MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABBES LAGHROUR KHENCHELA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département de Math et Informatique

N° de série :……

Mémoire de fin d'études

Pour l'obtention du diplôme de Master (L.M.D)

Spécialité: Informatique
Option: Génie Logiciel et Système Distribué

L'apprentissage automatique pour la reconnaissance et la lecture des panneaux d'information

Réalisé par : Nadia ABAIDIA

Membres de jury :
Dr. Adel ABDELHADI
Dr. Rafik MAHDAOUI

Dirigé par : Pr. Souad SAADI

Présenté le 26/06/2019

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH
UNIVERSITY ABBES LAGHROUR KHENCHELA
FACULT OF SCIENCE AND TECHNOLOGY

Department of Math and Computer Science

Serial number:……

Memory to obtain the diploma of Master's degree(L.M.D)

Specialty: Computer Science
Option: Software Engineering and Distributed System

Machine learning for recognition and reading of information signboards

Realized by: Nadia ABAIDIA

Jury members:
Dr. Adel ABDELHADI
Dr. Rafik MAHDAOUI

Directed by: Pr. Souad SAADI

Presented on 26/06/2019

Acknowledgements

This thesis is dedicated to the sake of Allah, my Creator and my Master, without him I wouldn't have done anything. He has given me the courage and the strength to continue. Thank you for enlightening the path of success for me.
I sincerely thank my supervisor Pr. Souad Saadi, this work would never have happened without her scientific contribution, her understanding, patience and her support to me.
I also thank the members of the jury who do us honor by agreeing to judge this work.
To my great family without exception (my dad, my mom, my brother and sisters, and all the others) who never stop giving of themselves in countless ways.
I do not forget to also to thank Pr. Fayçal Abbas for his help and his guidance.
Finally, I thank all the friends, and those who have contributed from near or far to this modest work to see the light of day.

Dedication

To Those I Love.

Abstract

In this day, there are many information signboards recognition systems, these systems have two major axes, which are the detection of the signboard and the recognition of its words. However, these systems are controlled by various conditions. So this memory proposes a new signboards recognition method based on a new machine learning approach which is transfer learning.

Our method consists of three major steps, the first one is the pretreatment step, which includes a step of text region detection, this step formed the part of the transfer learning from a convolutional neuron network (CNN). The second is using one of the famous text recognition engines to extract the words from the signboard this step will be based on Long Short-Term Memory neural network (LSTM). The last one is to transform the written text to voice. Our method gives us a satisfying result.

Key words: Detection, Recognition, Machine Learning, CNN, LSTM, Transfer Learning.

Résumé

De nos jours il existe de nombreux systèmes de reconnaissance des plaques d'information, ces systèmes sont composés par deux axes majeurs, qui sont la détection de la plaque d'information et la reconnaissance de mots. Cependant ces systèmes sont contrôlés par plusieurs conditions. Par conséquent ce mémoire propose une méthode de reconnaissance des plaques d'information basé sur une nouvelle approche d'apprentissage automatique qui s'appelle l'apprentissage par transfert.

Notre méthode est constituée de trois étapes majeures, qui sont la phase de prétraitements, qui inclue une étape de la détection de la région de texte, cette étape constitue la partie de transfert d'apprentissage à partir de réseau de neurone convolution (CNN). Le deuxième axe constitue à utiliser un moteur de reconnaissance de texte pour extraire les mots du panneau. Cette étape sera basée sur le réseau de neurones LSTM. La dernière étape consiste à transformer le texte écrit en audio. Les résultats obtenus par notre méthode sont satisfaisants.

Mots clés: La Détection, La Reconnaissance, l'Apprentissage Automatique. CNN, LSTM, Le Transfert d'Apprentissage.

ملخص

في الوقت الحاضر ، هناك العديد من أنظمة التعرف على لوحات المعلومات ، وهذه الأنظمة لها محورين رئيسيين ، هما اكتشاف اللوحة ثم التعرف على الكلمات التي تحتويها. و يتم التحكم في هذه الأنظمة من خلال ظروف مختلفة. لذلك تقترح هذه المذكرة طريقة جديدة للتعرف على لوحة المعلومات استنادًا إلى نهج جديد من التعلم الآلي وهو نقل التعلم.

تتكون طريقتنا من ثلاث خطوات رئيسية ، المرحلة الاولى ما قبل المعالجة ، والتي تتضمن خطوة لاكتشاف منطقة النص ، تشكل هذه الخطوة الجزء الخاص بنقل التعلم من شبكة الخلايا العصبية التلافيفية (CNN) .المرحلة الثانية هي استخدام محرك التعرف على النص لاستخراج الكلمات من اللوحة. تعتمد هذه الخطوة على الشبكة العصبيةLSTM .في آخر مرحلة نقوم بتحويل النص المكتوب إلى صوت. طريقتنا أعطت نتائج جد مرضية .

الكلمات المفتاحية: الاكتشاف, الاعتراف, والتعلم الآلي, CNN, LSTM, نقل التعلم.

Contents

List of Figures

General Introduction

Visibility is one of the most remarkable processes. it's the main sense which the brain is depends on it to recognize and understand also analyze the external environment. But unfortunately, there are 39 million blind people in the world and 285 million people are visually impaired according to World Health Organization. This number is increasing day-by-day due to diabetes, accidents, aging population and other reasons.

Blind or visually impaired people need assistance from another human being to perform simple tasks such as reading label of food packets or information signboards. In simple words, most of them lose confidence in living an independent life.

In order to compensate a part of this loss we will present a new method which is a combination of existing models of detection and recognition to extract text and words from information signboards images.

This thesis is organized in five chapters:

- **In the first chapter**, we will present the principle of the artificial intelligence learning, for this we defined the basis associated with learning, as well as the learning characteristics and also the different classifications of machine learning.
- **In the second chapter**,we will present some learning approaches, which are best known in the field of AI, and in the last part of this chapter we will discuss the two neural networks convolutional and long short term memory, that we will use through our work.
- **In the third chapter**,we will present some previous approaches and models, which they are trying to address the same subject like us, then we will allocate some space to talk about the model that we used in our work.
- **In the fourth chapter**, we will show the implementation part of our work.
- **In the fifth chapter**,we will final our work by discussing the different results obtained, and we will proposed some perspectives to doing in a future work.

Chapter 1

Machine Learning

Introdution

Artificial intelligence (AI) is one of the most recent domains of science and engineering. AI is currently including a wide variety of extensions one of them is machine learning. This one aims to provide a system that is optimized according to the environment [1].

1 Artificial Intelligence

1.1 Definition

Artificial intelligence (AI) is a new technology in full development, it's aims to gives machines the ability to acquire and retain knowledge, to learn or understand through experience. This technology was born in 1950 with Alan Turing who is the author of the famous Test of Turing (first conversation between human and Artificial Intelligence) [2].

1.2 History of AI

Artificial Intelligence (AI) is a field that has a long history and it's always evolving. We chose the shorter ones as following:

- 1943: McCulloch and Pitts create the model of the formal neuron.
- 1949: Hebb establishes the first rule of neural learning.
- 1950: Shannon, 1952 Samuel, 1953 Turing: Machine to play chess.
- From 1955 to 1956, Allen Bewell, John Shaw and Herbert Simon produced the first artificial intelligence program called Logic Theorist. This program is makes it possible to demonstrate 38 of the 52 demonstrations of the textbook of the time called "Principa Mathematica".
- 1960 John McCarthy, Allen Bewell and Simon Herbert: The computer can be used for anything other than calculations "manipulating symbols" (this idea proposed by Ada Lovelage friend of Babbage 1842).
- 1969-1979: expert systems.
- Since 1986: return of the neuron networks [3].

1.3 Fields of Artificial Intelligence

The human has created artificial intelligence in order to be able to put our own intelligence abilities into machines in order to be similar to his behavior. but creating intelligent agents is not so simple. For this reason, AI was divided into many subfields, each one has trying to handling part of the problem [4]. The most important subfields were:

1.3.1 Natural Language Processing

In Artificial Intelligence, Natural Language Processing is a discipline that aims to model, through computer science, the language that is written or spoken [4].

1.3.2 Expert Systems

An expert system is a software capable of answering questions, reasoning from known facts and rules. It can be used as a tool for decision support [4].

1.3.3 Games Problem Solving

Representation, analysis and resolution of concrete problems. This is the case of games of reflection such as chess, backgammon or the ladies. In the case of backgammon the world champion is being a program [4].

1.3.4 Artificial Vision

The goal of this discipline is to allow computers to understand images and video (for example, to recognize road signs or numbers) and to try to imitate the human or animal vision [4].

1.3.5 Robotics

This discipline aims to achieve physical agents, to allow to this agent to imitate human behaviors[4]. Look to the figure1.1. quote

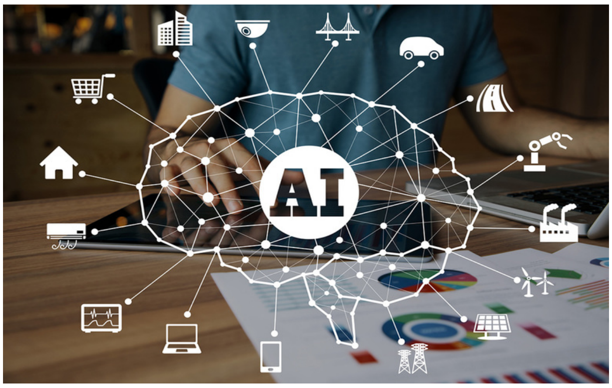


Figure 1.1: The Several Use of Artificial Intelligence.

