



**Ministry of Higher Education and Scientific  
Research**

**Abbes Laghrou University of Khenchela**

**Faculty of Science and Technology**

**Mathematics and Computer Science Department**



# **Master's Thesis**

*Presented to obtain a Master's degree in Computer Science  
(L.M.D)*

*By:*

***MESSAOUDIA Mohamed Islam***

***LAHOUEL Abdennour***

**Domain: Security and Web Technology (STW)**

---

## **Image classification using Convolutional Neural Network**

---

*Examined on June 2022 by the jury:*

- ***Dr. HIOUAL Ouassila (President)***
- ***Dr. DJELLAB Issam (Supervisor)***
- ***Dr. AZIZI Nabil (Examiner)***

*University year: 2021/2022*

# Dedication

“

*This work is wholeheartedly dedicated to my beloved parents, who have been my source of inspiration and gave me strength when i thought of giving up, who continually provide their moral, spiritual, emotional, and financial support. ,*

*To my beloved brothers, sister, relatives, friends, and classmates who shared their words of advice and encouragement to finish this study. ,*

*To all those who believed in me ,*

*Thank you.*

”

**- Islam**

# Dedication

“

*This work is wholeheartedly dedicated to my beloved parents, who have been my source of inspiration and gave us strength when i thought of giving up, who continually provide their moral, spiritual, emotional, and financial support. ,*

*To my beloved brothers, sister, relatives, friends, and classmates who shared their words of advice and encouragement to finish this study. ,*

*To all those who believed in me ,*

*Thank you.*

”

**- Abdennour**

# Acknowledgements

Above all, we thank Allah, the almighty for having given us the strength and patience to undertake and complete this work.

First and foremost, our heartfelt gratitude to our supervisor **Dr.DJELLAB Issam** for the patience and care throughout the whole research process, he was a vast source of insightful guidance and advice, we would forever be thankful for his time and words of encouragement..

Our greatest thanks, also, go to the board of examiners for their acceptance to evaluate our work and for any valuable comments they would make to refine it..

We're very much thankful to our family for their emotional and financial support, for their love, friendship, understanding, prayers and continuous support to complete this research work.

We similarly express our thanks to our friends and colleagues, for their support and valuable memories..

Finally , thanks to all who contributed in a way or another to the achievement of this work.

# Abstract

Image classification is a challenging task in computer vision, great progress has been achieved in recent years due to the application of methods of deep learning, especially the Convolutional neural network (CNN). In this work we suggest four convolutional neural network models trained using the intel image classification challenge dataset, this dataset consists of natural scene images from over the world it contains around 25 000 images distributed under six categories, there are around 14 000 images in the train set, 3 000 in the test set and 7 000 in prediction set. Our four models differ in the number of hidden layers (10 to 18), we took this approach seeking to achieve the best results, and we have applied some techniques to improve the performance of our convolutional neural network models as data augmentation, dropout, and batch normalization. Our solution produces good results in terms of accuracy and performance.

---

**Mots clés :** Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Deep Learning (DL), Image classification, Machine Learning (ML).

---

# Résumé

La classification des images est une tâche difficile dans la vision par ordinateur, de grands progrès ont été réalisés ces dernières années en raison de l'application des méthodes d'apprentissage profond, en particulier le réseau de neurones convolutifs. Dans ce travail, nous proposons quatre modèles de réseau de neurones convolutifs entraînés à l'aide de base de données intel image classification challenge, cette base de données se compose des images de scènes naturelles du monde entier, elle contient environ 25 000 images réparties en six catégories, il y a environ 14 000 images dans l'ensemble d'entraînement, 3 000 dans l'ensemble de test et 7 000 dans l'ensemble de prédiction. Nos quatre modèles diffèrent par le nombre de couches cachées (10 à 18), nous avons adopté cette approche à la recherche d'obtenir les meilleurs résultats, en outre, nous avons appliqué certaines techniques pour améliorer les performances de nos modèles de réseau de neurones convolutifs comme l'augmentation des données, dropout et batch normalisation. Notre solution produit de bons résultats en termes de précision et performance.

---

**Mots clés :** Apprentissage automatique, Deep Learning, Réseau de neurones convolutifs, CNN, ConvNet, Classification des Images, Réseau de neurones artificiels.

---

## المخلص

يعد تصنيف الصور مهمة صعبة في مجال رؤية حاسوبية، وقد تم تحقيق تقدم كبير في السنوات الأخيرة في هذا المجال بسبب تطبيق طرق التعلم العميق، وخاصة الشبكة العصبونية الإلتقافية. في هذا العمل نقترح أربعة نماذج شبكات عصبونية الإلتقافية مدربة باستخدام مجموعة بيانات تحدي تصنيف صور لأنتل، تتكون مجموعة البيانات هذه من صور مشاهد طبيعية من جميع أنحاء العالم تحتوي على حوالي 25000 صورة موزعة على ست فئات، هناك حوالي 14000 صورة في مجموعة التدريب، 3000 في مجموعة الاختبار و 7000 في مجموعة التنبؤ. تختلف نماذجنا الأربعة في عدد الطبقات المخفية (10 إلى 18)، وقد اتخذنا هذا النهج سعياً لتحقيق أفضل النتائج، وقمنا بتطبيق بعض التقنيات لتحسين أداء نماذج الشبكات العصبونية الإلتقافية لدينا كزيادة البيانات والتخلي و تطبيع الدفعة. توصلنا الى نتائج جيدة من حيث الدقة والأداء.

**الكلمات المفتاحية:** تصنيف الصور, التعلم الآلي, التعلم العميق , الشبكة العصبونية الإلتقافية

# Contents

- Dedication** **I**
  
- Acknowledgements** **III**
  
- Abstract** **IV**
  
- Contents** **VII**
  
- List of Figures** **XI**
  
- List of Tables** **XIII**
  
- List of Abbreviations** **XIV**
  
- 1 Machine learning and deep learning** **3**
  - 1 Introduction . . . . . 3
  - 2 Artificial intelligence . . . . . 3
  - 3 Machine learning . . . . . 4
    - 3.1 Definition of machine learning . . . . . 4
    - 3.2 Types of machine learning . . . . . 4
      - 3.2.1 Supervised Machine Learning . . . . . 5
      - 3.2.2 Unsupervised Machine Learning: . . . . . 5
      - 3.2.3 Reinforcement learning: . . . . . 6
      - 3.2.4 Semi-Supervised Machine Learning: . . . . . 6
    - 3.3 Classification technique . . . . . 6
      - 3.3.1 Perceptron based techniques . . . . . 6
      - 3.3.2 Logic based algorithms . . . . . 7
      - 3.3.3 Statistical learning algorithms . . . . . 7

3.3.4	Instance-based learning . . . . .	8
3.3.5	Support Vector Machines: . . . . .	8
4	Deep learning: . . . . .	8
4.1	Definition of deep learning . . . . .	8
4.2	Difference between Machine . . . . .	8
4.3	Biological Neuron . . . . .	9
4.4	Artificial Neurons . . . . .	9
4.5	Artificial Neural Network: . . . . .	10
4.6	Learning in artificial neural network . . . . .	10
4.7	Activation functions: . . . . .	10
4.7.1	Rectified Linear Unit function . . . . .	11
4.7.2	Sigmoid function . . . . .	11
4.7.3	Hyperbolic Tangent function . . . . .	12
4.7.4	Softmax function . . . . .	12
4.8	Loss functions . . . . .	13
4.8.1	Perceptron loss function . . . . .	13
4.8.2	Logarithmic loss function . . . . .	14
4.8.3	Binary cross entropy loss . . . . .	14
4.8.4	Categorical cross entropy . . . . .	14
4.9	Optimizers . . . . .	14
4.9.1	Batch Gradient Descent . . . . .	15
4.9.2	Stochastic Gradient Decent . . . . .	15
4.9.3	Adam . . . . .	15
4.9.4	RMSProp . . . . .	15
5	Different types of Neural Networks . . . . .	15
5.1	Feedforward Neural Network . . . . .	15
5.2	Recurrent Neural Networks . . . . .	16
5.3	Radial basis function Neural Network . . . . .	16
5.4	Kohonen Self Organizing Neural Network . . . . .	17
5.5	Convolutional Neural Networks . . . . .	18
6	Conclusion . . . . .	18

<b>2</b>	<b>Convolutional Neural Networks for Image Classification</b>	<b>20</b>
1	Introduction . . . . .	20
2	Computer vision . . . . .	20
3	Image classification . . . . .	21
3.1	Definition of image . . . . .	21
3.2	Image types . . . . .	21
3.2.1	Binary image . . . . .	21
3.2.2	Grayscale image . . . . .	21
3.2.3	Rgb image . . . . .	21
3.3	Characteristic of image . . . . .	22
3.3.1	Pixel . . . . .	22
3.3.2	Image size . . . . .	22
3.3.3	Image dimensions . . . . .	22
3.3.4	Brightness . . . . .	22
4	Convolutional neural networks . . . . .	22
5	Convolutional Neural Network Layers . . . . .	23
5.1	Convolutional Layer . . . . .	23
5.1.1	Padding . . . . .	23
5.1.2	Kernel size . . . . .	24
5.1.3	Stride . . . . .	25
5.2	Pooling Layer . . . . .	26
5.2.1	Max pooling . . . . .	26
5.2.2	Average pooling: . . . . .	26
5.3	Fully Connected Layer . . . . .	26
6	Popular CNN architecture . . . . .	27
6.1	AlexNet . . . . .	27
6.2	LeNet . . . . .	28
6.3	VGGNet . . . . .	28
7	Training Considerations: Batches, Epochs . . . . .	29
7.1	Batch-size . . . . .	29
7.2	Epochs . . . . .	29
8	Regularization . . . . .	29

## Contents

---

8.1	Data augmentation . . . . .	29
8.2	Dropout . . . . .	30
8.3	Batch normalization . . . . .	30
8.4	Early stopping . . . . .	31
9	Transfer Learning . . . . .	32
10	Conclusion . . . . .	33
<b>3</b>	<b>Implementation</b>	<b>34</b>
1	Introduction . . . . .	34
2	Software and libraries used in the implementation: . . . . .	34
2.1	Python . . . . .	34
2.1.1	Tensorflow . . . . .	35
2.1.2	Keras . . . . .	35
2.1.3	Matplotlib . . . . .	35
2.2	Google Colab Pro . . . . .	36
3	Configuration used in the implementation . . . . .	36
4	Dataset . . . . .	37
5	Network architecture . . . . .	38
6	Results and discussion . . . . .	44
7	Conclusion . . . . .	50
	<b>Bibliography</b>	<b>52</b>

# List of Figures

1.1	Relation between artificial intelligence,machine learning and deep learning . . .	4
1.2	Workflow of machine learning [1] . . . . .	4
1.3	Machine learning types . . . . .	5
1.4	The structure of a Biological Neuron . . . . .	9
1.5	A hypothetical example of Multilayer Perceptron Network. . . . .	10
1.6	Rectified Linear Unit . . . . .	11
1.7	Sigmoid function . . . . .	12
1.8	Hyperbolic Tangent function . . . . .	12
1.9	Softmax function . . . . .	13
1.10	Feedforward neural network with one hidden layer and multiple neurons at the output layer. . . . .	16
1.11	Architecture of RNN with single hidden layer [2] . . . . .	17
1.12	A Radial Basis Function Network. . . . .	17
1.13	Architecture of a SOM network . . . . .	18
2.1	General CNN architecture . . . . .	23
2.2	Examples of the convolution operation with filter (3 X 3 ) and stride of 1 . . . .	24
2.3	Examples of adding a padding to an image . . . . .	25
2.4	Max pooling versus Average pooling . . . . .	27
2.5	AlexNet introduced by Krizhevsky 2014 . . . . .	27
2.6	LeNet introduced by Yan LeCun . . . . .	28
2.7	VGGNet network architecture [3] . . . . .	28
2.8	Example of applying data augmentation . . . . .	30
2.9	Visualisation of dropout technique . . . . .	31
2.10	Example of Early stopping . . . . .	32

## List of Figures

---

3.1	Google Colab interface . . . . .	36
3.2	Example of images of the dataset . . . . .	37
3.3	Example of images of the dataset . . . . .	38
3.4	Model A architecture . . . . .	39
3.5	Model B architecture . . . . .	40
3.6	Model C architecture . . . . .	41
3.7	Model D architecture . . . . .	42
3.8	Visualizing of model A, B and C . . . . .	43
3.9	Visualizing of model D . . . . .	44
3.10	Test accuracy and loss plots for model A without data augmentation . . . . .	44
3.11	Test accuracy and loss plots for model A . . . . .	45
3.12	Test accuracy and loss plots for model B . . . . .	45
3.13	Test accuracy and loss plots for model C . . . . .	46
3.14	Test accuracy and loss plots for model D . . . . .	46
3.15	Confusion matrix for model D . . . . .	47
3.16	Predictions by Model D . . . . .	48
3.17	Bad predictions by Model D on test dataset . . . . .	49

# List of Tables

1.1	Comparison of Machine learning and deep learning . . . . .	9
3.1	Configuration used in the implementation. . . . .	37
3.2	Results Comparison Table . . . . .	47

# List of Abbreviations

<b>AI</b>	<i>Artificial Intelligence</i>
<b>ANN</b>	<i>Artificial Neural Network</i>
<b>CNN,ConvNet</b>	<i>Convolution Neural Network</i>
<b>CUDA</b>	<i>Compute Unified Device Architecture</i>
<b>CV</b>	<i>Computer Vision</i>
<b>DL</b>	<i>Deep Learning</i>
<b>DNN</b>	<i>Deep Neural Network</i>
<b>FC</b>	<i>Fully Connected Layer</i>
<b>KNN</b>	<i>K-Nearest Neighbors</i>
<b>ML</b>	<i>Machine Learning</i>
<b>MLP</b>	<i>Multilayer Perceptron</i>
<b>NB</b>	<i>Naive Bayes classifier</i>
<b>RBFN</b>	<i>Radial Basis Function Network</i>
<b>RELU</b>	<i>Rectified Linear Unit</i>
<b>RGB</b>	<i>Red Green Blue</i>
<b>RL</b>	<i>Reinforcement Learning</i>
<b>RNN</b>	<i>Recurrent Neural Network</i>
<b>SOM</b>	<i>Self-organizing map</i>

## List of Abbreviations

---

**SVM**      *Support Vector Machine*

# Introduction

The significant advancement in technology raises the amount of data volume, which needed efficient ways and techniques for processing data, most of the relevant techniques are powered by artificial intelligence which led to a huge progress in different modern life fields.

Machine Learning is a branch of Artificial intelligence (AI) that aims to teach the computer to perform various tasks without being explicitly programmed. The advances in machine learning especially deep learning have provided us with powerful tools for solving complex problems that seemed to be impossible to solve in the past few decades.[4]

Deep learning is a subfield of machine learning that uses algorithms and techniques based on the theory of statistical learning, artificial neural networks results have improved exponentially in the contemporary era due to advances in computing power and abundance of plethora of data.

Image classification is a challenging computer vision problem that seeks to make machines able of categorizing images to different classes. In this thesis We will explore the domain of image classification using deep learning first, we will focus on Convolutional neural network and its foundations going through its different layers, Afterwards, we will apply different CNN models to natural scene images dataset, and we will experiment with different techniques to improve our models and reach sufficient results.

We have organized our work into 3 chapters :

**In the first chapter** we highlight and introduce the concepts and basics of machine learning, then we talk briefly about Machine Learning (ML) and its different types, after that we will focus our attention on certain Supervised machine learning technique which is classification. In the second part of this chapter, we spotlight Deep learning concept, afterward we discuss artificial neurons, finally, we mention some popular artificial neural network architectures.

**In the second chapter** we take off by talking about Computer vision and the task of image classification, also we define different types of images. In the next section, we deconstruct the concept of convolutional neural network and its architecture, afterward will present some popular CNN architectures, and in the end we will indicate regularization techniques to improve models.

**In the third chapter** we start off by introducing the software, libraries, and hardware used in our models and the dataset used to train the model, then we present four implementations that we built based on our experiments with different architectures from different research papers, in the end, we discuss the results of each model.

# Chapter 1

## Machine learning and deep learning

### 1 Introduction

One of the fields of modern computing is machine learning. There has been a lot of research towards making machines intelligent in the other side learning is a natural behavior of humans, that has been made an integral part of machine learning. Machine learning includes deep learning as a subset. Deep learning has only been used in a few applications so far.[5]

This chapter highlight and introduces the concepts of basics machine learning, we will talk briefly about Machine Learning (ML) and its different types, we will focus on supervised learning technique which is classification. In the second part of this chapter, we spotlight Deep learning concept, afterward we discuss artificial neurons, finally, we mention some popular artificial neural network architectures.

