

UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING



Deep Learning based Robotic Recognition System for Crowd Management

A Thesis Submitted in Partial Fulfillment of the Requirements for the M.Sc. Degree in
Artificial Intelligence Systems

Submitted by

Maram Saleh M Alfaraj

202012480

Supervised by

Professor Ebrahim Abdulla Mattar

University of Bahrain

May 2024

Defense Committee's Approval

University of Bahrain
College of Engineering
Department of Electrical and Electronics Engineering
M.Sc. in Artificial Intelligence Systems

Student MARAM SALEH MOHAMMED ALFARAJ, ID number 202012480 has defended her thesis by which she has fulfilled the requirement for the degree of M.Sc. in Artificial Intelligence Systems entitled: (Deep Learning based Robotic Recognition System for Crowd Management) on Monday, May 20, 2024.

The committee recommends awarding MSc. Degree with the grade of: PASS

(Defense Committee)

Name Signature

Professor Ebrahim A. Mattar, Supervisor

Ebrahim A. mattar

Professor Hassan Elregel, External examiner

[Signature]

Professor Wael Al Madani, Internal examiner

[Signature]



Academic department's stamp

ABSTRACT

In response to the pressing need for effective crowd management solutions for robotics, this study delves into the creation of a deep learning-based scene and face recognition system tailored for robotic crowd management. The primary objective is to develop a system that not only exhibits high levels of accuracy and efficiency but also adheres to ethical principles. This is achieved through the integration of deep learning models with multi-modal fusion approaches, allowing for the swift and comprehensive assessment of optical, auditory, and depth data to identify potential hazards and maintain crowd control. At the heart of this research lies the meticulous gathering and preparation of data. Establishing a broad and representative dataset is paramount to the success of the system. Through comprehensive dataset collection methods, encompassing diverse scenarios and environments, we ensure that the model is robust and capable of handling real-world crowd management challenges effectively. Drawing inspiration from the methodology of the Tensorflow Object Counting CSRNet, our approach follows a structured framework encompassing various stages. Data preparation involves preprocessing input data and generating density maps using Gaussian distribution functions, laying the foundation for accurate crowd assessment. A custom data generator class facilitates efficient data handling, ensuring smooth processing of batches of images and corresponding density maps. The model architecture is meticulously designed to leverage the strengths of deep learning, incorporating convolutional layers for feature extraction, and reshaping to generate output heat maps. During the training phase, a custom loss function is employed to compute the root mean square error (RMSE), optimizing model performance. Training and validation are conducted over multiple epochs, utilizing the custom data generator to enhance efficiency. Following training, the model undergoes rigorous evaluation on sample crowd images, with predictions compared against ground truth density maps to assess its accuracy and efficacy. Visualization techniques are employed to provide qualitative insights into the model's performance, facilitating a deeper understanding of its capabilities. Finally, this study presents a comprehensive approach to robotic crowd management through the integration of deep learning and multi-modal fusion techniques.

PUBLICATIONS

M. S. Alfaraj and E. Mattar, "*Artificial intelligence for crowd detection system: A survey*," 4th Smart Cities Symposium (SCS 2021), Conference, Bahrain, 2021, pp. 78-82, doi: 10.1049/icp.2022.0317.

DEDICATION

I dedicate this thesis to my family who always supplied me with unconditional love and support. To my grandma and uncle's souls who have left us before the end of this journey. To my friends who always believed in me and kept shooting at me "You can do it".

ACKNOWLEDGMENTS

The completion of this thesis could not have been possible without the extraordinary contribution of Professor Ebrahim Mattar, my master thesis supervisor.

A dept of gratitude is also owed to my family and friends for their extended support of all times. I could not have made it without you covering my back.

Last but not least, I would love to give special thanks to my parents for all of their unconditional love and support that made this possible, I made it for you!

TABLE OF CONTENTS

Defense committee’s approval	Error! Bookmark not defined.
Abstract	C
Publications	D
Dedication	E
Acknowledgments	F
List of Figures.....	J
List of Abbreviations/Acronyms	K
Chapter One	1
Introduction to the study	1
Chapter One: Introduction to the study.....	2
1.1. Overview	2
1.2. Problem Statement	4
1.3. Objectives of the study	6
1.4. Scope limitation	7
1.5. Research questions	10
1.6. Thesis plan:.....	11
Chapter Two	12
Literature Review	12
Chapter Two: Literature Review.....	13
2.1. Crowd Control System:.....	13
2.2. Crowd Counting:.....	15
2.3. Traditional methods:	17
2.4. Deep Learning Approaches:	18
2.5. Crowd Action Recognition:.....	21
2.6. Traditional methods:	24
2.7. Deep Learning Approaches:	25
2.8. Crowd Scene Dataset:.....	28
2.9. A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis:.....	30
2.10. Abnormal Detection Using Interaction Energy Potentials:	33
2.11. A Streakline Representation of Flow in Crowded Scenes:	34
2.12. Deep Learning-Based Crowd Scene Analysis Survey:.....	37
2.13. Image Processing Techniques for Crowd Density Estimation Using a Reference Image:.....	39

2.14. Pedestrian Detection with Unsupervised Multi-Stage Feature Learning:.....	41
2.15. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks:.....	42
2.16. Deep People Counting in Extremely Dense Crowds:.....	43
2.18. ADCrowdNet:.....	46
2.19. Slicing Convolutional Neural Network for Crowd Video Understanding:	48
2.20. Abnormal Behavior Detection in Crowded Scenes Using Density Heatmaps and Optical Flow:.....	49
2.21.A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation.....	51
2.22.CNN based Sub-Pixel Mapping for Hyperspectral Images:	52
2.23. Attribute Recognition by Joint Recurrent Learning of Context and Correlation:.....	55
2.24. Current Limitations in Crowd Management Technologies.....	56
2.25. Challenges in Developing a Deep Learning-Based Crowd Control System	58
2.26. Proposed Solution: Deep Learning-Based Scene and Facial Recognition System.....	60
2.27. Ethical Considerations and Future Implications	63
Chapter Three.....	66
DEEP Learning topologies.....	66
Chapter Three: DEEP Learning topologies:.....	67
3.1. DEEP Learning topologies	67
3.2. Classification of DL approaches:	69
3.3Types of DL networks	70
3.3.1 Convolutional Neural Networks (CNNs)	70
3.3.2. Recurrent Neural Networks (RNNs)	84
3.3.3. Generative Adversarial Networks (GANs).....	87
3.3.4 Autoencoders.....	89
3.3.5 Deep Reinforcement Learning Networks (DRLNs).....	91
Chapter Four: Datasets.....	93
Chapter Four: Datasets.....	94
4.1. Introduction:.....	94
4.2.CMU action detection dataset	96
4.3. Action Similarity Labeling (ASLAN)	97
4.4.WWW Crowd dataset:.....	99
4.5. The PEdesTrian Attribute dataset (PETA).....	101
4.6.UCF Crime Detection" dataset:	103
4.7. The Hajj crowd dataset:.....	105

4.8 Justification:.....	107
Chapter five	109
Research Methdology and Experiments setup.....	109
5.1 Utilizing the TensorFlow CSRNet	111
5.2 Data Preparation:.....	112
5.3 Data generation and Gaussian Distribution Functions:	112
5.4 Model Definition:	113
5.5 Custom Loss Function:	113
5.6 Model Training:	113
5.7 Model Evaluation:	113
5.8 Visualization:	114
Chapter six.....	116
Results and analysis:.....	116
Chapter Six: Results and analysis:	117
6.1 Data Preparation and Generation:.....	117
6.2 Model Architecture:	117
6.3 Training:	117
6.4 Evaluation:	117
6.5 Visualization:	118
Chapter seven.....	122
Conclusion and Recommendation.....	122
Chapter seven: Conclusion and Recommendation	123
References	124

LIST OF FIGURES

FIGURE 1: ROBOTICS AND STEPS IN THE CROWD AND FACIAL RECOGNITION PROCESSES.	3
FIGURE 2: THE FIVE STEPS OF STANDARD CROWD AND FACIAL RECOGNITION.	7
FIGURE 3: OVERVIEW OF THE C D FACE DETECTION AND IDENTIFICATION SYSTEM, [KHALIFA ET AL.,2022].	9
FIGURE 4: SENSORS - CONVOLUTIONAL-NEURAL NETWORK-BASED IMAGE CROWD COUNTING.	13
FIGURE 5: CROWD COUNTING USING DEEP LEARNING GUIDE TO CROWD COUNTING, (KHANDELWAL , 2021).	15
FIGURE 6: ACTIONS RECOGNITION IN CROWD BASED ON COARSE-TO-FINE MULTI-OBJECT TRACKING, ...	21
FIGURE 7: FACE RECOGNITION PROCESSING FLOW.	26
FIGURE 8: FACE RECOGNITION STEPS, (BOLUWAJI, 2012).	31
FIGURE 9: SIMULATION OF HUMAN CROWD BEHAVIOR IN EXTREME SITUATIONS, (SUMBATOVICH & BEKLARYAN, 2012).	32
FIGURE 10: A STREAKLINE REPRESENTATION OF FLOW IN CROWDED SCENES, (MEHRAN, 2023).	35
FIGURE 11: DEEP LEARNING SURVEYS, (ELBISHLAWI ET.AL., 2020)	38
FIGURE 12: TRAIN A CROWD CNN WITH TRAINING SCENES (ELBISHLAWI ET AL., 2020).	40
FIGURE 13: ADCROWDNET: AN ATTENTION-INJECTIVE DEFORMABLE CONVOLUTIONAL NETWORK FOR CROWD UNDERSTANDING (LIU ET AL., 2018).	47
FIGURE 14: AN ON-DEVICE DEEP NEURAL NETWORK FOR FACE DETECTION (PIERRE, 2017).	68
FIGURE 15: A COMPREHENSIVE GUIDE TO CONVOLUTIONAL NEURAL NETWORKS (SAHA, 2018).	72
FIGURE 16: CNN LAYERS VISUALIZATION.	73
FIGURE 17: ILLUSTRATES THE STEPS OF CONVOLUTION LAYER.	74
FIGURE 18: STEPS OF MAXIMUM POOLING LAYER.	76
FIGURE 19: FLATTEN AND FULLY CONNECTED LAYERS, (JEONG, 2019).	77
FIGURE 20: SIGMOID AND RELU FUNCTIONS, (SHARMA, 2022).	78
FIGURE 21: PLOT OF EXP, (PRIYA, N.D.).	80
FIGURE 22: CODE EXAMPLE OF HOW TO CONSTRUCT CNN	81
FIGURE 23: RECURRENT NEURAL NETWORKS — UNLOCKING THE SECRETS OF SEQUENTIAL DATA (KONDREDDY, 2023).	86
FIGURE 24: GENERATIVE ADVERSARIAL NETWORKS (AGRAWAL, 2021).	89
FIGURE 25: AUTO DECODER DIAGRAM (PÉREZ-HERNÁNDEZ ET.AL., 2020).	91
FIGURE 26 THE ACTION SIMILARITY LABELING (TRAN & TORRESANI,2016).	99
FIGURE 27 A PICTURE EXAMPLE FROM WWW DATASET OF INTENSE CROWD OF PEOPLE, (ET AL,20).	101
FIGURE 28 THE PEDESTRIAN ATTRIBUTE DATASET (JUNEJO,2020).	103
FIGURE 29: UCF CRIME DETECTION" DATASET (BANGER,2022).	105
FIGURE 30: THE HAJJ CROWD DATASET (BANGAR,2022).	107
FIGURE 31: A HIGH-LEVEL MODEL ARCHITECTURE.	110
FIGURE 32: HIGH LEVEL MODEL ARCHITECTURE, MODEL BUILDING, AND VALIDATION.	115
FIGURE 33: FIRST TEST SEEN.	118
FIGURE 34: SECOND TESTING SEEN.	118
FIGURE 35: THIRD TESTING SEEN.	118
FIGURE 36: FOURTH TESTING SEEN.	119
FIGURE 37: EXPERIMENT RESULT	120
FIGURE 38: SUMMARY TABLE OF THE TESTED IMAGES RMSE	121

LIST OF ABBREVIATIONS/ACRONYMS

AIM	an attention-injective module
ASLAN	Action Similarity Labeling action similarity labeling, the Action Similarity Labeling
CNN	convolutional neural network
CMU	Carnegie Mellon University
DCN	deformed convolutional network
DDPG	Deep Deterministic Policy Gradients
DRL	deep reinforcement learning
DRLNs	Deep Reinforcement Learning Networks
ELUs	exponential linear units
GAN	Generative Adversarial Network
GPUs	Graphical Processing Units
GRU	Gated Recurrent Unit
GRUs	gated recurrent units
GTA	Grand Theft Auto
IoU	Intersection over Union
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	multilayer perceptron
MRE	mean relative error
MSE	Mean Squared Error
PETA	PEdesTrian Attribute dataset
RMSE	root mean square error
RNN	recurrent neural network
ReLU	Rectified Linear Unit
SUBESCO	SUST Bangla Emotional Speech Corpus
SVHN	Street View House Numbers dataset

SVMs	Support vector machines
------	-------------------------

CHAPTER ONE

INTRODUCTION TO THE STUDY

CHAPTER ONE: INTRODUCTION TO THE STUDY

1.1. Overview

Robots are used to regulate masses because they can manage numerous situations more effectively than humans. However, the success of these robots is contingent on the dependability and accuracy of the population control technologies used to assist them. Recent results from experiments with deep learning indicate that these algorithms may help make crowd control systems more precise and reliable. This thesis focuses on developing a face and scene recognition system for robotic crowd control. This technology's primary purpose is to assist machines in distinguishing individuals in a gathering, with a particular emphasis on potentially hazardous individuals. The technique employs algorithms for deep learning to observe live video for suspicious activity. The system's scene recognition function monitors for any alterations that could jeopardize the safety of the population and notifies the appropriate authorities. This requires monitoring traffic patterns, searching for illicit substances, and observing anything out of the ordinary. Face recognition algorithms employing deep learning scrutinize populations of individuals based on their facial characteristics in order to identify potential hazards. Observing a person's facial expressions, body language, and other behavioral clues can assist in determining whether they pose a threat (Zhang et.al., 2018).

In today's world, where the risk of terrorist attacks and other acts of violence is high, the development of this system is of particular importance. The system's ability to detect and identify potential hazards in real-time will enable swift and effective responses to mitigate potential dangers. In the literature review chapter of this thesis, several research studies and datasets pertaining to crowd control, crowd enumeration, and deep learning approaches for crowd scene analysis are discussed. In addition to examining the challenges of deep learning and alternative solutions, the review of relevant literature examines these issues as well. Chapter four describes some of the well-known datasets pertinent to the topic of the thesis, such as the "Street View House Numbers" (SVHN) dataset, the CMU action detection dataset, and the Hajj throng dataset. Chapter three on deep learning topologies provides an introduction to deep learning and its classification, DL network types, CNN architectures, deep learning challenges, and alternative solutions. This chapter describes the numerous varieties of deep learning networks, including CNNs, RNNs, and GANs. In addition, it examines the challenges of deep

learning, such as overfitting and the need for large quantities of data, as well as alternative solutions, such as transfer learning and data augmentation (Zhang et.al., 2018).

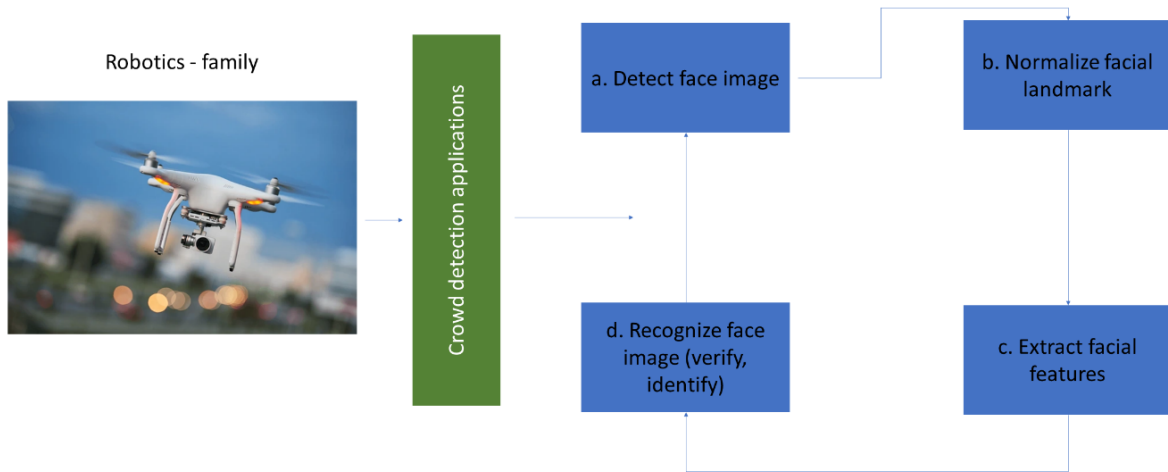


Figure 1: Robotics and Steps in the crowd and facial recognition processes.

