REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté *des Sciences et de la technologie*

Département d'électronique.

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : **Télécommunications**

Spécialité : Systèmes de Télécommunications

Par

- **AMMAR BOUDJELAL AYOUB**
- **BENARROUDJ CHAHINEZ**

Intitulé

Intelligent drone-based crowd counting using transfer learning

Soutenu le : 16 / 06 / 2022

Devant le Jury composé de :

Nom & Prénom	*Grade*	*Qualité*	*Etablissement*
M. ATIA SALIM	*MCA*	*Président*	*Univ-BBA*
M. IDRIS MESSAOUDENE	*MCA*	*Encadreur*	*Univ-BBA*
M. HIMEUR YASSINE	*Dr.*	*Co- Encadreur*	*Univ-Qatar*
M. TALBI MOHAMED LAMINE	*MCA*	*Examinateur*	*Univ-BBA*

Année Universitaire 2021/2022

Acknowledgement

First and foremost, we would like to praise Allah the Almighty, the Most Gracious, and the Most Merciful for His blessing given to us during our studies and in completing this thesis. May Allah's blessing goes to His final Prophet Muhammad (peace be up on him), his family and his companions.

We would like to express our sincere gratitude to all who have supported and encouraged us in various ways through the years. This dissertation would not have been possible without you.

We would like to express our very great appreciations and honor to our supervisor Dr. ***MESSAOUDENE IDRIS*** for his valuable and constructive suggestions and instructions during the planning and the development of this dissertation. Our Co-Supervisor ***Dr. HIMEUR YASSINE*** who deserves all the respect and the appreciations for helping us a lot about the challenges that we faced. He shared his knowledge and experiences in this environment. A special thanks to ***Dr. Atia Salim*** for his willingness to give his time and support so generously has been very much considered. Furthermore, all respect to Dr. **Feneniche Wafa** and **Dr. Belazzoug Massinissa** for their considerations and good intentions, for their advice and encouragement.

Next, our sincere thanks to our **friends** for their support, for the stimulating discussions, for the sleepless nights we were working together , and for all the fun we have had in the last six months.

Last but not the least, we would love to thank our **families**, our **parents** for their continuous support, encouragement and motivation with their helps and prayers this thesis have seen the light.

Abstract

This work presents two situations of crowd (low and high density), and the problems that may happen. To avoid its risks the Crowd management has become a crucial apparatus for monitoring crowds in various places, and an essential task to ensure the safety and smoothness of any events. It uses the new technologies as drones and the artificial intelligence. We present in our thesis two methods of crowd counting using different technics based on neural networks: detection and density estimation.

Ce travail présente deux situations de foule (faible et forte densité), et les problèmes qui peuvent survenir. Pour éviter ses risques le Crowd management est devenu un appareil indispensable pour surveiller les foules en divers lieux, et une tâche essentielle pour assurer la sécurité et la fluidité de tous les événements. Il utilise les nouvelles technologies comme les drones et l'intelligence artificielle. Nous présentons dans notre thèse deux méthodes de comptage de foule utilisant différentes techniques basées sur les réseaux de neurones : détection et estimation de densité.

يقدم هذا العمل حالتين من الحشود (كثافة منخفضة وعالية)، والمشاكل التي قد تحدث .لتجنب مخاطرها، أصبحت إدارة الحشود جهازًا مهمًا لرصد الحشود في أماكن مختلفة، ومهمة أساسية لضمان سلامة وسلاسة أي حدث. تستخدم التقنيات الجديدة مثل الطائرات بدون طيار والذكاء الاصطناعي. نقدم في أطروحتنا طريقتان لعد الحشود باستخدام تقنيات مختلفة تعتمد على الشبكات العصبية: الكشف وتقدير الكثافة.

TABLE OF CONTENTS

List of Figures

List of Tables

General introduction

During religious pilgrimages or huge entertainment events, where a large number of people meet and move in a compact space, crowd accidents are common. Many deaths and injuries occur as a result of pushing and the domino effect of people leaning against each other.

Crowd counting often reduces and prevents these kinds of disasters, where crowd density information is crucial to determine the maximum occupancy of a room or public area to address safety concerns. Counting people can be done in different ways but limited by the time required for the analysis. Therefore counting people in crowded scenes by hand is not a sufficient way even that it can provide accurate values and moving to automated crowd counting is needed to count and maybe predict its behavior ahead of time.

This work presents more details about crowd counting in three chapters.

In the first chapter we will present the meaning of crowd with examples of high and low density, its risks, challenges and how using drones can contribute to ease and pave the way to overcome it. Also, it presents some datasets that are used in this work.

Furthermore, in the second chapter, Artificial Intelligence, Machine Learning, and Deep Learning were all discussed where using them can go way beyond human-level accuracy when dealing with multiple tasks while the main factor is time, we go over the Convolutional Neural Network (CNN) and explain how each parameter affects the network's performance. We presented two approaches for scene analysis: detecting-then-counting which identifies the whole body to identify individuals in the scene, and density map that uses Gaussian kernel to generate a density map. The research classified crowd scenes analysis into three categories: methods based regression, methods based on detection, and methods based on density estimation.

Finally, the last chapter shows the implementation of some crowd counting methods with details. We discussed the steps of each technique with showing the results and comparing theme, ending by a method that can estimate the crowd number accurately in real time.

Chapter I

Crowd counting problems and challenges

I.1 Introduction

Nowadays, gathering in events like sport games, festivals and religious events has become a usual thing to see due to the development of transport means and media. However, the number of gathered people depends on the event's type and this number may increase exponentially in small spaces causing the domino effect of people leaning against each other facing the gamble of group related calamities such as stampedes, which can be deadly.

crowd counting and analysis has become one of the most major research fields that aims to ensure public safety and to prevent efficiently from serious disasters through counting the number of objects in photos or videos under study. Both of them like other fields have many problems in term of data availability and quality.

In this chapter we will make difference between dense and sparse crowd situations and study their risks, challenges. Besides, it will be shown how using drones can contribute to ease and pave the way to overcome these drawbacks. Moreover, we will talk about the different data-sets that are used for the analysis and researches.

I.2 Dense vs sparse crowd

Public places can contain different numbers of humans; few tens of them means low density crowd depending on the surface area. This situation, as presented in figure I.2-1, often contain sparse crowd which allows almost all the body to be visible. Meanwhile hundreds to thousands of humans, as presented in figure I.2-2, is called dense crowd, where distinguishing the whole human body is more likely an impossible thing to do.

Figure I.2-1: sparse crowd [4]

Figure I.2-2: Dense crowd [4]

I.3 Risks of crowded situations

In crowded situations pushing and swaying is an inevitable act causing the domino effect and leading to crushing between people and against fixed structures like shown in figure I.3-1. Sometimes falling brings the high possibility of being trampled underfoot.

FigureI.3-1: Example of crushing between people [4]

Fear from death and getting crushed can lead people to make dangerous behaviors such as climbing onto each other or structures as presented in figure I.3-3, which brings the risks of falling.

Figure I.3-2: Example of people climbing structures [4]

Sometimes, especially in sport events, huge number of crowd might cause terrible and unexpected accidents like structures falling under feet due to the unsupported weight as shown in figure I.3-3. Such situations make escaping the danger an impossible task and usually end up with serious injuries and deaths as depicted in figure I.3-4.

Figure I.3-3: stadium structure falling [30]

Figure I.3-4: Deaths in sports events [30]

I.4 Crowd counting challenges

Crowd counting analysis from images is facing many problems, some are related to angle of capturing and others are related to cameras sensors as presented in figure I.4-1.

Figure I.4-1: Different angles of capturing crowd [4]

- **Perspective variation**

Perspective in photography is quite a confusing topic; it is about how you show a three-dimensional world on a two-dimensional plane. A lot depends on where you place yourself when capturing a scene which may cause human heads to appear at different scales

- **Occlusion**

Where most or a part of heads is occluded or hidden partially. In this case the camera can only take a small portion of the needed information.

- **Diverse scenes**

Taking images in different times of day or different seasons brings diversity which can affect the quality of images and incompatible histogram of pixel values.

- **Low resolution**

Blurry or pixelated image can cause a major problem in crowd counting task because the number of the responsible pixels representing human head in this case is too low which makes distinguishing the needed information from the unneeded one very hard.

I.5 Drones and UAVs

UAV or unmanned aerial vehicle is an aircraft piloted by remote control or onboard computers. However, it should be noted that there is a distinct difference between drones and UAVs. In fact, drones like the one showed in figure I.5-1 exceed the meaning of UAV by the autonomous flight capabilities [1].

Figure I.5-1: Drone (unmanned vehicle area) [1]

Using drones made the crowd counting task easier due to the autonomous flight capabilities and facility of choosing precisely the right angle and height.

Capturing images or videos using drones or UAVs solved the problem of occlusion when working from high point, without forgetting the ability of moving freely.

These last years the **VisDrone (Vision Meets Drones)** international challenge in crowd counting is made by collecting dataset from drones at Lab of Machine Learning and Data Mining by The AISKYEYE team [2]. All the video sequences were taken by drones to make the task realistic and avoid wasting time on the implementation without making good results.

I.6 Datasets for crowd counting

Datasets are the key to good methods when it comes to machine learning and deep learning tasks, for the crowd counting task researchers used different datasets to improve their models. For this purpose, we can find different datasets containing huge numbers of images and their annotations. Researchers and collectors provide manually annotated datasets to train and test models. These datasets differ in the type of annotations, size of images, number of people per image, angle of capture, diverse scenes and many more differences. The crowd counting task is benefiting from these variations by getting trained on more situations.

I.6.1 UCF-CC-50 dataset

Introduced by **Idrees et al**. in Multi-source Multi-scale Counting in Extremely Dense Crowd Images paper [3], the UCF-CC-50 shown in figure I.6-1 is a dataset for crowd counting and consists of images of extremely dense crowds as follows:

- It has 50 images with 63,974 head center annotations in total
- The head counts range between 94 and 4,543 per image.

The small dataset size and large variance make it a very challenging counting dataset.

Figure I.6-1: UCF-CC-50 dataset [3]

I.6.2 UCF-QNRF dataset

The **UCF-QNRF** dataset of figure I.6-2 is a crowd counting dataset and it contains large diversity both in scenes, as well as in background types [3], with the following individualities:

- consists of 1535 images
- High-resolution images from Flickr, Web Search and Hajj footage
- The number of people varies from 50 to 12,000 across images.

Figure I.6-2: UCF-QNRF [3]

I.6.3 ShanghaiTech dataset

ShanghaiTech dataset, presented in figure I.6-3, is a large-scale crowd counting dataset introduced by Zhang et al. in Single-Image Crowd Counting via Multi-Column Convolutional Neural Network [4], with the following proprieties:

- Consists of 1198 annotated crowd images.
- The dataset is divided into two parts, Part A and Part B

- Part-A containing 482 images and Part-B containing 716 images.
- Part-A is split into train and test subsets consisting of 300 and 182 images, respectively.
- Part-B is split into train and test subsets consisting of 400 and 316 images.
- The dataset consists of 330,165 annotated people.
- Images from Part-A were collected from the Internet, while images from Part-B were collected on the busy streets of Shanghai.

Figure I.6-3 ShanghaiTech dataset [4]

I.6.4 Visdrone Competition Dataset

Visdrone Competition Dataset, presented in figure I.6-4, was held on the 16th European Conference on Computer Vision (ECCV 2020) [5]. It contains small object inference, background clutter and wide viewpoints. It is constituted of 3360 images among which 2460 are for training and 600 for testing.

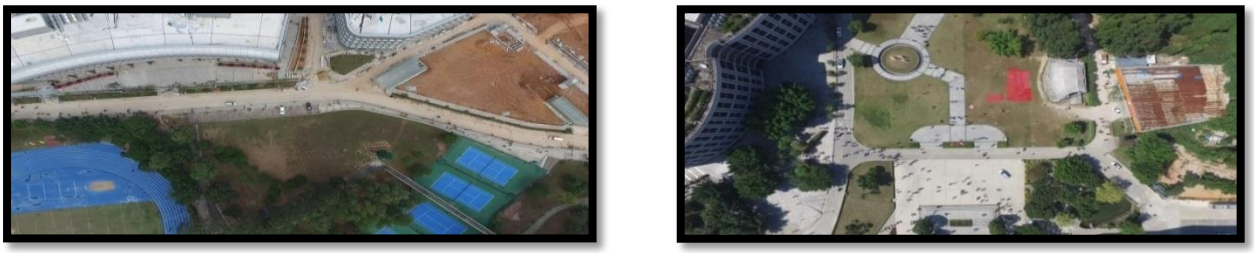


Figure I.6-4 VisDrone 2020 dataset [5]

I.7 Annotations

Machine learning and deep learning models need answers to get trained or briefly adding metadata to a dataset. These answers can be integers, strings, floats and arrays as they can be labels or pixel positions in an image depending on the used task to show presence, location, count and other information resumed in Table I.7-1

Table I.7-1types of metadata used in deep learning.

	classification	Object detection	Semantic segmentation	Instance segmentation	Crowd counting
presence	✓	✓	✓	✓	✓
location	×	✓	✓	✓	✓
count	×	✓	×	✓	•
size	×	×	•	✓	×
shape	×	×	•	✓	×

✓ : presence × : absence • : optional

In crowed counting, there are three methods of annotations that are given in the following.

I.7.1 Rectangles

Drawing rectangle that confine the boundaries of the human body or head, also naming (labeling) the rectangles as a person is a technique used in object detection. An illustration of this method is given in figure I.7-1.

Figure I.7-1 Rectangle annotations [4]

I.7.2 Polygons

Polygons, as shown in figure I.7-2, delineate boundaries between similar objects and label them under the same identification. They are used in object detection method.

Figure I.7-2 Polygons annotations [4]

I.7.3 Point annotations

One of the latest types of annotations is making a point in the center of each head in the crowd. An example of this type is showed in figure I.7-3, where a red point is assigned to every human head. This type of annotations is used in density map method.

Figure I.7-3 Point annotations [4]

I.8 Conclusion

This chapter introduces the crowd counting task by showing its necessity nowadays to avoid some disasters through presenting two types of crowd (high and low density) and giving some of its challenges.

We showed that using stationary cameras and drones has made the crowd counting task easier and more precise than before.

Finally, we presented a various datasets that contain images of different size, with a number of people in each picture and other parameters.

Chapter II

Artificial intelligence and crowd counting methods

II.1 Introduction

Nowadays, humans are using electronic machines more than previously to such a level that almost everything is automated and includes electronics, as in hospitals, cars, homes, stadiums…. These electronics has become very smart and useful due to their development and algorithm that made it learnable and behaves like human minds, this act is called **Artificial Intelligence (AI)**.

Researchers has made a big step in the crowd counting field by applying several methods using **Machine Learning** (**ML**) and **Deep Learning (DL)** through **Computer Vision (CV)** to overcome the crowd counting problems.

This race made the crowd counting systems differs into non-supervised and supervised algorithms.

Crowd counting methods can be divided or classified into three main methods which are based on: **detection**, **regression** and **density map estimation**.

Due to the problems facing these methods, their classification is compulsory. In fact, the difference between dense and sparse crowd situations causes many problems like defining two close pedestrian shapes as a single object. Such difficulties cause that some of existing methods are not applicable in detecting human instances in crowd scenes.

In this chapter, the concepts of **AI**, **ML, DL** and **CV** are firstly defined and characterized. Then, some of the famous CNNs used in each method are given and discussed and their contribution in making better results in the crowd counting analysis is studied without forgetting the loss functions and metrics used to analyze each method.

II.2 Artificial intelligence

AI is the ability of a digital computer or "computer-controlled robot" to perform tasks commonly associated with intelligent beings. It is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience [6].

Since the development of the digital computer in the 1940s, it has been demonstrated that computers can be programmed to carry out very complex tasks as, for example, discovering proofs for mathematical theorems or playing chess with great proficiency [6].

AI contains two main branches which are Machine learning and deep learning as shown in figure II.2-1.

