PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF DJILALI BOUNAAMA KHEMIS MILIANA

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

THESIS PRESENTED TO OBTAIN

THE MASTER DEGREE ON « COMPUTER SCIENCE»

OPTION:

«SOFTWARE ENGINEERING AND DISTRIBUTED SYSTEMS»

Traffic Congestion Detection Based On Deep Learning

Realized by:
BELHENNICHE Marwa

The jury composed of:
Mr.N.AZZOUZA: President
Mr.O.BOUKADOUM: Examiner
Mr. A.KHALFI : Supervisor

Academic year: 2022/2023.

Dedication

“

To my beloved father, I dedicate this memoir as a testament to my deep love and endless gratitude. May you be proud of me, for everything I have achieved is largely thanks to you. I love you with all my heart, Dad.

To my wonderful mother, I am forever grateful for your unwavering belief in me and the endless sacrifices you have made. With all my heart, I appreciate you, Mom.

To my wonderful sister Meryem, you are more than just a sister to me. You are my confidante. You have always encouraged me to pursue my dreams.

To my dear brother Abd El Kader and my sister-in-law Fatima, your love and unwavering support are precious treasures.

To my dear nephew Seddik, your presence in my life is a source of joy and inspiration. I love you with all my heart.

To all my close friends who have supported me and have been unforgettable and amazing individuals, especially Nadjlaa, Safaa, and Hadjer, whom I love dearly.

To all those dear to me, to all of you, thank you

”

- Marwa

Acknowledgements

I thank Allah the Almighty for granting me the courage and patience necessary to complete this work.

I would like to express my heartfelt gratitude to ***Mr. Ali Khalfi*** *my supervisor, for his immense assistance and for providing me with invaluable advice and information with an unparalleled level of patience and professionalism.*

I would also like to thank all my professors at Djillali Bounaama University, especially ***Dr. Imène Ait Abderrahim*** *and* ***Mr. Riadh Meghatria****.*

May the members of the jury find here the expression of my sincere gratitude for the honor they bestow upon me by taking the time to read and evaluate this work..

I would also like to express my gratitude to the academic and administrative staff of the Faculty of Science and Technology for their efforts in providing us with an excellent education.

To conclude, I would like to express my gratitude to everyone who has been instrumental in the successful completion of this thesis.

Abstract

Traffic congestion is a common phenomenon that is not only stressful for drivers but also has a negative impact on a country's economy, the well-being of its population, and the stability of its ecosystem.

Artificial intelligence, particularly deep learning, can be used to detect traffic congestion by analyzing traffic data from various sources, such as surveillance cameras. Deep learning models can identify congestion patterns by analyzing this data and recognizing signs of slowdown or congestion on the roads. This enables the provision of information about traffic conditions and helps in the management of traffic flow.

In this work, we conducted a comparative study considering two scenarios. The first scenario involved using a complete CNN classifier for both feature extraction and classification. The second scenario utilized CNN solely for feature extraction, with the extracted features then fed into traditional machine learning classifiers such as KNN, SVM, RF, and XGB.

Among the evaluated models, the combination of CNN and SVM, as well as CNN and KNN, demonstrated the highest accuracy, achieving 99.78% for binary traffic state classification.

Keywords : Machine learning, Deep learning, CNN, XGBoost, Random forest, SVM ,KNN ,Traffic congestion detection.

Résumé

La congestion routière est un phénomène courant qui est non seulement stressant pour les conducteurs, mais qui a également un impact négatif sur l'économie d'un pays, le bien-être de sa population et la stabilité de son écosystème.

L'intelligence artificielle, en particulier l'apprentissage en profondeur, peut être utilisée pour détecter la congestion routière en analysant les données de circulation provenant de différentes sources, telles que les caméras de surveillance. Les modèles d'apprentissage en profondeur peuvent identifier les schémas de congestion en analysant ces données et en reconnaissant les signes de ralentissement ou de congestion sur les routes. Cela permet de fournir des informations sur les conditions de circulation et aide à la gestion et à l'optimisation du flux de trafic.

Dans ce travail, nous avons réalisé une étude comparative en considérant deux scé- narios. Le premier scénario consistait à utiliser un classifieur CNN complet à la fois

Le deuxième scénario utilisait uniquement le CNN pour l'extraction des caractéristiques, les caractéristiques extraites étant ensuite alimentées dans des classifieurs traditionnels d'apprentissage automatique tels que KNN, SVM, RF et XGB.

Parmi les modèles évalués, la combinaison du CNN et du SVM, ainsi que du CNN et du KNN, a démontré la plus grande précision, atteignant 99,78 % pour la classification binaire de l'état de la circulation .

Mots clés : Apprentissage automatique, Apprentissage en profondeur, CNN, Détection de la congestion routière, Classification d'images.

ملخص

الازدحام المروري هو ظاهرة شائعة لا تُسبب الاجهاد فقط على السائقين، بل لها تأثير سلبي على اقتصاد البلدان ورفاهية سكانها واستقرار البيئة الطبيعية.

يمكن استخدام الذكاء الاصطناعي، وبشكل خاص التعلم العميق، لكشف الازدحام المروري من خلال تحليل بيانات المرور من مصادر مختلفة مثل كاميرات المراقبة . يمكن لنماذج التعلم العميق التعرف على أنماط الاختناق عن طريق تحليل هذه البيانات واكتشاف علامات التباطؤ أو الاختناق على الطرق. وهذا يمكنه توفير معلومات حول ظروف المرور ومساعدة في إدارة وتحسين تدفق المرور

في هذا العمل، أجرينا دراسة مقارنة تنظر في سيناريوهين. السيناريو الأول ينطوي على استخدام مصنف شبكات العصب الاصطناعي التكوينية كامل لاستخراج الميزات والتصنيف. أما السيناريو الثاني، فيستخدم شبكات العصب الاصطناعي التكوينية فقط لاستخراج الميزات، ثم يتم تغذية اَمْة، الميزات المستخرجة إلى مصنفات تقليدية للتعلم الآلي مثل ك و صطي و غذ و .نجـ من بين النماذج المقي أظهرت ث ■ك أعلى دقة، حيث حققت نسبة دقة تبلغ ٩٩٠ %٧٨ في

كلمات مفتاحية :
التعلم الآلي، التعلم العميق، شبكات العصب الاصطناعي التكوينية، اكتشاف ازدحام المرور، تصنيف الصور.

Contents

List of Tables

List of Figures

Acronyms

Adam: Adaptive Moment estimation.

CNN: Convolutional Neural Network.

FN: False Negative.

FP: False Positive.

GAN: Generative Adversarial Network.

ReLU: Rectified Linear Unit.

TN: True Negative.

TP: True Positive.

KNN: K-Nearest Neighbors.

IA: Intelligence Artificielle.

LSTM: Long Short-Term Memory.

SVM: Support Vector Machine.

SGD: Stochastic Gradient Descent.

ROI: Region Of Interest.

IA: Intelligence Artificielle.

ML: Machine Learning.

DL: Deep Learning.

XGBoost: eXtreme Gradient Boosting.

RF: Random Forest.

RNN: Recurrent Neural Networks.

RMSprop: Root Mean Square Propagation.

RGB: Red, Green, and Blue.

Introduction

Artificial Intelligence, specifically machine learning, has found a prominent place in various domains of our daily lives, including industry, commerce, healthcare, transportation management, and cybersecurity .(1) These technologies have improved efficiency, accuracy, and user-friendliness in these sectors.

In the current context, traffic congestion has become a major challenge for our modern society. This situation is exacerbated by the increasing demand for travel and transportation, leading to significant issues such as increased congestion and decreased road safety .(2) Factors contributing to this congestion problem include accidents, adverse weather conditions, slow-moving heavy vehicles, sudden vehicle breakdowns, non-compliance with traffic regulations, unexpected traffic jams at intersections, and physical road conditions (3).

However, the integration of modern technologies such as machine learning, particularly deep learning, offers a potential solution to this problem (4). By utilizing advanced deep learning techniques, it becomes possible to process and analyze vast amounts of data from various sources, such as surveillance cameras. This enables the extraction of precise information about traffic patterns, frequent congestion points, and critical moments, which aids in better understanding and predicting traffic flow patterns.

In this context, our objective is to design a Convolutional Neural Network (CNN) model capable of detecting traffic congestion stats from a query image associated with a congestion situation. To enhance the performance of our CNN, we are considering an approach that combines different classification methods with the CNN, such as combining the CNN with SVM, Random Forest, XGBoost (XGB), and KNN models. This hybrid approach will allow us to leverage the advantages of each algorithm and improve the accuracy and overall performance of traffic congestion detection.

Our thesis is organized into four chapters:

The first chapter, **"Generalities on Images"** presents a general overview of images and introduces various image processing techniques.

The second chapter, **"State of the Art"** presents the main terms in machine learning and deep learning. Additionally, it offers a review of the most relevant and innovative studies in this field, based on recent works.

The third chapter, **"The proposed system conception"** presents the global architecture of the system proposed and its different modules that allow the detection.

The fourth chapter, **"Implementation and discussion of the results"** represents the experimental part of our work, where we discuss the different results obtained by our proposed system. We will finish by comparing our results to other recent works.

Chapter 1

Generalities on Images

1.1 Introduction

Today, images have become one of the most essential means for humans to communicate with others. They are a universal form of communication that enables people of all ages and cultures to understand each other easily.

In this chapter, we will cover some general concepts. Next, we will explore some transformations that can be applied to images. Then, the image processing system. Finally, the visual similarity between images.

1.2 Image definition

An image is a representation of something visually or mentally, such as an object, a living being, or a concept. In the case of 2D (x,y) images which are the most common, the points are called pixels. And this kind of image works well for display on computer screen (also pixel oriented).

From a mathematical perspective, the image may be viewed as a function of R * R in R, where R is a matrix of numbers that represents a signal, and there are various tools available to manipulate this signal.

From a human perspective, many semantic details can be found in an image. It is necessary to interpret the content in light of the numerical value.(5)

1.3 Digital image

A digital image is a digital signal composed of elementary units (called pixels) that each represent a portion of the image. Unlike the one-dimensional case, we will study only digital images (discrete). It's defined by:

- the number of pixels that compose it in width and height.
- the range of shades of gray or colors that can take each pixel (we speak of dynamics of the image). (6)

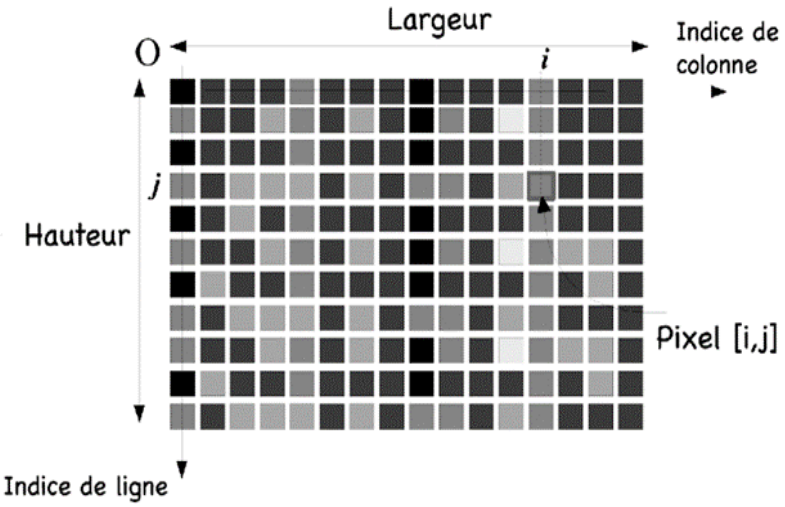

Figure 1.1: Digital image representation.
(6)

1.3.1 Different types of digital image format

Generally, there are two main types of digital images:

1.3.1.1 Bitmap images (also called matrix images)

They are made up of a grid of cells. i.e., an ensemble of points (pixels) contained in a table. Each of these pixels, is assigned a value of its color and luminance. (7) The majority of images used in the field of image processing and analysis are raster (matrix) images such as BMP, PCX, GIF, JPG, PNG and TIFF.

1.3.1.2 Victor images: (also called object-oriented images)

Due to the simple organization of the information in an image, its size has been greatly decreased. This simplicity lies in generating the objects that make up an image through geometric lines determined by calculations and mathematical formulas. The vector graphics are displayed from the coordinates of a line saved as a reference, which forms the objects from the mathematical definition of the points and lines straight or curved. (7)

1.3.2 Digital image types

We distinguish three types of images:

1.3.2.1 Binary image (black or white)

It's a rectangular matrix associated with the value of the pixel. 0 or 1 are the only possible values for the pixel (Figure 1.3). (8) White pixels have a value of 1, while black pixels have a value of 0.(9) As a result, only one bit is used to code a level. (8)

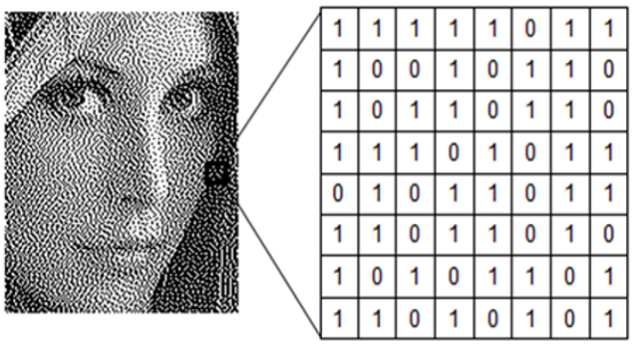

Figure 1.2: a binary image.
(10)

1.3.2.2 Grayscale images

The gray level is the value of the light intensity at particular point. When the light is at its maximum intensity, the value is 255 (2 8 - 1 = 255), which represents the color white. When the light is at its lowest intensity, the value is zero, which represents the color black. (8) The black and white colors are represented by values

between 0 and 255. (9) (Figure 1.4) shows this color gradient. The quantization of the image is related to the number 256. Grayscale image encoding requires 8 bits. (8)

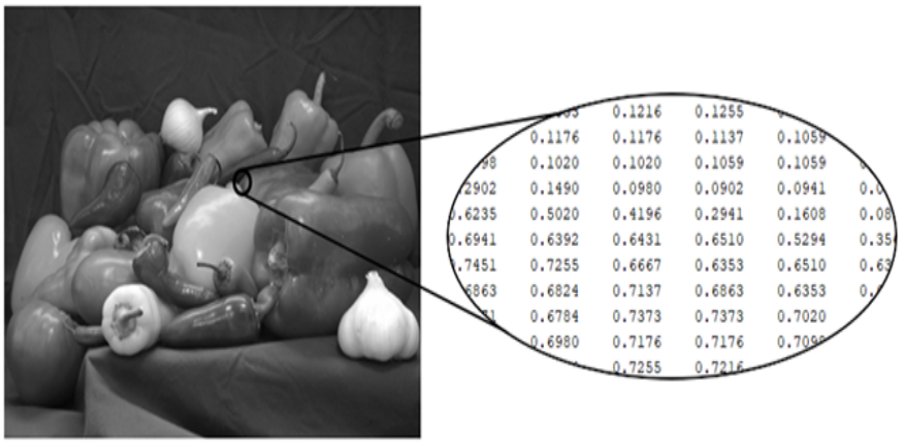

Figure 1.3: a grayscale image.
(10)

1.3.2.3 A color image

Since in the digital images processing the color space is the most used. A color image represented by three matrixes red, green, and blue (RGB) as showing in (figure 1.5).As each color component is generally represented by 8 bits, so every image pixel is represented using a 24-bit store. The combination of these major elements results in the final color of each pixel, which is equivalent to the dosage of the three primary colors (RGB) present in each bit. (8)

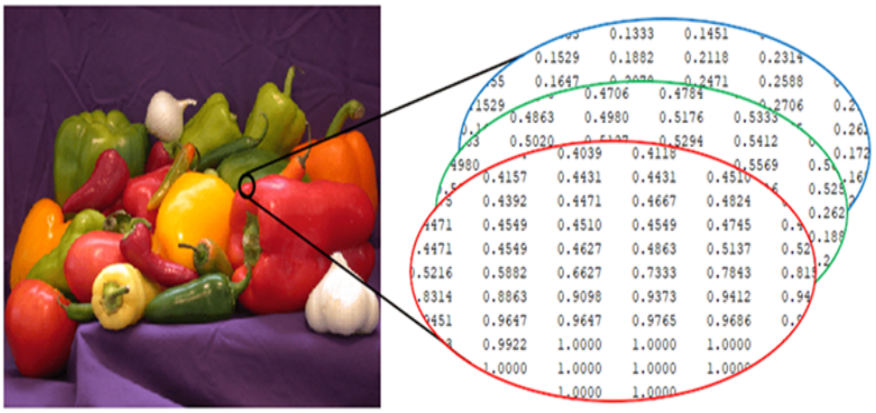

Figure 1.4: a color image (RGB).
(10)

1.3.3 Characteristics of a digital image

An image can be defined as a structured set of information that includes the following characteristics:

1.3.3.1 Pixel:

The term pixel is an abbreviation of (PICture Elements). It's the image's smallest component (11). The light intensity values of each pixel constitue an image. (9)

1.3.3.2 Dimension (size):

An image's dimension corresponds to the number of pixels that make up its height N and width M, and we note it M * N (9). The notions of pixel and image dimension are depicted in figure 1.6.