### 2 Artificial intelligence

Artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable[6]. Informally, the term AI is applied when a machine can perform functions that humans associate with other human minds, such as "learning" and "problem solving".[5]

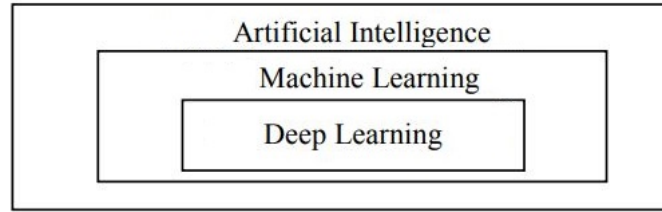


Figure 1.1: Relation between artificial intelligence, machine learning and deep learning

### 3 Machine learning

#### 3.1 Definition of machine learning

Machine learning is the study of designing machines that can learn from data. This is useful for solving a variety of tasks, including computer vision, for which the solution is too difficult for a human software engineer to specify in terms of a fixed piece of software. Moreover, since learning is a critical part of intelligence, studying machine learning can shed light on the principles of intelligence.[7]

Machine learning has progressed dramatically over the past two decades, from laboratory curiosity to a practical technology in widespread commercial use. Within artificial intelligence (AI), machine learning has emerged as the method of choice for developing practical software for computer vision, speech recognition, natural language processing, robot control, and other applications.[8]

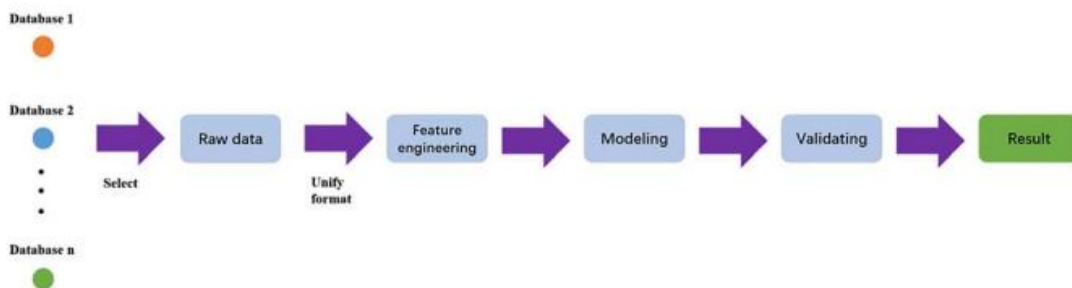


Figure 1.2: Workflow of machine learning [1]

#### 3.2 Types of machine learning

Based on the methods and way of learning, machine learning is divided into mainly four types which are :

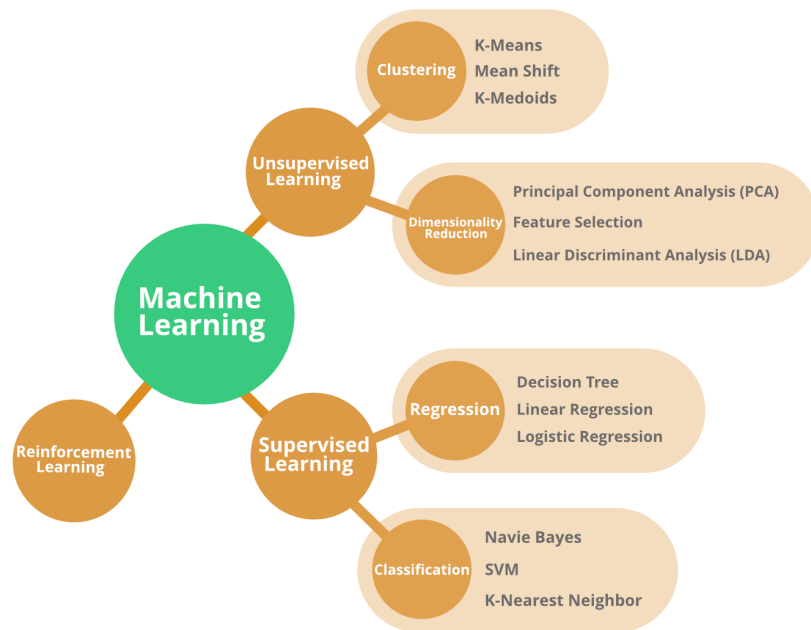


Figure 1.3: Machine learning types

### 3.2.1 Supervised Machine Learning

Supervised learning uses a training set to teach models to yield the desired output. This training data set includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.[9]

Supervised machine learning can be divided into the following two categories of problems:

**Classification** Classification is a process of categorizing a given dataset into classes, It can be performed on both structured and unstructured data. The process starts with predicting the class of given data points. The classes are often referred as target, label, or categories.[10]

**Regression** Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.[11]

### 3.2.2 Unsupervised Machine Learning:

Unsupervised learning is the use of artificial intelligence algorithms to discover patterns in data.[12] Sets containing data points that are neither classified nor labeled to classify its sets

without having any external guidance to performance this task, its ability to detect similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation and pattern recognition .

**Clustering** Clustering techniques are used when we want to find groups from data. It is a method of grouping objects into clusters so that the objects with the most similarity remain in a group and have less or no similarity with objects in other groups, as an example of a clustering algorithm is to group customers according to their purchasing behavior.[13]

**Association** Association learning is a rule-based device mastering and information mining method that could find out crucial relationships between variables or features in a dataset. In contrast to traditional affiliation algorithms that measure similarity, association rules gain knowledge of identifying hidden dependencies in databases with the aid of applying a sure level of hobby to generate affiliation rules for brand new searches.[14]

### 3.2.3 Reinforcement learning:

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path, it should take in a specific situation.[15]

### 3.2.4 Semi-Supervised Machine Learning:

Also known as classification with labeled and unlabeled data or partially labeled data, it is an extension to the supervised classification problem. The training data consists of both labeled instances and unlabeled instances, typically assumes that there is much more unlabeled data than labeled data. The goal of semi-supervised classification is to train a classifier from both labeled and unlabeled data, it does better than the supervised classification for certain tasks.[16]

## 3.3 Classification technique

### 3.3.1 Perceptron based techniques

**Single layer perceptron** The perceptron algorithm is a supervised learning method used to learn binary classifications. A binary classification is a function that can determine that an input is belong to this class or not.[17]

**Multilayer perceptron** A multilayer perceptron is a model used to represent a nonlinear mapping between an input vector and an output vector [18], in these networks the flow of information is based on the communications among three types of layers: input, hidden, and output layers.[19]

### 3.3.2 Logic based algorithms

**Decision tree** A decision tree is a collection of “If and Then” conditional rules for the assignment of class labels to instances of a dataset. Decision trees consist of nodes that specify a particular attribute of the data and branches that represent a test of each attribute value, and leaves that correspond to the terminal decision of class assignment for an instance in the dataset.[20]

**Learning set of rules:** Learning set of rules is a popular methods of classification in machine learning [21]. It used to find regularities in data that can be expressed in the form of an ”If-Then” rule. Depending on the type of rule that should be found, we can discriminate between descriptive and discovery rules and also predictive learning rule.[22]

### 3.3.3 Statistical learning algorithms

**Bayesian Networks** A Bayesian network is a graphical model for probabilistic relationships amongst a set of variables. Over the last decade, the Bayesian community has ended up a popular representation for encoding and understanding structures , researchers have evolved techniques For studying Bayesian networks from the information. The techniques which have been developed are new and nevertheless evolving,and it they had been shown to be remarkably effective for some information-analysis issues.[23]

**Naive Bayes classifiers** The naïve Bayes (NB) classifier is a family of simple probabilistic classifiers based totally on a common assumption that all Features are unbiased of each other, given the class variable.[24]

The specific NB classifiers fluctuate specifically utilizing the assumptions concerning the distribution of features. This assumptions are referred as event models of the NB classifier.[25]

### 3.3.4 Instance-based learning

**k-nearest neighbors** K-nearest neighbor Method, it measures the closest Neighbors at a value of k also define the number of closest neighbors Explore to explain the class of sample data points[26] .One of the major benefits of KNN is an effective method for large training data and also it is robust to noisy training data.[26]

### 3.3.5 Support Vector Machines:

Vapnik proposed a statistical mastering concept for systems based on gaining knowledge as a technique, that's known as Support Vector Machine (SVM).[27]

SVM has been taken into consideration as one of the maximum distinguished and handy approaches for solving issues associated with the classification of records [28].

The primary benefit of SVM is its capability to deal with an extensive variety of class problems includes excessive dimensional and also non linearly separable issues. One of the most disadvantages of SVM which is the wide variety of parameters to be set, for getting a remarkable classification effects.

## 4 Deep learning:

### 4.1 Definition of deep learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction[29]. The computer gathers knowledge from experiences, there is no need for a human computer operator formally to specify all of the knowledge needed by the computer. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones.[30]

### 4.2 Difference between Machine

Machine learning and deep learning are both types of AI, machine learning is AI that can automatically adapt with minimal human interference, while Deep learning is a subset of machine learning that uses artificial neural networks to mimic the learning process of the human brain. In the table 1.1 below we highlight important difference between ML and DL.

Machine learning	Deep learning
A subset of AI	A subset of machine learning
Can train on smaller data sets	Requires large amounts of data
Requires more human intervention to correct and learn	Learns on its own from environment and past mistakes
Shorter training and lower accuracy	Longer training and higher accuracy
Makes simple, linear correlations	Makes non-linear, complex correlations
Can train on a CPU (central processing unit)	Needs a specialized GPU (graphics processing unit) to train

Table 1.1: Comparison of Machine learning and deep learning

### 4.3 Biological Neuron

A neuron is a cell composed of a cell body and a nucleus. The cell body branches to form what are called dendrites. These are sometimes so numerous that we then speak of dendritic hair or dendritic arborization. It is through the dendrites that information is conveyed from the outside to the soma, the body of the neuron. The information processed by the neuron then travels along the (single) axon to be transmitted to the other neurons. The transmission between two neurons is not direct. In fact, there is an intercellular space of a few tens of Angströms (10-9m) between the axon of the afferent neuron and the dendrites (we say a dendrite) of the efferent neuron. The junction between two neurons is called the synapse.[31]

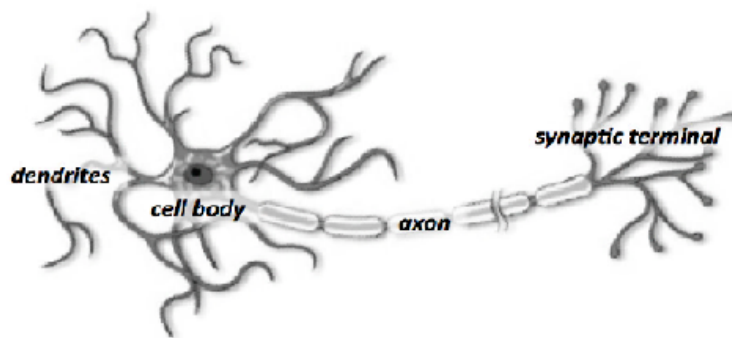


Figure 1.4: The structure of a Biological Neuron

### 4.4 Artificial Neurons

The artificial neuron is the model for the biological neuron. It takes a collection of inputs, multiplies each by weight, then sums all of these weighted plus a constant value known as a Bias. The total weight is given to an activation function, which sends its output to the next neuron as input.[32]

### 4.5 Artificial Neural Network:

The origin of modern neural networks is in Perceptron [33]. which is a neural network unit that processes data to identify features. The function  $F$  takes its input  $X$  and multiplies it by the learned weight coefficient  $w$  to generate an output value.[34]

Subsequent research showed that multilayer perceptrons is capable of preform too much complex functions . In the next section, we will introduce some of the most commonly used activation functions in DL era .[35]

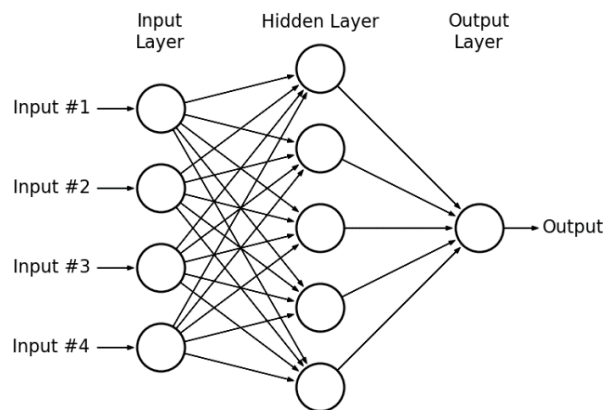


Figure 1.5: A hypothetical example of Multilayer Perceptron Network.

### 4.6 Learning in artificial neural network

Learning in an ANN optimization problem that consists of adjusting the weights  $W_i$  in an iterative process in order to improve model performance. The main goal is to minimize the loss which is presented by a function of the cost  $C$ , this function measures the difference between the predicted classes and the actual classes. During learning, each iteration is composed of two treatments are: forward propagation and back propagation.[36]

### 4.7 Activation functions:

The term activation function is used to refer the function  $F$  that converts the input  $z$  to the next nodes as an output [37]. The purpose of the activation functions is to convert an input signal from a node to an output signal, this conversion makes it possible to introduce nonlinear properties to ANN to solve nonlinear problems and process complex data type , such as images and videos.[36]

Different neurons need different activation functions for the best performance to specific tasks, in the next subsections, we are going to discuss the common popular activation functions on brief and their characteristics and a few of the tasks they're best suited to.

### 4.7.1 Rectified Linear Unit function

Rectified linear unit (ReLU)  $F(z)$  became widely used since it improved the convergence of stochastic Gradient descent in comparison to the sigmoid function, besides its smooth implementation by using Thresholding in the activation map. it's popular used in image processing.[38]

$$F(z) = \max(0, z), F(z) \geq 0, z \in R \quad (1.1)$$

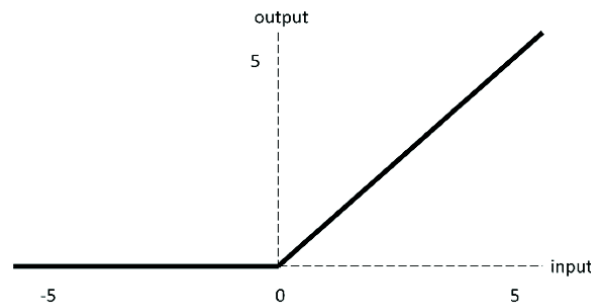


Figure 1.6: Rectified Linear Unit

### 4.7.2 Sigmoid function

Sigmoid activation function translates the input range from  $]-\infty, +\infty[$  to the range in  $[0; 1]$ . It is non-linear by nature and has a smooth derivative. Due to the output range of the Sigmoid  $[0; 1]$  the output of each unit is also squashed causing the gradient to vanish especially in a deep network. This behavior makes the network very hard to optimize and after a certain point. [39]

The Sigmoid function is defined as follows:

$$F(x) = \frac{1}{1 + e^{-x}} \quad (1.2)$$

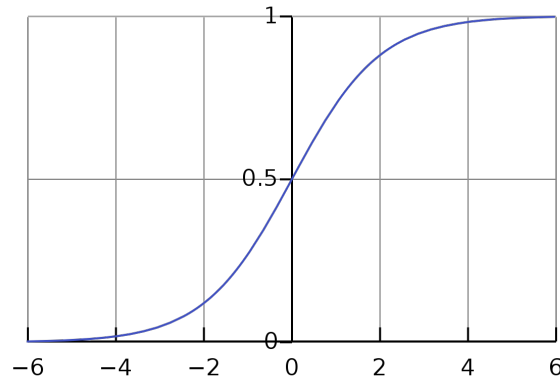


Figure 1.7: Sigmoid function

### 4.7.3 Hyperbolic Tangent function

Tanh is a non-linear function similar to a sigmoid. The difference between them unlike sigmoids in the range  $[0, 1]$  the Tanh function ranges  $[-1, 1]$ , the advantages of Tanh over sigmoid is that: its gradient is stronger and also its derivative is stronger, which makes vanishing gradient problems less likely to occur, for this reason it is widely used in Recurrent neural network.[39]

The Tanh function is defined as follows:

$$F(z) = \frac{2}{1 + e^{-2z}} - 1 \quad (1.3)$$

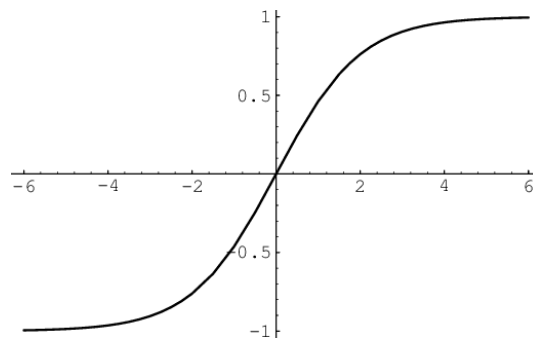


Figure 1.8: Hyperbolic Tangent function

### 4.7.4 Softmax function

The softmax function is a combination of several sigmoids function, as we know the sigmoid function its values ranges in  $[0, 1]$ , it used to predict the probability class of data points. Softmax characteristic are not like sigmoid while the sigmoid communally used for binary classification,

the softmax used for multiclass classification tasks. The softmax is used as an activation function for multi-class classification problems which required more than two class labels.[40]

It may be expressed as:

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (1.4)$$

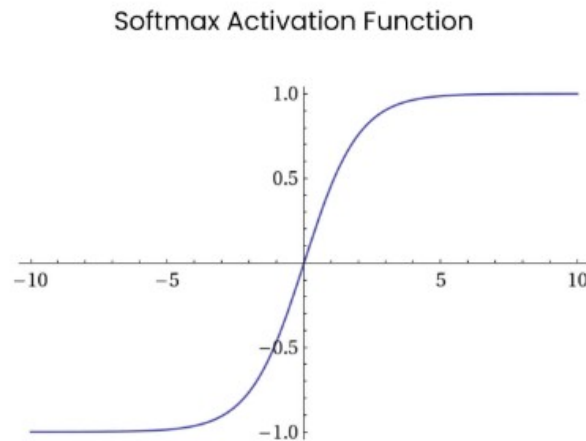


Figure 1.9: Softmax function

When we build a network for multiple class classification we need to have the same number of neurons which is equal to the number of classes.[40]

## 4.8 Loss functions

The loss function is a metric that helps the network to understand if the learning is in the right directions or not, quantifies the difference between expected outcome and outcome produced by the machine learning model. We have several standards loss function defined in deep learning as we will show in the next subsection.[41]

### 4.8.1 Perceptron loss function

Perceptron Loss Function also a partial function, when the samples predicted value has the same sign as the actual label the loss value is 0 ,otherwise the loss value is the absolute value of the predicted value. From the point of view of geometrical sense, if there is no loss means all classified samples are correct, while the method measures the distance from the predicted value and the expected value limit  $f(x) = 0$  and the larger the distance, the error, so the corresponding loss is larger.