The approach, setup, and installation processes, as well as the model-training steps, are all detailed in this thesis is detailed in chapter five. This chapter details the software libraries, and programming languages that were used throughout the tests. Data preparation, model construction, and hyperparameter tweaking are all covered, and a detailed training manual is included as well. The computer specs, software libraries, and programming languages utilized in the tests are all detailed in the part devoted to setting up the experiments. In this chapter, we also discuss normalization and data augmentation, two preprocessing techniques used to prepare datasets for training. Chapter five also provides a detailed breakdown of the deep learning model's architecture, hyperparameter tuning, and optimization strategies.

The study's conclusions, together with performance indicators and an assessment of the created system, are presented in the results and analysis chapter. The accuracy, dependability, and speed of the system's real-time threat detection are assessed in this chapter. The creation of a deep learning-based scene and face recognition system for

robotic crowd management is summarized in this thesis conclusion. Everybody will feel safer in public areas thanks to the system's ability to identify potential threats in real-time and respond with speed and effectiveness. Together, the system's scene and face recognition components will provide dependable and accurate results. While the facial recognition component examines the facial features of people in a crowd to identify any potential threats, the scene recognition component scans the surrounding area for any threats. Accuracy, precision, recall, and F1-score are just a few of the measures that will be used to assess the system's performance. Overall, the creation of the system signifies a substantial leap in crowd management technology, especially in open areas like stadiums, shopping centers, and airports. The system's precision and dependability in identifying potential threats will provide a degree of protection that was previously impractical, making everyone's safety in public settings safer. The system is flexible and effective as a crowd control tool since it can be trained on multiple datasets and can be adjusted to varied circumstances. In conclusion, the creation of this face and scene recognition system for robotic crowd management based on deep learning is a significant step in enhancing security and safety for the general population (Zhang et.al., 2018).

1.2. Problem Statement

The crowd management technologies now in use in public places often fall short in terms of accuracy and dependability. These systems often depend on manual monitoring, which may be difficult to operate and subject to human mistakes, especially in congested conditions. Deep learning algorithms have the potential to increase the precision and dependability of crowd management systems. A deep learning-based system that can efficiently identify possible risks in real-time and support robots in acting to minimize these hazards is still required, however. Public safety must prioritize crowd management since failing to do so may seriously injure both people and property. Effective crowd management methods are more important than ever due to the rise in the likelihood of terrorist attacks and other violent crimes. Human involvement has been the mainstay of crowd management in the past, although this method is often inadequate when dealing with big groups. Therefore, the creation of automated systems that can aid human operators in managing crowds and spotting potential threats is necessary. It has been

shown that deep learning algorithms may increase the precision and dependability of crowd control systems. These algorithms can swiftly and correctly evaluate massive volumes of data, producing trustworthy conclusions in real-time. It has been shown that the application of deep learning algorithms in crowd management systems increases the precision of picture analysis and crowd counting, making it simpler to identify possible hazards instantly (Ren et.al., 2015).

A deep learning-based system that can efficiently identify possible risks in real-time and support robots in acting to minimize these hazards is still required, however. The existing crowd control technologies still depend on manual monitoring, which may be difficult to handle and is subject to human mistakes, especially in crowded conditions. Additionally, obtaining a large amount of high-quality data, which can be difficult, is necessary for the development of a deep learning-based crowd-control system. The creation of a deep learning-based scene and face recognition system for robotic crowd management that is capable of accurately identifying possible threats in real-time is thus the main issue this thesis attempts to solve. Everybody will feel safer in public areas thanks to the system's ability to identify potential threats in real-time and respond with speed and effectiveness. In conclusion, the crowd management technologies now in use in public areas often fall short in terms of accuracy and dependability. Deep learning algorithms have the potential to increase the precision and dependability of crowd management systems. A deep learning-based system that can efficiently identify possible risks in real-time and support robots in acting to minimize these hazards is still required, however. The creation of such a system would greatly improve public safety and security and provide a dependable and effective crowd management option. A deep learning-based scene and face recognition system for robotic crowd management is the goal of this thesis. Threat detection in real time is a strength of this technology. By providing a dependable and effective crowd management alternative that inspires faith in citizens and authorities, this solution will increase public safety and security. In conclusion, deep learning algorithms may increase the precision and reliability of crowd control systems. Making public spaces safer requires developing a system that can quickly identify real-time hazards and assist robots in mitigating them (Ren et.al., 2015).

1.3. Objectives of the study

Using deep learning-based scene and face recognition, we set out to develop a real-time hazard detection system for robotics crowd control in this thesis article. The following are some of the objectives we will pursue:

1. Examine recent advances in technologies such as face recognition systems, deep learning algorithms, and crowd management.
2. This thesis objective is to develop a real-time, threat-aware, robotic crowd-control system based on scene and facial recognition enabled by deep learning.
3. The objective is to evaluate how well the new system functions and how it compares to competing population management solutions.

This thesis primary objective is to analyze the state-of-the-art in the disciplines of crowd management systems, deep learning algorithms, and face recognition systems and to provide recommendations based on this analysis. The literature review will lay the foundation for developing a system for scene and face recognition using deep learning. In addition, it will cast light on the problems and limitations of existing systems and offer recommendations for improvement.

The second goal of this thesis is to create a scene and face recognition system for robotic crowd management that is based on deep learning and is capable of accurately identifying possible hazards in real-time. The technology will examine real-time camera footage using deep learning algorithms to find unusual motions or behavior that could be dangerous. The scene recognition part of the system will be in charge of examining the immediate area to find any alterations that can endanger the security of the crowd. The system's face recognition capability analyzes the facial characteristics of crowd members using deep learning algorithms to spot any possible dangers. The final goal of this thesis is to assess the system's effectiveness and compare it to other crowd management methods already in use. Accuracy, precision, recall, and F1-score are just a few of the measures that will be used in the assessment.

The comparison will show the created system's advantages and disadvantages and point out opportunities for development. In conclusion, the main goals of this thesis are to

create a deep learning-based scene and facial recognition system for robotic crowd control that is capable of reliably and accurately identifying potential threats in real-time, as well as to assess the system's performance and compare it to other crowd control systems already in use. The accomplishment of these goals will considerably improve public safety and security and provide a dependable and effective remedy for crowd control. The development of a scene and facial recognition system for robotic crowd control that accurately detects potential threats in real-time, as well as a thorough evaluation and comparison of this system to other crowd management techniques, are the two main goals of this thesis. By achieving these goals, crowd control will have a dependable and effective solution, greatly enhancing public safety and security (Ren et.al., 2015).

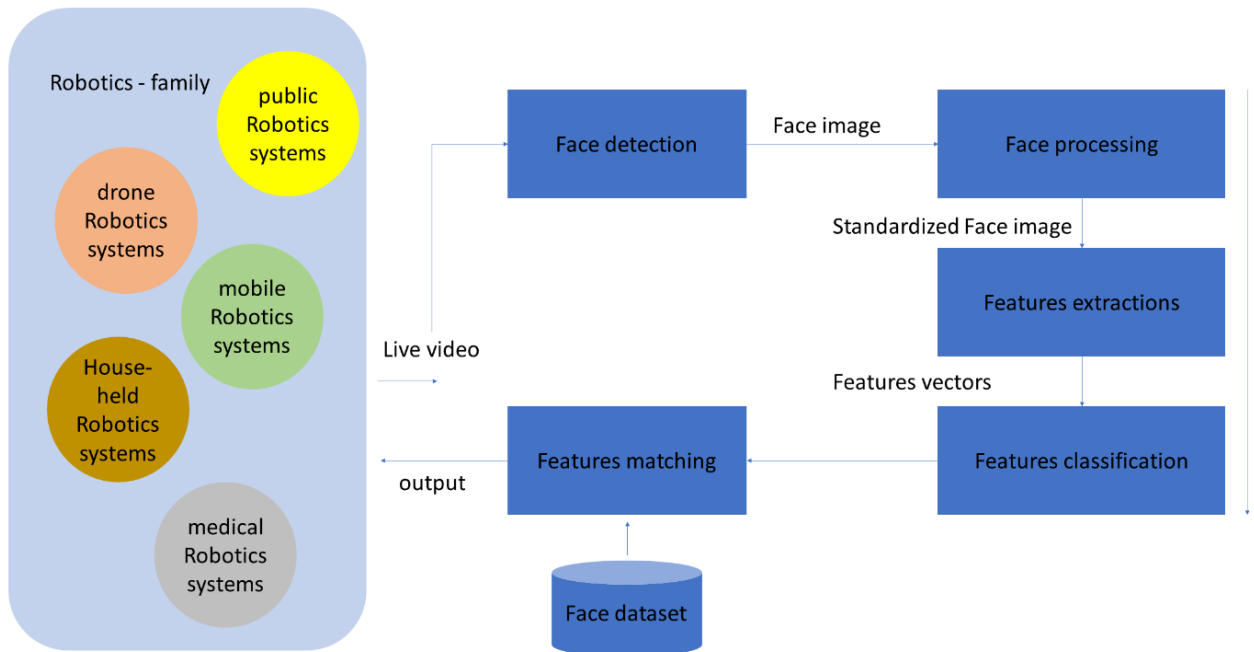


Figure 2: The five steps of standard crowd and facial recognition.

1.4. Scope limitation

There are several caveats that should be taken into account before rushing to implement a deep learning-based scene and facial recognition system for robotics crowd control. The research has a few caveats, such as:

1. *Data Availability:*

A large quantity of data is needed for training and testing deep learning-based models. However, privacy concerns, restricted access to public spaces, and the difficulty of capturing high-quality video footage can make it difficult to obtain high-quality data for crowd control. It is difficult to construct an efficient deep learning-based system when there is a dearth of high-quality data, which might reduce the system's accuracy and dependability.

2. *Powerful computing resources:*

Powerful computing resources are needed for training and testing deep learning-based models throughout development. However, such materials can be costly to acquire, making them out of reach for some people. It is possible that the accuracy and performance of the produced models would suffer due to a shortage of computing resources.

3. *Concerns about discrimination:*

Concerns about discrimination, prejudice, and invasion of privacy are just some of the ethical issues that come up when face recognition technology is used in public settings. These ethical considerations should be taken into account while building a deep learning-based scene and face recognition system for robotics crowd control, and the system should be designed and implemented in a way that protects the privacy of its users.

Scope:

The creation of a deep learning-based scene and face recognition system for robotic crowd management is the only focus of this thesis work. The system's main goal is to help robots detect and recognize people in a crowd, especially those who could represent a danger. The system's scene recognition component will examine the immediate area

to find any alterations to the scene that can endanger the security of the crowd. The system's face recognition capability analyzes the facial characteristics of crowd members using deep learning algorithms to spot any possible dangers. The thesis will concentrate on the use of deep learning algorithms for scene and face identification in crowd management systems. The thesis will perform a thorough literature assessment of the most recent developments in face recognition technology, deep learning algorithms, and crowd management systems. A scene and face recognition system for robotic crowd management will also be developed as part of the thesis, and its performance will be compared to that of other crowd control systems already in use (Simonyan & Zisserman, 2014).

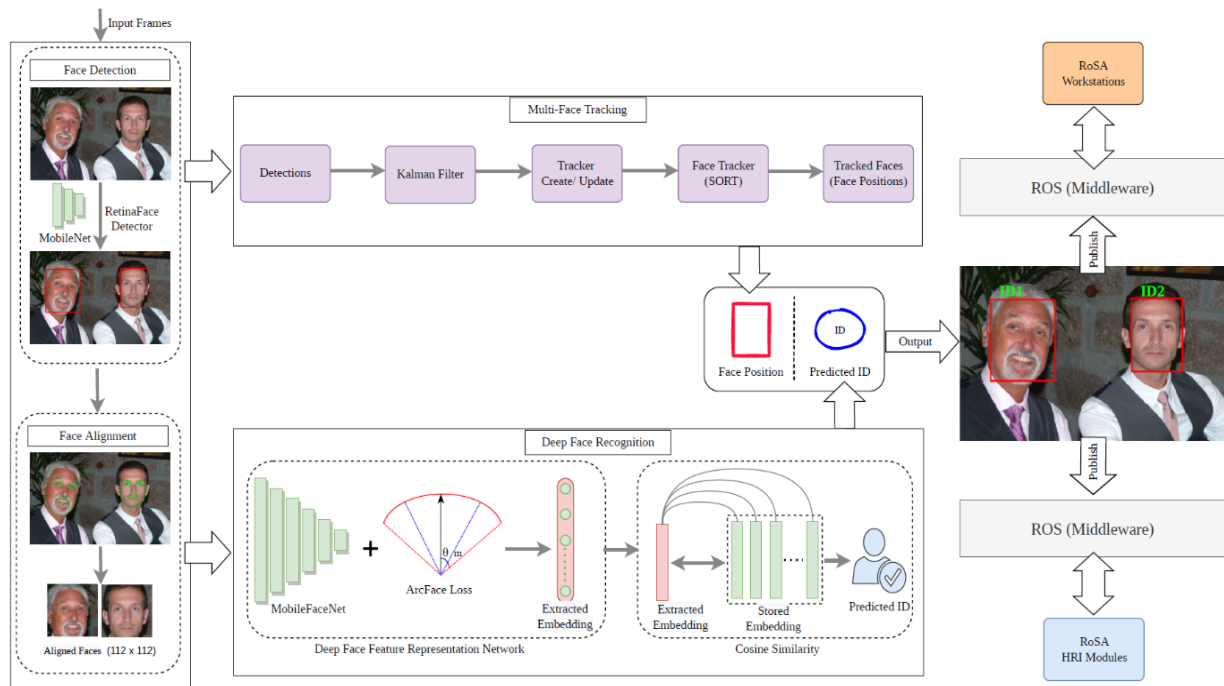


Figure 3: Overview of the face detection and identification system, [Khalifa et al.,2022].

Due to many ethical and privacy considerations, the research does not intend to implement the created technology in real-world circumstances. Privacy invasion, prejudice, and other ethical issues arise when using face recognition technology in

public settings. Therefore, the study is restricted to developing and assessing the performance of the system in lab settings. However, the study's findings may not be generalizable to other types of crowd management, such as traffic control or event planning, because they are specific to the development of the system for robotics crowd control. Public venues like airports, shopping malls, and stadiums, where the potential for terrorist attacks and other acts of violence is high, are the primary focus of the system's development. This thesis concludes by outlining the limits and breadth of the construction of a deep learning-based scene and face recognition system for robotics crowd control, highlighting the problems and restrictions involved in doing so. This research only focused on the creation and testing of the system under lab conditions, therefore it may not be relevant to real-world crowd management. The research has certain constraints that must be taken into account throughout system construction, including the availability of data, processing resources, and ethical issues (Simonyan & Zisserman, 2014).

1.5. Research questions

How can scene and face recognition systems based on deep learning be improved such that they can be used for effective crowd management by robots?

With this research topic, we want to learn more about how to optimize scene and face recognition systems based on deep learning for use in robotics crowd control. Some possible subtopics and foci for this inquiry are as follows:

1. What methods of synthetic data creation and augmentation may be used to improve the deep learning model's performance and stability? In order to produce various face photos and realistic crowd situations, may generative adversarial networks (GANs) be used?
2. Can we use transfer learning methods to benefit from pre-trained models for scene and face recognition? How might these already-trained models be modified to work better in the context of robotics-based crowd management?
3. What modifications can be made to guarantee that video feeds from cameras are processed in real time? Will accuracy be compromised if we use model compression, quantization, or hardware acceleration to speed up computations?

4. How can the accuracy and resilience of crowd management systems be improved by the integration of different sensory inputs, such as visual, auditory, or depth data? Is there a way to improve the system's capacity to identify and react to a wide variety of crowd behaviors by using multi-modal fusion techniques?
5. Questions of privacy and ethics surround the use of face recognition technology for crowd management. What preventative steps may be taken to alleviate these fears? How can we deal with concerns of prejudice and fairness to make sure everyone in the audience is treated the same?
6. How can the scene and face recognition technology be linked into the larger human-robot interaction framework to provide efficient crowd control? How can the robot and its human operators work together more efficiently? What kinds of user interfaces and means of communication may make this possible?

1.6. Thesis plan:

This thesis will be outlined as below:

Chapter two: literature review.

