UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---

Applicazione del framework *concept-bottleneck model* per la diagnosi di tumori alla pelle basato sulla regola *7-point checklist*

Application of the concept-bottleneck model framework for the diagnosis of skin cancer based on the 7-point checklist rule

Candidate:
Giulia Evangelisti

Advisor:
Prof. Simone Fiori

Coadvisor:
Dr. Hwee Kuan Lee
Dr. Davide Coppola

Academic Year 2020-2021

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---

Applicazione del framework *concept-bottleneck model* per la diagnosi di tumori alla pelle basato sulla regola *7-point checklist*

Application of the concept-bottleneck model framework for the diagnosis of skin cancer based on the 7-point checklist rule

Candidate:
Giulia Evangelisti

Advisor:
Prof. Simone Fiori

Coadvisor:
Dr. Hwee Kuan Lee
Dr. Davide Coppola

Academic Year 2020-2021

Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione
Via Brecce Bianche – 60131 Ancona (AN), Italy

Acknowledgments

I would like to dedicate this space to those who, with dedication and patience, have contributed to the realization of this paper.

A special thanks go to my supervisor Simone Fiori who has supported me, with his infinite availability.

Thanks also to my co-advisors Hwee Kuan Lee and Davide Coppola for their precious advice and for their guidance during the writing process of my thesis.

Thank you very much to my mother and my boyfriend Marco, because, without their support, this thesis would not even exist.

Thanks to all my colleagues, for always encouraging me since the beginning of my university career.

Ancona, Giugno 2021

Giulia Evangelisti

Sommario

Il melanoma è una delle forme più comuni e mortali di cancro, che può svilupparsi a partire da lesioni cutanee. La diagnosi tempestiva permette di salvare la vita delle persone, nella maggior parte dei casi. Per diagnosticare questo tumore, gli esperti oltre a raccogliere informazioni sulla storia dermatologica del paziente, effettuano un'ispezione visiva e dei test diagnostici. La dermatoscopia, dermoscopia o epiluminescenza, è una tecnica non invasiva rivolta alla diagnosi precoce del melanoma, in cui l'analisi della lesione cutanea viene condotta per mezzo di un dermatoscopio; uno strumento che favorisce l'ispezione della pelle in-vivo, rimuovendo i riflessi della superficie della pelle. Uno dei metodi più diffusi in dermatologia per la diagnosi di cancro alla pelle, è la regola 7-pt checklist che prevede l'identificazione di una serie di attributi sulla lesione, ad ognuno dei quali è assegnato un punteggio. La lesione viene diagnosticata come melanoma se supera una certa soglia. Ad oggi, molti medici utilizzano il *7-point checklist rule* sfruttando immagini prese tramite dermoscopio. Tuttavia, l'ispezione della cute tramite un dermoscopio è spesso influenzata dall'esperienza del medico, così come da altri fattori come l'attenzione, lo stress o la fatica. Per affrontare questo problema, sforzi nel settore della ricerca negli ultimi anni si sono rivolti allo sviluppo di sistemi Computer-Aided Diagnosis (CAD), ovvero sistemi di diagnosi assistita da computer, con l'obiettivo di supportare le decisioni degli specialisti. In letteratura è presente un'ampia gamma di applicazioni CAD che hanno raggiunto performance equivalenti o superiori rispetto a quelle di dermatologi esperti. Tuttavia, i modelli di Deep Learning (DL), sui quali si basano i sistemi CAD non sono ancora parte integrante delle attività di diagnosi, a causa della loro natura *black-box*: il processo di ragionamento del sistema di Intelligenza Artificiale risulta non visibile all'utente finale. Per questa ragione i medici hanno poca fiducia nel risultato restituito dal software: perché la macchina ha predetto la diagnosi di meloma? Quali sono i fattori identificati nell'immagine che hanno condotto il sistema di DL a prendere una determinata decisione? Da questo limite nasce l'obiettivo di questo lavoro, che è quello di sviluppare un modello di deep learning che espliciti l'evoluzione del suo ragionamento circa le sue predizioni. La vision è la creazione di un futuro ecosistema umano-artificiale nel quale entrambi gli attori (il dottore e il sistema di AI) collaborino per il raggiungimento del medesimo scopo. Se i modelli di AI saranno accettati, potranno essere di supporto alle decisioni diagnostiche. Questi modelli posso ingerire molti più dati e più rapidamente rispetto alle capacità di un

essere umano, e questo potrebbe arricchire la conoscenza di un medico, che avrebbe a disposizione una quantità di informazioni nettamente superiore.
Il framework *Concept-Bottleneck Model* (CBM) è un design pattern proposto in letteratura, in cui la struttura del modello viene riorganizzata in due componenti: un modello addestrato sulle immagini, che fa predizioni su concetti espressi in linguaggio naturale, e quindi comprensibili da un essere umano; e un modello addestrato su questi concetti che fa predizioni di uno specifico output target. In questo lavoro i concetti sono i task della *7-point checklist* e il target di output è la diagnosi binaria di Melanoma o Naevi. In questo modo, il modello AI mostra una spiegazione spiegazione concettuale della diagnosi finale. Questo studio mira a sviluppare *explainable models* per la predizione di cancro alla pelle utilizzando il framework CBM, e li confronta con modelli più tradizionali come il *Single-Task Learning* (STL) e il *Multi-Task Learning*, anch'essi implementati e utilizzati negli esperimenti. Per tutte le configurazioni proposte, sono state utilizzate architetture popolari di reti neurali per l'impelentazione dell'estrazione delle features nelle immagini. Le architetture utilizzate come *base*sono: ResNet50, InceptionV3, e DenseNet121. La parte *head* delle reti neurali, volta alla classificazione delle label finali, è stata invece progettata ed implementata per ogni modello proposto. Il dataset utilizzato è il *derm7pt*, pubblicamente rilasciato. Si tratta di un database di oltre 2000 immagini e annotazioni sugli attributi della 7-pt checklist e sulla diagnosi. I risultati ottenuti sono promettenti e dimostrano come i CBM ha delle performance equiparabili rispetto ai modelli standard. Questo suggerisce che ulteriori studi possono essere condotti per testare architetture più sofisticate basate su questo framework.

Abstract

Melanoma is one of the most common and deadly forms of cancer arising from skin lesions. An early dectection has been shown to aid in reversing the odds of survival in the majority of cases. Physicians use evaluations of the patient that include gathering information, visual inspection and diagnostic tests for skin disorders. In dermatoscopy the inspection of skin lesion is conducted by means a dermatoscope, that allows a visual in-vivo analysis unobstructed by skin surface reflection. The *7-point checklist* rule is one of the most used rule-based method in dermatology for the diagnosis of skin cancer by dermatologists, which consists in identifying a set of attributes on the lesion and assigning a score to them. The lesion is diagnosed as melanoma if it exceeds a certain threshold. However, the visual inspection with dermatoscope is often influenced by the experience of doctor, as well as other factors such as attention, stress, and fatigue. To deal with this challange, over the last years, research efforts have been directed towards the development of Computer-Aided Diagnosis (CAD) systems to support the physicians' decisions. Many applications of CAD were presented in the literature, and they have shown to achieve performance comparable or better to experienced dermatologists. Despite the good performance, Deep Learning (DL) models have not been accepted by the medical community yet, due to their *black-box* nature in which the reasoning process of the Artificial Intelligence system remains "opaque" to the final user. For this reason, physicians do not trust in the output of the model: why does the machine say that the diagnosis is cancer? What are the factors found in the image that have guided a particular output? From this gap arises the objective of this work: to develop a trustworthy deep learning model, which exposes its reasoning process to the final user. The vision is to have, in the future, a hybrid human-artificial ecosystem where both actors (the doctor and the AI system) work together in a collaborative way. The *Concept-Bottleneck Model* (CBM) framework is a design pattern proposed in the literature, in which the architecture of the model is re-organized in two components: one model trained on raw images, which predicts human-understandable concepts, and a second model trained on these concepts which predicts a specific target.

In this work the concepts are the *7-point checklist* patterns and the target is the binary diagnosis of Melanoma. In this way, the AI model shows a conceptual explanation of the final diagnosis. This study aims to develop explainable models for the prediction of melanoma using the CBM framework, and to compare them to

more traditional models such as *Single-Task Learning* (STL) and simple *Multi-Task Learning*. For all the proposed configurations, popular neural network architectures are used for the features extraction: ResNet50, InceptionV3, and DenseNet121 and customized heads have been built in order to classify the final labels. The dataset used in this work is the publicly released *derm7pt*: a database of over 2000 dermoscopy images and annotations for the attributes of the checklist. In the experiments, the models achieve promising performance and show that CBM yields equiparable results in comparison to standard models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most common and deadly forms of cancer is Melanoma, which arising from skin lesions. However, an early diagnosis has been shown to aid in reversing the odds in the majority of cases [9]. Physicians use evaluations of the patient that include gathering information, visual inspection and diagnostic tests for skin disorders, such as biopsy, scraping, diascopy, and so on. The focus of this work is on visual inspection through dermoscopy images. Different strategies have been developed by the medical community to recognize a skin lesion. The 7-point checklist method accounted for in this project is a rule-based algorithm that requires identifying seven dermoscopic patterns correlated with melanoma. Each criterion is assigned a score and the skin lesion is diagnosed as melanoma when the sum of the score passes the given threshold. The last decade has seen an increasing interest in the development of Computer-Aided Diagnosis (CAD) systems thanks to the improvement of machine learning algorithms and the spreading accessibility of computational power. CADs help physicians in their analysis and are able to work on a higher number of images in less time compared to human efficiency as they are not subject to stress or fatigue. Nowadays, deep learning methods for skin lesion diagnosis have reached dermatologist-level performance [10]. However, machine learning models are not able to explicit the equivalent abilities of a reasoning human. As a matter of fact models can inference relationships as it only links statistical associations in the data. Critical applications (e.g. justice or healthcare) are limited due to the lack of understanding of the inference process of ML models. Complex reasoning patterns need the ability to understand the context and also to master trivial abilities simultaneously. For instance, in the field of computer vision: spatial awareness, recognizing of colors and shapes, and so on. In pshychology this is called *transfer of learning* that occurs when people apply strategies, skills, and information they have learned to a new problem or context [11]. For humans, the information re-utilization from one task to another is common: a solution or conclusion rarely ever involves a single knowledge or skill. In Multi-Task Learning (MTL) [12] this scheme is an inspiration and models learn multiple tasks simultaneously, leveraging on the shared features that they might have. The drawback of this approach is the increase of model complexity and consequentially

the interpretability and explanation of its decision process. This is a crucial point for the usability of models, in fact, regardless of tested performances on new data, the human operators are less likely to trust ML without a clear understanding of the reason behind a prediction or choice.

Thus, it is necessary to understand and retrace the decision process of the model not only at a low level (i.e. mathematical/algorithmic) but also at high-level explanations (i.e. natural language representations/concepts) [13]. The idea proposed by the *Concept Bottleneck Model* framework [14] is to develop models that first predict a set of interpretable concepts, and then use these concepts to predict the main target label. Hence the model conveys and shows additional information more understandable by users and this should encourage more trust towards the system.

This work proposes architectures for single-task learning and multi-task learning for detecting melanoma in dermoscopy skin images. In fact, the application of such methods to the analysis of bio-medical images will have a critical impact once it becomes part of the daily workflows [13]. Furthermore different concept bottleneck models have been proposed, inspired by [14]. This architecture aims to enable better interpretability by humans (i.e. dermatologists) in real applications. The manuscript will be divided in the following chapters:

- Chapter 2 — **State of the Art**: this chapter begins with a description of the medical background 2.1 which introduces the essential backdrop to understand the diagnosis problem and which dermatologist algorithms and tools have used to design the Computer-Aided Diagnosis (CAD) system. Paragraphs about technical backgrounds 2.2 follow, which explain Deep Learning (DL) methods for image classification, Convolutional Neural Network (CNN) and the use of CNN in Multi-task learning. Lastly, this chapter shows machine learning applications to skin lesion diagnosis 2.2.3 and methods proposed on the same dataset used in this work (Derm7pt dataset).

- Chapter 3 — **Methods**: firstly the dataset used in the experiments is presented 3.1, as well as the various adjustements that have been applied to it. Then Single-Task Learning (STL) deep learning architecture is studied 3.3.1 with a variety of parameters and configurations. This is used only to predict diagnosis of melanoma or not melanoma (i.e. naevi). The analysis is expanded to the case of MTL 3.3.2, which aims to learn 8 different tasks related to 7-point checklist criteria: *diagnosis, pigment network, blue whitish veil, vascular structure, pigmentation, dots and globules, regression structures, and streaks.* Different variations of architectures have been proposed. Lastly, three different concept bottleneck model 3.3.3 setups have been implemented: *independent* model, *sequential* model, and *joint* model.

- Chapter 4 — **Results**: explores the setup used for experiments 4.1, and the definition of the metrics used to evaluate the performance of the models. The chapter containts the results obtained through the different methods proposed in Sec. 4.2.
- Chapter 5 — **Conclusion**: an overview of the entire work with some final considerations.

Chapter 2

State of the Art

2.1 Medical background

2.1.1 Skin Lesion and use of Dermatoscope

Definition of Skin Lesion

Skin is a complex tissue vital for the functions of mechanical protection, thermoregulation, immunosurveillance, and maintenance of a barrier that prevents insensible loss of body fluids.
The skin lesion is a superficial growth or patch of the skin that does not resemble the area surrounding it. Skin lesions can be grouped into two categories: primary and secondary.[15]

Primary Skin Lesion

This category groups all the variations of the skin in color or texture that may be present at birth (for example nevus or birthmarks) or that may be acquired during a person's lifetime. The latter represents all the skin lesions associated with infectious disease such as warts, acne, or psoriasis; allergic reactions, such as hives or contact dermatitis; environmental agents such as sunburn, pressure, or temperature extreme.

Secondary Skin Lesion

This category groups all the lesions that involve changes in the skin that result from primary skin lesions, either as a natural progression or as a result of a person manipulating a primary lesion.

Dermatoscope

A dermatoscope is a device used by dermatologists to examine the skin at low magnification (e.g. 10X). Older instruments consist of a low-power (10x) magnifier, a nonpolarized light source, a transparent plate, and a light layer of mineral oil between the instrument and the skin. The role of mineral oil is to facilitate inspection of skin lesions without reflection from the skin surface. Recent dermatoscope use polarized light to eliminate skin surface reflections [16], and allow digital image capture.