Apparently, the idea of the perceptron loss is clear and easy to understand so it is easy to optimize, however its purpose is to determine the exact sample category, therefore if the model obtained bad decisions means it has poor generalization performance, perceptron loss function is not robust against noisy data.[42]

### 4.8.2 Logarithmic loss function

Logarithmic Loss Function is a function that predict sample probability value, where the predicted probability value is obtained through a conditional probability distribution. In particular, the greater probability means that the sample is well predicted to its label, which also means a smaller the corresponding loss value is ; on the contrary, the higher loss value means the lower probability is that lead to bad predictions to the desired class. The probability define as predicted is a positive class expressed by  $P$ , while the other probability is  $1 - P$ , and increases rapidly with decreasing probability of correct prediction. This trend will make the prediction output closer to the label, which is beneficial for the convergence of the algorithm.[42]

### 4.8.3 Binary cross entropy loss

Binary cross entropy loss function used on binary problems 0 or 1 to make decisions, it measures how far away the true value to the prediction so which of the classes and it belongs. The last neurons must have a Sigmoid as activation function to perform a binary classification.[43]

### 4.8.4 Categorical cross entropy

Categorical cross entropy is a loss function that used for multi classification. Precisely in multi classification problems where a sample can belong to multiple classes at the same time, the model tries to decide for each class whether the sample is belongs it.[44]

## 4.9 Optimizers

The optimization algorithm chosen by the deep learning expert determines the learning rate and the final prediction performance of the model. To date, there is no theory which fully explains how to make the selection of optimizers.[45]

### 4.9.1 Batch Gradient Descent

This method calculates the loss of the entire dataset. The batch gradient descent method tends to converge slowly, because the gradient is calculated for the whole dataset after that to do the update. It cannot be used with a large data set because of memory limitation. [46]

### 4.9.2 Stochastic Gradient Descent

SGD is much faster than batch gradient, regular updates often cause extreme volatility by moving towards the minimum. These fluctuations reduce the probability of getting stuck at a local minimum and allow the algorithm to switch to the new algorithm and the local minimum is likely better. However, it has been proven that when we reduce the learning rate SGD converges somewhat favorably.[46]

### 4.9.3 Adam

Adaptive Moment estimation is another method of calculating adaptive learning rates for each parameter. In addition to storing the exponentially decreasing average of past square gradients like RMSprop, Adam also keeps an exponential descending average of past gradients. [46]

### 4.9.4 RMSProp

RMSProp is a variant of the optimizer Rprop which is suitable for mini batch learning. This method is considered to be a combination of the methods Rprop and SGD and known by its similarity to the AdaDelta optimizer. The main part of this strategy is to solve the problem of degradation of the learning rate. RMSProp divides the learning rate by the exponentially moving average of the square and a fixed gradient.[47][48]

## 5 Different types of Neural Networks

### 5.1 Feedforward Neural Network

This neural network is a simple form of ANN, in which the data travels in one direction, the data travels through the input nodes and exits on the output nodes, this neural network might or might not have a hidden layer, the wave consists of forward propagation and back propagation also it uses an activation function.

Feedforward neural networks are often used in computer vision and speech recognition where it is difficult to determine what classes the target data belongs to. These types of Neural Networks are able to handle noisy data well and are easy to maintain.[49]

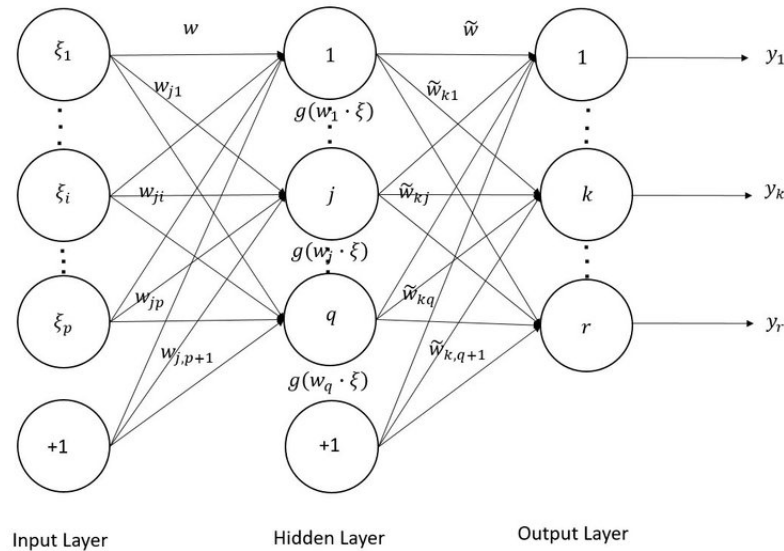


Figure 1.10: Feedforward neural network with one hidden layer and multiple neurons at the output layer.

## 5.2 Recurrent Neural Networks

A recurrent neural network is a feedback neural network enhanced by consists of edges spanning contiguous time steps, introducing the concept of time into the model. RNNs can have no period between regular edges, however the edges that connect adjacent time steps, are called repeating edges, which can form cycles, including cycles of length one that are self-connections of a node to itself in time.[50]

## 5.3 Radial basis function Neural Network

In its simplest form, an RBF network is a three-layer feedforward neural network. The first layer corresponds to the input of the network, and the second layer is a hidden layer composed of multiple RBFs Non-linear activation units, the last one corresponds to the final output of the network. activation Functions in RBFN are traditionally implemented as Gaussian functions.[51]

Most of the classical methods in the literature for training RBFNs are performed in two steps stage. In the first stage, determine the middle and width, for example using some unattended

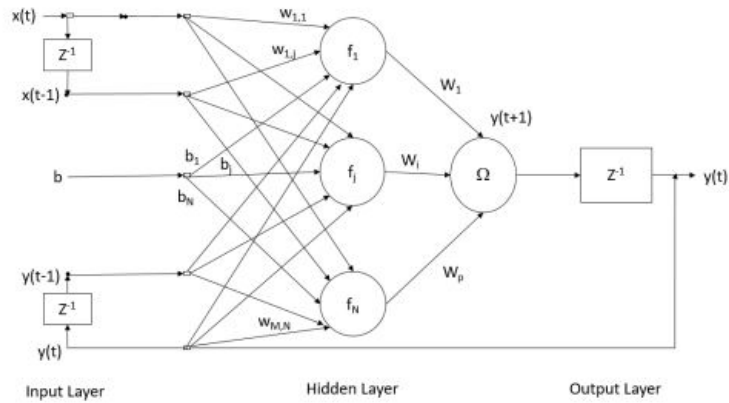


Figure 1.11: Architecture of RNN with single hidden layer [2]

clustering algorithm, while the hidden layer in the second stage is used to find the error criterion for the output layer somehow, like the usual mean squared error (MSE) is minimized over the entire dataset.[51]

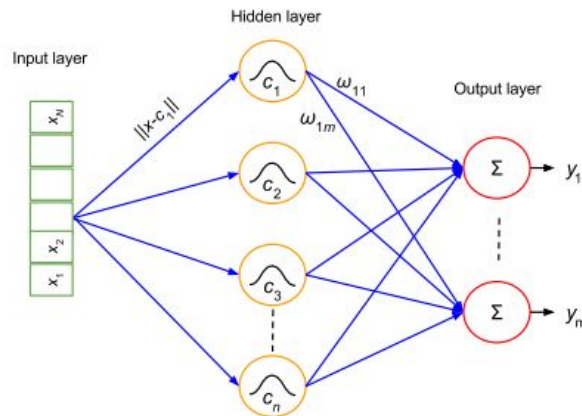


Figure 1.12: A Radial Basis Function Network.

## 5.4 Kohonen Self Organizing Neural Network

Kohonen Self-Organizing Feature Map (SOM) is a self-modifying neural network Respond to input patterns, this property is called self-organization and realizes use competitive learning. Basic competitive learning means competition of process that happens before each learning cycle. the winner of the competition process elements are selected based on certain criteria. Usually the criterion is to minimize a Euclidean distance between the input vector and the weight vec-

tor. After winning select the processing element and adjust its weight vector according to the learning law second hand. SOM is different from basic competitive learning, not just adaptation the winning handling element is also a neighbor of the winning handling element custom made. Self-organizing properties of SOM are based on using neighborhoods winning processing element.[52]

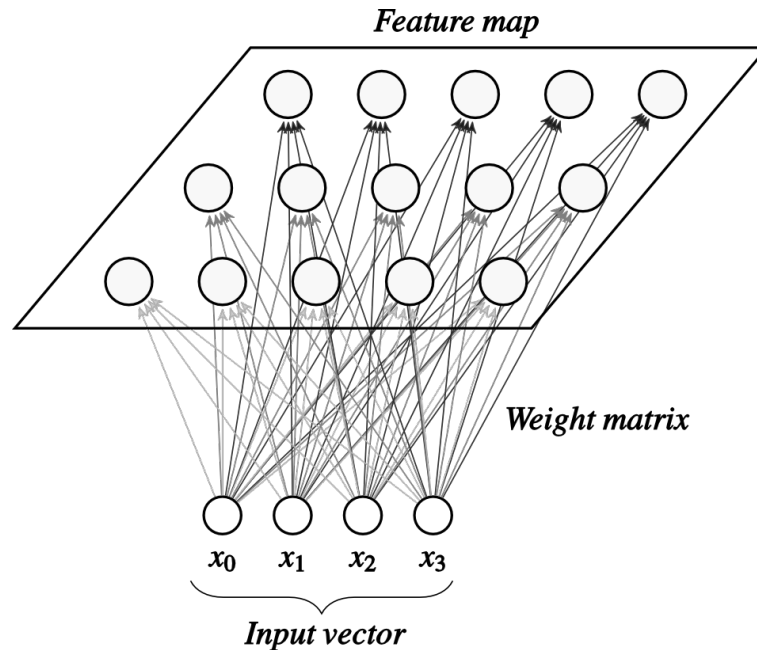


Figure 1.13: Architecture of a SOM network

## 5.5 Convolutional Neural Networks

A convolutional neural network is a particular type of neural network that is based on the convolution operation. Convolutional neural networks are derived from Multi Layer Perceptron (MLP) architectures, however they use shared weights, linked to the convolution window, which allow them to implicitly extract local features. CNNs are particularly well suited for image classification. A network is said to be convolutional when each neuron receives its information not from the whole previous layer, but only from the neurons located in its receptive field.[53]

## 6 Conclusion

In this chapter, we introduced the Machine learning field passing through its different type, we indicate different supervised classification techniques. More over we defined the deep learning

## **Chapter 1. Machine learning and deep learning**

---

concept and it's learning method with different component, also we cited different artificial neural network architectures.

# Chapter 2

## Convolutional Neural Networks for Image Classification

### 1 Introduction

The immense increase in data storage in the past few years has put engineering in challenges to make use of this data, image is one of the most common types of data for the purpose makes the best of it. Deep learning proved its capability in image classification tasks using the most famous neural network CNN in many fields, medical [54], industrial[55] and agriculture[56].

In this chapter we will start by talking about the Computer vision field and then the image classification task with defining images and their types, In the next section we will spotlight convolutional neural networks and their architecture, afterward will show some popular CNN architecture, lastly, we will indicate some of the regularization techniques to build better models.

### 2 Computer vision

AI has proved its capability of solving difficult problems that few years before were so hard to solve for classical programming especially when we come to Computer vision tasks, CV stands for making computers able to gain high-level understanding from digital images or videos. The evolution of computers and the exponentially growth of data pushed deep learning to be one of the best techniques in the computer vision domain like classification, segmentation or object detection.[57]

### 3 Image classification

Classification of images is a challenging problem in computer vision. The computer vision task of localization, detection, and segmentation is based on the foundation of understanding how objects are represented in digital space. Although Tasks can be considered second nature to humans, it's more challenging for automated systems[58] Some of the problems that can occur include the variability of object viewpoints and the high variability of having many object types.[59]

#### 3.1 Definition of image

An image may be defined as a two-dimensional function,  $f(x,y)$ , where  $x$  and  $y$  are spatial (plane) coordinates, and also the magnitude of  $f$  at any pair of  $(x,y)$  coordinates is termed the intensity or gray level of the image at now. When  $x$ ,  $y$  and also the intensity values of  $f$  are all finite.

#### 3.2 Image types

##### 3.2.1 Binary image

Logical array containing only 0s and 1s, interpreted as black and white, respectively.[60]

##### 3.2.2 Grayscale image

It is also known as an intensity image.Array of class uint8, uint16, int16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0,255]. For uint16, values range from [0, 65535]. For int16, values range from [-32768, 32767].[60]

##### 3.2.3 Rgb image

The abbreviation of the three colors (Red,Green,Blue),rgb image is an image in which each pixel is specified by three values one each for the red, blue, and green components of the pixel scalar. Mby-n-by-3 array of class uint8, uint16, single, or double whose pixel values specify intensity values. For single or double arrays,values range from [0, 1]. For uint8, values range from [0, 255]. For uint16, values range from [0, 65535].[60]

### 3.3 Characteristic of image

#### 3.3.1 Pixel

A digital image is made up of a finite number of elements, each with its own position or value called pixels.

#### 3.3.2 Image size

The image size refers to the number of pixels that make up a digital image, which are represented in columns and rows. For example, the image contains 100 x 100 pixels, which equals to an image height and width.

#### 3.3.3 Image dimensions

This is the image's dimensions. The latter is provided as a matrix with numerical values representing light intensities as its elements (pixels). The total number of pixels in an image is calculated by multiplying the number of rows in this matrix by the number of columns.[61]

#### 3.3.4 Brightness

The brightness can be defined as the amount of energy emitted by a light source. Sometimes it's easy to see that the picture is bright, and sometimes it's hard to see.

## 4 Convolutional neural networks

Convolutional neural networks (CNNs) have been around for a long time, and their popularity is evidenced by the many different architectures that are available. CNN's are designed to process multiple arrays of data, rather than relying on a single layer of neurons like MLPs. This allows CNNs to more accurately simulate how our eyes perceive images. This shows that the computer can solve computer vision problems very well, such as image classification, recognition, and understanding.[62] [63]

CNN proved its capability in image classification, it captured a lot of attention after winning a competition of ImageNet.[64]

We will discuss in the next subsection the different layers of the Convolutional neural network.

