResNet50 network

Number of Parameters	Number of Trained Parameters	Number of Layers	Number of Convolution Layers	Number of Pooling layers	Pooling used
23,597,957	23,544,837	178	53	2	Max Pooling

This neural network was first proposed by He, Kaiming and Zhang, Xiangyu, in 2015 [33], as part of a novel way of creating **CNNs!** (**CNNs!**), as it takes advantage of both the convolutional operations and residual vectors from the images. It is composed of a combination of two distinct types of blocks whose composition can be seen in Figure 4.5.

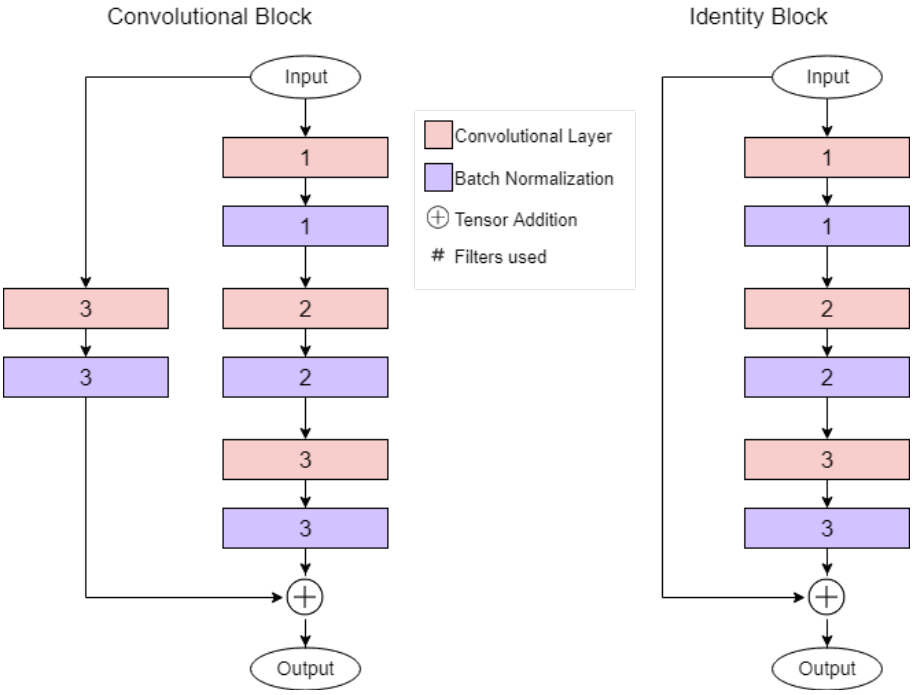


Figure 4.5: Blocks that serve as the main building blocks for the Resnet neural network, whilst being the same structure, a different number of filters is used on different depths.

Identity blocks are so named due to computing the residual with respect to a single copy of the input. Convolutional blocks, however, compute the residual with respect to the output of further convolutional layers.

The version of this network which was used in this work was the ResNet50 which is so called from the number of layers used. This name was coined in [33], which introduced these neural networks. In the same document a top-five error of 5.25% was reported on the ImageNet dataset [67].

Table 4.11: ResNet50 Architecture

Layer/Block type	Filters 1	Filters 2	Filters 3
Convolutional layer	64	-	-
Batch normalization	-	-	-
MaxPooling	-	-	-
Convolutional Block	64	64	256
2× Identity Block	64	64	256
Convolutional Block	128	128	512
3× Identity Block	128	128	512
Convolutional Block	256	256	1024
5× Identity Block	256	256	1024
Convolutional Block	512	512	2048
2× Identity Block	512	512	2048
Global Average Pooling	-	-	-
Dense	# Classes	-	-

In Table 4.11 The column number are respecting to the numbered layers of Figure 4.5, which is to say filters 1 is used in the layers numbered 1.
This version of the ResNet neural network uses ReLu activations, with a kernel size of 3 and, when applicable, a stride of (2 , 2), and the architecture is as described in the previous table 4.11.

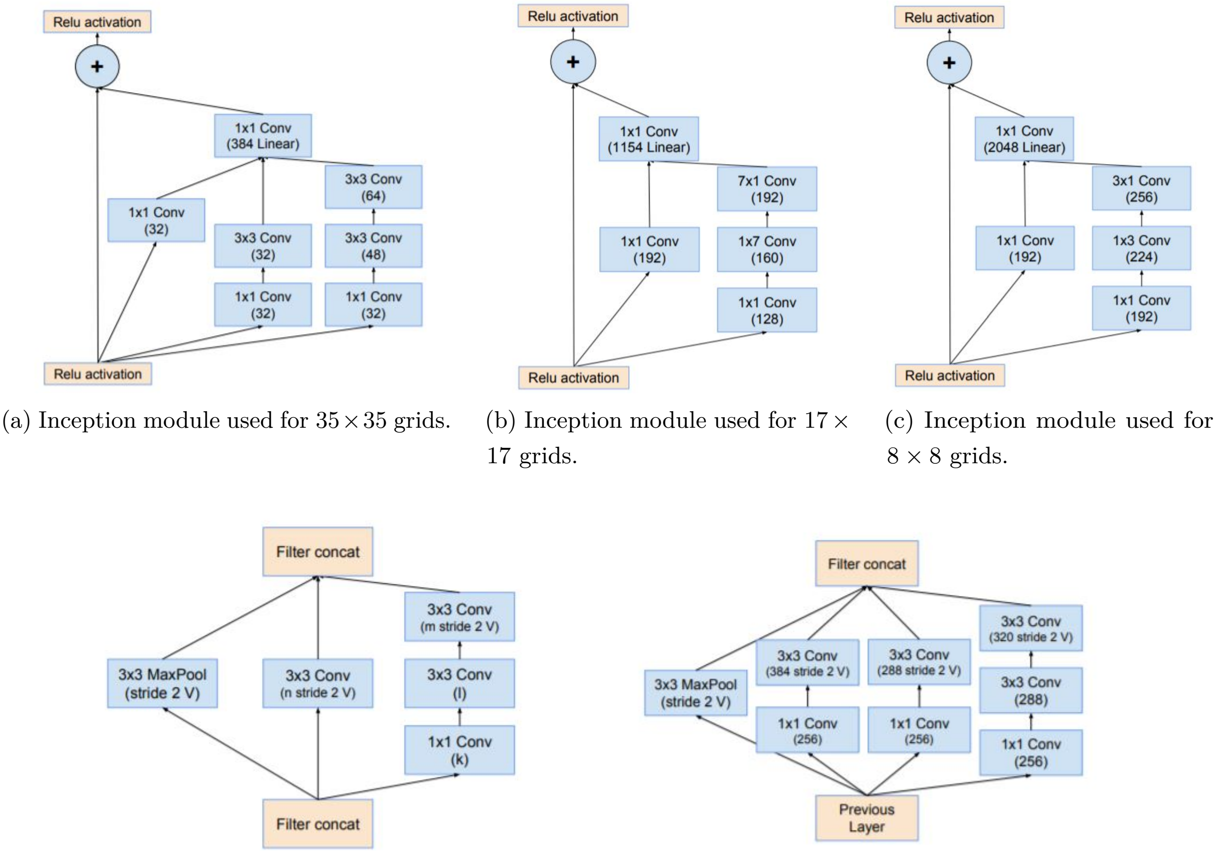
InceptionResnet

Table 4.12: Details for the InceptionResnet V2 network

Number of Parameters	Number of Trained Parameters	Number of Layers	Number of Convolution Layers	Number of Pooling layers	Pooling used
54,344,421	54,283,877	783	244	6	Max, Avg Pooling

This neural network was first proposed by Christian Szegedy [76] as a further improvement on the inception networks, described in the above Section 4.5.2, by combining the inception modules with residual learning concepts from the resnet architecture, described in the above Section 4.5.3.