2.1.2 Dermatologist Diagnosis Methods

Skin Cancer is by far the most common type of cancer [17], and early detection is a major factor in reducing mortality rates associated with this type of skin cancer. There are two major types of Skin Lesion diagnosis:

- Evaluation of the Dermatologic Patient: which include the gathering of dermatologic history and dermatologic examination (i.e. visual inspection)
- Diagnostic Tests for Skin Disorders: that are indicated when the previous approach is not effective. This method includes patch testing, biopsy, scrapings,examination by wood light, tzanck testing, and diascopy. [18]

This work focuses on the first one and in particular on the dermoscopy examination method. The aid of dermoscopic images allows the application of different algorithms that are commonly accepted and used for skin lesion diagnosis by science. These diagnostic methods are the following: pattern analysis, ABCD rule; 7-point checklist criteria; Menzies methods, and the revised pattern analysis.[19] The *Board of the Consensus Netmeeting* agreed on a two-step procedure for the classification of pigmented lesions of the skin.[19]

1. The first step is the differentiation between nonmelanocytic and melanocytic lesions. For this decision phase, the algorithm in Figure 2.1 is used. The classification of the melanocytic lesion is based on the presence of *aggregated globules, pigment network, branched streaks, homogeneous blue pigmentation, or a parallel pattern (palms, soles, mucosa).* If any of the previous characteristics are present then the lesion should be evaluated for the presence of textitcomedo-like plugs, multiple milia-like cysts, and comedo-like openings, irregular crypts, light brown fingerprint-like structures, or "fissures and ridges" (brain-like appearance) pattern. If so, the lesion is suggestive of a seborrheic keratosis. If not, the lesion has to be evaluated for the presence of textitarborizing blood vessels (telangiectasia), leaf-like areas, large blue- gray ovoid nests, multiple blue-gray globules, spoke wheel areas, or ulceration. If present, the lesion is suggestive of basal cell carcinoma. If not, one has to look for textitred or red-blue (to black) lagoons. If these structures are present, the lesion should be considered a hemangioma or an angiokeratoma. If all the preceding questions were answered with "no," the lesion should still be considered to be a melanocytic lesion.
2. The step after the lesion identification in melanocytic origin, is the decision whether the lesion is benign, suspect, or malignant. To perform this, four different methods are the most commonly used and today considered the basic of dermoscopy knowledge[20]:

- Modified pattern analysis
- ABCD rule of dermatoscopy
- 7-Point checklist criteria
- Menzies method

In this dissertation, the focus will be on the Seven-Point Checklist method (see chapter 3). The most relevant comparison was made by dermatologists [21] which encompass the distinctions between ABCD rule and Seven-Point criteria: the 7-point checklist gave a sensitivity of 95% and specificity of 75% compared with 85% sensitivity and 66% specificity using the ABCD rule and 91% sensitivity and 90% specificity using standard pattern analysis (overall ELM diagnosis). Compared with the ABCD rule, the 7-point method allowed less experienced observers to obtain higher diagnostic accuracy values These methods are confirmed as valid by Argenziano, et al. in their 2021 publication. [20]
In the next paragraph, there is an explanation of Seven-Point checklist criteria.

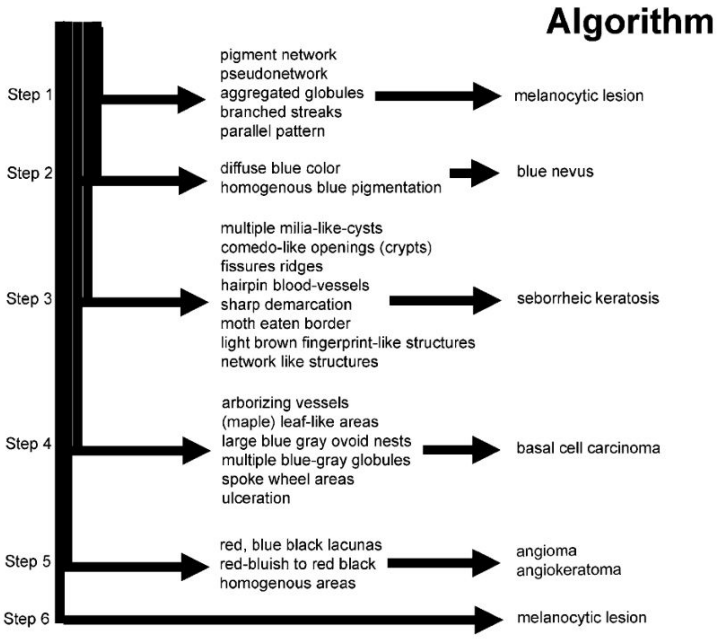


Figure 2.1: Algorithm for the determination of melanocytic versus nonmelanocytic lesions according to the proposition of the Board of the Consensus Netmeeting.[1]

2.1.3 7- Point Checklist Criteria

The 7-Point Checklist diagnostic method is a rule-based algorithm that requires identifying seven dermoscopic criteria associated with melanoma. For each criterion

is assigned a score and the skin lesion is diagnosed as melanoma when the sum of the score passes a given threshold [21, 22]. The two thresholds are:

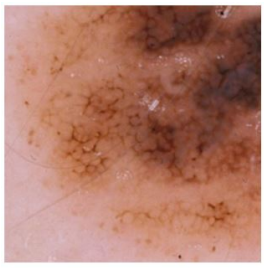
- Classic — excision is recommended if the total score is ≥ 3
- Revised — excision is recommended if the total score is ≥ 1

Table 2.1: Seven-point checklist: dermoscopic criteria and scores for the classic and the revised version of the algorithm

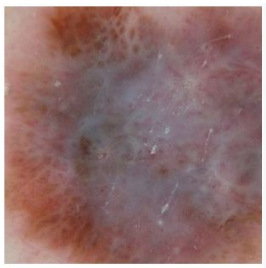
Dermoscopic pattern	Classic algorighm score	Revised algorithm score
Atypical network	+2	+1
Blue-white veil	+2	+1
Atypical vascular pattern	+2	+1
Irregular dots/globules	+1	+1
Irregular streaks	+1	+1
Irregular blotches	+1	+1
Regression structures	+1	+1

The following is a dermatologic description [22] of each pattern shown in 2.1. The images of each criterion are shown in Fig. 2.2 and 2.3

- Atypical network — Combination of at least two types of pigment network (in terms of color and thickness of the lines) asymmetrically distributed within the lesion
- Blue-white veil — Irregular, structureless area of confluent blue pigmentation with an overlying white 'ground-glass' film. The pigmentation cannot occupy the entire lesion and usually corresponds to a clinically elevated part of the lesion
- Atypical vascular pattern — Linear-irregular vessels, dotted vessels, and /or milky-red areas not clearly seen within regression structures
- Irregular dots /globules — More than three round to oval structures, brown or black in color, asymmetrically distributed within the lesion
- Irregular streaks — More than three brown to black, bulbous or finger-like projections asymmetrically distributed at the edge of the lesion and not clearly arising from network structures
- Irregular blotches — Black, brown, and/or grey structureless areas asymmetrically distributed within the lesion (Fig.
- Regression structures — White scar-like depigmentation and/or blue pepper-like granules usually corresponding to a clinically flat part of the lesion

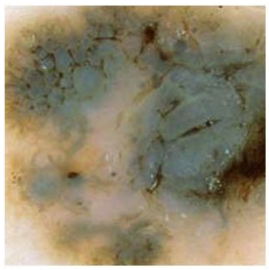


(a) *Atypical pigment network*



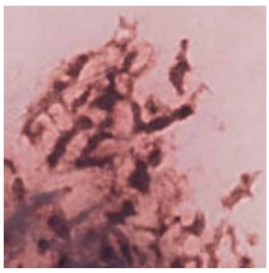
(b) *Presence of blue whithis veil*

(c) *Irregular vascular structure*



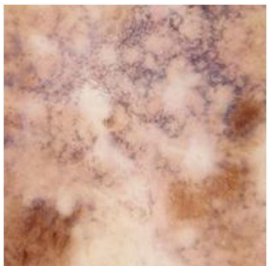
(d) *Irregular pigmentation*

Figure 2.2: Dermoscopy images of seven point checklist patterns: *atypical pigment network, presence of blue whithis veil, irregular vascular structure, irregular pigmentation*



(a) *Irregular streaks*

(b) *Irregular Dots and globules*



(c) *Presence of regression structures*

Figure 2.3: Dermoscopy images of seven point checklist patterns: *dots and globules, presence of regression structures, irregular streaks*

2.2 Technical background

2.2.1 Deep Learning for Image Classification

Pattern Recognition and Image Classification

Pattern Recognition is a field of automated discovery of regularities and patterns in data. These regularities are recognized through the use of computer algorithms and allow them to perform actions on data such as classification in various categories. Pattern recognition has several applications: image analysis, machine learning, signal processing, statistical data analysis, computer graphics, bioinformatics, information retrieval, and data compression.
Nowadays, thanks to the increased availability of processing power and the plentitude of Big Data, some modern pushes to pattern recognition involve the use of Machine Learning (ML). In ML, pattern recognition regards the assignment of a label to a given input value [23]. In this thesis, the methods used focus on a specific type of pattern recognition task: *Image Classification*, which tries to predict for each sample in the input one of a given set of classes (for instance, decide whether a given image represents "melanoma" or "not melanoma").
However, pattern recognition is a more general problem that deals with other types of output as well, for example:

- Regression, which assigns a real-valued output to each input;[24]
- Sequence labeling, which assigns a class to each member of a sequence of values[25] (e.g. part of speech tagging, which assigns a part of speech to each word in an input sentence);
- Parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.[26]

Pattern Recognition Algorithms mostly make effort to provide a reasonable answer for every possible input and to return the "most likely" matching of the inputs, taking into account their statistical variation. It is in contrast with *Pattern Matching Algorithms* which look for exact matches in the input using pre-defined patterns [1]

[1] An example is a regular expression matching that searches for patterns in textual data sorted in a determined way

Pattern recognition — supervised, unsupervised and semi-supervised

Pattern recognition is by and large categorized in line with the learning procedure used to grasp the regularities in data in order to predict an output value. There are three types of learning methods

- Supervised learning — expects a set of data (called training data) that contains observations that have been properly labeled with the correct output, by the hand of one or more domain experts. The goal of this learning procedure is to generate a model that endeavors to meet to objectives: perform as well as possible on the training data and generalize as well as possible on new data (i.e. can correctly label data never seen)

- Unsupervised learning — accepts training data that has not been hand-labeled, and tries to discover patterns that it can successively use to decide the correct output label for new data instances.

- Semi-supervised learning — is a recent combination of the two previous, which use both unlabeled and labeled.

One observation in the dataset (i.e. an input sample) for which an output value is generated is formally termed as an instance. Each instance is formally defined by a features vector, which together represents a characterization of all known about the instance. Features vectors can be seen as defining points in a multidimensional space, which means that it is possible to apply methods for manipulating vectors in vector spaces to the instances of a dataset. The feature can be:

- Categorical (or nominal) — set of unsorted items (e.g. gender of "male" or "female", profession "software engineer", "chef", and so on)

- Ordinal (i.e. set of ordered items such as "small", "medium" or "large")

- integer-valued (e.g. a count of number of occurrences of a particular word in an email)

- real-valued (e.g. measurement of body temperature)

Deep Learning

Deep Learning is a subfield of Machine Learning, in which the model is inspired by the biological brain. It is based on Artificial Neural Networks that work with representation learning. These two topics will be briefly introduces in the next paragraph:

- **Artificial Neural Networks** (ANNs), often called only Neural Networks (NNs) are inspired by distributed communication nodes and information processing in biological systems, but have some distinctions. Specifically, neural networks are more symbolic and static, while the brain of most living organisms is dynamic (i.e. plastic) and analogue. [27, 28, 29]. ANNs are computing systems with a structure formed by a collection of nodes called artificial neurons. These units are connected in a way that resembles the connections of neurons in the biological brain. Each connection is an edge between two nodes, and can send a signal. The behavior of each neuron is summarizable in three steps:

  1. receive the signal from another unit;
  2. process it;
  3. transmit the processed output to other units.

  The signal at a connection is a real number and represents the input message received by the neuron from another, and the output of each neuron is computed by function (linear or non-linear) of the sum of its inputs, in Fig. 2.4 is shown an example of linear unit. Edges typically have a weight that adjusts as learning

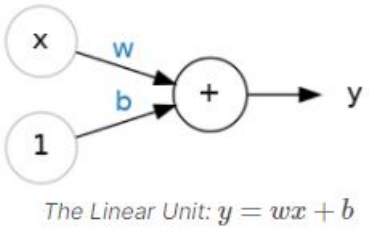


Figure 2.4: Example of linear unit with output that is a linear combination of one input x, weighted by w, added with a *bias* b[2]

  proceeds, multiplies the signal, and decreases or increases its strength.

  An ANN has a structure in which all neurons are organized typically in a layer. Different layers may carry out different transformations in their inputs. Signals go forward from the first layer, called the input layer, to the last layer, called the output layer. In Deep Learning there are hidden layers between the input and the output, and the overall lenght of the chain gives the depth of the model, It is from this terminology that the name "deep learning" arises. [30]

- **Feature learning** (also known as representation learning) is a set of techniques that allow a computer system to find out a representation for feature detection or classification from raw data. In contrast to feature engineering in which domain knowledge expertise is needed, in feature learning the machine learns both the features and the specific task to perform. [31]

Deep Learning belongs to supervised methods.

Nowadays there exist very powerful frameworks in Deep Learning for supervised learning. By creating a model architecture with more layers and more units within a layer, a Deep Neural Network (DNN) can map functions of increasing complexity. DNNs can accomplish most of the tasks that consist of mapping an input vector to an output vector, given relatively large models and datasets of labeled training examples.

The organization of neurons, connections, and layers can be designed in different ways. In this work, there was be used feedforward neural networks. Feedforward neural networks (also known as Deep feedforwards networks or multilayer perceptrons (MLPs) are Artificial Neural Networks wherein edges between neurons do not form a cycle. The goal of a feedforward network is to approximate some function f^*. For instance: a classifier, $y = f^*(x)$ maps an input x to a category y. A feedforward network defines a mapping $y = f(x; \theta)$ and learns the value of the parameters θ that result in the best function approximation. [30]

2.2.2 Convolutional Neural Networks Principal Methods for Image Classification

Convolutional Neural Networks (CNNs) are a class of deep feedforward networks most commonly used to analyze visual imagery and to categorize images into one of several predefined classes. A convolutional neural network consists of three layers – an input layer, hidden layer, and output layer. The specificity of a conv net is that the hidden layers follow this strucutre:

1. Convolutional Layer followed by detection stage

2. Sub-sumpling or pooling Layer

3. Fully-Connected Layer

These are the fundamental types of layers used in CNN, and an architecture example is shown in Fig. 2.5

The convolutional layer applies a convolutional operation through multiple filters, on every image. This operation generates one feature for each filter, extracting the high-level features such as vertical and horizontal edges. This mechanism is adjusted by the network during the training, learning the values of each filter (also known as weights). More precisely the inputs and the filters are multidimensional arrays, respectively of pixels and parameters (also referred as tensors). Formally the convolutional operation is, as explained in [30] defined by:

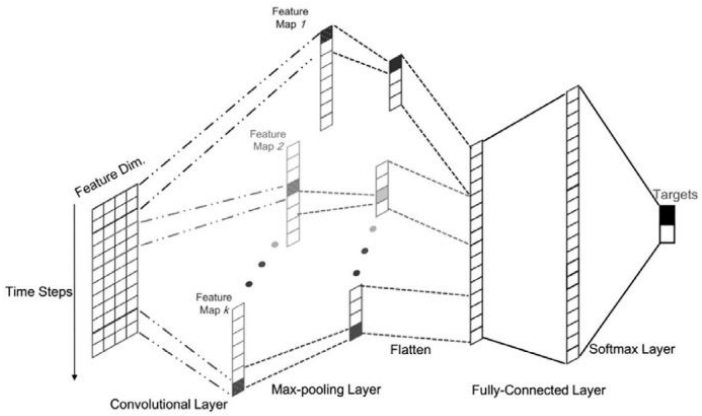


Figure 2.5: Simplified Convolutional Neural Network for classification task [3]

$$S(i,j) = (I * K)(i,k) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.1}$$

This operation is schematized, according to [30] in Fig. 2.6.

The convolutional operation produces a linear combination that is given in input to a nonlinear activation function (commonly rectified linear activation function shown in Fig. 2.7) This stage is often called as detection stage and for ReLU istance rectifies to zero all the negative values.