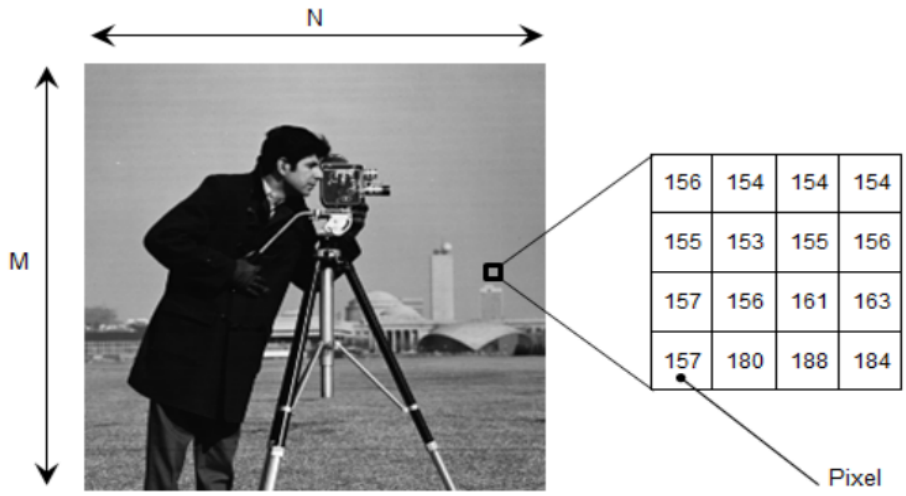

Figure 1.5: Example showing each of the image's dimensions and pixels n (cameraman.tif).

(12)

1.3.3.3 Color:

color in the image is defined as being a set of pixels each having three red, green and blue components as a result of after the scene has been digitalized by a color camera. Other color spaces can also be used to express these components. (13)

1.3.3.4 Contours and textures:

Contours represent the boundary between objects in the image, or the boundary between two pixels with significantly different grey levels. Textures describe the structure of these objects (pixels). Finding the points in the image where two different textures are separated is known as contour extraction. (11)

Figure 1.6: Contour of an image.
(12)

1.3.3.5 Contrast:

It is the marked opposition between two regions (dark and light) of an image. the contrast C is defined by the ratio:

C=L1-L2/L1+L2.

such as: the luminosities of two adjacent areas in the image are L1 and L2, respectively. (11)

Figure 1.7: Example showing a low-contrast image (a) and high-contrast image (b).
(14)

1.3.3.6 Luminance:

For an image, it's the degree of brightness of its points. It's defined as being the quotient of a surface's apparent area and luminous intensity. A bright image, strong contrast, and the absence of parasites are all indicators of good brightness. (11)

1.3.3.7 Noise:

Noise is the blurring of visual detail caused by unintentional variations in color or brightness. (15) Generally, there are two types of image noise:

- chrominance noise, which is the color component of noisy pixels: it is visible as random colored spots.
- luminance noise, which is the light component of noisy pixels: it is visible as darker or lighter spots. (11)

Figure 1.8: (a) Image without noise (b) Image with noise
(16)

1.3.3.8 Histogram

The image histogram is the histogram of the data series corresponding to the grey levels of the pixels. Its definition as a discrete function is:

$$\forall p \in \{0, ..., 255\} \; ; \; h_p = The \quad number \quad of \quad pixels \quad with \quad a \quad grayscale \quad level \quad 'p'.$$

A "continuous"; version h of the histogram can be defined by interpolating (e.g. piecewise linear) the hp values so that: (17)

$$\forall p \in \{0, ..., 255\} \; ; \qquad H(P) = h_p$$

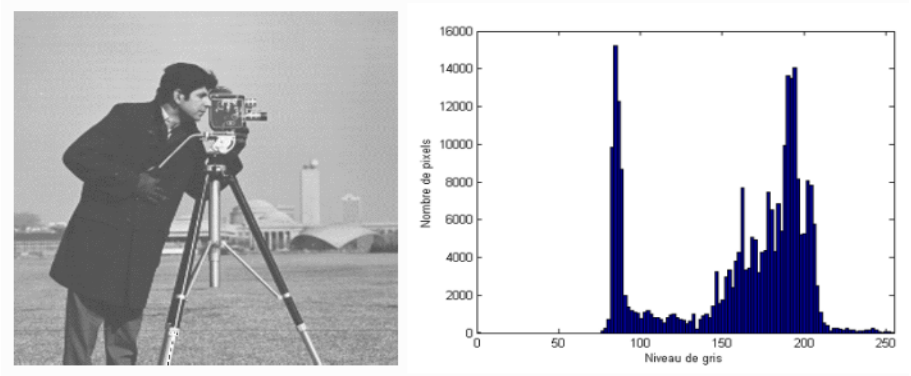

Figure 1.9: Example showing a grayscale image with its histogram. (17)

1.4 Transformation applied to images

There are many different types of image transformations that can be applied, depending on the desired outcome. Some of the most common ones include:

1.4.1 Rotation

The rotation, noted r (center, degree) r (center, degree), is a geometric transformation which allows to obtain the image of an initial figure following a ”shift”, defined by a degree and a direction, around the rotation’s center.(18)

Figure 1.10: (a) Original Image (b) Image rotated by 180 degree. (18)

1.4.2 Scaling

Image scaling describes the process of resizing an image to a larger or smaller version while maintaining the same aspect ratio. This can be done to change the size of the image, its resolution, or the level of detail in the image.(18)

Figure 1.11: (a) Original Image (b) Image after Scaling.
(18)

1.4.3 Translation

Translation basically means shifting the object's location, either horizontally or vertically by some defined off-set (measured in pixels).(18)

Figure 1.12: (a) Original Image (b)Image after Translation.
(18)

1.4.4 Shearing

Basically, this shifts some part of an image to a direction and other parts to some other direction. For example, horizontal shearing will shift the upper part to the right and lower part to the left. This concept can be better understood through visualization. (18)

Figure 1.13: a original image, b the sheared image in the direction of the X-axis, and c the sheared image in the direction of the Y-axis.
(19)

1.5 Digital image processing

Image processing refers to the technique of converting a visual image into a digital form and performing certain operations to get some useful information from it. (20) There are various image processing techniques that can be employed for different purposes. Here are a few examples:

1.5.1 Image acquisition

Image acquisition is a fundamental step in machine vision (MV) workflows, where it involves capturing digital images of objects using various hardware systems, such as cameras, encoders, and sensors. Without a high-quality image, the subsequent steps in the MV workflow will be rendered useless.

Since machine vision systems don't analyze the acquired digital image of the object and not the object itself, gaining an image with the right clarity and contrast is paramount.

During the image acquisition process, the incoming light wave from an object is converted into an electrical signal by a combination of photo-sensitive sensors. These small subsystems fulfill the role of providing your machine vision algorithms with an accurate description of the object. (21)

1.5.2 Image pre-processing (Filtering)

Digital images as they are acquired are often unusable for image processing. They contain noisy signals. To remedy this, different pre-processing techniques for improvement or correction are applied. Generally, filters are categorized into two parts: (22)

1.5.2.1 Linear filter

Linear filters apply a two-dimensional convolution, which is a mathematical operation used to transform an input data set into an output data set. They are commonly used to eliminate noise from digital images. The size of each filter is typically represented as an odd number N x N, where N refers to the number of rows and columns in the filter.

The most well-known linear filters are the low-pass and high-pass filters. (23)

1. **Low-pass filter** This filter does not affect the low-frequency components in image data, but should attenuate the high-frequency components. The smoothing operation is often used to reduce image noise and irregularities, and it can be repeated multiple times, resulting in a blurring effect. In practice, a compromise must be made between noise attenuation and the preservation of important details and edges.

$$1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 1.14: Filter mask (smoothing).
(23)

2. **High-pass filter** The sharpening of edges and their extraction are achieved in the frequency domain by applying a high-pass filter. The digital high-pass filter has characteristics that are the inverse of the low-pass filter. This filter does not affect the high-frequency components of a signal, but it must attenuate the low-frequency components.(23)

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 1.15: Filter mask (Sharpening).
(23)

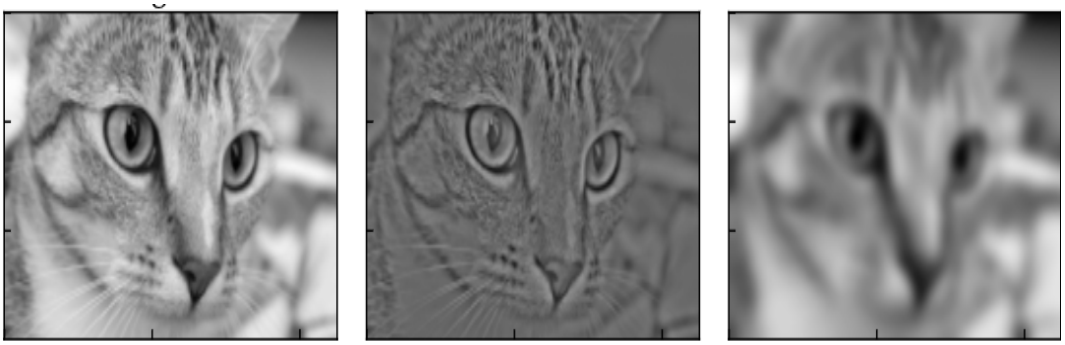

Figure 1.16: Digital image without filter (left), with low-pass filter (center), and with high-pass filter (right).
(24)

1.5.2.2 Nonlinear filters

(23) They are designed to address the limitations of linear filters. The principle behind nonlinear filters is the same as linear filters, which involves replacing the value of each pixel with the value of a function calculated in its neighborhood. However, the major difference is that this function is no longer linear, but can be any function (including operators for comparison or classification). The most well-known nonlinear filters are:

1. **Median filter** This filter is a commonly used technique for removing noise from an image, which can originate from various sources such as dust, small clouds, momentary drops in electrical intensity on sensors, etc. The algorithm for the median filter is as follows:

   (a) Sort the pixel values in ascending order.

   (b) Replace the value of the central pixel with the value located in the middle of the sorted list.

   (c) Repeat this process for all pixels in the image.

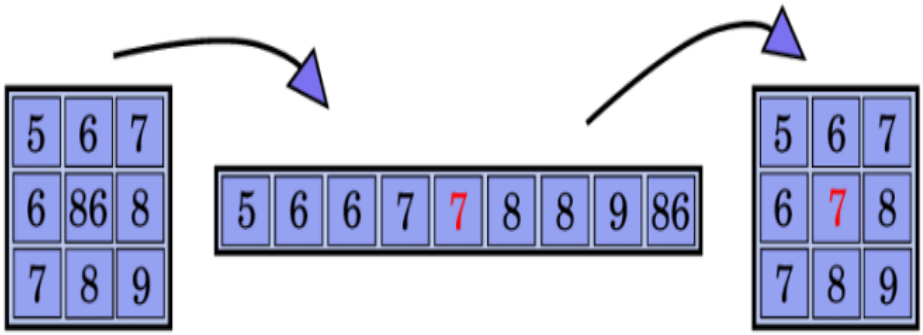

Figure 1.17: The principle of the median filter.
(25)

2. **Min Max Filter** The same process as the median filter is applied, but instead of replacing the central pixel value with the median value of its neighboring pixels, it is replaced with either the maximum or minimum value.

1.5.3 Segmentation

Segmentation is the most important part in image processing. Fence off an entire image into several parts which is something more meaningful and easier for further process. These several parts that are rejoined will cover the entire image. Segmentation may also depend on various features that are contained in the image. It may be either color or texture. Before denoising an image, it is segmented to recover the original image. The main motto of segmentation is to reduce the information for easy analysis.(26)

1.6 Visual similarity between images

1.6.1 Image descriptor vector

The image descriptor vector is a feature vector of the image, which is constructed from the attributes from the image. It is usually presented as a vector with n real components. The attributes extracted from images are of different types and are expressed in different units depending on whether they belong to color, texture, shape, etc. (27)

1.6.2 Similarity measure

The similarity measure is generally estimated by the ratio between the similarity and dissimilarity of two images. Two objects (images) can be considered perfectly similar if their similarity is very high and their dissimilarity tends towards zero, or when the ratio between the two measures tends towards infinity. (28)

1.6.3 Similarity distance

The similarity (or dissimilarity) distance is a mathematical measure that quantifies the difference or similarity between two objects.

1.6.3.1 Euclidean distance

(29) It is also called the L2 distance. If u= (x1, y1) and v= (x2, y2) are two points, then the Euclidean distance between u and v is given by:

$$EU_(u, v) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \; ; \qquad (1)$$

With two dimensions (m =2).

If the points have n-dimensions, eq. 1 can be generalized by defining the Euclidean distance between a and b as:

$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \; ; \qquad (2)$$

1.6.3.2 Manhattan distance:

(29) It is also called the L1 distance. If u= (x1, y1) and v= (x2, y2) are two points, then the Manhattan distance between u and v is given by:

$$MH_(a, b) = | \quad x_1 - x_2 \quad | + | \quad y_1 - y_2 \quad | \; ; \qquad (3)$$

With two dimensions (m =2).