2 The Learning

2.1 Machine Learning

2.1.1 Definition

Machine learning is a technique allows computers to learn without being explicitly programmed and use existing data to predict future trends, results, and behaviors [5].

2.1.2 The Goal of Machine Learning

Machine learning aims to uses computers to simulate human learning and enables to computers to identify and acquire the real world knowledge and to improve the performance of certain tasks based on this new knowledge [6].

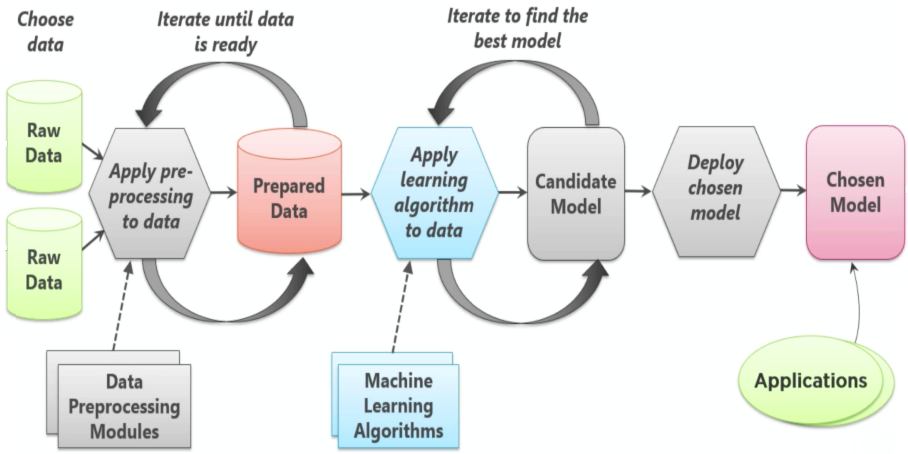


Figure 1.2: The Process of Machine Learning .

2.1.3 Application Fields of Machine Learning in AI

Machine learning applies to a large number of human activities. For example:

- Analysis and automatic processing of images.
- Search for information (internet engine ...).
- Discover the relationships between data in large databases.
- Establish a medical diagnosis from the clinical description of a patient.
- Give an answer to the request for a bank loan from a client on the basis of his personal situation.

2.1.4 Different Types of Machine Learning

Machine learning techniques are classified into four main categories:

A Supervised Learning

Supervised learning approaches use a set of sample data to construct a model, where each sample data has a specific class label, and the system learns to classify according to this ranking model in order to deduce the best possible separations so that it can identify new examples with a great precision. The goal of this paradigm is to determine typical rules that can be used to differentiate between data of different classes [7].

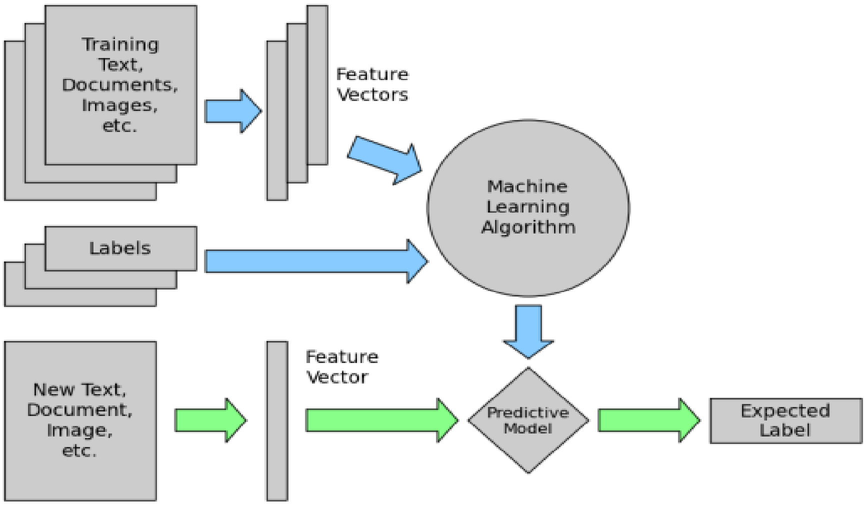


Figure 1.3: Supervised Learning Model.

B Unsupervised learning

It is mainly used in pattern detection and descriptive modeling. However, there are no output categories or labels based on which the algorithm can try to model relationships here. This paradigm tries to use special techniques at the input data to extract rules, detect patterns..., which help to describe the data to the users in better way [8].

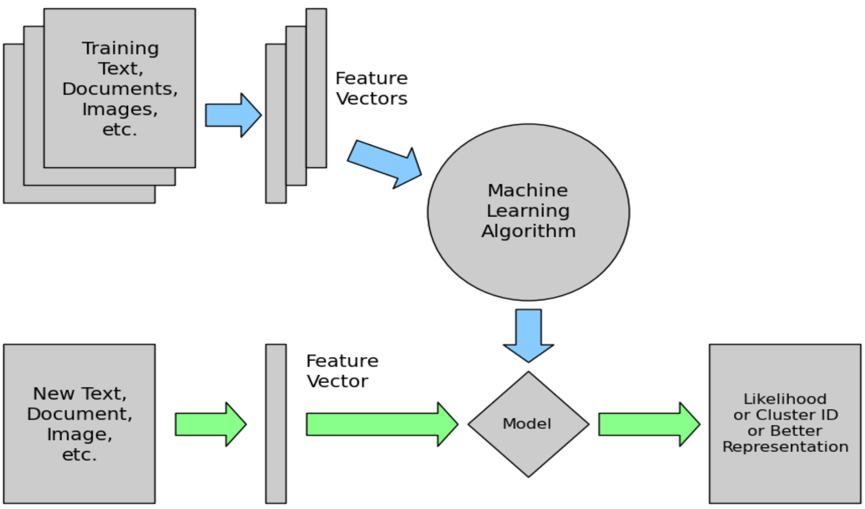


Figure 1.4: Unsupervised Learning Model.

C Semi-Supervised Learning

Semi-supervised learning falls between these two previous types. The input data consists of labeled and unlabeled examples. because it allows you to use all the information, especially if this data carries important information [8].

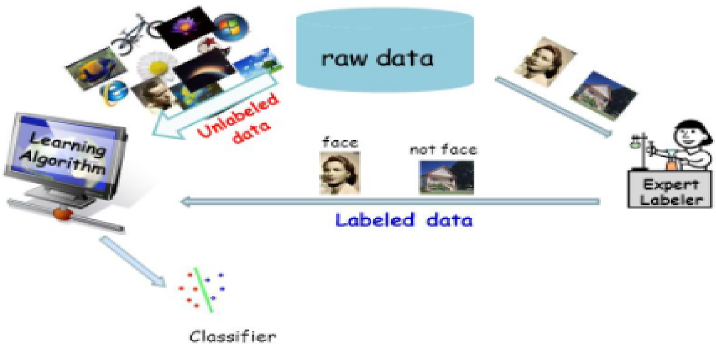


Figure 1.5: Semi-Supervised Learning.

D Reinforcement Learning

Reinforcement learning algorithm (called the agent) continuously learns from the environment in an iterative fashion. In the process, the agent learns from its experiences of the environment until it explores the full range of possible states. Some applications of the reinforcement learning algorithms are computer played board games (Chess, Go), robotic hands, and self-driving cars [8].

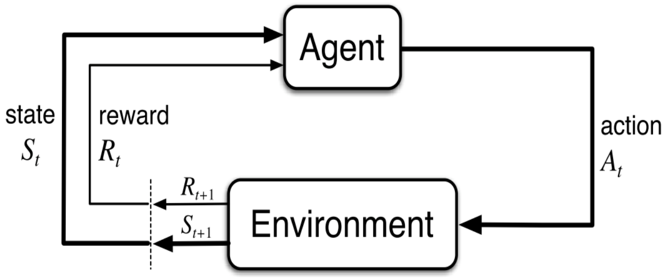


Figure 1.6: The Process of Reinforcement Learning.

2.1.5 The Difference Between Machine Learning Types

The table below explain the difference between machine learning types:

Supervised	**Unsupervised**	**Semi-supervised**	**Reinforcement**
•Data has known labels or output.	•Labels or output unknown. •Focus on finding patterns and gaining insight from the data.	•Labels or output known for a subset of data. •A blend of supervised and unsupervised learning.	•Focus on making decisions based on previous experience.

Table 1.1: Styles of Learning.

2.2 Deep Learning

2.2.1 Definition

Deep learning is a subset of machine learning, it is using a family of neural network methods to teaches a computer model how to perform classification tasks directly from images, text or audio. Deep learning models can achieve an exceptional level of accuracy, sometimes surpassing human performance [9].

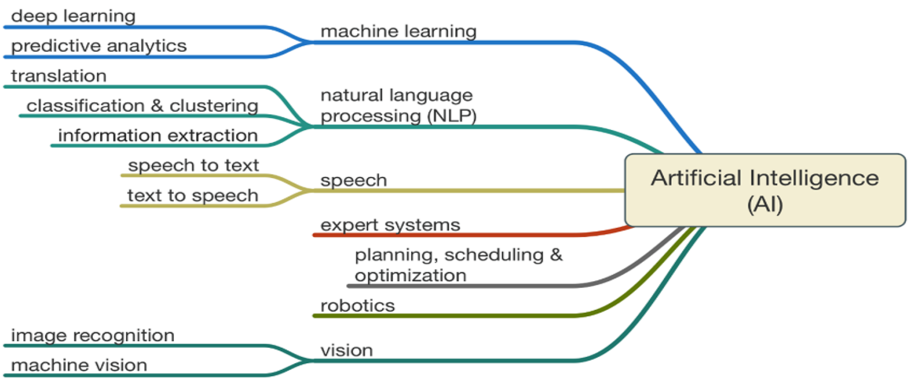


Figure 1.7: Schematic Representation of The Artificial Intelligence.