The last phase is handled by the pooling function, the most used is the Max Pooling function. This technique subsamples the input tensor, dividing the area of an input slice into chunks of non-overlapping squares and giving the maximum values within that area. Thus, a max-pooling operation can reduce the dimension of the tensor extracted by a feature map and helps pull smooth features.
The output of the pooling operation is flattened to a one-dimensional vector and used as the input of a fully connected feed-forward network. The last layer needs to compute the loss function in order to train the CNN. In this work, the last layer uses a softmax activation function to simulate pseudo-probability. During the training the network adjusts weights in order to minimize the loss function (i.e. the difference between ground truth and predicted label).

Famous architectures

In this section there is a brief introduction some popular CNN architectures for image classification *AlexNet, ResNet, InceptionNet, and DenseNet121*

AlexNet [4] — The AlexNet architecture is displayed in 2.8. It was designed by

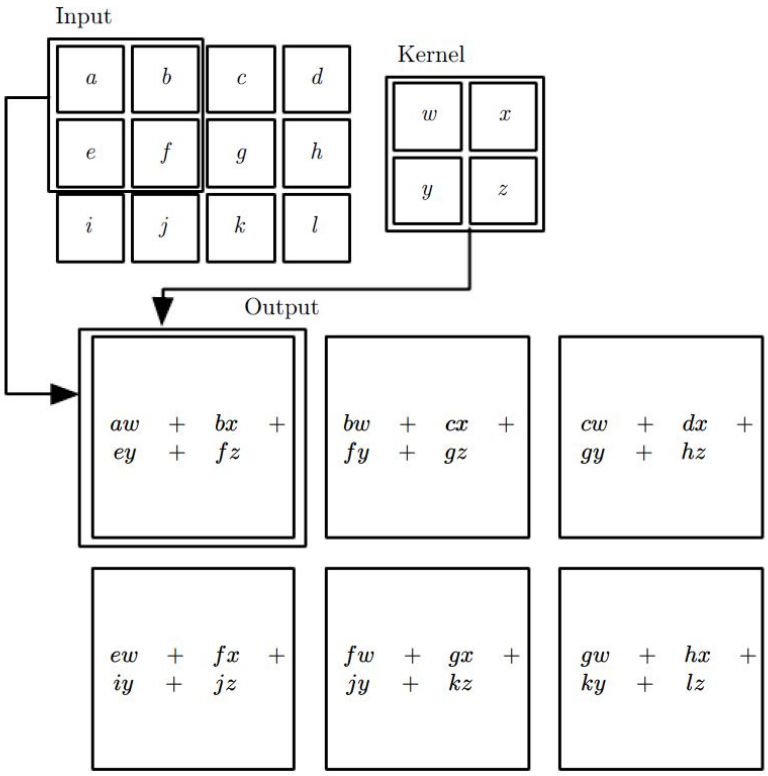


Figure 2.6: An example of 2-D convolution

Ilya Sutskever, Geoffrey Hinton, and Alex Krizhevsky in 2012 [4].
This network is a large convolutional neural network trained on ImageNet during the LSVRC-2010 contest, to classify 1.2 million of high-resolution images in 1000 different classes.
It is composed of five Conv layers, some of which are followed by max-pooling layers, and three fully-connected layers with an output layer that has 1000-way softmax neurons, to classify the 1000 different labels. During the test phase, they achieved top-1 and top-5 error rates of 37.5% and 17.0% respectively.

Deep Residual Network - ResNet [5] — In Deep Neural Network with high depth, creating a stack of layers does not work properly, the repeated multiplication may present a *vanishing gradient*. This issue occurs when the gradient is extremely small, and it prevents the weights from significantly changing their values.
The core objective of ResNet is to construct a shortcut connection that skips one or more layers to transport the input value in the output of another layer, through a direct edge (as shown in Fig. 2.9). The authors of this work want to address a degradation problem in which accuracy gets saturated and degrades rapidly. They show that it is not only caused by overfitting. In fact adding more layers leads to higher training error [5].

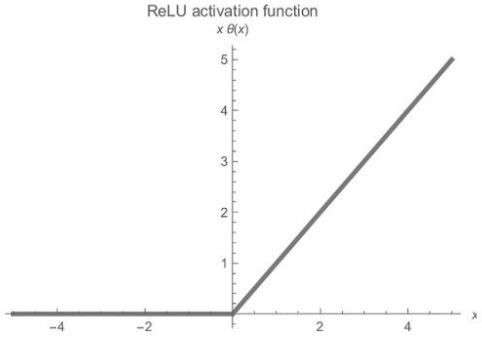


Figure 2.7: ReLu function

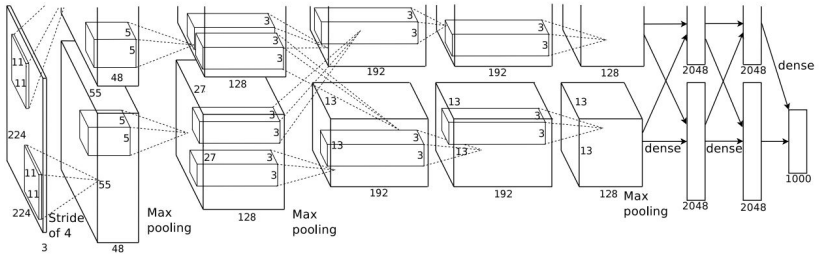


Figure 2.8: Structure of AlexNet[4]

With ResNet50 architecture a deeper model would not produce a greater training error compared to its shallower counterparts. This is achieved by explicitly letting each few stacked layers fit a residual mapping instead of hoping they would directly fit a desidered underlying mapping.
This formulation can be realized by feedforward neural networks with "shortcut connections" (Fig. 2.9), that are those skipping one or more layers and in this case perform *identity* mapping, and their ouputs are added to the output of the stacked layers. In literature there exists different ResNet architectures. The most commonly used are 50-layer, 101-layer, and 152-layer. Recently variants were proposed, such as denseley connected convolutional networks (DenseNet) and RexNeXT. These networks aim to increase the feature reuse and width of ResNet to increase the

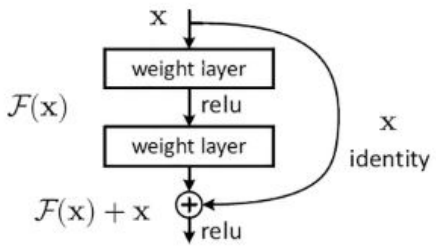


Figure 2.9: Residual learning: a building block [5]

accuracy and obtain a network highly parameter-efficient. [32, 33, 34]

InceptionNet [6] — The GoogLeNet architecture shown in 2.10, also named Inception v1 has 22 layers in total, resulting in a more deep model respect AlexNet. It uses average pooling instead fully-connected layer to reduce the computational overhead. Furthermore, a 1x1 convolution layer is used to debase the computational complexity. Two auxiliary classifiers are attached to the outputs of two of the inception modules for forwarding propagation, to elude *vanishing gradient* that makes the network hard to train.

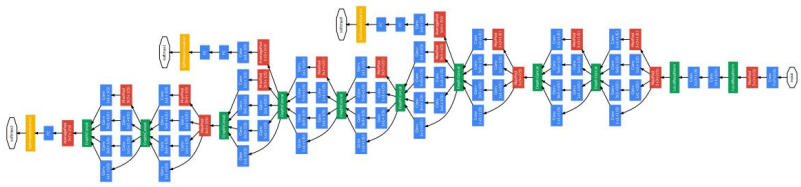


Figure 2.10: GoogLeNet network [6]

There are also other two versions of this architecture: Inception v2, and Inception v3 [35]. These are improved networks, that use other techniques to increase the accuracy and reduce the computational complexity (such as dimensionality reduction, convolution kernel factorization/decomposition, and batch normalization).

DenseNet121 [7] — The authors grasp the idea of shorter connections between layers and connect each layer to every other subsequent layer in a feed-forward fashion. This means that any feature maps of each layer are used as inputs for the subsequent layers, as well as receiving feature maps from all preceeding layers. DenseNet has the following pros: reduction of *vanish gradient* problem, strengthen feature propagation, encourage feature reuse and reduce the number of parameters. A dense block architecture is shown in Fig. 2.11

2.2.3 Convolutional Neural Network and Multi-Task Learning

Multi-task learning (MTL) is a subfield of machine learning, in which a model learns multiple tasks simultaneously. Such approaches offer advantages like improved data efficiency, reduced overfitting through shared representations, and fast learning by leveraging auxiliary information. [36]

Multi-Task Architectures for Computer Vision: in literature several architectures were proposed for different aims: Computer Vision, Language Processing, Reinforcement Learning, Multi-Modal (that is capable to handle multiple tasks from multiple domains, e.g. visual and linguistic data). In this section, there will be a brief overview of principal architecture in the field of Computer Vision. Many MTL architectures in CV branch the network into task-specific parts and use shared components for

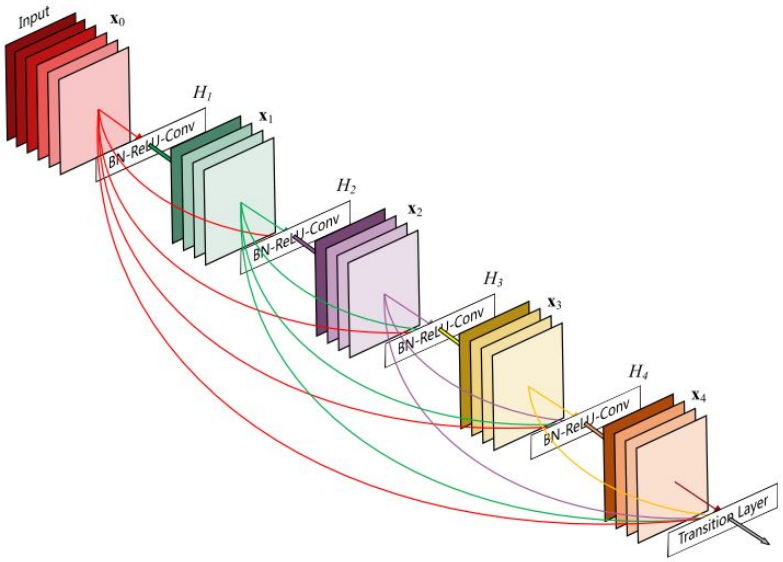


Figure 2.11: A 5-layer dense block. Each layer takes all preceding feature-maps as input. [7]

improving generalization over sharing and information flow between tasks, at the same time minimize the negative transfer.

Shared Trunk [36] — The more traditional multi-task architecture is the following: a base network used for features extraction on all tasks (i.e. a global features extraction)

Cross-Talk [36] — This architecture has a separate Neural Network for each task, and the information stream between parallel layers in the task networks. These layers are named *cross-talk*. The idea is that the input of every layer is a linear combination of the output of the previous layers from all task networks. The weights of each linear combination is learned and task-specific, in this way each layer can select which tasks to use for its advantage (i.e. from which tasks get information).

Prediction Distillation [36] — This type of architectures are founded on the following principle: learned features from one task may be useful in performing another related task (i.e. helping to learn another task)

Task Routing [36] — Both shared trunk and cross-talk architecture are founded on an unflexible scheme about parameter sharing. The novel component of this architecture is a layer that applies a task-specific binary mask, named Task Routing Layer to the output of a Convolutional Layer. The sharing occurs at the features level instead of the layer level The binary masks are not learned, the user has control over the degree of sharing between tasks using a hyperparameter "sharing ration" σ which can take values from 0 to 1. This binary mask is randomly initialized at the beginning of training and then fixed from that point on. The range of these hyperparameters specify the proportion of units in each layer which are task-specific.

The drawback of this architecture is an increase of number of parameters.

2.3 Machine Learning application to Skin Lesion Diagnosis

2.3.1 Methods proposed

There are a range of Machine Learning algorithms frequently used in dermatology, as summarized in [37]

In the following section the most recent and relevant works will be presented.

- **Hameed, et. al in** [38], proposed hybrid approach i.e. using deep convolution neural network and support vector machine (SVM). The proposed scheme is designed, implemented and tested to classify skin lesion image into one of five categories, i.e. healthy, acne, eczema, benign, or malignant melanoma. Experiments were performed on 9,144 images obtained from different sources. AlexNET, a pre-trained CNN model was used to extract the features. the overall accuracy achieved is 86.21%.

- **Jainesh Rathod et. al** [39] This system will utilize computational technique to analyze, process, and relegate the image data predicated on various features of the images. Skin images are filtered to remove unwanted noise and also process it for enhancement of the image. Feature extraction using complex techniques such as Convolutional Neural Network (CNN), classify the image based on the algorithm of softmax classifier and obtain the diagnosis report as an output. An initial training gives the output accuracy of 70% approximately.

2.3.2 Methods proposed on 7ptDerm dataset

The dataset used in this work is Derm7pt publicly available. Related works on this dataset are the following:

- **Kawahara et al., 2019 [40]** - this work proposes a deep convolutional neural network multi-task trained on multi-modal data (i.e. on images both clinical and dermoscopic, and patient meta-data). The aim is to classify each sample through the 7-point melanoma checklist criteria and predict skin lesion diagnosis. The neural network uses a single optimization (i.e. multi-task structure). They use different loss functions, each one considers a diverse combination of input modalities. They reported results for experiments setup with all the input modalities and the unbalanced dataset, in other words, the original dataset without the application of any balancing technique.

- **Coppola et al., 2020 [10]** - The purpose of this work is to obtain a single neural network capable of classyfing skin lesion diagnosis, as well as the seven

patterns, belong to the 7-point checklist. The network take only dermoscopic images as input. Furthermore they develop a mechanism to share between tasks only information relevant to them from other tasks. They use learnable gates for each task, that regulate wich features coming from other tasks would be useful for the specific task they refer to.

- **Somfai et al., 2021** [41] - the goal is to bridge machine learning applications and human in melanoma detection scenario. They build a system combining: visual pre-processing, deep learning, and ensembling to provide explanations to experts and minimize false-negative rate, whilst maintaining high accuracy. They assemble a skin lesion classifier by building a number of deep CNN-based classifiers (called *feature classifiers*) which receive preprocessed input images typically focused on the individual criteria of the ABCD melanoma classification rules, and fuse their prediction via a shallow neural net.

- **Li et al., 2020** [42] - their research focuses on the lack of generalization in deep neural networks when trained on limited datasets (this is very common with medical image).[2]. They propose to learn a representative feature space through variational encoding with a novel linear-dependency regularization term to capture the shareable information among medical data collected from different domains. They adopt seven public skin lesion datasets, including HAM10000, Dermofit, Derm7pt, MSK, PH2, SONIC , and UDA. Their method is based on two different medical imaging classification tasks: skin lesion classification task and gray matter segmentation task of spinal cord.

- **Lucieri et al., 2020** [43] - the work aims to clarify a deep learning based medical image classifier developed by REasoning for COmplex Data (RECOD) Lab for classification of three skin tumors: Melanoma, Seborrhei Keratosis, and Melanocytic Naevi. They used derm7pt and PH2 skin disease datasets for experimentation. Concepts understandable by human are mapped to RECORD image classification model with the help of Concept Activation Vectors. The authors sustain that the work increase confidence of dermatologists on Computer-Aided Diagnosis (CAD) systems and can play a fundamental role for further development of CAV-based neural network interpretation methods.

- **Bdair et al., 2021** [44] proposed `FedPerl`, a semi-supervised federated learning method that utilizes peer learning from social sciences and ensemble averaging from committee machines, to build communities and encourage its member to learn from each other such that they produce more accurate pseudo labels. They validated `FedPerl` on 38,000 skin lesion images coming from

[2]Trained deep neural network on data within a certain distribution may not be able to generalize to the data with another distribution [42]

ISIC19, HAM10000, Derm7pt, and PAD-UFES. They opt for EfficientNet as a backbone architecture and initialized weights through Xavier.