## 5 Convolutional Neural Network Layers

ConvNet is a sequence of layers, each of it applies a different function from set of activations function. We use three main types of layers in Convolutional Neural Networks (CNNs): Convolutional Layer, Pooling Layer, and Fully-Connected Layer, we will stack these layers to build a full neural network architecture.[65]

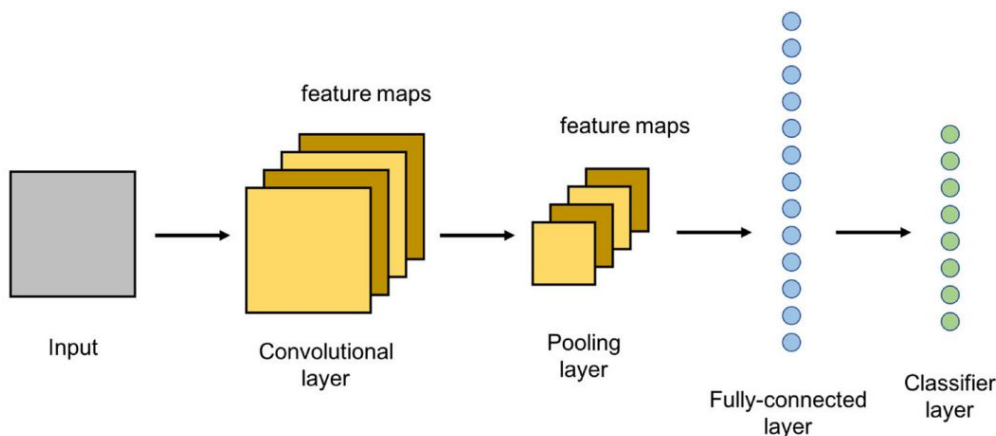


Figure 2.1: General CNN architecture

### 5.1 Convolutional Layer

The Convolution Layer is the most basic in a CNN, but it is also very important. The activation map for an image is generated by multiplying the pixel matrix of the image by a corresponding activation function. The main advantage of an activation map is that stores all the distinguishing features of a given image while at the same time reducing the amount of data of it. The matrix with which the data is convolved is a feature detector which is a set of values that the machine is compatible with. Different versions of the image are generated based on the value of the feature detector. The ConvNet use backpropagation in the training in order to ascertain minimal error in each layer. [66] At the end of each convolution operation, the activation function ReLu is applied to convolution layer to enhance the generalization.

There are some hyperparameters we can specify in our Conv layer :

#### 5.1.1 Padding

It is often necessary to extend the kernel beyond the activation map in order to achieve desired results. Padding helps to save data at the boundaries of activation maps, which can lead to

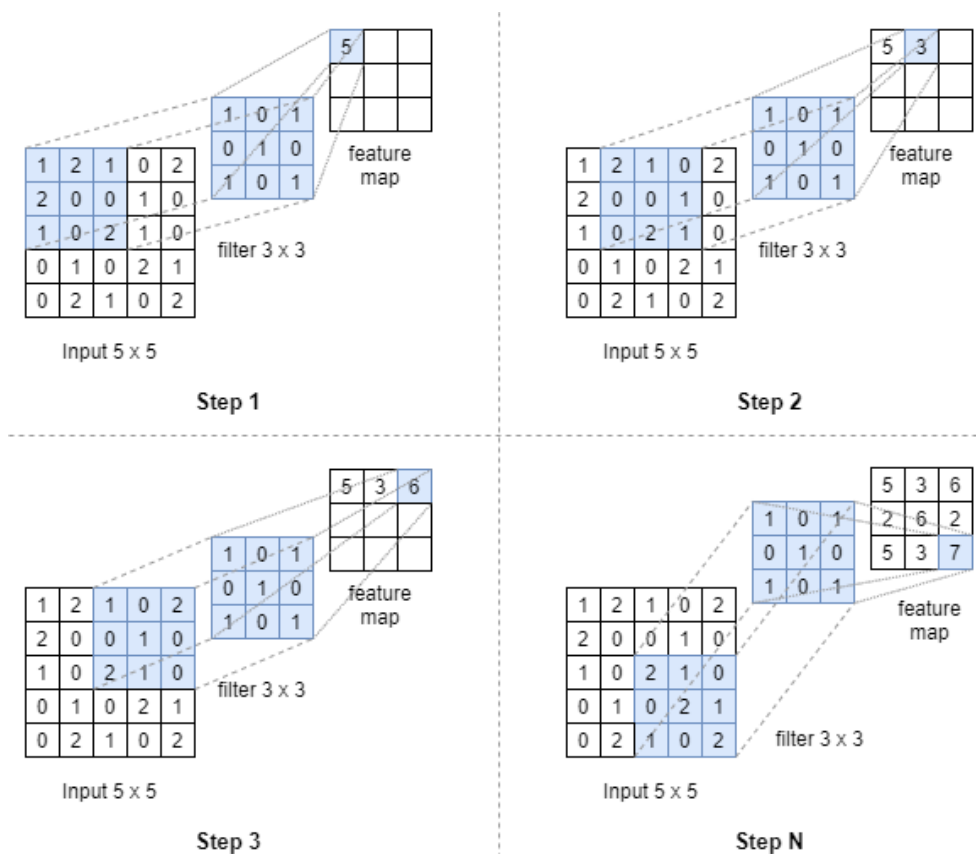


Figure 2.2: Examples of the convolution operation with filter (3 X 3) and stride of 1

better performance. and it can help preserve the input's spatial size [67], which allows an architecture designer to build deeper networks for reaching a higher performance. There exist many padding techniques[68], The most commonly used approach is to zero-pad data because of its performance, simplicity, and computational efficiency. The technique involves adding zeros symmetrically around the edges of an input. Many high-performing CNNs adopt this approach like AlexNet.[64]

As we keep applying convolution layers to our input the spatial dimensions of images decreases, if we want to keep image size the same for preserving the information we could add padding as like shows the figure ( 2.3). The input volume is 32 x 32. If we imagine two borders of zeros around the volume, this gives us a 36 x 36 x 3 volume, then we apply our Conv layer with our three 5 x 5 x 3 filters and stride of 1, then will also get 32 x 32 x 3 output volume.

### 5.1.2 Kernel size

The choice of a hyperparameter has a big impact on the image classification task. Small kernel sizes are able to extract a great deal of information from the input containing local features

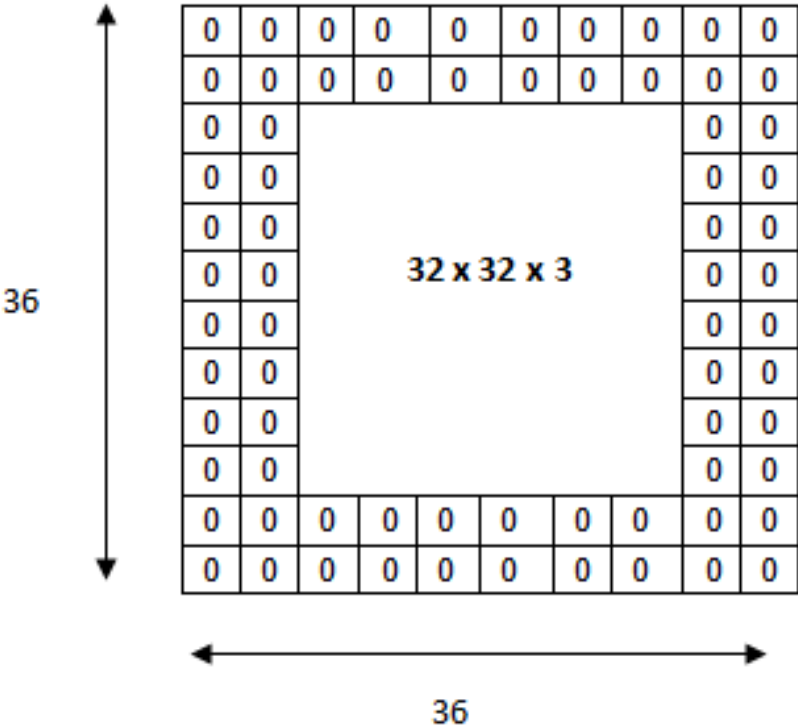


Figure 2.3: Examples of adding a padding to an image

very well. The smaller the kernel size, the smaller the reduction in layer dimensions, which allows for a deeper architecture. Large kernel sizes result in less information being extracted, which can lead to slower layer reduction and often poorer performance. Large kernels are better suited for extracting larger features. Ultimately, the kernel size you choose will be based on the task and dataset you are working with, but in general, a larger kernel will be more efficient for certain tasks and smaller kernels will be more efficient for other tasks, a smaller kernel size leads to better performance for image classification task because an architectural designer can stack more and more layers together to learn more and more complex features.[3]

**5.1.3 Stride**

The choice of stride value in convolutional has a large impact on the number of calculations to be performed by the network, since, basically a stride higher than 1 is original to skipping several pixels, thus performing smaller multiplications and additions.[69]

### 5.2 Pooling Layer

A pooling layer that provides a typical downsampling operation reduce the dimension of the feature map to introduce a translation that is invariant at small change and distortion, and reduce the number of subsequent learnable parameters. Note that there are no learnable parameters in any of the grouping layers, while the filter size, step, and padding are the hyper-parameters pooling layer.[70]

#### 5.2.1 Max pooling

The max pooling operation requires the kernel size and stride to be chosen during architecture design. Once selected, the operation slides the kernel over the input with the specified stride, while selecting only the maximum value of each kernel slice from the input to obtain the output value.

#### 5.2.2 Average pooling:

Average pooling is a pooling operation that computes the average of patches in a feature map and uses it to create a downsampled feature map. It is usually used after convolutional layers. It adds a small amount of translation invariance meaning that translating the image by a small amount does not significantly affect the value of most pooling outputs. It extracts features more smoothly than Max Pooling which extracts more obvious features like edges.

### 5.3 Fully Connected Layer

In a CNN, the fully connected layers (FC) have the same structure as an MLP. The purpose of these layers is to learn the nonlinear combinations between the features extracted by the convolution layers. The result of the last convolution layer  $[N, N, N_c]$  is flattened into a vector of size  $[N * N * N_c]$ . This vector presents the input layer to the set of fully connected layers. In supervised classification, the last layer is used for prediction based on the Softmax activation function and sigmoid for binary classification.[71]

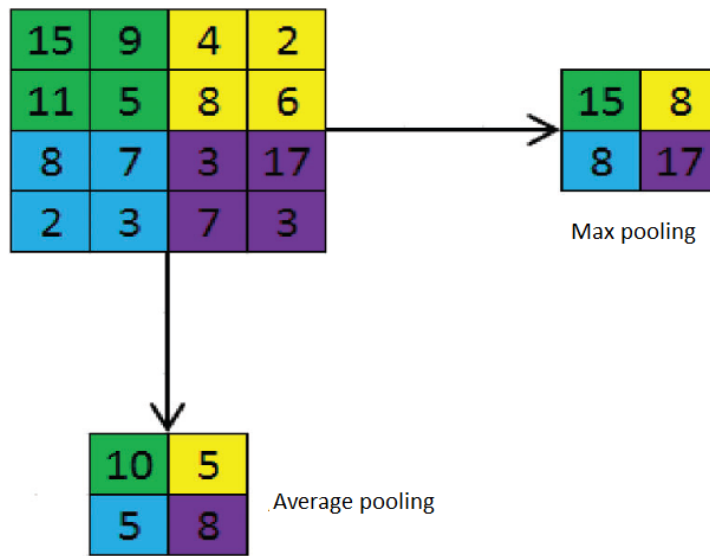


Figure 2.4: Max pooling versus Average pooling

## 6 Popular CNN architecture

### 6.1 AlexNet

AlexNet contains 5 complex classes as well as 2 classes fully connected layers for feature learning, it has maxpooling after the first, second and fifth convolutions Class. In total, it has 650,000 neurons, 60,000 parameters and 630M connections. AlexNet is the first page to show in-depth learning effectively in computer vision tasks.[64]



Figure 2.5: AlexNet introduced by Krizhevsky 2014

## 6.2 LeNet

Yan LeCun introduced convolution network called LeNet for digit recognition, it has 5 convolutional layers and one fully connected layer. [72]

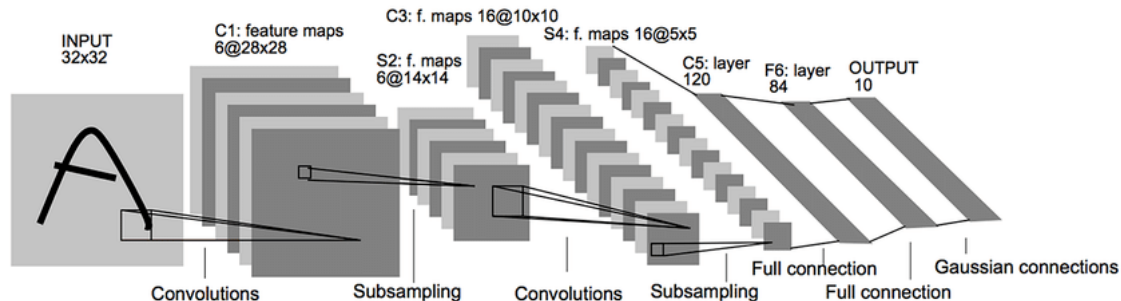


Figure 2.6: LeNet introduced by Yan LeCun

## 6.3 VGGNet

VGGNet is a convolutional neural network which is based on the same principles of the AlexNet network. The purpose of this version is to provide configurations deep (16 to 19 layers) based on the technique of structural stabilization. This technique makes it possible to control the number parameters in deep networks in order to reduce the risk of over-learning. The experimental study on the ImageNet learning base showed the positive effect of depth on performance, where the most deep are the most efficient.[3]

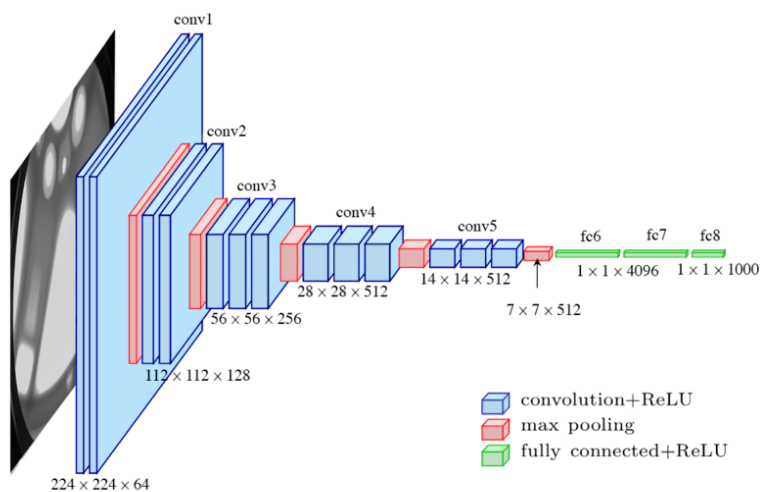


Figure 2.7: VGGNet network architecture [3]

# 7 Training Considerations: Batches, Epochs

Understanding those parameters could save us time and decrease computation load for getting better accuracy results.

## 7.1 Batch-size

Because deep learning requires large amounts of data to train, large batches may not be required. Optimally kept in memory, may cause Overfitting if batches are too small. Can take a long time to converge and can be noisy[73]. Batch size is (32-512) samples are commonly used in deep learning.[74]

## 7.2 Epochs

As mentioned, a single pass of data may not be optimal, so multiple passes. The data is created for training purposes and the full data is always passed will require  $\frac{n}{b-size}$  batch, where  $n$  is the number of records and  $b-size$  is batch size. In addition to previous experience and heuristics, the number of epochs is determined by observing the convergence plot of train/validation errors as the epoch progresses training process. The weights of the network can be stored as checkpoints, the networks could started from that checkpoint. [75]

# 8 Regularization

Overfitting is one of the most difficult problems that machine learning algorithms faces. When a model performs well in the training data but fails to categorize the validation data we call it Overfitting. Furthermore, the model is learning to a point the subtleties and noise in the training data. where it is unable to distinguish new data due to the noise it has acquired from the training data. As a result the models capacity to generalize begins to deteriorate. In the the subsections we will showsome of the regularization techniques for tackling overfitting in ConvNets.[76]

## 8.1 Data augmentation

Increasing data is another way to reduce Overfitting on models where we increase the number of training data using only the information contained in our training data. The field of data augmentation is not new and in fact, various data augmentation techniques have been applied

for specific problems. The main technique of data augmentation is an approach to find a way to directly increasing dataset by doing certain transformation to images or using gans algorithms to generate more samples for each class.[77].