To achieve this, multiple new formulations of the inception modules were created and split into two groups: [76], inception modules and reduction modules represented in Figure 4.6.



(a) Inception module used for 35 × 35 grids.

(b) Inception module used for 17 × 17 grids.

(c) Inception module used for 8 × 8 grids.

(d) This reduction module reduces 35 × 35 grids to 17 × 17 grids.

(e) This reduction module reduces 17 × 17 grids to 8 × 8 grids.

Figure 4.6: Different inception and reduction modules as suggested in the paper [76]
With k=256, l=256, m=384, and n=384 being filter sizes.
Taken from the paper

The modules in Figure 4.6 are used as the building blocks for the main feature extractor of the neural network. Before these, there is another module that is referred to as the stem module of the neural network Figure 4.7.

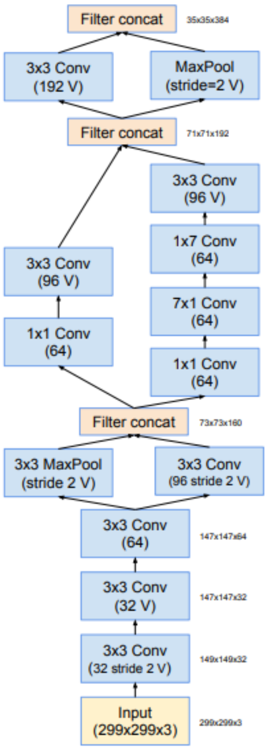


Figure 4.7: Stem Module
Taken from the paper[76]

This neural network architecture is described in Table 4.13. This network achieved 4.1% top-5 error score on the ImageNet dataset [67], therefore being the most successful among the neural networks considered in this work for the classification of the ImageNet dataset [67].
.

Table 4.13: InceptionResnet-v2 Architecture

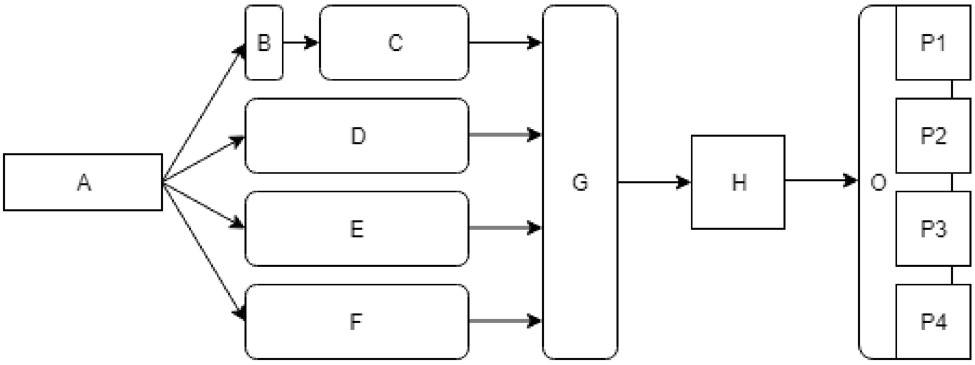
Layer/Module type as in Figure 4.6	Comments/Filter Sizes
Input	Image 299 × 299 pixels
Inception (a)	5 modules
Reduction (d)	-
Inception (b)	10 modules
Reduction (e)	-
Inception (c)	5 modules
Average Pooling	-
Dropout	Rate: 0.2
Dense	# Classes

Ensemble Structure

Table 4.14: Details for the Ensemble network

Parameters	Trained Parameters	Layers	Convolution Layers	Pooling layers	Pooling used
236,614,208	236,466,112	1304	404	27	Max, Avg Pooling

The ensemble proposed is mostly an effort to parallelize all of the aforementioned networks into a single trainable neural network.



A - Input images 299x299 px
B - Image resize 224x224 px
C - Neural Network VGG
D - Neural Network ResNet
E - Neural Network Inception_v3
F - Neural Network InceptionResNet

G - Concatenation
H - Classification dense layers
O - Output Vector
P#- Probability for class #

Figure 4.8: Ensemble Architecture

The neural networks being used in tandem in this ensemble have their classification sections removed, this means that the outputs concatenated in G are the outputs from the last pooling layers of each of the aforementioned networks. Furthermore the ensemble is trained in a single back propagation training the whole ensemble as a single neural network. The general structure of the ensemble can be seen in Figure 4.8. Block B, uses TensorFlow's resize_images function, which applies a bi-linear interpolation to reduce the size of the images. Block H's definition is in Table 4.15.

Table 4.15: Block H structure

Layer type	Comments/Filter Sizes
Dense	5000 nodes
Dropout	rate = 0.2
Dense	500 nodes
Dense	Softmax

This network uses the ReLu activation in Block H and has not been tested against the ImageNet dataset.

Implementing the system

Efforts were made towards the practical implementation of this system as a service which could be accessed externally from the machine running the neural networks. This was done via the creation of a server running a RESTful API [20] that would receive a JSON [9] object with a Base64 [42] encoded image, via HTTP [19]. The system would then decode the image and send it to the neural network for classification. This classification would then be sent back to the original requester. Due to a difficulty found with the permanency of the model in memory, the model had to be loaded again for each request, thus adding a big overhead to the systems responsiveness. The following scheme describes the system:

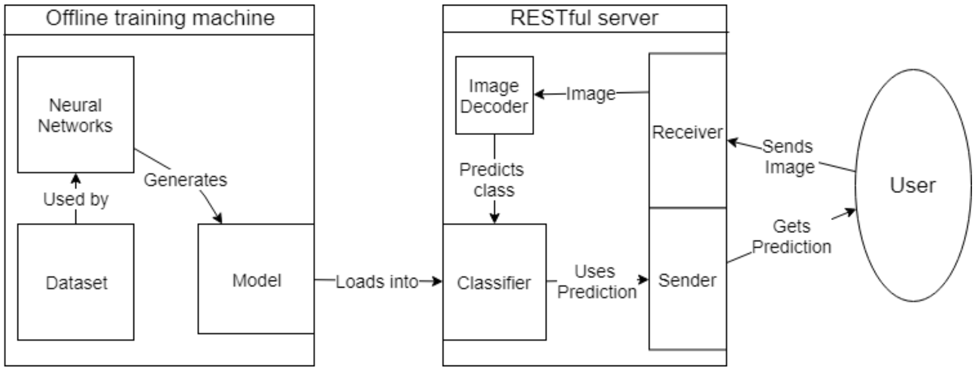


Figure 4.9: Servicing the system

Figure 4.9 illustrates how the system could be serviced. This kind of system distribution was implemented, and is capable of delivering the prediction to the user in up to 2 minutes after the image being sent: this time includes the loading of the model and classifying the image, whereas classification itself took around 45 milliseconds.

Chapter 5

Tests and Results

In this chapter, we present a description of the different tests performed as well as a report on the results of said tests, with a forthcoming analysis of the results.

Tests

Methodology

This section contains a discussion on the testing methodology used in this work, as well as the rationale behind each methodological choice.