Chapter 3

Methods

The present work proposes different methods to address both the diagnosis task and the 7-pt checklist tasks. Firstly the dataset used is presented, together with pre-processing steps made. Secondly, different approaches belonging to *Single-Task Learning, Multi-Task Learning, and Concept-Bottleneck Models* are introduced. Single-Task Learning (STL) methods aim to learn one task; i.e. diagnosis. Multi-Task Learning (MTL) approaches have been developed to predict seven tasks; i.e. the tasks related to 7-pt checklist rule. The idea of MTL is that information acquired by a learning system regarding one specific pattern can influence the learning of other patterns and vice-versa. The Concept-Bottleneck Models (CBM) are inspired by [14] and they have been choosen in order to improve the explainability of Machine Learning models as well as their trustability.

3.1 Dataset description and preprocessing strategies

The dataset used is Derm7pt (from the *Interactive Atlas of Dermoscopy* [45]), which was publicly released with [40]. It is a database provided for evaluating computerized image-based prediction of a skin lesion diagnosis and of the the seven-point checklist criteria. It consists of over 2000 dermoscopy and clinical color images, with corresponding structured patient meta-data that includes other types of information, such as patient gender and lesion location. This dataset has been noted to have "excellent interobserver agreements", and was used to teach dermatologists, suggesting that it is a suitable source for training machine learning algorithms [40]. In Tab. 3.1 there is a summary of the database. Each entry in the dataset is named *case*, and has data available in multiple modalities (i.e. clinical and dermoscopic images and metadata). For each case labels are available for 8 categories: *pigment network, regression structures, pigmentation, blue whitish veil, vascular structures, streaks, dots and globules, diagnosis*. These categories will be used as the 8 tasks to learn in the MTL architecture explained in 3.3.2, so in the following the terms *task* and *category* will be used interchangeably. These categories also will be used for evaluating of the 7-point checklist criteria, introduced in 2.1.3.

Table 3.1: Section headers indicate the categories; *abbrev* indicate the abbreviation for the label and the grouping used in the experiments, *name* is the full name of the label; *7-pt score* is the contribution to the criteria ("-" is no contribution); *no. of images* indicates how many images exist with the particular label

abbrev	name	7-pt score	no. of images
DIAGNOSIS (DIAG)			
NEV	nevus	-	575
MEL	melanoma	-	252
Seven point criteria			
1. Pigment Network (PN)			
ABS	absent	0	276
TYP	typical	0	335
ATP	atypical	2	216
2. Blue Whitish Veil (BWV)			
ABS	absent	0	644
PRS	present	2	183
3. Vascular Structures (VS)			
ABS	absent	0	690
REG	regular	0	73
IR	irregular	2	64
4. Pigmentation (PIG)			
ABS	absent	0	484
REG	regular	0	82
IR	irregular	1	261
5. Streaks (STR)			
ABS	absent	0	494
REG	regular	0	96
IR	irregular	1	237
6. Dots and Globules (DaG)			
ABS	absent	0	134
REG	regular	0	301
IR	irregular	1	392
7. Regression Structures (RS)			
ABS	absent	0	594
PRS	blue areas	1	223

3.1.1 Dataset pre-processing

Among the available modalities in the dataset, only the dermoscopic images have been used in the experiments because they have a higher resolution and allow to appreciate better the patterns on the lesion that are necessary for the seven-point criteria.

Grouping of granular labels

The original dataset is defined with 43 different labels at the most granular level. However, most labels occur infrequently and have comparable clinical interpretation (e.g. types of benign nevi), thus in [40] they grouped infrequent labels with similar clinical interpretation into a single label, and obtain a total number of in 24 labels with higher granularity. For example, in the diagnosis category, the NEV label groups all the nevi labels (e.g. blue nevus, clark nevus, etc) into a single label. Table 3.1 shows the available labels for each of the tasks studied in this work and the relative number of sample in that category.

Adjustments to dataset

The dataset was modified through these steps:

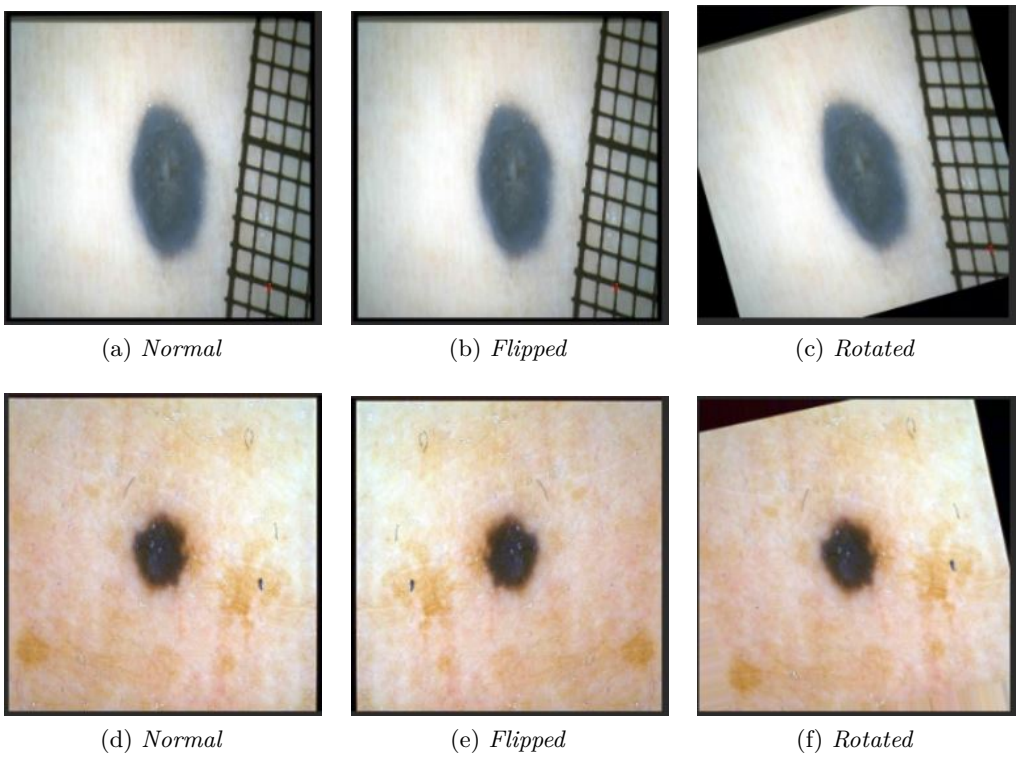
1. Removal of unnecessary samples: the focus of this work is building a model to distinguish nevi (NEV) from melanoma (MEL); thus the samples with other diagnoses present in the dataset (such as Basal-cell carcinoma, seborrheic keratosis and others) have been excluded in the experiments. Table 3.1 shows the tasks with corresponding labels, and how many images exist with the particular labels, after this pre-processing step.

2. Dataset splitting: the samples in the dataset have been split in training, validation, and testing subset following [40]. Therefore the number of samples per subset is as follows:

   - Train set: 346
   - Valid set: 161
   - Test set: 320

3. Sanity check: verification that images contained in the split datasets aren't duplicates. As a matter of fact, whether an image of val or test sets are also present in the training set we would have Data Leakage[1]. The `case_num` (i.e.

[1] Data Leakage, sometimes called train-test contamination occurs when validation or test data corrupt training data. In this way the model is trained and evaluated on the same data, so it may get good performance but perform poorly when it will be deployed to make decisions[46]



(a) *Normal* (b) *Flipped* (c) *Rotated*

(d) *Normal* (e) *Flipped* (f) *Rotated*

Figure 3.1: Example of data augmentation of 2 samples, with random horizontal/vertical flip, and random rotation within a range of 20°.

ID) of skin lesion is beign used to address this task.

Data Augmentation

One way to improve the generalization performance of machine learning model is to train on more data: the more samples the network has to learn from, the better it will be able to identify which differences in images matter and which do not for the classification. The data augmentation technique allows using of the data owned to obtain more data. The idea is to transform the images in the database in ways that preserve the labels in categories. In this way, the classifier learns to ignore those kinds of transformations. For instance, whether melanoma is facing left or right in a photo doesn't change the fact that it is a Melanoma and not a Naevi. An example of data augmentation is shown in Fig. 3.1. In this case, the model will learn that modifies such as rotation or flips, are diversity that it should overlook. In fact, this project works whit a relatively small dataset and the following types of data augmentation have been used:

- Random horizontal and vertical flip

- Random rotation within a range of 20° and the points outside the boundaries of the input are filled according to a mode called *nearest*, in which points are occupied with the same value of the nearest pixel.

An example of data augmentation implemented is shown in 3.1

Imbalance problem: balanced mini-batch sampling at training time

The dataset is imbalanced, thus it may happen that several mini-batches do not include one of the unique labels, meaning that the neural network will seldom be optimized for that label. For this reason, this work employs the same mini-batch sampling scheme described in [40]: at each training iteration, k cases are randomly sampled from the training set for each of the unique target labels present in the dataset. Consequentially, in every mini-batch there are at least k elements belonging to each unique label. As observed in Tab. 3.1, unique labels are 21 across the 8 tasks considered. This means that $b = 21k$ is the size of a each mini-batch during training.

3.2 Problem definition and objective functions

3.2.1 Formal encoding: one-hot method

Formally, a set is given of n color images $x_s \in X$ and their corresponding one-hot encoded labels for T tasks

$$\mathbf{y}_s = \{\mathbf{y}_s^1, \mathbf{y}_s^2, ...\mathbf{y}_s^T\} \in Y \subset \mathbb{R}^{J^1} \otimes \mathbb{R}^{J^2} \otimes ...\mathbb{R}^{J^T} \tag{3.1}$$

where $\otimes$ is the direct product, $s = 1, ...n$ is the sample index and $J^1, ...J^T$ indicate the number of labels in each task t. Thus each

$$\mathbf{y}_s^t = \left[y_{s,1}^t, y_{s,2}^t, ...y_{s,J^t}^t\right] \tag{3.2}$$

is the vector of one-hot encoded lables for sample s for task t. The aims of both STL and MTL problems is to find a neural network g_θ, which has a set of parameters such that $g_\theta : X \to Y$ minimizes a chosen objective function.

3.2.2 Activation function

The output function is a transformation that is applied the output vectors of a neural network before the loss computation. For all architectures proposed *Softmax* activation function has been used. This function, also known as *softargmax* or *normalized exponential function* is a generalization of the logistic function to multiple dimensions. Given a categorical distribution also referred as multinoulli distribution (i.e. a discrete probability distribution that describe the possible results of a random

variable that can take on one of K possible categories), the softmax function is often used to predict the probabilities associated with this types of distributions [30]. The softmax function is defind to be $\sigma : \mathbb{R}^K \rightarrow [0, 1]^K$

$$softmax(x_i) = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)} \tag{3.3}$$

A softmax function take as input a vector $\mathbf{z}$ of K real numbers, and normalize it into a probability distribution consisting of K probabilities proportional to the exponential of the input numbers. This means that the input of the function can be any real number, but its output belong to the interval (0,1).

3.2.3 Loss function

Categorical cross entropy

The categorical cross entropy (CE) for each sample s is defined as

$$\phi(\mathbf{y}_s, \hat{\mathbf{y}}_s) = -\sum_{j=1}^{J^t} y_{s,j} log(\hat{y}_{s,j}) \tag{3.4}$$

Where J^t is the cardinality of label set of a task t, $\mathbf{y}_s \in \mathbb{R}^{1 \times J^t}$ and $\hat{\mathbf{y}}_s \in \mathbb{R}^{1 \times J^t}$ are respectively the one-hot encoded ground truth and the softmax activated predicted output of the network. *Cross entropy* is *entropy* between two probability distributions as defined by Shannon in [47]. This function measures the average number of bits required to send a message from distribution A to Distribution B. Thus, it is used for compute the difference between two probabilities: ground truth distribution and prediction distribution. In this way the algorithm attempts to minimize this function in order to achieve its goal; i.e. predict values equal as much is possible to ground truth. In other words, the ML model ideally wants to obtain two equal probability distributions.

Focal categorical cross entropy

Focal cross-entropy (FCE) was first introduced in [48]. This loss function scales the standard cross-entropy by a factor, which expresses the confidence of the model in classifying a sample; i.e. inputs that are classified easily have major confidence and should have a minor impact on the loss function, thus they should have a higher scale value. The classification loss function for one sample s is:

$$\xi(\mathbf{y}_s, \hat{\mathbf{y}}_s) = -\sum_{j}^{J^t} y_{s,j}(1 - \hat{y}_{s,j})^{\beta} log(\hat{y}_{s,j}) \tag{3.5}$$

For both CE and FCE J^t is the cardinality of label set of a task t. While $\mathbf{y}_s^t \in \mathbb{R}^{1\times J^t}$ and $\hat{\mathbf{y}}_s^t \in \mathbb{R}^{1\times J^t}$ are respectively the one-hot encoded ground truth and the softmax activated predicted output of the network, as defined in Eq. 3.1 3.2. The hyperparameter β is equal to $\beta = 2$, how is presented in findings in [48].

3.3 Proposed Architecures

In this section different approaches are presented. Firstly models belonging to different methods: *single-task learning and multi-task learning*. Secondly different models which implement *concept bottleneck framework*. The first approach aims to classify only the task related to the diagnosis (i.e. DIAG category), the latter focuses also on the classification of the seven-point checklist tasks, thus the networks presented in *multi-task learning* approach compute 8 tasks in total (the diagnosis and the seven-point). For some architectures, the transfer learning technique has been used, defined in [49] as follows: inductive transfer refers to any algorithmic process by which structure or knowledge derived from a learning problem is used to enhance learning on a related problem. Specifically, state-of-the-art architectures pre-trained on the ImageNet dataset[2]

3.3.1 Single-task architectures

This paragraph describes the single-task models used for the classification of diagnosis. These models consist of a base architecture taken from the literature, which is used as a feature extractor, followed by a series of layers to process the features into classification logits. The structure of the single-task models is represented in Fig. 3.2. The base architectures used in the experiments are IncepionV3 [35], ResNet50 [5], and DenseNet121 [7]: a brief summary of the characteristics of these models have been given in Sec. 2.2.2. These base models are initialized with weights pre-trained on the ImageNet dataset. For add regularization, a 30% dropout has been introduced at the deeper levels after the Global Average Pooling (Fig. 3.2). These STL models have trained with the categorical cross-entropy objective function, as defined in 3.4.