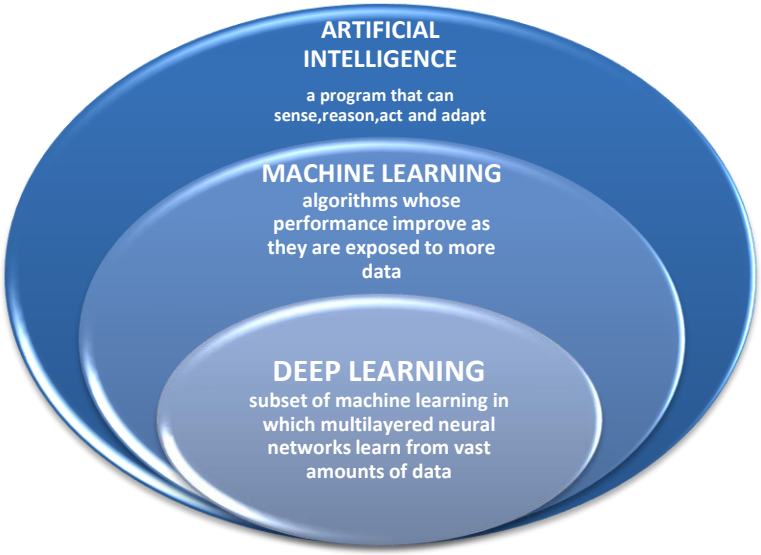


Figure II.2-1 Artificial Intelligence and its fields

II.3 Machine learning and deep learning

II.3.1 Machine learning

Machine learning is a term included when a computer learn from data, where algorithms are used to perform a specific task without giving any instructions about the path to the results.

The learning process can either be supervised or unsupervised and sometimes might be reinforced. Once programmed, a computer can take in new data indefinitely, sorting and acting on it without the need for further human intervention.

Over time, the computer may be able to recognize things even if the supervision is stopped. This 'self-reliance' is so fundamental to machine learning that the field breaks down into subsets based on how much ongoing human help is involved.

Machine learning is generally divided into three main sub-fields as shown in figure II.3-1.

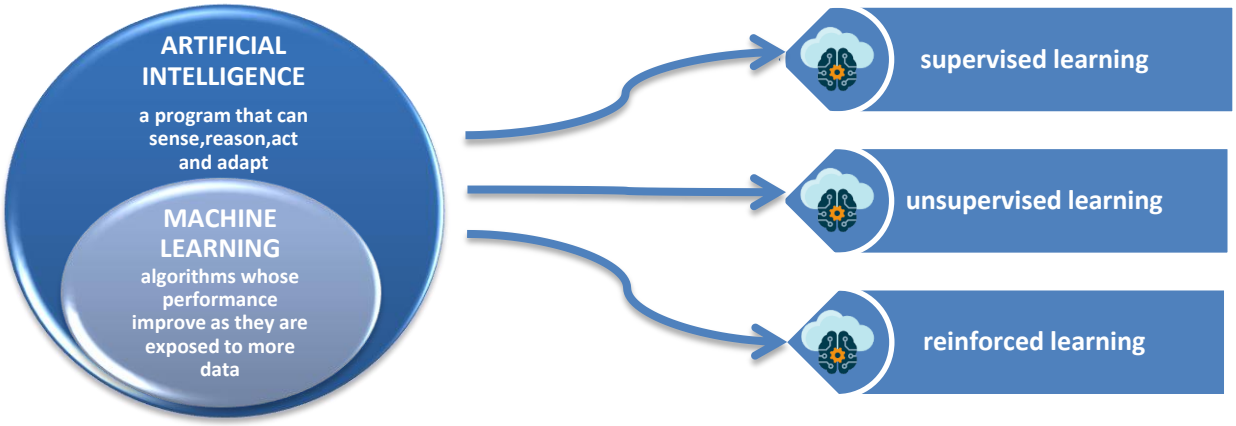


Figure II.3-1 Machine learning subfields

II.3.1.1 Supervised learning

In supervised learning, the human gives labeled training data to the computer and explicit the design to reach the results or respond on data.

This level of supervised learning may lead the model to predict on new datasets of the same field very accurately. However, keep monitoring the computer can cause the model to get out of bounds and lose its efficiency.

In semi-supervised learning, the computer is fed a mixture of correctly labeled data and unlabeled data, and searches for patterns on its own. The labeled data serves as 'guidance' from the programmer, but they do not issue ongoing corrections.

II.3.1.2 Unsupervised learning

Using unlabeled data made the computer free by choosing suitable patterns to reach solutions that improved the learning to such unlimited level.

Unsupervised learning can be used in **clustering,** where the computer can organize data into classes and makes the decision depending on the user needs.

II.3.1.3 Reinforcement learning

The keywords of this method are **agent, actions, rewards** and **observation.** The agent learns to perform a task through trial and error in a dynamic environment by making a series of decisions to maximize the reward for successfully performing a task.

II.3.2 Deep learning

DL is a part of Machine learning that helps to teach computers or machines to replace humans in various fields like biology, engineering, solving math problems, daily life and many more, based on what the brain can do naturally. For example: learn to drive cars, recognizing faces, feelings, making a conversation…. DL is getting lots of attention lately after found reaching better than the expected results.

Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance.

Models are trained by using a large set of labeled data and neural network architectures that contain many layers [7].

II.4 Deep learning vs traditional programing

Giving rules or logic to get results in a program is required in traditional programing, where a programmer needs to give the rules manually step by step, feed them to the computer and then analyze the results, as illustrated in the block diagram in figures II.4-1.

In traditional programing, rules are the guide to the results that can't be developed by time or the computer interferences, finally the output can only be one certain result.

On the other hand, in DL the matter is not the same, since giving rules is clearly unneeded. However, giving inputs and answers is a must as shown in figure II.4-2. Here, the computer will do necessary studies and analysis using the provided data and at the end comes up with a model or program that is able to solve the main and the related problems every time.

Figure II.4-1 Traditional programming

Figure II.4-2 Deep learning

II.5 Computer vision

CV is a branch of AI that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and takes actions or makes recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand [8]. Some other formal textbook definitions are given as follows:

- "The construction of explicit, meaningful descriptions of physical objects from images" [9].
- "Computing properties of the 3D world from one or more digital images" [10].
- "To make useful decisions about real physical objects and scenes based on sensed images" [11].

II.5.1 The most common fields of computer vision

II.5.1.1 Image Classification

Convolutional Networks (**ConvNets**) are currently the most efficient deep models for classifying images data [12].

Identifying and extracting low level features from images and then learning to recognize them and then combine them to learn more complicated patterns. Using convolution and different types of activations, has made image classification problems easier.

II.5.1.2 Object detection

Classifying images is a huge part of object detection, identifying the class of the object then labeling where it can be found is called object detection.

Object detection can be used in many places for example detecting damages on an assembly line or detecting face masks.

II.5.1.3 Object tracking

Detected objects in a video can be followed or tracked by applying object detection on every image sequence in a video.

Object tracking helped automated engineering by adding the factor of time (speed) as in the cases of autonomous vehicles or real time surgery, where classifying and detection process is not enough and the task need to be continuous.

II.6 Artificial neural networks (ANNs)

Simply called neural networks (NNs), are a combination of layers made by artificial neurons (perceptron) inspired from biological neurons.

II.6.1 Artificial neuron

The truth behind these neurons doesn't exceed a mathematical combination of equations, where one neuron accepts n inputs (x1, x2, x3…xn) multiplied by n weights (w1, w2 ,w3…wn) to generate an output based on a well-chosen transfer function (activation) **σ** [13], see Figure II.6-1. This gives the following equation:

$$y_k = \sigma\left(\sum_{i=0}^{n} w_{ki} x_i\right) \tag{1}$$

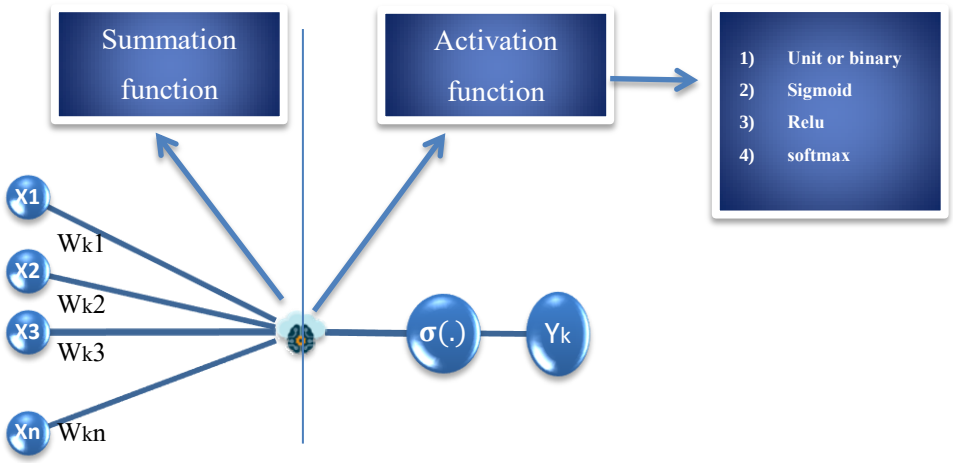


Figure II.6-1 Basic structure of a neuron

Training a neuron is based on giving it data or initial values as a starting point then beginning the progress of the summation with random initial weights (not always). After that, the error is computed and finally the new weights are updated to decrease the error value.

The error function can be given as follows:

$$\boldsymbol{error = n(d - y)x} \tag{2}$$

Where **n, d, y** and **x** are, respectively, the learning rate value, the actual output, the perceptron output and finally the input value.

II.6.2 Activation functions

The responsible part that determines the output of a neural network is the activation function, presented in Figure II.6-2, by deciding whether the neuron should be activated or

not, and it determines the output of a model, its accuracy and finally the computational efficiency.

There are several activation functions that have been used in DL. We can't mention all of them here but we will provide the most used once.

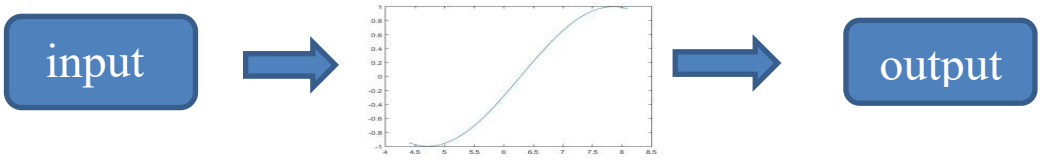


Figure II.6-2 activation function

II.6.2.1 Binary step function

Binary step function is based on a threshold value to decide whether if the neuron should be activated or not. It can be summarized by the equation (3) whose plot is presented in Figure II.6-3.

$$f(x) = \begin{cases} 0 & for\ X < 0 \\ 1 & for\ X \geq 0 \end{cases} \tag{3}$$

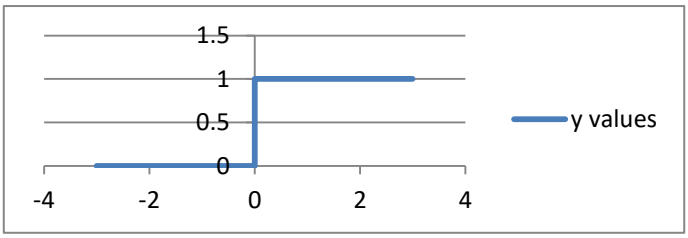


Figure II.6-3 Binary step function

II.6.2.2 Sigmoid function

This function can take any input and give the output in range between 0 and 1, it can be summarized in the equation (4) and its plot in Figure II.6-4.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

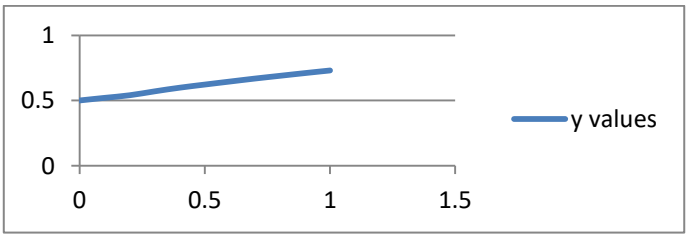


Figure II.6-4 sigmoid activation

II.6.2.3 Relu function

Relu function stands for rectified linear unit, and it represented by the following equation whose plot is shown in Figure II.6-5.

$$f(x) = max(0, X) \tag{5}$$

It gives an impression of a linear function, which has a derivative function, and it allows for backpropagation while simultaneously making it computationally efficient [14]. However, this function doesn't activate all the neurons at the same time.

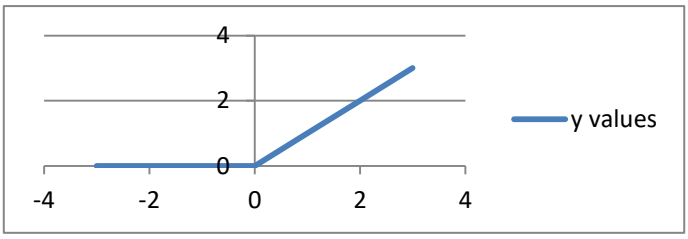


Figure II.6-5 Relu activation

II.7 Neural networks and deep neural networks

Deep Neural Networks (DNN) are an important method for machine learning, which has been widely used in many fields.

Compared with the shallow neural networks (NN), DNN has better feature expression and the ability to fit the complex mapping [15] because it contains more hidden layers and complex shapes as Figure II.7-1 shows.

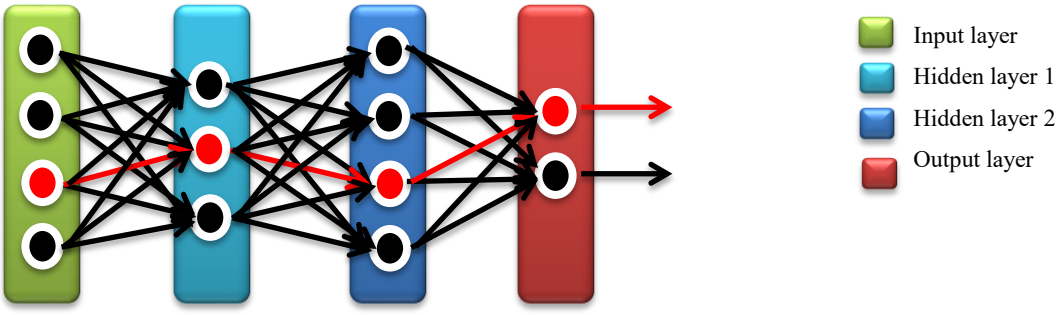


Figure II.7-1 Deep neural network

Deep neural networks are based on the same basics of a perceptron because they are formed by perceptrons and this last gives them the ability to learn better with higher number of trainable parameters.

II.8 Convolutional neural networks (CNNs)

The main difference between the fully connected neural networks (FCNN) and CNNs is the number of connections between neurons. Both network's layers can have the same number of neurons but CNNs have les connections in between [16].

Using CNNs is not confined only on compensation in the unneeded connections between layers, but also in the time of the processing and resources. As example, we can take an image of shape (64*64*3) in both networks and the difference appears immediately when calculating the number of weights. In a DNN every layer will have exactly the multiplication result of (64*64*3=12288) weight while CNNs layer can have fewer weights and sometimes the ratio between both can be slightly less than a half.

II.8.1 CNN elements

A basic CNN can contain three types of elements as shown in Figure II.8-1

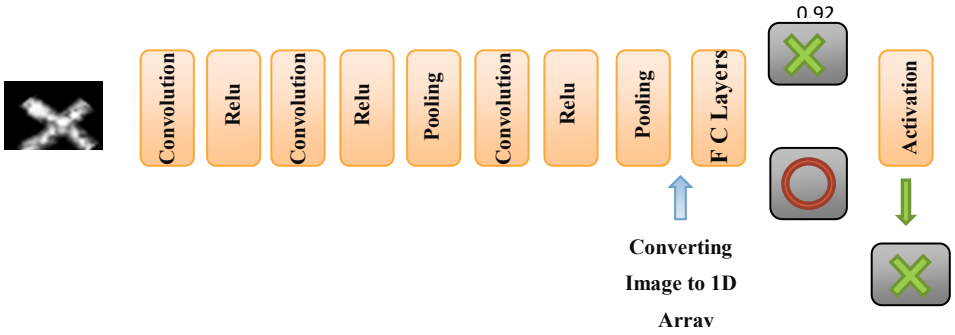


Figure II.8-1 example of CNN architecture

II.8.1.1 Convolutional layer

Using convolutional layer in a NN is a good strategy to extract weights and look for local regions in an image by manually picking a matrix of size n*n which can be set to detect edges, sharpen edges, add Gaussian blur and many more tasks.

II.8.1.2 Stride

CNN has more options which provide a lot of opportunities to even decrease the parameters more and more, and at the same time reduces some of the side effects [16]. Among these parameters, there is the stride which informs how much pixels to overlap while moving along the image or how many slide to move the filter.