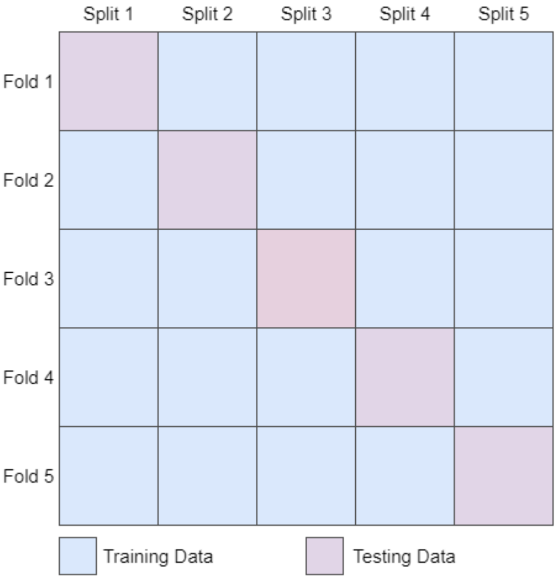
Testing Methodology

Validation is an important step in model selection, which uses a set of images that are "new" to the network, meaning that they were not used for training. Many techniques can be used for validating a model, the chosen one was 5-fold cross-validation, which involves the creation of 5 "folds", i.e. 5 sub-datasets generated from a single dataset. However, in this work, we are not using this technique for the usual goal of model selection, where multiple models from the same architecture or algorithm are generated and compared to each other in each of the folds to pick the best one. We are using this technique to test the performance of multiple architectures concerning their performance averaged across the multiple folds, as a way to determine which model can be expected to perform better for this task. The validation data used for the training phase of each of these networks is a subset of the training set generated using the n-fold cross-validation. The folds were created in the following manner, from the original dataset:

1. Randomise the order of the dataset.
2. Split the shuffled dataset into 5 separate subsets, with the union of all subsets being equal to the full dataset.

3. On each of the subsets a check is run, this check calculated the count of instances for each of the classes ensuring that the difference between the minimum value and the maximum value is lower than an arbitrary value, the value 20 was selected. Should this fail the folds would reset, and the process starts again.

The arbitrary value mentioned in point 3. is motivated by the difficulty of generating an exact split with the same number of examples per class, as there was no guarantee that the full dataset would have the same number of images per class. The folds will then be split like the following visualization:



	Split 1	Split 2	Split 3	Split 4	Split 5
Fold 1	🟪	🟦	🟦	🟦	🟦
Fold 2	🟦	🟪	🟦	🟦	🟦
Fold 3	🟦	🟦	🟪	🟦	🟦
Fold 4	🟦	🟦	🟦	🟪	🟦
Fold 5	🟦	🟦	🟦	🟦	🟪

🟦 Training Data 🟪 Testing Data

Figure 5.1: Training and testing data splits

As can be seen in Figure 5.1, the process described above ensures that no image is in any two different testing data splits, as well as ensuring that every image is at some point used for testing. This provides us with five different sets of weights for each different neural network which allows not only to understand which model provides the best results but also which models is the most stable so that a measure of confidence that this model would be a good fit for this kind of problem as it would appear to be able to provide with stable results for different data points on the same problem.

Confusion Matrices

Confusion matrices are a common data representation for the output of statistical classification algorithms, such as neural networks, which allow for a quick look up on multiple metrics.

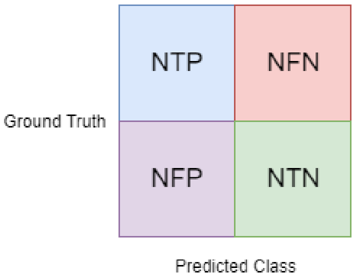


Figure 5.2: Confusion matrix,
NTP, NTN,
NFP, NFN

These are usually applied to, and designed for, binary classification problems. However these can be generalized for multiclass classification, i.e., a three class classification confusion matrix can be reduced into three binary classification confusion matrices (Figure 5.3).

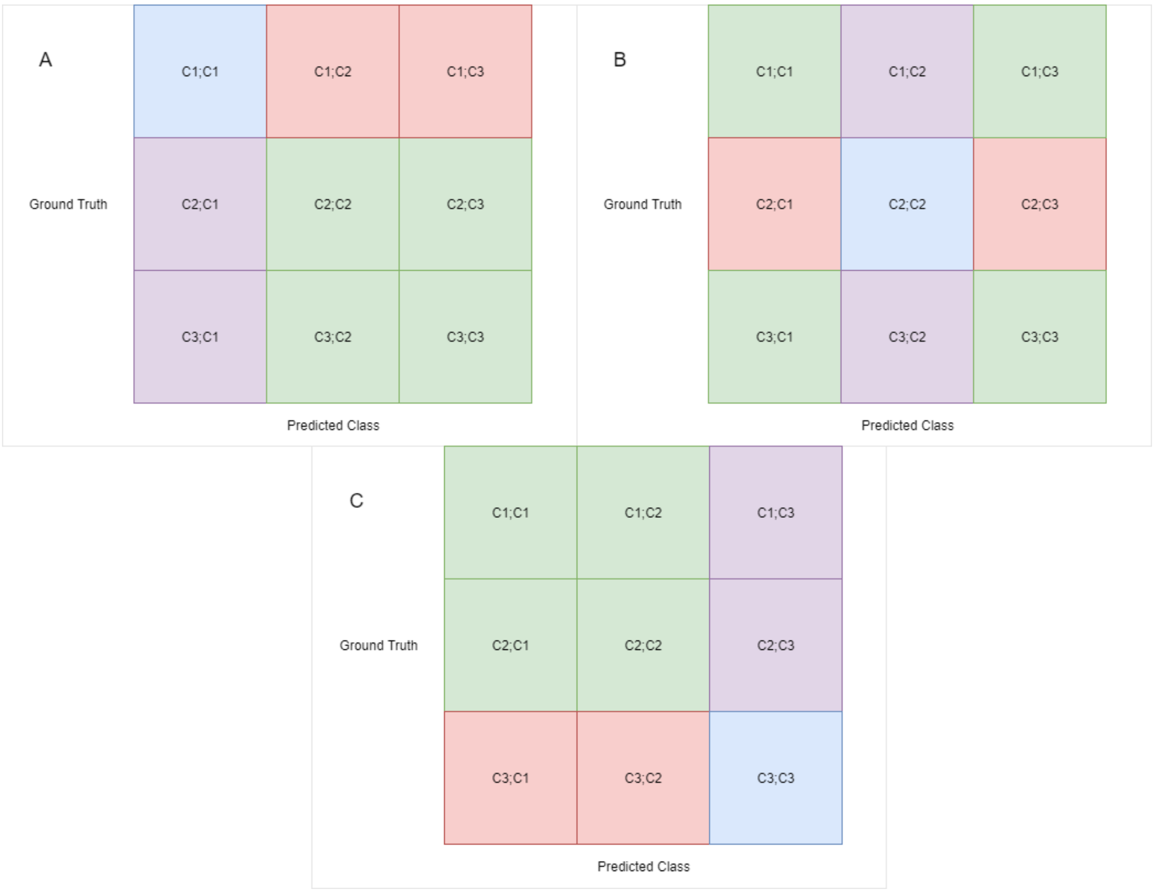


Figure 5.3: Collapsing of a N way confusion matrix to a binary confusion matrix

In Figure 5.3 the structure of a confusion matrix is represented. The colours are in line with the previous example of a confusion matrix, as seen in Figure 5.2

The reduction is done by adding up all the squares of the same colour and using those values as the true values for the binary problem for each class, basically turning a class A, class B, class C problem into a A vs not A, B vs not B, C vs not C. Consider Figure 5.3 A:

- NTP: has the same value as the blue square, i.e., number of samples from class C1 classified as C1.
- NTN: this value will be the sum of: $(C2; C2) + (C2; C3) + (C3; C2) + (C3; C3)$, i.e., number of samples from classes C2 or C3 that are classified as either C2 or C3.
- NFP: this value will be the sum of: $(C2; C1) + (C3; C1)$, i.e., number of samples from classes C2 or C3 that are classified as C1
- NFN: this value will be the sum of: $(C1; C2) + (C1; C3)$, i.e., number of samples from class C1 that are classified as C2 or C3.

Such reduction results in a matrix similar to the one in Figure 5.2. However, this gives the confusion metric of the binary classification problem, not the multiclass classification problem. In order to obtain the desired metrics for the multi-class classification problem, we can average over the metrics obtained for the different binary classification problems considered in the previous reduction procedure.

The metrics that were considered were: Accuracy, Sensitivity, Specificity, F1-score (Table 5.1).