If the points have n-dimensions, eq. 3 can be generalized by defining the Manhattan distance between a and b as:

$$\sum_{i=1}^{n} | \quad x_i - y_i \quad | \; ; \tag{4}$$

1.6.3.3 Minkowski distance

(30) Suppose two objects X and Y are represented by two p dimensional vectors, The Minkowski distance d(X; Y) is defined as:

$$D(X,Y) = (\sum_{i=1}^{p} | \quad x_i - y_i \quad |^r)^{\frac{1}{r}} \; ; \tag{5}$$

1.7 Conclusion

In this chapter, we summarized the definitions and basic concepts of images. Additionally, we discussed several image processing methods and explored the topic of visual similarity between images.

Chapter 2

State of the Art

2.1 Introduction

Traffic congestion is a major issue in our cities with significant consequences on the economy and quality of life of citizens. Therefore, detecting traffic congestion is essential for effectively managing traffic and reducing congestion problems, and recent advances in deep learning have led to significant development in these techniques.

In this chapter, we have presented the traffic congestion problem as well, as some notions in the field of automatic and deep learning. We have also provided an analysis of the most important research works in this domain, with a focus on deep learning-based methods.

2.2 Traffic congestion problem

Traffic congestion is a common problem in many urban areas around the world where demand for transportation outpaces the capacity of the road network, which causes slower speeds, longer travel times, and more delays for drivers, pedestrians, and transit users.

In this context our problematic detection of the traffic congestion which allows to detect the state of the congestion of the traffic starting from an images request associated a situation of a level of congestion.

Following that, traffic congestion detection is important for maintaining traffic conditions and improving the efficiency of the road network.

2.3 Artificial intelligence

(AI) includes the implementation of many techniques designed to make machines mimic some form of real intelligence. Artificial intelligence is being used in a growing number of fields of application.

The concept was born in the 1950s thanks to the mathematician Alan Turing. In his book Computers and Intelligence, Turing posed the question of giving machines a form of intelligence and wrote a test now known as the "Turing test", in which an object blindly interacts with another human and then with a machine programmed to formulate a reasonable response. If the subjects are indistinguishable, then the machine has passed the test and, according to the authors, can be considered as truly "intelligent". (31) The figure Figure 2.1: illustrates artificial intelligence and its subsets.

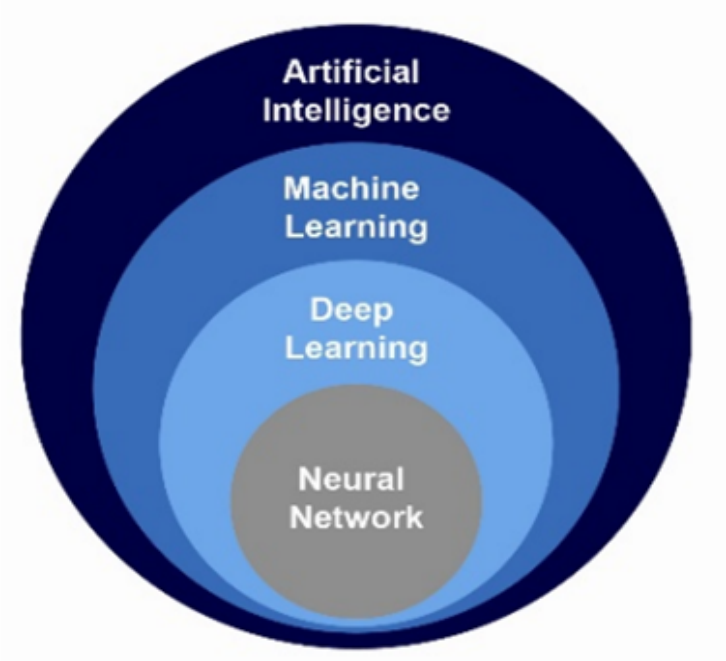

Figure 2.1: the relationship between Artificial Intelligence, Machine Learning, Neural Network and Deep Learning.
(32)

2.4 Machine learning

Machine learning (ML) is a subset of artificial intelligence, it demonstrates experiential learning linked to human intellect. while having the ability to learn and improve its analyses through the use of computational algorithms. These algorithms make effective use of large collections of data inputs and outputs to recognize models and effectively learn in order to train the machine to make recommendations or take independent decisions. The machine can take an input and predict an output once the method has been modified and repeated enough times. (33)

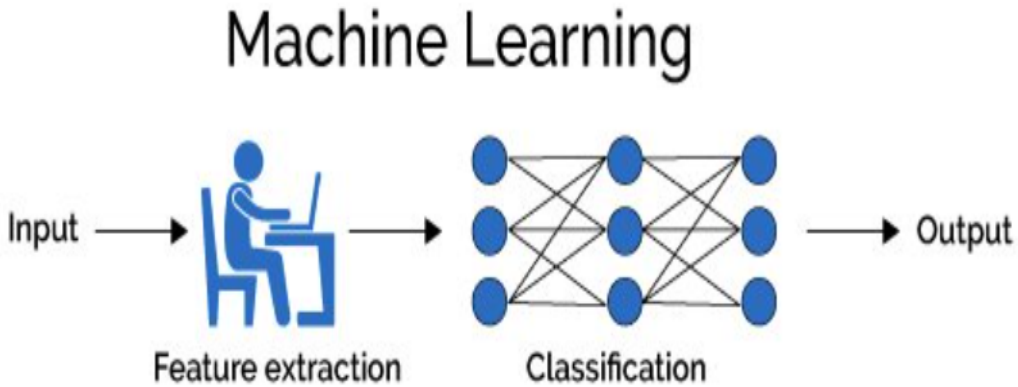

Figure 2.2: General schema of machine learning .
(34)

2.4.1 Machine learning algorithms

Among the machine learning algorithms that are widely used, we cite:

2.4.1.1 Decision tree

The decision tree uses an arborescent structure to build classification or regression models, it decomposes a data set into ever smaller subsets while gradually growing the decision tree that goes with it.

It is given this name because it starts out as a single or root decision and then splits into several branches before a decision or prediction is made, like a tree. (35)

2.4.1.2 Naive Bayes

The naive bayes is a classification technique based on Bayes Theorem with an assumption of independence among predictors.

Naive Bayes classifier, to put it simply, assumes that the presence of a particular characteristic in a class has nothing to do with the presence of any other characteristic. (36)

2.4.1.3 SVM

Support Vector Machine (SVM) is a ML-technique. This algorithm has strong regularization and can be applied to classification or regression problems. It is distinguished by the usage of noyaux, the parity of the solution, and the capacity control gained by modifying the margin or the quantity of support vectors, etc. (35)

2.4.1.4 XGBoost

XGBoost (eXtreme Gradient Boosting) is a machine learning algorithm that belongs to the family of gradient boosting methods. It can be used to solve regression and classification problems. XGBoost is favored by data scientists for its fast execution speed and efficient memory management.(37)

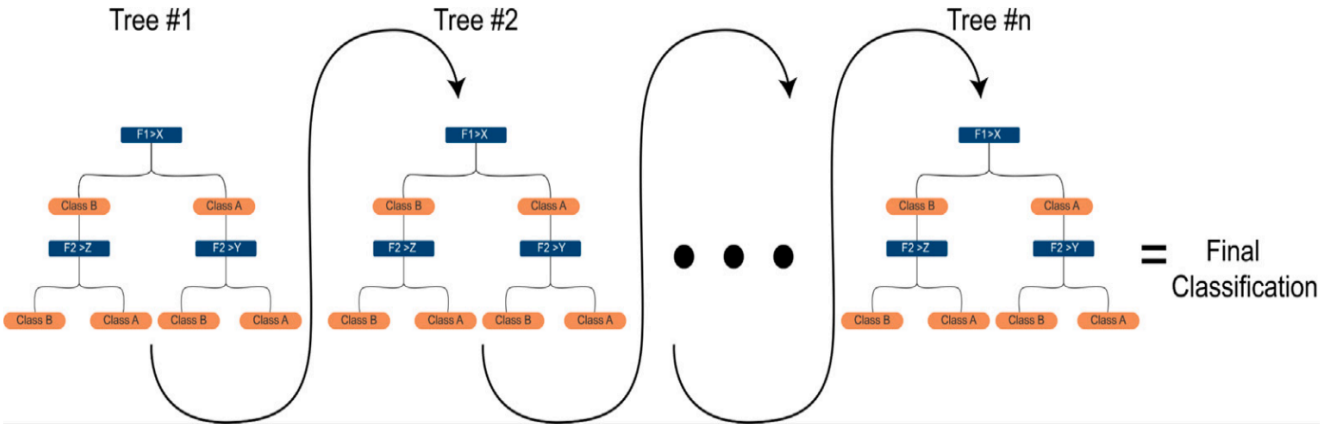

Figure 2.3: eXtreme Gradient Boosting (XGBoost) Schematic Representation. (38)

2.4.1.5 Logistic regression

Logistic regression is one of the most popular supervised ML algorithms. Unlike its name, the logistic regression model is more of a classification model than a regression model. It is one of the industry's most widely utilized regression methods which is extensively applied to fraud detection, credit card scoring and clinical trials. (35)

2.4.1.6 KNN

The KNN algorithm is the fundamental and simplest classification technique when there is little or no prior knowledge of the data distribution.

It can be used for regression as well as for classification, but it is mainly used for classification problems. The KNN allows the classification of instances from the closest training instances in the feature space. (39)

2.4.1.7 Random Forest

It is an improved version of the decision tree that is used for supervised learning. This classifier creates a number of decision trees and then uses a majority vote to determine the majority vote to determine the final prediction.

This algorithm is more efficient than decision trees, because decision trees work together as a group together as a group and correct each other's errors, while in the random forest algorithm the random forest algorithm the trees are not related to each other. This algorithm can be used for both classification and regression tasks. (40)

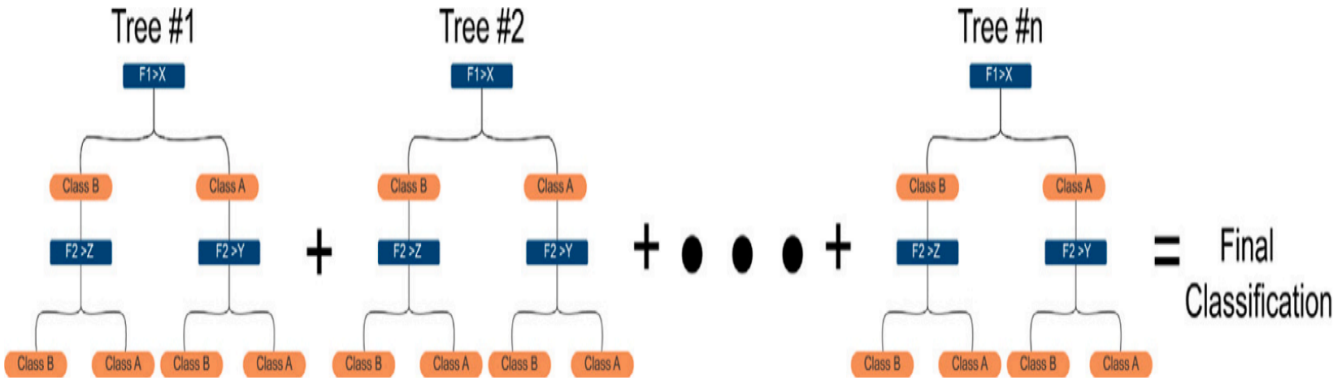

Figure 2.4: Random Forest Schematic Representation.
(38)

2.4.1.8 Extra trees

Extra Trees is an algorithm that aims to fit random decision trees on various subsets of the dataset. It differs from traditional decision trees in terms of construction. Instead of searching for the optimal split at each node, Extra Trees randomly selects maximum features and generates random splits for them. The best split is then chosen from these random splits. This approach helps to increase accuracy and control overfitting by incorporating randomness and averaging. (41)

2.5 Deep learning

Deep Learning (DL) is a subset of machine learning inspired by the structure of the human brain. It incorporates computer models and algorithms that mimic the architecture of biological neural networks in the brain (artificial neural networks).

Whenever the brain receives new information, it tries to compare it with known information to make sense of it. The brain decrypts the information by labeling it and assigning the elements to various categories, and DL employs the same concept. (42)

With its architectures, DL has a strong ability to extract inherent features from data from the lowest level to the highest level and perform complex calculations. Theoretically, any type of non-linear function can be approximated thanks to the hierarchical structure of neural units, which are the fundamental building block of Deep Learning architectures. (43)

Deep learning can thus represent traffic features without prior knowledge and gives good performances. (44)

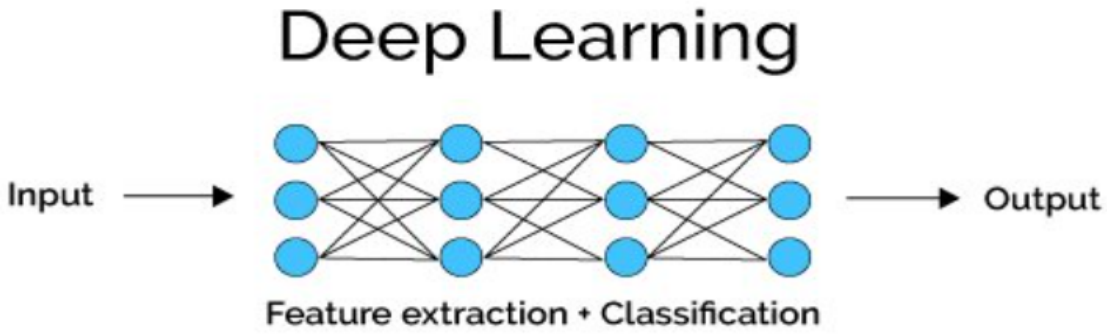

Figure 2.5: General schema of deep learning.
(34)

Among the most popular deep learning models, we mention:

2.5.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a variant of standard neural network which is specifically designed to process sequence data such as image. It requires minimal data preprocessing and it can automatically detect the invariant and extract important features. (45)

CNNs have historically been utilized often in image-classification tasks, due to their capacity to capture the correlation between nearby pixels in an image. A deep-CNN is able to capture the correlations between pixels placed far apart in the image.