2.2.2 Difference Between Artificial Intelligence and Machine Learning and Deep Learning

- Artificial intelligence is a technique that allows computers to imitate human behavior.
- Machine learning is a subset of artificial intelligence techniques that use statistical methods to enable machines to improve with experiments.
- Deep learning is a subset of machine learning that calculates multilayer neural networks.

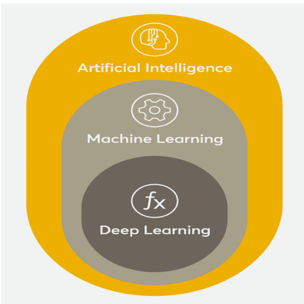


Figure 1.8: Difference Between AI and ML and DL.

2.2.3 Examples of Deep Learning Application

Millions of people are already benefiting from the progress made through deep learning. For example:

- Search engines, for example: Google, Yahoo...
- Self-driving cars.
- Translation.
- In the field of medicine, for example: finding cancer cells...
- Robotics.

2.2.4 Some Deep Learning Algorithms

There are different algorithms of deep learning. We can mention :

A Long Short-Term Memory

LSTM network has the following three aspects that differentiate it from an usual neuron in a recurrent neural network:

- It has control on deciding when to let the input enter the neuron.
- It has control on deciding when to remember what was computed in the previous time step.
- It has control on deciding when to let the output pass on to the next time stamp.

The beauty of the LSTM is that it decides all this based on the current input itself. LSTM is inspired by how our brains work and can handle sudden context switches based on the input [10].

B Convolutional Neural Networks

Convolutional neural networks (CNNs) are to date the most efficient models for classifying images. The CNN problem is divided into subparts, and for each part, a "group" of neurons will be created to study that specific part. For example, for a color image, it is possible to divide the image into width, height and depth (colors) [10].

C Deep Boltzmann Machine

In machine learning, Deep Boltzmann machine is a type of artificial neural network for unsupervised learning. It is commonly used to estimate the probabilistic distribution of a dataset. It was originally invented as Harmonium in 1986 by Paul Smolenski [10].

2.3 Learning by Transfer

Transfer learning is a machine learning technique. It is differing from traditional Machine Learning in that it is the use of pre-trained models that have been used for another task to jump start the development process on a new task or problem [11].

Transfer learning is a popular approach in deep learning. It's only works in deep learning if the model features learned from the first task are general [12].

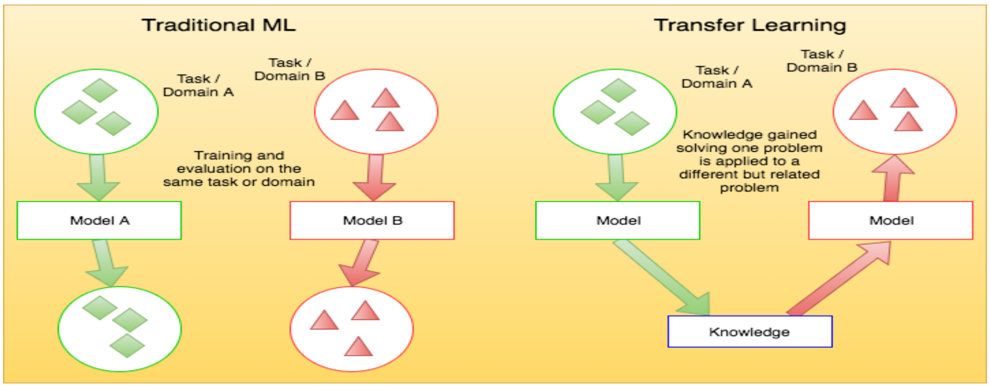


Figure 1.9: Learning by Transfer and Traditional ML.

Conclusion

In this chapter, we presented the definition of artificial intelligence and some historical and learning in several types. The next chapter we will talk about one of the artificial intelligence important approaches which is the neural networks.

Chapter 2

The Artificial Neural Networks

Introdution

Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

1 Definition of Artificial Neural Networks

Artificial neural networks are highly connected networks of elementary processors operating in parallel. Each elementary processor calculates a single output based on the information it receives [13].

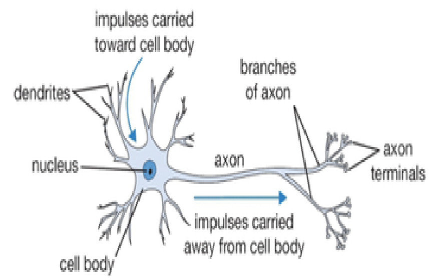
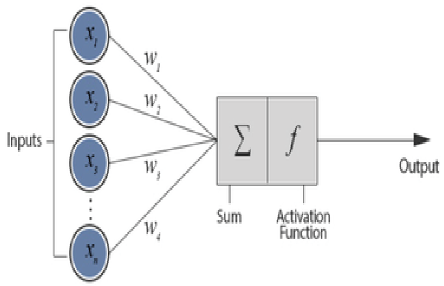
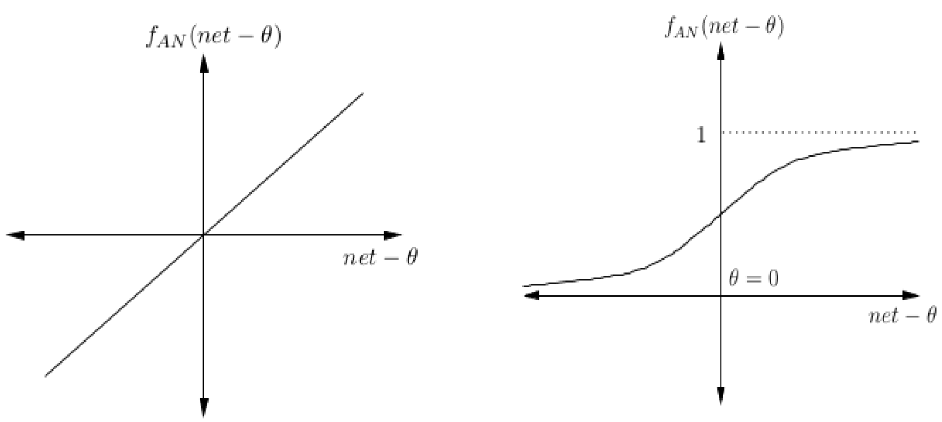
<table>
<tr><th>The biological neuron</th><th>The formal neuron</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Neuron is a cell that is the basis of the functioning of the nervous system. The neuron is mainly composed of three parts:
• The cell body.
• The axons that pass the information.
• The dendrites that receive it from other neurons.</td><td>The formal neuron is a model characterized by an internal state $s \in S$,input signals $x_1, x_2, ..., x_p$ and a state transition function [14]
$$s = h(x_1, ..., x_p) = f(\beta_0 + \sum_{j=1}^{p} \beta_0 x_j)$$</td></tr>
</table>

Table 2.1: The biological and formal neuron.

2 The Transition Function of Neural Networks

The transition function operates a transformation of an affine combination of the input signals,β_0 being called through the neuron. This affine combination is determined by a weight vector $[\beta_0, \beta_1, ..., \beta_p]$ associated with each neuron and whose values are estimated in the learning phase. They constitute "memory" or "distributed knowledge" of the network.

The different types of neurons are distinguished by the nature of their transition function [14].in below the main types:



(a) Linear function (d) Sigmoid function

Figure 2.1: The Main Activation Functions.

3 The Different Types of Neural Networks

The neural networks are classified according to the nature of their learning algorithm in two main types:

3.1 Feedforward Networks "not looped"

(perceptron multilayer, perceptron...) the data flows from the entry to the exit through the links without possibility of return using these same links.

3.1.1 Simple Perceptron

The perceptron was introduced in 1958 by Franck Rosenblatt, In its simplest version, the perceptron has only one output y to which all inputs xi are connected, the perceptron inputs and outputs are Boolean [15].

3.1.2 Perceptron Multilayer

A multilayer perceptron consists of three types of layers [16]:

- An input layer that matches the input data $x = [x_1, ..., x_n]^T$. This layer does not contain neurons.
- An output layer consisting of K neurons and producing the outputs of the network $y = [y_1, ..., y_n]^T$.
- Hidden layers each consisting of several neurons. These layers allow the non-linear transformation of the input signal to the output signal.

In the multilayer perceptron, all the neurons of a layer are connected to the neurons of the previous layer.

3.2 Recurrent Networks

(Hopfield networks, Kohonen networks...) with the existence of returns between entry and exit.

3.2.1 Hopfield Networks(HNN)

Hopfield networks are networks in which all neurons are interconnected. Here, there is only one layer and the output layer is assimilated to the input layer. In addition, each neuron has a binary activation value (0/1 or -1/1). The set of activation values of the neurons of a hopfield network represents its state and its output [17] .