Table 5.1: Metrics collected

Accuracy	$= \frac{NTP+NTN}{NTP+NTN+NFP+NFN}$	Fraction of samples classified correctly
Sensitivity	$= \frac{NTP}{NTP+NFN}$	Fraction of correct classifications of a class from all samples classified as that class.
Specificity	$= \frac{NTN}{NTN+NFP}$	Fraction of correct classifications as not a class from all samples classified as not being from that class.
F1 Score	$= 2 * \frac{NTP}{NTP+NFP+NFN}$	A measure for the model's accuracy, which represents a tradeoff between sensitivity and specificity.

Testing strategy

This section contains a description of the studies performed, as well as the resources used for this purpose. The computer used to train and run the neural networks has the following characteristics:

Table 5.2: Computer specs for the computer used

Component type	Component model	Comments
CPU	i5 7600	3.5 GHz clock speed, water cooled
GPU	NVIDIA GTX 1080 TI	3586 Cuda cores, 11GB GDDR5X vram, 1582 MHz clock speed
Ram	DDR4	16Gb, 2.4 MHz clock
HDD	Toshiba 2TB	7200RPM

All testing was done in this machine as this allows for a performance comparison between the different neural networks, with the only varying factor being the neural network being trained itself.

Due to a large number of images and the time limitations there was a need to create a subset of the data available, the first subset was approximately 1200 images split amongst 5 different classes, Basal Cell Carcinoma (BCC), Melanoma (MM), Nevus, multiple kinds (NV), Squamous Cell Carcinoma (SCC), Seborrheic Keratosis (SK). A first study was conducted using the images of this first dataset. The purpose of first study being mainly to discover if the neural networks were, in fact, able to learn to classify the different classes, discover any classes that could be joined to improve upon the accuracy of the system, and finally to find a reasonable set of hyper-parameters that would work well with all the neural networks in testing.

Then, a second, extended dataset was considered. In this case, nearly 4000 images are split equally over 4 different classes, these being: MM, Non-Melanoma carcinoma (NMC), NV, and SK. This new sets of classes is obtained by merging in a single class NMC both BCC and SCC images This choice has been motivated by the observation that these where the classes the neural networks trained in the previous study had the most difficulty in separating, another factor is that this would allow for the number of samples in each class to be balanced which would help alleviate any biases learned by the networks. From a clinical point of view once there is a suspicion of either a BCC or SCC diagnosis then the gold standard for the diagnosis is a biopsy [57], therefore this change, while altering the descriptive capabilities of the neural network, should not be disruptive.

Note that the images selected in both the first and second dataset were picked due to their quality, with the effort made to avoid watermarked images, and when this was not possible, to ensure that the number of watermarked images was balanced across the multiple classes also ensuring that said watermarks did not intersect with the lesion itself, as well as allowing for similar splits of dermoscopic/clinical images, whilst this was not always possible due to the available amount of images for some of the classes.

A second study is conducted over the second dataset. The purpose of the second study was first, to apply and confirm the findings of the first study over a larger dataset, as well as showing that an increase in the number of images used for training could lead to a better generalization ability of the considered neural networks.

Finally, the proposed network ensemble is tested over the second dataset, in what is called an ensemble study. The purpose of this study is to determine if the ensemble strategy would be a good approach to this problem, which is to say if this approach would improve upon the systems metrics and stability.

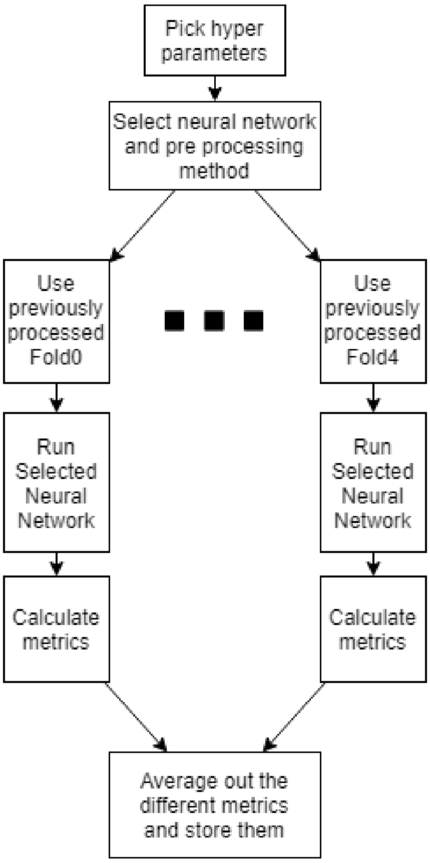

Pick hyper parameters
Select neural network and pre processing method
Use previously processed Fold0
Use previously processed Fold4
Run Selected Neural Network
Run Selected Neural Network
Calculate metrics
Calculate metrics
Average out the different metrics and store them

Figure 5.4: Study workflow

Figure 5.4 details how the acquisition of metrics, as well as how the studies are performed.

Results

The following are the results as well as their analysis, which will involve an overview of the general results with some of the generated confusion matrices and metrics reported directly.

In the first study, multiple variations of hyperparameters were tested. Batch size was limited by hardware restrictions, learning rate and learning decay were experimented upon in steps correlating to powers of 10. These values were found empirically via training neural networks from scratch and comparing the results. The hyperparameters that were finally settled upon were:

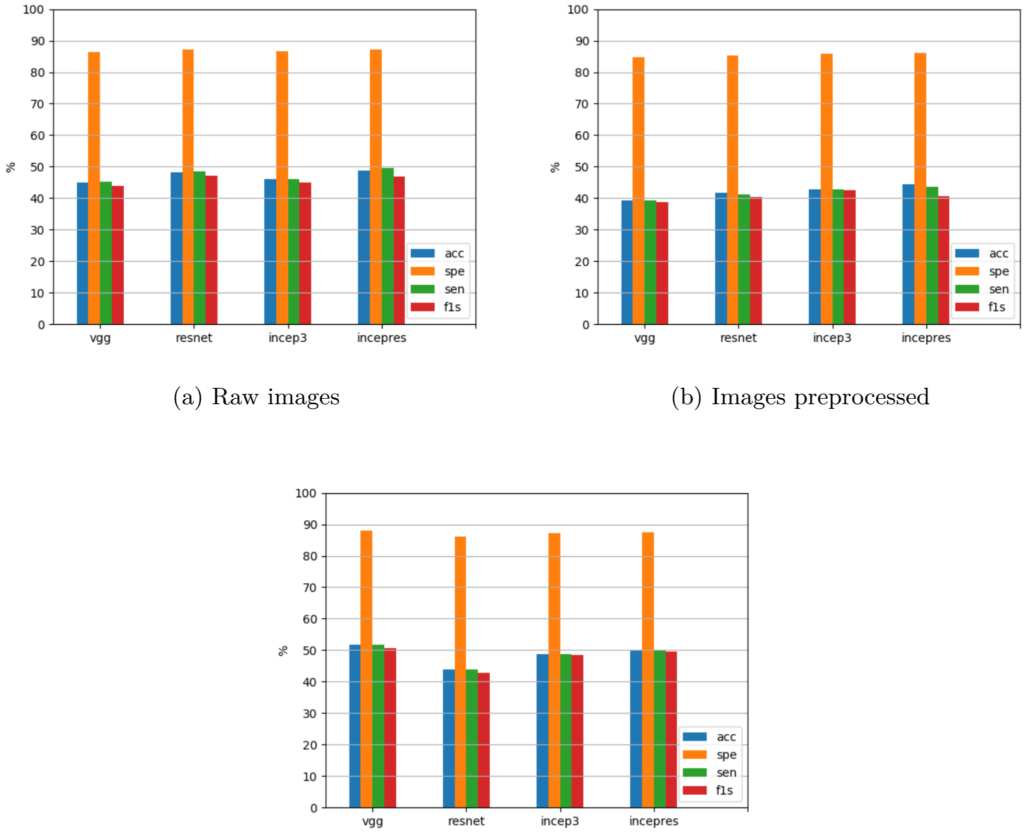
Table 5.3: Hyper parameters selected for use in neural network training

	Non-Augmented	Augmented
Momentum	0.9	0.9
Batch size	4	16
Learning rate	1×10^{-3}	1×10^{-5}
Learning decay	1×10^{-5}	1×10^{-7}

First study

This test was separated in two parts, one where the network was trained from scratch and another where the networks were initiated with weights obtained from the training of said network on the ImageNet dataset [67], thus implementing a transfer learning strategy..