In a typical CNN architecture, the first few layers are convolutional blocks, interspersed with pooling layers. Fully connected layers are present just before the output layer. Pooling is a down sampling technique used to report summary statistics from a neighborhood. The most commonly used pooling method with CNNs is max-pooling, wherein the maximum value of the activation is selected from a region. Pooling helps reduce the complexity of the deep learning model and also learn representations that are invariant to small local translations of the input data.(46)

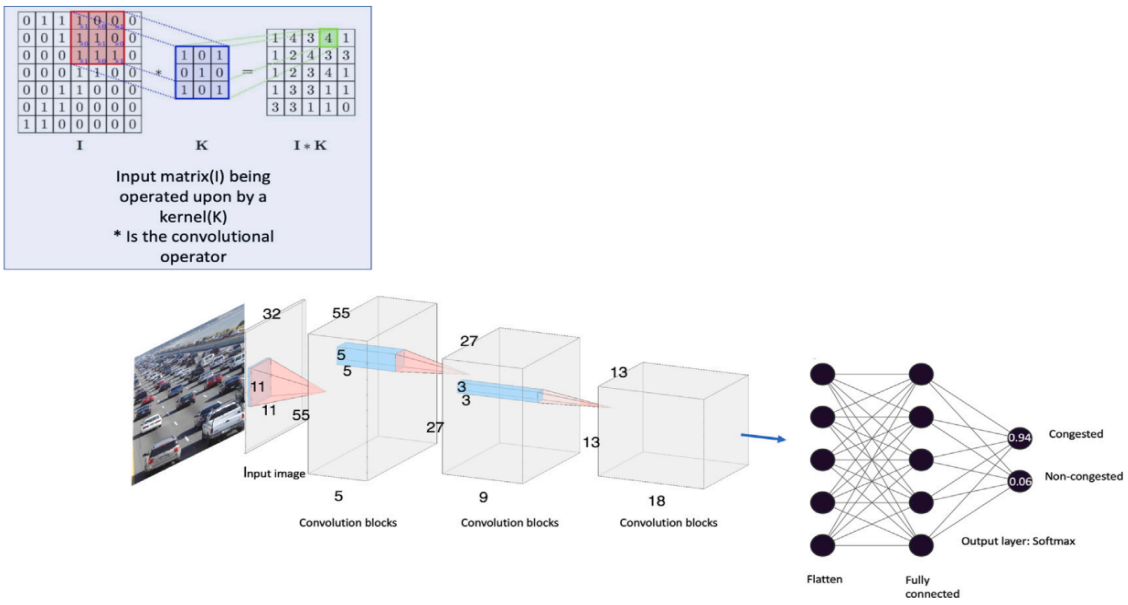

Figure 2.6: A deep convolutional neural network (CNN) .
(46)

The structure of a convolutional neural network includes several layers, among which we mention:

2.5.1.1 Convolutional layer

Convolutional layers are important components of convolutional neural networks and are always at least their first layer. Their goal is to determine the presence of a set of features in the input images.

The operation of the convolutional layer involves the implementation of a convolution filter, which slides a window representing the object over the image and calculates the convolution between the object and each part of the scanned image. This operation is illustrated in Figure 2.5 . (47)

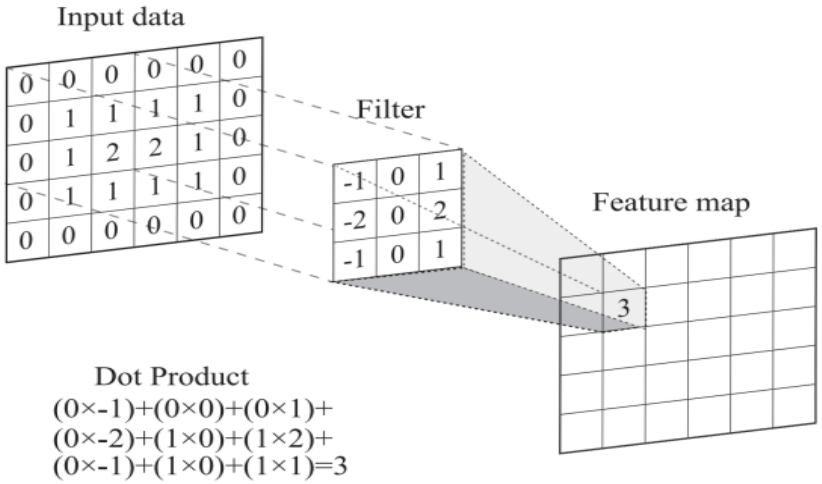

Figure 2.7: Convolution Operation .
(48)

2.5.1.2 Activation Function

Activation functions are responsible for determining if the information (weight, bias) received by a neuron from the previous layer is enough to activate the neuron. In other words, if the value falls within the threshold range specified for the cell's activation, then the output is sent to the next layer of neurons as input. The choice of activation function depends on the nature of the problem being solved, and there are several activation functions available, including: (49)

Sigmoid: is a non-linear function that produces values between 0 and 1, making it suitable for solving binary classification problems. If the weight of the value reaches the minimum threshold, it gives the value 0 and if it reaches to the upper limit, give the value (Figure 2.8) (49)

$$g(z) = \frac{1}{(1 + e^{-z})} \tag{2.2}$$

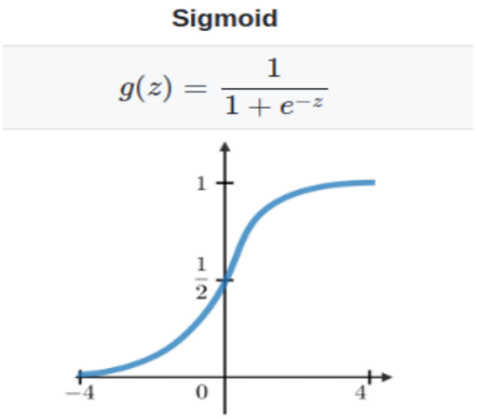

Figure 2.8: graph of sigmoid function.
(50)

Relu : A linear function for positive values and null for negative values (figure 2.7), in other words, it gives the largest value between 0 and . The Relu function cancels out negative values, and gives positive values differential importance.(49)

$$g(z) = max(0, z) \tag{2.3}$$

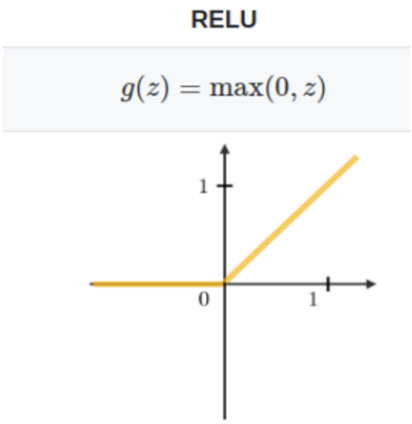

Figure 2.9: graph of ReLU function.
(50)

Tanh : A non-linear activation function is similar to the exponential function, except that its values range between -1 and 1 (figure 2.8). Therefore, negative inputs give negative results, and only zero-valued inputs are assigned to outputs close to Zero. (49)

$$g(x) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{1 - e^{-2z}}{1 + e^{-2z}} \tag{2.4}$$

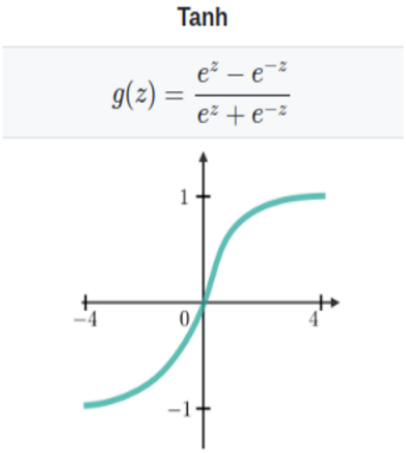

Figure 2.10: graph of Tanh function.
(50)

SoftMax: SoftMax function limits the output's function in the range [0,1]. It means that the output can be interpreted as a probability. In other words, SoftMax functions are sigmoid functions with more classes, which means that they are used to determine the probability of more classes simultaneously. (49)

2.5.1.3 pooling layer

This type of layer is usually placed between two convolutional layers. It receives multiple feature maps as input and applies a pooling operation to each of them. Pooling operations aim to reduce the size of the images while preserving their important features. To do this, the image is divided into regular cells, and the maximum value in each cell is retained.

In practice, small square cells are often used to avoid losing too much information. The most common choices are non-overlapping adjacent 2 x 2-pixel cells or 3 x 3-pixel cells separated by 2 pixels. The output has the same number of feature maps as the input but is much smaller. (51)

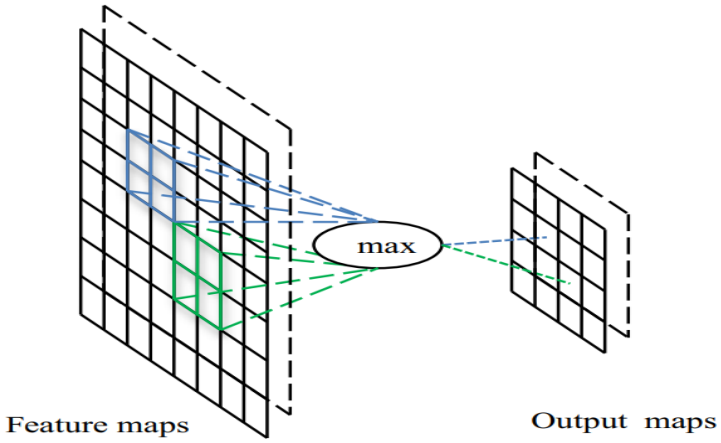

Figure 2.11: The operation of the max pooling layer. (48)

2.5.1.4 The fully-connected layer (dense layer)

A fully-connected layer can be used as the final layer in both convolutional and non- convolutional networks. This type of layer takes a vector as input and produces a new vector as output by applying a linear combination to the input values, followed by an activation function.

In classification tasks, the last fully-connected layer is used to classify input images by returning a vector of size NN, where NN is the number of classes. Each element of the vector represents the probability that the input image belongs to a particular class.(51)

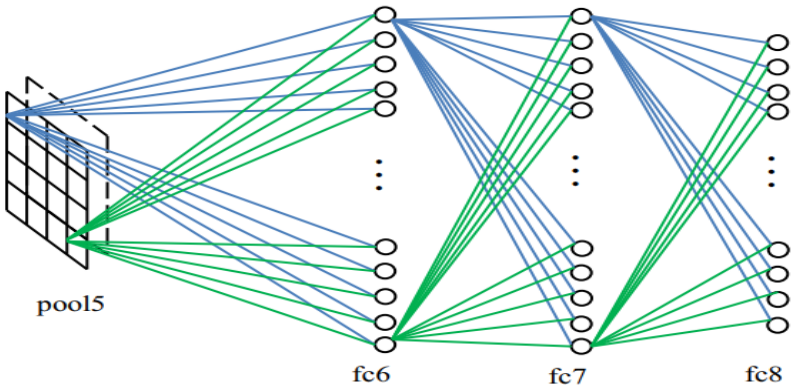

Figure 2.12: The operation of the fully-connected layer.
(48)

2.5.1.5 Loss layer

The loss layer is the final layer of the network. It computes the error between the network prediction and the true value. In classification tasks, the random variable is discrete because it can only take the values 0 or 1, representing membership (1) or non-membership (0) to a class. This is why the most common and appropriate loss function is the cross-entropy function. (51)

2.5.2 Recurrent Neural Network (RNN)

RNNs are supervised neural networks that preserve the temporal dimension of series data by using a recurrent hidden state that is updated by sequential information obtained from input time series data. The output of a sequence is dependent on this hidden state, meaning that the current output of a sequence is linked to the previous output (recurrent neural network), as shown in figure. They are based on the repeated use of classifiers, which avoids the need for a large number of historical classifiers. (52)

Additionally, this process is justified by the ability of RNNs to store in memory what has been processed in the sequence, enabling them to learn long-term dependencies in the training data. (53)

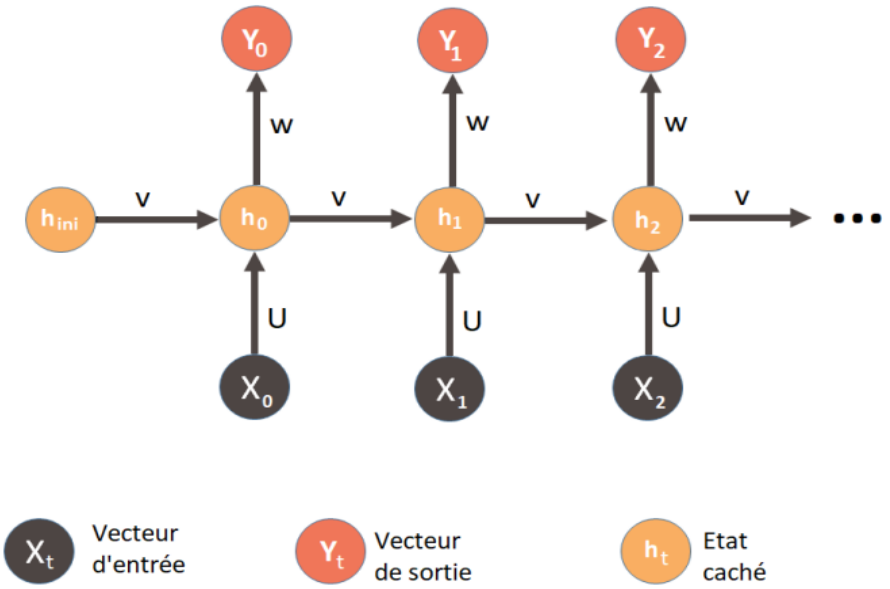

Figure 2.13: Structure of a recurrent neural network (RNN).
(54)

One of these types is the long short-term memory (LSTM) cell or unit. LSTM consists of a cell state and a carry in addition to the current word vector being processed when the sequence is being processed at each time step. The carry is responsible for ensuring that there is no loss of information during the sequential process. They can become dependent on learning and memory for long periods of time. Therefore, LSTM can retain stored information for a long period of time. (53)

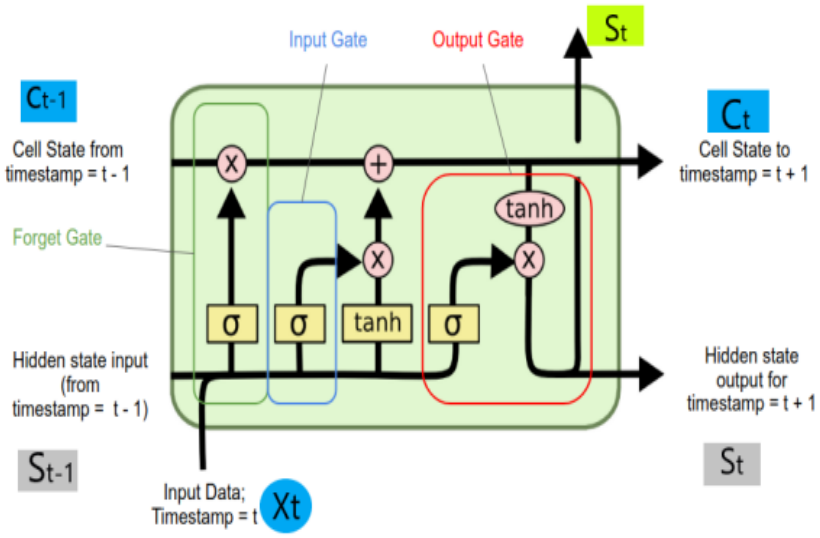

Figure 2.14: LSTM cell.
(55)

2.5.3 Generative Adversarial Networks (GAN)

The GAN model produces synthetic images that resemble the original, and then applies data augmentation and image modification techniques to enhance them. It utilizes a min-max game with a value function called V (D, G), where D is the discriminator and G is the generator.