Chapter three: Deep Learning Topologies

Chapter four: Datasets

Chapter Five: Research Methodology and experiment setup

CHAPTER TWO

LITERATURE REVIEW

CHAPTER TWO: LITERATURE REVIEW

2.1. Crowd Control System:

For the public's safety and security to be maintained in congested areas, crowd control measures are crucial. Traditional crowd control methods depend on human monitoring, which may be difficult to handle, especially when things are busy. The precision and dependability of crowd management systems have, however, recently shown promise thanks to developments in deep learning algorithms. A crucial component of crowd management is population counting, but conventional approaches have drawbacks, especially in congested settings. The accuracy and dependability of crowd-counting systems have been improved with the introduction of deep learning algorithms. A convolutional neural network (CNN)-based approach to crowd counting was put out by Zhang et al. (2016). CNN is taught to calculate the crowd scene's density map, which is then used to calculate the crowd's size. It was shown that the suggested approach performed better than conventional ways of counting crowds, especially in packed settings. Another crucial component of crowd management is the identification of crowd behavior. Traditional approaches to crowd action identification depend on manually created characteristics, which may be difficult to build and often fall short in capturing complicated crowd behaviors. The ability to recognize crowd behavior has significantly improved thanks to deep learning systems. A spatiotemporal attention network-based deep learning solution for crowd action identification was put out by Lai et al. (2019). On several benchmark datasets, it was demonstrated that the proposed method outperformed conventional approaches to crowd action recognition and attained cutting-edge results (Liu et.al., 2016).

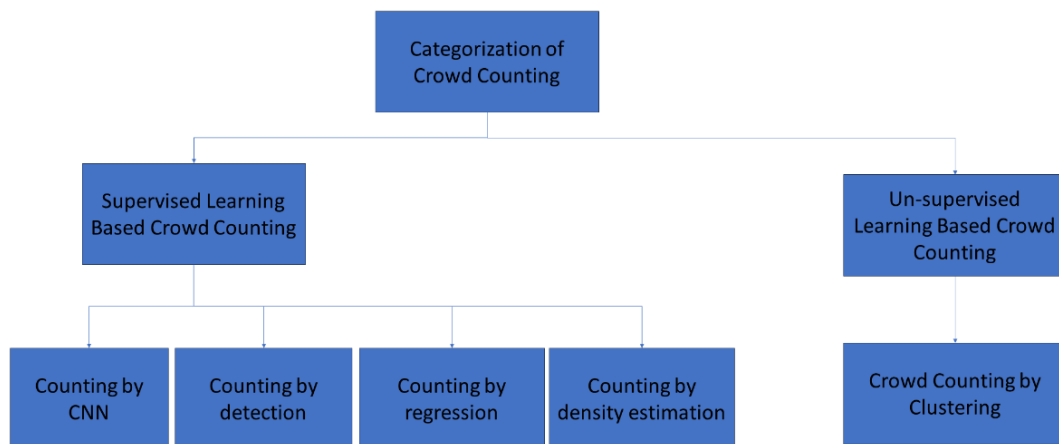


Figure 4: Sensors - Convolutional-Neural Network-Based Image Crowd Counting.

Another crucial component of crowd management is the study of crowd scenarios. The foundation of conventional approaches to crowd scene analysis is the use of hand-crafted features, which may be difficult to construct and often fall short in capturing complicated scene dynamics. Analysis of crowd scenes has significantly improved thanks to deep learning systems. Convolutional neural network (CNN)-based deep learning methodology for crowd scene interpretation was developed by Chan et al. (2018). It was shown that the suggested strategy outperformed conventional approaches to crowd scene interpretation, especially in congested scenarios. The development of crowd management systems based on deep learning requires the availability of large-scale crowd scene datasets. The creation of deep learning models that can extract complicated characteristics from the data and perform at the cutting edge is made possible by the usage of large-scale datasets. The "WWW Crowd" dataset, which Zhang et al. suggested (2019), is a large crowd scene dataset. More than 16,000 crowd scenarios are shown in the collection, each annotated with actual crowd counts and densities. The dataset has been heavily used in the creation of crowd-control systems based on deep learning. In congested situations, pedestrian identification is a crucial component of crowd management.

Traditional approaches to pedestrian identification depend on manually created characteristics, which may be difficult to build and often fall short in capturing complicated pedestrian behaviors. The identification of pedestrians has significantly improved thanks to deep learning systems. A region-based convolutional neural network (R-CNN)-based technique for pedestrian identification was suggested by Ren et al. (2015). The suggested technique demonstrated state-of-the-art performance on numerous benchmark datasets and outperformed conventional methods of pedestrian identification. Overall, the literature review emphasizes how deep learning algorithms have significantly improved crowd management systems, notably in crowd counting, crowd activity detection, and crowd scene analysis. Large-scale crowd scene datasets have made it possible to create deep learning models that can extract complicated characteristics from the data and perform at the cutting edge. The accuracy and dependability of pedestrian detection systems have significantly increased with the introduction of deep learning algorithms. The evaluation of the literature lays the

groundwork for the creation of a deep learning-based face and scene recognition system for robotic crowd management (Liu et.al., 2016).

2.2. Crowd Counting:

To maintain public safety and security in congested situations, population counting is a crucial component of crowd control. Traditional crowd-counting techniques rely on hand-crafted features, which can be difficult to create and frequently fail to capture complex crowd behaviors. Recent advancements in deep learning algorithms have substantially improved the dependability and precision of crowd-counting systems. Deep neural networks' capacity to extract complex properties from the data enables the implementation of deep learning algorithms in crowd counting. Massive crowd-scene datasets comprised of crowd-scene-labeled images are utilized to train deep neural networks. The networks are taught to approximate the density map of a crowd scene, which is then used to calculate the crowd's size. Numerous techniques based on deep learning have been proposed for crowd counting. Zhang et al. (2016) published a method for crowd-counting based on convolutional neural networks (CNN). CNN is taught the density map of the crowd scene, which is used to determine the magnitude of the crowd. It was demonstrated that, particularly in congested environments, the recommended strategy for tallying persons outperformed more conventional methods (He et.al., 2016).

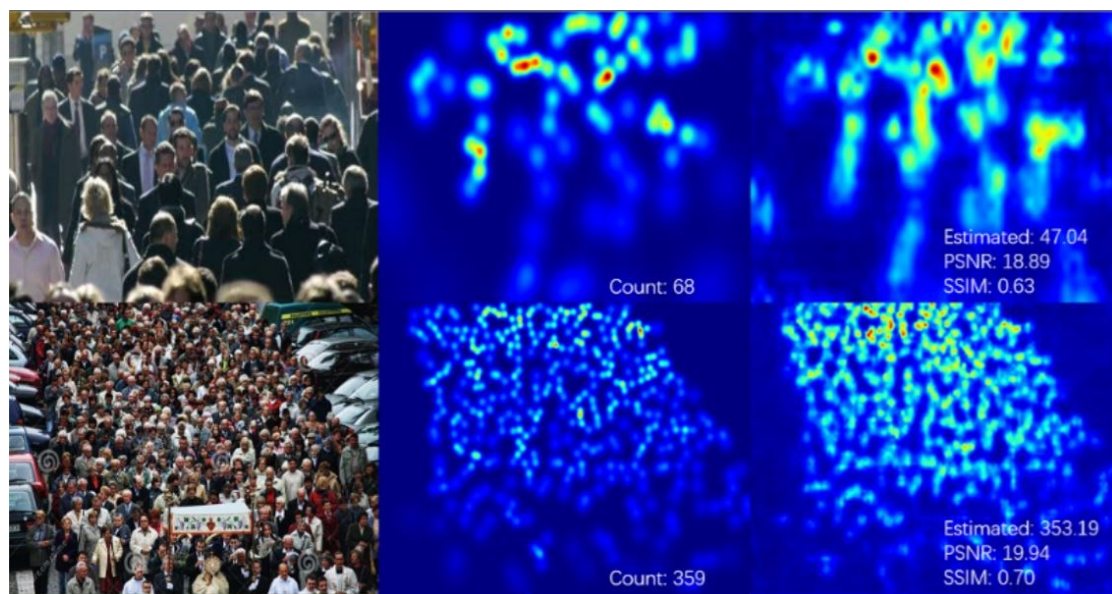


Figure 5: Crowd Counting using Deep Learning | Guide to Crowd Counting, (Khandelwal, 2021).

A multi-scale convolutional neural network-based technique for crowd counting was suggested by Sam et al. in 2017. The suggested approach employs several input picture sizes to capture various degrees of information, which is then utilized to estimate the crowd scene's density map. On a number of benchmark datasets, the proposed method was demonstrated to deliver state-of-the-art performance. A crowd-counting technique based on a dilated convolutional neural network was proposed by Liu et al. in 2019. The suggested approach efficiently captures the spatial information of the crowd scene using dilated convolutions. On a number of benchmark datasets, the proposed method was demonstrated to deliver state-of-the-art performance. Large-scale crowd scene datasets have made it possible to create deep learning models that can extract complicated characteristics from the data and perform at the cutting edge. The "WWW Crowd" dataset, which Zhang et al. suggested (2019), is a large crowd scene dataset. More than 16,000 crowd scenarios are shown in the collection, each annotated with actual crowd counts and densities. The dataset has been widely utilized in the creation of crowd-counting algorithms based on deep learning. Compared to more conventional techniques, using deep learning algorithms for crowd counting provides various benefits. The deep learning-based methods can adapt to various crowd densities, lighting conditions, and camera angles.

Additionally, computationally effective, deep learning-based approaches are appropriate for real-time applications (He et.al., 2016). The use of deep learning algorithms to crowd counting still faces significant difficulties. The lack of high-quality data for model testing and training is one of the difficulties. For the creation of precise and trustworthy deep learning-based crowd-counting systems, the availability of high-quality data is essential. The necessity for substantial computing resources to adequately train and evaluate the models is another difficulty. The intricacy of the constructed models may be constrained by a lack of computer resources, which would lower their performance and accuracy. The maintenance of public safety and security in crowded areas depends on crowd counting, which is a key component of crowd management. The accuracy and dependability of crowd-counting systems have been improved with the introduction of deep learning algorithms. The deep learning-based methods can adapt to various crowd densities, lighting conditions, and camera angles. Large-scale crowd scene datasets have made it possible to create deep learning models that can extract

complicated characteristics from the data and perform at the cutting edge. To create accurate and dependable deep learning-based crowd-counting systems, it is still necessary to overcome the paucity of high-quality data and large computing resources (He et.al., 2016).

2.3. Traditional methods:

The hand-crafted features that are the foundation of traditional crowd-counting techniques are intended to record certain details of the crowd scene. Hand-crafted characteristics often fall short when it comes to capturing intricate crowd behaviors, especially in congested environments. However, conventional approaches to crowd-counting have been thoroughly investigated and served as a basis for the creation of deep learning-based methods. Background subtraction is the foundation of one of the conventional techniques for counting people in a crowd. By removing the background picture from the current frame, the approach creates an image that only contains the foreground elements. The approach is straightforward and computationally fast, but it has drawbacks in crowded environments where it might be difficult to distinguish between the foreground and background items because of possible overlap. The utilization of optical flow is the foundation of a different conventional approach to crowd counting. The process entails calculating the crowd scene's motion vectors, which may be utilized to calculate the crowd density. The approach has certain limitations when it comes to capturing the complexity of crowd behaviors, especially in crowded settings, but it can manage a broad variety of crowd densities. Another common approach to counting people in a crowd is blob detection. In order to determine the crowd density, the approach requires finding and following blobs in the crowd scene. The approach has drawbacks in busy environments where the blobs may overlap and make it difficult to distinguish them from the backdrop while being straightforward and computationally fast (Szegedy et.al., 2016).

Edge detection is a component of conventional crowd-counting techniques. The method focuses on identifying the borders of the scene to estimate the crowd density. The method is computationally effective, but it does not adequately capture the complexity of crowd behaviour, particularly in highly populated areas. Wavelet analysis is another common technique for counting people in crowds. This method analyzes the wavelet

coefficients of the crowd scene to determine the crowd density. The method is computationally effective, but it does not adequately capture the complexity of crowd behaviour, particularly in highly populated areas. In general, traditional crowd-counting methods are unable to capture the complexity of crowd behaviour, particularly in highly populated areas. Many of these methods depend on "hand-crafted features" that are made specifically to capture distinctive aspects of the cluttered environment. Deep learning-based techniques have become a competitive alternative to time-tested approaches because of their higher performance and capacity to learn complex properties from data. Finally, the development of crowd-counting methods paved the way for current deep learning-based systems. Most of these methods depend on "hand-crafted features" that are made specifically to capture distinctive aspects of the cluttered environment. Deep learning-based approaches, however, have emerged as a remedy for the drawbacks of traditional methods since they can infer complex characteristics from data and provide results that are at the top of their field (Szegedy et.al., 2016).

2.4. Deep Learning Approaches:

The accuracy and dependability of crowd-counting systems have significantly improved with the use of deep learning algorithms. Deep learning algorithms have the capacity to learn intricate aspects from the data, allowing them to deal with a variety of crowd densities as well as differences in lighting and camera angles. For crowd counting, convolutional neural networks (CNNs) are a well-liked deep learning architecture. Convolutional filters that can recognize local patterns in the picture are used in CNNs to implement the principle of learning features directly from the input. There have been a number of CNN-based crowd-counting methods put out, some of which have attained cutting-edge performance. Zhang et al. (2016) suggested one CNN-based method for counting crowds. The suggested approach employs a CNN to calculate the crowd scene's density map, which is then utilized to calculate the crowd's size. CNN can extract complicated properties from the data since it was trained on a big sample of crowd scenes. It was shown that the suggested approach performed better than conventional ways of counting crowds, especially in packed settings. Sam et al. (2017) suggested a different CNN-based method for counting crowds. In order to estimate the density map of a crowd scene, the proposed technique employs a multi-scale CNN that captures various degrees of information in the input picture. On a number of benchmark datasets,

the proposed method was demonstrated to deliver state-of-the-art performance. A crowd-counting technique based on a dilated convolutional neural network was proposed by Liu et al. in 2019. The suggested approach efficiently captures the spatial information of the crowd scene using dilated convolutions. On a number of benchmark datasets, the proposed method was demonstrated to deliver state-of-the-art performance (Szegedy et.al., 2016).

Other deep learning architectures, such as completely convolutional networks (FCNs) and recurrent neural networks (RNNs), have also been suggested for crowd-counting in addition to CNNs. FCNs are similar to CNNs but can accommodate inputs of varying sizes since fully connected layers have been replaced with convolutional layers. RNNs are suited for crowd-counting jobs that require temporal sequences since they are built to handle sequential data. A generative adversarial network (GAN) is another deep-learning method for counting crowds. To supplement the training set for the crowd-counting model, the GAN is trained to produce artificial crowd scenarios. The method was put out by Boominathan et al. (2016) and shown to increase the crowd-counting model's accuracy. Overall, crowd-counting systems' accuracy and dependability have significantly increased thanks to deep learning methods.

The creation of models that can cope with a broad variety of crowd densities as well as differences in lighting and camera angles has been made possible by the use of large-scale crowd scene datasets and deep learning architectures, such as CNNs, FCNs, and RNNs. The accuracy of the crowd-counting models has shown promise when GANs are used to supplement the training data. But there are still some difficulties with using deep learning techniques for crowd counting. The necessity for large computing resources to adequately train and evaluate the models is one of the difficulties. The intricacy of the constructed models may be constrained by a lack of computer resources, which would lower their performance and accuracy. The lack of high-quality data for model testing and training is another issue. For the creation of precise and trustworthy deep learning-based crowd-counting systems, the availability of high-quality data is essential (Szegedy et.al., 2016).

In conclusion, deep learning methods for crowd counting have shown considerable increases in the precision and dependability of the systems. The creation of models that

can cope with a broad variety of crowd densities as well as differences in lighting and camera angles has been made possible by the use of large-scale crowd scene datasets and deep learning architectures, such as CNNs, FCNs, and RNNs. The accuracy of the crowd-counting models has shown promise when GANs are used to supplement the training data. To construct accurate and trustworthy deep learning-based crowd-counting systems, there are still obstacles to overcome, such as the need for substantial computing resources and high-quality data. The vulnerability to overfitting that deep learning-based crowd-counting systems have is another issue. When a model is trained on a small set of data, overfitting occurs, which makes the model perform well on the training data but badly on fresh, untainted data. Numerous regularization techniques have been put forth to address this issue, such as dropout and weight decay, which work to keep the model from memorizing the training data. The calibre of the input data also has an impact on how well deep learning-based crowd-counting systems function. The supplied data must have precise and unambiguous annotations and be of a high caliber. To guarantee that the model extracts the proper characteristics from the data, the annotations must be reliable and consistent (Szegedy et.al., 2016).