On the networks trained from scratch these were the achieved results:



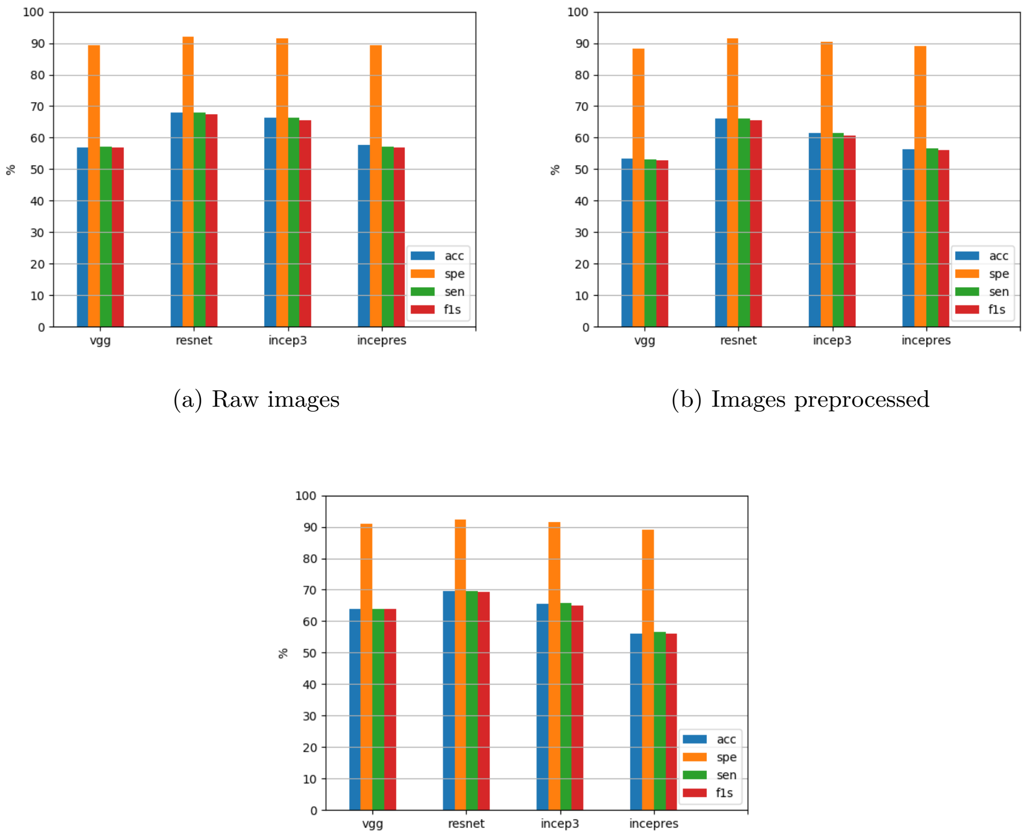
(a) Raw images

(b) Images preprocessed

(c) Images preprocessed and Augmentated

Figure 5.5: Average metric histogram across all folds trained from scratch per architecture.

As for the networks which were trained with transfer learning these were the achieved results:



(a) Raw images

(b) Images preprocessed

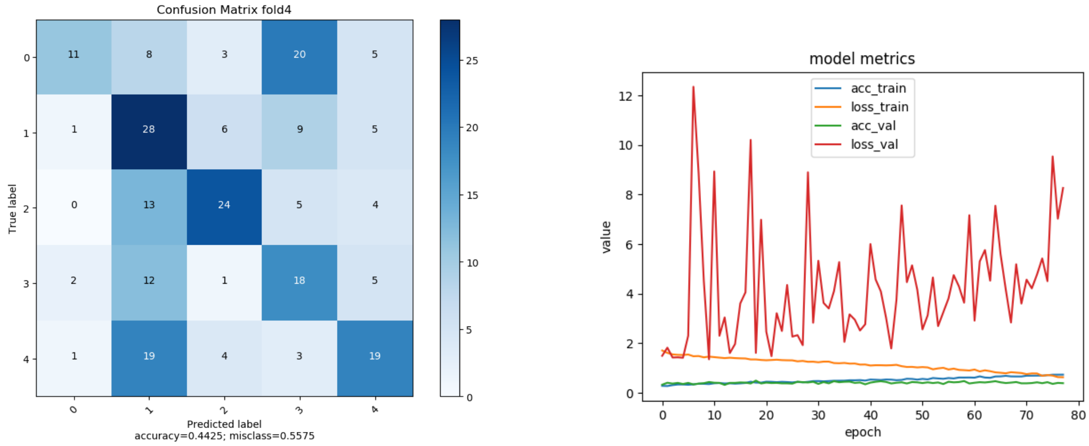
(c) Images preprocessed and Augmentated

Figure 5.6: Average metric histogram across all folds trained with transfer learning per architecture.

In Figures 5.5 it can be observed that the networks are learning to distinguish between the 5 classes of the first dataset are as their results are above the random chance value of 20% which could be expected in this scenario. Further, it is possible to note that, with data augmentation, all networks perform at a comparable rate to each other, whilst losing some performance on one of the networks, the inception-ResNet-v2 which seems to perform better with fewer samples. The similarity between the performances of the networks shows that these could perform well in an ensemble, as all the networks proved that they are capable of capturing good features for this classification task and therefore neither will overpower the others.

In Figures 5.6 transfer learning techniques were added to the neural networks, with no other alterations to the training so the improved results when compared directly with the reported metrics in Figures 5.5 can be attributed to this technique. These show an interesting quirk of the inception-ResNet-v2 network which appears to be worsening its results until the introduction of data augmentation techniques. However, the overall performance of the neural networks improves significantly when this technique is used. This would reflect that these weights, created for the general challenge ImageNet, can bring useful information to the specific task of skin lesion classification. And since these weights are obtained from a more general task it also appears to reduce the effects of overfitting.

In the following, the specific behavior of some of the networks will be analyzed more in details by reporting the confusion matrices and learning curves associated with some of the folds of the 5-fold cross-validation system used to compute the overall metrics. As they serve to illustrate some of the setbacks that can occur in this sort of model.