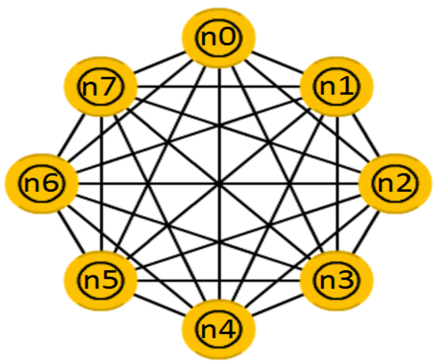


Figure 2.2: Hopfield Network Containing 7 Neurons.

3.2.2 Kohonen Networks

The Kohonen card is a self-organizing card in two layers, one input layer and another output layer. The output layer also called competitive layer is in two dimensions. Each input neuron is connected to the set of output neurons by weights $W_{i,j}$ [18].

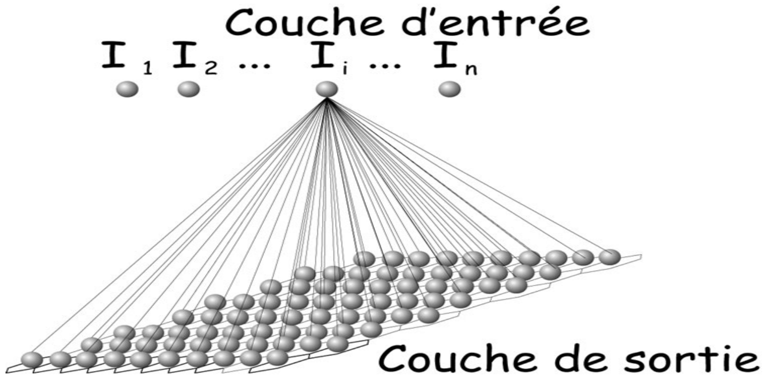


Figure 2.3: Kohonen Networks model.

4 The Learning of Neural Networks

The purpose of learning is to provide a method for the network so that it can adjust its parameters when presenting examples to it. Indeed, neural networks have been successfully applied to the learning of classification tasks. Learning is the process of adapting the parameters of a system to give a desired response to any input or stimulation.

5 Convolution Neural Networks(CNN)

Convolutional neural networks or CNNs are an extension of Perceptron Multilayer. They are designed to automatically extract the characteristics of input images. The first Convolutional Neural Network (CNN) was introduced in the late 1980s by LeCun 1989. It is the first neural network for image recognition. This network allowed the recognition of handwritten figures [19].

5.1 The different layers of a convolutional neural network

We present in this part the different modules used in CNN:

5.1.1 The convolutional layer

The convolution layer is the key component of the CNN, which applies a specified number of convolution filters to the input image. This layer is used to perform a set of mathematical operations to produce a unique value at the output. The resulting output is called a feature map. The following formula explains how to calculate the output of a given neuron in a convolution layer [20].

$$g_{i,j,k} = \sum_{u=1}^{f_h} \sum_{v=1}^{f_w} \sum_{k'=1}^{f_{n'}} x_{i',j',k'} * w_{u,v,k',k} \quad with \begin{cases} i' = u * s_h + f_h - 1 \\ j' = v * s_w + f_w - 1 \end{cases}$$

- $g_{i,j,k}$corresponds with the output of the neuron located in line i and in column j in the characteristic map k of the convolution layer (layer l).

- s_h and and s_w are the vertical and horizontal steps, f_h and f_w are the height and width of the receiver field, and $f_{n'}$ are the numbers of feature maps in the previous layer (layer $l-1$).
- $x_{i',j',k'}$ corresponds with the output of the neuron located in the layer $l-1$, line i' , column j' , characteristic map k' (or channel k' if the previous layer is the layer of Entrance).
- b_k is the constant term of the characteristic map k (in layer l). it can be seen as an adjustment of the overall brightness of the k feature map.
- $w_{u,v,k',k}$ is the weight of the connection between any neuron in the feature map k of layer l and its input in line u and column v (relative to the neuron receiver field) in the map of characteristics k'.

Example of calculates:
$1*1+1*0+1*1+0*0+1*1+1*0+0*1+0*0+1*1 = 4$

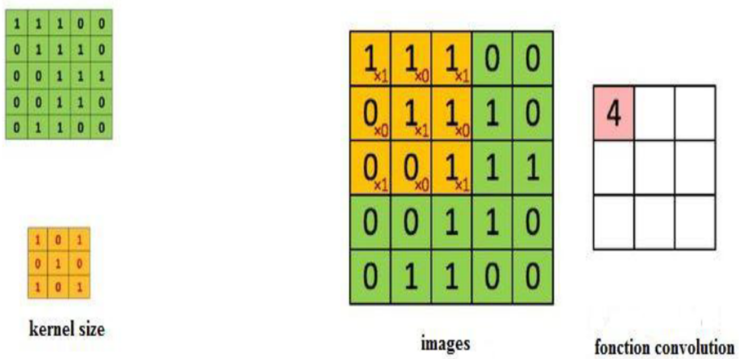


Figure 2.4: An Illustration Of A Convolution Layer.

5.1.2 The pooling layer

The pooling layer (or agglomeration) makes it possible to add spatial invariance when extracting characteristics while reducing the size of the inputs. It can be of different natures but the most used types of pooling are the Max Pooling and the Average pooling. Max Pooling returns the maximum element on a calculation window. Average pooling makes it possible to return the average of the elements on a calculation window [19].

5.1.3 The Maxpooling layer

The Maxpooling layer is generally used after the convolutional layer. It is used to reduce the dimensions of an image in order to reduce the calculation time and minimize the occupied memory space. The formula below illustrates the calculation of the size of Maxpooling [21].

$$W_2 = \frac{W_1 - F}{S} + 1$$
$$H_2 = \frac{H_1 - F}{S} + 1$$

- **W1, H1 :** the size of the input volume.
- **F :** The spatial size of the output volume.
- **S :** the step.
- **W2, H2 :** the size of the output volume.

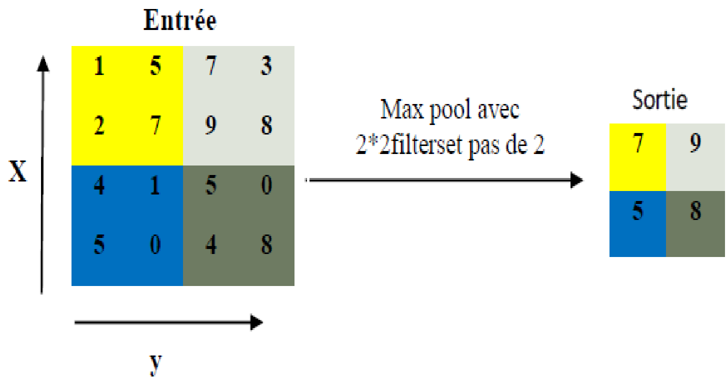


Figure 2.5: An Illustration Of A Maxpooling Layer.

5.1.4 The Activation Functions

There are different activation functions. Among the best known:

A ReLU Activation Function

The activation function (ReLU) is used almost in all convolutional neural networks and in deep learning methods. It is an elementary function that is generally performed according to two cases: the first case; the function is disabled if the inputs are negative (the output is zero), the second case where the inputs are positive, so the output is the same as the input. It is used to gain the non-linearity of the network. As illustrated by the following formula [22]:

$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases}$$

ReLU function.
f : The activation function and x: the value of the neuron.

B Sigmoid Activation Function

The sigmoid activation function is the most appropriate choice. It is particularly used for models, where output possibilities must be provided.

In our work we have chosen this function to ensure the correct detection (text, not text), its property is: for all real $x, y \in R : 0 < y < 1$, such that y is the output of the sigmoid function [23].

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

Sigmoid function.
f: The activation function and x: the value of the neuron.

C Softmax Activation Function

Softmax is another kind of reactivation function and it's different one, which normalizes a set of pre-activations so that each one can be interpreted as a probability.

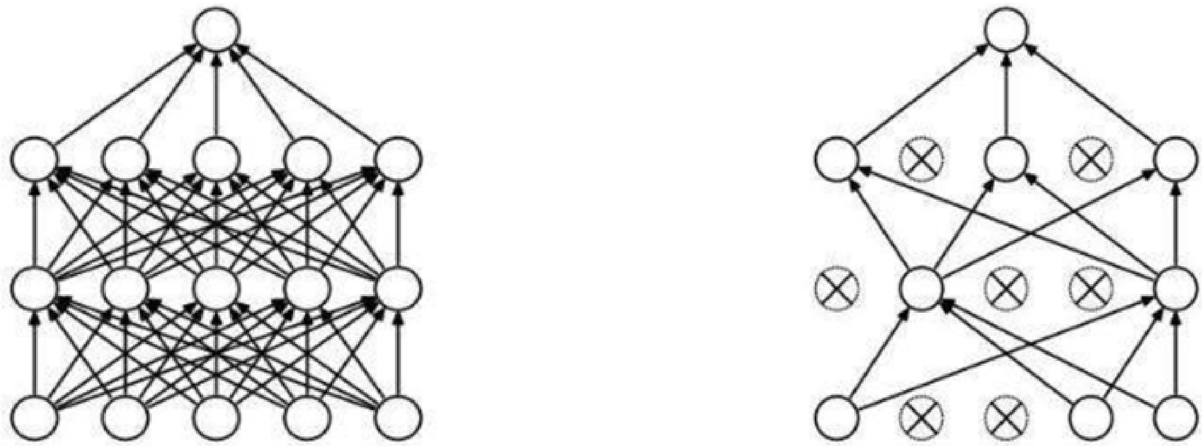
Usually, If there are only two classes, only one Sigmoid unit is used. However, if there are more than two classes, the Softmax is used. Considering x as a vector of pre-activations C, Softmax on these classes C is defined as follows [24].

$$f(x) = \frac{e^x}{\sum_{i=1}^{c} e^i}$$

Softmax function.