The discriminator aims to distinguish between real and fake images, while the generator tries to produce images that can fool the discriminator. The generator produces images by adding random noise (rnv) to the input data (x), and every component of the noise vector can be considered a feature provided to the generator. The discriminator evaluates the probability of a given instance being genuine using a mathematical formula D(G(rnv)).

$$min[max[V(D,G)]] = -E_{\$\sim pd(X)}[LogD(X)] + E_{rno\sim Prno}[log(1-D(G(rnv))] \; ; \; (1)$$

$$L_{recons}^{pixel} = -E_{II \; D_{Encoder(\$),\$\sim I_{real}}}[\quad || \quad K(q) - \gamma(\$) \quad || \quad] \; ; \qquad (2)$$

In order to trick the discriminator, the generator attempts to produce images that are very close to being flawless while the discriminator strives to improve its performance by distinguishing between real and fake samples, even though it is impossible to distinguish between them. (56)

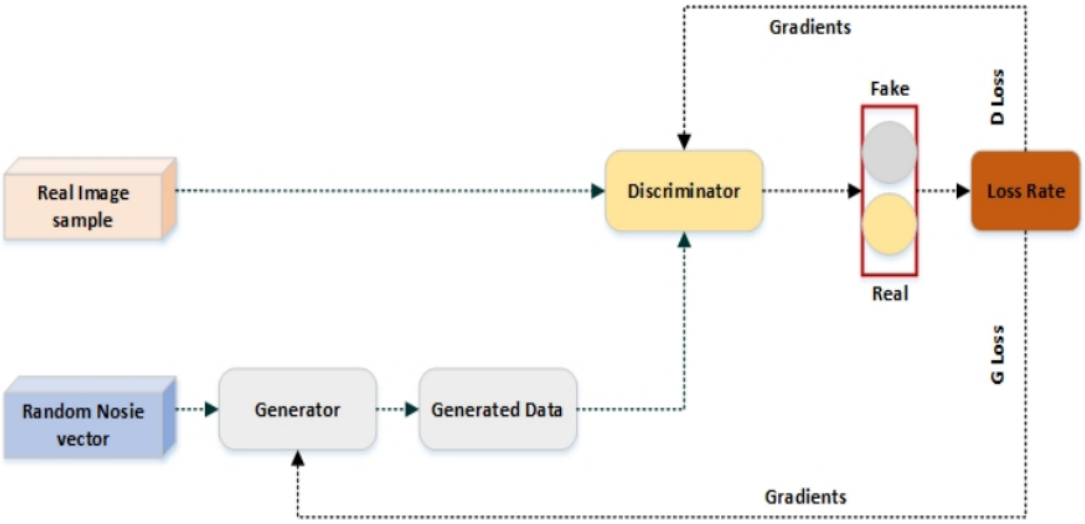

Figure 2.15: Generative adversarial networks architecture.
(57)

2.6 Categorization of traffic congestion detection approaches

There are several approaches for traffic congestion detection, among which we have mentioned image-based analysis and machine learning-based methods. We have focused on approaches based on deep learning, and reviewed works that used this approach to improve the accuracy and efficiency of traffic congestion detection.

2.6.1 Detection Based on Image Analysis

In this research paper(58), Amruta K. Dhayafule and Dr. S.R.Gengaje proposed a method for detecting traffic congestion based on texture analysis. The algorithm consists of five steps, including video frame acquisition, vehicle area calibration, GLCM calculation, feature extraction, and road congestion recognition.

The GLCM calculation is done by condensing 256 levels of gray. Feature extraction is done using equations (1) and (2) to extract energy and entropy features from the GLCM, and is used to estimate vehicle density. The characteristic S indicates the number of vehicles present in the image, and a larger S indicates more traffic congestion.

The method has been validated through experiments, and it has been shown that a non- congested road can be identified when S ¡ ST, while a road with many cars and traffic congestion can be identified when S ¿ ST . Overall, this method provides a promising approach to detecting traffic congestion using texture analysis.

The authors (59) (60) propose a traffic congestion detection system. This system utilizes the background subtraction technique using OpenCV software to detect moving objects. Subsequently, the contours of these objects are detected using the Canny algorithm. Black and white pixels are employed to differentiate between stationary and moving objects. A clustering mechanism is then utilized to predict future congestion.

For road intersections, a traffic control system model has been developed using the Haar Cascade and background subtraction techniques. The server retrieves real-time traffic images, traffic density, and other statistics.

2.6.2 Method based on Machine learning

The authors (61) propose an approach for evaluating road traffic and detecting congestion using sound sensors and machine learning. They addressed two important problems: assessing traffic conditions from audio data and analyzing audio in uncontrolled environments. By using audio data as a learning source to model traffic parameters and sound generation from passing vehicles, the authors proposed a low-cost and easy-to-deploy solution for monitoring road traffic. Results showed that the accuracy of congestion detection with sound sensors was comparable to that of video-based monitoring, without some of the problems associated with image-based monitoring.

The authors suggested that the use of sound sensors and video monitoring could be complementary and proposed avenues for improving the efficiency of their method in future work.

2.6.3 Method based on Deep learning

To overcome the problem of traffic congestion detection, many researchers have started to use a technique based on deep learning. Deep learning has been widely applied for congestion detection from traffic images. Well-known deep learning models that perform well for computer vision tasks have been adopted to detect traffic congestion : (42)

The Researchers (62) compared the effectiveness of two deep learning techniques (DCNN and YOLO) with a shallow algorithm (SVM) for detecting traffic jams from camera images. For all camera conditions, demonstrating that the models can perform well under challenging conditions.

The results show that YOLO achieves 91.5% accuracy and DCNN achieves 90.2% accuracy in detecting congestion, while SVM achieves 85.2% accuracy.YOLO achieved the highest accuracy, closely followed by DCNN, while SVM achieved comparatively lower results.

The authors (63) compared two different approaches for traffic state classification on the trafficdb dataset. The first approach utilized computer vision algorithms to extract visual features. A preliminary comparison between object detectors, performed on the GRAM Road Traffic Monitoring video dataset, revealed that YOLO v3 can be used as a real-time vehicle detector with a detection accuracy of over 80%. This was followed by classification using traditional machine learning algorithms. The second approach employed deep learning models to automatically extract and classify features.

The results indicate that the deep learning approach outperforms traditional machine learning techniques, achieving an accuracy of 98.6% for multi-class classification. However, there is still room for improvement, particularly in refining object detection algorithms and validating on more diverse traffic datasets that consider different road environments and weather conditions. Overall, this research provides promising results for implementing real-time traffic flow classification systems.

The authors (64) have introduced a promising new approach to improve the accuracy of traffic congestion classification using synthetic data augmentation techniques based on GANs and a 5-layer convolutional neural network with three pooling layers. The GAN model, introduced by Chatterjee, Subhajit et al., aims to improve image resolution by creating synthetic images that closely resemble the original, which are then further improved using data augmentation and image modification techniques.

Basically, GAN utilizes a min-max game in conjunction with a value function known as V (D, G)), where D is the discriminator and G is the generator. The generator uses random noise to produce samples, while the discriminator evaluates the probability that the samples are real or fake. Both networks aim to improve their respective performances, but are in direct opposition to each other. To trick the discriminator, the generator seeks to produce samples that are very close to perfection, while the discriminator seeks to distinguish between real and fake samples. Adding an encoder to the discriminator can help to better learn discriminative features.

The accuracy of the proposed model for the GAN-augmented dataset is 98.63%, which is better than the accuracy achieved by the model that did not use the GAN-augmented dataset (90.18%). In other words, the use of GAN augmentation significantly improved the classification results.

The authors (65) developed a technique for video traffic classification using a color-coding scheme before feeding the traffic data into a deep convolutional neural network (CNN) consisting of 4 convolution layers and 2 max-pooling layers.

Firstly, the video data is transformed into a set of image data, and a region of interest (ROI) is manually defined to minimize irrelevant external objects in the scenes. Then, the authors performed vehicle detection using the You Only Look Once algorithm.

A color-coding scheme was adopted to transform the image data set into a binary data set (the detected vehicle region is colored red, the non-detected region in white). For data augmentation, the authors randomly flipped sample images horizontally and vertically. For image classification, the binary images obtained are fed into a deep convolutional neural network.

They concluded that without color coding, the model failed to detect the validation set properly (with an accuracy of only 10%).

2.7 Conclusion

This chapter presented the main terms related to machine learning. We have presented a state of the art in using image processing and machine learning for traffic congestion detection. The research presented in this chapter highlights the significance of our problem and the continuous advancement of techniques employed to address it. In the next chapter, we will present our proposed system for detecting traffic congestion deep learning ,as well as some machine learning classifiers.

Chapter 3

The proposed system conception

3.1 Introduction

Recently, in various computer vision tasks, convolutional neural networks (CNNs) have demonstrated their ability to automatically extract high-level image features, leading to a significant improvement in classification performance (66). This makes them immensely valuable for feature extraction and classification in machine learning classifiers.

In this chapter, we present a comprehensive overview of the proposed system's architecture, the different modules of our system that enable detection, along with detailed descriptions of each module's functionalities.

3.2 System Architecture

The system architecture is a conceptual representation that describes the structure and operation of the various components and subsystems of a system.

In our study, we drew inspiration from the state of the art which highlighted the effectiveness of convolutional neural networks (CNNs) for classification. We approached this work in two scenarios:

In the first approach, we utilized a complete CNN classifier that performed both feature extraction and classification.

In the second approach, we used our CNN trained solely as a feature extraction method, and the extracted features were then fed into traditional machine learning classifiers.

By comparing these two approaches, our aim was to demonstrate the effectiveness of our CNN model in extracting informative features and performing classification tasks. To deepen our research, we employed a combination of trained CNN and traditional machine learning classifiers. Figure 3.1 presents the overall system architecture, highlighting the modules involved in traffic congestion classification.

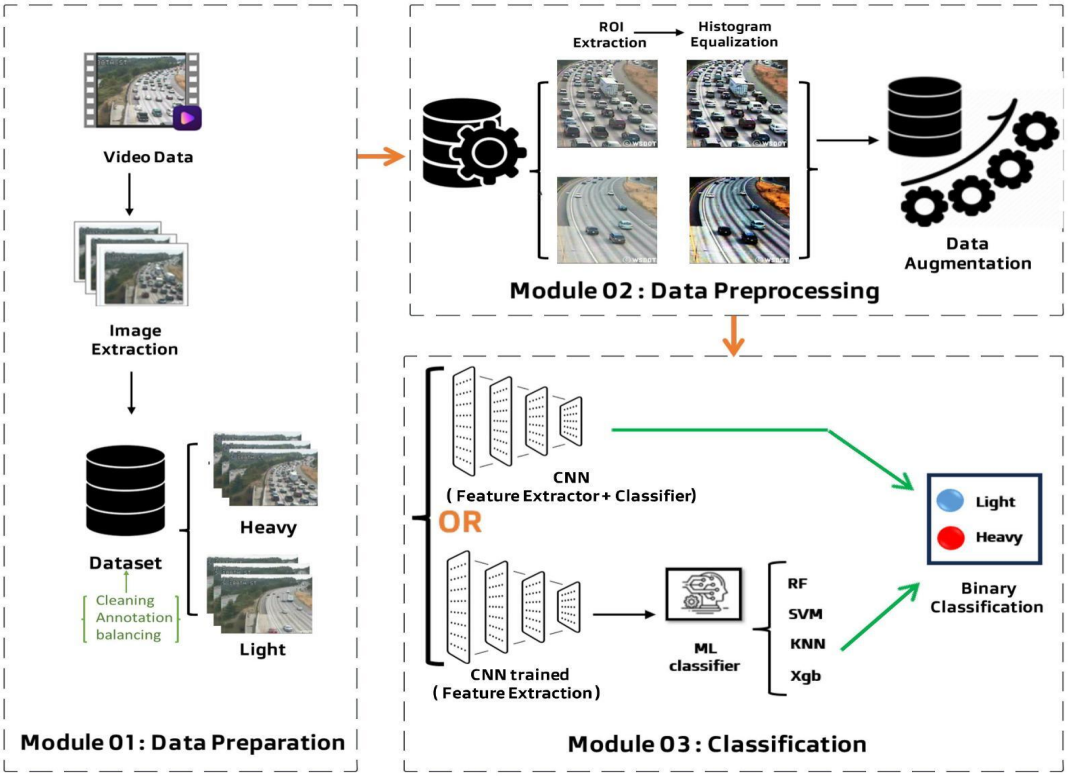

Figure 3.1: The proposed system for binary classification.

3.3 Presentation of Modules in Our System

3.3.1 Module 01: Data preparation

3.3.1.1 Images Extraction

In this part, we converted the set of videos into sequences of images, aligning with our research objective of "image classification".

3.3.1.2 Data Cleaning

In this case, we perform a manual removal of defective images, which are typically the initial frames of each video. These images may have defects due to recording issues, capture artifacts.

3.3.1.3 Data annotation

Data annotation involves assigning categories to data in order to classify or organize them based on specific criteria.

3.3.1.4 Data balancing

Data balancing plays a crucial role in optimizing model performance. By balancing the classes, we promote balanced learning, avoiding biases towards the majority class and reducing the risk of overfitting.

3.3.2 Module 02: Data preprocessing

3.3.2.1 Region of interest

In this part, the region of interest (ROI) is defined with the aim of minimizing irrelevant external objects in the scenes, focusing solely on relevant features related to the detected objects. We have selected the bottom right corner of each image to extract our region of interest, defined by a height and width of 180 respectively.

3.3.2.2 Image enhancement

Image enhancement plays a crucial role in improving visual perception for computer vision, pattern recognition, and digital image processing.(67)

To enhance the visibility of details in images, we employed a histogram equalization-based approach for image enhancement. This technique improves contrast by redistributing color levels and maximizing the dynamic range of the image. Therefore,it facilitates the identification of relevant features in the images.

3.3.2.3 Data augmentation

For reliable results, the development of a machine learning model indeed requires an adequate amount of data (68). Image augmentation is a vital technique that allows for the diversification and expansion of the training dataset by generating new images from existing ones. This technique enhances the model's ability to generalize and become more robust when presented with new images.

In this part, we used the ImageDataGenerator technique, which applies a series of transformations and manipulations to existing images in order to generate new images. These transformations can include random rotation, shifts, shear, flips, translations, etc.

3.3.3 Module 03: Classification

3.3.3.1 CNN model used for classification

In this approach, we proposed a CNN model. the hierarchy of the CNN model consists of three convolutional layers and three pooling layers. Each convolutional layer has its own trainable parameters, and Table 3.1 provides detailed information on the input and output resolutions, as well as the corresponding filters used for training. The Rectified Linear Unit (ReLU) activation function is used for each convolutional layer. The fully connected layer employs the sigmoid activation function.

Layers	***Nos of Filters***	***filter Size***	***Activation Function***	***Padding***	***Feature Map Size***	***Trainable Parameters***
Input	-	-	-	-	180x180x3	-
Conv2D(1)	16	5x5	ReLU	Same	180x180x16	1216
Batch-normalization	-	-	-	-	180x180x16	64
Max-Pooling(1)	-	-	-	-	90x90x16	-
Drop-out	-	-	-	-	90x90x16	-
Conv2D(2)	32	5x5	ReLU	Same	90x90x32	12832
Batch-normalization	-	-	-	-	90x90x32	128
Max-Pooling(2)	-	-	-	-	45x45x32	-
Drop-out	-	-	-	-	45x45x32	-
Conv2D(3)	64	5x5	ReLU	Same	45x45x64	51264
Batch-normalization	-	-	-	-	45x45x64	256
Max-Pooling(3)	-	-	-	-	22x22x64	-
Drop-out	-	-	-	-	22x22x64	-
Flatten	-	-	-	-	30976	-
Dense	-	-	-	-	1024	31720448
Drop-out	-	-	-	-	1024	-
Fully connected layer	-	-	sigmoid	-	2x3	1025
Total	-	-	-	-	-	31787009

Table 3.1: Tabular representation of the CNN model for binary traffic congestion classification.