3.3.2 Multi-task architectures

In this paragraph the architectures that aim to predict 8 different tasks simultaneously (Tab. 3.1) will be presented. Similarly to the single-task architectures, these architectures use popular architectures as base of CNN. The base networks used are ResNet50, InceptionV3, and DenseNet121. The latter choice has been made in

[2]ImageNet is a database of images formed according to the WordNet hierarchy (presently only the nouns) in which each node of the hierarchy is depicted by hundreds and thousands of images. The models used are ResNet50, InceptionV3 and DenseNet121, briefly recapped in Sec. 2.2.2 have been employed in the experiments. `https://www.image-net.org/`

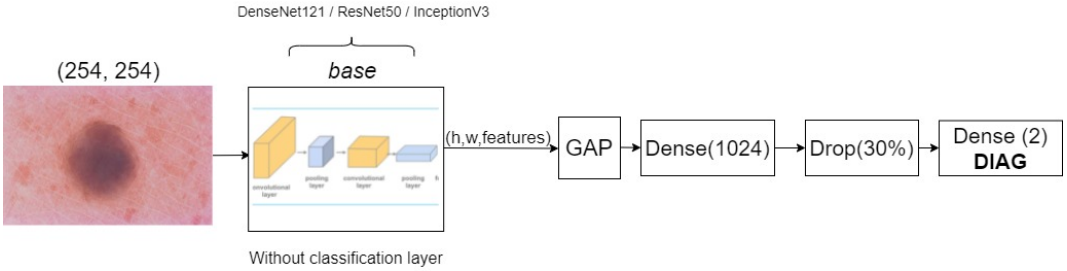


Figure 3.2: Single-Task proposed architectures with use of different bases: InceptionV3, ResNet50, and DenseNet121. The weitghts are initialized on pre-trained bases on ImageNet dataset.

order to reduce overfitting. In fact, the number of parameters of DenseNet121 is approximately one-third compared to ResNet50 and Inceptionv3[3] (i.e. 8,062,504). The classification layer of each base architecture has been removed and 8 different branches are added: one for different tasks. Each task branch has the same architecture. Thus different *bases* provide different features extraction and get different performances. The common structure is showed in Fig. 3.3 The choice to put equal branches structure allows comparing better the models. **Loss function** is the same for all models. In particular for each task a *focal cross entropy* $\xi(\cdot)$ has been applied, as characterized in 3.5, and the total loss of MTL models is defined as

$$\Xi = \sum_{t \in T} \xi_t(\mathbf{y}_s^t, \hat{\mathbf{y}}_s^t) \tag{3.6}$$

where $T = \{DIAG, PN, BWV, VS, PIG, STR, DaG, RS\}$

3.3.3 Concept bottleneck model architectures

In this paragraph three *Concept Bottleneck Models* (CBM) [14] approaches will be presented. The first part will briefly present the idea of the CMB, the subsequent one introduces the architecutres proposed in this work. The authors of the original paper [14], seek to learn models that allow to interact with high-level concepts. Whether a model is deployed, it will be used by physicians, thus the essential idea is that clinicians can manipulate concepts predicted by it, and propagating these changes to the final prediction. For instance, in this work the concepts are the seven criteria of 7-pt checklist rule and the final prediction is the skin lesion diagnosis. This type of model also empowers human-model interaction: the authors show that the accuracy increases significantly if it may correct model misconceptions at test time [14]. This type of interaction has not been studied in this manuscript.
The aim is to predict a target y, given an input $x \in \mathbb{R}^d$. The training points also

[3]ResNet50 and InceptionV3 have 25,636,712 and 23,851,784 parameters respectively

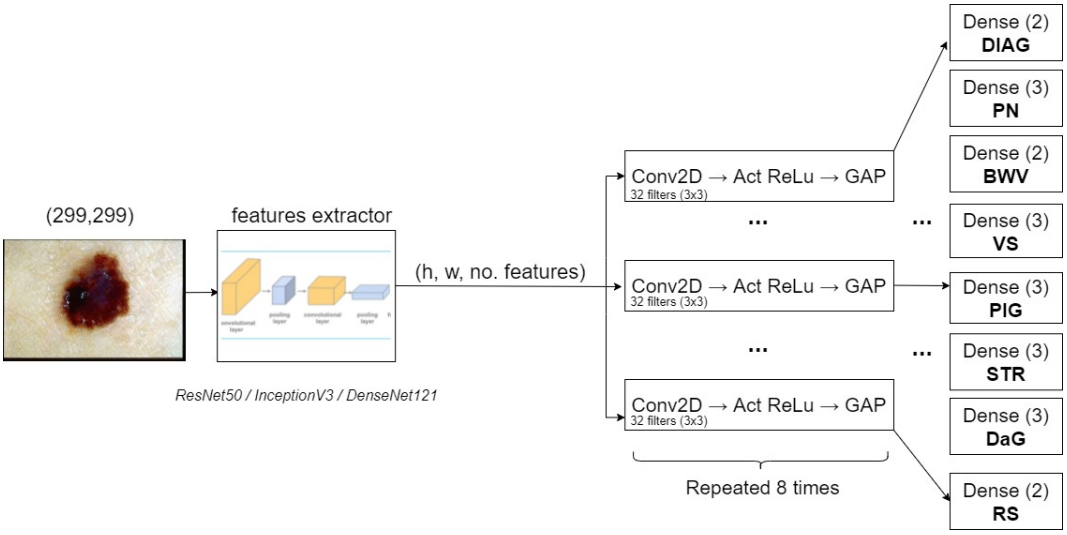


Figure 3.3: Multi-task Architecture with 8 branches: one for DIAG task and 7 for Seven-Point checklist patterns. This strcture represents the three different archictectures implemented, in which the difference is in the base: ResNet50, InceptionV3 or DenseNet121.

convey an intermediate representation defined as $c \in \mathbb{R}^k$, that is a vector of k concepts. Thus training samples are defined in the following way $\{x^{(i)}, y^{(i)}, c^{(i)}\}$ for $i = 1...n$. Where n is the number of available training samples.
A CBM is characterized by $f(g(x))$, where $g(\cdot)$ maps an input image into the concept space (for istance: "pigmentation", "dots and globules", "streaks", etc.) and $f(\cdot)$ maps these concepts into final label (i.e. the presence or absence of melanoma on skin lesion).
The fundamental difference of CMB compared to standard end-to-end systems, is that the prediction $\hat{y} = f(g(x))$ depends on the $\hat{c} = g(x)$ that is trained to achieve a component-wise to the concepts c.

The authors of *Concept Bottleneck Models* [14] methodically experiment three configurations of $f(\cdot)$ and $g(\cdot)$:

- *Independent bottleneck*: $g(\cdot)$ and $f(\cdot)$ are trained indipendently using the ground truth labels. This means that there are two different models with distinct loss functions.

- *Sequential bottleneck*: in this case there are two different models which are trained sequentially. Firstly, $g(\cdot)$ is trained to predict the concepts vector. Secondly, the prediction of $g(\cdot)$ are used as input to $f(\cdot)$ during the training phase.

- *Joint bottleneck*: is defined similarly to the *Sequential bottleneck*, with the difference that the two models are trained simultaneously; i.e. the loss function

is a sum of the loss functions of the two models.

The following paragraphs will describe the concept bottleneck model configurations that have been used in this work.
Figures 3.4a and 3.4b illustrate $g(\cdot)$ and $f(\cdot)$ architectures respectively.

Independent configuration

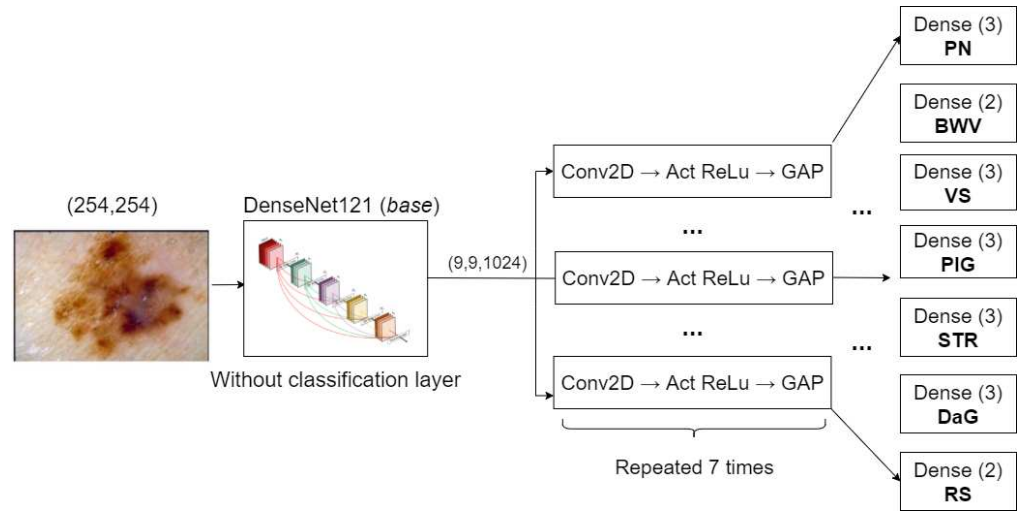
This structure consists of two models, trained indipendently on actual images for $g(\cdot)$ and actual concepts for $f(\cdot)$.

1. $g(\cdot)$ for $x \rightarrow c$
   A convolutional neural network, that given an input image x, make predictions on concepts c. DenseNet121 has been chosen as base architecture, as it showed the best results in the multi-task learning setting experiments (see section 4.2.2). The architecture is presented in section 3.3.2 in which the diagnosis task has been removed. The neural network has been trained on actual images. The loss function is *focal cross entropy* for each task, as defined in Eq. 3.5. Thus, the total loss is computed as:

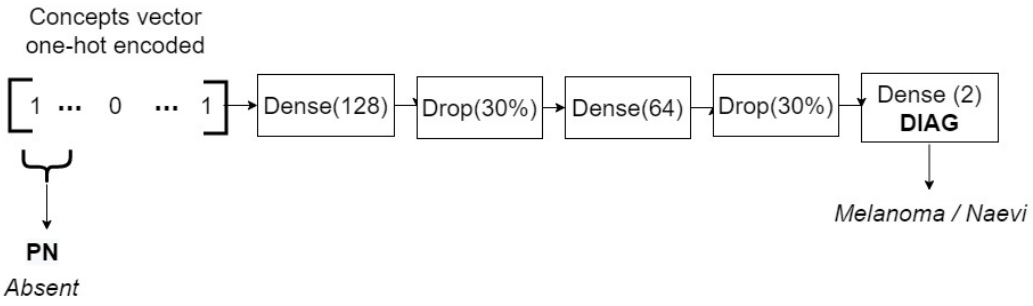
$$\Xi = \sum_{s}^{b} \sum_{t \in T} \xi_t(\mathbf{y}_s^t, \hat{\mathbf{y}}_s^t) \tag{3.7}$$

   where $T = \{PN, BWV, VS, PIG, STR, DaG, RS\}$; i.e. the 7 attributes of the 7-point checklist rule.

2. $f(\cdot)$ $c \rightarrow y$
   A multi-layer perceptron, that given the concepts c make prediction of the diagnosis task y. The neural network has been trained using the true concepts as input, however at test time it receives as input the concepts predicted by the model $g(\cdot)$. The classification loss function for the diagnosis task for one sample is *categorical cross entropy* because this is the same loss function used in a STL (section 3.3.1), and this allows to compare the results. The categorical cross entropy for each sample s is defined in Eq. 3.4.



(a) $g(\cdot)$



(b) $f(\cdot)$

Figure 3.4: CBM: architectures of $g(\cdot)$ which predict concepts from raw images, and $f(\cdot)$ which predicts output target.

Sequential configuration

For this structure, the same models used in independent configuration have been chosen, and they are trained at different times. The difference compared to Independent CBM, is that MLP is trained on predicted concepts instead of actual concepts. **Loss function** is *categorical cross entropy*, defined in Eq. 3.4, because the final prediction is single-task: the diagnosis label. Within a mini-batch the loss function is

$$\Phi = \sum_{s}^{b} \sum_{t \in T} \phi_t(\mathbf{y}_s^t, \hat{\mathbf{y}}_s^t) \tag{3.8}$$

where $\mathbf{y}_s^t = \hat{g}(x)$ and $T = \{PN, BWV, VS, PIG, STR, DaG, RS\}$, the 7 patterns of the 7-point checklist.

Joint configuration

This structure aims to minimize both task and diagnosis losses simultaneously. The *focal categorical cross entropy* (Eq. 3.5) and *categorical cross entropy* (Eq. 3.4) presented for above configurations have been used. The loss function of joint model is

$$L_{joint} = \sum_{s}^{b} [\ \sum_{t}^{T} \lambda \xi(\mathbf{y}_s^t, \hat{\mathbf{y}}_s^t) - \phi_{DIAG}(\mathbf{y}_s^{DIAG}, \hat{\mathbf{y}}_s^{DIAG})] \tag{3.9}$$

where λ regulates the weight of concepts loss in contrast to task loss.

Chapter 4

Results

This chapter describes and discusses the most relevant results obtained while experimenting with the proposed architectures.

4.1 Experimental setup

The proposed architectures have been implemented in Python 3.8 using the popular deep learning framework TensorFlow [50] (version 2.3.0). The experiments were carried out on GPU NVIDIA GeForce GTX TitanX which has 12 GB of memory. Despite the GPU is faster than a CPU, the less dedicated RAM allows a smaller mini-batch size given the size of the model and the inputs. The images have been resized to 254 x 254 to improve efficiency.
The dataset has been split into train, valid, and test sets as defined in Section 3.1. The training has been performed on the entire training dataset for all the experiments. In machine learning, many strategies are explicitly designed to reduce the error of the model with new data, and they are known collectively as regularization [30]. In this research, the model performance at the end of each epoch has been evaluated through the validation set, with a regularization technique called *early stopping*.
Tests have been performed in a *5-fold validation* fashion: the whole test dataset is equally divided into five parts, and for each iteration, only one part constitutes the test set. Successively, mean and standard deviation across the five test folds have been calculated.

4.1.1 Metrics

This section summarizes the metrics used for the evaluation of experiments. They have been implemented using the scikit-learn library [8].
All the confusion matrices are calculated on the entire test set. The remaining metrics have been computed on 5-folds test set and the results are reported as mean and standard deviation of these split. Furthermore note that only accuracy and F-measure are reported in this chapter. Precision and recall can be found in the Appendix 6. Precision and recall are evaluated on the positive label (MEL =

melanoma) for the diagnosis task. For non-binary tasks, these metrics are evaluated as a weighted average (based on the number of true samples for each label).

Confusion Matrix

A Confusion Matrix is built for each model trained to evaluate the metrics of a classification. By definition, a confusion matrix C, is such that each entry of the matrix, is equal to the number of observations belong to the label indicated in the row but predicted to label indicated by column. For example, in binary classification, the count of true positive is tp, false positive is fp, true negative tn and false negative fn. An example is shown in Tab. 4.1. In this work the confusion matrices are computed without any normalization, which means that the information of number they convey exhibits the exact sum of samples predicted to any class.

Accuracy

The accuracy function computes the fraction (i.e. with normalization) of correct predictions. Both in STL and in MTL, the accuracy is computed for any task separately. If the $\hat{y}_i$ is the predicted value of the $i-th$ sample and y_i is the corresponding true value, then the fraction of correct predictions over $n_{samples}$

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbb{1}(\hat{y}_i = y_i) \tag{4.1}$$

where $\mathbb{1}(x)$ is the indicator function.

Precision

Precision is the ratio

$$\frac{t_p}{t_p + f_p} \tag{4.2}$$

where t_p represents the number of true positive, while f_p is the number of false positive. The best score is 1 and the worst one is 0. An intuitive explanation of

Tot. population = p + n	**Predicted positive**	**Predicted negative**
Actual positive	*tp*	*fn*
Actual negative	*fp*	*tn*

Table 4.1: Definition of confusion matrix C with *tp, tn, fp*, and *fn* indicated.

this metric is the following: it is the ability of the model not to label as positive the negative samples.

Recall

Recall is the ratio

$$\frac{t_p}{t_p + f_n} \tag{4.3}$$

where t_p represents the number of true positive, while f_n is the number of false negative. The best value is 1 and the worst value is 0. This metrics is intuitively the ability of the classifier to find all positive samples

F1-score

F1-score, also known as *balanced F-score* or *F-measure* can be interpred as a weighted average of the precision and recall. Its best value is 1 and the worst score is 0. Both precision and recall give the same relative contribution to the F1 score. F-measure is formally defined as

$$2\ \frac{(precision \times recall)}{precision + recall} \tag{4.4}$$

4.2 Experiments

In the following pages the most significant experiments have been selected among all experimentations carried out. This section is intended to show the thought process leading to the final setting of the *concept-bottleneck* models. Initial experiments have been carried out on single-task learning models with the melanoma diagnosis as main task. These were followed by multi-task learning models over the 8 available tasks (diagnosis + 7 checklist attributes). Finally, the best performing base architecture was chosen as backbone for the experiments employing the concept-bottleneck model framework. A summary of the experiments including the relevant hyperparameters and performance can be found in Tab. 4.2. In-depth performance for the experiments are reported in Tab 4.4 to Tab. 4.7.