II.8.1.3 Padding

Padding is a solution to the drawbacks of reducing the size of the image when applying convolution. It helps to keep boundaries of the image.

Zero padding (adding zeros) or same padding (giving the new pixels the same value as the boundaries pixel values) are two methods to use in deep learning.

II.9 Methods of crowd counting

Crowd counting is divided into three main methods, as presented in Figure II.9-1,

Including: **detection**, **regression** and **density map estimation.**

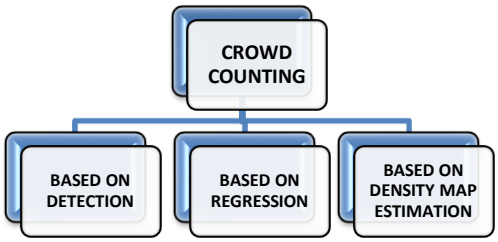


Figure II.9-1 crowd counting methods

II.9.1 Detection based methods

This approach uses sliding window to detect and classify objects in the image and then count the total number of detected objects [17]. It can be found in various applications (autonomous cars, smart houses alarms, face recognition applications) as it can be applied to detect humans, animals, tools and any other object that is trained to detect.

Some of the best CNNs used in this method for crowd counting are R-CNN (Region-based Convolutional Neural Networks) [18], Fast R-CNN [19] and Mask R-CNN [19].

II.9.1.1 Region-based Convolutional Neural Networks (R-CNN)

This method uses **ROI**s (region of interest) then computes CNN features after that a simple task is used to classify objects; this later is called image classification.

The basic architecture of the R-CNN is presented in Figure II.9-2.

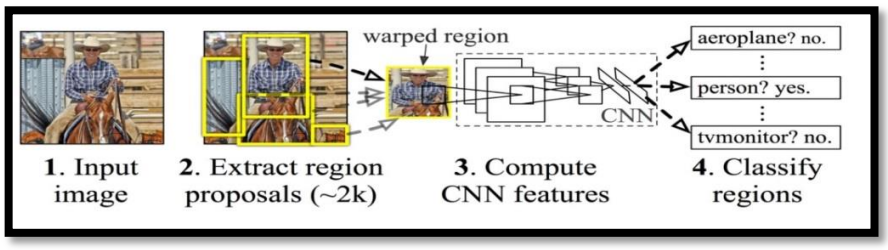


Figure II.9-2 Architecture of R-CNN [18]

II.9.1.2 Fast RCNN

This network is a better version of the R-CNN since it forwards the entire image through a CN (convolutional network). The **ROI** will be used on smaller size feature map which makes the process faster and less demanding to resources. The architecture is shows in Figure II.9-3.

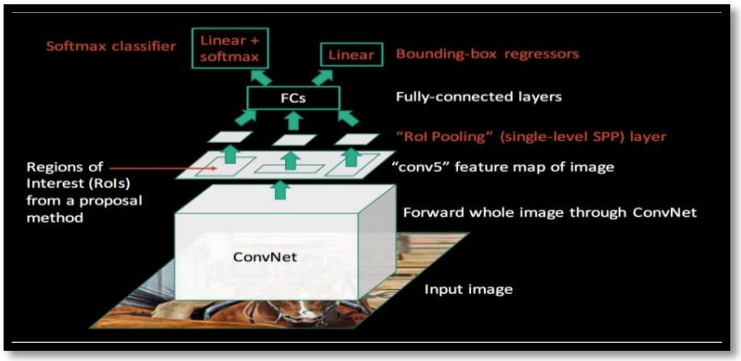


Figure II.9-3 Architecture of Fast RCNN [19]

A newer version of this network, called Mask-RCNN makes the region of interest more accurate by annotating the objects (heads) with polygons so the area is defined only on the highlighted pixels.

II.9.2 Regression based methods

Regression approaches are suggested to solve object detection problems. These methods extract a low level features from the image like histogram or local features like SIFT (**scale-invariant feature transform**), HOG (**Histograms of Oriented Gradients**), LBP (**Local Binary Patterns**) and execute a regression modelling to map the features to their corresponding counting results.

This method is based on Foreground segmentation and Perspective normalization [20], where estimation is seeking to minimize the errors through the Bayesian Poisson Regression. This latter is formed by equation (6).which is an approach to linear regression in which the statistical analysis is undertaken within the context of Bayesian inference. The response variable of Bayesian Poisson regression (also called the dependent variable) is given by:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \varepsilon \quad (6)$$

Where β's are the weights (known as the model parameters), $x's$ are the values of the predictor variables, and ε is an error term representing random sampling noise or the effect of variables not included. A plotting example of equation (6) is given in Figure II.9-4.

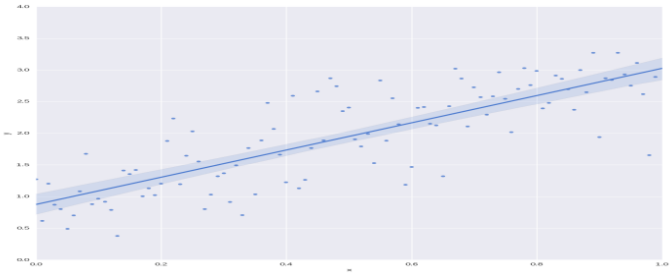


Figure II.9-4 Bayesian Poisson regression example

II.9.3 Density map estimation based methods

Density map estimation methods are state of the art of nowadays that took crowd analysis to another level, where the goal is mapping low level features. It solves dense crowd problems by generating the ground truth that is generated by using point annotations to extract features with infinite accuracy. This method uses the 2D Gaussian distribution for all annotated pixels which is given by the following equation.

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (7)$$

Where G_σ is 2D Gaussian distribution for kernel centered on the point m of coordinates (x, y) and σ is the standard deviation. Sum of the points inside a Gaussian kernel is equal to 1. σ is used to compute the weighted average of the neighboring points (pixels) in an image while changing it will change the size of points as presented in Figure II.9-5.

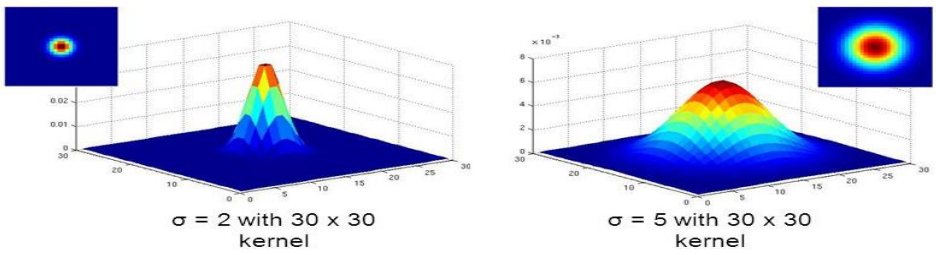


Figure II.9-5 Gaussian kernel 2D with different sigma values

Later, the integration of a CNN method in density based crowd counting has become widely used [17]. CNN method can be split into three categories that are based on: □ Basic CNN, Multi-column and Single column.

II.9.3.1 Multi column convolutional neural network

Multi column convolutional neural network (MCNN), usually made by several CNN columns, is used to capture and extract features of different size. For the example presented in

Figure II.9-6, three CNN Columns (small, medium, large) are used. The goal of this MCNN is achieved by using different convolution kernels in the back-end of each CNN column. The density map at the output is obtained by merging the outputs of all three CNN columns in one layer and then forwarding them through a convolution of size [1*1].

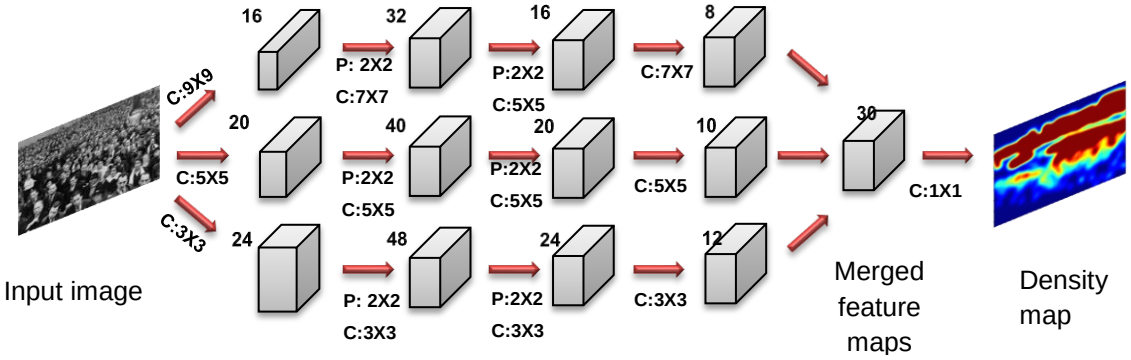


Figure II.9-6 Architecture of MCNN

II.9.3.2 Congested Scene Recognition Network

Congested Scene Recognition Network (CSRNet) is based on transfer of learning from a pretrained model. It uses the first ten layers from VGG-16 [21] model as a front-end and dilation convolution as a back-end. The CSRNet structure, created by Y. Li, X. Zhang, and D. Chen [22], is presented in Figure II.9-7.

There are many configurations of the **CSRNet** that differ only in the back-end structure.

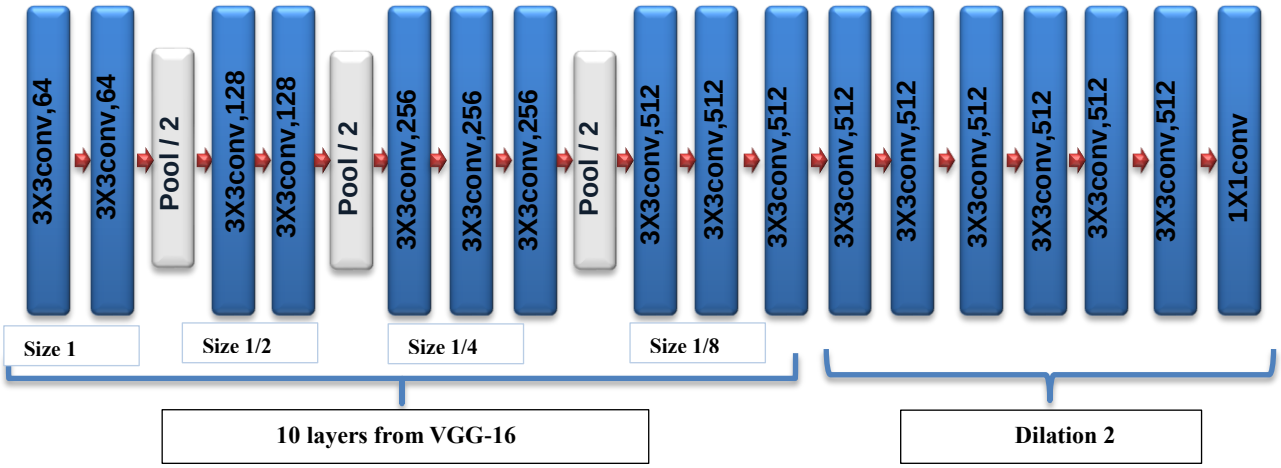


Figure II.9-7 Architecture of CSRNet [22]

II.9.4 W network

W network (Wnet), as presented in Figure II.9-8, is created by two branches in order to keep the same size of image; the alphabet W stands for its shape. The first branch is made

for extracting features and downscaling image size. The other branch is composed by the two following parts:

1. Upscaling purpose and extracting feature maps mixed with features from branch 1.
2. Similar to the first part but ending with an additional convolution of size 1*1.

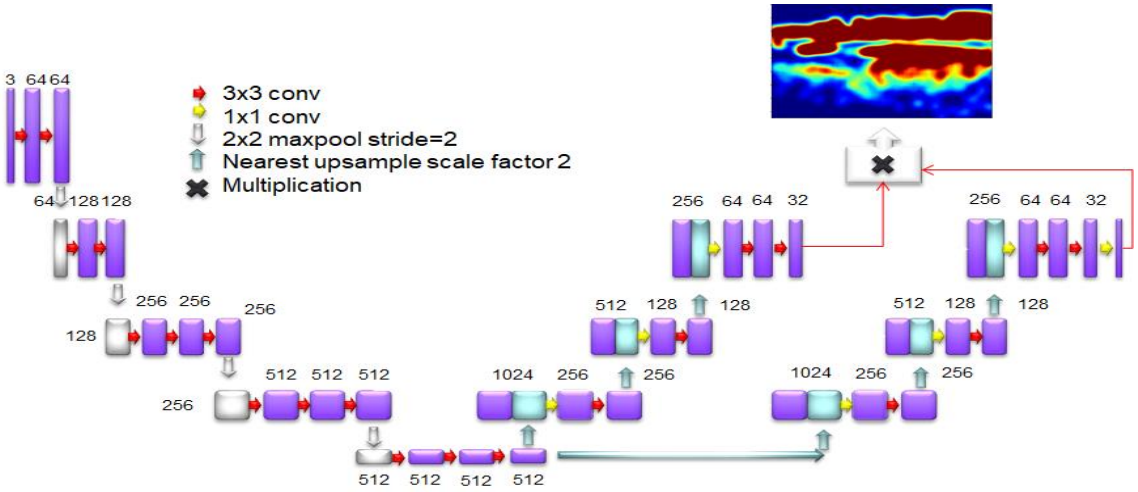


Figure II.9-8 Architecture of Wnet

II.10 Loss functions and metrics

Deep learning models need to be evaluated by loss functions and metrics to learn and validate results. There are a lot of loss functions and metrics in deep learning but we are going to describe exclusively those used in crowd counting.

Object detection method requires two main tasks, the image classification and the object positioning that are the reason of using two different loss functions; the image classification may use the binary-cross entropy, the categorical cross entropy or sometimes the accuracy computation. The object positioning necessitates using mean squared error (MSE) or mean absolute error (MAE) for the bounding box offset prediction.

Density map method requires MAE, MSE and related functions to be evaluated.

II.10.1 Categorical cross entropy

Categorical cross entropy is a loss function used in multi-class classification tasks. It is used to compute the distortion of the output class, by the following equation:

$$logloss = -\frac{1}{N}\sum_{I=1}^{N}\sum_{J=1}^{M} Y_{IJ} log(P_{IJ}) \quad (8)$$

Where N is the length of the fully connected layer output array and M is the number of classes, P_{IJ} is the class probabilities of the class and Y_{IJ} is the class values.

II.10.2 Binary cross entropy

Binary cross entropy is used on one class to compare it with the predicted probability. Thus, after replacing M with 2 in equation (8), it gives the following simplified equation:

$$logloss = -\frac{1}{N}\sum_{I=1}^{N}[Y_I \log(P_I) + (1 - Y_I)\log(1 - P_I)] \quad (9)$$

II.10.3 Mean absolute error

MAE is generally formed by the sum of the absolute difference between correct and wrong values divided by the total number of values. Mathematically, it is represented in equation (10).

$$MAE = \frac{1}{N}\sum_{0}^{N}|X_N - Y_N| \quad (10)$$

With N the total number of values, X_N is the correct value and $Y_N$is the wrong one.

II.10.4 Mean squared error

In general, it is formed by square root of the sum of the squared difference between correct and wrong values divided by the total number of values and it can be represented in the following equation:

$$MSE = \frac{1}{N}\sum_{0}^{N}(X_N - Y_N)^2 \quad (11)$$

Where N is the total number of values, X_N are correct value and Y_N are predicted.

II.10.5 Accuracy

Accuracy is the sum of true positive and negative predictions divided by the sum of all the predictions. For more clarification, the Table II.10-1 and equation (12) are introduced.

Table II.10-1 accuracy table

		prediction	
		0	1
actual	0	True negative TN	False Positive FP
	1	False Negative FN	True Positive TP

$$accuracy = \frac{TP + TN}{TP + FP + TN + TP} \quad (12)$$

II.11 Conclusion

In this chapter we introduced the artificial intelligence, machine learning, and deep learning. We discussed the Convolutional Neural Network (CNN) and explain the effect of activation functions, loss functions and the metrics criteria on the performance of the network.

We also presented diverse crowd counting tasks based on CNNs (CNN-CC) techniques that are able to count crowd and perform other tasks such as classification, segmentation, uncertainty estimation and analysis of crowd behavior. These multitask CNN-CC are designed through computer vision and deep learning in order to show a small portion of what researchers has done in the field. Meanwhile, we tried to give a small part of what is needed to go through the crowd counting analysis using deep learning.