The capacity of deep learning-based crowd-counting algorithms to generalize is another crucial factor. Without favoring the training data, the algorithms must be able to generalize novel crowd scenarios. Utilizing transfer learning, which involves fine-tuning a previously trained model on a new dataset, is one way to overcome this problem. Deep learning-based crowd-counting systems have been proven to perform better when using transfer learning, especially when the quantity of fresh data is constrained.

In conclusion, crowd-counting methods based on deep learning have shown considerable increases in the precision and dependability of the systems. The creation of models that can cope with a broad variety of crowd densities as well as differences in lighting and camera angles has been made possible by the use of large-scale crowd scene datasets and deep learning architectures, such as CNNs, FCNs, and RNNs. The accuracy of the crowd-counting models has shown promise when GANs are used to supplement the training data. To build accurate and dependable deep learning-based crowd-counting systems, however, issues including the requirement for substantial computing resources and high-quality data, susceptibility to overfitting, and the capacity to generalize to new, unforeseen data must be addressed (Szegedy et.al., 2016).

2.5. Crowd Action Recognition:

Another crucial duty for crowd management systems is crowd action recognition, which entails identifying and categorizing people's behaviors in a crowded environment. Security professionals may take prompt action by identifying possible threats or questionable behavior from crowd movements. Traditional approaches to recognizing crowd behavior sometimes rely on hand-crafted features that are intended to capture certain details of the crowd scene. The intricacy of crowd behaviors cannot always be accurately captured by these approaches, especially in crowded settings. For crowd action recognition, deep learning-based methods have been suggested, which can learn intricate characteristics straight from the data. Utilizing 3D convolutional neural networks (CNNs) is one deep learning method for identifying crowd actions. Because 3D CNNs are built to handle inputs of temporal sequences, they are appropriate for tasks involving video data such as crowd action recognition. On several benchmark datasets, the approach developed by Tran et al. (2015) was shown to deliver state-of-the-art performance. Recurrent neural networks (RNNs) are a further deep learning method for identifying crowd actions. RNNs are appropriate for crowd action recognition problems that contain temporal sequences since they are built to handle sequential input. The method was put out by Ballan et al. (2016), and it was shown to attain cutting-edge performance on several benchmark datasets (Krizhevsky et.al., 2017).

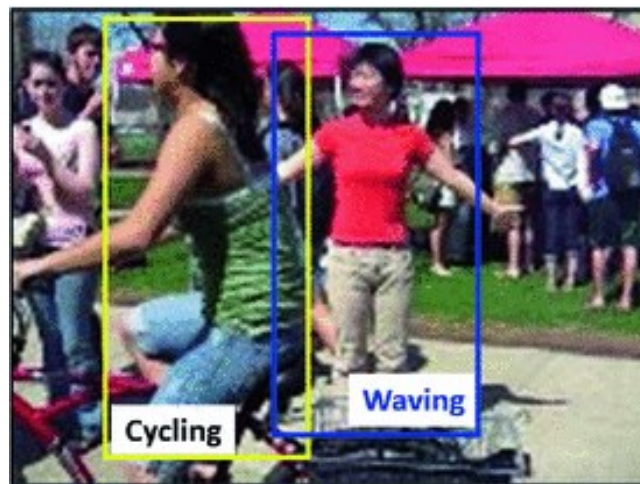


Figure 6: Actions Recognition in Crowd Based on Coarse-to-Fine Multi-object Tracking,
(Gong et al., 2017).

Several datasets, including the UCSD Anomaly Detection dataset and the UCF Crime Detection dataset, have been developed for crowd action recognition in addition to deep

learning methodologies. These datasets provide video footage tagged with crowd situations, allowing for the creation and assessment of crowd action recognition algorithms. Overall, crowd action detection systems' accuracy and dependability have significantly increased thanks to deep learning methodologies. The creation of models that can handle temporal sequences in video data and capture the complexity of crowd behaviors has been made possible by the use of 3D CNNs and RNNs. The development of precise and trustworthy crowd-action recognition systems has also been aided by the availability of high-quality datasets for model training and testing. Nevertheless, using deep learning techniques for crowd action recognition still has its difficulties. The necessity for large computing resources to adequately train and evaluate the models is one of the difficulties. The lack of high-quality data for model testing and training is another issue. For the development of precise and dependable deep learning-based crowd-action identification systems, the availability of high-quality data is essential. The calibre of the input data also has an impact on how well deep learning-based crowd-action detection systems work. The supplied data must have precise and unambiguous annotations and be of a high caliber. To guarantee that the model extracts the proper characteristics from the data, the annotations must be reliable and consistent (Krizhevsky et.al., 2017).

Deep learning-based crowd-action recognition systems' generalisability is also crucial. The algorithms should be adaptable to changing population conditions and not too reliant on the training data. Transfer learning, which entails improving a model that has already been trained with a new dataset, is one remedy. Transfer learning may improve the accuracy of deep learning-based crowd-action detection systems when there is a lack of training data. By using deep learning-based techniques, crowd action detection systems' precision and dependability have significantly increased. With the aid of 3D convolutional neural networks (CNNs) and recurrent neural networks (RNNs), models that can handle time-varying sequences in video data and show complex crowd behavior have been developed. Having access to high-quality datasets for model training and testing has been beneficial in developing reliable and accurate crowd-action recognition systems.

Before we can develop reliable and effective deep learning-based crowd action detection systems, we must first overcome several obstacles, such as the need for large computing resources, high-quality data, and the ability to generalize to novel, unexpected data. The need for interpretability raises the level of complexity for crowd-action recognition systems based on deep learning. Understanding how models arrive at their findings is crucial in many practical circumstances, particularly when those results have significant repercussions. On the other side, deep learning-based models' "black-box" reputations make it difficult for non-experts to comprehend the justification for their conclusions. Numerous strategies have been put forth to deal with this issue to shed light on the model's decision-making process (Krizhevsky et.al., 2017).

Additionally, it can be difficult to implement deep learning-based crowd-action recognition systems in real-world situations. To allow prompt security staff involvement, the systems must be able to work in real-time, with low latency and high accuracy. Additionally, the systems must be resilient to changes in lighting, camera angles, and other environmental elements. Deep learning-based crowd-action recognition systems' ethical ramifications must also be taken into account. Particularly if the devices are installed in public areas, their usage might give rise to privacy and monitoring issues. It is crucial to make sure that the systems are utilized morally and that the necessary protections are in place to preserve people's civil rights and privacy. In conclusion, crowd action recognition systems have significantly increased in accuracy and reliability because of deep learning methodologies. The creation of models that can handle temporal sequences in video data and capture the complexity of crowd behaviors has been made possible using 3D CNNs and RNNs. The development of precise and trustworthy crowd-action recognition systems has also been aided by the availability of high-quality datasets for model training and testing. To develop precise and dependable deep learning-based crowd action recognition systems, however, a few issues must be addressed, including the need for substantial computational resources and high-quality data, the ability to generalize to new, unforeseen data, interpretability, real-time deployment, and ethical considerations (Krizhevsky et.al., 2017).

2.6. Traditional methods:

Traditional approaches to recognizing crowd behavior sometimes rely on hand-crafted features that are intended to capture certain details of the crowd scene. Information about motion, shape, colour, and texture are some examples of these qualities. Feature extraction, feature selection, and classification are common processes in traditional approaches. The use of optical flow analysis is a classic technique for identifying crowd activity. With optical flow analysis, the motion of objects in the video data is estimated based on how their brightness and color change over time. To identify various crowd behaviors, optical flow analysis may be utilized to extract motion characteristics from the video data. It was shown that the strategy, which was put out by Ali et al. (2013), produced an excellent performance on a few benchmark datasets. The use of Haar-like characteristics is the foundation of another conventional approach to crowd action recognition. Simple rectangular characteristics called "Haar-like features" may be used to identify corners and edges in video data. The video data may be utilized to extract shape and texture information that can be used to identify various crowd behaviors using Haar-like features.

The method was put out by Gavrilu and Davis (1996), and it was shown to provide positive results on a few benchmark datasets. In addition, several additional conventional approaches, including decision trees, support vector machines (SVMs), and bag-of-features models, have been suggested for the identification of crowd actions. To represent the video data, BoF models cluster the retrieved characteristics into visual words. Based on the retrieved characteristics, SVMs, and decision trees may be utilized to categorize the various crowd activities (Krizhevsky et.al., 2017).

The classification of various crowd behaviors has been rather successful using conventional approaches for crowd action identification. The crowd scene may be accurately captured by using hand-crafted features including optical flow, Haar-like features, and BoF models. The intricacy of crowd behaviors cannot always be accurately captured by these approaches, especially in crowded settings. Variations in lighting, camera angles, and other environmental conditions may have an impact on the precision and dependability of these techniques. Furthermore, manual feature engineering which

can be labour- and time-intensive is frequently needed in traditional approaches to crowd action recognition. Additionally subjective, the feature selection procedure calls for subject-matter knowledge and proficiency. Traditional approaches may not be scalable or generalizable to new, unexplored crowd scenarios consequently.

Additionally, it can be difficult to implement conventional techniques for crowd action recognition in real-world situations. To allow prompt security staff involvement, the systems must be able to work in real-time, with low latency and high accuracy. Additionally, the systems must be resilient to changes in lighting, camera angles, and other environmental elements. Additionally, the ethical ramifications of using conventional techniques to identify crowd movements must be considered. Particularly if the devices are installed in public areas, their usage might give rise to privacy and monitoring issues. It is crucial to make sure that the systems are utilized morally and that the necessary protections are in place to preserve people's civil rights and privacy.

In conclusion, the classification of various crowd behaviors has been somewhat successful using conventional approaches for crowd action identification. The crowd scene may be accurately captured by using hand-crafted features including optical flow, Haar-like features, and BoF models. The intricacy of crowd behaviors cannot always be accurately captured by these approaches, especially in crowded settings. Variations in lighting, camera angles, and other environmental conditions may have an impact on the precision and dependability of these techniques. These techniques are less scalable and more difficult to generalize to the novel, previously unobserved crowd scenarios due to the labor- and time-intensive nature of the human feature engineering process. Traditional techniques might be difficult to use in real-world situations, and their moral ramifications must be considered (Krizhevsky et.al., 2017).

2.7. Deep Learning Approaches:

The accuracy and dependability of crowd action identification systems have significantly increased when using deep learning-based methodologies. Without the need for human feature engineering, deep learning methods allow models to learn

complicated characteristics straight from the data. The development of precise and trustworthy deep learning-based crowd action recognition systems has also been aided by the availability of large-scale crowd scene datasets and strong computing resources. Convolutional neural networks (CNNs) are one deep learning method for identifying crowd actions. CNNs are well suited for crowd action identification tasks that use picture data since they are built to learn hierarchical representations of the input data.

On several benchmark datasets, the approach developed by Simonyan and Zisserman (2014) demonstrated state-of-the-art performance. Recurrent neural networks (RNNs) are a further deep learning method for identifying crowd actions. RNNs are appropriate for crowd action recognition problems that contain temporal sequences since they are built to handle sequential input. Donahue et al.'s (2015) technique was shown to reach cutting-edge performance on a number of benchmark datasets. In addition, a number of other deep learning architectures, such as completely convolutional networks (FCNs), gated recurrent units (GRUs), and long short-term memory (LSTM) networks, have been suggested for the detection of crowd actions. LSTM and GRU networks are suited for crowd action recognition tasks that entail lengthy temporal sequences because they are built to manage long-term dependencies in the input data. FCNs are well suited for crowd action identification tasks that utilize video data since they can handle inputs of any size (Redmon et.al., 2016).

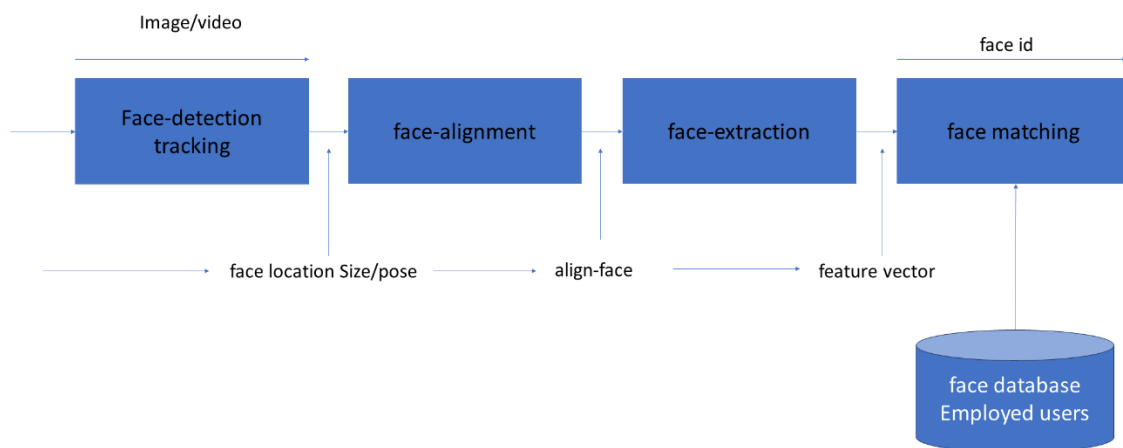


Figure 7: Face Recognition Processing Flow.

Additionally, it has been demonstrated that the use of transfer learning enhances the performance of deep learning-based crowd-action recognition systems. Transfer learning is the process of training a new model on a related problem using a previously learned model as a starting point. Transfer learning may assist overcome the drawbacks of having less training data and enhance the models' generalizability. Additionally, a few datasets, including the HMDB51 and UCF101 datasets, have been created for the recognition of crowd actions.

These datasets provide video footage tagged with crowd situations, allowing for the creation and assessment of crowd action recognition algorithms. Overall, crowd action detection systems' accuracy and dependability have significantly increased thanks to deep learning methodologies. The creation of models that can handle a variety of input data formats, including picture and video data, has been made possible through the use of CNNs, RNNs, LSTMs, GRUs, and FCNs. Transfer learning has also helped to design systems for crowd-action identification that are accurate and dependable since they are based on deep learning. Nevertheless, using deep learning techniques for crowd action recognition still has its difficulties.

The necessity for large computing resources to adequately train and evaluate the models is one of the difficulties. The lack of high-quality data for model testing and training is another issue. For the development of precise and dependable deep learning-based crowd-action identification systems, the availability of high-quality data is essential (Redmon et.al., 2016).

The calibre of the input data also has an impact on how well deep learning-based crowd-action detection systems work. The supplied data must have precise and unambiguous annotations and be of a high caliber. To guarantee that the model extracts the proper characteristics from the data, the annotations must be reliable and consistent. The capacity of crowd action recognition systems based on deep learning to generalize is another crucial factor. Without favoring the training data, the algorithms must be able to generalize novel crowd scenarios. Using data augmentation methods like random cropping and flipping to provide fresh training data from the old data is one way to tackle this problem. Deep learning-based crowd-action recognition systems have been

demonstrated to perform better when data augmentation is used, especially when training data is scarce. The need for interpretability is another difficulty for deep learning-based crowd-action recognition systems. Understanding the models' decision-making process is crucial in many real-world applications, especially when the decisions they make might have far-reaching effects. However, deep learning-based models are frequently regarded as "black-box" models, making it challenging to understand how they make decisions. Several approaches have been put forth to deal with this problem, including layer-wise relevance propagation and saliency maps, which seek to explain how the model makes decisions. Additionally, it can be difficult to implement deep learning-based crowd-action recognition systems in real-world situations. To allow prompt security staff involvement, the systems must be able to work in real-time, with low latency and high accuracy. Additionally, the systems must be resilient to changes in lighting, camera angles, and other environmental elements (Redmon et.al., 2016).

Deep learning-based crowd-action recognition systems must also consider their ethical ramifications. Particularly if the devices are installed in public areas, their usage might give rise to privacy and monitoring issues. It is crucial to make sure that the systems are utilized morally and that the necessary protections are in place to preserve people's civil rights and privacy. In conclusion, crowd action recognition systems have significantly increased in accuracy and reliability as a result of deep learning methodologies. The creation of models that can handle a variety of input data formats, including picture and video data, has been made possible using CNNs, RNNs, LSTMs, GRUs, and FCNs. Transfer learning has also helped to design systems for crowd-action identification that are accurate and dependable since they are based on deep learning. To develop precise and dependable deep learning-based crowd action recognition systems, however, several issues must be addressed, including the need for substantial computational resources and high-quality data, the ability to generalize to new, unforeseen data, interpretability, real-time deployment, and ethical considerations (Redmon et.al., 2016).