5.1.5 The Dropout Layer

The dropout layer was introduced by Srivastava in 2014. It is introduced to avoid over-learning. This layer is used during learning. It allows to randomly disable neurons during the different iterations of learning. In other words, dropout allows the network to learn subnets with fewer parameters and therefore less subject to over-learning. This way of doing things allows you to learn more generic parameters that do not focus on details of the learning base. Once the learning is complete, all the neurons are reactivated [19].



(a) standard neural network. (b) after the application of dropout.

Figure 2.6: A Neural Network Before and After Dropout Application.

5.1.6 Fully Connected Layer (FC)

After several layers of convolution and max-pooling, the high-level reasoning in the neural network is via fully connected layers. Neurons in a fully connected layer have connections to all the outputs of the previous layer. Their activation functions can therefore be calculated with a matrix multiplication followed by a polarization shift [19].

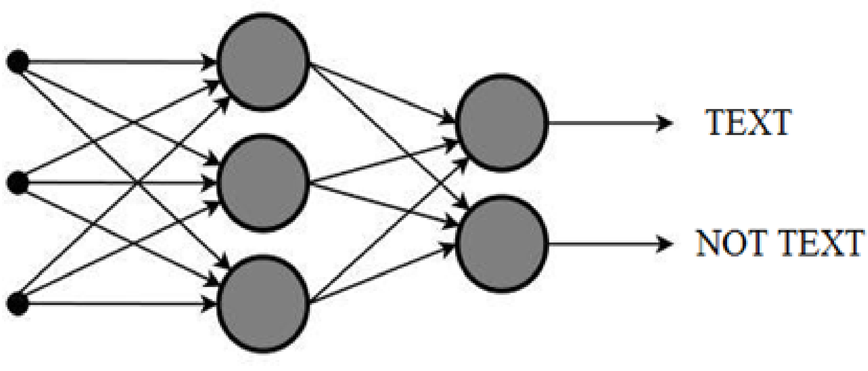


Figure 2.7: Architecture of the Fully Connected Layer.

6 Long Short-Term Memory Neural Network(LSTM)

Long Short-Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter and Schmidhuber (1997), they work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! [25].

6.1 general architecture of LSTM

LSTMs have four neural network layers interacting in a very special way [25]:

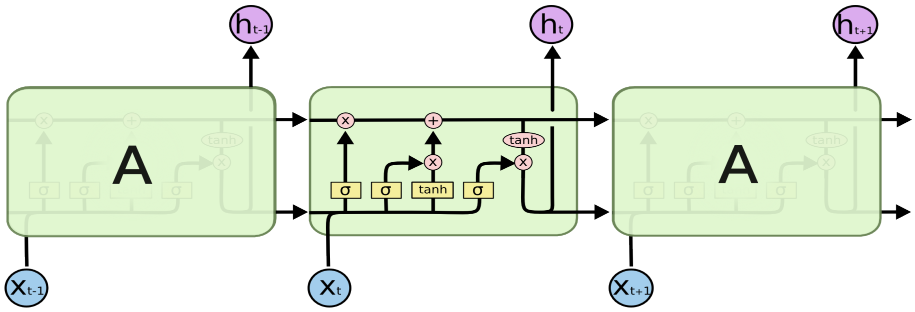


Figure 2.8: Chain of LSTMs Contains Four Interacting Layers.

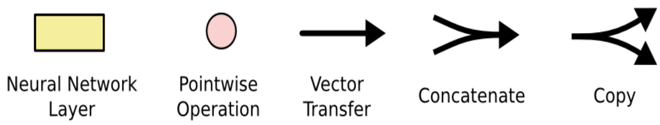
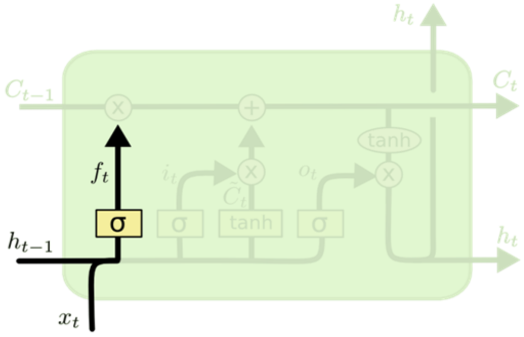


Figure 2.9: The Notation of The LSTM Architecture.

In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

6.1.1 Forget Gate Layer

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer" It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. the 1 represents "completely keep this" while a 0 represents "completely get rid of this" [25].

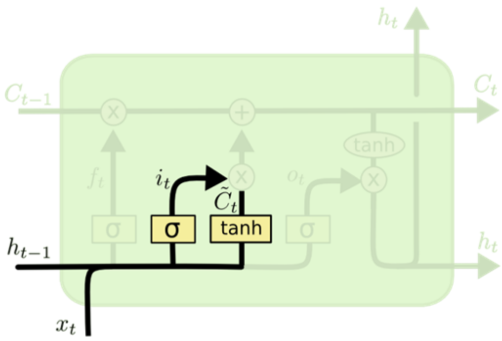


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 2.10: Forget Gate Architecture.

6.1.2 Input Gate Layer

The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a **tanh** layer creates a vector of new candidate values, ˜C_t, that could be added to the state. In the next step, we'll combine these two to create an update to the state [25].



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 2.11: Input Gate Architecture.

6.1.3 Update the Cell State

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it. We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \backsim C_t$. This is the new candidate values, scaled by how much we decided to update each state value [25].

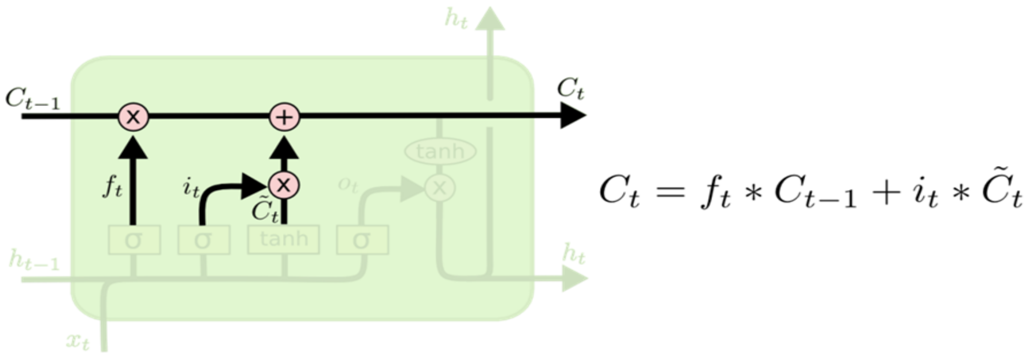


Figure 2.12: Cell State Architecture.

6.1.4 output Gate Layer

Finally, we need to decide what we are going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we are going to output. Then, we put the cell state through **tanh**(to push the values to be between ±1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to [25].

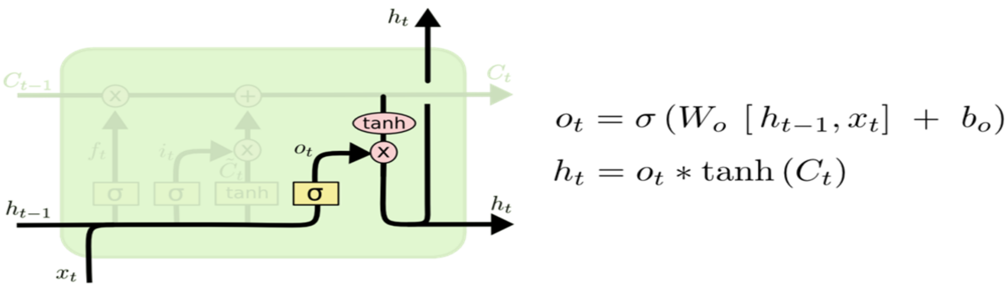


Figure 2.13: Output Gate Architecture.

LSTMs are working really a lot better for most tasks, they were a big step in what we can accomplish with RNNs.

Conclusion

We presented in this chapter the definition and the different types of artificial neural network. Then we have devoted the last part in the chapter to the tow type that we will use in our work, which is convolution neural networks(CNN) and Long Short-Term Memory neural networks(LSTM).

In the next chapter, we will talk about several approaches and models, which they are working on a same subject like as, after this we will focus on the model that we will used in our work.

Chapter 3

Detection and Recognition of Signboards

Introduction

Signboards is the design or use of signs and symbols or words to communicate a message to a specific group, usually for the purpose of marketing or a kind of advocacy. The term signboards is documented to have been popularized in 1975 to 1980 [26].

1 Definition of Signboard

Signboards is a plate of wood or metal which has been painted with pictures or words. The purpose from it is to give information about a particular place, product, or event. Newer signboards may also use digital or electronic displays [27].

2 The Purpose of Signboards

Typically, signboards tend to serve a few common purposes: to promote, identify, provide information, give directions or to raise safety awareness. For more expansion [28][29]:

- Advertising and Marketing
- Recognition
- Directional
- Health, Safety and Regulatory
- Identification
- Appearance

3 Detection and Recognition of Signboards

A lot of models and approaches for text detection and recognition has appeared in the last few years. Every one of them has some advantages and disadvantages. We will explain briefly some of the popular models in the field:

3.1 Text Detection Technics

3.1.1 MSER (Maximally Stable Extremal Regions)

In computer vision, maximally stable extremal regions (MSER) are used as a method of blob detection in images. This technique was proposed by Matas et al [30]. to find correspondences between image elements from two images with different viewpoints. This method has led to better object recognition algorithms.