However, making good results in this field requires the knowledge of many essential things starting by those presented in this chapter and lots more. We propose a variety of CNNs architectures that specifically incorporate various aspects such as global/local context information that can be applied to obtain a crowd count in real time due to their simple network architecture.

Chapter III

Implementation of crowd counting methods

III.1 Introduction

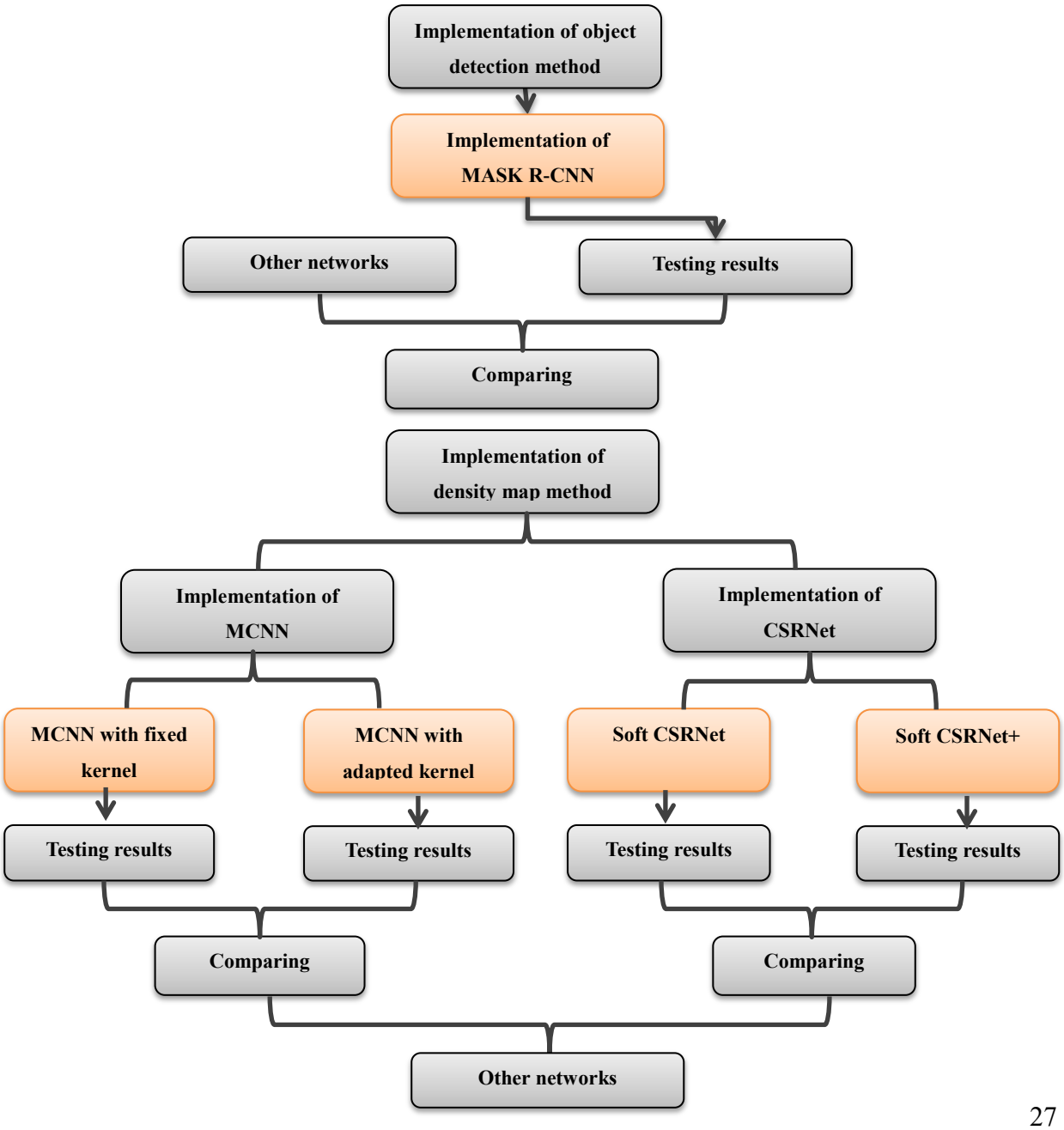
In this chapter, some crowd counting methods were implemented, tested and compared to several recent existing ones. These methods can be broadly categorized into object detection and density map estimation, to address the field related challenges like real time prediction, which is crucial to prevent crowded situations risks, scale variations, occlusion..

We also tried to solve some of the task problems through studying and searching deeper in pros and cons of each method given in the state of the art.

Finally, the implementations of some models with good performances are presented.

III.2 Global organizational chart

The global organizational chart of all the implementations selected in the state of art.



III.3 Implementation of object detection method

The Mask R-CNN that is latest version of R-CNN networks, has been chosen in the following according to its performance superiority as explained in the state of art (chapter two) [19].

III.3.1 Implementation of Mask R-CNN

After downloading the right dataset from the open source "**COCO dataset"** [23] the path of the implementation is explained and showed in Figure III.3-1.

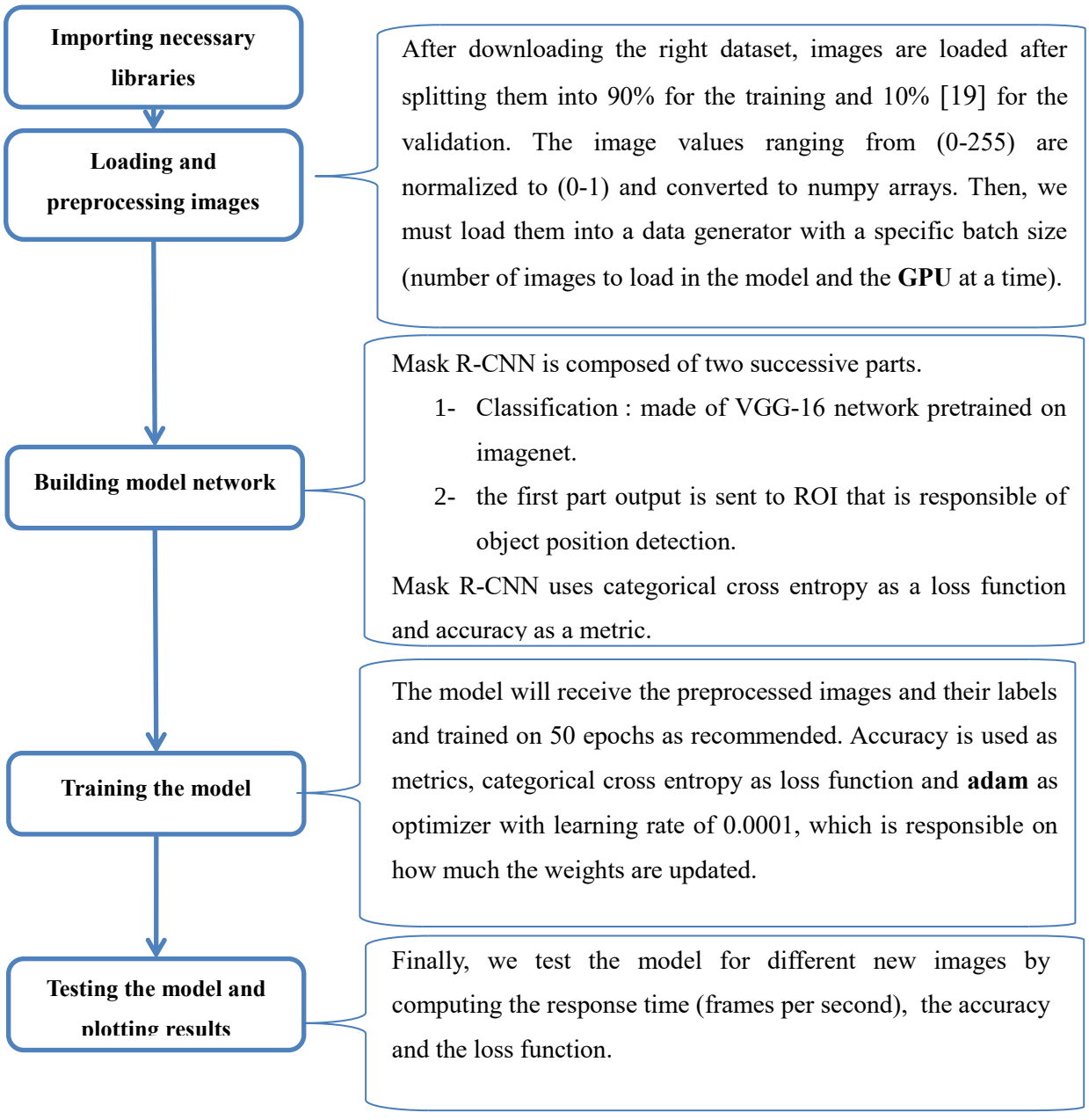


Figure III.3-1 implementation of Mask R-CNN

III.3.2 Building the model

From the implementation path details aforementioned, the building blocks of the model can be structured as shown in Figure III.3-2.

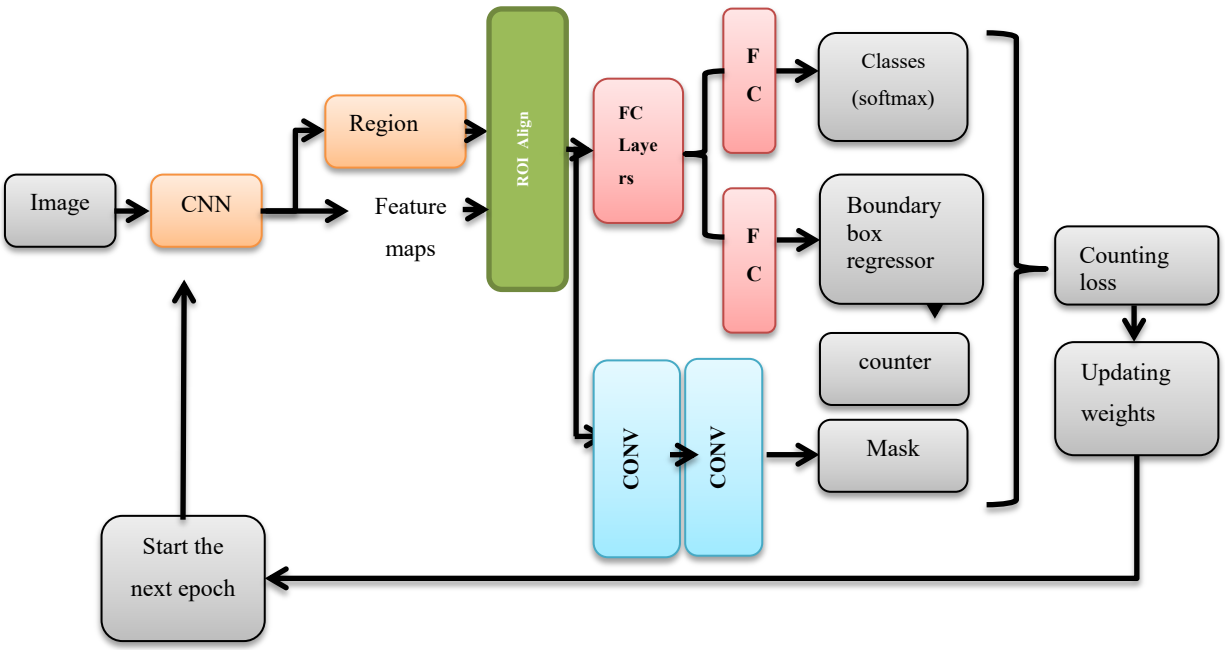


Figure III.3-2 building of Mask R-CNN model

III.3.3 Results and discussion

After training the Mask R-CNN model for 50 epochs the obtained loss and accuracy values are shown in Figure III.3-3. As we can see, the training and validation losses are both decreasing to approximately zero, which means that the model is learning. However, the validation loss decreases more quickly for the first epochs which can be explained by the effect of chronological order.

Besides, we notice that the training and validation accuracy are both converging the constant value 0.8, even though the training accuracy converges slower for the first epochs.

The detection accuracy, the mask accuracy and the running speed were derived, respectively, from the realized tests on different images.

The simulation results are presented in Table III.3-1.

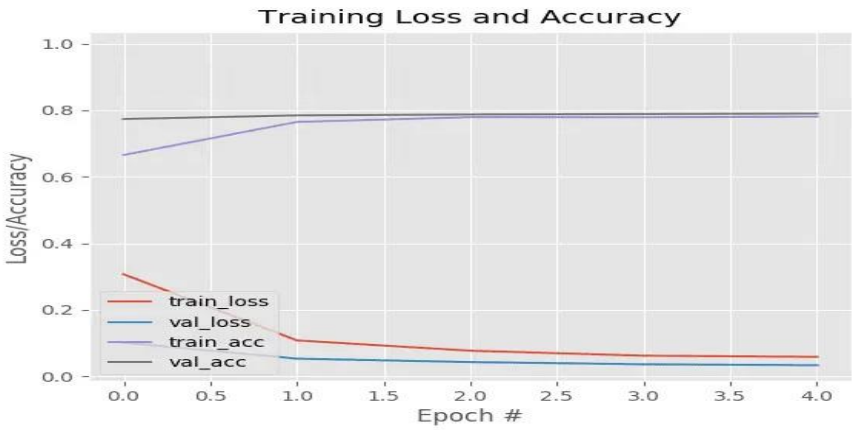


Figure III.3-3: training loss and accuracy of Mask RCNN

Table III.3-1 Mask RCNN performances

	Detection accuracy (%)	Mask accuracy (%)	Running speed (fps)
MASK RCNN	79.8	81.25	4.26

A comparative analysis of the implemented Mask R-CNN network and other existing networks is presented in Table III.3-2. Here the values of R-CNN and Fast R-CNN were taken from the results of a direct application downloaded from the open source GitHub applied on the same images.

Table III.3-2 comparison between R-CNN, Fast R-CNN and Mask R-CNN

	R_CNN	Fast R-CNN	Mask R-CNN
Region proposals method	Selective search	Selective search	Region proposal network
Prediction timing	40-50 seconds	≈2 seconds	≈0,23 seconds
Computation	High computation time	Low computation time	Low computation time
Full Detection accuracy (%)	60	92.1	95.3
Running speed (fps)	0.02	0.5	4.34

An example of Mask R-CNN prediction is showed in Figure III.3-4.

Figure III.3-4: Example of Mask RCNN prediction

III.3.4 Testing Mask R-CNN on sparse crowd scenes

The model is tested on different situations especially on moving people in public spaces for a better evaluation. As a result we found that Mask R-CNN can't be enough for the crowd counting purpose. As presented in Figure III.3-5 and Figure III.3-6, the presence of any occlusion may influence the detection which means this method is not applicable in crowded situations.

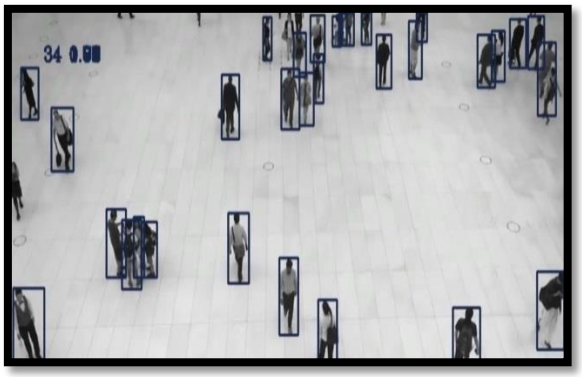


Figure III.3-5: occlusion 1[29]

Figure III.3-6: occlusion 2[29]

III.3.5 Implementation of the Gaussian density map estimation method

The implementation of Gaussian density map method follows the steps showed in Figure III.3-7.

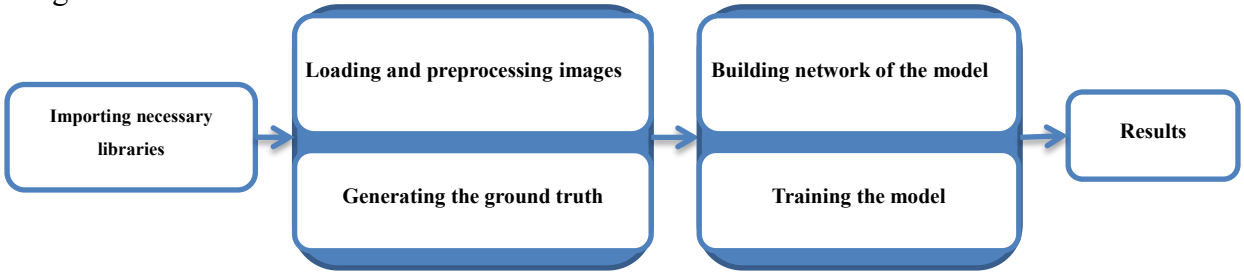


Figure III.3-7 steps of implementing Gaussian density map method

III.3.5.1 Implementation of MCNN

Implementing the MCNN necessitates downloading the datasets (UCF-CC-50, UCF-QNRF and ShanghaiTech). This variety of datasets serves for better evaluation.