3.3.3.2 Combining CNN and Traditional Machine Learning Classifiers

In this approach, we utilized the trained CNN model for feature extraction by removing the classification layer. We then combined it with the following four machine learning classifiers:

1. k-Nearest Neighbors (k-NN)
2. Random Forest (RF)
3. Support Vector Machines (SVM)
4. eXtreme Gradient Boosting (XGBoost)

3.4 Data Splitting

To train the CNN model, we divided the dataset into three distinct subsets:

- **Training set:** Used to train the model and enable it to learn the hidden features within the data.
- **Validation set:** Used to evaluate the model's performance during training, providing intermediate feedback on its ability to generalize.
- **Test set:** Used to impartially evaluate the final model, measuring its performance on unseen data.

When combining the trained CNN model with the machine learning algorithm, we only required the training and test datasets.

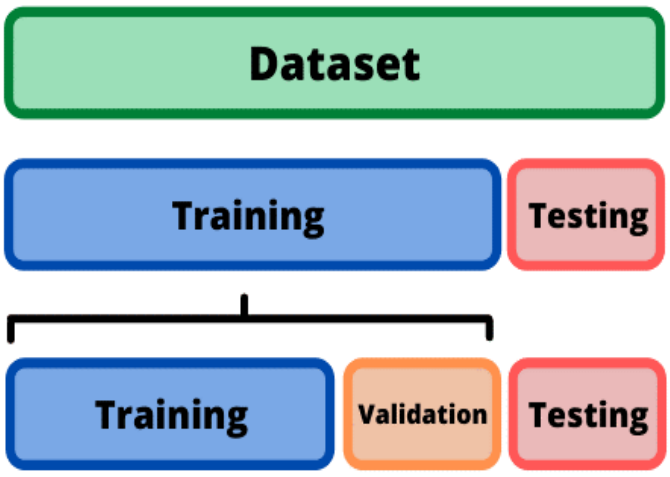

Figure 3.2: Splitting Data.
(69)

3.5 Optimization strategy

3.5.1 Optimizer

Optimizers are algorithms used to adjust the attributes of a neural network, such as weights and learning rate, in order to minimize losses. Some commonly used optimizer examples include Stochastic Gradient Descent (SGD), Batch Gradient Descent, Adam, Rmsprop, etc.

3.5.2 Batch size

Batch size is the number of training examples utilized in each iteration. It is an adjustable parameter that dictates the quantity of samples processed before updating the internal parameters of the model.

3.5.3 Learning rate

The learning rate is an adjustable hyperparameter in machine learning algorithms. It controls the speed at which the model's weights are updated during training, typically ranging between 0.0 and 1.0.

3.5.4 Number of epochs

The number of epochs plays a vital role as an important hyperparameter in the algorithm. It specifies the number of complete iterations the learning algorithm undergoes over the entire dataset during training. At each epoch, the internal model parameters are updated, allowing the model to learn from the data through multiple iterations.

3.5.5 Early stopping

Early stopping is a regularization method used during the training of machine learning models. It involves monitoring the model's performance on a validation set and stopping the training when the training loss becomes lower than the validation loss or when the training accuracy becomes lower than the validation loss.

3.6 Regularization strategies

3.6.1 Batch normalization

Batch normalization is a technique used in the field of deep learning to prevent overfitting. Its main objective is to address the issue of internal covariate shift and improve the stability and performance of neural networks.

3.6.2 Dropout

Dropout is a technique used in neural networks to prevent overfitting and improve generalization capacity. It involves randomly deactivating certain neurons during training, which prevents their interdependence and makes the network more robust.

3.7 Evaluation Metrics

3.7.1 Confusion Matrix

The confusion matrix provides a summary of a classification model's performance in predicting examples belonging to different classes. It is represented as an n x n table, where n is the number of classes. One axis of the matrix corresponds to the predicted labels by the model, while the other axis represents the true labels. The figure illustrates a confusion matrix for n = 2, where the entries have the following interpretations:(70)

- **TP (true positives):** Represents the correct prediction of positive values.
- **TN (true negatives):** Represents the correct prediction of negative values.
- **FP (false positives):** Represents the incorrect prediction of positive values as negative.
- **FN (false negatives):** Represents the incorrect prediction of negative values as positive.

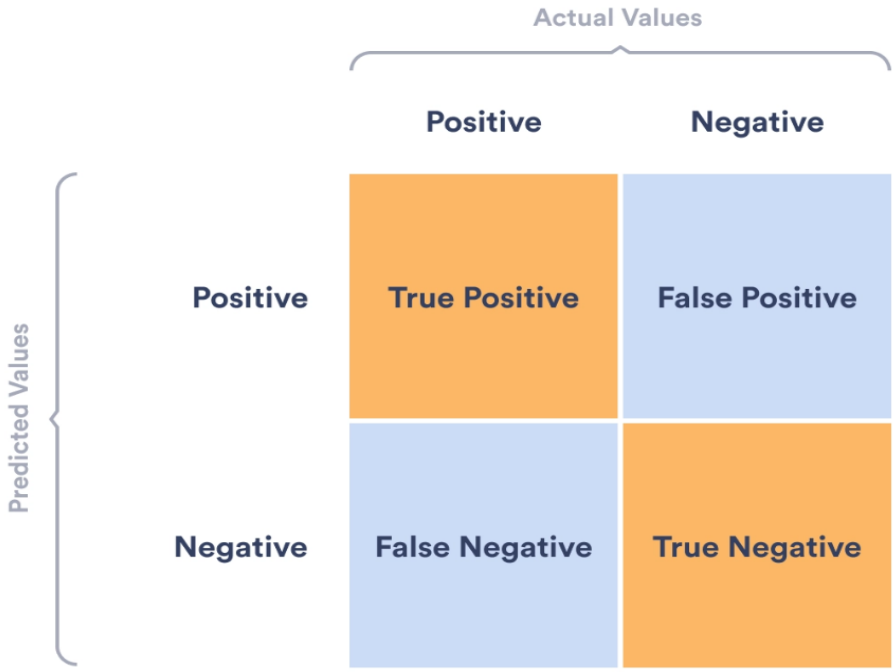

Figure 3.3: Confusion matrix for binary classification.
(71)

Accuracy: The accuracy (Acc) is the ratio of number of correct predictions to the total number of input samples:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} .$$

Recall: The recall (R) measures the percentage of non-functional requirements that were correctly retrieved and categorized:

$$R = \frac{TP}{TP+FN} \ .$$

Precision : The precision (P) as the proportion of corrected predicted classifications against all predictions against the classification under test:

$$P = TP/(TP + FP) \ .$$

F1-Score : The F1 measure is the harmonic mean of precision and recall, giving an equal weight to both elements:

$$F_1 = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R} \ .$$

3.8 Conclusion

In this chapter, we have provided a detailed overview of our proposed system for detecting traffic congestion from images. Our system employs two scenarios: the first one utilizes the CNN model as a classifier, and the second scenario utilizes the CNN as a feature extractor. We further combine these features with SVM, KNN, RF, and Xbg classifiers. The next chapter will be dedicated to the implementation of the systems and the discussion of their results.

Chapter 4

Implementation and discussion of the results

4.1 Introduction

This chapter presents the implementation steps of the system proposed in the previous chapter, including the dataset used and the techniques applied during the implementation phase. It also presents the results obtained from the experiments, which will be discussed and compared with other works.

4.2 Hardware Technologies

A physical machine from Dell was used in this project, featuring an Intel(R) Core(TM) i5-8350U CPU and 8.00 GB of RAM.

4.3 Software Technologies

4.3.1 Programming Language

4.3.1.1 Python

We chose Python as a programming language because of its stability, flexibility, and reliability. It is an interpreted, object-oriented, high-level language with dynamic semantics. Python's syntax is simple and easy to learn, prioritizing readability and reducing program maintenance costs. It has a module and package management system that facilitates program modularity and code reuse. As a high-level programming language, Python is versatile and enjoys great popularity. (72)

Figure 4.1: Python.

4.3.2 Development environment

4.3.2.1 Kaggle

Kaggle is an online platform for data scientists and machine learning enthusiasts. It provides tools for collaboration, dataset access and sharing, GPU-powered notebooks, and data science competitions. It was founded in 2010 by Anthony Goldbloom and Jeremy Howard and later acquired by Google in 2017. With over 8 million registered users, Kaggle aims to support professionals and students in achieving their data science goals. (73)

Figure 4.2: Kaggle.

4.3.3 Libraries

4.3.3.1 TensorFlow

TensorFlow is an open-source library dedicated to Machine Learning, designed by Google to simplify the solving of complex mathematical problems. This toolkit enables the development and execution of Deep Learning and Machine Learning applications with great ease. This library has been helpful in our work during the training and execution phase of neural networks. (74)

Figure 4.3: TensorFlow.

4.3.3.2 Karas:

Keras is a deep learning library that provides minimal yet highly productive functionalities. It is written in Python and runs on top of TensorFlow, a machine learning platform. Keras is designed to enable fast experimentation and offers a productive interface for solving various machine learning problems. It focuses on a modern approach to deep learning. A major advantage of Keras is that it can take a developer's idea and guide them towards concrete results.(75)

Figure 4.4: Karas.

4.3.3.3 NumPy

NumPy is a fundamental library for scientific computing in Python and forms the foundation of SciPy. It enables manipulation of multidimensional arrays and

performs mathematical operations on these arrays. NumPy offers a wide variety of functions, including creating arrays from files, saving them, and performing operations on vectors, matrices, and polynomials. (76)

Figure 4.5: NumPy.

4.3.3.4 Matplotlib

Matplotlib is a Python programming library that enables plotting and visualization of data in graphical form. It can be used in conjunction with the NumPy and SciPy libraries for scientific computations in Python. Matplotlib provides an object-oriented API that allows easy integration of plots into applications using popular graphical user interface tools such as Tkinter,wxPython,Qt,or GTK. (77)

Figure 4.6: Matplotlib.

4.3.3.5 Scikit-learn

Scikit-learn is a powerful and robust Python library for machine learning, widely used in industry and research. It provides a comprehensive range of efficient tools for automatic learning and statistical modeling, covering various tasks such as classification, regression, and clustering. (78)

It offers a wide selection of commonly used classifiers in our study. It also offers a wide range of additional tools for model development, selection, evaluation, and data preprocessing.

Figure 4.7: Scikit-learn.

4.3.3.6 OpenCV

OpenCV is an open-source library for computer vision and machine learning. It provides a common infrastructure for computer vision applications, with over 2500 optimized algorithms. These powerful algorithms are designed for real-time image processing.(79) In our work, we utilize this library for image enhancement tasks such as equalizeHist and Region of Interest (ROI) operations.

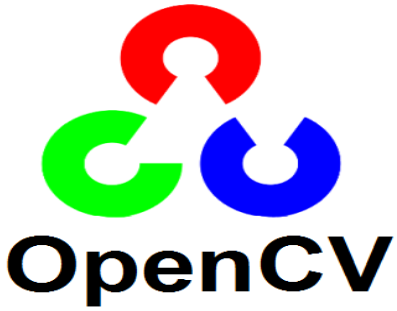

Figure 4.8: OpenCV.

4.3.4 Other Tools

4.3.4.1 Overleaf

Overleaf is an online, real-time collaborative LaTeX editor that allows multiple people to work simultaneously on a document. It supports almost all LaTeX features, including inserting images, bibliographies, equations, and much more. We used it for the writing and drafting of our thesis.(80)

Figure 4.9: Overleaf.

4.3.4.2 Lucidchart

Lucidchart is a cloud-based platform that enables online collaboration for creating diagrams, visualizing data, and designing conceptual schemas. We used this platform to describe the complete architecture of the system in our proposal. (81)

4.3.4.3 Video to JPG converter

A video to JPG converter is a tool used to convert a video into a series of images in JPG format. We utilized it for extracting images from our database.(82)

4.4 TrafficDB dataset

In our experimental part, we utilized the 'trafficd' (83) dataset, which includes videos representing traffic congestion situations. These videos were captured over a two-day period from a fixed camera positioned above the I-5 highway in Seattle, Washington. The original dataset consists of three different congestion levels: high, medium, and light, and it includes a total of 254 videos. The figure below displays the number of videos for each congestion level:

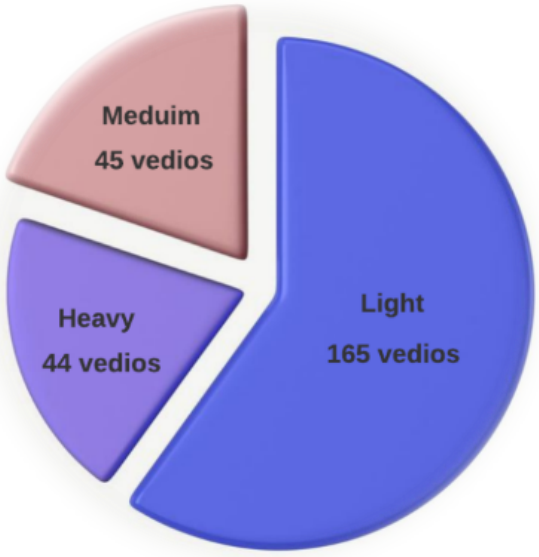

Figure 4.10: Number of videos for each congestion level..

In this study, after extracting images from the video dataset, for a binary classification between the "heavy" (high congestion) and "light"(low congestion) classes, excluding the "medium" class. Each image was labeled as either "heavy" or "light". We obtained a total of 2298 congested images and over 8000 non-congested images. Due to class imbalance, we selected a subset of only 2298 images for each class. The description of this version is shown in Table 4.1

Image class	**Number of images**
Light	2298
Heavy	2298
Total	4596

Table 4.1: Description of the TrafficDB dataset.

4.4.1 Dataset Visualization

The figure ... below presents a sample of images from the TrafficDB dataset before they were preprocessed.