The MSER algorithm has been used in text detection by Chen by combining MSER with Canny edges. Canny edges are used to help cope with the weakness of MSER to blur. MSER is applied to the image in question to determine the character regions [31].

Figure 3.1: Example of MSER Text Detection.

3.1.2 SWT (Stroke Width Transform)

Stroke Width Transform is an image processing technique for text detection in natural images. SWT present a novel image operator that seeks to find the value of stroke width for each image pixel, and demonstrate its use on the task of text detection in natural images. The suggested operator is local and data dependent, which makes it fast and robust enough to eliminate the need for multi-scale computation or scanning windows. Extensive testing shows that the suggested scheme outperforms the latest published algorithms. Its simplicity allows the algorithm to detect texts in many fonts and languages [32].

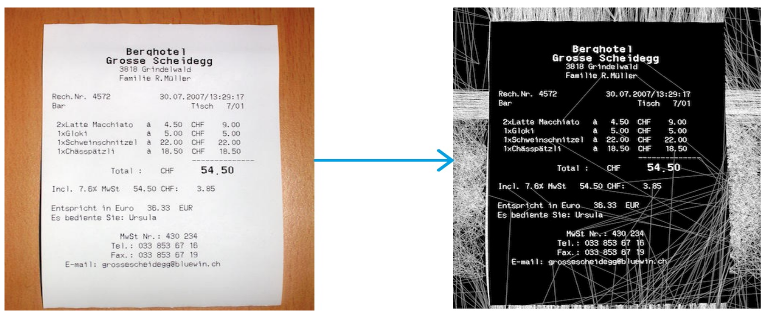


Figure 3.2: Example of SWT Text Detection.

3.1.3 EAST (Efficient and Accurate Scene Text Detector)

EAST is a pre-trained neural network model. It can be decomposed in to three parts: feature extractor stem, feature-merging branch and output layer. The figure bellow explains the structure of the model:

The feature extractor is a convolutional network pre-trained on dataset, with interleaving convolution and pooling layers. In the feature-merging branch, we gradually merge them, in each merging stage, the feature map from the last stage is first fed to an unpooling layer to double its size, and then concatenated with the current feature map. finally, we will have:

- The first output gives us the probability of a region containing text or not.
- The second output is the feature map that represents the "geometry" of the image [33].

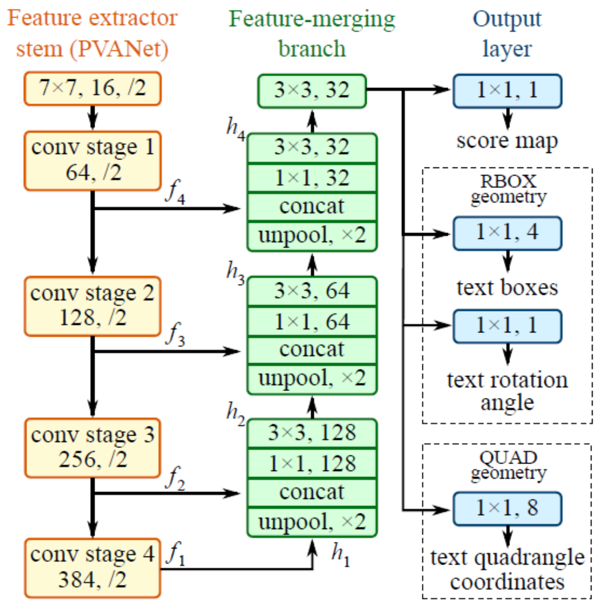


Figure 3.3: The Structure of The EAST Text detection.

3.2 Text Recognition Technics

3.2.1 Tree-Structured Character Recognition Model

Scene text recognition has inspired great interests from the computer vision community in recent years. Tree-Structured Model propose a novel scene text recognition method using part-based tree-structured character detection. Tree-structure is used to model each type of character so as to detect and recognize the characters at the same time. While for word recognition, and the final word recognition result is obtained by minimizing the cost function. Experimental results on a range of challenging public datasets (ICDAR 2003, ICDAR 2011, SVT) demonstrate that the Tree-Structured Model outperforms other methods significantly both for character detection and word recognition [34].

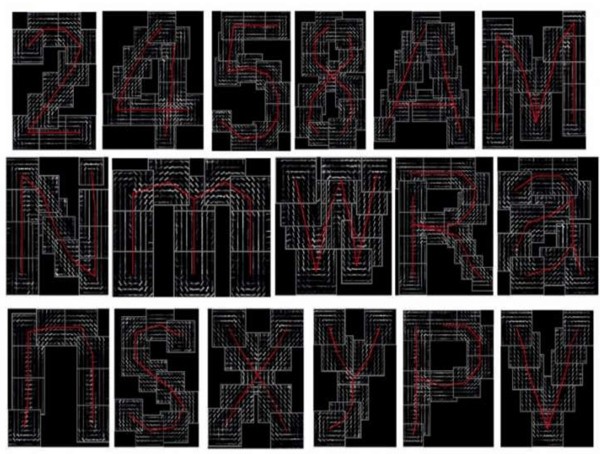


Figure 3.4: Example of Tree-Structured Character Recognition Model.

3.2.2 Focusing Attention Towards Accurate Text Recognition

Scene text recognition has been a hot research topic in computer vision due to its various applications. The FAN (Focusing Attention Network) method is employs a focusing attention mechanism to automatically draw back the drifted attention. FAN consists of two major components: an attention network (AN) that is responsible for recognizing character targets as in the existing methods, and a focusing network (FN) that is responsible for adjusting attention by evaluating whether AN pays attention properly on the target areas in the images. Extensive experiments on various benchmarks, including the IIIT5k, SVT and ICDAR datasets, show that the FAN method substantially outperforms the existing methods [35].

3.2.3 Tesseract OCR (Optical Character Recognition) Engine

Tesseract, a highly popular OCR engine, it is the technology that enables computers to extract text data from images by using Long Short-Term Memory (LSTM) neural networks. Tesseract was originally developed by Hewlett Packard in the 1980s. We can access Tesseract via the Python programming language by using the pytesseract library.

Tesseract can recognize more than 100 languages. It's probably powers many of the systems in services that we use daily. Some of the applications of OCR include automatic data entry for business documents, translation apps, online databases like Google Books, security cameras that automatically recognize license plates, and more [36].

4 The Chosen Models

In our work, we created a new method which is a combination of the EAST model and the Tesseract OCR engine, we have chosen this two models because they are simpler and more effective, also they have less error rate than the others. We will use this combination to extract text and words from information signboards images, then we will convert the results we have get to voice.

Conclusion

In this chapter, we briefly presented the definition and the purpose of signboard. Then we showed some popular models that is dealing with the same subject that we interested in.

The next chapter we will show the implementation of our work and discussing some of the limitations and drawbacks of our method.

Chapter 4

Implementation

Introduction

In this chapter, we will define the architecture of our method that we proposed to detect and recognize text in natural scene images, then we will apply this method on dataset of signboard images to test the final performance. For this, we will work with python programming language and we use some libraries, like Open CV, Tensorflow and others, to make our work more flexible.

1 Configuration Used in The Implementation

We have implemented our method in machine which has the following feature:

- DELL laptop with an Intel Core i5-4210U 1.70 GHz processor.
- 8 GB of RAM and NVIDIA GeForce 820M.
- The operating system was Windows 7.

2 Software and Libraries Used in The Implementation

2.1 Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics...), that aims to simplify package management and deployment. Package versions are managed by the package management system conda [37].

2.2 Python

Python is a powerful, high-level, easy-to-learn programming language that supports multiple programming models (procedural, functional, and object-oriented). Python libraries (packages) encourage modularity and reusability of existing codes. It is considered to be one of the most widely used programming languages in the world [38].

2.3 Spyder

Spyder is a powerful interactive development environment for the Python language, with advanced features, interactive testing, debugging, and introspection [39].

2.4 OpenCV

OpenCV (Open Computer Vision) is a free graphics library, originally developed by Intel, specializing in real-time image processing. The OpenCV library provides many features very diverse to create programs from raw data to the creation of GUIs basic [40].

2.5 Tensorflow

TensorFlow is an open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications [41].

3 The General Architecture of Our Method

In this part we will review our method architecture and we will explained each step that we had taken:

1. First, we will perform a step of text detection using EAST a highly accurate deep learning text detector, used to detect text in natural scene images.
2. Once we have detected the text regions of interest, we'll pass them into Tesseract OCR Engine.
3. Finally, when Tesseract OCR Engine give us the final results in the form of a text, we will take the last step by transforming the text to a voice.