III.3.5.1.1 MCNN with fixed Gaussian kernel size

The implementation steps of MCNN with fixed Gaussian kernel size are:

1- Importing necessary libraries

- Importing libraries for processing images, arrays, matrixes, (math, numpy, opencv, pandas, matplotlib…)
- Importing the end-to-end open source platform **Tensorflow** and **Keras (an API designed for reducing cognitive load)** to build and train the network

2- Loading and preprocessing images

Preprocessing images is one of the main steps of implementing MCNN, as presented in Figure III.3-8. Images need to be with the same size and data type as those of the network input. For this preprocess we need to normalize (0-1) images and convert them to float32 data type. MCNN inputs must be grayscale.

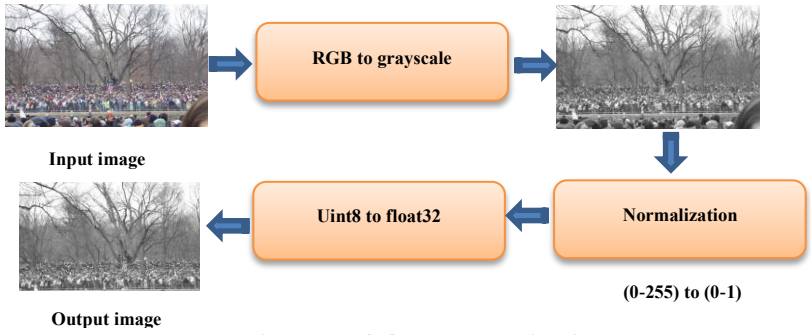


Figure III.3-8 Preprocessing images

3- Generating ground truth

The first step of generating the ground truth (GT) is to create a black image (pixel values 0) using matrix, and then a generator is built to generate the Gaussian density map using two inputs (annotations and empty images) basing on the equation (7). To explain more the organizational chart is designed and presented in Figure III.3-9. Results of this generator are illustrated in Figure III.3-10.

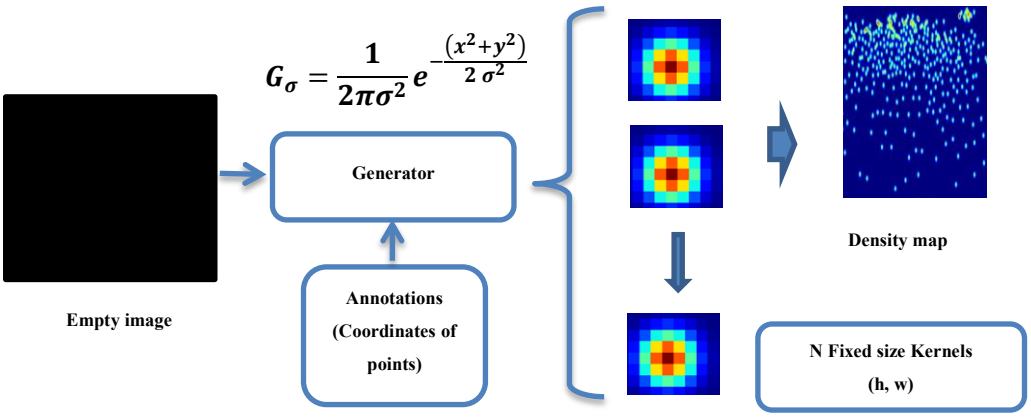


Figure III.3-9 GT generator

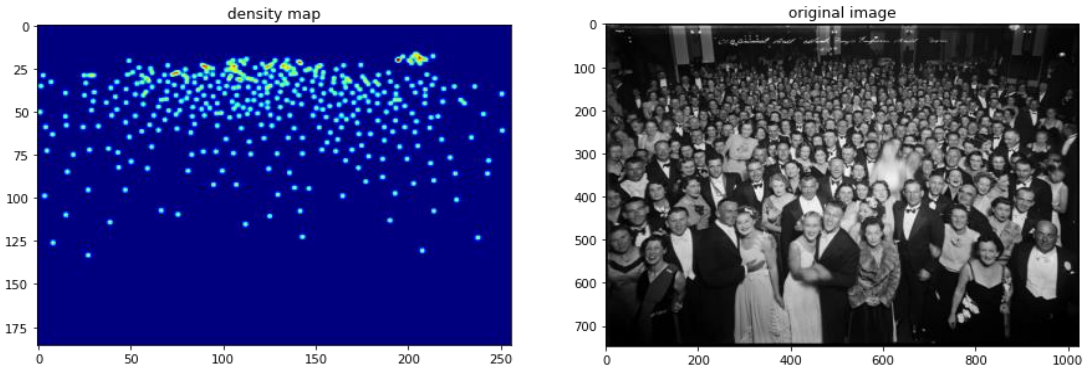


Figure III.3-10 Fixed kernel Gaussian density map

4- Building MCNN

MCNN is based on 3 columns having each a specified number of kernels. The first column starts by 16 convolutional kernels of size 9*9 to extract big features, forwarded by pooling layer of size 2*2 (Maxpooling). The next layers are produced with the same way with slight changes in the size of the convolutional layers as presented in figure III.3-11. The other 2 columns are similar to the first one but with smaller kernels. Finally, the output is applied to Adam optimizer function with learning rate, followed by Loss function with metrics.

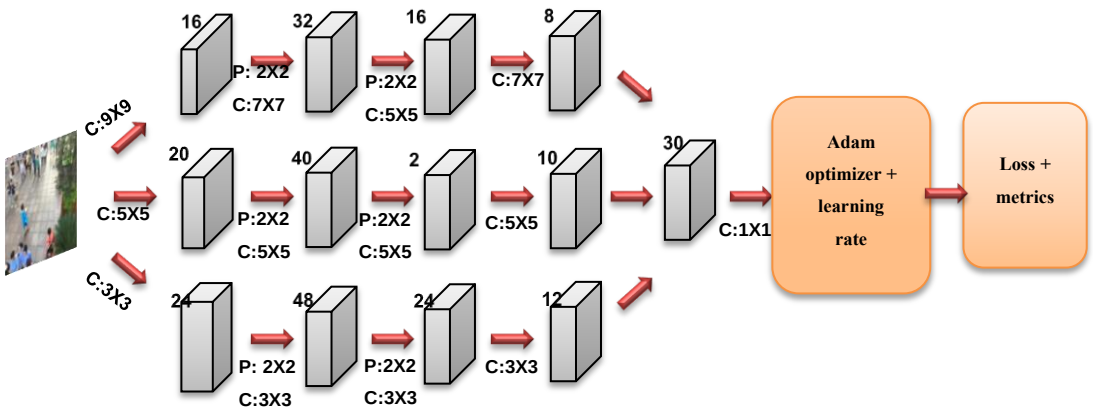


Figure III.3-11 Building the MCNN

5- Training the model

The model is trained for 140 epochs; every epoch took around (227s). The steps of training are clarified in Figure III.3-12.

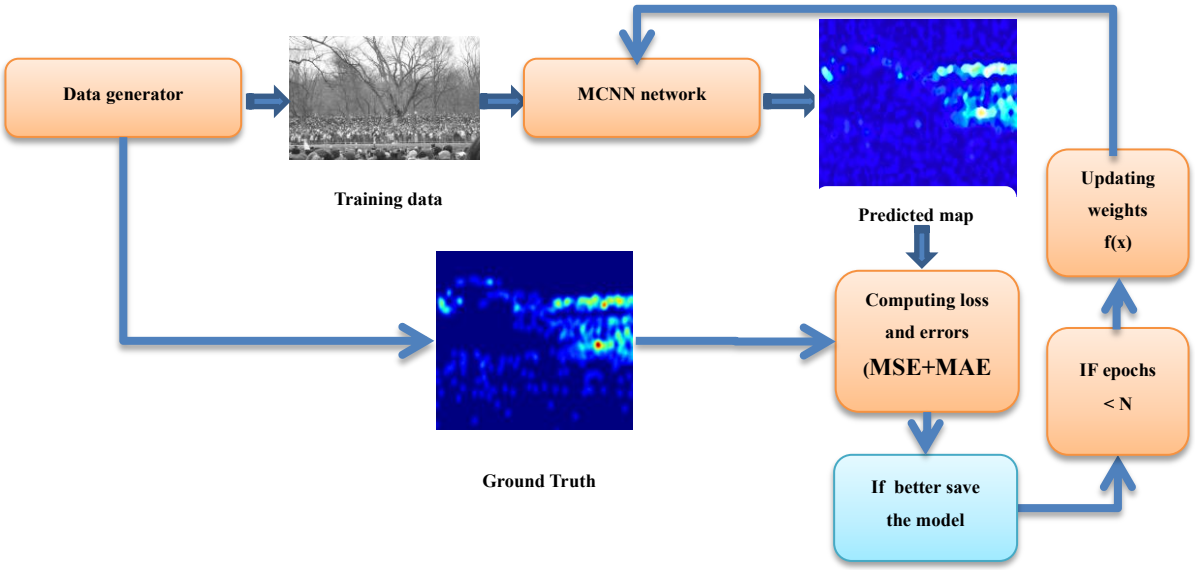


Figure III.3-12 Organizational chart of the model training (MCNN)

6- Results

We start by plotting the used loss function and metrics (MAE, MSE), respectively, as shown in Figures III.3-13, III.3-14 and III.3-15.

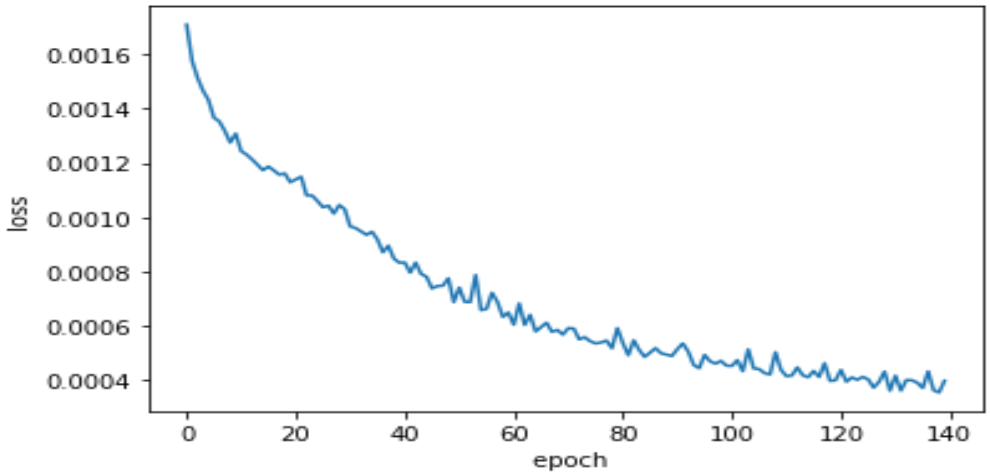


Figure III.3-13 Loss function of MCNN with fixed kernel

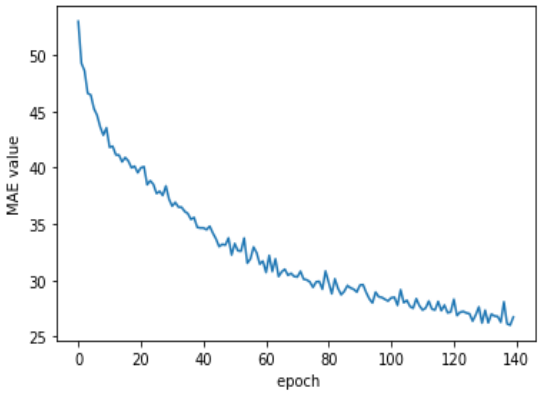


Figure III.3-14 MAE metric

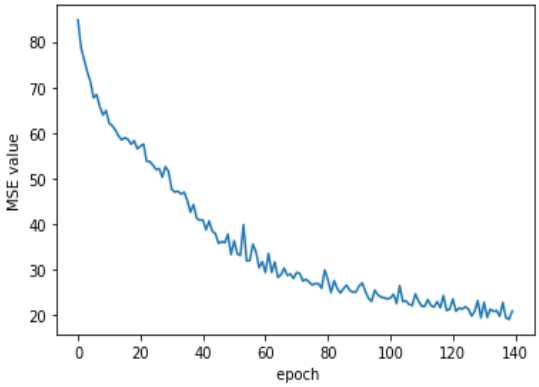


Figure III.3-15 MSE metric

Here we see that the loss function is decreasing almost in every epoch, which means the model is getting better and the network is updating correct weights.

MAE and MSE metrics are both decreasing also, which means that the model is predicting better after each epoch. These results can be checked through comparing predictions and GTs presented in Figure III.3-16.

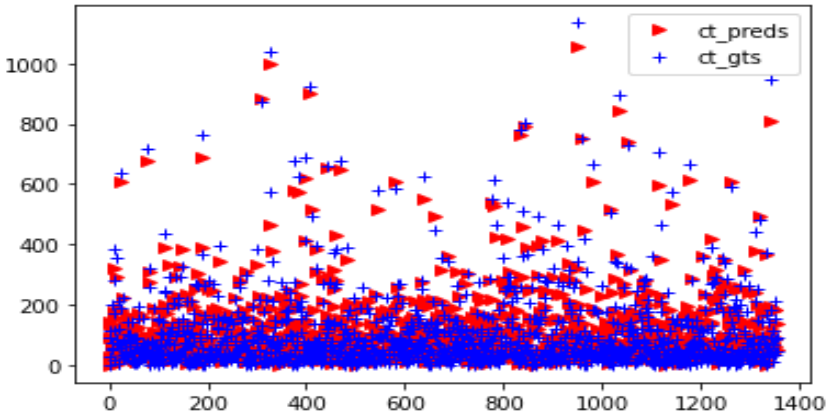


Figure III.3-16 predictions vs ground truth

The model shows good prediction values on the full dataset with small errors even in images containing high density of crowd. To compute the predicted crowd number we just have to compute the sum of points (pixel values) present in the Gaussian density map. For confirmation purpose we show two examples in Figure III.3-17. Pictures in the left corner are original images entitled “original image”, in the center we find GT entitled “gt image” followed by the actual crowd number. Prediction Gaussian density maps can be found in the right side entitled “prediction” followed by the predicted number.

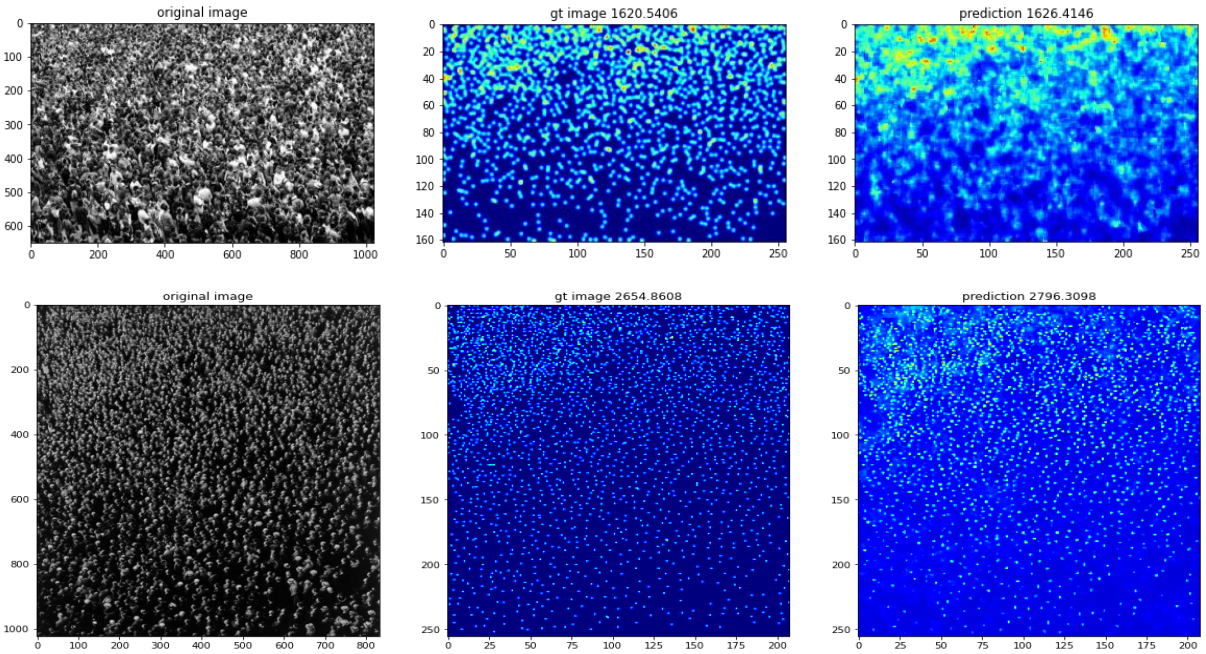


Figure III.3-17 ground truth vs predictions examples

The model reaches **MAE = 97.14** and **MSE = 131.45.**

III.3.5.1.2 MCNN with adapted Gaussian kernel size

The implementation of MCNN with adapted kernel size is similar to the one with the fixed kernel size for all the steps. The difference is in the way of generating the Gaussian density map so we will only show how to do that in this implementation.