Base	**type**	**F. Ar.**	**Loss**	η	**Im.N.**	**m.b.s**	**Dr.**	**7p**$_\mu$	**DIAG**
RN	STL	3.2	ϕ	10^{-3}	Yes	40	30%	-	79.1
Inc	STL	3.2	ϕ	10^{-3}	Yes	40	30%	-	79.4
DN	STL	3.2	ϕ	10^{-3}	Yes	40	30%	-	80.0
RN	MTL	3.3	ξ	10^{-4}	Yes	42	-	47.2	34.1
Inc	MTL	3.3	ξ	10^{-4}	Yes	42	-	69.34	80.0
DN	MTL	3.3	ξ	10^{-4}	Yes	42	-	71.24	81.2
DN	CBM-CNN	3.4a	ξ	10^{-4}	Yes	42	-	68.2	-
-	CBM-MLP	3.4b	ϕ	10^{-4}	-	21	30%	-	76.2
-	CBM-seq	3.4	ϕ	10^{-4}	Yes	42	30%	-	78.4
.	CBM-joint	3.4	$\phi + \xi$	10^{-4}	Yes	42	30%	71.9	82.2

Table 4.2: Summary of the experiments. Column **Base** refers to the features extraction base of each model, if it is applicabile, where RN = ResNet50, Inc = InceptionV3, and DN = DenseNet121; **type** indicates whether the model belongs to single-task, multi-task learning or concept-bottleneck framework; **F. Ar.** collects the direct referiments to full architectures; **Loss** is the loss funcion (i.e. categorical cross entropy ϕ, defined in Eq. 3.4 or focal cross entropy defined in Eq. ξ 3.5); η represents learning rate; **Im.N.** is whether the model use the ImageNet as initialization of parameters for the base. **m.b.s** is mini-batch size; **Dr.** is dropout where applicable; **7p**$_\mu$ refers to the accuracy average of seven-point pattner of Tables 4.6 and 4.7; **DIAG** refers to the accuracy of diagnosis task.

Experiments have been conducted in chronological order as they are presented in the following paragraphs: from single-task learning, through multi-task learning and finally concept bottleneck models. In all the results concerning the diagnosis task, f1-score, precision and recall are computed with regard to the MEL label; i.e. diagnosis of melanoma.

4.2.1 Single-task learning

The architectures used in these experiments are presented in section 3.3.1. The proposed structure is common for all convolutional neural networks, the difference is in the three bases used as features extractors: ResNet50, InceptionV3, and DenseNet121. All the experiments for the single-task learning are carried out with the balancing sampling method introduced in paragraph 3.1.1. The step of the process is the following:

1. Evaluation of which labels to select for the balancing. In this phase, models are trained using both balancing on 'NEV'-'MEL' as unique labels as well as the 21 labels, presented in 3.1. Only results using ST ResNet50 have been reported, however the others architectures display comparable metrics. Results in Tab. 4.3 show that balancing on 2 unique labels 'NEV' and 'MEL' gives a slightly better performance, hence this is the balancing method used for all the subsequent STL experiments. The k parameter which defines the dimension of mini-batches is set to 20, in order to obtain a mini-batch size equal to $b = 20 \times 2$. Furthermore, the training curves reported in appendix 6.1.1 display that the balancing on 21 labels return a singular behaviour during the training. This is confirmed also by confusion matrices that have been reported in the appendix which express a better classification performance whether the model is trained with the balancing on 'NEV' and 'MEL'.

Model	μ_{ac}	$\pm\sigma_{ac}$	μ_{pr}	$\pm\sigma_{pr}$	μ_{rec}	$\pm\sigma_{rec}$	μ_{f1}	$\pm\sigma_{f1}$
ResNet50$_{21label}$	70.6	4.9	54.9	8.6	50.4	7.8	52.0	5.4
ResNet50$_{NEV-MEL}$	**79.1**	3.0	**63.6**	10.8	**69.2**	3.8	**61.1**	18.1

Table 4.3: STL: comparison between Balanced Batch Sampling on 21 unique labels and NEV-MEL labels. The mean and standard deviation of all metrics, are been computed on five folds of test set obtained with `StratifiedKFold` provided by [8].

Model	μ_{ac}	$\pm\sigma_{ac}$	μ_{f1}	$\pm\sigma_{f1}$	μ_{pr}	$\pm\sigma_{pr}$	μ_{rec}	$\pm\sigma_{rec}$
STResNet50	79.1	3.0	63.6	10.8	69.2	3.8	61.1	18.1
STInceptionV3	79.4	2.8	60.8	8.7	**75.2**	6.7	52.3	13.1
STDenseNet121	**80.0**	4.7	**66.2**	11.2	69.7	7.5	**66.2**	11.2

Table 4.4: STL: The mean and standard deviation are been computed on five folds of the test set obtained with `StratifiedKFold` provided by [8].

2. Experiments involving the other bases, are carried out with the finer balancing method studied in the first step. The results are reported in Tab. 4.4. The better accuracy is obtained with DenseNet121 as base. STInceptionV3 is the neural network which has higher precision, hence it is better in the prediction of true positive (i.e. melanoma cases). The preeminent recall is obtained wit STDenseNet121, thus this neural network has a lower error with false negatives. This suggests that the latter is the most effective model.

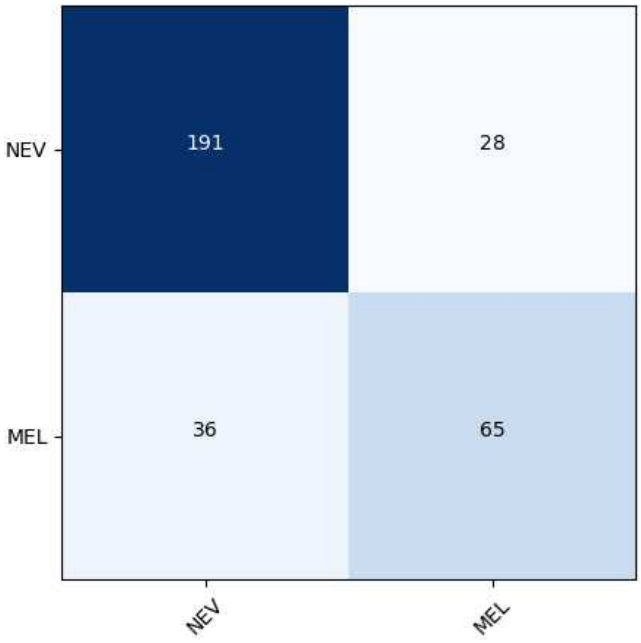


Figure 4.1: STL: STDenseNet121: confusion matrix. Columns are the predictions and rows are the ground truth.

The best performant model is 'STDenseNet121' for the accuracy and F-measure. The confusion matrix in Fig. 4.1 shows that the neural network recognize fairly well true positives and true negatives.
Additional experiments have been carried out without any balancing. In the table 4.5 are shown results obtained, that are perfectly aligned with what we expect: for all models the performance are significantly worst.

Model	μ_{ac}	$\pm\sigma_{ac}$	μ_{f1}	$\pm\sigma_{f1}$	μ_{pr}	$\pm\sigma_{pr}$	μ_{rec}	$\pm\sigma_{rec}$
STResNet50	**68.4**	0.6	55.6	0.9	46.8	0.9	68.4	0.6
STInceptionV3	58.75	4.6	57.1	5.0	56.0	5.3	58.7	4.6
STDenseNet121	58.1	4.8	**58.1**	5.01	**58.32**	5.3	**58.1**	4.8

Table 4.5: STL: The mean and standard deviation of all metrics, are been computed on five folds of test set obtained with `kStratifiedKFold` provided by [8]. The models have trained without any balancing

4.2.2 Multi-task learning

Multi-task learning architectures have a common structure as defined in Sec. 3.3.2. They are composed of a base that executes the extraction of features. These bases are the same used for the STL; i.e. ResNet50, InceptionV3, and DenseNet121. The classification layer of these bases has been removed, and an equivalent branch for each task is added, for the classification of diagnosis and seven-point patterns. The weights of the bases are initialized with the pre-trained models on *ImageNet* dataset, and during the training they have not been frozen. In order to fine-tune all parameters. The learning rate is set to 10^{-4} instead of 10^{-3} of the *single-task learning* based on experimental findings. In fact, the training curves for learning rate equal to 10^{-3}, behave singularly, with a constant trend for all tasks (see appendix 6.1.2). All the experiments for the multi-task learning are carried out with the balancing sampling methods introduced in paragraph 3.1.1, computed on the 21 unique labels. The results confirm that the best model is the one with DenseNet121 as feature extractor, as shown in Tab. 4.6. Thus, MT DenseNet121 is the model selected to execute experiments on the concept-bottleneck models in the following paragraph (Sec. 4.2.3).

Model	μ_{DIAG}	$\pm\sigma_{DIAG}$	μ_{PN}	$\pm\sigma_{PN}$	μ_{BWV}	$\pm\sigma_{BWV}$	μ_{VS}	$\pm\sigma_{VS}$	μ_{PI}	$\pm\sigma_{PI}$	μ_{ST}	$\pm\sigma_{ST}$	μ_{DaG}	$\pm\sigma_{DaG}$	μ_{RS}	$\pm\sigma_{RS}$
Accuracy																
MTDenseNet121	**81.2**	5.6	**60.3**	10.2	**86.6**	1.4	**82.5**	10.2	**70.6**	6.3	**65.0**	2.1	60.9	5.8	72.8	3.9
MTInceptionV3	80.0	3.0	52.5	7.8	86.6	5.9	81.6	5.2	67.2	7.2	62.2	7.8	**62.5**	8.5	**72.8**	3.0
MTResNet50	34.1	3.0	32.8	2.5	25.0	3.1	81.2	0.0	51.6	5.3	28.7	0.9	49.4	0.9	61.9	4.1
F1-score																
MTDenseNet121	**65.9**	12.3	**60.1**	10.4	86.3	1.4	**81.9**	9.0	**66.8**	5.7	**64.2**	2.3	**60.1**	4.6	**70.7**	3.8
MTInceptionV3	65.6	7.3	52.3	7.9	**86.9**	5.3	79.1	5.6	66.6	7.3	62.4	7.2	59.6	7.9	66.9	4.9
MTResNet50	47.9	2.4	24.5	1.8	15.3	6.3	73.6	1.1	48.9	5.5	12.8	0.7	32.6	0.9	59.1	3.2
Precision																
MTDenseNet121	**76.9**	12.0	**60.9**	9.9	87.1	2.1	**82.9**	8.7	64.6	3.8	**65.7**	5.8	**61.3**	4.8	71.7	5.3
MTInceptionV3	71.5	6.3	54.0	8.1	**87.8**	3.5	80.4	8.7	**69.6**	5.7	64.8	7.4	60.1	7.9	**73.3**	6.4
MTResNet50	31.9	1.8	20.1	1.5	51.0	30.3	67.7	2.7	53.4	2.9	8.3	0.5	24.4	0.8	58.0	3.8
Recall																
MTDenseNet121	59.3	15.7	**60.3**	10.2	**86.6**	1.4	**82.5**	10.2	**70.6**	6.3	**65.0**	2.1	60.9	5.8	72.8	3.9
MTInceptionV3	61.3	10.6	52.5	7.8	86.6	5.9	81.6	5.2	67.2	7.2	62.2	7.8	**62.5**	8.5	**72.8**	3.0
MTResNet50	**96.0**	4.2	32.8	2.5	25.0	3.1	81.2	0.0	51.6	5.3	28.7	0.9	49.4	0.9	61.9	4.1

Table 4.6: MTL: The mean and standard deviation of all metrics, are been computed on five folds of test set obtained with `kStratifiedKFold` provided by [8]. The models have trained with balancing.

4.2.3 Concept Bottleneck Models

In this section, the most promitent results obtained during the experimentation of CBM frameworks are summarized. The different configurations and their training process have been defined in Sec. 3.3.3. Given the findings discussed in Sec. 4.2.2 the learning rate is set to 10^{-4}. All the experiments for both $g(\cdot)$ and $f(\cdot)$ are carried out with the balancing sampling method introduced in paragraph 3.1.1 and computed on the 21 unique labels.
Following the findings of the experiments using STL and MTL models, the DenseNet121 was chosen as base model for the CBMs employed in this section.
The results are shown in Tab. 4.7 and are commented below:

- Diagnosis task: over the three configurations, training according to the joint configuration yields higher accuracy, F-measure, precision and recall. The confusion matrix for this configuration is shown in Fig. 4.2: the model misclassifies more MEL samples rather than NEV.
- 7-pt checklist patterns: also for the concepts predictions, the joint configuration performs better for all concepts except for VS, in which the Convolutional Neural Network of independent configuration perform slightly better.

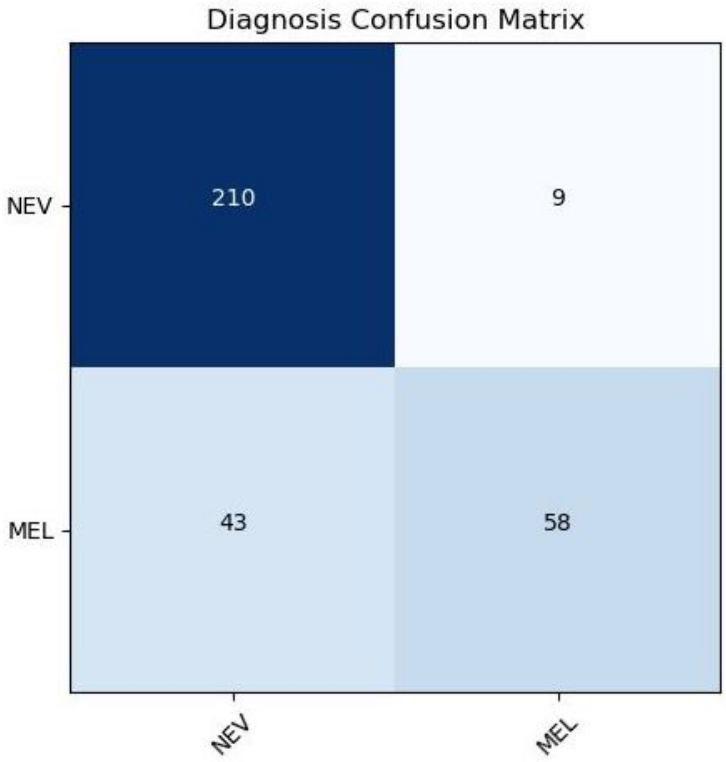


Figure 4.2: CBM joint: confusion matrix for the diagnosis target. Rows are true label, columns are predicted labels.