The next figure we will illustrated our method architecture:

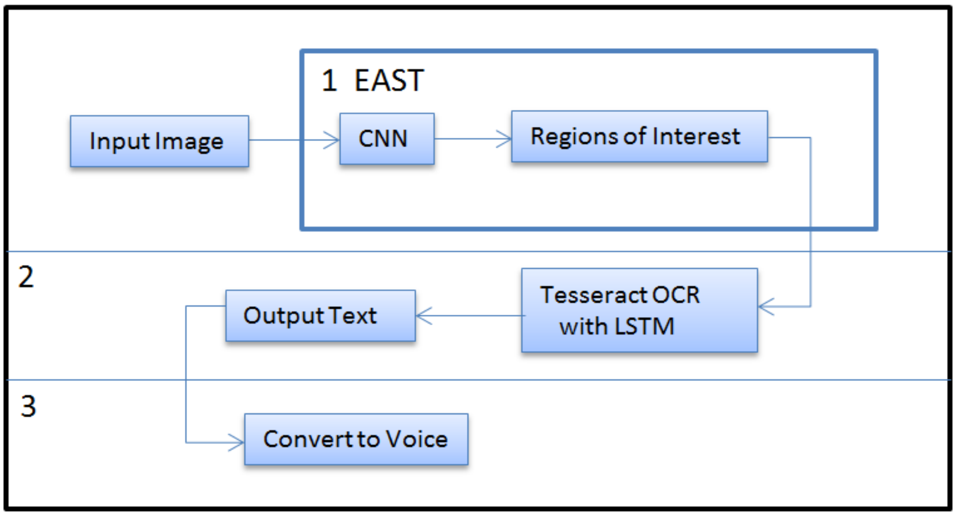


Figure 4.1: General Architecture.

3.1 Input Image

In our experience we used tow dataset to test the final performance of our method. One is Street View Text dataset. It was harvested from Google Street View, and mostly deals with outdoor street level signboards. the other one contains images captured in indoors scenes. To ensure good test, images are captured under different lighting conditions (clear day, night, strong artificial lights…). The two Dataset together includes 1000 images. Each image with different size. In addition to this two dataset we have used images from our environment to make our experiments more effective.

Figure 4.2: Outdoor Signboards Images.

Figure 4.3: Indoor Signboards Images.

3.2 EAST (Efficient and Accurate Scene Text Detector) model

In our experiments, we Used pre-trained EAST neural network model. it consists of two steps:

3.2.1 Convolution neural network (CNN)

We will pass the input image to EAST convolution neural network and in return we will have two outputs:

1. The first output is the probability of a region containing text or not.
2. The second output is the feature map that represents the "geometry" of the image (we'll be able to use this geometry to draw the bounding box coordinates of the text in the input image).

3.2.2 Regions of Interest

We'll use the geometry and the probability that we have get from the previous step to extract the text regions of interest (ROIs) then use them in the next step.

Figure 4.4: First Step Text Detection.

3.3 Tesseract OCR engine with LSTM

we will use a special function in Pytesseract library, this function will pass the regions of interest to Tesseract deep learning text recognition engine. Then the long short-term memory neural network of Tesseract OCR engine will start to recognize the text and give us the final results as a text.

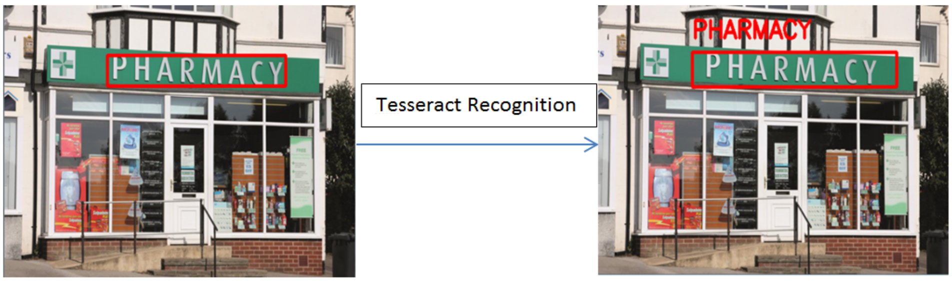


Figure 4.5: Second Step Text Recognition.

3.4 Output Text

Now we have our final result as a text, we will take a last step in our architecture by passing this result to the final function.

3.5 Convert To Voice

Finally, we will use a simple function. This function transforms any text into spoken word. So, we will use the output text from the previous steps as input in this function and in return it will give us spoken text.

Conclusion

In this chapter we have presented the detection and recognition method based on convolutional neural networks. We will show in the following chapter the results obtained in terms of precision and errors.

Chapter 5

Results

Introduction

In this chapter, we will present the results that we have obtained from our method, and we will show some wrong results that we faced through our experiences. Then we will present the reasons that make some results unsuitable.

1 Results

In this part we will evaluated our method in terms of precision and performance to show the effectiveness. The figures below show some results obtained by our method.

First we will start with a images form our environment, this images contains signboards with a Latin words :

Figure 5.1: The First Test

Figure 5.2: The Second Test

Figure 5.3: The third Test

Figure 5.4: The Fourth Test

Figure 5.5: Test Images Contains a Latin Words and the results we obtained.

The next images is also from our environment but with one different, this images contains signboards with Arabic words. In this experiment we have obtained our results normally, but we have faced one issue with displaying results. The OpenCV library is not able to draw the Arabic text. So to solve this problem we will display the console of Python that contain the results.

Figure 5.6: The First Test

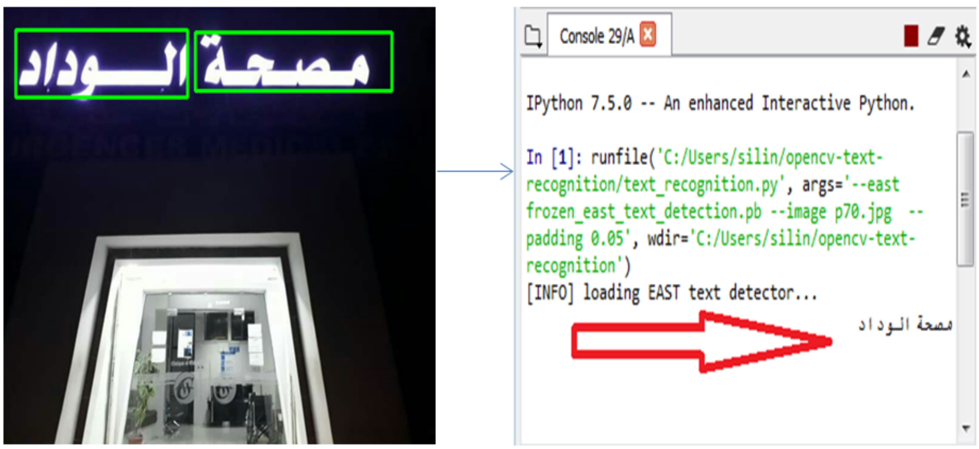


Figure 5.7: The Second Test

Figure 5.8: Test Images Contains a Arabic Words and The Console Results.

Now, we will display some results that we have obtained from our method using images from the dataset. The images are outdoor and indoor signboards with different lighting conditions:

Figure 5.9: The First Test.

Figure 5.10: The Second Test.

Figure 5.11: The Third Test.

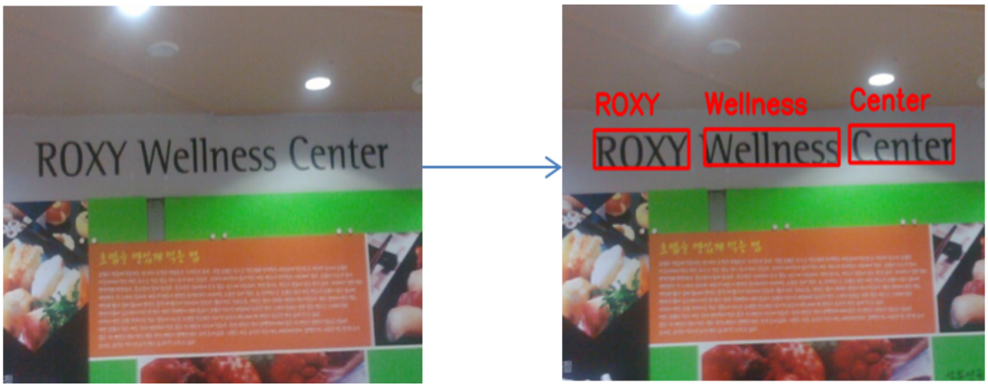


Figure 5.12: The Fourth Test.

Figure 5.13: Signboards Images and The Results from our Method.

Finally, in the next Figure we will try signboards images in different night lighting:

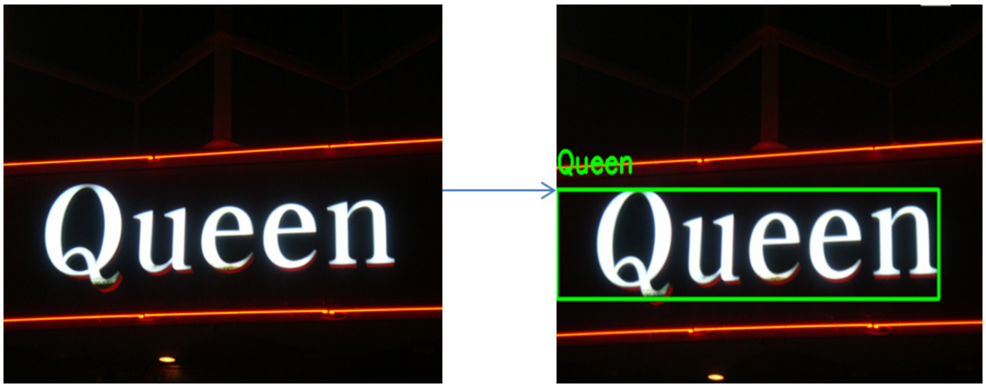


Figure 5.14: The First Test.

Figure 5.15: The Second Test.