(a) Confusion matrix of fold 4 Resnet50 network with pre0 pre-processing

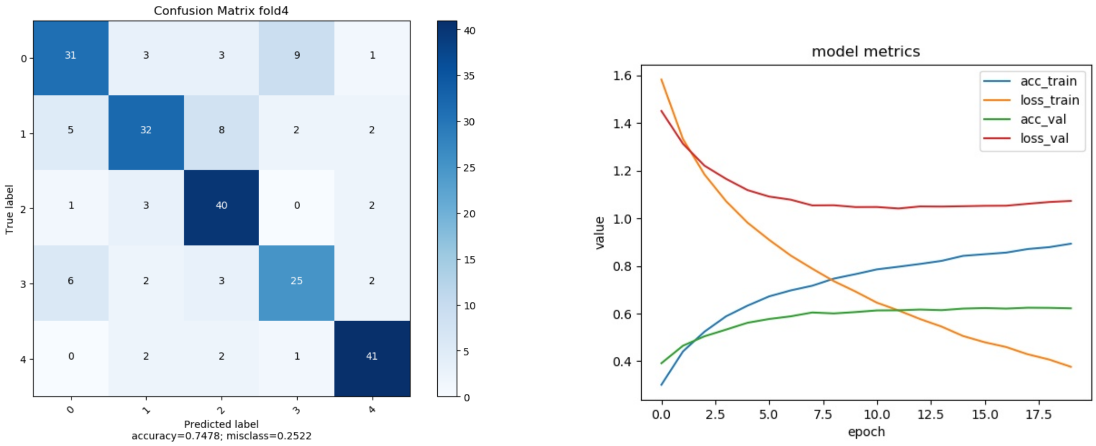
(b) History of fold 4 Resnet50 network with pre0 pre-processing

Figure 5.7: Examples of problems during the training of the neural networks. Class labels: 0 - BCC; 1 - MM; 2 - NV; 3 - SCC; 4 - SK

The confusion matrix in Figure 5.7a shows that a particular case of overfitting occurred, where the trained neural network fails to generalize well. This network was trained with no transfer learning Figure 5.7a, shows that the network is not learning well enough to be able to separate the classes meaningfully, therefore failing to classify the samples correctly. Whilst Figure 5.7b shows the problems this training had on the stability of the validation loss. This is not caused by insufficient information but by not learning information that is not relevant to the separation of the class and subsequently using this in the classification process.

This problem can be caused by either or the combination of the following deficiencies on the neural network training: 1. Poor initialization values for the neural network; 2. An insufficient amount of training samples;

These factors will be experimented upon with both the introduction of transfer learning and data augmentation. In spite of the fact that generating new samples via data augmentation is not comparable to the acquisition of new independent samples, this the best option for the data available.



(a) Confusion matrix of fold 4 Resnet50 network with pre0 pre-processing and data augmentation

(b) History of fold 4 Resnet50 network with pre0 pre-processing and data augmentation

Figure 5.8: Corrected training of the neural networks previously mentioned.
Class labels: 0 - BCC; 1 - MM; 2 - NV; 3 - SCC; 4 - SK

In Figure 5.8 is the homologous training to the reported in Figure 5.7, in which the aforementioned training problems have been corrected. The introduction of both data augmentation and transfer learning has improved the training of these neural networks. Not only it is possible to observe the improvements in the confusion matrix, but also the curves in the training history are smoother. Of particular note is the validation loss curve which shows that the network was able to progressively improve with no noticeable peaks as in Figure 5.7b. This leads towards the acceptance that both data augmentation and transfer learning are tools which should be used in the following studies.

Second Study

Besides the use of the same hyper parameters as in the first study, this study also uses data augmentation and preprocessing. In addition there was a change in the dataset structure. This change consists on the joining of two classes, BCC and SCC. These classes were joined due to the observation of high misclassification rate between these classes, as can be seen in Figure 5.9.

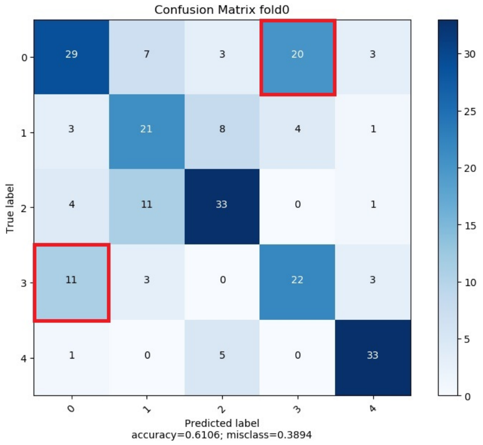
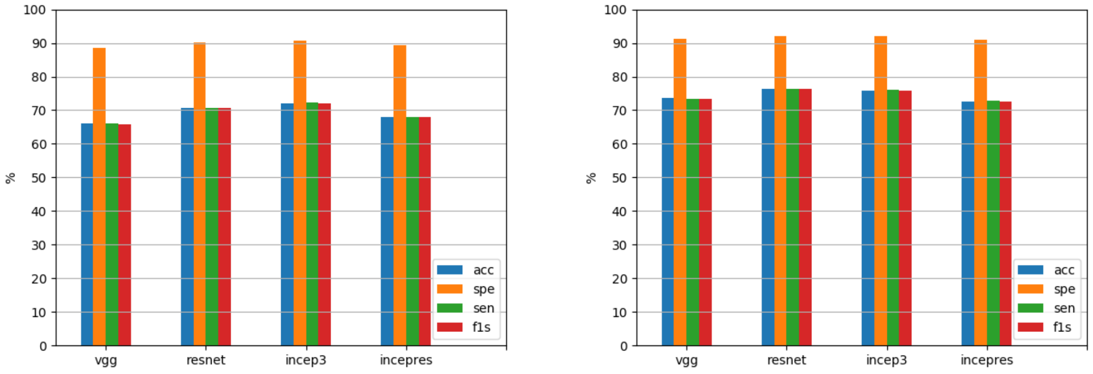


Figure 5.9: Confusion matrix illustrating the incorrect classification between classes BCC and SCC, confusion matrix from Resnet50 with data preprocessed and augmented.
Class labels: 0 - BCC; 1 - MM; 2 - NV; 3 - SCC; 4 - SK

The average results achieved in this study are as follows:

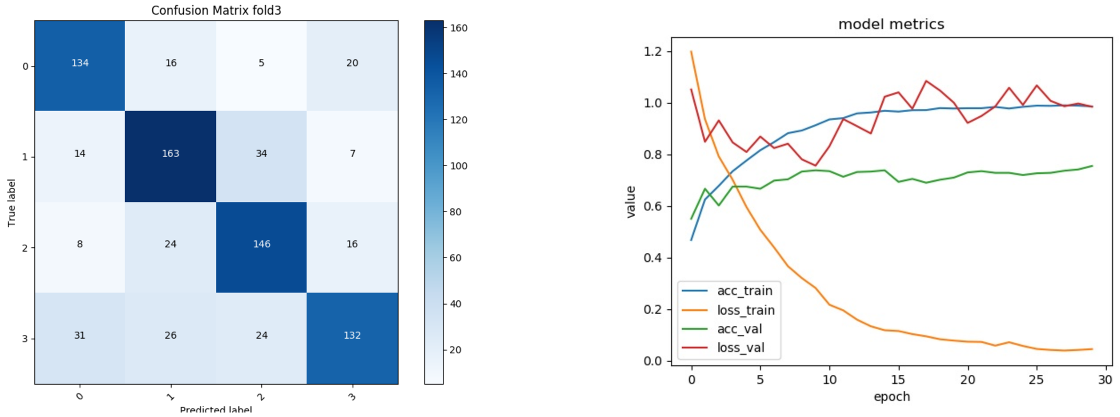


(a) Images preprocessed

(b) Images preprocessed and augmented

Figure 5.10: Average metric histogram across all folds trained from scratch per architecture.

The changes to the dataset generated better results than those reported in Figure 5.6c by a substantial margin, as well as reinforcing the notion that data augmentation is crucial for the good performance of such systems. Another interesting observation is that the neural networks are stabilizing achieving similar results relative to one another.



(a) Confusion matrix of fold 3 Resnet50 network with pre0 pre-processing and data augmentation

(b) History of fold 3 Resnet50 network with pre0 pre-processing and data augmentation

Figure 5.11: Average metric histogram across all folds trained from scratch per architecture. Class labels: 0 - NMC; 1 - MM; 2 - NV; 3 - SK

A typical training performed in this study behaved as reported in Figure 5.11, where it can be observed that there is a good separation of the classes being classified 5.11a, as well as, the convergence of the training and validation curves.