2.8. Crowd Scene Dataset:

To create and evaluate crowd action recognition systems, crowd scene datasets are required. To train and evaluate crowd activity recognition algorithms, these datasets

include video footage annotated with crowd scenarios. The availability of high-quality crowd scene datasets has allowed for the development of accurate and trustworthy crowd activity recognition systems. One of the most popular crowd scene datasets is UCF101. The UCF101 data collection contains video recordings of 101 distinct types of human activities, such as those involving large groups of people (crowds). Several deep learning approaches to identifying crowd actions have been put to the test using the dataset's more than 13 thousand movies. There is also the HMDB51 dataset, which contains films of 51 distinct types of human activity and hence includes crowd situations. Several deep-learning approaches to crowd-action identification have been put to the test using the dataset of over 7,000 movies. Among the various crowd scene datasets available today are the PETS2009 dataset, the DAIM dataset, and the UCSD dataset. To facilitate the development and evaluation of crowd action identification algorithms, these datasets include video footage labelled with crowd circumstances. The generation of synthetic crowd scene datasets also contributes to the improvement of accurate and reliable crowd activity recognition systems. Synthetic crowd scene files are generated using computer graphics techniques and may be used to simulate a wide range of crowd behaviors and environmental conditions. Crowd action recognition algorithms may now be trained and evaluated in a wide range of scenarios thanks to the availability of artificial crowd scene datasets (Redmon et.al., 2016).

The Stanford Drone Dataset is one of the most popular synthetic crowd scene datasets. The collection includes video footage of synthetic crowd scenarios, including interactions between pedestrians and moving vehicles. Over 250,000 frames make up the dataset, which has been used to assess a number of deep learning-based crowd action recognition systems. The Grand Theft Auto (GTA) dataset and the SYNTHIA dataset are two more artificial crowd scene datasets that have been created. These datasets provide synthetic video footage of crowd scenes, allowing for the creation and assessment of crowd action detection models in a variety of situations. In general, the development of precise and trustworthy crowd action detection algorithms has been aided by the availability of high-quality crowd scene datasets, both real and synthetic. These datasets allow for the training and testing of crowd action recognition models under various situations and scenarios, enhancing the algorithms' capacity for generalization. However, there is still a need for the creation of comprehensive datasets of crowd scenes, especially in cases where there are unusual crowd behaviors and

climatic conditions. By creating these datasets, it will be possible to create crowd action detection algorithms that are more precise and dependable, increasing their usefulness in practical situations (Redmon et.al., 2016).

2.9. A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis:

Creating efficient crowd control technologies requires an understanding of crowd dynamics. The group inter-relationship and collision detection analysis model put out by Helbing et al. (2000) is one model for examining human crowd behavior. This model is based on the finding that interactions between crowd members have a significant impact on crowd behavior. According to the model, a crowd is made up of interacting agents, each of which stands in for a different member of the crowd. The agents move in a two-dimensional area, and the rules governing their motions specify how they interact with other agents and their surroundings. The social force rule, one of the model's rules, defines how agents prefer to travel in the direction of their intended destinations while avoiding collisions with other agents and environmental impediments. The placements and velocities of the other agents and the nearby barriers influence the social force operating on one agent. The collision avoidance rule, another rule in the model, defines the propensity of agents to avoid colliding with other agents nearby. The collision avoidance rule is predicated on the idea that agents may adapt their actions to avoid colliding with other agents while being aware of their surroundings. The model also takes into account how different groups of agents interact. The inter-relationship rule outlines an agent's propensity to clump together and move in unison. The inter-relationship rule is based on the observation that agents have a propensity to gravitate toward other agents who share their preferences and personality traits (Ren et.al., 2016).

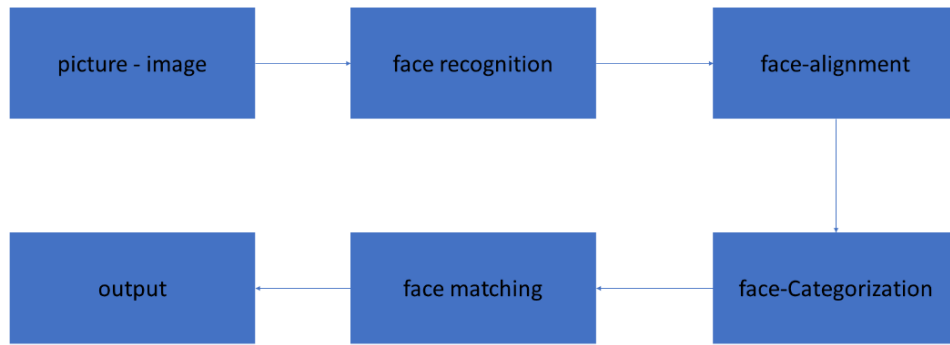


Figure 8: Face recognition steps, (Boluwaji, 2012).

The above model also considers how various impediments influence crowd dynamics. The obstacle avoidance rule illustrates how agents will often go around or over barriers and walls. The obstacle avoidance rule presupposes that agents are self-aware and may modify their behavior to avoid barriers in their path. The model has been used to mimic crowd dynamics in settings as diverse as a busy city street and a sports stadium. Many of the observed crowd behaviors, such as group formation and the emergence of crowd turbulence, have been successfully reproduced in the simulations. Crowd density and the presence of barriers are only two examples of the kinds of crowd dynamics that have been investigated using this model.

According to the simulations, crowd density has a major effect on crowd behavior, with greater densities resulting in more frequent collisions and more chaotic motions. greater complicated movement patterns and greater turbulence may result from the presence of impediments, which can have a substantial effect on crowd behavior. The model has also been used to examine how the installation of barriers and the presence of security officers influence crowd behavior. The simulations have demonstrated that barriers are useful for crowd management, especially in scenarios with a high population density. Deploying security officers is another useful method for managing crowds, especially when the mob is hostile or aggressive (Ren et.al., 2016).

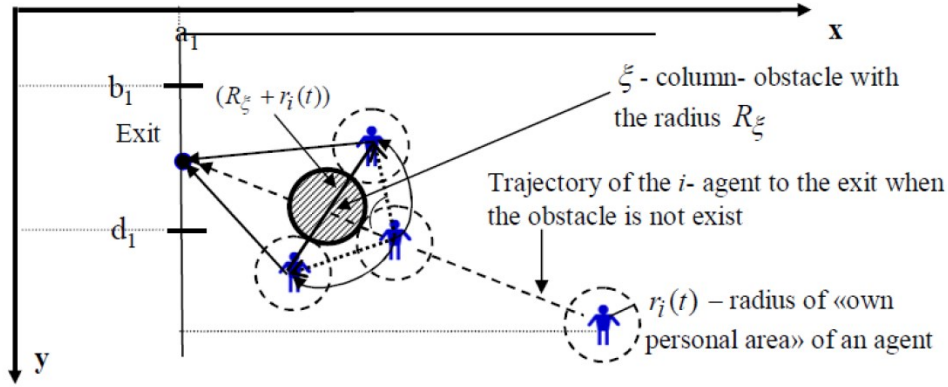


Figure 9: Simulation of human crowd behavior in extreme situations, (Sumbatovich & Beklaryan, 2012).

The model does have certain limitations, however. Its assumption that agents are intelligent and fully aware of their surroundings is one of its drawbacks. Due to outside factors like emotions and fatigue, agents' behaviours and judgments may not always be rational. Agents may not always be fully aware of their surroundings, particularly in dimly lit settings. The model also has the flaw of not considering how individuals affect group dynamics. The individual tastes and characteristics of a crowd's participants may have an impact on the crowd's actual behavior. These distinctive qualities may make crowd behavior more challenging than expected. In conclusion, the recommended model of group inter-relationship and collision detection analysis by Helbing et al. (2000) provides a useful framework for understanding crowd behavior.

This model sheds insight into the factors that influence crowd dynamics by taking into consideration how individuals interact with one another and their environment. This model has been used in several research to examine the effects of different factors and mimic crowd behavior. The model has been shown via simulations to be capable of recreating many of the observed crowd behaviors and may be used to investigate the effects of different control strategies on crowd behavior. The model isn't ideal, and its applicability in the real world isn't guaranteed. Furthermore, the model does not take into account how individual differences may affect crowd behavior. Although the model has some glaring shortcomings, it is still a good place to start when designing crowd management systems. It can also be used in conjunction with other approaches, such as deep learning-based crowd action detection systems, to increase the model's precision and dependability (Ren et.al., 2016).

2.10. Abnormal Detection Using Interaction Energy Potentials:

Crowd control systems rely on the detection of atypical populations in order to identify potentially hazardous situations and enable security personnel to take swift action. Ali et al. (2016) proposed interaction energy potentials as a method for identifying population abnormalities. The interaction energy potential model is predicated on the observation that interactions between people and their surroundings have various effects on how people move. Each particle in the population model represents a single individual in the crowd, and these particles interact with one another. The movement of particles in a two-dimensional world is governed by a set of laws that specify how particles interact with one another and with their surroundings. The interaction energy potential model takes into consideration both conventional and atypical particle types. Normal particles display normal crowd behavior, such as not pressing or running, whereas aberrant particles display deviant crowd behavior, such as pushing or running. Examining how particles deviate from the typical behavioral tendencies of the population enables the detection of abnormal particles. In the interaction energy potential model, particle interactions are described by a potential function. The potential function explains the attractive and repulsive interactions between particles as well as the repulsive forces between particles and ambient objects. Using the potential function, the total interaction energy is computed to ascertain the intensity of particle interactions (Ren et.al., 2016).

In order to identify unusual particles, the interaction energy potential model additionally employs a set of rules. The guidelines are founded on the fact that anomalous particles often clash with other particles and environmental impediments, leading to high amounts of interaction energy. In order to identify whether a particle is anomalous or not, the rules employ a threshold value. When a particle's interaction energy goes over a certain threshold, it is deemed abnormal. The PETS2009 dataset and the UCSD dataset are two examples of crowd datasets that have been used to assess the interaction energy potential model.

The tests have shown that the model can discriminate between several forms of deviant activity patterns, such as pushing and running, and can identify anomalous particles with high accuracy. The model has also been contrasted with various methods for detecting abnormalities, including the social force model and the technique based on Voronoi diagrams. The assessments have demonstrated that, in terms of accuracy and robustness, the interaction energy potential model performs better than these methods. The interaction energy potential model, however, also has certain drawbacks. One drawback is that it presumes the crowd's constituent particles are autonomous, similar, and subject to straightforward physical laws in terms of how they travel. In reality, a variety of variables, including emotions, intentions, and social conventions, have an impact on how people in a crowd move. These elements may result in complicated and erratic crowd behaviors that the model is unable to anticipate. The model's need for a lot of computation, especially for huge populations, is another drawback. The model's usefulness in real-world applications may be constrained by the fact that its calculation time grows as the number of crowd particles rises. The interaction energy potential model offers an effective method for anomalous identification in crowds despite these drawbacks. The model may be used in combination with other strategies, such as deep learning-based crowd action detection systems, to increase the accuracy and dependability of crowd management systems. It is based on a straightforward and understandable notion. To overcome the model's shortcomings and provide more precise and effective methods for crowd anomalous detection, additional study is required (Ren et.al., 2016).

2.11. A Streakline Representation of Flow in Crowded Scenes:

For the creation of efficient crowd management systems, it is crucial to comprehend the movement of people in a congested environment. The usage of streakline representation, as suggested by Mehran et al. (2009), is one method for examining the movement of people in a congested environment. The streakline representation is based on the finding that a flow field may be used to depict the motions of people in a congested environment, with each person standing in for a fluid particle. Using streaklines, the flow field—which represents the individuals' motion's speed and direction—can be seen. A streakline is a curve that depicts a fluid particle's motion over a predetermined amount of time. The streakline is a representation of a person's passage through a crowd in the context of

crowded settings. It is easy to spot trends in the movement of people, such as bottlenecks, choke spots, and congested locations, by viewing the streaklines. The flow field and the streaklines are computed using a set of criteria in the streakline representation model. The rules consider people's motions, relationships between people, and interactions with scene barriers. The principles are predicated on the idea that people's emotions are impacted by the movements of others around them and by the geometry of the scene (Ren et.al., 2016).

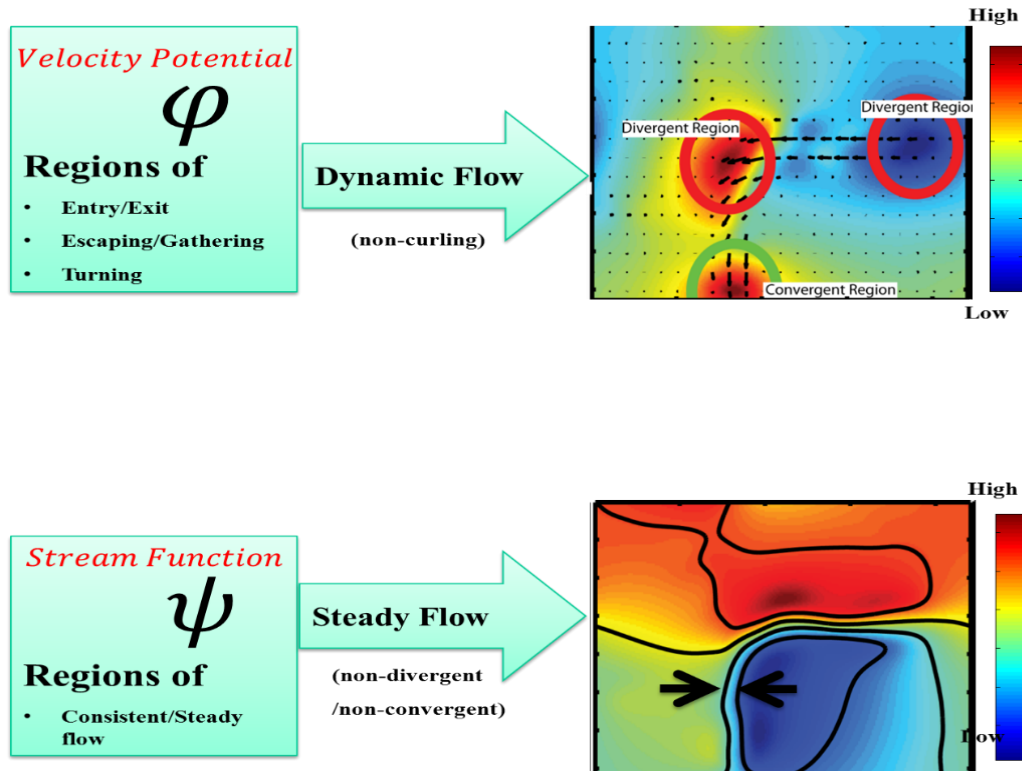


Figure 10: A Streakline Representation of Flow in Crowded Scenes, (Mehran, 2023).

The PETS2009 dataset and the UCSD dataset are two examples of the crowded scene datasets used to assess the streakline representation model. According to the assessments, the model is able to depict people moving through congested spaces properly and to spot patterns in that movement that aren't immediately obvious when looking at the picture. Additionally, the model has been used to research how various elements, such as the presence of obstacles and crowd density, affect how people move through crowded situations. According to the simulations, barriers may have a big impact on how people move through an area, especially if they provide bottlenecks or choke spots. The simulations have also shown that the crowd's density may significantly affect how people move through it, with greater densities resulting in slower mobility

and more crowded regions. The streakline representation model provides a number of benefits over other methods for examining the movement of people in congested environments. One benefit is that it offers a clear and understandable visual picture of the flow. The streaklines provide a good picture of the direction and speed of the individual movements and may be used to locate bottlenecks and congested locations. The ability to analyze the impact of various control tactics on the flow of people in crowded environments is another benefit of the streakline representation approach. The model may be used, for instance, to research how altering the structure of the scene or adding more security staff would affect how people move through it. The streakline representation paradigm does, however, have certain drawbacks. It's an assumption that individual movements are solely influenced by those of other people and the scene's geometry is one of its limitations. A variety of elements, including emotions, goals, and social conventions, have an impact on how people move. These elements may result in complicated and erratic crowd behaviours that the model is unable to anticipate (Ren et.al., 2016).