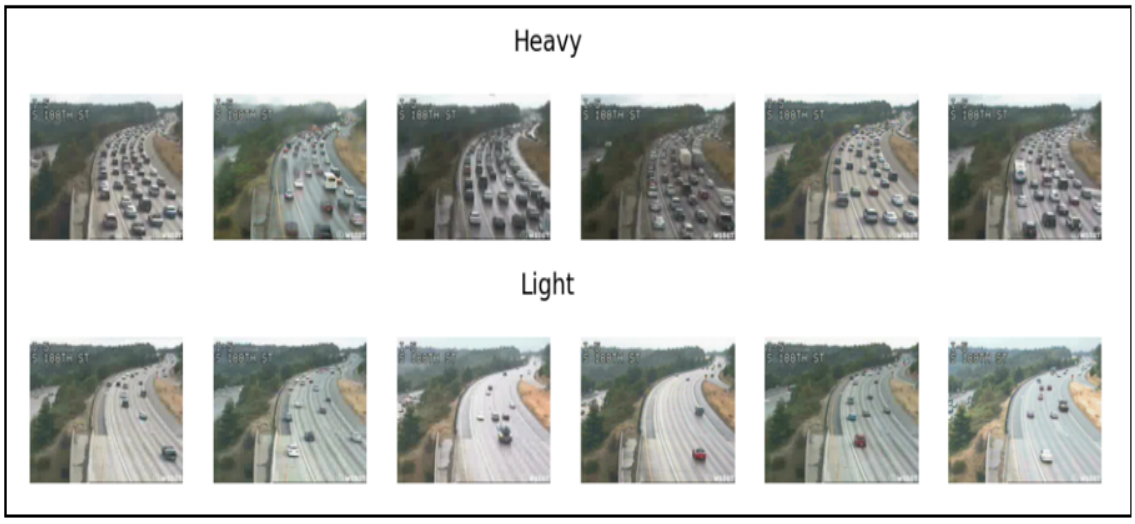

Figure 4.11: Dataset Visualisation.

4.4.2 TrafficDB preprocessing

We preprocessed our dataset using two different preprocessing techniques: Firstly, we defined a region of interest (ROI) to exclude irrelevant external objects and focus on the traffic congestion area. By selecting the bottom corner of the image, we concentrate on the most important part for congestion detection. The original images are sized at 320*240, and with the ROI selection, they become 180*180. This approach reduces the model's complexity, prevents memory issues, and improves classification accuracy by focusing solely on congestion-related features. Figure... illustrates an example for the selection of the interest region (ROI) on image.

Figure 4.12: ROI extraction.

Secondly, we applied a histogram equalization-based method to enhance the images. This approach improves the contrast and visibility of details in the used images. By redistributing the color levels, it facilitates the identification of relevant features associated with traffic congestion. Figure 4.13 depicts the visualization of the data after preprocessing (region of interest selection and histogram equalization).

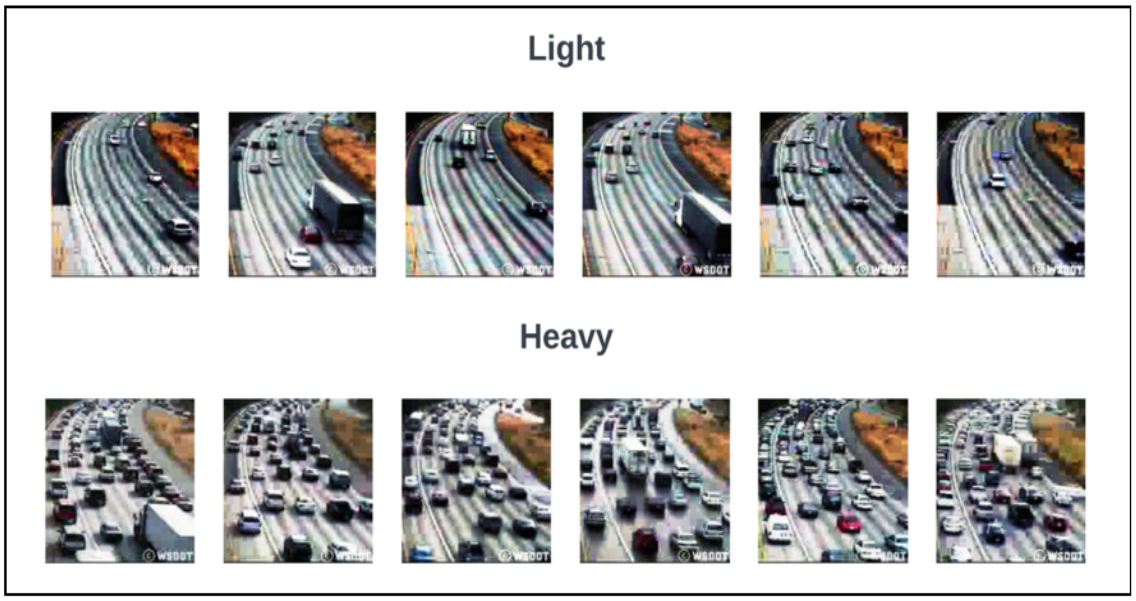

Figure 4.13: Dataset Visualization after preprocessing.

4.4.3 Dataset Splitting

To train, valid and test the CNN model, we divided our dataset into three distinct groups in a stratified manner. This division was done with the purpose of avoiding overfitting, ensuring that our model is evaluated impartially on independent data. For the combination of the CNN with machine learning classifiers, we used only the training set and the test set.

Initially, we divided the dataset into two distinct parts: the training set and the test set, with respective percentages of 90% and 10%. Subsequently, we set aside 10% of the training set for validation.

The above table 4.2 presents the distribution of the image dataset for training, validation, and testing.

Image class	**Training**	**validation**	**Testing**
Heavy	1861	207	230
Light	1861	207	230
Total	3722	414	460

Table 4.2: The Trafficdb dataset after the division.

4.4.4 Data augmentation

To enhance the quantity and diversity of our training data, we utilized the ImageDataGenerator method for data augmentation. This technique aims to improve the robustness and performance of our model by providing a wider range of training examples.

Firstly, the training set underwent data normalization, where pixel values were scaled between 0 and 1. Subsequently, the resulting normalized images were aug-

mented by applying rotations, horizontal and vertical flips to generate mirror variations of the images, as well as shear range transformations.

Figure 4.14 visualizes some of the transformations applied by the ImageDataGenerator method, while Table 4.3 displays the resulting increase in the number of augmented images.

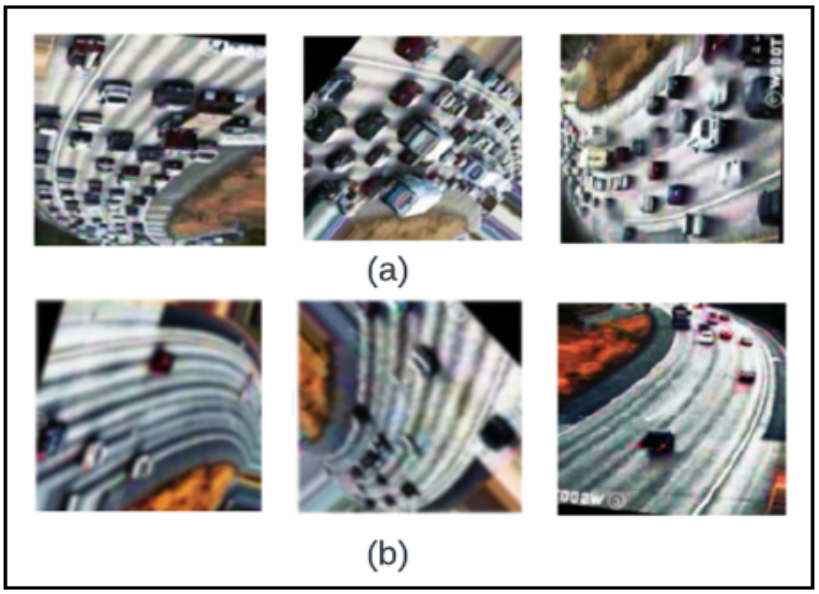

Figure 4.14: (a) Transformed sample images of congestion.(b) Transformed sample images of no congestion.

set	**Original images**	**Images augmented by ImageDataGenerator**	**Total images**
Training	3722	14884	18606

Table 4.3: The number of images in the training set before and after augmentation.

4.5 CNN Model Implementation

We trained our convolutional neural network (CNN) model for 50 epochs with early stopping callbacks. By applying early stopping with a patience of 6, we were able to prevent overfitting while minimizing the required training time for our CNN model. Firstly, we compared two batch sizes, and based on the results obtained in Table 4.4, we continued the experiments with a batch size of 32.

Batch size	**Accuracy**
32	98,69
64	95,43

Table 4.4: Performance Comparison for Different Batch Sizes.

Secondly, we compared three optimizers, ADAM, RMSprop, and SGD, with a learning_rate of 0.0001 and momentum of 0.9. Table 4.5 highlights the obtained results, confirming the effectiveness of the SGD optimizer.

Optimizer	**Accuracy**
ADAM	76%
SGD	98,69%
RmsProp	82,4%

Table 4.5: Accuracy of CNN with three types of optimizers.

After several attempts, we have fixed the chosen hyperparameters for training the CNN, as illustrated in Table 4.6.

Batch size	**Learning rate**	**Epochs**	**Early stopping**	**optimzer**	**Momentum**
32	0,0001	50	6	SGD	0,9

Table 4.6: The final hyperparameters chosen for our model.

4.5.1 Results and discussion

Figure 4.15 shows a graphical illustration of the loss and accuracy observed throughout the training and validation. It can be observed that the CNN model trains until it stops without reaching the 50 epochs by applying the Early Stopping Callback which is set to 06.

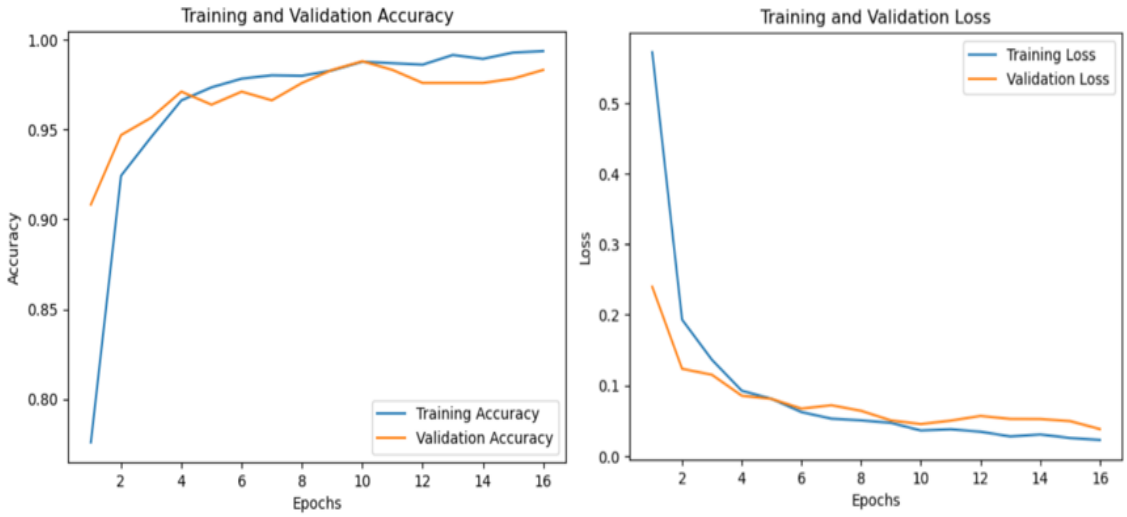

Figure 4.15: (Left) Training and validation accuracy (Right) Training and validation loss.

As can be seen from the table below the obtained results for accuracy and loss are as follows:

Classification type	Accuracy		Loss	
	Training	Validation	Training	Validation
Our model	99,36%	98,31%	0,0227	0.0382

Table 4.7: The results for accuracy and loss.

The confusion matrix provided by the CNN model is reported in Table 4.7 below:

CNN model	**Heavy (Pred)**	**Light(pred)**
Heavy	0,974	0,026
Light	0	1

Table 4.8: Confusion matrix for the CNN.

Based on the previous table, we can observe that all samples from the 'Light' class were correctly predicted by the CNN. However, we also notice some misclassifications, with a certain number of 'Heavy' samples being wrongly classified as 'Light'.

The following table 4.8 summarizes the classification results obtained for traffic congestion using the proposed CNN model as a classifier:

model	**Precesion**	**Recall**	**F1Score**	**Accuracy**
CNN model	97,45%	100%	98,71%	98,69%

Table 4.9: Traffic congestion classification results using the CNN model.

4.6 The implementation of combining CNN with machine learning algorithms

First, we trained our CNN model. The results are shown in the table 4.8 Then, we used the CNN as a feature extractor instead of a classifier, and combined it with the following four machine learning algorithms: SVM, KNN, XGBoost, and Random Forest.

In order to adjust and optimize the machine learning algorithms used in our approach, we utilized the key parameters mentioned in the table 4.9 .

Algorithms	Parameters	
SVM	C=10 ,	$kernal = rbf$
XGB	n-estimators=100 ,	$max - depth = 3$
RF	n-estimators=100 ,	$max - depth = 5$
KNN	$n_neighbors = 7$,	weights =uniform

Table 4.10: Classification results of traffic congestion using CNN as feature extractor combined with machine learning classifiers.

4.6.1 Results and discussion

The confusion matrices provided by the combination of CNN and machine learning classification algorithms are reported in the tables below:

CNN + SVM	**Heavy (Pred)**	**Light(pred)**
Heavy	1	0
Light	0,004	0,995

Table 4.11: Confusion matrix for the CNN and SVM.

CNN + XGB	**Heavy (Pred)**	**Light(pred)**
Heavy	0,995	0,004
Light	0,008	0,991

Table 4.12: Confusion matrix for the CNN and XGB.

CNN + RF	**Heavy (Pred)**	**Light(pred)**
Heavy	1	0
Light	0,008	0,991

Table 4.13: Confusion matrix for the CNN and RF.

CNN + KNN	**Heavy (Pred)**	**Light(pred)**
Heavy	1	0
Light	0,004	0,995

Table 4.14: Confusion matrix for the CNN and KNN.

According to the previous tables, the confusion matrices provided by the combination of CNN and machine learning classification algorithms show that all samples from the 'Heavy' class were correctly predicted by CNN+KNN, CNN+SVM, and CNN+RF. However, we observe some misclassifications, with a certain number of 'Heavy' samples being wrongly classified as 'Light' by CNN+XGB.

Additionally, we can observe that the majority of samples from the 'Light' class were correctly predicted by all machine learning classifiers. However, we also notice some misclassifications, with a certain number of 'Light' samples being wrongly classified as 'Heavy'.

The following table summarizes the results of classifying traffic congestion images using the combination of CNN with machine learning algorithms. We observe that the best results are obtained by CNN+SVM and CNN+KNN. We notice that the highest accuracy of 99.78is achieved by both classifiers, CNN+SVM and CNN+KNN.

Model	Accuracy	Precision	Recall	F1-Score
CNN+SVM	99,782%	99,783%	99,782%	99,782%
CNN+KNN	99,782%	99,783%	99,782%	99,782%
CNN+RF	99,565%	99,568%	99,565%	99,565
CNN+Xgb	99,347%	99,348%	99,347%	99,347%

Table 4.15: Classification results of traffic congestion using CNN as feature extractor combined with machine learning classifiers.

4.7 Comparison between the results of our models and the baseline

We take [2022](65) as the baseline for our work, as they utilize TrafficDB as the dataset. In their study, the authors [2022](65) proposed an approach for road congestion classification using a color-coding scheme on image data. This color-coding scheme was adopted to transform the image dataset into a binary dataset before feeding it into a convolutional neural network (CNN).

Model	Accuracy	Precision	Recall	F1-Score
CNN+SVM	**99,782**%	**99,783**%	**99,782**%	**99,782**%
CNN+KNN	**99,782**%	**99,783**%	**99,782**%	**99,782**%
CNN+RF	99,565%	99,568%	99,565%	99,565%
CNN+Xgb	99,347%	99,348%	99,347%	99,347%
CNN	98,69%	97,45%	100%	98,69 %
[2022] (65)	98,2%	98,2%	95,6%	_

Table 4.16: Comparison between the results of the models.