The most common method today to increase data is to perform a multiple combination of transformations and color modifiers like rotation, reflection, scaling (zoom in/out), and crop.[78]

The ImageDataGenerator class in Keras implements data augmentation in such a way that it takes place in random access memory and has no effect on the original data on the hard disk, all we have to do is decide what changes to make and tune their parameters.[79]

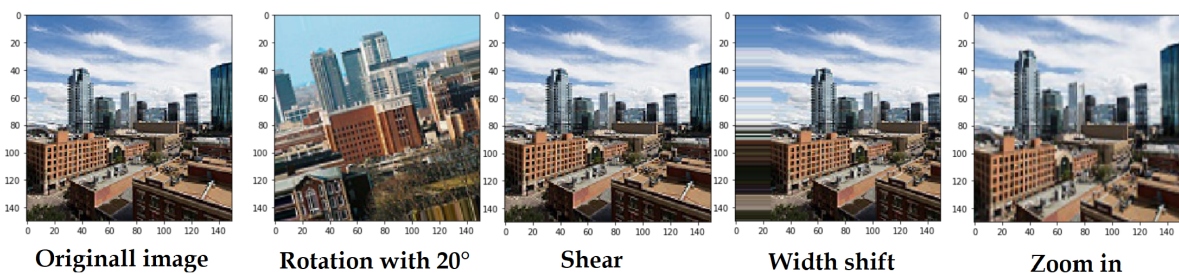


Figure 2.8: Example of applying data augmentation

## 8.2 Dropout

A technique called dropout is used for regularization [80], During training some random nodes of a chosen layer will be dropped, the percentage of nodes to be dropped in a layer is controlled by a parameter. That leads the network to reduce the number of trained parameters of smaller weights and similar effect. Dropout applied during training only, during the testing phase it reactivated. Often this could lead to smaller test loss than training loss for some epochs.[73]

## 8.3 Batch normalization

Distribution of layer's inputs changes during training because the parameters of the previous layers change complicates Deep Neural Network training as consequence the training require longer time since it necessitates slower learning rates and more careful parameter management. It's famously difficult to train models with saturating nonlinearities, batch normalization could be a simple and chic approach to re-parametrize nearly any deep network. The re-parametrization

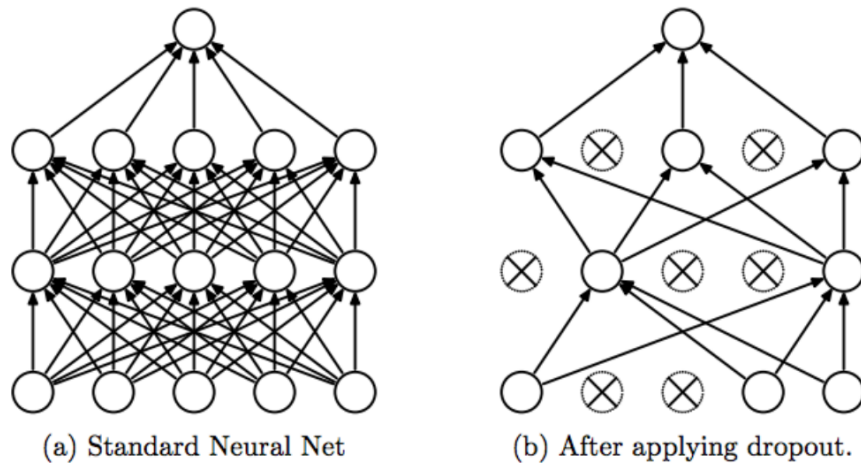


Figure 2.9: Visualisation of dropout technique

solves the problem of coordinating updates across several systems. In the other side lot of layers BatchNorm encompasses a significant impact on network training: it changes the environment of network training to substantially smoother this assures faster network convergence and better learning.[81]

### 8.4 Early stopping

Early stopping provides guidance on the number of iterations required to execute to minimize the cost function. Early stopping is often used to prevent the poor generalization of trained models. If model runs into small numbers of training epochs it tends to underfit also too high number of training epoch may leads the model to overfitting. Early Stopping is a deep neural network regularization strategy that terminates training when parameter updates no longer provide improvements on a validation set. In other words, during training, we save and update the current best parameters, and when parameter changes no longer generate an improvement (after a certain number of iterations), we terminate training and use the previous best values. It functions as a regularizer by limiting the optimization approach to a smaller parameter space.[82]

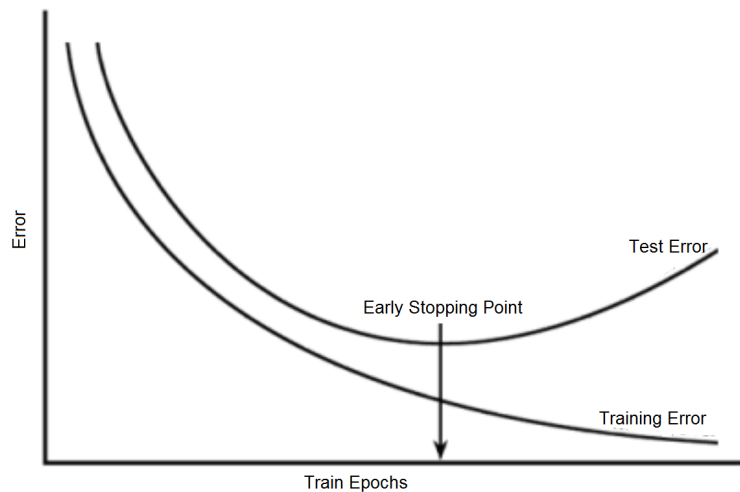


Figure 2.10: Example of Early stopping

## 9 Transfer Learning

It is common knowledge that powerful models require a large amount of training data, high-performance equipment, and a long period of training time to realize their full potential. Despite this, they have a high level of precision. Transfer learning makes use of the information gained by these large-scale models by using it to solve issues that are similar but more particular.[83]

Transfer learning is the process of applying a trained model to one problem to solve a related problem in some way. Jason Yosinski et al explained transfer learning as "In transfer learning, we first train a base network on a base dataset and task, and then we re-purpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task".[84]

Transfer learning has the advantage of reducing the training time of the neural network. This is a model and it can reduce the generalization error. The weight of the reusable layer is used as a starting point for training and adapting to new situations and problems. This usage treats transfer learning as a kind of weight initialization scheme. This is useful if the first related problem has more labeled data than the target problem and the structural similarity of the problem is useful in both contexts. Using a pre-trained model involves downloading the model, using it in your application, and using it to classify new images.[85]

## **10 Conclusion**

In this chapter, we have defined the computer vision field, then we throw images and its types and characteristics, in the next section we dive into convolutional neural networks by highlighting its layers (convolutional layer, pooling layer, fully connected layer), we expose some CNN architecture, Finally, we talked about regularization techniques for better network training.

# Chapter 3

## Implementation

### 1 Introduction

During our experiments, We have created many convolutional neural network models based on different parameters.in This chapter introduces the software, libraries, and hardware also the dataset used for the experiments, we present the implementation of our convolutional neural network models for image classification, then we will discuss the results of each model and compare them to each other . We will use three techniques for regularization data augmentation[86], batch normalization and dropout [87] to avoid overfitting and for better learning processes.

### 2 Software and libraries used in the implementation:

#### 2.1 Python

The Python programming language is gaining tremendous popularity among data scientists and software developers , different from the R programming language that is mainly intended for statistical data analysis, Python shows up in a much wider range of applications such as Internet and website development, database access, desktop GUIs, scientific computation, and software and game development , Python is not a compiled language, meaning that it does not pre compile the code into binary. [88]

### 2.1.1 Tensorflow

TensorFlow is a flexible and extensible software library for numerical computation using dataflow graphs. This library and related tools make it easy to program and train neural networks and other machine learning models, and then deploy them to production. The core algorithms of TensorFlow are written in a highly optimized C++ and CUDA platform, which was created by NVIDIA. The software has APIs available in several languages. The Python API is the most comprehensive and reliable one. Other languages that are officially supported by the company are JavaScript, C++, Java, Go, and Swift. Third party packages available for other languages such as C sharp and Ruby. The examples in this review are all based on Python, which is the easiest language to read and use.[89]

### 2.1.2 Keras

Keras is a library that makes deep learning easy to use, and it can be used with TensorFlow, Theano, this toolkit simplifies the process of creating neural networks by providing layers and other essential components, while taking care of the mathematical details of tensors. TensorFlow is the best option for back-end data processing for Keras. You can use Keras for deep learning applications without needing to use TensorFlow or Theano, which can be relatively complex programs. There are two major types of programming frameworks: the sequential API and the functional API, the sequential API is based on the idea of a sequence of layers; this is the most common use of Keras and the simplest part of Keras. The sequential model can be thought of as a linear stack of layers.[75]

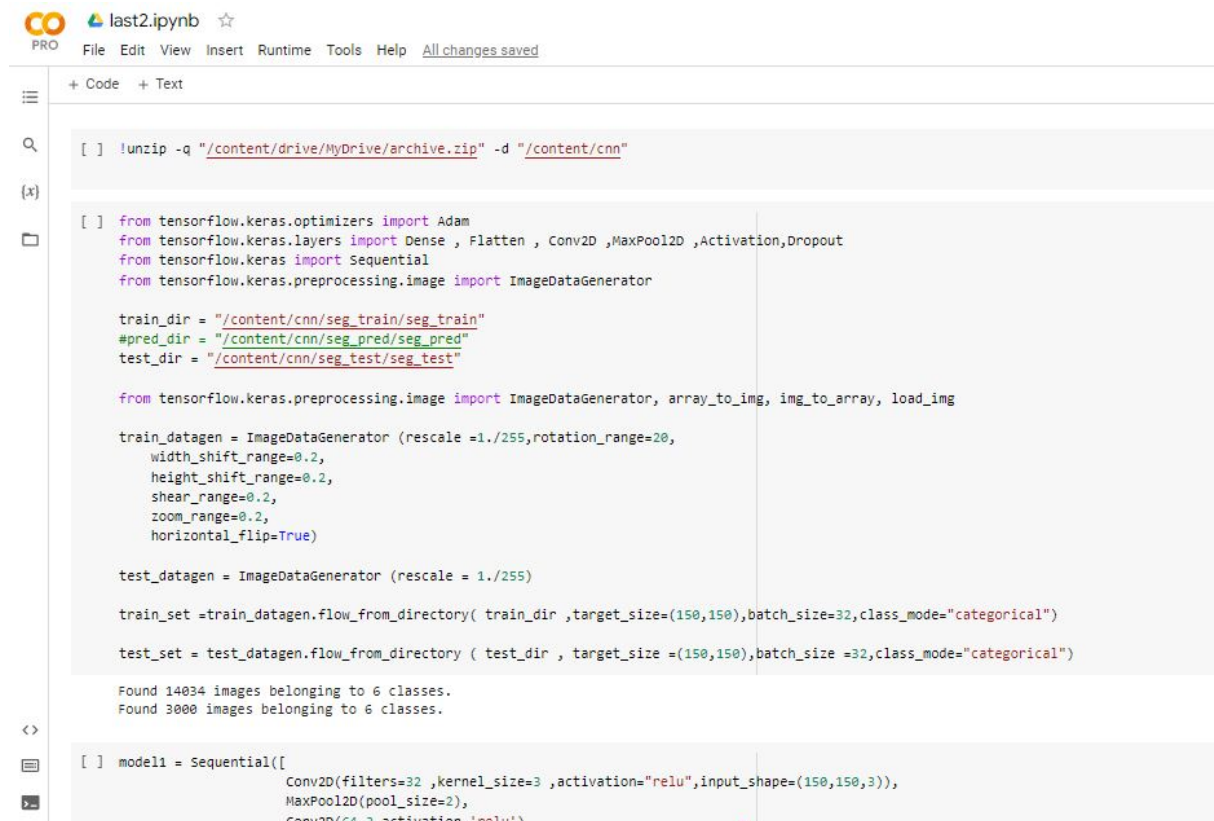
### 2.1.3 Matplotlib

Matplotlib is a portable 2D plotting and visualization package for mainly the visualization of scientific, technical and financial data. matplotlib can be used interactively from the Python shell, called from python script or integrated into a graphical application (GTK, Wx, Tk, Windows). Much Common paper outputs are supported, including JPEG, PNG, PostScript, and SVG. Features include creating multiple axes and shapes per page, interactive navigation, multiple predefined line styles and symbols, images, anti-aliasing, alpha matching, date and financial graphics, W3C compliant font handling and FreeType2 support, legends and tables, pseudo-color cells, math text and more. It works with both numarray and Numeric. Package target, core architecture, current features (illustrated with examples) and forecast Improvements will be described.

[90]

### 2.2 Google Colab Pro

Google Colaboratory widely known as Google Colab is an open source service provided by Google to any person having a Gmail account. Google Colab provides GPU for research to the people who do not have enough resources or cannot afford one. The Google Colab service provides 13 GB of RAM and 108 GB of hard disk space in one runtime. Every runtime lasts for 12 hours after which the runtime is reset and the user has to establish a connection again. [91]



```
[ ] !unzip -q "/content/drive/MyDrive/archive.zip" -d "/content/cnn"

[ ] from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Activation, Dropout
from tensorflow.keras import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_dir = "/content/cnn/seg_train/seg_train"
#pred_dir = "/content/cnn/seg_pred/seg_pred"
test_dir = "/content/cnn/seg_test/seg_test"

from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img

train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale = 1./255)

train_set = train_datagen.flow_from_directory( train_dir ,target_size=(150,150),batch_size=32,class_mode="categorical")

test_set = test_datagen.flow_from_directory ( test_dir , target_size =(150,150),batch_size =32,class_mode="categorical")

Found 14034 images belonging to 6 classes.
Found 3000 images belonging to 6 classes.

[ ] model1 = Sequential([
    Conv2D(filters=32, kernel_size=3, activation="relu", input_shape=(150,150,3)),
    MaxPool2D(pool_size=2),
    Conv2D(64, 3, activation='relu')
```

Figure 3.1: Google Colab interface

## 3 Configuration used in the implementation

This our configuration that we have used during the learning process.

GPU	Tesla P100 16GB
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
RAM	13 GB
OS	Google Colab runs on Linux
DISK STORAGE	108 GB

Table 3.1: Configuration used in the implementation.

## 4 Dataset

Intel image classification challenge dataset[92] consists of different types of images of dimensions (150 x 150 x 3 ), which are natural scene images divided into 6 categories (Buildings, Forest, Glacier, Mountain, Sea, and Street). The dataset is split into training , test set and prediction. For the training, we got around 14000 images, 3000 images for the testing and 7300 for prediction the figure 3.2 shows the number of samples for training and testing of each class.