1- Generating the ground truth

Adapted kernel size means changing the size of the distributed points by varying the standard deviation sigma according to the distances between them. This method serves to cover the surface area of the pedestrian faces efficiently.

Taking images can face perspective variation as it is mentioned before (chapter1), which produces face scale variation. Generating the same kernel size on the whole image will cause two possible problems in covering the area of the face:

1. The kernel can be bigger than the face size.
2. The kernel can be smaller than the face size.

To prevent this problem we had to compute the average of distances between N annotations points (faces) and to multiply the standard deviation by that factor. The differences between fixed and adapted Gaussian kernel are shown in Figure III.3-18. 4 points were chosen to make relation as much as we can between both sizes. The results of this technique are shown in Figure III.3-19.

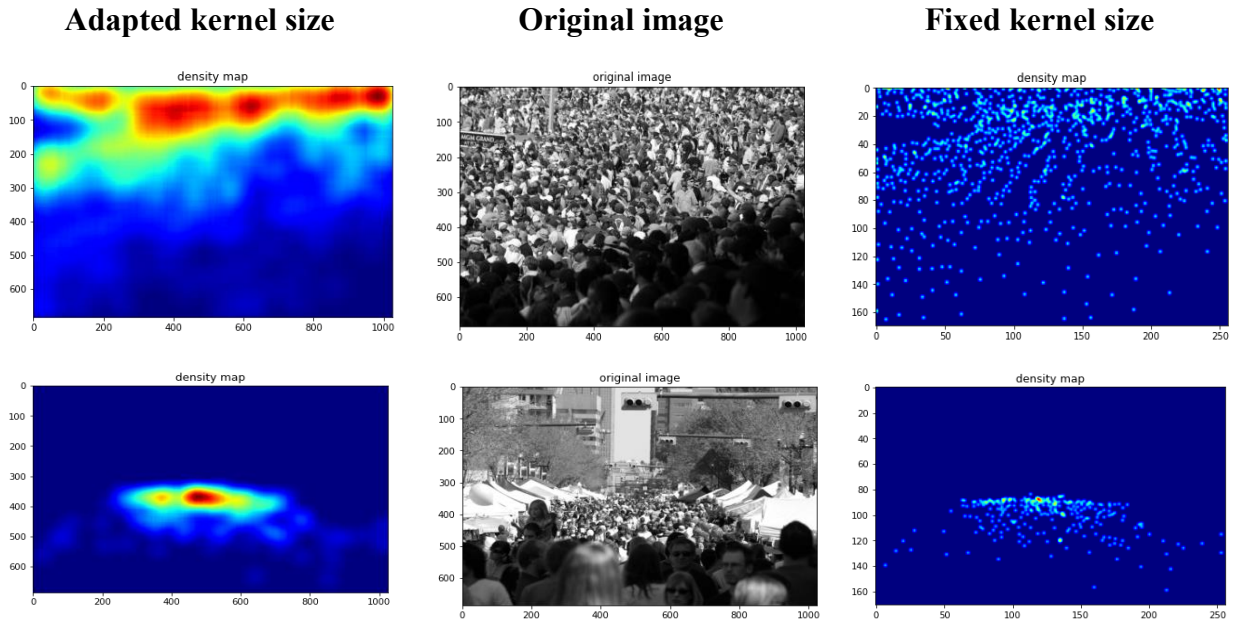


Figure III.3-18 Gaussian density maps using fixed and adapted kernel size

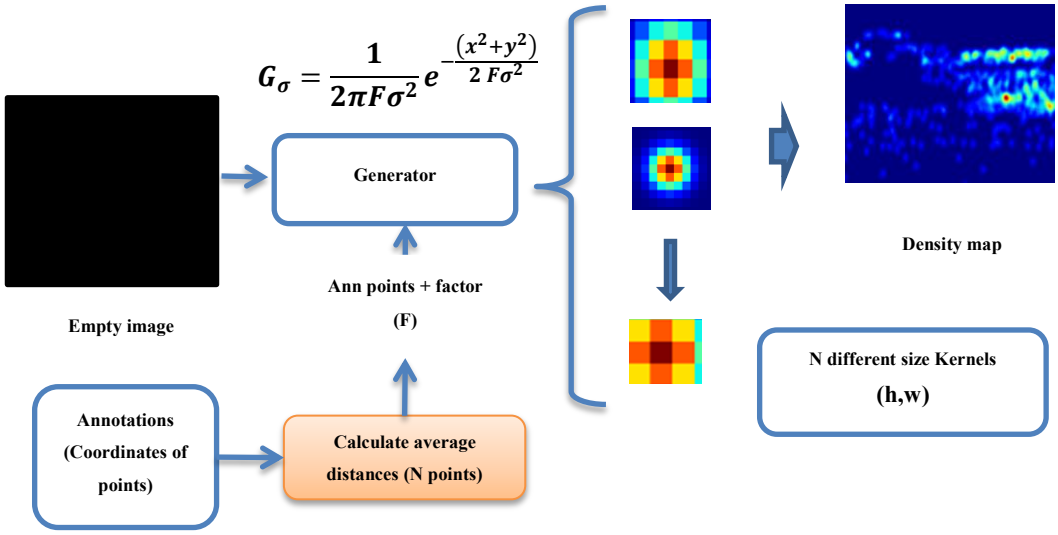


Figure III.3-19 Generating ground truth adapted kernel size

2- Results:

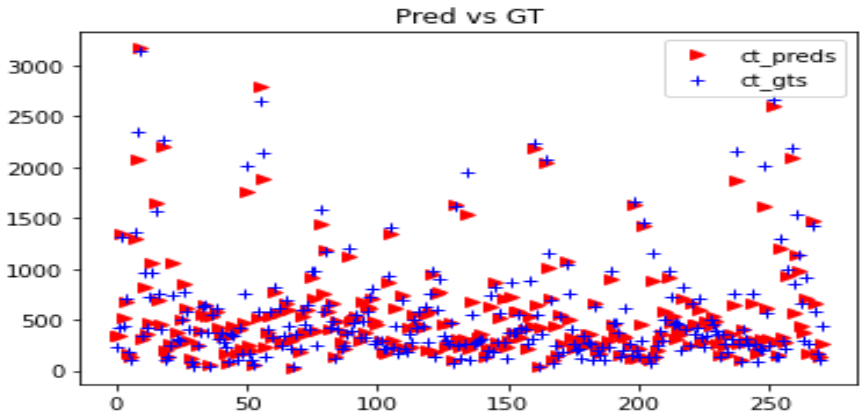


Figure III.3-20 predictions vs ground truth adapted kernel size

Figure III.3-20 illustrates a point presentation of predictions and GTs. The closer the red triangle is to the plus sign, the better the result is.

The adapted kernel size shows very good similarity to the Gaussian density map of the GTs which means the number of predicted points is closer to the actual count.

Examples of the prediction on images are given in Figure III.3-21.

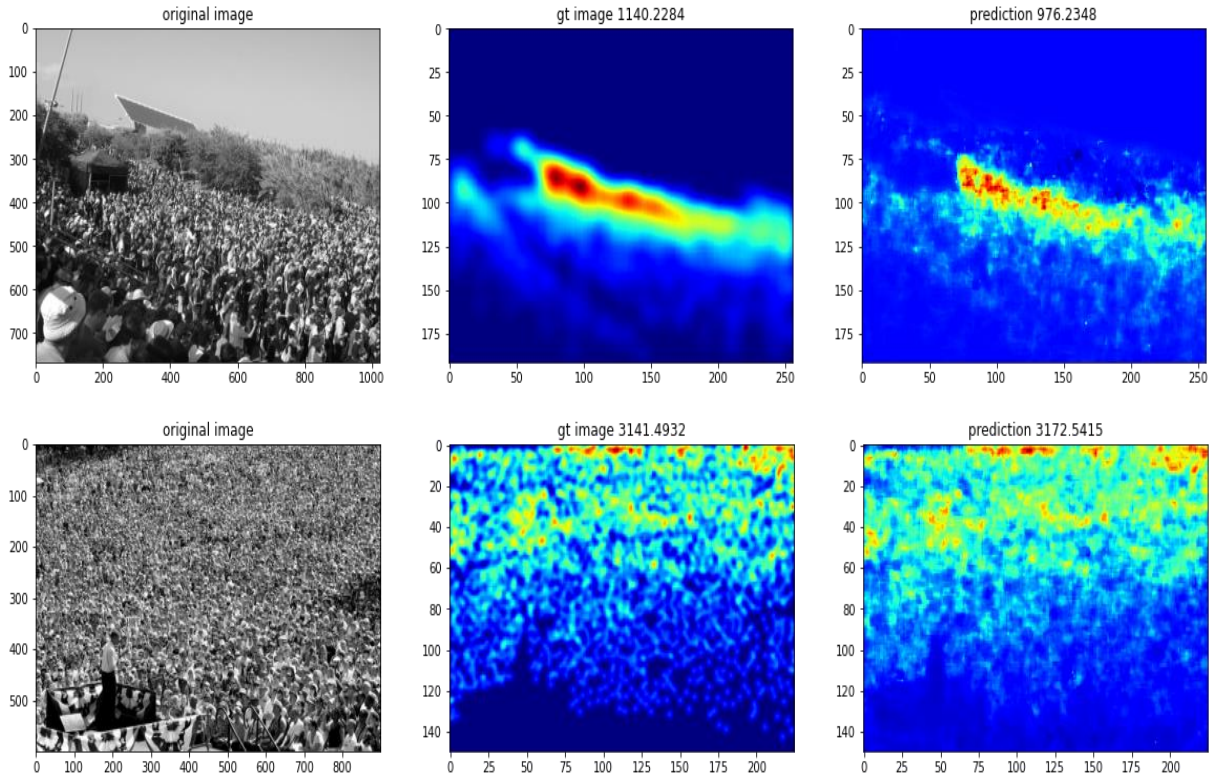


Figure III.3-21 Some predictions with adapted kernel size

The model reaches best results when **MAE** = **79.22** and **MSE** = **111.97**

III.3.5.1.3 Comparative study

Both aforementioned technics in terms of MAE and MSE were compared as shown in Table III.3-3 and tested on the same dataset (ShanghaiTech). The prediction on images is illustrated in Figure III.3-22, where it shows in Figure III.3-22-(a) points of fixed kernel technic and in Figure III.3-22-(b) points of adapted kernel technic. We can see that adapted kernel method shows better results in terms of MSE and MAE even with using the same network (MCNN). However, this method can't reach real time prediction due to high computation.

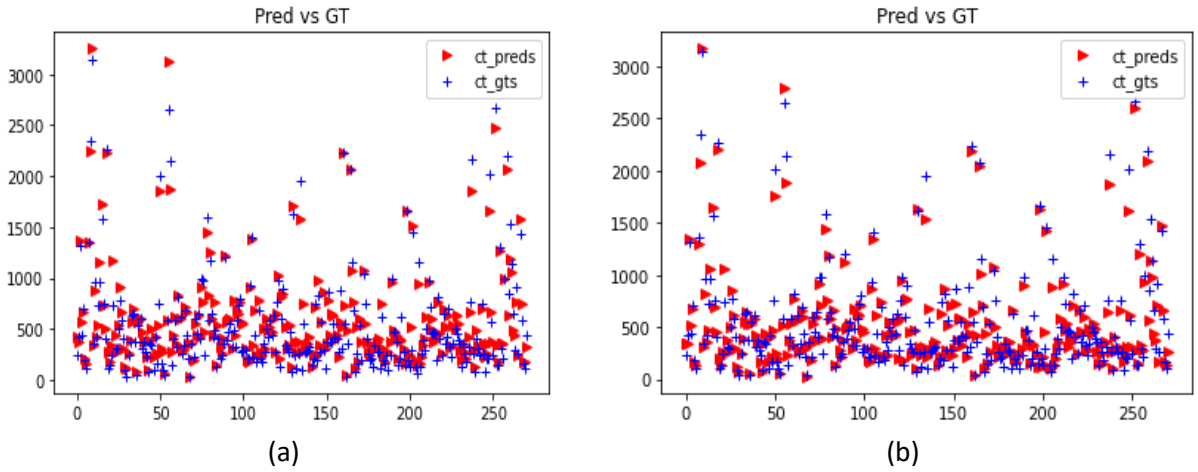


Figure III.3-22 Comparison between MCNN With fixed and adapted kernel size

Method	MAE	MSE	Response time
MCNN with fixed kernel	97.14	131.45	2.9s
MCNN adapted kernel	79.22	111.97	2.9s

Table III.3-3 MAE and MSE in both adapted and fixed kernel

III.3.6 Soft CSRNet and soft CSRNet+ implementations

These networks were created and developed by Yuhong Li, Xiaofan Zhang, Deming Chen [12] to understand highly congested scenes and perform accurate count estimation as well as to present high-quality Gaussian density maps.

III.3.6.1 Soft CSRNet implementation

Steps of Soft CSRNet implementation are almost the same as in **MCNN**; the main difference is in the network.

1- Building the soft CSRNet

We start by transfer learning from a pretrained model on large database of controlled quality called **ImageNet.** The building process is illustrated in Figure III.3-23**.**

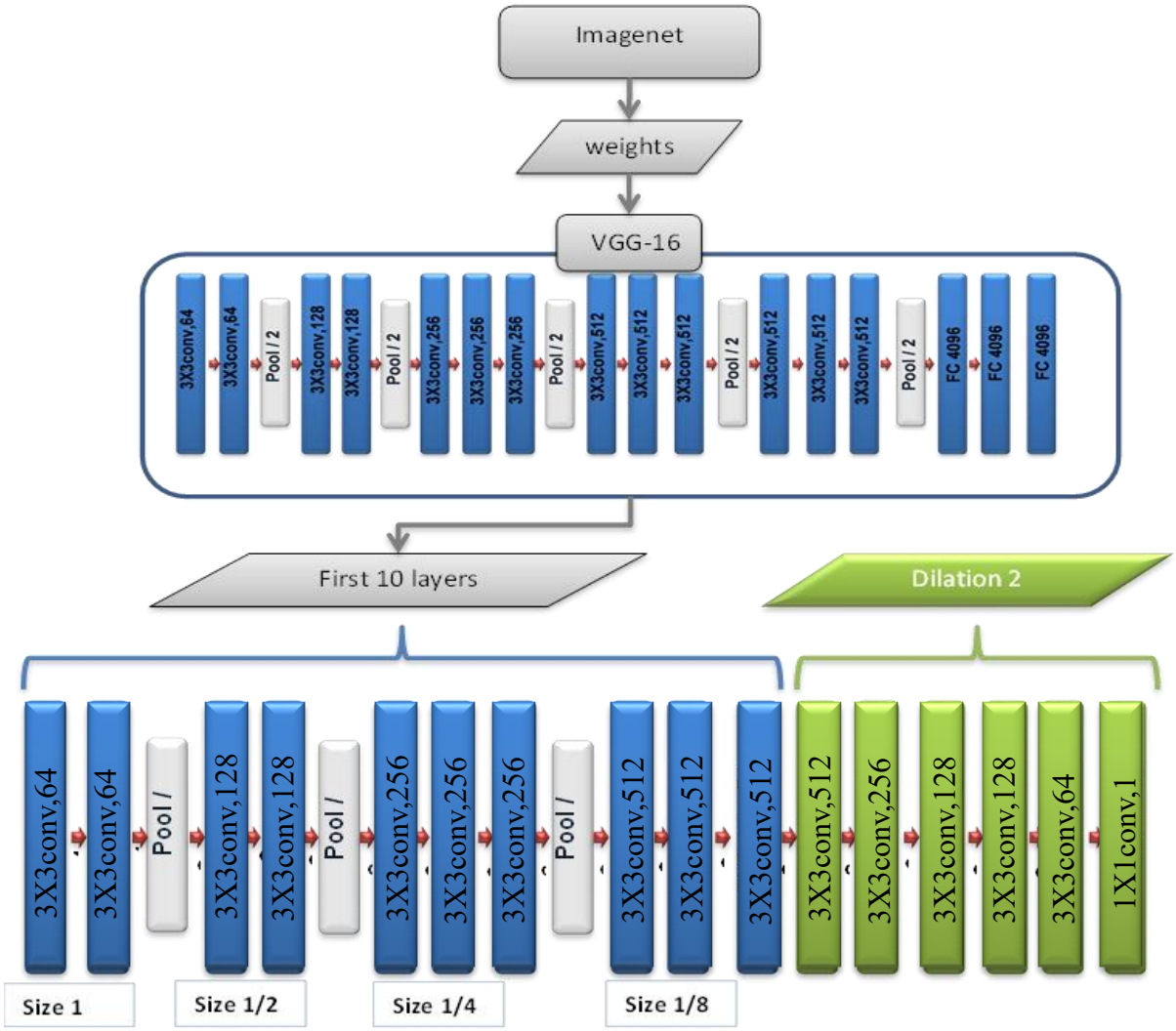


Figure III.3-23 Building Soft CSRNet

2- Training the network

The first 10 layers are kept non trainable because the VGG-16 is a pretrained model on huge dataset (ImageNet) and we keep passing images through the whole model like other Gaussian density map methods.