Model	μ_D	$\pm\sigma_D$	μ_{PN}	$\pm\sigma_{PN}$	μ_{BWV}	$\pm\sigma_{BWV}$	μ_{VS}	$\pm\sigma_{VS}$	μ_{PI}	$\pm\sigma_{PI}$	μ_{ST}	$\pm\sigma_{ST}$	μ_{DaG}	$\pm\sigma_{DaG}$	μ_{RS}	$\pm\sigma_{RS}$
								Accuracy								
Independent$_{CNN}$	-	-	59.1	10.6	83.8	3.6	81.9	5.1	68.1	2.1	65.0	3.2	54.4	6.8	72.5	5.5
Independent$_{MLP}$	76.2	7.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sequential	78.4	6.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Joint	**82.5**	4.2	**63.1**	4.6	**89.7**	4.4	**81.6**	**0.7**	**70.6**	5.8	**68.8**	3.3	**60.3**	11.8	**78.1**	5.2
								F1-score								
Independent$_{CNN}$	-	-	58.0	10.4	83.3	2.9	**77.3**	4.0	67.9	2.5	64.7	3.1	54.2	6.5	69.7	5.5
Independent$_{MLP}$	55.3	19.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sequential	57.9	15.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Joint	**65.9**	10.6	**61.0**	5.3	**89.1**	4.9	73.3	1.0	**70.0**	6.8	**67.4**	2.3	**59.3**	12.2	**76.8**	5.4
								Precision								
Independent$_{CNN}$	-	-	58.8	11.2	85.2	2.6	73.4	3.7	70.5	1.7	66.6	3.4	56.7	6.1	71.6	8.4
Independent$_{MLP}$	64.6	16.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sequential	72.0	10.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Joint	**83.0**	4.6	**64.3**	6.3	**90.0**	5.0	**66.5**	1.1	**72.4**	6.3	**69.4**	3.4	**61.5**	10.8	**77.6**	6.0
								Recall								
Independent$_{CNN}$	-	-	59.1	10.6	83.8	3.6	81.9	5.1	68.1	2.1	65.0	3.2	54.4	6.8	72.5	5.5
Independent$_{MLP}$	49.3	20.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Sequential	49.3	17.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Joint	**55.2**	13.2	**63.1**	4.6	**89.7**	4.4	**81.6**	**0.7**	**70.6**	5.8	**68.8**	3.3	**60.3**	11.8	**78.1**	5.2

Table 4.7: CBM: The mean and standard deviation of all metrics, are been computed on five folds of test set obtained with `kStratifiedKFold` provided by [8].

Model	7pt-avg
Accuracy	
CBM-Joint	73.1
F1-score	
CBM-Joint	70.9
Precision	
CBM-Joint	71.6
Recall	
CBM-Joint	73.1

Table 4.8: Mean of the best CBM model on seven patterns of 7-pt checklist patterns

4.2.4 Application of 7-pt rule on ground truth

In order to compare with the clinical ground truth, the 7-point checklist rule has been applied to the ground truth labels of the attributes of the test set. The predictions obtained in this way represent the diagnosis that would be obtained in clinical practice when applying the rule to correctly identified attributes.
For standard threshold $\tau = 3$ the findings about the accuracy of the are comparable to the STL, MTL., and CBM. While for the precision, the application of seven-point clinical rule is better only in Independent$_{MLP}$, which means that it gives less false positive. In opposite the recall is significantly higher with the application of seven-point score instead of any models presented. This suggests that the clinical rule rarely returns a false negative (i.e. it dignoses well negatives samples). The results are shown in Tab. 4.9.

4.3 Discussion

In this section, the results reported in the previous sections will be discussed and compared among each other and to another work that uses the same dataset.
Two macro-categories of architectures are proposed in this work: STL and MTL/CBM. The first does not yield the interpretation of the diagnosis, the latter includes both MTL and CBM. An overall comparison shows that across the models they do not differ a lot one to each other for the accuracy of the diagnosis. Further comparison between MTL and CBM insights that they are equiparable for the average accuracy on seven patterns of 7-pt checklist, for the DenseNet121 as base. Thus, the latter is could be selected in order to pursue future studies, in that it impelements hierarchy between concepts and the final diagnosis, which is intrinsic of the seven point rule.
For Single-Task Learning (STL) (Sec. 4.2.1) the best model is 'STDenseNet121' with

	μ_{ac}	$\pm\sigma_{ac}$	μ_{pr}	$\pm\sigma_{pr}$	μ_{rec}	$\pm\sigma_{rec}$	μ_{f1}	$\pm\sigma_{f1}$
$\tau = 3$	83.4	4.9	67.5	7.4	94.0	2.3	78.4	5.3
$\tau = 1$	56.9	3.6	42.3	2.3	99.0	2.2	59.2	2.4

Table 4.9: Seven-point rule on ground truth

acc = 80.0%±4.7 , F1-score 66.2±11.2 %, and recall = 66.2%±11.2. The precision metric is higher in the model 'STInceptionV3' with a value equal to 75.2%±6.7,

The Multi-Task Learning (MTL) (4.2.2) models have been trained to learn 8 tasks (diagnosis and attributes) without a hierarchy assigned to the tasks. For the diagnosis task the best model is 'MTDenseNet121' with a diagnosis accuracy of 81.2%±5.6, and a mean among seven-point patterns accuracy equal to 81.6%. 'MTDenseNet121' is also better with regard to the F1-score and precision metrics among the tasks. Surprisingly, the MTResNet50 model reaches a recall value equal to 96% for the melanoma diagnosis and for the seven clinical patterns the mean is 71.2%. This suggests that 'MTResNet50' achieves a low false negative rate, which is a desiderable behaviour in this setting. The use of 'MTResNet50' may be considered for further refinements on the architecture as well as the hyperparameters. Overall the MTL models appear to perform better than their STL counterparts, probably due to the higher generalization that comes from learning multiple correlated tasks at the same time.

Given the findings from the STL and MTL experiments, the DenseNet121 was chosen as base architecture for the experiments with the Concept-Bottleneck Models (CBM) configurations (Sec. 4.2.3). The Joint configuration is the best for accuracy = 82.5%±4.2, F-measure = 65.9%±10.6, Precision = 83.0%±4.6, and Recall 55.2%±13.2 for the diagnosis task. This is also true for the mean of seven patterns as can been seen in Tab. 4.8. Finally, between MTL and CBM we can say that for the diagnosis task the accuracy is higher in CBM. The difference in performance with the best MTL models is not excessive, but CBM models have the benefint of learning the hierarchy present between the seven-point patterns and the diagnosis task. The experiments in the CBM also show a better performance than the STL models, probably due to the same generalization benefits coming from MTL. Further comparison has been done among the implemented models and the clinical ground truth. This is obtained by applying the 7-point checklist rule to the true labels of the attributes in the test. This shows the efficacy of the rule when the attributes are correctly identified; the metrics are reported in Tab. 4.9. This rule is highly efficient in identifying the cases of melanoma, as shown by the high values of recall (over 90%) for both threshold values. The models proposed in this manuscript don't yield such a hight recall rate. At the same time the rule suffers from a high false-positive rate, especially when the threshold is low, meaning that many benign lesion will be classified as melanoma.

The implemented models perform better in this regard, being able to identify a higher number of naevi.

Comparing with other approaches is challeging as often different subset of the same dataset or other sources are used in the training. A comparison is somewhat possible with [40], which uses the same split of the Derm-7pt dataset. In this work they explored using different input modalities as training data for their system (i.e. clinical and dermoscopic images and metadata) both in a single- and multi-modal input configurations. The work of [40] employs a mini-batch sampling strategy equivalent to the one used in this work. Additionally, they weight each sample based on the frequency of the sample's labels in the mini-batch. This weighting strategy is not implemented in this work, but it is a possibile future employ. Furthermore they use more samples, because in this work only NEV and MEL labels have been selected for diagnosis task. For this reason the diagnosis task of [40] is a 5-label task while the one proposed in this research is a binary task. Two others significant difference involve loss function and multi-modal input of [40]. The comparison will be made with the accuracy obtained in [40] during training with only dermoscopic images, and in the following referred as x_{d-Kaw}, and with training done combining all possible inputs (clinical, dermoscopic and meta-data images), in the following referred as $\mathrm{x}_{combine-Kaw}$. The 'MTDenseNet' achieved an average accuracy on the 8 tasks of 72.4% while the 'CBM-join' of 74.3%. Kawahara's x_{d-Kaw} and $\mathrm{x}_{combine-Kaw}$ accuracies achieved performances of 72.5% and 73.7%, respectively. In conclusion the best models obtained in this work are competitive compared to the model of [40].

Chapter 5

Conclusion

Visual inspection of dermoscopic images of skin lesions is a standard between experienced and practitioners dermatologists. The *7-pt checklist* is one of the most used rule-based algorithms for the diagnosis of skin cancer. This technique consists in identifying seven clinically significant patterns in the dermoscopic image. Each pattern is assigned a score and the diagnosis of cancer is made if the sum across all patterns is greater than the chosen threshold. The problem is that, human inspection of the skin is influenced by different factors such as experience, attention, stress, or fatigue. Thus, over the last years research has been conducted to develop CAD systems in order to support the physicians' decisions. However, against the prominent performances already reached by CADs proposed in the literature, they are encountering barriers in their diffusion in the real-world. The main reason is related to *black-box* nature of deep learning systems, that causes a lack of trustability by users. From this point the goal of this research study arises. This work focuses on developing a model for melanoma diagnosis based on the *Concept-Bottleneck Framework*. The framework proposes a re-organization of standard the deep learning model into two components: one that is trained from raw images to predict a set of human-understandable concepts; and a second component trained on these concepts, to predict the final target (i.e. the diagnosis). Additionaly, standard models of single-task learning and multi-task learning have been developed to make comparisons. The first has been designed to predict only the diagnosis task, while the latter computed 8 tasks simultaneously: the diagnosis and the 7-pt checklist patterns. The experiments have been carried out on a publicly released dataset derm7pt consisting of 1011 images labeled by experts for the seven-point patterns as well as the diagnosis. Overall, in total 10 experiments have been reported in this manuscript, divided in the three main frameworks of Single-Task Learning (STL), Multi-Task Learning (MTL) and Concept-Bottleneck Models (CBM). Three popular architectures for image classification have been tested as viable bases for our models, with the DenseNet121 emerging as best performing base architecture. Overall, experiments have yielded satisfying results. The best model appears to be the CBM trained using the Joint configuration, which shows better performance

across the eight inspected tasks. In addition to a better performance, this model also addresses the problem of a better explainability of the prediction of the model. This suggests that further studies could be conducted in this direction for the design of CBM architectures that could achieve higher performance while providing additional details to the final diagnosis, and incorporating clinically relevant hierarchies. This could facilitate the spreading of CADs systems in the health sector with a view to a future in which machines and humans could work in a collaborative ecosystem.

Chapter 6

Appendix

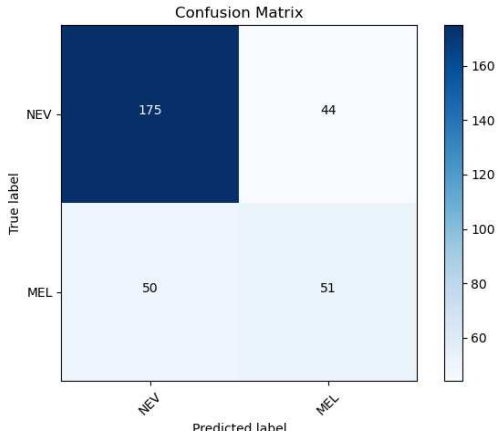
6.1 Appendix Results

In this section additional metrics, training curves or confusion matrices of the most relevant experiments have been reported.

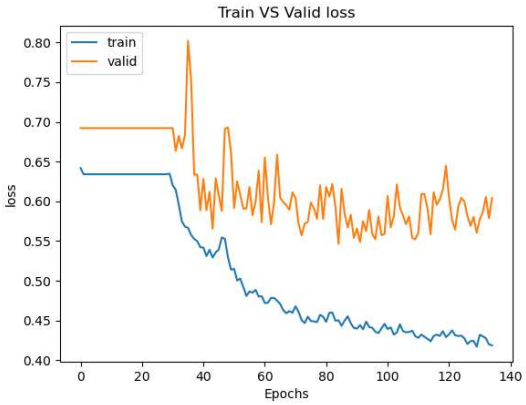
6.1.1 Appendix Single-task learning

Comparison between balancing on 21 and NEV-MEL unique labels

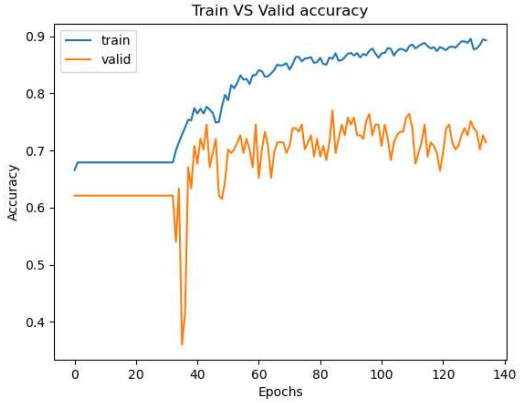
Figure 6.1 shows results with balancing on 21 labels, while 6.2 represents the balancing on NEV-MEL labels. The confusion matrices show that for balancing on 21 labels 6.1a, the model misclassifies more than the balancing on 2 labels 6.2a. The training curves display that the balancing on 21 labels return a singular behaviour during the training. Furthermore the confusion matrices show a better classification performance whether the model is trained with the balancing on 'NEV' and 'MEL'. This confirm that the latter balancing method is better than the first.



(a) *Confusion matrix on 21 labels*

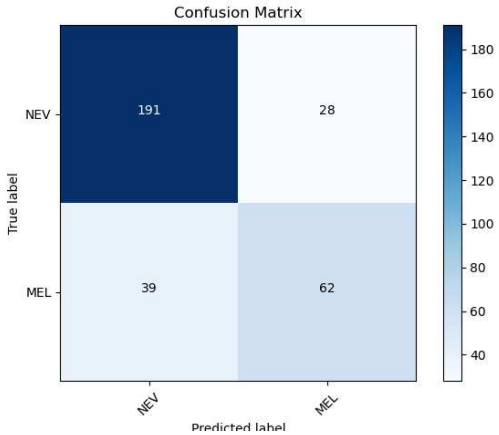


(b) *Loss on 21 labels*

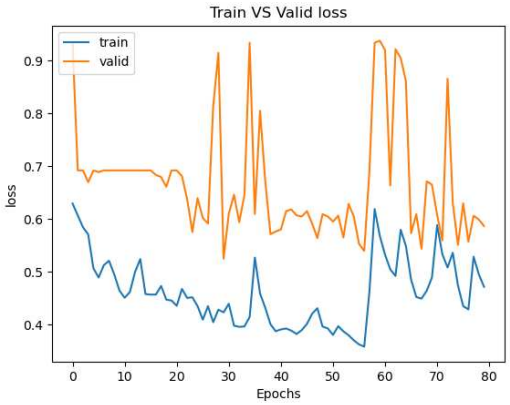


(c) *Accuracy on 21 labels*

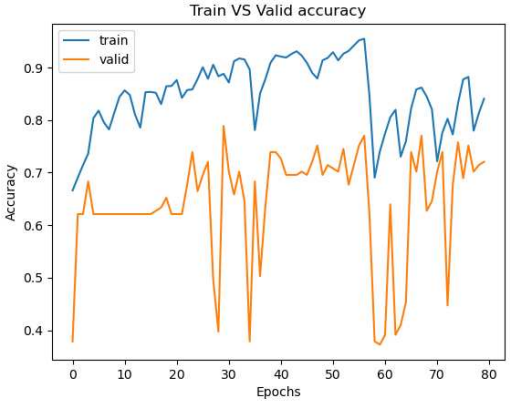
Figure 6.1: STL: comparison between confusion matrix and training curves of balancing on 21 and NEV-MEL unique labels. These are metrics related to 21 labels balancing.