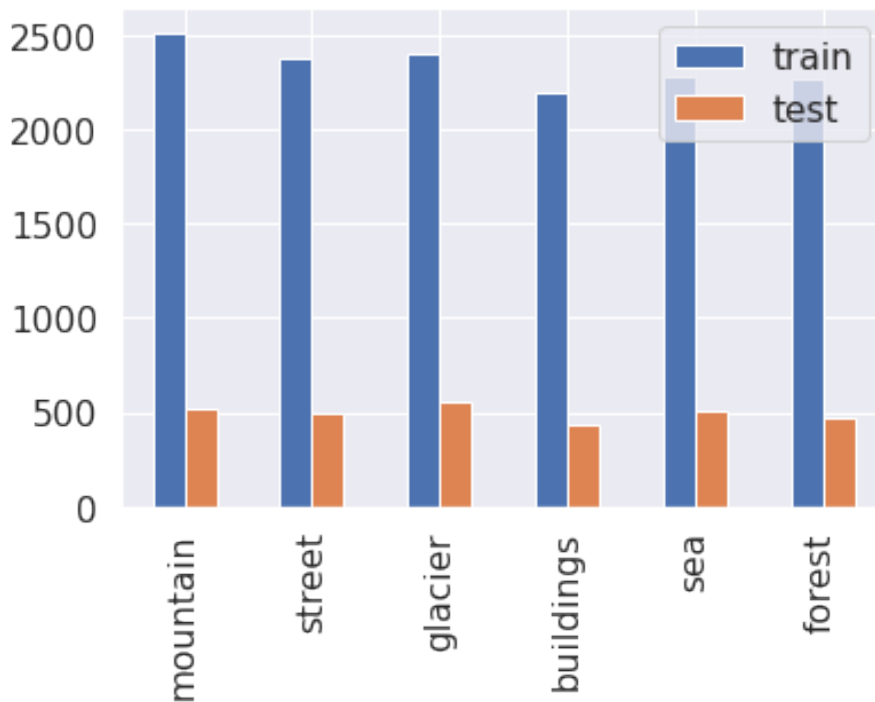


Figure 3.2: Example of images of the dataset

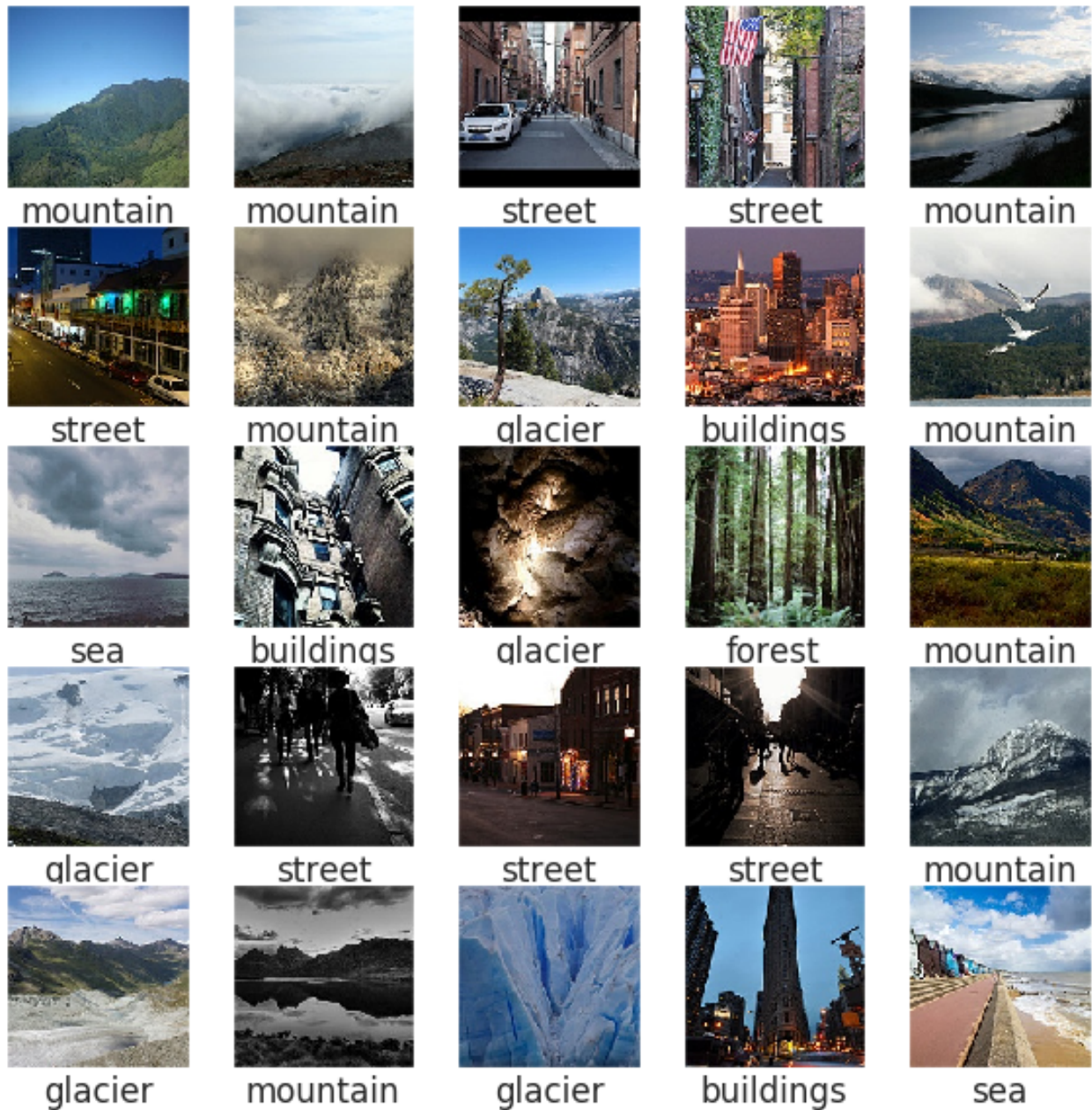


Figure 3.3: Example of images of the dataset

## 5 Network architecture

In this section, we have created 3 different convolution neural networks as shown below :

- **Model A** : composed of 10 different layers (3 convolutuionals, 3 maxpooling, 1 flatten, 1 dropout, 2 dense) figure (3.4).

- **Model B** : composed of 12 different layers (4 convolutions, 4 maxpooling, 1 flatten, 1 dropout, 2 dense) figure 3.5.
- **Model C** : composed of 14 different layers (6 convolutions, 4 maxpooling, 1 flatten, 1 dropout, 2 dense) figure 3.6.
- **Model D** : composed of 16 different layers (4 convolutions, 4 maxpooling, 4 Batchnormalization, 1 flatten, 1 dropout, 2 dense) figure 3.7.