Ensemble Study

The ensemble study was performed using the same framework as the second study, but with a new architecture for the neural networks. This new architecture is the concatenation of the features extracted by each of the neural networks from the same input and then working this input in a single classification system, as described in Section 4.5.5.
This approach is aided by the stability of the networks, as seen in the second study, achieving the following results compared to those in the second study:

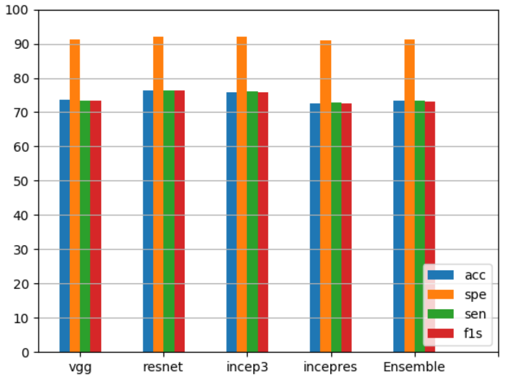
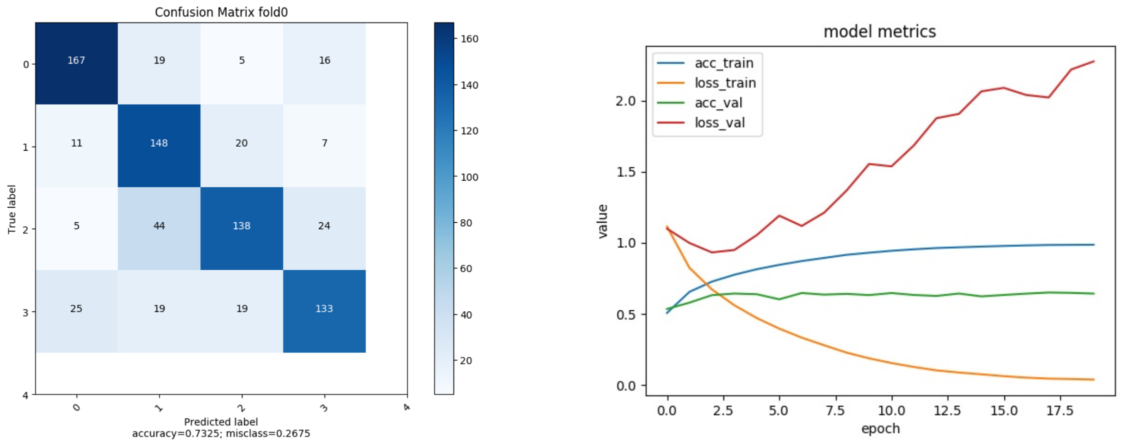


Figure 5.12: Average results for the trained ensemble, included with the equivalent dataset training from the second study.



(a) Confusion matrix of fold 0 trained on the ensemble with pre0 pre-processing and data augmentation

(b) History of fold 0 trained on the ensemble with pre0 pre-processing and data augmentation

Figure 5.13: Typical training on the ensemble neural network.
Class labels: 0 - NMC; 1 - MM; 2 - NV; 3 - SK

As it can be seen in Figure 5.12, the final results for the ensemble show that its performance is comparable to the previously tested neural network architectures.

Figure 5.13 shows a typical training for this network. The confusion matrix represented in 5.13a shows that the network is having some difficulties in the separation of some classes, particularly the MM and NV classes. As for the training curves, Figure 5.13b, these are smooth and appear to have reached convergence, and the behavior of the validation loss curve is consistent with the network overfitting, which is likely to be due to an excessive amount of epochs being run, as well as the scarcity of data.

There are some interesting observations taken from these studies. Firstly the increase of complexity in the neural network will not always correlate to better results, as can be attested by the ensemble results when compared to the other neural networks trained. Techniques like data augmentation and transfer learning appear to be extremely important to the performance of these algorithms as for every single instance when these techniques were applied the results improved.

The amount of data seems to be another very important aspect for the training of these networks, as can be seen from the first to second studies results, once more samples are available for the training the results tend to improve. Another is that these networks seem to be overfitting. This is most likely related to the small training sample size, as well as an excessive amount of epoch run. It is very likely that had an early stopping strategy been put into place the results of the trained networks would have improved. Future work could go towards the acquisition of more data that could potentially improve the performances of the neural networks further.

Chapter 6

Conclusion

Final remarks

From the results reported in this work, it can be concluded that deep neural networks represent a good approach for the classification of skin lesions. Although most works in the state-of-the-art deal with Melanoma classification exclusively, this need not be the case. This work focused on the classification of a wider number of possible diagnosis, which ranges among the most common of skin cancers to the less troublesome of skin lesions.

Good results were achieved as the networks proved they are not only capable of separating the classes of skin lesions but also appear to be capable of achieving better results than those here reported. This can be evidenced by the capability of similar systems that have been developed since this work started, such as [49] and this is, in fact, the direction that the scientific community seems to be trending towards as there are new challenges being put forth like the Isic-2019 challenge [73] which asks for the classification of 8 different diagnosis with a none of the above class.

The usage of data augmentation and transfer learning techniques proved to be an integral part for achieving the results reported in this work as they improved the results by a substantial margin in every test performed.

Another interesting factor is that whilst the ensemble has a very naive and simplistic architecture, it has competing capabilities with the other networks with good generalization abilities, in spite of overfitting with the available data.

Furthermore, a functional serviceable tool for the access of the neural networks by a user is discussed in Section 4.6, and it shows that such systems can be used for both an individual to keep track of their skin lesions, and a dermatologist to use such a system as a first indicator for problematic skin lesions. This system could also be integrated into a potential screening system for the general population, which could prove interesting from a preventive medicine perspective.

In this sense, this work represents an interesting effort in bridging the gap between theoretical performance studies and a practical application of deep learning for skin lesion classification.

Future Work

In the future, there should be an effort made towards experimenting with different neural networks in the ensemble, as well as additional work in the classification section of said network. Finding a good combination of neural networks with a better classification section could prove to be very fortuitous in terms of the performance for this kind of systems.

Additionally obtaining more samples with trustworthy labels and good quality of image, i.e., not watermarked, should improve upon the capability of this system to generalize. Another factor that could prove beneficial is the utilization of not only the image but also metadata (i.e., age of the subject, location of the lesion, gender of the subject).

Furthermore, the study of a different structure for the application could provide better results for a more general classifier for these diagnoses. A neural network would first classify the lesion based on their characteristics, then a second neural network trained for the diagnosis of the particular classification from the previous network would provide a more detailed diagnosis, i.e., a neural network would separate the lesion into a rash or a tumour, then another neural network would classify a potential tumour into the classes used in this work.

Bibliography

[1] Murad Alam and Désirée Ratner. Cutaneous squamous-cell carcinoma. *New England Journal of Medicine*, 344(13):975–983, 2001.

[2] Hellenic Derm Atlas. Hellenic derm atlas website, 2019-1-0325.

[3] Marina Bacac and Ivan Stamenkovic. Metastatic cancer cell. *Annu. Rev. pathmechdis. Mech. Dis.*, 3:221–247, 2008.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35 (8):1798–1828, 2013.

[5] J Gordon Betts. *Anatomy and physiology.* Houston, Texas: OpenStax College, Rice University, 2013.

[6] David E Bloom, Axel Boersch-Supan, Patrick McGee, Atsushi Seike, et al. Population aging: facts, challenges, and responses. *Benefits and compensation International*, 41(1):22, 2011.

[7] Almut Boer, KC Nischal, et al. www. derm101. com: A growing online resource for learning dermatology and dermatopathology. *Indian Journal of Dermatology, Venereology, and Leprology*, 73(2):138, 2007.

[8] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 2018.

[9] Tim Bray. The javascript object notation (json) data interchange format. 2014.

[10] Jay Burmeister and Janet Wiles. The challenge of go as a domain for ai research: a comparison between go and chess. In *Intelligent Information Systems, 1995. ANZIIS-95. Proceedings of the Third Australian and New Zealand Conference on*, pages 181–186. IEEE, 1995.