Looking at the table, we can observe that our proposed models, CNN+SVM and CNN+KNN, achieve superior results (99.78%) compared to the baseline (98.2%). Furthermore, we notice that all our models using CNN as a classifier achieved an accuracy of 98.69%. Additionally, the combinations of CNN+RF and CNN+XGB outperformed the baselines, achieving accuracies of 99.56% and 99.32%, respectively.

In summary, our overall results demonstrate:

- A significant improvement compared to the baseline, with superior performance in all proposed models.
- The CNN proves to be both an effective feature extractor and a high-performing classifier.
- The combination of CNN as a feature extractor with machine learning classifiers enables more accurate and reliable results in road congestion classification.

4.8 Conclusion

In this chapter, we provided a comprehensive overview of the implementation process of our proposed system for detecting traffic congestion from images. We discussed in detail the dataset preparation, the techniques applied, and the tools utilized in our work. Furthermore, we presented and analyzed the results obtained at each stage of the implementation. The discussions provided valuable insights into the performance and effectiveness of our system in detecting traffic congestion.

General conclusion

Our study explored the use of machine learning, specifically deep learning, using Convolutional Neural Networks (CNN) to classify road congestion images. We also tested the effectiveness of combining CNN with other machine learning classifiers. Two approaches were implemented, one using only CNN and the other combining CNN with classifiers such as SVM, KNN, RF, and XGB.

In the experimental phase, the results showed that the hybrid approach, particularly the combination of CNN and SVM, as well as CNN + KNN, was the most effective, achieving an accuracy score of 99.78%, which surpasses some baseline works using the same dataset (TrafficDB).

We then analyzed and discussed these results, attempting to provide an interpretation of our observations. In summary, our research demonstrated that the approach combining CNN and SVM , as well as CNN and KNN, proved to be particularly effective, offering promising results for road congestion detection.

Future Perspectives

To conclude this thesis, the obtained results are encouraging and competitive. In terms of future perspectives for this modest work, we emphasize the following points:

- Explore the use of GANs to generate even more realistic samples from latent vectors, thereby improving the quality of the generated data.
- Utilize alternative combinations of data augmentation methods to diversify and enrich the dataset, aiming to enhance performance.
- Further explore data enhancement techniques to strengthen the system's performance.
- Combine multiple models to improve overall accuracy and performance of the system.

Bibliography

[1] A. Khalfi and M. Guerroumi, "Transfer learning with image data augmentation for parking occupancy detection," in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)*, pp. 1–4, IEEE, 2023.

[2] U. Jilani, M. Asif, M. Rashid, A. A. Siddique, S. M. U. Talha, and M. Aamir, "Traffic congestion classification using gan-based synthetic data augmentation and a novel 5-layer convolutional neural network model," *Electronics*, vol. 11, no. 15, p. 2290, 2022.

[3] C.-Y. Wu, M.-B. Hu, R. Jiang, and Q.-Y. Hao, "Effects of road network structure on the performance of urban traffic systems," *Physica A: Statistical Mechanics and its Applications*, vol. 563, p. 125361, 2021.

[4] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[5] M. Sandeli, "Traitement d'images par des approches bio-inspirées application à la segmentation d'images," *Université Constantine*, vol. 2, p. s1, 2014.

[6] M. Bergounioux, "Quelques méthodes de filtrage en traitement d'image," 2010.

[7] A. J. Qasim and F. Q. A. Alyousuf, "History of image digital formats using in information technology," *QALAAI ZANIST JOURNAL*, vol. 6, no. 2, pp. 1098–1112, 2021.

[8] L. Selsabile, "Compression of color image by dct," 2022.

[9] A. MESSAOUDI, *Contribution à l'amélioration et la mise en oeuvre de nouveaux algorithmes pour la compression des signaux*. PhD thesis, Université de Batna 2, 2017.

[10] `https://fr.mathworks.com/help/images/image-types-in-the-toolbox.html`, April 2023.

[11] F. REGRAGUI, A. TOUZANI, N. ZAHID, and A. TAMTAOUI, "Outils de compression et de crypto-compression: Applications aux images fixes et video,"

[12] N. P. Galatsanos, C. A. Segall, and A. K. Katsaggelos, "Digital image enhancement," *Encyclopedia of optical engineering*, pp. 388–402, 2003.

[13] D. Attia, *Segmentation d'images par combinaison adaptative couleur-texture et classification de pixels.: Applications à la caractérisation de l'environnement de réception de signaux GNSS*. PhD thesis, Université de Technologie de Belfort-Montbeliard, 2013.

[14] T. G. Hawarden, S. Leggett, M. B. Letawsky, D. R. Ballantyne, and M. M. Casali, "Jhk standard stars for large telescopes: the ukirt fundamental and extended lists," *Monthly Notices of the Royal Astronomical Society*, vol. 325, no. 2, pp. 563–574, 2001.

[15] S. Blanchard, D. Rousseau, D. Gindre, and F. Chapeau-Blondeau, "Transmission d'image assistée par le bruit en imagerie cohérente et incohérente," in *Proceedings*, 2007.

[16] A. KAAZAOUI and K. KAAZAOUI, "La fusion d'image multifocale," 2018.

[17] M. Bergounioux, *Introduction au traitement mathématique des images-méthodes déterministes*, vol. 76. Springer, 2015.

[18] `https://www.cronj.com/blog/geometric-transformation-images-affine-transformatio`

[19] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, pp. 1–27, 2022.

[20] J. Won, G. L. Monroy, R. I. Dsouza, D. R. Spillman Jr, J. McJunkin, R. G. Porter, J. Shi, E. Aksamitiene, M. Sherwood, L. Stiger, *et al.*, "Handheld briefcase optical coherence tomography with real-time machine learning classifier for middle ear infections," *Biosensors*, vol. 11, no. 5, p. 143, 2021.

[21] R. R. Chand, M. Farik, and N. A. Sharma, "Digital image processing using noise removal technique: A non- linear approach," in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–5, IEEE, 2022.

[22] S. BENFRIHA and A. HAMEL, "Segmentation d'image par coopération région-contours," 2016.

[23] C. Kavya *et al.*, "Performance analysis of different filters for digital image processing," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 2, pp. 2572–2576, 2021.

[24] `https://glq2200.clberube.org/chapitres/docs/signal-filtres`, April 2023.

[25] J. Hosdez, *Fissuration par fatigue de fontes à graphite sphéroïdal et vermiculaire: caractérisation des effets de la plasticité et d'un vieillissement thermique.* PhD thesis, Thèse de Doctorat, Centrale Lille, p 35, 2017.

[26] R. Yogamangalam, B. Karthikeyan, *et al.*, "Segmentation techniques comparison in image processing," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 1, pp. 307–313, 2013.

[27] J. Landré, *Analyse multirésolution pour la recherche et l'indexation d'images par le contenu dans les bases de données images-Application à la base d'images paléontologique Trans' Tyfipal.* PhD thesis, Université de Bourgogne, 2005.

[28] M. Sarifuddin, R. Missaoui, J. Vaillancourt, Y. Hamouda, and M. Zaremba, "Analyse statistique de similarité dans une collection d'images," *Revue des Nouvelles Technologies de l'Information*, vol. 1, no. 1, pp. 239–250, 2003.

[29] M. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pp. 503–507, IET, 2013.

[30] B. Li, E. Chang, and Y. Wu, "Discovery of a perceptual distance function for measuring image similarity," *Multimedia systems*, vol. 8, no. 6, pp. 512–522, 2003.

[31] M. Les, "Les modes de régulation de l'intelligence artificielle par le droit européen: entre droit souple et droit dur,"

[32] H. Castañé, G. Baiges-Gaya, A. Hernández-Aguilera, E. Rodríguez-Tomàs, S. Fernández-Arroyo, P. Herrero, A. Delpino-Rius, N. Canela, J. A. Menendez, J. Camps, *et al.*, "Coupling machine learning and lipidomics as a tool to investigate metabolic dysfunction-associated fatty liver disease. a general overview," *Biomolecules*, vol. 11, no. 3, p. 473, 2021.

[33] J. M. Helm, A. M. Swiergosz, H. S. Haeberle, J. M. Karnuta, J. L. Schaffer, V. E. Krebs, A. I. Spitzer, and P. N. Ramkumar, "Machine learning and artificial intelligence: definitions, applications, and future directions," *Current reviews in musculoskeletal medicine*, vol. 13, pp. 69–76, 2020.

[34] M. P. S. Prasad, D. Senthilrajan, and D. Senthilrajan, "Leaf features extraction for plant classification using cnn," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 2, no. 2, pp. 148–154, 2021.

[35] H. Tran, "A survey of machine learning and data mining techniques used in multimedia system," *no*, vol. 113, pp. 13–21, 2019.

[36] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[37] A. I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, and A. El-Shafie, "Extreme gradient boosting (xgboost) model to predict the groundwater levels in selangor malaysia," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 1545–1556, 2021.

[38] J. Abualdenien and A. Borrmann, "Ensemble-learning approach for the classification of levels of geometry (log) of building elements," *Advanced Engineering Informatics*, vol. 51, p. 101497, 2022.

[39] S. B. Imandoust, M. Bolandraftar, *et al.*, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *International journal of engineering research and applications*, vol. 3, no. 5, pp. 605–610, 2013.

[40] T. Ahmad, M. S. Faisal, A. Rizwan, R. Alkanhel, P. W. Khan, and A. Muthanna, "Efficient fake news detection mechanism using enhanced deep learning model," *Applied Sciences*, vol. 12, no. 3, p. 1743, 2022.

[41] L. Abhishek, "Optical character recognition using ensemble of svm, mlp and extra trees classifier," in *2020 International Conference for Emerging Technology (INCET)*, pp. 1–4, IEEE, 2020.

[42] D. Jakhar and I. Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences," *Clinical and experimental dermatology*, vol. 45, no. 1, pp. 131–132, 2020.

[43] E. L. Manibardo, I. Laña, and J. Del Ser, "Deep learning for road traffic forecasting: Does it make a difference?," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6164–6188, 2021.

[44] S. IBTISSAM, "La prédiction de flux de trafic routier par une méthode d'apprentissage profond," 2020.

[45] J. Kurniawan, S. G. Syahra, C. K. Dewa, *et al.*, "Traffic congestion detection: learning from cctv monitoring images using convolutional neural network," *Procedia computer science*, vol. 144, pp. 291–297, 2018.

[46] N. Kumar and M. Raubal, "Applications of deep learning in congestion detection, prediction and alleviation: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 133, p. 103432, 2021.

[47] K. Akram, *Classification des images utilisant les réseaux de neurones de convolution*. PhD thesis, Faculté des Sciences et Technologies, 2021.

[48] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Convolutional neural networks," in *Machine learning*, pp. 173–191, Elsevier, 2020.

[49] M. A. Mercioni and S. Holban, "The most used activation functions: Classic versus current," in *2020 International Conference on Development and Application Systems (DAS)*, pp. 141–145, IEEE, 2020.

[50] K. A. L. Conza, "Deep learning for decision support in dermatology,"

[51] K. Akram, *Classification des images utilisant les réseaux de neurones de convolution*. PhD thesis, Faculté des Sciences et Technologies, 2021.

[52] A. Boukerche, Y. Tao, and P. Sun, "Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems," *Computer networks*, vol. 182, p. 107484, 2020.

[53] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid cnn-rnn based deep learning approach," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100007, 2021.

[54] `https://lbourdois.github.io/blog/nlp/RNN-LSTM-GRU-ELMO/`, April 2023.

[55] `https://medium.com/analytics-vidhya/introduction-to-long-short-term-memory-lstm` April 2023.

[56] S. Chatterjee, D. Hazra, Y.-C. Byun, and Y.-W. Kim, "Enhancement of image classification using transfer learning and gan-based synthetic data augmentation," *Mathematics*, vol. 10, no. 9, p. 1541, 2022.

[57] S. Chatterjee, D. Hazra, Y.-C. Byun, and Y.-W. Kim, "Enhancement of image classification using transfer learning and gan-based synthetic data augmentation," *Mathematics*, vol. 10, no. 9, p. 1541, 2022.

[58] A. K. Dhayafule and S. Gengaje, "Road congestion detection using image texture analysis," 2021.

[59] A. Rao, A. Phadnis, A. Patil, T. Rajput, and P. Futane, "Dynamic traffic system based on real time detection of traffic congestion," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, IEEE, 2018.

[60] S. Kulkarni, S. Ugale, H. Jawale, and A. Shinde, "Review on traffic congestion detection using image processing,"

[61] R. C. Gatto and C. H. Q. Forster, "Audio-based machine learning model for traffic congestion detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7200–7207, 2020.

[62] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transportation Research Record*, vol. 2672, no. 45, pp. 222–231, 2018.

[63] D. Impedovo, F. Balducci, V. Dentamaro, and G. Pirlo, "Vehicular traffic congestion classification by visual features and deep learning approaches: a comparison," *Sensors*, vol. 19, no. 23, p. 5213, 2019.

[64] U. Jilani, M. Asif, M. Rashid, A. A. Siddique, S. M. U. Talha, and M. Aamir, "Traffic congestion classification using gan-based synthetic data augmentation and a novel 5-layer convolutional neural network model," *Electronics*, vol. 11, no. 15, p. 2290, 2022.

[65] M. F. Adnan, N. Ahmed, I. Ishraque, M. S. Al Amin, and M. S. Hasan, "Traffic congestion prediction using deep convolutional neural networks: A color-coding approach," in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–5, IEEE, 2022.

[66] C.-C. Chang, Y.-Z. Li, H.-C. Wu, and M.-H. Tseng, "Melanoma detection using xgb classifier combined with feature extraction and k-means smote techniques," *Diagnostics*, vol. 12, no. 7, p. 1747, 2022.

[67] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020.

[68] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020.

[69] `https://dhavalpatel2101992.wordpress.com/2021/05/21/kaggle-titanic-dataset-cleaning-split-data-into-train-validation-and-test-set`, April 2023.

[70] `https://datascientest.com/matrice-de-confusion`, April 2023.

[71] `https://plat.ai/blog/confusion-matrix-in-machine-learning/`, April 2023.

[72] `https://www.python.org/doc/`, June 2023.

[73] `https://WhatisKaggle?|DataCamp`, May 2023.

[74] `https://www.lebigdata.fr/tensorflow-definition-tout-savoir`, May 2023.

[75] `https://keras.io/about/`, May 2023.

[76] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[77] E. Bisong *et al.*, *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.

[78] `https://www.data-transitionnumerique.com/scikit-learn-python/`, May 2023.

[79] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc.", 2008.

[80] `https://Overleaf,ÃĽditeurLaTeXenligne`, June 2023.

[81] `https://VisualisezetmettezenÅŞuvrevosprojets|Lucidchart`, June 2023.

[82] `https://ezgif.com/video-to-jpg`, June 2023.

[83] A. B. Chan and N. Vasconcelos. `http://visal.cs.cityu.edu.hk/downloads/trafficdb/`, May 2023.