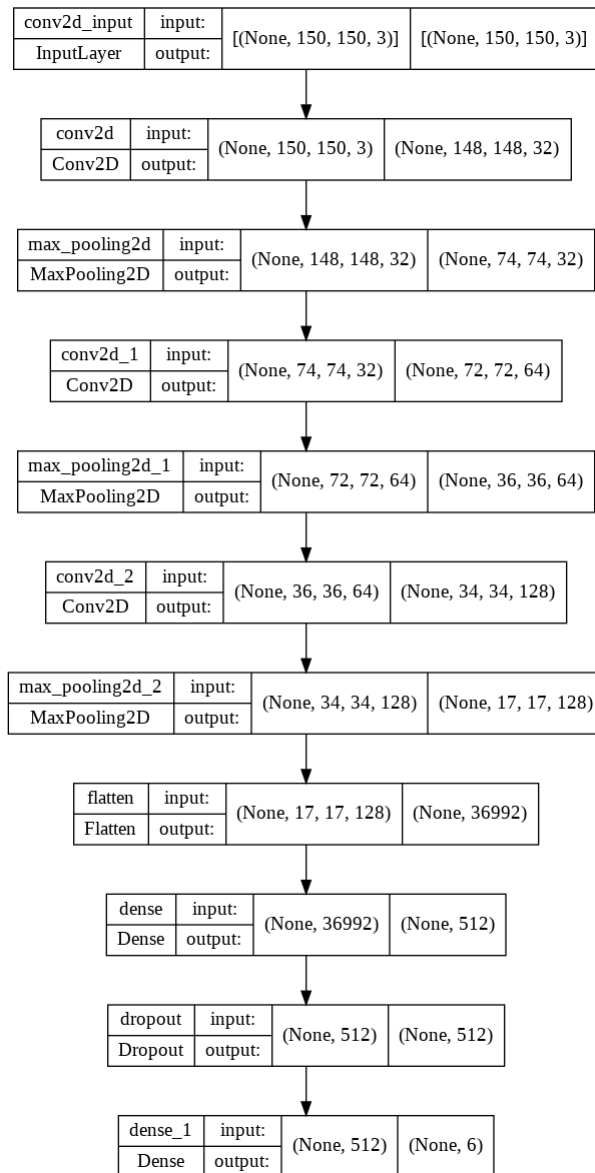


Figure 3.4: Model A architecture

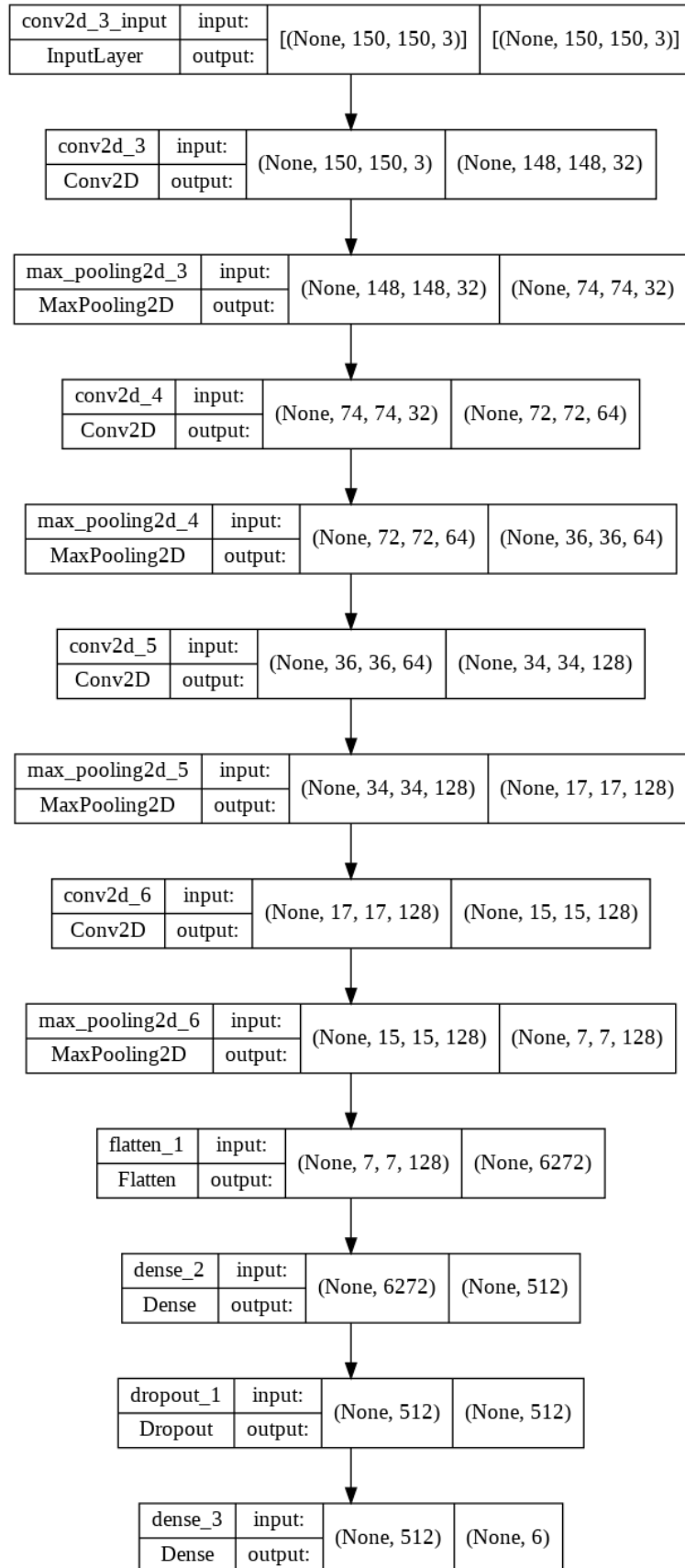


Figure 3.5: Model B architecture

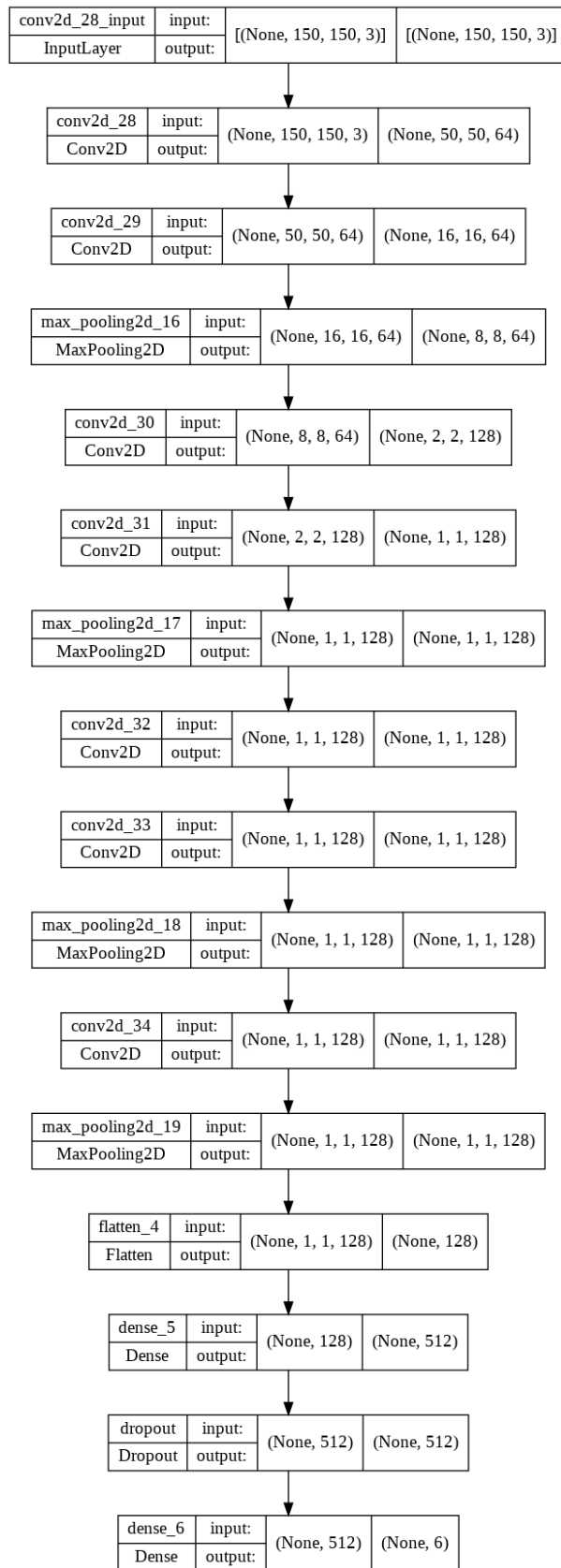


Figure 3.6: Model C architecture

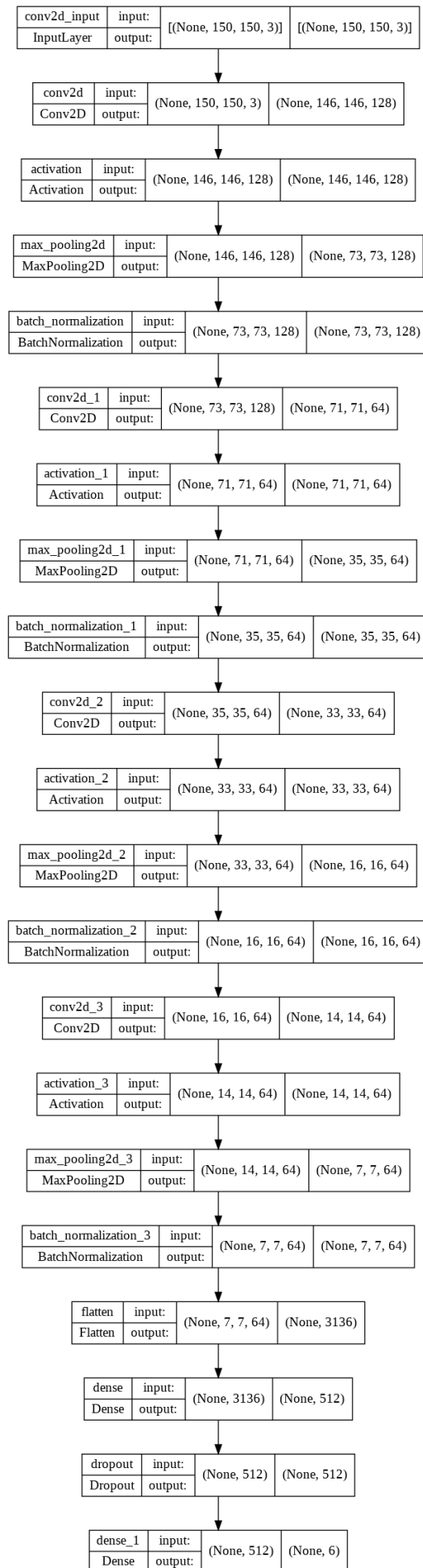


Figure 3.7: Model D architecture

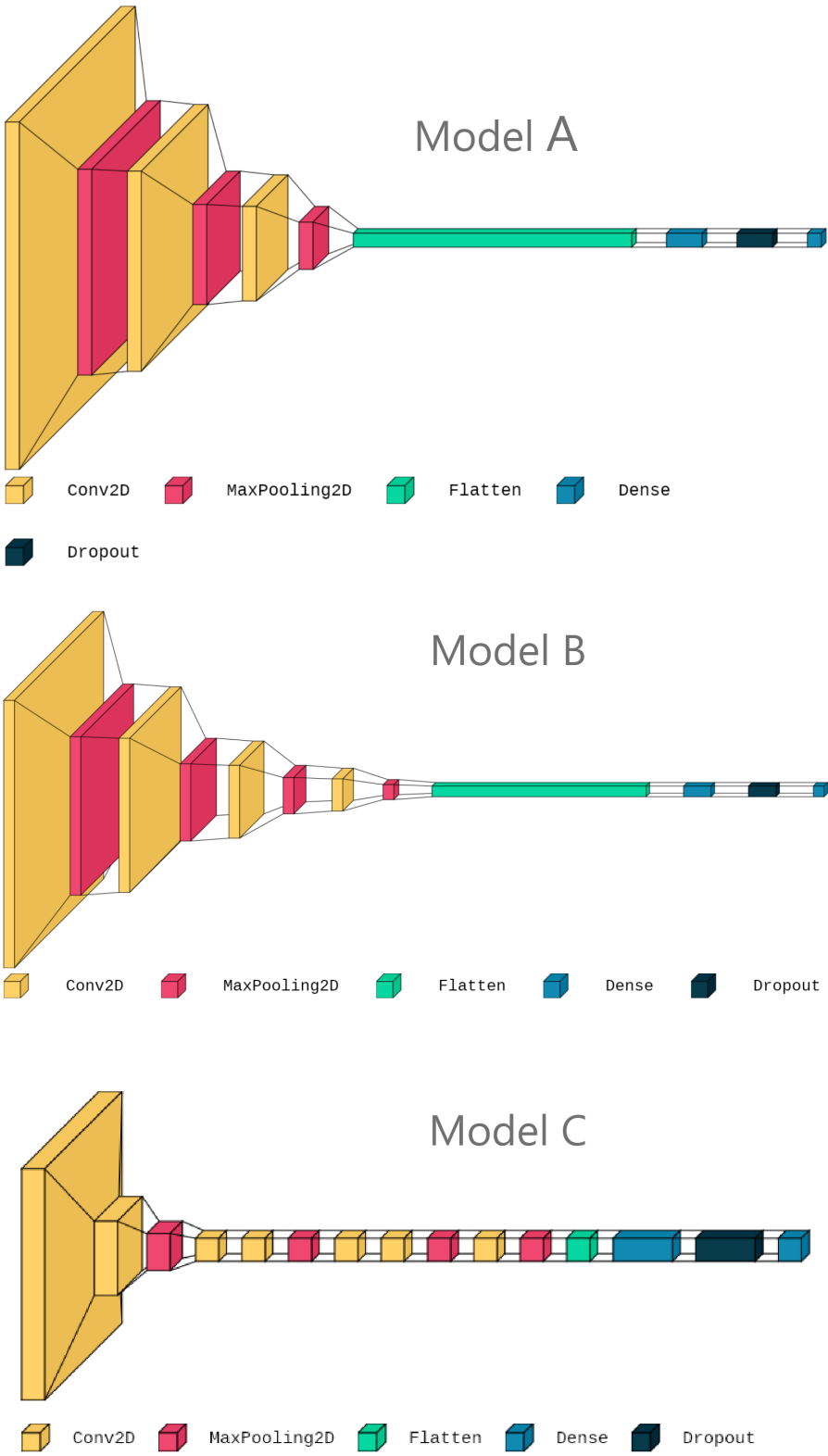


Figure 3.8: Visualizing of model A, B and C

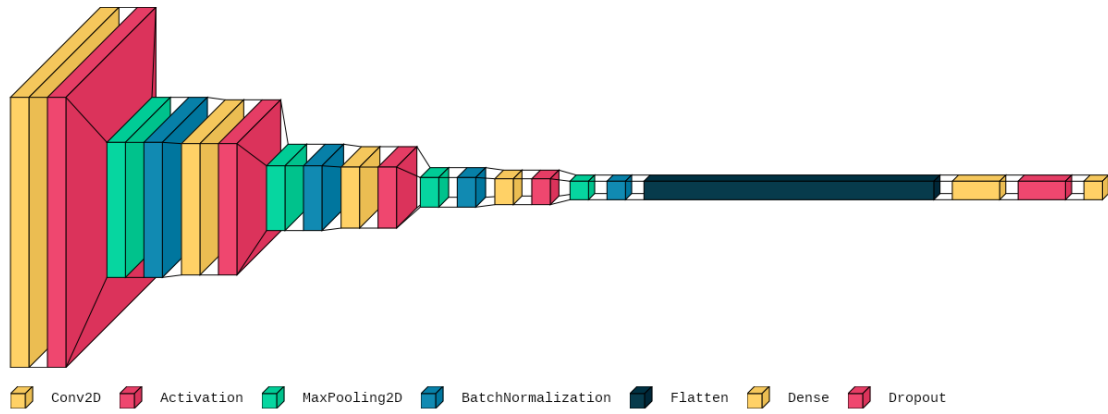


Figure 3.9: Visualizing of model D

## 6 Results and discussion

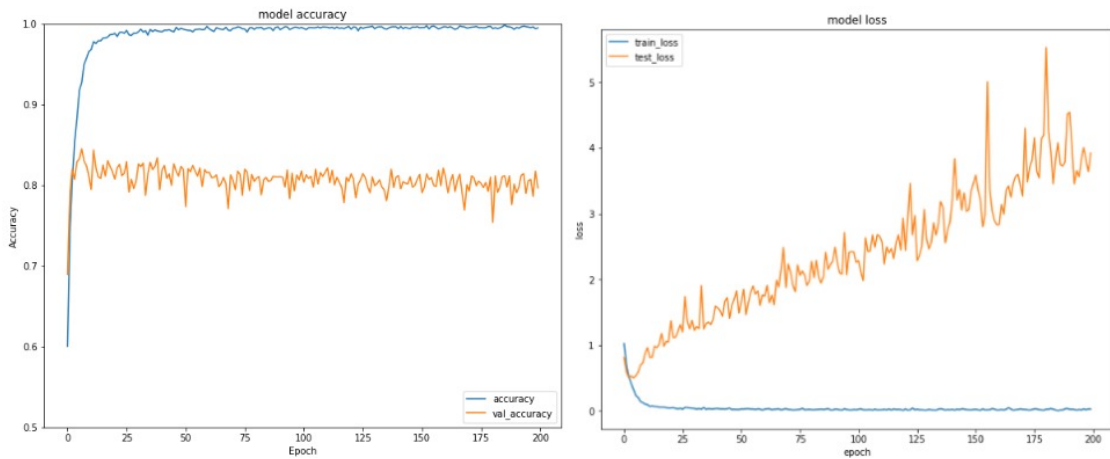


Figure 3.10: Test accuracy and loss plots for model A without data augmentation

From figure 3.10 , during the training, we observe that the training accuracy curve is increasing rapidly close to reaching the peak, while the growth of validation accuracy is a logarithmic function of epochs with the presence of fluctuations, in the other side validation loss is increasing so quickly while the training loss curve is decreasing quickly close to 0.

We observe that our model is doing well on the training data set but poorly on the validation set which we call that overfitting. For solving the overfitting problem we are going to use a technique mentioned in chapter 2 called data augmentation in the next experiments.

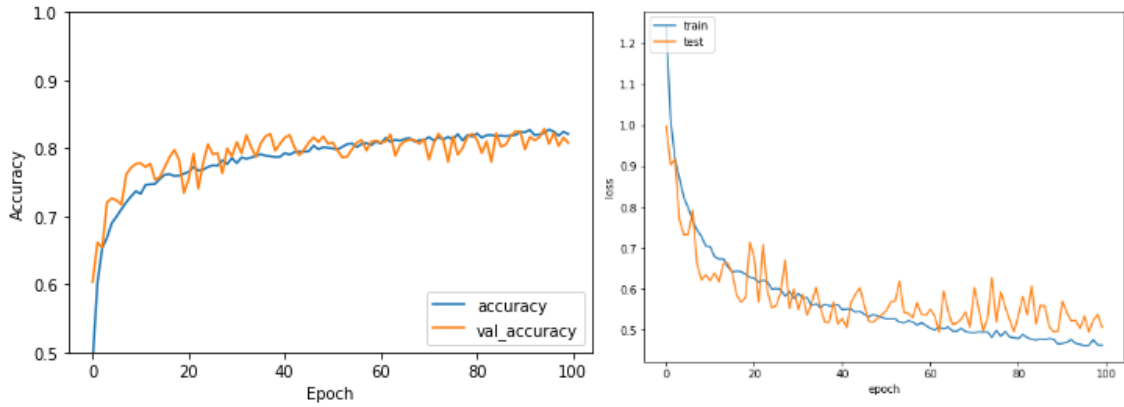


Figure 3.11: Test accuracy and loss plots for model A

In the figure 3.11 that represents the testing Accuracy and loss for model A after implementing the data augmentation technique. The training accuracy and validation accuracy are growing rapidly but the validation accuracy is a bit fluctuating. On the other side, the loss curve of training and validation is decreasing. As we see above that we have solved the overfitting problem, our model now can generalize better.

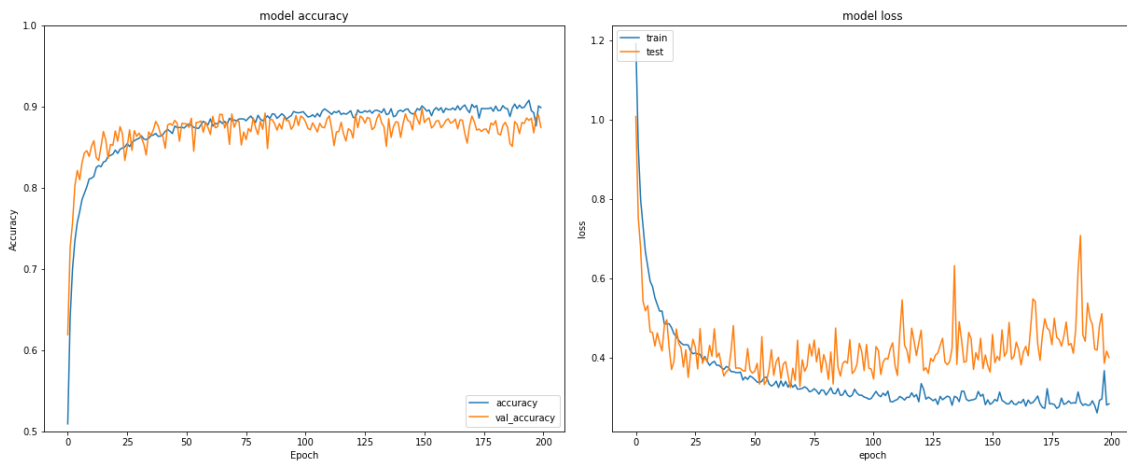


Figure 3.12: Test accuracy and loss plots for model B

Now we aim to achieve better accuracy, the figure 3.12 represents the loss and accuracy of model B, as we observe the accuracy of training and validation are increasing with each epoch in a logarithmic way. In the other part, the loss curve of training is decreasing rapidly while the validation starts to increase fast in the few first epochs then it keeps growing slightly in the presence of fluctuations.

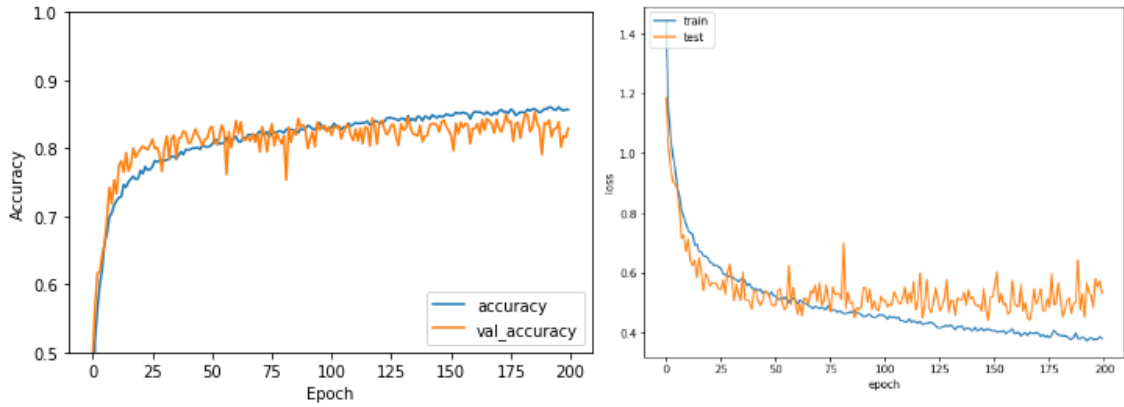


Figure 3.13: Test accuracy and loss plots for model C

For model C, from figure 3.13 the training accuracy is increasing by the number of epochs in a logarithmic way also validation accuracy slightly fluctuating, while the training loss is falling rapidly and the validation loss started falling in the first few epochs then increases again slightly.

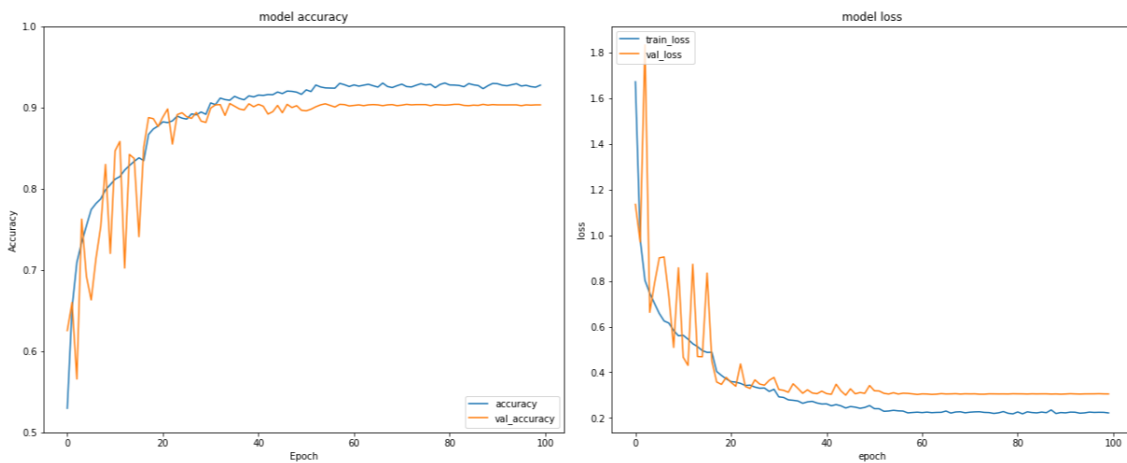


Figure 3.14: Test accuracy and loss plots for model D

For model D, from figure 3.14 both training and validation accuracy are increasing by the number of epochs rapidly, then it starts to stabilize after the 50th epoch, while the training and validation loss decreasing rapidly .

### Chapter 3. Implementation

Model name	Training accuracy	Validation accuracy	Training loss	Validation loss	Epoch Number	Run time
Model A	82,12%	80,80%	47,32%	50,09%	200	273,5 min
Model B	89,23%	87,43%	29,32%	40,01%	200	274,33 min
Model C	84,29%	82,93%	39,32%	53,23%	200	300,5 min
Model D	92,15%	90,86%	22,12%	30,34%	100	150,83 min

Table 3.2: Results Comparison Table

From the table 3.2 we can observe that model D got the highest accuracy compared to all other models with a validation accuracy of 90,86% and for training 92,15% as consequence in the further sections we will use model D for our prediction.



Figure 3.15: Confusion matrix for model D

A confusion matrix is a summary of classification problem prediction outcomes. It resumes the number of correct and bad predictions. As we see on figure 3.15 confusion matrix of model D which is doing great work for the most of classes, however it is getting a bit confused between (buildings, street) classes ,also ( glacier and mountain) classes.