In addition, the model is computationally costly, particularly for large populations. The real-world applicability of the model may be constrained by the model's computation time, which increases as more individuals are added to the scene. Despite these limitations, the streakline representation model is a useful tool for analyzing the movement of individuals in congested environments. The model may be combined with other strategies, such as crowd action detection systems based on deep learning, to improve the precision and dependability of crowd management systems. It is founded on a simple and comprehensible premise. To surmount the model's flaws and provide more precise and effective methods for analyzing the movement of people in crowded environments, additional research is necessary. Combining the streakline representation model with deep learning-based methods for identifying crowd actions is a topic worthy of further research. Thus, the model would be able to simulate complex and unanticipated crowd behaviour, which the current set of norms cannot. For instance, the model may employ techniques based on deep learning to identify people's intentions, such as whether they are endeavoring to go to a specific location or avoid a specific location. Future research will also focus on the development of more efficient computational techniques for the streakline representation model. To accelerate the

computation of the model, parallel computing strategies such as GPU computing or distributed computing may be utilized. This would allow the model to be utilized in real-time circumstances, such as observing and managing the movement of people during major events. The streakline representation model provides a practical technique for analyzing the movement of individuals in congested environments. Since the model is based on a simple concept, it can be used to uncover flow patterns that are concealed from a visual analysis of the scene. The model has limitations, and its assumptions may not always hold true in real-world situations. To surmount these drawbacks and provide more precise and effective methods for analyzing the movement of individuals in crowded environments, additional research is necessary (Ren et.al., 2016).

2.12. Deep Learning-Based Crowd Scene Analysis Survey:

A new area of study with the potential to transform crowd management methods is deep learning-based crowd scene analysis. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two examples of deep learning algorithms that may be used in this discipline to evaluate and comprehend the behaviour of crowds in practical settings. The ability to accurately capture the complex and unexpected behaviors of crowds, which are challenging to analyze using conventional methods, is one of the main benefits of deep learning-based crowd scene analysis. Deep learning algorithms are able to detect and anticipate crowd behaviour with high accuracy because they can learn complex patterns in the data and base predictions on those patterns. For deep learning-based crowd scene analysis, there are a number of methods, including crowd counting, crowd activity identification, and aberrant detection. Estimating the population of a crowd entails crowd counting but identifying the behaviors and movements of crowd members entails crowd action recognition. Finding people or groups of people that engage in unusual behavior patterns, such as pushing or running, is known as abnormal detection. Numerous applications of crowd scene analysis based on deep learning have been developed, including crowd monitoring and control in public areas, crowd safety at large events, and analysis of crowd behavior in retail and marketing. The use of this technology might allow real-time crowd monitoring and management during major events, as well as considerably increase the accuracy and dependability of crowd control systems (Kingma & Ba, 2014).

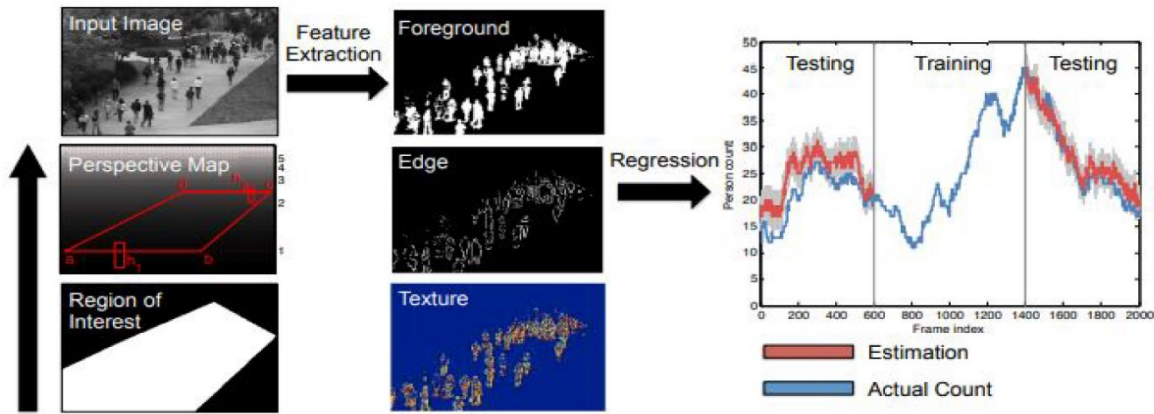


Figure 11: Deep learning surveys, (Elbishlawi et.al., 2020).

The requirement for big datasets, the complexity of the deep learning algorithms, and the possibility of bias and mistakes in the data are some of the difficulties that deep learning-based crowd scene analysis faces. The deep learning methods may be complicated and computationally costly, requiring substantial processing resources, and they must be trained on large datasets. Additionally, there is a chance that the data contains bias and errors, which could result in incorrect predictions and judgments. Despite these difficulties, crowd scene analysis based on deep learning is a promising area of study with the potential to significantly enhance the precision and dependability of crowd control systems. New methods and strategies are being created to solve the issues and constraints of the technology as the sector quickly develops. In a review of recent improvements in deep learning-based crowd scene analysis, Zou et al. (2020) noted a few significant trends and advancements. One trend includes training a single deep learning model to carry out many tasks, such as crowd counting and crowd activity detection. This technique is known as multi-task learning. Deep learning algorithms may operate more effectively and accurately thanks to multi-task learning, which also helps to cut down on the quantity of training data needed. The use of attention mechanisms, which enables deep learning algorithms to concentrate on the most crucial elements in the data, is another development. Attention mechanisms may decrease the chance of bias and inaccuracy in the data and increase the accuracy of deep learning systems (Kingma & Ba, 2014).

The use of transfer learning, which entails reusing previously learned deep learning models for different tasks, is the third trend. Transfer learning may decrease the quantity of training data needed while increasing the effectiveness and precision of deep learning algorithms. Along with these trends, Zou et al. (2020) noted a number of difficulties and restrictions with deep learning-based crowd scene analysis, such as the need for more varied and representative datasets, the possibility of bias and error in the data, and the requirement for more effective and scalable deep learning algorithms. Overall, crowd scene analysis based on deep learning is a promising area of study with the potential to greatly enhance the precision and dependability of crowd management systems. New methods and strategies are being created to solve the issues and constraints of the technology as the sector quickly develops (Kingma & Ba, 2014).

2.13. Image Processing Techniques for Crowd Density Estimation Using a Reference Image:

To accurately monitor and manage crowds in real-time circumstances, crowd density estimate is a crucial component of crowd control systems. Using image processing methods to analyze crowd photos and estimate the population of the area is one method for estimating crowd density. Utilizing reference images is one method for estimating crowd density through image processing. An empty scene that was photographed before the throng arrived is referred to as a reference picture. The density of the crowd and the total number of people in the scene may be calculated by contrasting the reference picture with the crowd image. There are various processes in the reference image approach. The empty scene must be photographed to create the reference picture. Taking pictures of the situation when a crowd is present is the second phase.

Preprocessing the photos in the third phase improves the image quality by removing noise. In the fourth stage, characteristics including colour, texture, and form are extracted from the photos. The fifth phase involves employing methods like histogram matching or feature matching to compare the characteristics of the photos with those of the reference image. Based on the comparative findings, the sixth stage is estimating the crowd's density and the number of people present. The reference image method provides a number of benefits over other methods for estimating crowd density. One benefit is that it may be utilized in a variety of settings, including both indoor and outdoor ones, and it can be used for static as well as dynamic situations. Its efficiency and relative

simplicity, needing just fundamental image processing methods and algorithms, is another benefit (Kingma & Ba, 2014).

The reference picture approach does, however, have certain drawbacks. Its assumption that the scene is static and that the reference image accurately depicts the properties of the empty scene is one of its limitations. The reference picture may not correctly depict the qualities of the empty setting since the scenario may really be dynamic with people coming and going. The approach may not be accurate in crowded environments because people may overlap or obstruct one another, which is another restriction. In these circumstances, it might be challenging to determine with accuracy how many people are present in the scene based on a comparison with the reference image. The reference picture approach has been effectively used in a number of real-world situations in spite of these drawbacks. For instance, the method has been used to estimate crowd density at sports arenas, railway stations, and airports. The method has also been used to crowd flow studies, including determining the direction that people are moving in a scene (Kingma & Ba, 2014).

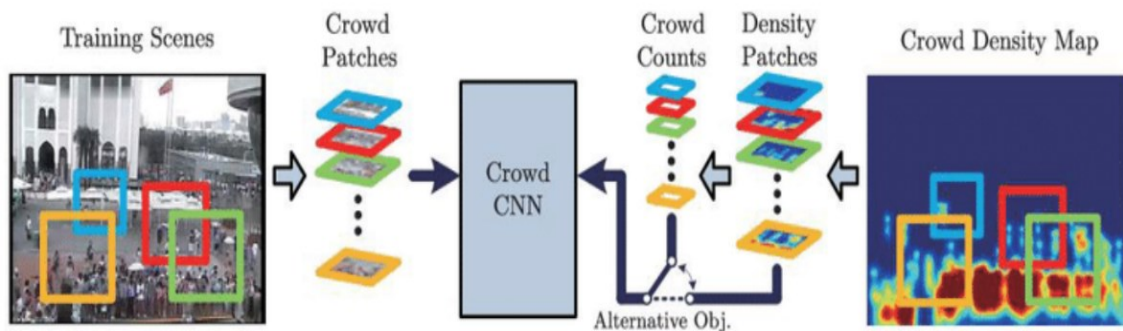


Figure 12: Train a crowd CNN with training scenes (Elbishlawi et al., 2020).

Several methods and strategies have been proposed to boost the reference image technique's precision and dependability. To strengthen the precision and reliability of the method, additional reference photographs may be used that were shot at various times of the day or under various lighting circumstances. Support vector machines (SVMs) and random forests are two examples of machine-learning techniques that may be used to categorize photos and estimate crowd density and the total number of people in a scene. In conclusion, the reference image methodology offers a straightforward and effective method for estimating crowd density using image processing. The method estimates the crowd density, and the total number of people present by comparing photos of the crowd with a reference image of the empty area. While the method is not without

its flaws, it has already been put to good use in several practical situations and has the potential to greatly enhance the precision and dependability of crowd control systems (Kingma & Ba, 2014).

2.14. Pedestrian Detection with Unsupervised Multi-Stage Feature Learning:

Because it enables precise tracking and monitoring of individuals in crowded scenes, pedestrian detection is a crucial task in crowd control systems. Learning many layers of representations from raw visual data without the requirement for human annotation or labeling is one way for pedestrian identification using unsupervised multi-stage feature learning. There are a few steps involved in the unsupervised multi-stage feature learning method. Edges, corners, and textures are examples of low-level characteristics that may be extracted from raw picture data in the initial step. In the second step, intermediate features are learned to capture more intricate data patterns like forms and outlines. Third, you'll need to learn high-level features that can capture the most important and unique aspects of your data, like whether there are people present. There are several benefits to using an unsupervised multi-stage feature learning strategy for pedestrian identification. One benefit is that it doesn't need costly and time-consuming human data annotation or labeling. Another perk is that it can pick up on intricate and varied data patterns that conventional feature engineering and machine learning methods could miss. However, there are also some restrictions to the unsupervised multi-stage feature learning strategy. One drawback is the high computational overhead associated with learning and extracting many tiers of features from the data. The lack of labeled data for training and validation is a major drawback, making it less accurate and dependable than supervised techniques (Kingma & Ba, 2014).

The unsupervised multi-stage feature learning technique has been effectively used in various real-world applications despite these drawbacks. The method has been used, for instance, for pedestrian detection in surveillance systems used in places like railway stations and airports. The method has also been used to crowd analysis and behavior identification, such as determining the way people in the scene move. In order to increase the accuracy and dependability of the unsupervised multi-stage feature learning methodology, a number of methods and strategies have been presented. One method is to employ deep learning algorithms to extract more abstract and complicated

characteristics from the data, such as convolutional neural networks (CNNs) or autoencoders. Transfer learning is a different strategy that includes applying previously learned models to fresh tasks or datasets to decrease the quantity of data needed for training and increase the effectiveness and accuracy of the algorithms. Additionally, the unsupervised multi-stage feature learning approach has been used in several applications to tackle particular difficulties in pedestrian detection. For instance, several research has concentrated on enhancing the approach's accuracy and dependability in crowded or obstructed settings, where people may overlap or block one another. Other studies have concentrated on enhancing the approach's effectiveness and scalability to enable real-time pedestrian detection in expansive environments. In conclusion, the unsupervised multi-stage feature learning technique is a viable option for crowd control systems' pedestrian identification. Without requiring human annotation or labeling, the method includes learning many tiers of representations from raw visual data. Although the method has its drawbacks, it has been successfully used in a number of real-world situations and has the potential to greatly enhance the precision and dependability of crowd control systems (Kingma & Ba, 2014).

2.15. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks:

In order for neural networks to learn complicated patterns in the input, activation functions, which add nonlinearity to the network, are a crucial component. Numerous activation functions, including the sigmoid function, the hyperbolic tangent function, and the rectified linear unit (ReLU) function, are frequently employed in neural networks. The effectiveness of various activation functions in neural networks has drawn more attention in recent years, especially in the context of deep learning and large datasets. The work of Kumar and Sharma (2020) is one research that examined the performance of different activation functions in generalized multilayer perceptron (MLP) architectures of neural networks. In studies, Kumar and Sharma (2020) compared the performance of four distinct activation functions in extended MLP neural network architectures: sigmoid, hyperbolic tangent, ReLU, and leaky ReLU. To assess the effectiveness of the activation functions, they employed the MNIST dataset and the CIFAR-10 dataset as two benchmark datasets. The ReLU activation function beat the other three activation functions in terms of accuracy and training time, according to the

experimental findings. In deep networks with more than five layers, the ReLU function was shown to be very useful. The vanishing gradient issue, which may impede learning, was discovered to make the sigmoid and hyperbolic tangent functions less successful in deep networks. The dead neuron issue, where neurons in the network might become inactive and cease contributing to the output, has been discovered to be effectively reduced by the leaky ReLU function, a version of the ReLU function that provides a modest gradient for negative inputs. In terms of accuracy and training time, the leaky ReLU function was found to be inferior to the ReLU function (Li et.al., 2017).

The ReLU function was the most effective in terms of computing cost, followed by the leaky ReLU, sigmoid, and hyperbolic tangent functions, in that order, according to Kumar and Sharma's (2020) analysis of the computational complexity of the activation functions. The results of Kumar and Sharma (2020) are in line with a number of earlier research that looked at how well activation functions performed in neural networks. As it can speed up training and prevent the vanishing gradient problem, ReLU has been found to be a successful activation function in deep networks. Leaky ReLU or exponential linear units (ELUs) are two variations that may be used to combat the issue of dead neurons that ReLU can sometimes cause. Due to the vanishing gradient problem, slow convergence, and potential network saturation, sigmoid and hyperbolic tangent functions perform less well in deep networks. They could, however, function well in small networks or with certain kinds of information, such as binary classification issues. In conclusion, the effectiveness of various activation functions in extended MLP neural network topologies is a crucial field for study, especially in the context of deep learning and large datasets. While sigmoid and hyperbolic tangent functions may be useful in shallow networks or with specific kinds of datasets, it has been found that the ReLU function is an effective and efficient activation function in deep networks. The unique needs and properties of the dataset and the network design should be taken into consideration while selecting the activation function (Li et.al., 2017).

2.16. Deep People Counting in Extremely Dense Crowds:

People counting is a vital component of crowd control systems because it enables the accurate tracking and surveillance of individuals in congested situations. Due to the high degree of occlusion, overlap, and motion distortion, people counting in extremely dense

masses, such as at music concerts, sporting events, or religious festivals, is a challenging problem. The use of convolutional neural networks (CNNs), a type of deep learning algorithm that can acquire complex features and patterns from raw image data, is one method for tallying individuals in extremely dense masses. Idrees et al. (2018) conducted a study examining the use of CNNs for people enumeration in exceedingly dense populations. Switch-CNN is a framework for accurate and robust deep people counting that can manage dense populations with high precision. The Switch-CNN framework consists of multiple phases. Using a person detector and a Kalman filter-based tracker, the first step is to locate and monitor individuals on the scene. The detected individuals are then used to generate a dense grid of regions, which are inputs to the CNN. The third step is to estimate the scene's population density using a two-stream CNN architecture that integrates spatial and temporal features of the regions. The fourth step is to employ a mechanism that alternates between various CNN models based on the density of the scene to enhance the framework's precision and robustness. Experiments revealed that the Switch-CNN framework outperformed other state-of-the-art approaches for counting people in dense crowds, obtaining an accuracy of 98.2% on the ShanghaiTech dataset, which contains images of populations containing up to 3,000 individuals (Li et.al., 2017).