3- Results

After training the network for 50 epochs on VisDrone CC2020 dataset the resulting loss function, and MSE and MAE are respectively presented in Figure III.3-24 and Figure III.3-25.

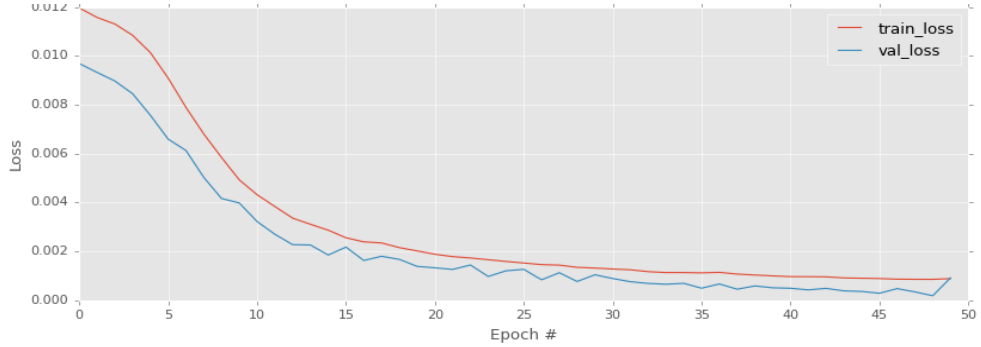


Figure III.3-24 Loss function of soft CSRNet on Visdrone dataset

We can see that the training loss function on dataset is improved in each epoch.

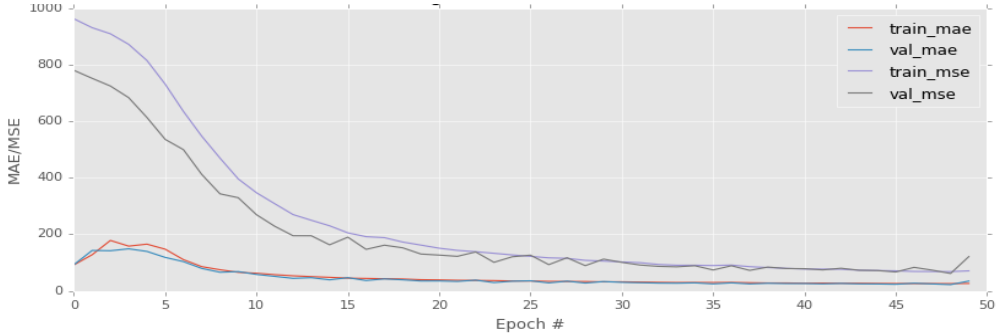


Figure III.3-25 MAE and MSE on soft CSRNet

MAEs in both train and validation are getting statistically better after each epoch.

To compare predictions and GTs a plot of their difference is generated for each image as presented in Figure III.3-26.

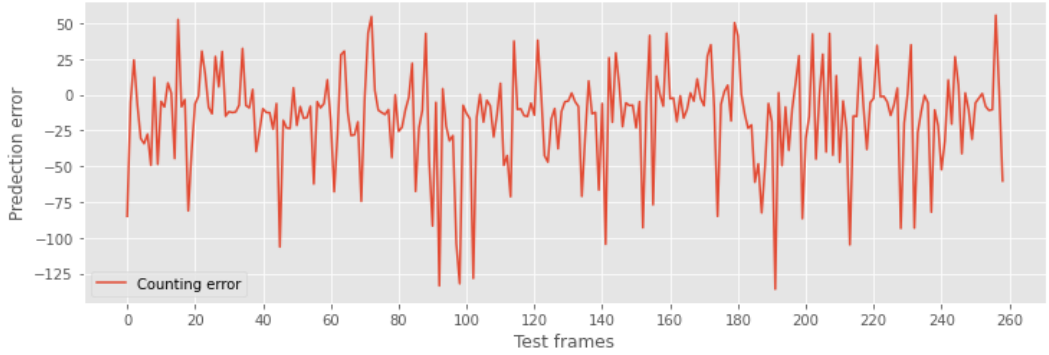


Figure III.3-26 Predictions - ground truth CSRNet

MAE in this dataset is equal to 25.76 and MSE = 43.5.

To clarify the results a plot of points representing both prediction and GTs is made and presented in Figure III.3-27.

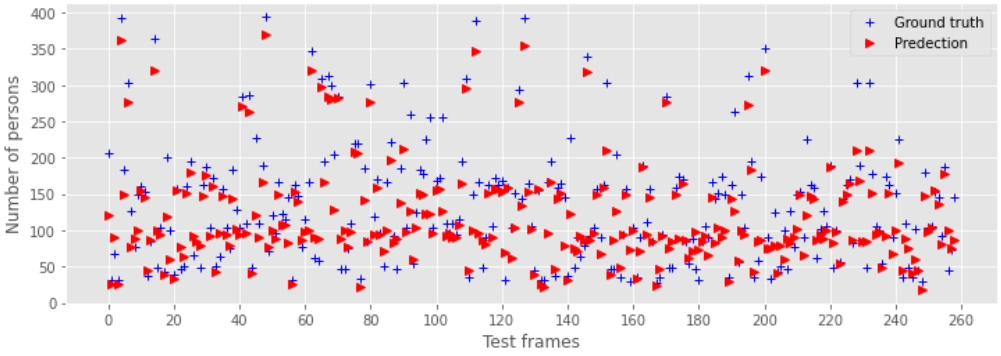


Figure III.3-27 predictions vs ground truth soft CSRNet

Soft CSRNet can capture human instances in pictures taken from considerable heights; this later made the use of drones more convenient. Some examples are given in Figure III.3-28. The model reached a response time of **7 fps** on our system and around **15 fps** on **google Colaboratory.**

Response time is computed by the following equation:

$$response\ time = \frac{1}{N}\sum_{i=0}^{N} t_i \tag{13}$$

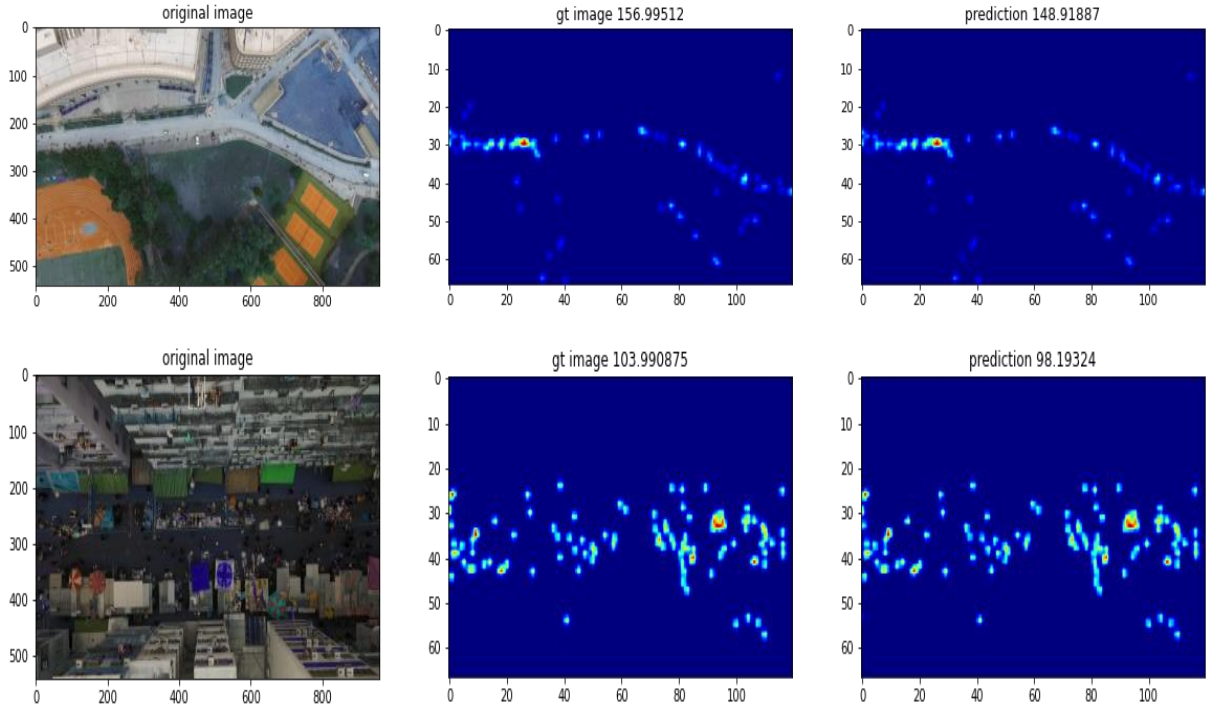


Figure III.3-28 examples of prediction on images soft CSRNet

III.3.6.2 Soft CSRNet+ implementation

Soft CSRNet+ implementation is an improved network version that is developed from the soft CSRNet network by I. Bakour, H. N. Bouchali, S. Allali and H. Lacheheb [12]. It is same as soft CSRNet in all the layers of VGG-16; the main difference is in the back-end dilation convolution where they left only 5 layers instead of 6 and decreased the number of the filters. This modification may decrease accuracy by reducing the number of the trainable parameters but permits less computation time.

The implementation of the soft CSRNet+ is almost the same as the soft CSRNet so we will consider, as shown in Figure III.3-29, only the parts that are different.

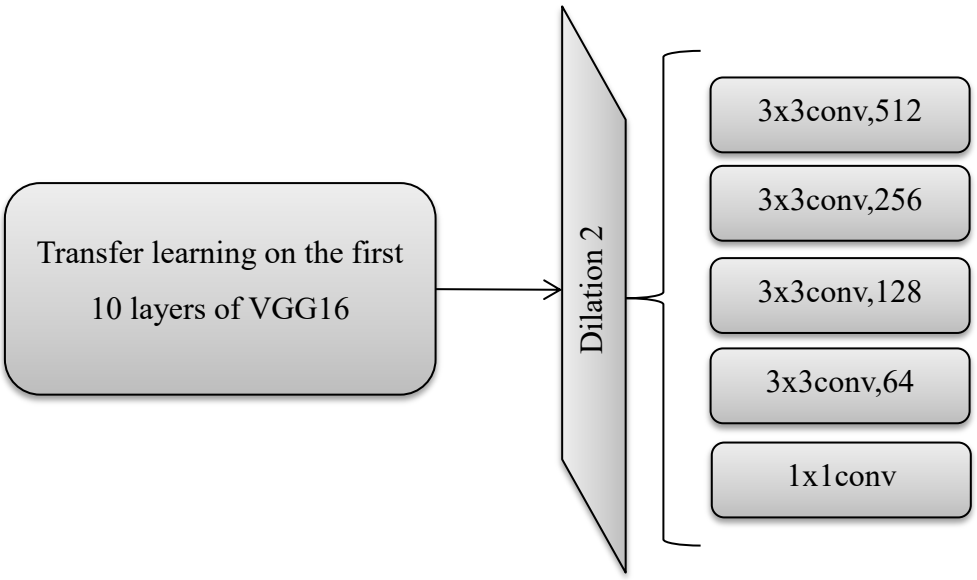


Figure III.3-29 Architecture of soft CSRNet +

1- Results:

After training the network on the same datasets, the plot of the loss function, and the MSE and MAE can be derived as depicted in Figures III.3-30 and III.3-31, respectively.

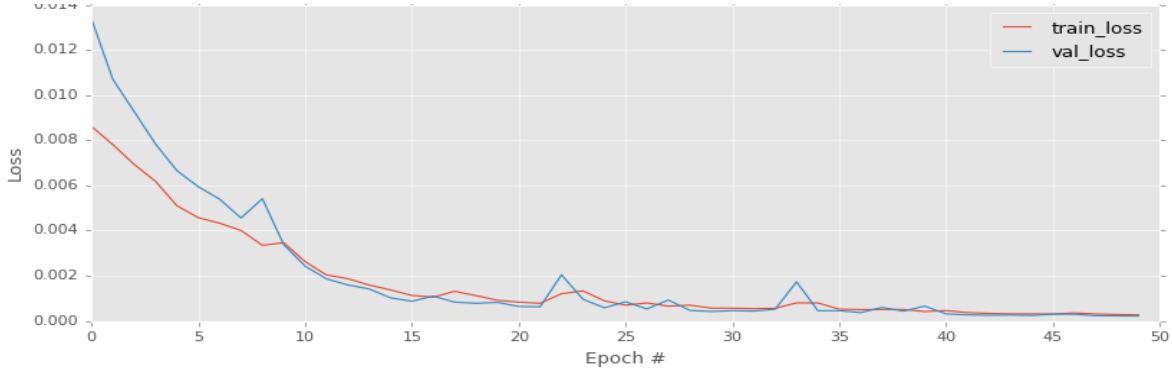


Figure III.3-30 Loss function of soft CSRNet+

Loss function of training and validation are both diminishing after each epoch which means the model is improving.

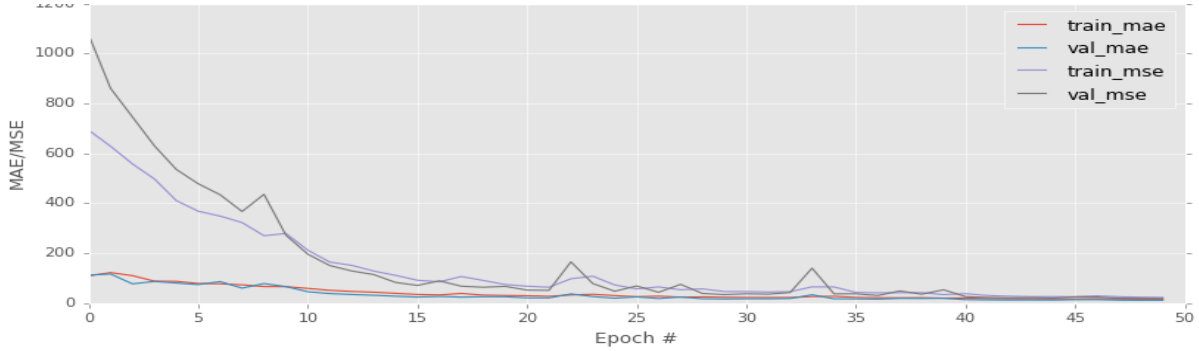


Figure III.3-31 MAE and MSE soft CSRNet+

MAE and MSE of both train and validation are falling off after epoch which means the model is updating the suitable weights. To visualize the results a plot of points representing both prediction and GTs is made and presented in Figure III.3-32.