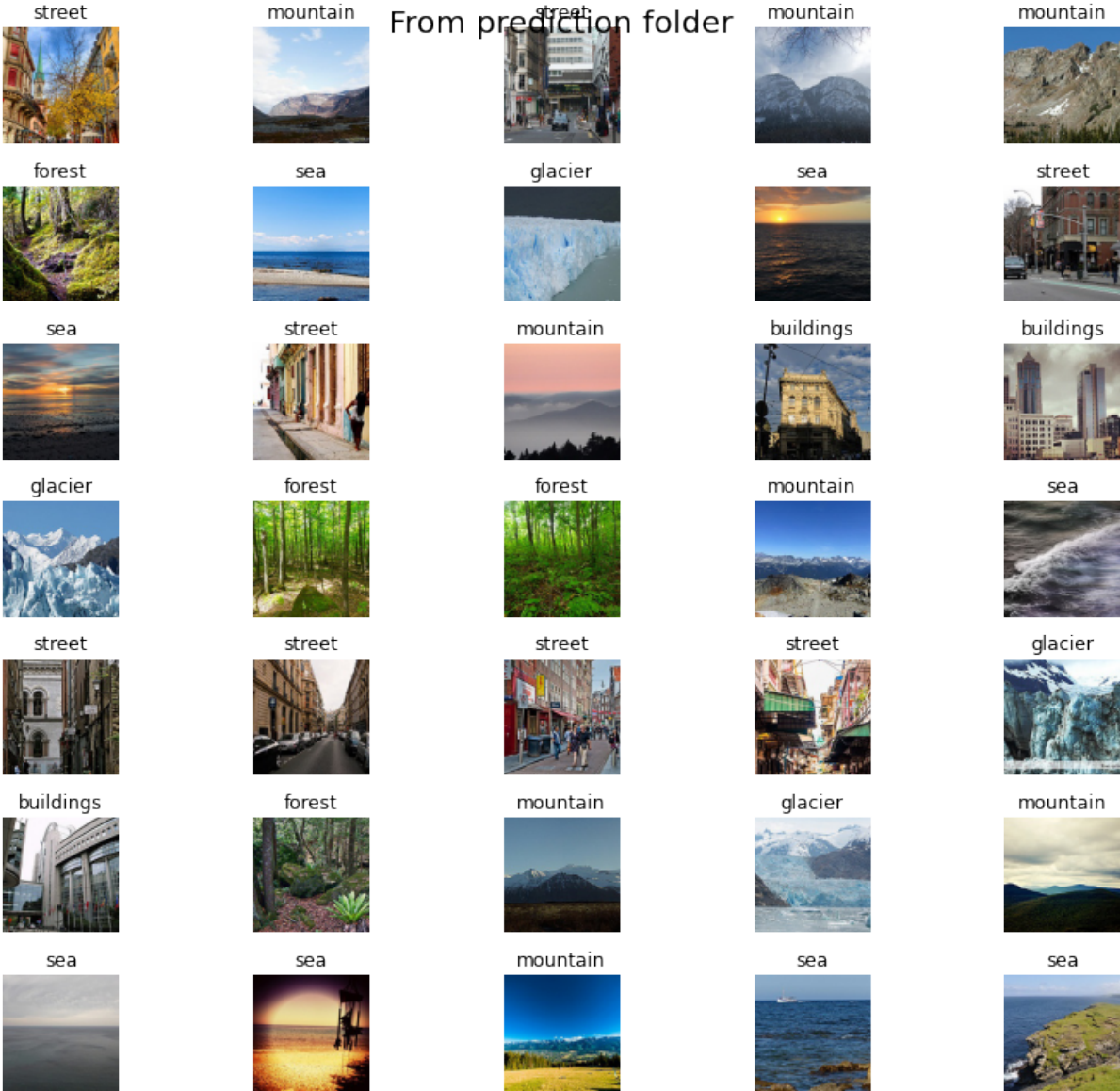


Figure 3.16: Predictions by Model D

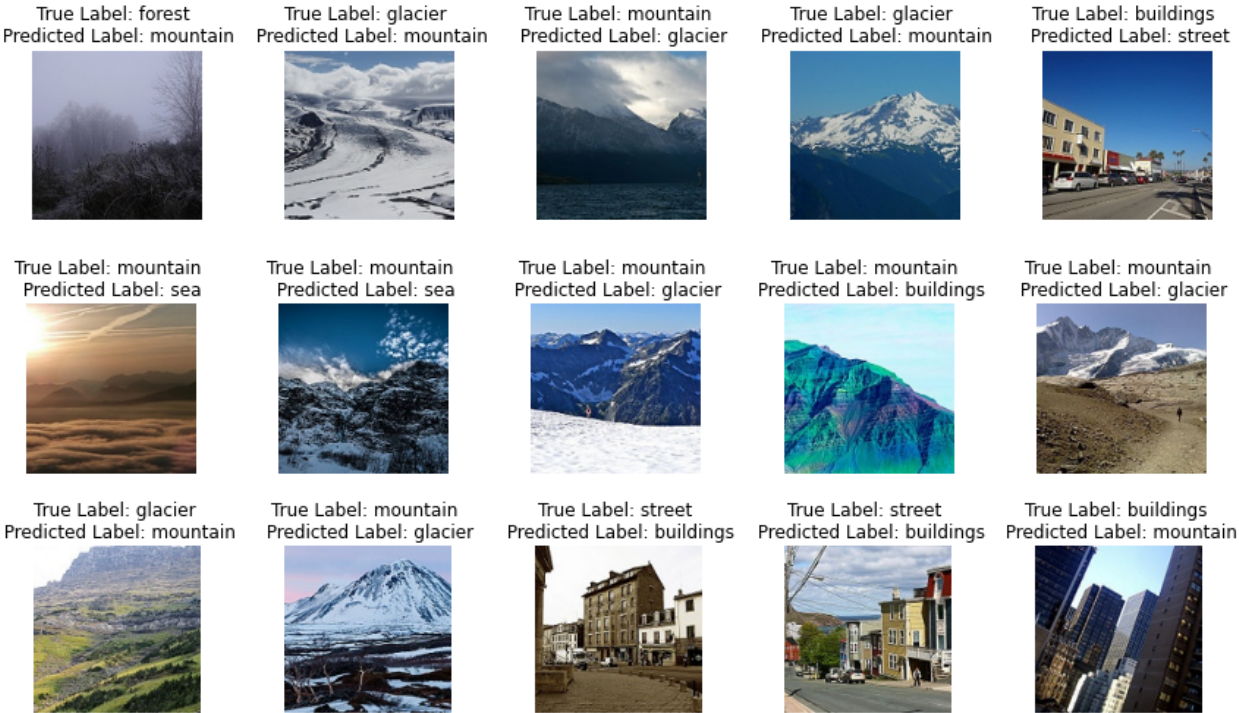


Figure 3.17: Bad predictions by Model D on test dataset

### 7 Conclusion

We have presented in this chapter different convolutional neural network models for the image classification that we have built, we go through some regularization techniques such as data augmentation, batch normalization, and dropout seeking for best results. As result, we have seen from our experiments that there are several parameters (epoch, batch size ) and techniques (data augmentation, batch normalization) that improves CNN model and to achieve a good result.

# Conclusion

In this work we have discussed the fundamental concepts behind machine learning, we focused our discussion on the most fruitful class of machine learning, namely neural networks also known as deep learning, we paid a special attention to convolutional neural networks algorithms, where we have introduced convolutional neural networks and explained the role of different types of layers that are used in the classification process (convolution layer, pooling layer, and fully connected layer), we have also presented some popular CNN models, after that we have discussed the different parameters that effect the training process of CNN : activation functions, loss functions, optimizers, and regularization technique.

In this work we found out that the classification task requires a significant computing power as a consequence of this requirement we have used Google Colab Pro platform, which saved us allot of computing during our experimentation with different architectures. We also have found out that the choice of the dataset is critical for CNN training, in order to achieve optimal results we recommend using high resolution and polyvalent dataset.

We have spent a fairly large amount of learning about the concepts of CNN from various primary sources, in order to comprehend the mechanisms that govern CNN models function, which played a major role in our attempt to design an optimal CNN model in the most efficient way possible, during which we have discovered that the task of creating an efficient CNN model is indeed challenging because of the different parameters that effect the CNN learning process, nonetheless in our search for improving the results of our models we have discovered the existence of multiple techniques that are applied to improve the results of the CNN model, such as batch normalization, dropout, and data augmentation.

This work has been a milestone for us, as perspectives for future work, we aim at using Transfer Learning for solving more complex and challenging problems.

# Bibliography

- [1] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. Machine learning in materials science. *InfoMat*, 1(3):338–358, 2019.
- [2] Jennifer S Raj, J Vijitha Ananthi, et al. Recurrent neural networks and nonlinear prediction in support vector machines. *Journal of Soft Computing Paradigm (JSCP)*, 1(01):33–40, 2019.
- [3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [4] Arthur L Samuel. Machine learning. *The Technology Review*, 62(1):42–45, 1959.
- [5] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.
- [6] John McCarthy. What is artificial intelligence. URL: <http://www-formal.stanford.edu/jmc/whatisai.html>, 2004.
- [7] Ian Goodfellow. Deep learning of representations and its application to computer vision. 2015.
- [8] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.
- [9] Tammy Jiang, Jaimie L Gradus, and Anthony J Rosellini. Supervised machine learning: a brief primer. *Behavior Therapy*, 51(5):675–687, 2020.
- [10] Aized Amin Soofi and Arshad Awan. Classification techniques in machine learning: applications and issues. *Journal of Basic and Applied Sciences*, 13:459–465, 2017.

- [11] Dastan Maulud and Adnan M Abdulazeez. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1(4):140–147, 2020.
- [12] M Emre Celebi and Kemal Aydin. *Unsupervised learning algorithms*. Springer, 2016.
- [13] T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.
- [14] Krzysztof J Cios, Roman W Swiniarski, Witold Pedrycz, and Lukasz A Kurgan. Unsupervised learning: Association rules. In *Data Mining*, pages 289–306. Springer, 2007.
- [15] Marco A Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729, 2012.
- [16] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [17] Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.
- [18] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.
- [19] Ali Asghar Heidari, Hossam Faris, Seyedali Mirjalili, Ibrahim Aljarah, and Majdi Mafarja. Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. *Nature-inspired optimizers*, pages 23–46, 2020.
- [20] Michael D Twa, Srinivasan Parthasarathy, Cynthia Roberts, Ashraf M Mahmoud, Thomas W Raasch, and Mark A Bullimore. Automated decision tree classification of corneal shape. *Optometry and vision science: official publication of the American Academy of Optometry*, 82(12):1038, 2005.
- [21] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer Science & Business Media, 2012.

- [22] Johannes Fürnkranz and Tomáš Kliegr. A brief overview of rule learning. In *International symposium on rules and rule markup languages for the semantic web*, pages 54–69. Springer, 2015.
- [23] David Heckerman. A tutorial on learning with bayesian networks. *Innovations in Bayesian networks*, pages 33–82, 2008.
- [24] J Russell Stuart and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 2009.
- [25] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. *arXiv preprint arXiv:1302.4964*, 2013.
- [26] Oliver Kramer. K-nearest neighbors. In *Dimensionality reduction with unsupervised nearest neighbors*, pages 13–23. Springer, 2013.
- [27] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [28] AH Nizar, ZY Dong, and Y Wang. Power utility nontechnical loss analysis with extreme learning machine method. *IEEE Transactions on Power Systems*, 23(3):946–955, 2008.
- [29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [31] John Bryden. Biologically inspired computing: The neural network. URL <https://www.academia.edu/35380032/BiologicallyInspiredComputingTheNeuralNetwork>.
- [32] Geonyeop Lee, Ji-Hwan Baek, Fan Ren, Stephen J Pearton, Gwan-Hyoung Lee, and Jihyun Kim. Artificial neuron and synapse devices based on 2d materials. *Small*, 17(20):2100640, 2021.
- [33] Marvin Minsky and Seymour Papert. *Perceptrons*. 1969.
- [34] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10:978–3, 2018.

- [35] Meha Desai and Manan Shah. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn). *Clinical eHealth*, 4:1–11, 2021.
- [36] Nassima Dif. *L'apprentissage profond pour le traitement des images*. PhD thesis, Djillali Liabes University, 2020.
- [37] Thamer M Jamel and Ban M Khammas. Implementation of a sigmoid activation function for neural network using fpga. In *13th Scientific Conference of Al-Ma'moon University College*, volume 13, 2012.
- [38] Yu-Dong Zhang, Chichun Pan, Xianqing Chen, and Fubin Wang. Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. *Journal of computational science*, 27:57–68, 2018.
- [39] Andrinandrasana David Rasamoelina, Fouzia Adjailia, and Peter Sinčák. A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 281–286. IEEE, 2020.
- [40] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.
- [41] Jojo Moolayil, Jojo Moolayil, and Suresh John. *Learn Keras for deep neural networks*. Springer, 2019.
- [42] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212, 2022.
- [43] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.
- [44] Kevin Koidl. Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*, 2013.
- [45] Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

- [46] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [47] T Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical Report*, 2017.
- [48] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11127–11135, 2019.
- [49] Dongwoo Lee, Sybil Derrible, and Francisco Camara Pereira. Comparison of four types of artificial neural network and a multinomial logit model for travel mode choice modeling. *Transportation Research Record*, 2672(49):101–112, 2018.
- [50] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.
- [51] Hossam Faris, Ibrahim Aljarah, and Seyedali Mirjalili. Evolving radial basis function networks using moth–flame optimizer. In *Handbook of neural computation*, pages 537–550. Elsevier, 2017.
- [52] Markus Törmä. Kohonen self-organizing feature map in pattern recognition. *Photogramm. J. Finland*, 15:1, 1995.
- [53] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [54] Muhammet Fatih Aslan, Kadir Sabanci, Akif Durdu, and Muhammed Fahri Unlarsen. Covid-19 diagnosis using state-of-the-art cnn architecture features and bayesian optimization. *Computers in Biology and Medicine*, page 105244, 2022.
- [55] Yusuf Satılmış, Furkan Tufan, Muhammed Şara, Münir Karşlı, Süleyman Eken, and Ahmet Sayar. Cnn based traffic sign recognition for mini autonomous vehicles. In *International Conference on Information Systems Architecture and Technology*, pages 85–94. Springer, 2018.
- [56] Andreas Kamilaris and Francesc X Prenafeta-Boldú. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3):312–322, 2018.

- [57] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):1–9, 2021.
- [58] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.
- [59] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.
- [60] Tarun Kumar and Karun Verma. A theory based on conversion of rgb image to gray image. *International Journal of Computer Applications*, 7(2):7–10, 2010.
- [61] Abderrahmane BENDJAAFER, Toufik MEDDAH, et al. *Classification d’image à l’aide d’un réseau Totalemment convolutionnelle*. PhD thesis, university of M’sila, 2021.
- [62] Qianru Zhang, Meng Zhang, Tinghuan Chen, Zhifei Sun, Yuzhe Ma, and Bei Yu. Recent advances in convolutional neural network acceleration. *Neurocomputing*, 323:37–51, 2019.
- [63] Shiqi Yu, Sen Jia, and Chunyan Xu. Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219:88–98, 2017.
- [64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [65] Asia Kausar, Mohsin Sharif, Jinhyuck Park, and Dong Ryeol Shin. Pure-cnn: A framework for fruit images classification. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 404–408. IEEE, 2018.
- [66] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.
- [67] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

- [68] Guilin Liu, Kevin J Shih, Ting-Chun Wang, Fitsum A Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. *arXiv preprint arXiv:1811.11718*, 2018.
- [69] Luiz Zaniolo and Oge Marques. On the use of variable stride in convolutional neural networks. *Multimedia Tools and Applications*, 79(19):13581–13598, 2020.
- [70] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.
- [71] Nassima Dif. *L'apprentissage profond pour le traitement des images*. PhD thesis, 12 2020.
- [72] Yann LeCun et al. Lenet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, 20(5):14, 2015.
- [73] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, 17(1):182–188, 2020.
- [74] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.
- [75] Navin Kumar Manaswi. Understanding and working with keras. In *Deep Learning with Applications Using Python*, pages 31–43. Springer, 2018.
- [76] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Regularization for deep learning. *Deep learning*, pages 216–261, 2016.
- [77] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
- [78] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

## Bibliography

---

- [79] Ramaprasad Poojary, Roma Raina, and Amit Kumar Mondal. Effect of data-augmentation on fine-tuned cnn model performance. *IAES International Journal of Artificial Intelligence*, 10(1):84, 2021.
- [80] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [81] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.
- [82] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020.
- [83] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.
- [84] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.
- [85] Pedro Marcelino. Transfer learning from pre-trained models. *Towards Data Science*, 10:23, 2018.
- [86] Chaoyun Zhang, Pan Zhou, Chenghua Li, and Lijun Liu. A convolutional neural network for leaves recognition using data augmentation. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2143–2150. IEEE, 2015.
- [87] Nitish Srivastava. Improving neural networks with dropout. *University of Toronto*, 182(566):7, 2013.
- [88] Jiangang Hao and Tin Kam Ho. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, 44(3):348–361, 2019.

## Bibliography

---

- [89] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020.
- [90] Paul Barrett, John Hunter, J Todd Miller, J-C Hsu, and Perry Greenfield. matplotlib—a portable python plotting package. In *Astronomical data analysis software and systems XIV*, volume 347, page 91, 2005.
- [91] Pratik Kanani and Mamta Padole. Deep learning to detect skin cancer using google colab. *International Journal of Engineering and Advanced Technology Regular Issue*, 8(6):2176–2183, 2019.
- [92] Intel image classification challenge. <https://www.kaggle.com/puneet6060/intel-image-classification>. Accessed: 15-04-2022.