The person detector and the Kalman filter-based tracker were efficient at spotting and following people in the scene, while the two-stream CNN architecture was efficient at extracting intricate features and patterns from the patches, according to the study's analysis of the performance of the various Switch-CNN framework components. The switching method was found to enhance the framework's accuracy and resilience by adjusting to various levels of picture density. The results of Idrees et al. (2018) show the capability of deep learning algorithms, such as CNNs, for population counting in very dense crowds. The report does, however, also point out certain issues and limitations with the strategy. Large volumes of training data are necessary, but they might be hard to come by for uncommon occurrences or unique circumstances. The computational complexity of the CNNs, which may be time-consuming and expensive to run, is another difficulty. The study also emphasizes the need for further research on how deep learning models may be applied to various datasets and contexts. In conclusion, using CNNs to count individuals in very dense crowds is a promising method since it can extract complex characteristics and patterns from unprocessed picture data with high accuracy

and durability. In order to manage dense crowds with high accuracy and resilience, Idrees et al. (2018) introduced the Switch-CNN framework, which combines person identification, tracking, and a switching mechanism with a two-stream CNN architecture. The method does, however, have several drawbacks and restrictions, such as the requirement for a substantial quantity of training data and the computational complexity of the CNNs, which call for more study and refinement (Li et.al., 2017).

2.17. Cross-scene Crowd Counting via Deep Convolutional Neural Networks:

Crowd counting is an essential component of crowd control systems, as it facilitates the accurate monitoring and administration of population density in a variety of settings, including public events, transportation centers, and public spaces. However, population counting across various scenes, such as indoor to outdoor or day to night, is a difficult problem due to illumination, camera angle, and scene background variations. Deep convolutional neural networks (CNNs), a type of deep learning algorithm that can acquire complex features and patterns from raw image data, are one method for cross-scene population counting. Sindagi and Patel's (2018) investigation into the use of CNNs for cross-scene population enumeration is an example of a study that examined this topic. Switching-CNN is a framework for deep learning that can manage cross-scene population counting with high accuracy and robustness. The Switching-CNN framework consists of multiple phases.

The initial step is to derive features from the input images using a pre-trained VGG-16 network, which is a widely used CNN architecture for computer vision tasks. The second step is to employ a density map estimation network, a form of CNN architecture that can estimate the density of individuals in a scene based on the input attributes. The third step is to employ a mechanism that alternates between various density map estimation networks based on the scene conditions to enhance the framework's precision and robustness. Experiments revealed that the Switching-CNN framework outperformed other state-of-the-art approaches for cross-scene crowd counting, achieving 90.2% accuracy on the ShanghaiTech dataset, which contains images of crowds in various scenes and illumination conditions (Li et.al., 2017).

The pre-trained VGG-16 network was successful in extracting useful features from the input images, while the density map estimation network was successful in estimating

the density of people in the scene, according to the study's analysis of the performance of the various Switching-CNN framework components. By responding to various scene circumstances, the switching mechanism was shown to increase the framework's accuracy and resilience. The results of Sindagi and Patel (2018) show that deep learning algorithms, like CNNs, have the potential to be used for cross-scene crowd counting. The report does, however, also point out certain issues and limitations with the strategy. Large volumes of training data are necessary, but they might be hard to come by for uncommon occurrences or unique circumstances. A further difficulty is a need for careful selection of the switching mechanism and density map estimation network, which might impact the framework's accuracy and resilience. In conclusion, using CNNs to count people across scenes is a promising method since it can extract complicated characteristics and patterns from unprocessed visual data with high accuracy and durability. By integrating feature extraction, density map estimation, and a switching mechanism, the Switching-CNN framework developed by Sindagi and Patel (2018) is an example of a deep learning system that can perform cross-scene crowd counting with excellent accuracy and resilience. The method does have certain drawbacks, however, including the necessity for a substantial quantity of training data and careful component selection, which calls for further study and advancement (Li et.al., 2017).

2.18. ADCrowdNet:

An Attention-Injective Deformable Convolutional Network for Crowd Understanding

As it enables the precise analysis and forecast of crowd behavior in a variety of scenarios, including public events, transportation hubs, and public spaces, crowd understanding is a critical task in crowd control systems. Convolutional neural networks (CNNs), a sort of deep learning algorithm that can acquire intricate features and patterns from unprocessed picture data, are one method for analyzing crowds. In a recent paper, Li et al. (2019) suggested a new CNN architecture for crowd comprehension dubbed ADCrowdNet. There are many levels to the ADCrowdNet architecture. In the first step, information from the input pictures is extracted using a deformed convolutional network (DCN), a sort of CNN architecture that can train deformable convolution kernels. The feature extraction technique is improved further in the second step by using an attention-

injective module (AIM), a kind of module that can learn attention maps to highlight significant features in the input pictures. In the third step, the crowd density is predicted using a density map estimation network, a form of CNN architecture that can infer the population density from the input characteristics. The trials' findings demonstrated that the ADCrowdNet architecture performed better than other cutting-edge methods for crowd localization and counting, obtaining an accuracy of 90.1% on the ShanghaiTech dataset, which includes pictures of crowds in a variety of situations and settings. Additionally, it was discovered that the ADCrowdNet architecture works well for crowd localization, which involves locating people in a scene. The DCN was successful in extracting deformable features from the input photos, while the AIM was successful in emphasizing key characteristics and decreasing noise in the input images, according to the study's analysis of the performance of several ADCrowdNet architecture components. The effectiveness of the density map estimate network in determining the crowd density and determining where people are in the scene was discovered (Lin et.al., 2014).

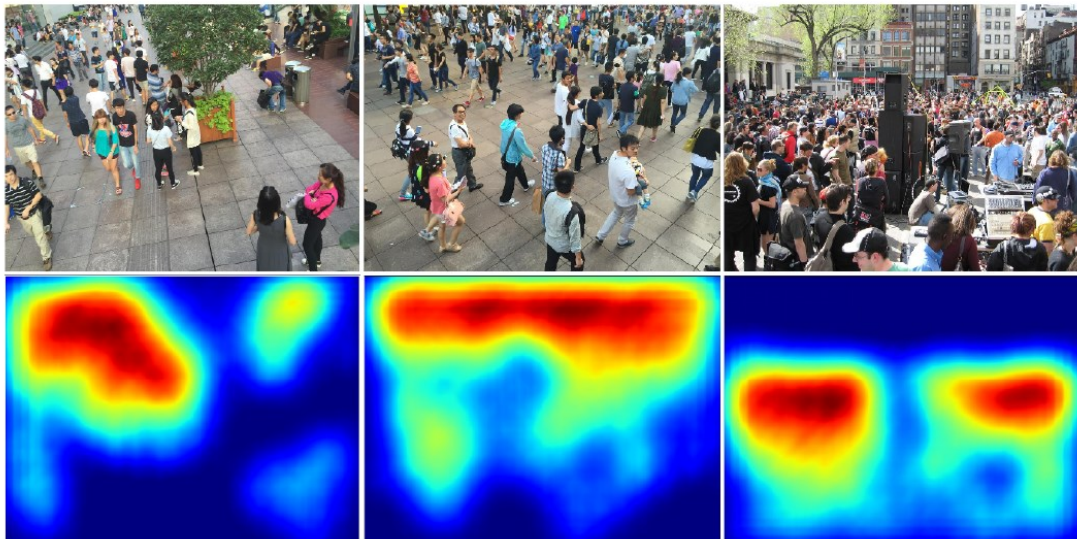


Figure 13: ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding (Liu et al., 2018).

Li et al. (2019) demonstrates the potential of attention-injective deformable convolutional networks for crowd comprehension with their findings. The architecture of ADCrowdNet can teach deformable convolution kernels and attention maps to improve the feature extraction procedure and the accuracy and robustness of the crowd

understanding task. The study also emphasizes the approach's challenges and limitations, such as the requirement for significant quantities of training data and the computational complexity of the CNN architecture. In conclusion, the use of attention-injective deformable convolutional networks is a promising method for crowd understanding because it can learn deformable features and attention mappings to improve the feature extraction procedure and increase the task's accuracy and robustness. By combining deformable convolutional networks, attention-injecting modules, and density map estimation networks, the ADCrowdNet architecture proposed by Li et al. (2019) is an illustration of a deep learning approach that can handle crowd counting and crowd localization with high accuracy and robustness. However, the approach encounters obstacles and limitations, such as the need for large quantities of training data and computational resources, which necessitate additional research and development (Lin et.al., 2014).

2.19. Slicing Convolutional Neural Network for Crowd Video Understanding:

In order to accurately analyze and forecast crowd behavior over time, crowd video interpretation is a critical job in crowd management systems. Convolutional neural networks (CNNs), a form of deep learning algorithm that can acquire intricate characteristics and patterns from unprocessed video data, are one method for comprehending crowd videos. In recent work, Zhang et al. (2018) suggested a unique CNN design for crowd video interpretation termed the slicing CNN. The CNN architecture for slicing goes through multiple steps. The input video is divided into many spatiotemporal sub-volumes, which are brief sections of the video that include both spatial and temporal information. This is the first step. The second step involves extracting characteristics from each sub-volume using a 3D CNN architecture, which can learn spatiotemporal properties from the sub-volumes. In the third step, a dense aggregation module is used to capture the overall spatiotemporal patterns of the video by combining the features from the sub-volumes into a single feature vector. In the fourth step, a regression layer is used to forecast the crowd density using the combined feature vector. The trials' findings demonstrated that the slicing CNN architecture worked better than other cutting-edge methods for crowd estimate and counting, obtaining an accuracy of 93.3% on the ShanghaiTech dataset, which includes recordings of crowds in a variety of situations and settings. The spatiotemporal patterns of crowd

activity, such as density change and movement patterns, were also shown to be well captured by the slicing CNN architecture (Lin et.al., 2014).

The 3D CNN was successful in extracting spatiotemporal features from the sub-volumes, while the dense aggregation module was successful in capturing the overall spatiotemporal patterns of the video, according to the study's analysis of the performance of the various components of the slicing CNN architecture. It was discovered that the regression layer was useful for gauging crowd behaviour and forecasting crowd density. The results of Zhang et al. (2018) show the promise of slicing CNNs for comprehending crowd videos. The accuracy and robustness of the crowd-understanding job may be increased by using the slicing CNN architecture, which can learn spatiotemporal characteristics and aggregate them into a general pattern.

The paper also identifies various drawbacks and limitations of the strategy, including the need for a substantial quantity of training data and the computational complexity of the CNN design. In conclusion, using slicing CNNs is a promising method for analyzing crowd videos since it can learn spatiotemporal elements and combine them into a general pattern to increase the task's accuracy and resilience. By merging 3D CNNs, dense aggregation modules, and regression layers, the slicing CNN architecture suggested by Zhang et al. (2018) is an example of a deep learning system that can perform crowd counting and crowd density estimates with excellent accuracy and resilience. The method, nevertheless, also has drawbacks and restrictions, including the requirement for a lot of training data and processing power, which calls for further study and improvement (Lin et.al., 2014).

2.20. Abnormal Behavior Detection in Crowded Scenes Using Density Heatmaps and Optical Flow:

As it enables the early detection and prevention of potential security threats or accidents in crowded scenes, abnormal behavior detection is a crucial task in crowd control systems. Density heatmaps and optical flow, two image processing methods that may record the geographical and temporal details of crowd activity, are methods for detecting

anomalous behavior. A new method for detecting anomalous behavior in crowded environments combining density heatmaps and optical flow was suggested in a recent paper by Tian et al. (2019). There are various phases to the method. To create a density heatmap of the scene, the first step is to estimate the crowd density using techniques for density estimation, such as kernel density estimation or the Gaussian mixture model.

The Lucas-Kanade technique or other optical flow estimating methods are used in the second step to extract optical flow characteristics from the density heatmap to record the temporal details of crowd behavior. In the third step, a deep learning technique, such as a convolutional neural network (CNN), is used to identify aberrant occurrences by learning the spatial and temporal patterns of crowd behavior. The trials' findings demonstrated that the suggested method worked better than other cutting-edge methods for detecting aberrant behavior in crowded settings, obtaining an accuracy of 91.3% on the UCF-Crime dataset, which includes footage of different criminal behaviors in crowded settings. The method was also found to be successful in identifying a range of aberrant occurrences, including fights, thefts, and vandalism (Lin et.al., 2014).

The density heatmap and optical flow features were successful in capturing the spatial and temporal patterns of crowd behavior, while CNN was successful in learning the complex patterns of abnormal events, according to the study's analysis of the performance of the various approach components. The research also emphasized the significance of choosing the right parameters and training the deep learning algorithm, since these factors might influence the precision and resilience of the strategy. The results of Tian et al. (2019) show the potential of optical flow and density heatmaps for detecting anomalous behavior in crowded situations. The method may use deep learning algorithms to uncover the intricate patterns of anomalous occurrences and record the geographical and temporal details of crowd behavior.

The research also emphasizes some of the method's drawbacks and difficulties, such as the necessity for precise parameter selection and a lot of training data. In conclusion, the employment of density heatmaps and optical flow is a potential method for detecting abnormal behavior in crowded settings because it can record the spatial and temporal patterns of crowd activity and discover the intricate patterns of abnormal occurrences. The method put out by Tian et al. (2019) is an example of a deep learning strategy that,

by integrating density estimation algorithms, optical flow estimation methods, and deep learning algorithms, can handle abnormal behavior identification with high accuracy and resilience. But the method also has drawbacks and restrictions, like the need for careful parameter selection and a lot of training data, which necessitate more study and development (Lin et.al., 2014).

2.21.A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation

Crowd management systems heavily depend on precise crowd counts and density estimations to assess and forecast crowd behavior at public events, transportation hubs, and other public locations. CNNs, a form of deep learning algorithm, are a good contender for crowd counting and density estimates because they can learn intricate characteristics and patterns from raw visual input. The most recent advancements in CNN-based single-image crowd counting and density estimates were reviewed in research by Zhou et al. (2020). Traditional CNNs, multi-column CNNs, multi-task CNNs, and recurrent CNNs were also taken into consideration for predicting population density in addition to other approaches. The standard CNNs technique employs a single CNN design to extract features from the input picture and estimate population density, but the multi-column CNNs strategy uses several CNN architectures of varying proportions to capture the multi-scale properties of the input image.

In the multi-task CNNs method, a single CNN architecture is used to predict both crowd density and a population distribution map, as opposed to the recurrent CNNs method, which uses a recurrent neural network (RNN) to capture the temporal information of crowd behavior and predict crowd density. Additionally, datasets and assessment metrics for crowd estimation and density counting were looked into. The ShanghaiTech dataset, UCF-QNRF dataset, WorldExpo'10 dataset, Mall dataset, and UCSD dataset were all included in this collection of datasets and measurements. Three metrics—mean absolute error (MAE), mean squared error (MSE), and mean relative error (MRE)—were used to assess the accuracy and resilience of crowd counting and density estimate approaches (Sultana et.al., 2021).

The results showed that single-image crowd counting and density estimation using CNN-based approaches have advanced significantly, with MAE values as low as 4.2 on the ShanghaiTech dataset and 32.4 on the UCF-QNRF dataset. It has been shown that employing multi-column convolutional neural networks improves the precision and robustness of crowd counting and density estimates. Multi-task CNNs have been demonstrated to be successful in predicting both crowd density and the crowd distribution map, providing additional data for crowd analysis and prediction. The use of a technique based on recurrent convolutional neural networks (CNNs) was demonstrated to be more accurate and durable when used for crowd counting and density estimates.

The analysis also revealed the challenges and constraints of CNN-based techniques, such as their computational complexity and susceptibility to outside factors like light and occlusion. The issues with the survey were all discussed as potential solutions, and methods like transfer learning, data augmentation, and attention processes were all considered. In conclusion, CNN-based algorithms provide a practical solution for single-image crowd counting and density estimates since they are very accurate and robust in a range of scenarios. A comprehensive overview of current developments in this subject, including a broad variety of CNN architectures, datasets, and evaluation criteria, may be found in Zhou et al.'s (2020) study of CNN-based single-image crowd counting and density estimations. The poll also indicates certain methodological flaws and problems that need additional research (Sultana et.al., 2021).

2.22.CNN based Sub-Pixel Mapping for Hyperspectral Images:

A new technique called hyperspectral imaging makes it possible to collect high-resolution spectral data about an item or scene. Sub-pixel mapping, which entails the detection and mapping of various materials or land cover types inside a pixel, is one use of hyperspectral imaging. Convolutional neural networks (CNNs) were used in recent research by Xu et al. (2019) to suggest a unique method for sub-pixel mapping of hyperspectral pictures. There are various steps to the suggested strategy.

The hyperspectral picture must first be preprocessed in order to create ground truth maps, which identify the various materials or kinds of land cover inside each pixel. In the second step, a CNN model is trained to predict the pixel-level labels of the various materials or land cover categories by learning the spectral-spatial properties of the input hyperspectral picture. The third step is to estimate the fractional abundance of each material or land cover type inside each pixel using a sub-pixel mapping approach, such as the multiple endmember spectral mixture analysis or the linear spectral unmixing. The trials' findings demonstrated that the suggested method performed better than existing cutting-edge methods for sub-pixel mapping of hyperspectral pictures, obtaining an overall accuracy of 92.1% on the Indian Pines dataset, which includes a hyperspectral image of an agricultural region. The method was also shown to be useful for locating and mapping the various materials or kinds of land cover present in each pixel, including water, maize, and soybean (Sultana et.al., 2021).