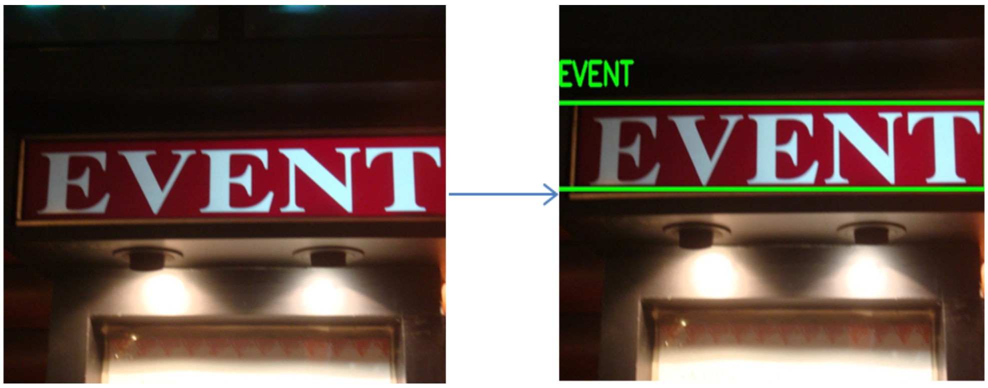


Figure 5.16: Signboards Images in Night Lighting and Our Method Results.

Noticing in the previous figures that the detection and recognition of text has been well done by our method. But detecting text in natural scene images in real-world conditions can be so challenging because like all algorithms there are some errors that appear when some unintended circumstances interfere with the functioning of our method leading to wrong results. The following figures illustrate some of the results we have obtained during the testing of the effectiveness, which seems to be wrong:

Figure 5.17: The First Test.

Figure 5.18: The Second Test.

Figure 5.19: Signboards Images and The Error Results from our Method.

To explain the error causes, we summed up the most important reasons that lead to errors during detection and recognition of text. It is as follows [42]:

- **Image/sensor noise:** Sensor noise from a handheld camera is typically higher than that of a traditional scanner. Additionally, low-priced cameras will typically interpolate the pixels of raw sensors to produce real colors.
- **Viewing angles:** Natural scene text can naturally have viewing angles that are not parallel to the text, making the text harder to recognize.
- **Blurring:** Uncontrolled environments tend to have blur, especially if the end user is utilizing a smartphone that does not have some form of stabilization.
- **Lighting conditions:** We cannot make any assumptions regarding our lighting conditions in natural scene images. It may be near dark, the flash on the camera may be on, or the sun may be shining brightly, saturating the entire image.

- **Resolution:** Not all cameras are created equal — we may be dealing with cameras with sub-par resolution.
- **Non-paper objects:** Most, but not all, paper is not reflective (at least in context of paper you are trying to scan). Text in natural scenes may be reflective, including logos, signs...
- **Non-planar objects:** Consider what happens when you wrap text around a bottle — the text on the surface becomes distorted and deformed. While humans may still be able to easily "detect" and read the text, our algorithms will struggle. We need to be able to handle such use cases.
- **Unknown layout:** We cannot use any a priori information to give our algorithms "clues" as to where the text resides.

Conclusion

This chapter we have shown the results obtained in terms of accuracy and error. We conclude from the previous figures that despite some of the errors that have arisen because of circumstances beyond our control, our method has shown a satisfactory for evaluation for our results in terms of detection and recognition of the text in natural scene images.

General Conclusion and Perspectives

The following work is the result of research in the field of signboards detection and recognition. The goal of our work is to develop a signboard detection and recognition method with ability of extracting text and words form the signboard then transform this text into voices. To achieve our goal, we have used a machine learning, exactly transfer learning by convolution neural networks (CNN) to detect the text region in the signboard which we were interested in. Then we extracted the words from the signboard using one of the famous text recognition engines, this engine using a long short term memory neural network (LSTM), and finally we transform it to voice.

The results obtained from our method show a remarkable improvement and great effectiveness .

We have come out with new perspectives after seeing the results of this work, the most important one is that we can make some adjustments on our method in a future work so it can detect and recognize round or vertical text. Also we can use video stream instead of images...

Bibliography

[1] Cornuejols, A and Miclet, L. 2002. Apprentissage artificiel, concepts et algorithms, Eyrolles. ISBN 2-212-11020-0.

[2] http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/l-ia-au-debut-du-xxeme-siecle.html, accessed:2019-02-03.

[3] Russell, S. J. and Norvig, P. 2016. Artificial intelligence : a modern approach. Malaysia, Pearson Education Limited, (Meghyn. Bienvenu. Introduction a l Intelligence Artificielle. Universite Aix-Marseille 1. 2007-2008 et 2008-2009.).

[4] Russell, S. 2009. Artificial intelligence A modern approach author Stuart russell, peter norvig, publisher Prentice hall pa.

[5] C.J, Gronlund. olprod, 12,07,2017 OpenLocalizationService Presentation de Machine Learning dans le cloud Azure, Microsoft azure.

[6] Portugal, I, Alencar, P, and Cowan, D 2017. The use of machine learning algorithms in recommender systems a systematic review. Expert Systems with Applications.

[7] Chen, D, and Stow, D 2002. The effect of training strategies on supervised classification at different spatial resolutions. Photogrammetric Engineering and Remote Sensing, 68,11, 1155-1162.

[8] https://en.proft.me/2015/12/24/types-machine-learning-algorithms/, accessed:2019-02-06.

[9] Bastien L, Deep Learning ou apprentissage profond : définition, qu'est-ce que c'est, comment ça marche, à quoi ça sert , le magazine Cloud and big data, data analytics, Novembre, 13, 2017.

[10] Kadous Djamila, MEMOIRE DE MASTER, Utilisation des reseaux de neurones comme outil du datamining Generation de modele comportemental d un processus physique a partir de donnees, UNIVERSITE ABOU BAKR BELKAID DE TLEMCEN,2012.

[11] https://medium.com/kansas-city-machine-learning-artificial-intelligen/an-introduction-to-transfer-learning-in-machine-learning-7efd104b6026, accessed:2019-02-20.

[12] https://machinelearningmastery.com/transfer-learning-for-deep-learning, accessed:2019-02-25.

[13] Touzet, C, 1992. Les reseaux de neurones artificiels introduction au connexionnisme.

[14] PHILIPPE, B. 2006. Apprentissage Statistique and Data mining. Institut national des sciences appliquees INSA de toulouse, Version Octobre.

[15] Pierre Buyssens, livre de Fusion de differents modes de capture pour la reconnaissance du visage appliquée aux transactions, 31 Oct 2014.

[16] Parizeau, M. 2004. Reseaux de neurones. GIF-21140 et GIF-64326, 124.

[17] Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79,8, 2554-2558.

[18] Kohonen, T. 1982. Self organized formation of topologically correct feature maps , Biological Cybernetics, Vol. 43, pp. 59-69.

[19] Chabot, F. 2017. Analyse fine 2D,3D de vehicules par reseaux de neurones profonds Doctoral dissertation, Universite Clermont Auvergne.

[20] Aurelien, G. 2017. HandsOn Machine Learning with Scikit Learn and TensorFlow.

[21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient based learning applied to document recognition. Proceedings of the IEEE, 8611, 2278-2324.

[22] Xiang, W., Tran, H. D., and Johnson, T. T. 2017. Reachable Set Computation and Safety Verification for Neural Networks with ReLU Activations. arXiv preprint arXiv 1712.08163.

[23] Kanestrom, P. 2017. Traffic flow forecasting with deep learning Master thesis, NTNU.

[24] Pezeshki, M. 2017. Towards deep semi supervised learning.

[25] http://colah.github.io/posts/2015-08-Understanding-LSTMs/, accessed:2019-02-27.

[26] Random House Kernerman Webster College Dictionary. K Dictionaries Ltd, Random House, Inc. 2010.

[27] https://www.collinsdictionary.com/dictionary/english/signboard, accessed:2019-03-01.

[28] https://octink.com/what-is-signage-and-what-is-its-purpose/, accessed:2019-03-03.

[29] https://en.wikipedia.org/wiki/Signage, accessed:2019-03-05.

[30] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. Proc. of British Machine Vision Conference, pages 384-396, 2002.

[31] Chen, Huizhong, Tsai, Sam, Schroth, Georg, Chen, David, Grzeszczuk, Radek, Girod, Bernd. Robust Text Detection in Natural Images with Edg enhanced Maximally Stable Extremal Regions. Proc. IEEE International Conference on Image Processing 2011.

[32] Epshtein, Boris, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

[33] Zhou, Xinyu, et al. EAST an efficient and accurate scene text detector. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017.

[34] Shi, Cunzhao, et al. Scene text recognition using part-based tree-structured character detection Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

[35] Cheng, Zhanzhan, et al. Focusing attention Towards accurate text recognition in natural images. Proceedings of the IEEE International Conference on Computer Vision. 2017.

[36] https://en.wikipedia.org/wiki/Tesseract-software, accessed:2019-03-18.

[37] https://en.wikipedia.org/wiki/Anaconda-Python-distribution, accessed:2019-03-21.

[38] Van Rossum, G. 1990. The python language.

[39] Raybaut, P. 2009. Spyder-Documentation.

[40] Bradski, G., and Kaehler, A. 2008. Learning OpenCV Computer vision with the OpenCV library. Reilly Media, Inc.

[41] https://www.tensorflow.org/, accessed:2019-04-15.

[42] Thillou, Celine Mancas, and Bernard Gosselin Natural scene text understanding Vision systems segmentation and pattern recognition. IntechOpen 2007.