(a) *Confusion matrix on NEV-MEL labels*



(b) *Loss on NEV-MEL labels*



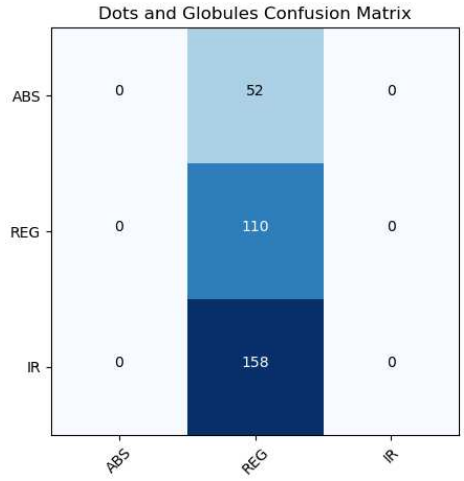
(c) *Accuracy on NEV-MEL labels*

Figure 6.2: STL: comparison between confusion matrix and training curves of balancing on 21 and NEV-MEL unique labels. These are metrics related to 2 labels balancing.

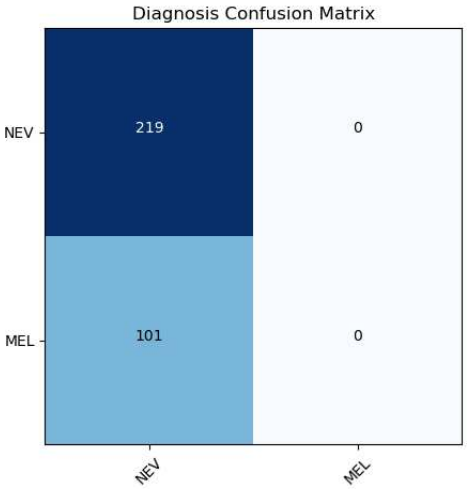
6.1.2 Appendix Multi-task learning

Confusion matrices and loss curve with learning rate 10^{-3}

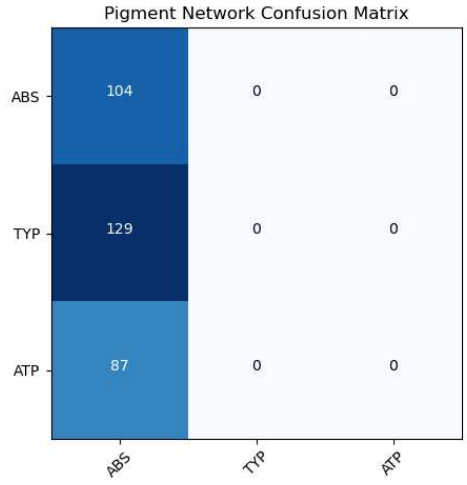
The following are the plots of training curves for *lista pattern*. This phenomena appear only for some tasks.



(a) *Dots and Globules*

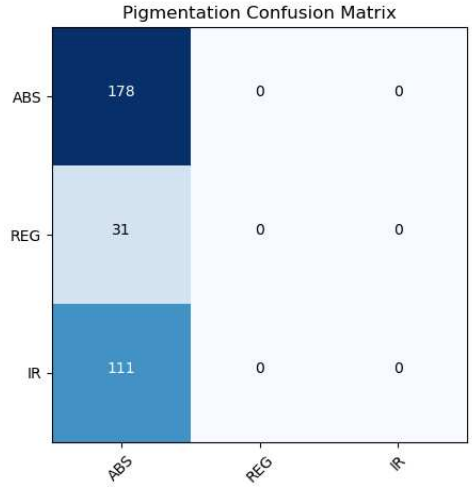


(b) *Diagnosis*

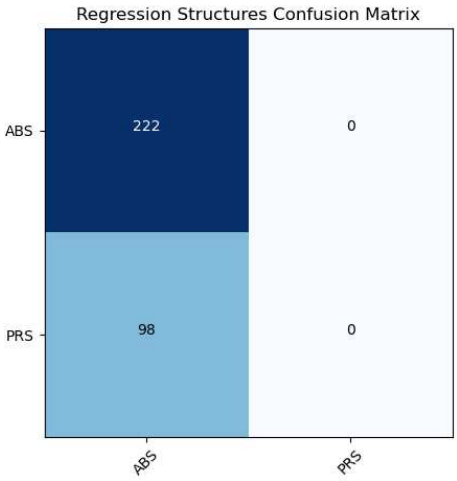


(c) *Pigment Network*

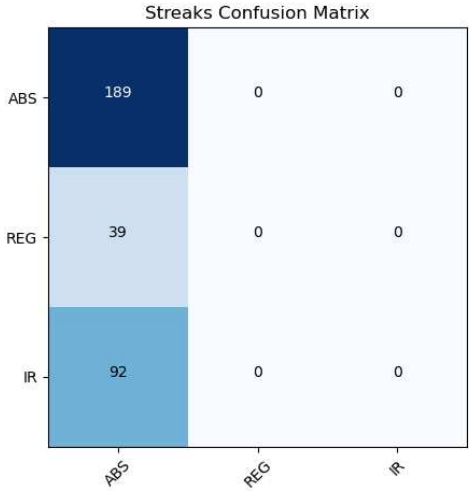
Figure 6.3: MTDenseNet121: confusion matrices of Diagnosis, Dots and Globules, and Pigment Network with learning rate equal to 10^{-3}



(a) *Pigmentation*

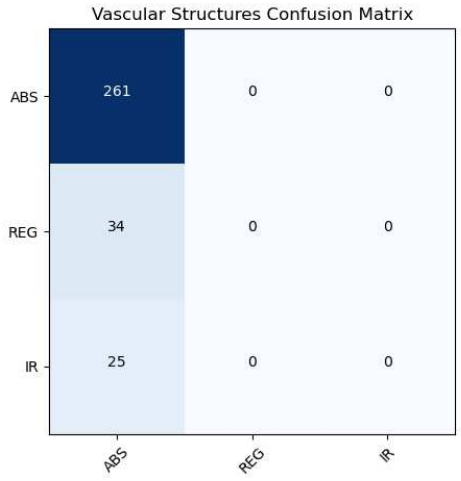


(b) *Regression Structures*

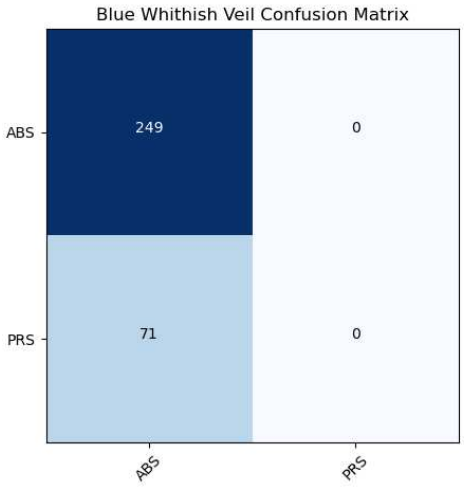


(c)

Figure 6.4: MTDenseNet121: confusion matrices of Pigmentation, and Regression Structures with learning rate equal to 10^{-3}



(a) *Vascular Structures*



(b) *Blue Whitsh Veil*

Figure 6.5: MTDenseNet121: confusion matrices of Vascular Structures, and Blue Whitsh Veil with learning rate equal to 10^{-3}

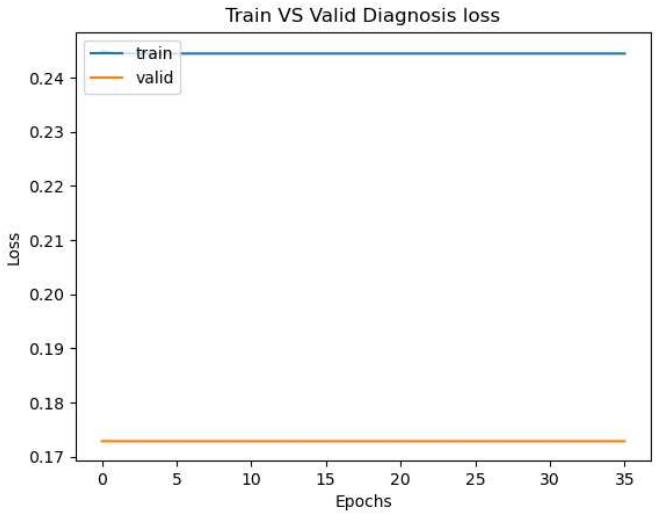


Figure 6.6: MTL DenseNet121: total loss function with learning rate equal to 10^{-3}

Bibliography

[1] Giuseppe Argenziano, H. Peter Soyer, Sergio Chimenti, Renato Talamini, Rosamaria Corona, Francesco Sera, Michael Binder, Lorenzo Cerroni, Gaetano De Rosa, Gerardo Ferrara, Rainer Hofmann-Wellenhof, Michael Landthaler, Scott W. Menzies, Hubert Pehamberger, Domenico Piccolo, Harold S. Rabinovitz, Roman Schiffner, Stefania Staibano, Wilhelm Stolz, Igor Bartenjev, Andreas Blum, Ralph Braun, Horacio Cabo, Paolo Carli, Vincenzo De Giorgi, Matthew G. Fleming, James M. Grichnik, Caron M. Grin, Allan C. Halpern, Robert Johr, Brian Katz, Robert O. Kenet, Harald Kittler, Jürgen Kreusch, Josep Malvehy, Giampiero Mazzocchetti, Margaret Oliviero, Fezal Özdemir, Ketty Peris, Roberto Perotti, Ana Perusquia, Maria Antonietta Pizzichetta, Susana Puig, Babar Rao, Pietro Rubegni, Toshiaki Saida, Massimiliano Scalvenzi, Stefania Seidenari, Ignazio Stanganelli, Masaru Tanaka, Karin Westerhoff, Ingrid H. Wolf, Otto Braun-Falco, Helmut Kerl, Takeji Nishikawa, Klaus Wolff, and Alfred W. Kopf. Dermoscopy of pigmented skin lesions: Results of a consensus meeting via the internet. *Journal of the American Academy of Dermatology*, 48:679–93, 2003.

[2] A Single Neuron | Kaggle.

[3] Hang Yu, Laurence T. Yang, Qingchen Zhang, David Armstrong, and M. Jamal Deen. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing*, 444:92–110, 2021.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Technical report, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 770–778. IEEE Computer Society, dec 2016.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. Technical report, 2015.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:2261–2269, aug 2016.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Édouard Duchesnay, and Matthieu Perrot. Softmax function - Wikipedia, 2011.

[9] Manu Goyal, Thomas Knackstedt, Shaofeng Yan, and Saeed Hassanpour. Artificial Intelligence-Based Image Classification for Diagnosis of Skin Cancer: Challenges and Opportunities. *Computers in Biology and Medicine*, 127, nov 2019.

[10] Davide Coppola, Hwee Kuan Lee, and Cuntai Guan. Interpreting mechanisms of prediction for skin cancer diagnosis using multi-task learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020-June:3162–3171, 2020.

[11] Transfer of learning - Wikipedia.

[12] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

[13] Davide Coppola. Biomedical image processing using machine/deep learning.

[14] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. Technical report, nov 2020.

[15] Gale Encyclopedia of Medicine. Definition of Skin Lesions by Medical dictionary.

[16] Inc. Farlex. Definition of Dermatoscope by Segen's Medical Dictionary.

[17] American Cancer Society. Skin Cancer | Skin Cancer Facts | Common Skin Cancer Types.

[18] Julia Benedetti. Evaluation of the Dermatologic Patient - Dermatologic Disorders - Merck Manuals Professional Edition, feb 2019.

[19] Ralph Peter Braun, Harold S. Rabinovitz, Margaret Oliviero, Alfred W. Kopf, and Jean Hilaire Saurat. Dermoscopy of pigmented skin lesions. *Journal of the American Academy of Dermatology*, 52(1), 2005.

[20] Aimilios Lallas and Giuseppe Argenziano. Research in Dermoscopy: The Best Is Yet to Come! *Dermatology Practical & Conceptual*, jan 2021.

[21] Giuseppe Argenziano, Gabriella Fabbrocini, Paolo Carli, Vincenzo De Giorgi, Elena Sammarco, and Mario Delfino. Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: Comparison of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis. *Archives of Dermatology*, 134(12):1563–1570, 1998.

[22] G. Argenziano, C. Catricalà, M. Ardigo, P. Buccini, P. De Simone, L. Eibenschutz, A. Ferrari, G. Mariani, V. Silipo, I. Sperduti, and I. Zalaudek. Seven-point checklist of dermoscopy revisited. *British Journal of Dermatology*, 164(4):785–790, apr 2011.

[23] Pattern recognition - Wikipedia.

[24] W.R. Howard. Pattern Recognition and Machine Learning. *Kybernetes*, 36(2):275–275, feb 2007.

[25] Information Extraction Sequence Labeling. Technical report.

[26] Ian Chiswell. *Mathematical logic*.

[27] Adam Marblestone, Greg Wayne, and Konrad Kording. Towards an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, 10(SEP), jun 2016.

[28] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[29] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards Biologically Plausible Deep Learning. feb 2015.

[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[31] 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.

[32] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, and Uc San Diego. Aggregated Residual Transformations for Deep Neural Networks. Technical report, 2016.

[33] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:2261–2269, aug 2016.

[34] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep Networks with Stochastic Depth. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS:646–661, mar 2016.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Rethinking the Inception Architecture for Computer Vision. Technical report, 2015.

[36] Michael Crawshaw. MULTI-TASK LEARNING WITH DEEP NEURAL NETWORKS: A SURVEY. Technical report, 2020.

[37] Ram Charan Mishra, Rahul Mishra, and Kusum Sharma. Review of Skin Diseases Classification Using Machine Learning. Technical Report 2, 2021.

[38] Nazia Hameed, Antesar M. Shabut, and M. A. Hossain. Multi-Class Skin Diseases Classification Using Deep Convolutional Neural Network and Support Vector Machine. In *International Conference on Software, Knowledge Information, Information Management and Applications, SKIMA*, volume 2018-Decem. Institute of Electrical and Electronics Engineers Inc., jan 2018.

[39] Jainesh Rathod, Vishal Wazhmode, Aniruddh Sodha, and Praseniit Bhavathankar. Diagnosis of skin diseases using Convolutional Neural Networks. In *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, pages 1048–1051. Institute of Electrical and Electronics Engineers Inc., sep 2018.

[40] Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546, 2018.

[41] Ellák Somfai, Benjámin Baffy, Kristian Fenech, Changlu Guo, Rita Hosszú, Dorina Korózs, Fabrizio Nunnari, Marcell Pólik, Daniel Sonntag, Attila Ulbert, and András Lőrincz. Minimizing false negative rate in melanoma detection and providing insight into the causes of classification. feb 2021.

[42] Haoliang Li, Yufei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex C Kot. Domain Generalization for Medical Imaging Classification with Linear-Dependency Regularization. Technical report, 2020.

[43] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. *arXiv*, 2020.

[44] Tariq Bdair, Nassir Navab, and Shadi Albarqouni. Peer Learning for Skin Lesion Classification. mar 2021.

[45] Giuseppe Argenziano, HP Soyer, V De Giorgi, Domenico Piccolo, Paolo Carli, and Mario Delfino. *Interactive atlas of dermoscopy*. EDRA Medical Publishing & New Media, 2020.

[46] Alexis Cook. Data Leakage | Kaggle.

[47] C.E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):519–520, 1948.

[48] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, aug 2017.

[49] Jeremy; Dan Ventura; Sean Warnick West. Spring Research Presentation: A Theoretical Foundation for Inductive Transfer. 2007.

[50] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Research. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Technical report.