[11] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[12] Skin cancer uk. Skin-cancer-uk website, 2019-1-09.

[13] Tanguy Chouard. The go files: Ai computer wraps up 4-1 victory against human champion. *Nature News*, 2016.

[14] Geoffrey M Cooper. *The cancer book: A guide to understanding the causes, prevention, and treatment of cancer.* Jones & Bartlett Learning, 1993.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[16] Samuel Freire da Silva. Atlas dermatologico website, 2019-1-25.

[17] DermNetNZ. Dermnetnz website, 2019-1-25.

[18] Nelson Marcos Ferrari Júnior, Helena Muller, Manoel Ribeiro, Marcus Maia, and José Antonio Sanches Júnior. Cutaneous melanoma: descriptive epidemiological study. *Sao Paulo Medical Journal*, 126(1):41–47, 2008.

[19] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Hypertext transfer protocol–http/1.1, 1999.

[20] Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Doctoral dissertation, 2000.

[21] Michael C Fu. Alphago and monte carlo tree search: the simulation optimization perspective. In *Proceedings of the 2016 Winter Simulation Conference*, pages 659–670. IEEE Press, 2016.

[22] Adrian Galdran, Aitor Alvarez-Gila, Maria Ines Meyer, Cristina L Saratxaga, Teresa Araújo, Estibaliz Garrote, Guilherme Aresta, Pedro Costa, Ana Maria Mendonça, and Aurélio Campilho. Data-driven color augmentation techniques for deep skin image analysis. *arXiv preprint arXiv:1703.03702*, 2017.

[23] Ioannis Giotis, Nynke Molders, Sander Land, Michael Biehl, Marcel F Jonkman, and Nicolai Petkov. Med-node: a computer-assisted melanoma diagnosis system using non-dermoscopic images. *Expert systems with applications*, 42(19):6578–6585, 2015.

[24] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[26] Google. Machine learning glossary | google developers, 2018-12-07.

[27] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003.

[28] Gaorav P Gupta and Joan Massagué. Cancer metastasis: building a framework. *Cell*, 127 (4):679–695, 2006.

[29] HA Haenssle, C Fink, R Schneiderbauer, F Toberer, T Buhl, A Blum, A Kalloo, A Ben Hadj Hassen, L Thomas, A Enk, et al. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*, 29(8):1836–1842, 2018.

[30] Balazs Harangi. Skin lesion detection based on an ensemble of deep convolutional neural network. *arXiv preprint arXiv:1705.03360*, 2017.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[35] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[36] ISIC. Isic archive website, 2018-12-03.

[37] ISIC. Isic challenges website, 2018-12-03.

[38] ISIC. Simple color balance algorithm, 2019.

[39] Arthur Jacob. Observations respecting an ulcer of peculiar character, which attacks the eye-lids and other parts of the face. *Dublin Hosp Rep*, 4:232–239, 1827.

[40] Anthony F Jerant, Jennifer T Johnson, Catherine Demastes Sheridan, and Timothy J Caffrey. Early detection and treatment of skin cancer. *American family physician*, 62(2), 2000.

[41] Giuseppe Argenziano Jeremy Kawahara, Sara Daneshvar and Ghassan Hamarneh. 7-point criteria evaluation database, 2019-1-23.

[42] Simon Josefsson. The base16, base32, and base64 data encodings. 2006.

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

[44] Yuexiang Li and Linlin Shen. Skin lesion analysis towards melanoma detection using deep learning network. *Sensors*, 18(2):556, 2018.

[45] Yuexiang Li, Linlin Shen, and Shiqi Yu. Hep-2 specimen image segmentation and classification using very deep fully convolutional network. *IEEE transactions on medical imaging*, 36(7): 1561–1572, 2017.

[46] A Lomas, J Leonardi-Bee, and F Bath-Hextall. A systematic review of worldwide incidence of nonmelanoma skin cancer. *British Journal of Dermatology*, 166(5):1069–1080, 2012.

[47] RM MacKie, Axel Hauschild, and AMM Eggermont. Epidemiology of invasive cutaneous melanoma. *Annals of Oncology*, 20(suppl_6):vi1–vi7, 2009.

[48] Elaine Nicpon Marieb and Katja Hoehn. *Human anatomy & physiology*. Pearson Education, 2007.

[49] Roman C Maron, Michael Weichenthal, Jochen S Utikal, Achim Hekler, Carola Berking, Axel Hauschild, Alexander H Enk, Sebastian Haferkamp, Joachim Klode, Dirk Schadendorf, et al. Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks. *European Journal of Cancer*, 119:57–65, 2019.

[50] Frederich H Martini, Judi L Nath, and Edwin F Bartholomew. *Anatomy and physiology*. New York: Prentice Hall, 2005.

[51] Hitoshi Matsubara, Hiroyuki Iida, and Reijer Grimbergen. Chess, shogi, go, natural developments in game research. *ICCA journal*, 19(2):103–112, 1996.

[52] Jonathan E Mayer, Susan M Swetter, Teresa Fu, and Alan C Geller. Screening, early detection, education, and trends for melanoma: current status (2007-2013) and future directions: Part ii. screening, education, and future directions. *Journal of the American Academy of Dermatology*, 71(4):611.e1–611.e10, 2014.

[53] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[54] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira. Ph2-a dermoscopic image database for research and benchmarking. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5437–5440, July 2013. doi:10.1109/EMBC.2013.6610779.

[55] Scott W Menzies, Christian Ingvar, Kerry A Crotty, and William H McCarthy. Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features. *Archives of Dermatology*, 132(10):1178–1182, 1996.

[56] Bogdan Miclut. Committees of deep feedforward networks trained with few data. In *German Conference on Pattern Recognition*, pages 736–742. Springer, 2014.

[57] Mette Mogensen and Gregor BE Jemec. Diagnosis of nonmelanoma skin cancer/keratinocyte carcinoma: a review of diagnostic accuracy of nonmelanoma skin cancer diagnostic tests and technologies. *Dermatologic Surgery*, 33(10):1158–1174, 2007.

[58] Franz Nachbar, Wilhelm Stolz, Tanja Merkle, Armand B Cognetta, Thomas Vogt, Michael Landthaler, Peter Bilek, Otto Braun-Falco, and Gerd Plewig. The abcd rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4):551–559, 1994.

[59] Danish national service of Dermato-Venereology. Danderm website, 2019-1-25.

[60] David T Netscher and Melvin Spira. Basal cell carcinoma: an overview of tumor biology and treatment. *Plastic and Reconstructive Surgery*, 113(5):74e–94e, 2004.

[61] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press USA, 2015.

[62] Anibal Pedraza, Gloria Bueno, Oscar Deniz, Gabriel Cristobal, Saúl Blanco, and María Borrego-Ramos. Automated diatom classification (part b): A deep learning approach. *Applied Sciences*, 7:460, 05 2017. doi:10.3390/app7050460.

[63] Fábio Perez, Cristina Vasconcelos, Sandra Avila, and Eduardo Valle. Data augmentation for skin lesion analysis. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, pages 303–311. Springer, 2018.

[64] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[65] Jonathan L Rees. Skin cancer 909: a textbook of skin cancer for medical students, 2019-12-03.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi:10.1007/s11263-015-0816-y.

[68] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[70] American Cancer Society. Cancer facts and figures 2018. Technical report, ACS, 2018.

[71] H Peter Soyer, Rainer Hofmann-Wellenhof, Robert H Johr, et al. *Color atlas of melanocytic lesions of the skin*. Springer Science & Business Media, 2007.

[72] Patricia S Steeg. Tumor metastasis: mechanistic insights and clinical challenges. *Nature medicine*, 12(8):895, 2006.

[73] Jason Su. Isic 2019 challenge "skin lesion analysis towards melanoma detection" call, 2019.

[74] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[76] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.