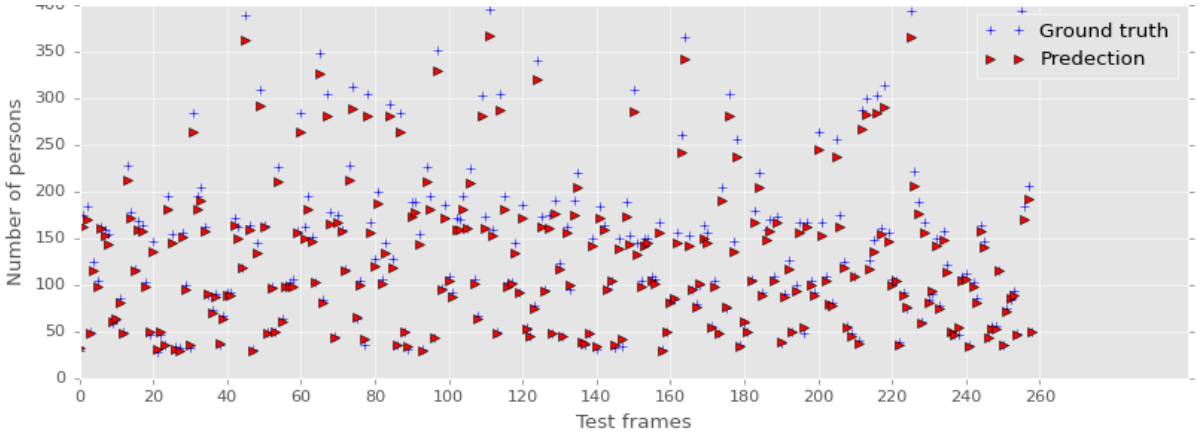


Figure III.3-32 predictions vs ground truth Soft CSRNet+

As we can see triangles that represent the predictions are closer to the plus signs representing GT (true predictions) which indicates that this model is better. To elucidate the difference between predictions and GTs values we made a graph representing the difference in each image presented in Figure III.3-33.

The pred – GT retrieved in 260 images shows that difference values are confined in the field (-45, 12), where the MAE = 12.86 and MSE=36.7.

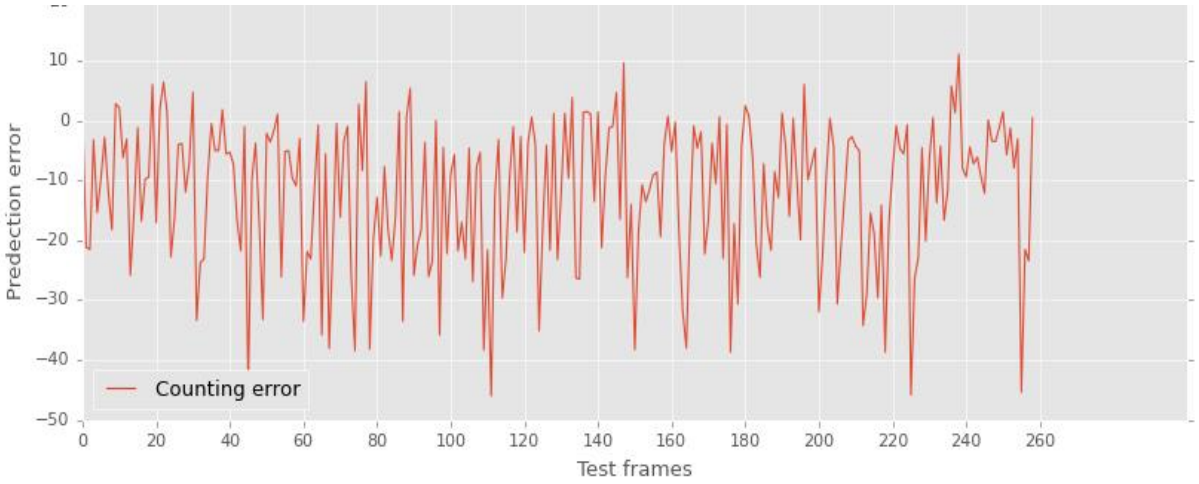


Figure III.3-33 Predictions - GT values

III.3.6.3 Comparative study

We tested both networks (soft CSRNet and soft CSRNet+) on three different datasets and computed MAE, MSE and the average response time. The results are presented in Table III.3-4.

Method	ShanTech A		ShanTech B		Visdrone ECCV 2020		UCF-CC-50		Response time
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	(fps)
Soft CSRNet	65.2	115	10.4	16.3	25.76	43.5	246.1	357.5	15
Soft CSRNet+	63.2	102	10.6	15	12.86	36.7	266.1	397.5	24

Table III.3-4 Soft CSRNet vs Soft CSRNet+

To demonstrate the efficiency of the soft CSRNet+ as well as the quality of the generated Gaussian density maps, two examples from two different sequences, are shown in

Figure III.3-34. These examples represent various crowd densities and different images illuminance from different scenes. From figure III.3-34 we can observe that the proposed soft CSRNet+ method can overcome the problem of the scale variations. Besides, the generated Gaussian density maps are more similar to the ground truth.

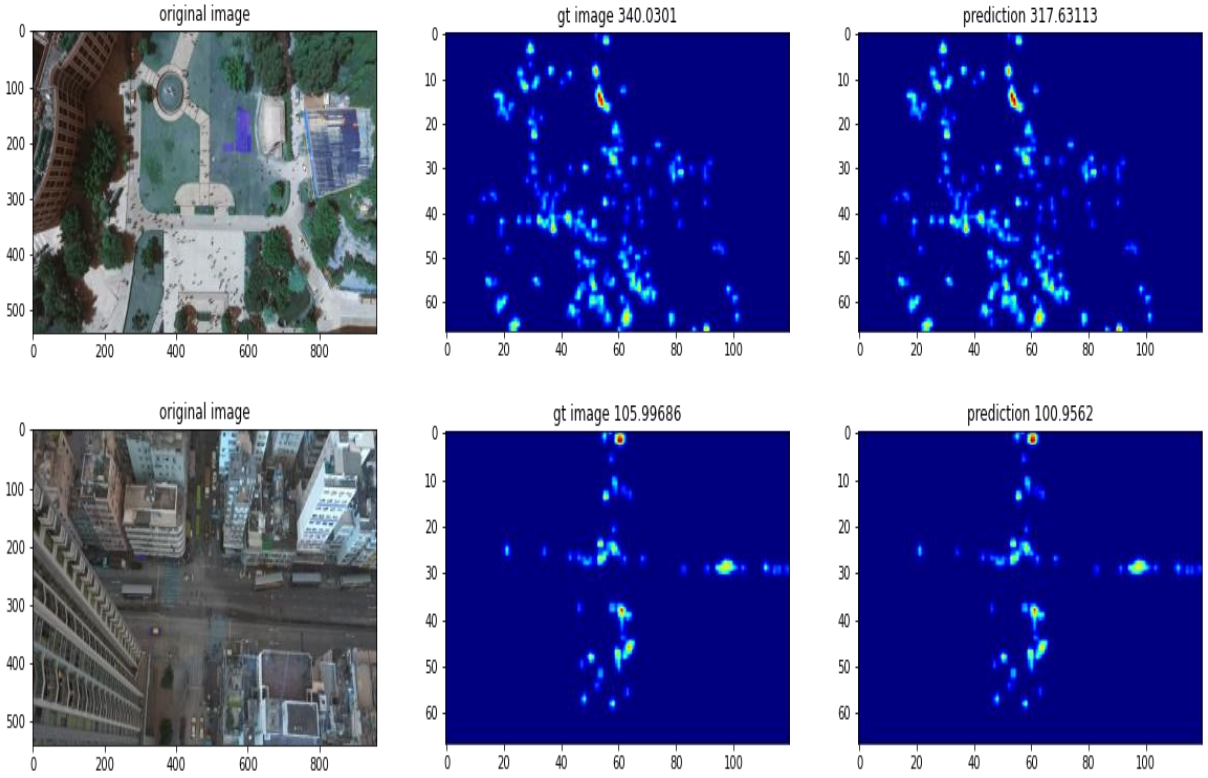


Figure III.3-34 Example of results from Soft CSRNet+ method

III.3.7 Overall Comparative study

Deep learning engineers made vast number of specialized networks in crowd counting task. This huge number of networks made the task very challenging.

To prove network efficiency it should be validated on the same dataset and achieved better results in term of errors and response time. In this part we will compare multiple networks and their values. The results are presented in Table III.3-5.

The additional used networks values in the comparison are found in authors articles.

Table III.3-5 Performance of each method on the existing crowd counting datasets

Method	ShanghaiTech A		ShanghaiTech B		UCF-QNRF		UCF-CC-50		Visdrone ECCV 2020	
	MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
SPN [25] (2019)	**61.7**	**99.5**	9.4	14.4	--	--	259.2	335.9	22.5	25.76
SKT [26](2020)	62.7	102.3	7.98	13.1	**96.4**	**156.8**	259.2	335.9	25.8	30.29
Soft CSRNet+	63.2	102	10.6	15	--	--	266.1	397.5	**12.86**	36.7
Soft CSRNet	65.2	115	10.4	16.3	--	--	246.1	357.5	25.76	43.5
DENet [27] (2020)	65.5	101.2	**9.6**	15.4	--	--	241.9	345.4	19.3	31.8
SCAR [28] (2019)	66.3	114.1	9.5	**12.2**	107	183	**212.2**	**243.7**	13.39	**17.98**
CSRNet[24] (2018)	68.2	115	10.6	16	--	--	266.1	397.5	25.3	31.81
MCNN adapted	79.2	111.9	10.8	41.3	272	420	377.6	509.1	--	--
MCNN fixed	97.1	131.4	11.6	43	277	426	385.1	517.2	--	--

Note: The **BOLD** fonts represent the first place (1^{st})

Networks in Table III.3-5 are classified in a descending order, as we can see Soft CSRNet+ is in the best three networks followed by Soft CSRNet. MAE and MSE metrics showed that this network is very good in almost all types of datasets; **ShanghaiTech A, ShanghaiTech B, UCF-CC-50** and **Visdrone ECCV 2020.** It scored the first place in **Visdrone** dataset. The soft CSRNet+ can be considered as a very efficient network especially in real time crowd counting, it could overcome the scale variation problem.

III.4 Conclusion

In this chapter we showed the implementation of two crowd counting methods. One implementation of object detection method (MASK R-CNN) and four implementations in density map estimation (MCNN with fixed and adapted kernel, Soft CSRNet and Soft CSRNet+). In the object detection method we implemented Mask R-CNN network on small sparse crowd and in density map method we implemented three networks. The first one (MCNN) is implemented using two technics which are fixed kernel and adapted kernel. The Other two networks are respectively soft CSRNet and soft CSRNet +.

Specifying the suitable method for crowd type is obligatory, Object detection method is only applicable for small and sparse crowd, meanwhile Gaussian density map method showed rewarding results in crowded situations. On the other hand different networks in this method are ranked in term of error and response time. Estimation the crowd number needs to be at real time with accurate values in order to prevent from bad coincidences.

Finally, we can say that Soft CSRNet+ is sufficient for the crowd counting in real time.

General conclusion

Crowd counting is an essential task in crowd image analysis due to nowadays related disasters. It is challenging when the proposed algorithm is particularly exposed to data collected in diverse conditions like scale variation, occlusion, irregular people distribution and diversity of scenes. However, researchers from the CV, DL, ML and transfer learning have shown significant progress, in particular in the last 10 years. This work was made in order to find better solutions for crowd counting using DL, ML, CV and transfer learning. To overcome the related challenges CNNs were used as feature extractors, this last showed high efficiency and precision. Diversity of crowd situations divided the crowd counting methods into three that are based on; Object detection, regression and Gaussian density map estimation.

Object detection method showed good prediction on sparse and small crowd scenes since it recognize the object then localize it, but it showed poor prediction in crowded ones. Meanwhile, using Gaussian density map method showed high efficiency in both; it uses convolved point annotations with the Gaussian kernel to generate the ground truth, this last made CNNs learn to capture details in the level of pixels. This method uses MAE and MSE as metrics and loss functions to increase the learning while giving answers to the network. The proposed networks are MCNN with adapted and fixed kernel, Soft CSRNet and Soft CSRNet+. However, these networks differ in the shape. Both MCNN methods use different size of convolution kernel to extract different size of features but the poor number of kernels made it unreliable in comparison with Soft CSRNet, this last uses a novel technic called transfer learning. It is composed by the first ten layers of VGG-16 as front-end of the structure and dilation convolution as back-end, transferring the learning from the pretrained model accelerates the training process which made this network very suitable for crowd counting.

Finally, the implementation of these methods showed that the crowd counting can be done by many different networks, however choosing the right one is depended on the accuracy, processing time and crowd situation. The proposed Soft CSRNet+ showed very good results after being tested on different datasets.

References

References:

[1] T. Yang, C. H. Foh, F. Heliot, C. Y. Leow and P. Chatzimisios, "Self-Organization Drone-Based Unmanned Aerial Vehicles (UAV) Networks," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761876.

[2] Zhu P, Wen L, Du D, et al. Detection and Tracking Meet Drones Challenge[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2021 (01): 1-1.

[3] https://paperswithcode.com/paper/multi-source-multi-scale-counting-in.

[4] https://paperswithcode.com/paper/single-image-crowd-counting-via-multi-column-1.

[5] Z. Liu et al., "VisDrone-CC2021: The Vision Meets Drone Crowd Counting Challenge Results," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 2830-2838, doi: 10.1109/ICCVW54120.2021.00317.

[6] Copeland, B.J.. "Artificial intelligence". Encyclopedia Britannica, 18 Mar. 2022, https://www.britannica.com/technology/artificial-intelligence. Accessed 27 April 2022.

[7] What Is Deep Learning https://www.mathworks.com/discovery/deep-learning.html.

[8] what is computer vision by IBM, https://www.ibm.com/topics/computer-vision.

[9] Computer Vision 1st Edition (Ballard & Brohttps, 1982).

[10] Introductory Techniques for 3-D Computer Vision (Trucco & Verri, 1998).

[11] Computer Vision 1st Edition (Sockman & Shapiro, 2001).

[12] N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2018, pp. 397-402, doi: 10.1109/ASET.2018.8379889.

[13] Maan, A. K.; Jayadevi, D. A.; James, A. P. (1 January 2016). "A Survey of Memristive Threshold Logic Circuits". IEEE Transactions on Neural Networks and Learning Systems. PP (99): 1734–1746. arXiv:1604.07121. Bibcode:2016arXiv160407121M. doi:10.1109/TNNLS.2016.2547842. ISSN 2162-237X. PMID 27164608. S2CID 1798273.

[14] https://www.v7labs.com/blog/neural-networks-activation-functions.

[15] Huang Yi, Sun Shiyu, Duan Xiusheng and Chen Zhigang, "A study on Deep Neural Networks framework," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1519-1522, doi: 10.1109/IMCEC.2016.7867471.

[16] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[17] Bakour, I., Bouchali, H. N., Allali, S. and Lacheheb, H. (2021, February). Soft-CSRNet: Real-time Dilated Convolutional Neural Networks for Crowd Counting with Drones. In 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH) (pp. 28-33). IEEE.

[18] C. C. Loy, K. Chen, S. Gong, and T. Xiang, “Crowd counting and profiling: Methodology and evaluation,” Modeling, Simulation and Visual Analysis of Crowds, p. 347, 2013.

[19] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[20] Mei Jiang1 and Yanyun Zhao, 3rd International Conference on Multimedia Technology（ICMT 2013), An Improved Method of Crowd Counting Based on Regression.

[21] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.

[22] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1091–1100.

[23] https://cocodataset.org/#home.

[24] Li, Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition .

[25] Chen, X., Bin, Y., Sang, N., & Gao, C. (2019, January). Scale pyramid network for crowd counting. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1941-1950). IEEE.

[26] Liu, L., Chen, J., Wu, H., Chen, T., Li, G., & Lin, L. (2020, October). Efficient crowd counting via structured knowledge transfer. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 2645-2654).

[27] Liu, L., Jiang, J., Jia, W., Amirgholipour, S., Wang, Y., Zeibots, M., & He, X. (2020). Denet: A universal network for counting crowd with varying densities and scales. IEEE Transactions on Multimedia, 23, 1060-1068.

[28] Gao, J., Wang, Q., & Yuan, Y. (2019). SCAR: Spatial-/channel-wise attention regression networks for crowd counting. Neurocomputing, 363, 1-8.

[29] https://www.youtube.com/watch?v=WvhYuDvH17I

[30] https://www.itv.com/news/2021-10-18/football-stand-collapses-as-fans-celebrate-victory