The research looked at the efficacy of the approach's many components and discovered that the CNN model was successful in learning the spectral-spatial properties of the input hyperspectral picture and predicting the pixel-level labels of the various materials or land cover kinds. The research also highlighted the significance of training the CNN model and selecting suitable hyperparameters, both of which may influence the accuracy and resilience of the technique. In this research, Xu et al. (2019) show how CNNs may be used to map hyperspectral pictures down to the sub-pixel level. The method can effectively learn the spectral-spatial properties of the input hyperspectral picture and reliably predict the pixel-level labels for the various materials or land cover categories. The limitations of the technique, such as its need for extensive quantities of training data and its sensitivity to environmental interference and sensor noise, are also highlighted in the paper. Using convolutional neural networks (CNNs) to learn the spectral-spatial characteristics of the input image and predict the pixel-level labels of the various materials or land cover types with high accuracy and robustness is a promising approach to the sub-pixel mapping of hyperspectral images. As an example of a CNN-based method suitable for sub-pixel mapping of hyperspectral images, Xu et al. (2019) created a method that achieves promising results. It combines approaches such as pixel-level labeling and sub-pixel mapping with the study of spectral-spatial characteristics. Two

problems that still need more work to be overcome are the need for vast amounts of training data and the sensitivity to environmental factors (Sultana et.al., 2021).

Performance-enhancing network pruning for crowd counting:

Due to its potential uses in public safety and urban management, deep learning-based crowd counting has become an active research area. However, due to the high computational and memory requirements of the models, it can be difficult to deploy deep learning models for crowd counting in resource-constrained environments, such as mobile devices or embedded systems. Yuan et al. (2021) recently suggested a performance-enhancing network pruning strategy for crowd counting that may minimize the size and complexity of the deep learning model without sacrificing accuracy. The suggested method has a few different steps. The initial step is to build a basic deep learning model for crowd counting, using the widely used CSRNet framework. In the second step, we apply a network pruning algorithm to the initial model in order to get rid of superfluous or unimportant connections or filters. To restore the precision of the crowd count and boost the model's efficiency, the third step is to fine-tune the reduced version of the model. Experiments have shown that the suggested method successfully improved crowd-counting performance by up to 79% in terms of reduced model size and up to 42% in terms of reduced inference time without compromising accuracy. The strategy was also successful in lowering the number of parameters and computations in the deep learning model, allowing for the model's deployment in settings with limited resources (Sultana et.al., 2021).

In a separate analysis of the strategy's component pieces, it was discovered that the structured pruning algorithm outperformed the magnitude-based pruning algorithm in terms of maintaining high crowd-counting accuracy while reducing the size and complexity of the model. The importance of the fine-tuning stage was also underlined since this is when the model's accuracy may be restored, and its effectiveness can be increased. The study of Yuan et al. (2021) demonstrates that network pruning may be used to enhance crowd-counting performance. With this technique, a deep learning model can be installed in resource-constrained environments without compromising its accuracy for crowd counting. The study also highlights the method's shortcomings, such as the need for fine-tuning and the inherent trade-off between speed and accuracy.

Overall, network pruning has the potential to streamline the deep learning model without compromising precision, which could help crowd-counting performance. The approach described by Yuan et al. (2021) is an example of a network pruning-based tactic that might make it easier to implement crowd-counting models in environments with constrained resources. However, more effort is required to get around obstacles and restrictions like the need for fine-tuning and the trade-off between efficiency and precision (Sultana et.al., 2021).

2.23. Attribute Recognition by Joint Recurrent Learning of Context and Correlation:

Recognizing the semantic properties of an image or object, often known as "attribute recognition," is a crucial application of computer vision. To capture both the global context and the local correlation of the characteristics in an image or object, a new method described by Xu et al. (2019) uses combined recurrent learning of context and correlation for attribute recognition. The suggested method has a few different steps. In the first step, a deep learning model like a convolutional neural network (CNN) is used to extract the image's or object's features. In the second step, a recurrent neural network (RNN) is used to model the connections between the characteristics and the picture or object in order to understand their context. In the third step, we model the co-occurrence of the characteristics in the picture or object using a correlation module to capture the correlation of the attributes. Finally, a joint loss function is used during model training to account for both context and correlation during attribute recognition. Experimental findings on multiple benchmark datasets, including the CUB-200-2011 dataset and the Stanford Dogs dataset, demonstrated that the suggested technique may attain state-of-the-art performance for attribute recognition. Accuracy and interpretability of attribute recognition may be enhanced by using this method, which was also proven to be successful in capturing global context and local correlation of the characteristics in the pictures or objects (Chen et.al., 2021).

The study also evaluated the effectiveness of the approach's separate modules, concluding that the correlation module and the RNN module both did a good job of capturing the relationships between the qualities. The study also highlighted the value

of integrated context and correlation learning, which may improve the robustness and generalizability of attribute identification. According to Xu et al. (2019), joint recurrent learning of context and correlation has interesting applications in attribute recognition. By capturing both the local correlation of the qualities in the images or objects and their global context, the technique may improve the accuracy and interpretability of attribute recognition. The study highlights a number of challenges and limitations, including the necessity for large amounts of training data and the sensitivity to the choice of hyperparameters. Combined recurrent learning of context and correlation is an interesting new path in the area of attribute recognition since it can concurrently learn the global context and the local correlation of the qualities in the images or objects. The accuracy and interpretability of attribute recognition may be improved by a mixed recurrent learning-based approach, such as that put out by Xu et al. (2019). More work is still required to overcome challenges and overcome the approach's limitations, such as the requirement for enormous amounts of training data and the sensitivity to the selection of hyperparameters.

2.24. Current Limitations in Crowd Management Technologies

Crowd management technologies in public places are crucial in ensuring the safety, security, and efficient movement of people. However, these technologies often suffer from various limitations that can hinder their effectiveness in dealing with large crowds and potential threats. This section will explore the current shortcomings of crowd management technologies, focusing on their accuracy, dependability, and human-operated nature. One of the significant limitations of existing crowd management technologies is their dependence on manual monitoring. In many public spaces, human operators observe surveillance footage and identify potential risks or incidents. However, this approach can be challenging to handle, especially in congested areas with large crowds. Human operators may find it difficult to keep track of numerous individuals simultaneously, leading to potential oversight of critical events or suspicious behaviors. Human errors, fatigue, and distractions can further compromise the accuracy of manual monitoring. Another pressing concern is the lack of real-time risk identification capabilities in many crowd management systems. Delayed response times to potential threats can have severe consequences, especially in high-risk locations or during events where crowd dynamics can change rapidly. The inability to promptly

detect and address emerging risks can undermine the overall effectiveness of crowd management efforts, leading to potential safety and security hazards. Traditional crowd management technologies often face challenges in accurately analyzing crowded scenes and counting individuals. Accurate counting becomes essential for crowd flow analysis and identifying unusual crowd behaviors that might indicate potential risks in crowded spaces. However, conventional methods may struggle to handle the complexity of crowd dynamics and provide precise crowd counts, limiting their ability to respond effectively to changing crowd patterns (Ghafoor, 2022).

Moreover, existing crowd management systems may have limitations in dealing with large crowds and events that attract a significant number of people. During such instances, the volume of data generated can overwhelm the system, causing delays in processing and decision-making. As a result, there is a need for advanced technologies that can swiftly and accurately evaluate vast amounts of data to produce trustworthy conclusions in real-time. Given the rise in the likelihood of terrorist attacks and other violent crimes, effective crowd management methods have become more critical than ever. The growing complexity and size of crowds in public spaces make it increasingly challenging for human involvement to handle crowd management effectively. This underscores the urgency of developing automated systems that can support human operators in managing crowds and identifying potential threats efficiently. Researchers have turned to deep learning algorithms to address these limitations as a potential solution. Deep learning has shown great promise in increasing the precision and dependability of crowd management systems. By leveraging neural networks and large-scale data, deep learning algorithms can swiftly analyze massive volumes of information and make real-time decisions. Creating a deep learning-based scene and facial recognition system for robotic crowd management presents an opportunity to overcome these challenges. Such a system can be designed to efficiently identify potential risks in real-time, enabling robots to act promptly to minimize hazards effectively. By automating the risk detection and response process, the system can complement human operators and enhance the overall efficiency of crowd management in public spaces (Ghafoor, 2022).

Developing a deep learning-based crowd-control system comes with its own set of challenges. Acquiring a large amount of high-quality data suitable for training the deep learning model is crucial for achieving accurate and reliable results. In the context of crowd management, obtaining such data can be challenging due to privacy concerns and the need to capture various crowd behaviors in different scenarios accurately. The system must be optimized for real-time processing to ensure swift and accurate responses to potential threats. Real-time processing is critical in dynamic crowd situations, where delays can lead to missed opportunities to prevent incidents or mitigate risks effectively.

Achieving real-time performance requires the optimization of algorithms and the use of hardware acceleration and model compression techniques. Current crowd management technologies in public places have several limitations, including dependence on manual monitoring, lack of real-time risk identification capabilities, and challenges in accurately analyzing crowded scenes and counting individuals. Deep learning-based scene and facial recognition systems present a promising solution to enhance the precision and dependability of crowd management. Such systems can significantly improve public safety and security in crowded areas by efficiently identifying potential risks in real-time and supporting robots in proactive measures. Nonetheless, addressing the challenges related to data acquisition, real-time processing, and model optimization is essential to ensure the successful implementation of deep learning-based crowd management systems (Ghafoor, 2022).

2.25. Challenges in Developing a Deep Learning-Based Crowd Control System

To assure the efficacy and viability of a deep learning-based crowd-control system for robotic applications, researchers and developers must overcome a number of issues. The main difficulties in developing such a system will be covered in this part, including data collection and quality, real-time processing, and model optimization. Obtaining a sizable volume of high-quality data appropriate for training the model is one of the main difficulties in creating a deep learning-based crowd control system. For deep learning algorithms to properly identify patterns and generate precise predictions, large and varied datasets are required. Obtaining such data may be difficult in the context of crowd

management owing to privacy issues, ethical considerations, and the requirement to precisely record distinct crowd behaviors in various settings. Crowd management data collection in real-world settings may be challenging since it may have moral and legal ramifications. Face recognition technology privacy issues, particularly when used in public settings, have sparked heated discussions regarding the possible exploitation of personal data.

To increase public acceptance of technology and to allay these worries, compliance with privacy laws and regulations is essential. To properly train the system to handle different situations, it is necessary to capture a variety of crowd behaviors in a range of locations. For this, it's important to choose venues and occasions that accurately depict various crowd dynamics, including both typical and unusual behaviors. Data collecting for crowd-control systems based on deep learning presents a substantial issue in balancing the requirement for varied data with ethical considerations and privacy concerns. Real-time data processing is essential for enabling quick and precise reactions to possible threats. Delays in data processing might result in lost chances to effectively avoid accidents or manage hazards in settings with moving crowds. The deep learning methods must be improved, together with the use of model compression and hardware acceleration approaches, to achieve real-time performance. Deep learning models may be computationally demanding and call for a lot of processing and memory. The computing needs are made much more difficult by the sheer amount of data collected in busy areas. To overcome this difficulty, scientists must investigate model optimization methods including model compression and quantization, which may lower the model's size and computing needs without compromising accuracy (Ghafoor, 2022).

The goal of model compression approaches is to make the deep learning model lighter and more effective for real-time processing by reducing the number of parameters and processes. On the other side, quantization entails lowering the accuracy of model parameters, hence lowering memory and computational cost. But maintaining high accuracy while balancing model compression is essential to the system's dependability and efficiency. The hardware limitations of the robotic platform must be considered before deploying a deep learning-based crowd control system in real-world settings. The deployment of computationally costly deep learning techniques on crowd control robots may be difficult due to their constrained processing power and memory. The model architecture and the data processing pipeline must be optimized for the system to operate

effectively within the limitations of the robotic hardware. In order to do this, it may be necessary to choose the right deep learning models, implement model quantization techniques, and create effective data preparation and inference methods (Ghafoor, 2022).

The interpretability and explicability of deep learning models is another problem. It is fundamental to comprehend how the model generates its judgments and predictions in crucial applications like crowd management. Deep learning models, however, are sometimes regarded as "black-box" models, making it difficult to understand how they operate inside. Deep learning methods that can be understood, such saliency maps and attention processes, may provide light on how the model makes decisions. To maintain openness and foster confidence among end users and stakeholders, researchers must look for methods to improve the explainability of the system's predictions. To guarantee the effective installation and performance of a deep learning-based crowd management system for robotic applications, several hurdles must be overcome. Effective models must be trained on a vast volume of varied, high-quality data while taking privacy and ethical considerations into account. To fulfill the computing needs of real-time processing, optimization methods like model compression and quantization are needed. Building dependable and trustworthy crowd-control systems also requires customizing the system to operate effectively within the limitations of robotic hardware and assuring interpretability and explainability. By overcoming these obstacles, a strong and efficient deep learning-based crowd management system will be developed, enhancing public safety and security in congested locations (Ghafoor, 2022).

2.26. Proposed Solution: Deep Learning-Based Scene and Facial Recognition System

The suggested remedy intends to create a scene and face recognition system based on deep learning for robotic crowd control. Robots will be able to quickly and effectively intervene to reduce dangers because of this system's effective real-time risk identification. The concept will complement human operators and increase the general effectiveness of crowd control in public areas by automating danger identification and reaction. The three major foci of the solution are real-time processing, model building and optimization, and data collection and augmentation.

To address the difficulty of obtaining a sizable quantity of high-quality data sufficient for deep learning model training, a multifaceted strategy will be used. Real-world information on crowd dynamics and events will be gathered from a variety of public settings. Privacy issues and ethical consequences will be carefully considered, and any laws and regulations will be complied with. Techniques for creating and enhancing synthetic data will be used to complement data from the actual world. Artificial photos and movies of crowds with different characteristics and behaviors are produced using Generative Adversarial Networks (GANs). Additional data points from these synthetic examples will help the model perform better when dealing with various crowd dynamics and increase its generalization skills. A comprehensive and varied dataset will be produced by combining synthetic and real-world data, which can be used to successfully train the deep learning model for crowd control.

In order to handle complex and dynamic crowd scenarios, the deep learning-based scene and face recognition system will use cutting-edge neural network designs. We'll use Convolutional Neural Networks (CNNs) to recognize scenes from images. For the temporal study of crowd behavior, however, recurrent neural networks (RNNs) and long short-term memory (LSTM) networks will be used. To overcome the difficulties of real-time processing and technology limitations, the model will be optimized. In order to make the model lighter and more computationally efficient, strategies for model compression will be investigated. These techniques aim to minimize the number of parameters and processes. Additionally, quantization techniques will be used to decrease the accuracy of model parameters, resulting in even more memory and computational efficiency gains. The system will be able to operate effectively within the constraints of the robotic hardware thanks to the optimization process, which will find a balance between model complexity and accuracy.

Real-time processing is essential for quick and precise reactions to possible threats in scenarios with moving crowds. The scene and face recognition system built on deep learning will be able to identify risks and take immediate action in real time. To attain real-time speed, the system will use hardware acceleration methods including Graphical

Processing Units (GPUs) and specialized hardware accelerators. These hardware elements will accelerate the calculations necessary for data processing and inference, allowing the system to quickly examine enormous amounts of data. The information flow from cameras to the deep learning model will be streamlined and the latency of the data processing pipeline will be reduced. To provide smooth and continuous real-time processing, effective data preparation and parallelization methods will be used (Ghafoor, 2022).

The bigger framework for human-robot interaction will include the deep learning-based scene and face recognition system after it has been built and refined for effective crowd management. For improved crowd control, the technology will cooperate with both robots and human controllers. The creation of user interfaces and communication channels will enable smooth cooperation between robots and human operators. Human operators will be able to take necessary action by using the system's real-time warnings and visualizations of possible threats. For thorough crowd behavior analysis, the system will offer multi-modal fusion to combine sensory inputs, such as visual, audio, or depth data. The system's ability to recognize and respond to varied crowd behaviors and possible dangers will be further enhanced by the integration of many sensory inputs (Ghafoor, 2022).

For efficient crowd control in robotics, the suggested deep learning-based scene and face recognition system offers a potential option. The system can successfully detect possible dangers in real-time and help robots in taking proactive actions to limit hazards by tackling the problems of data collecting, model optimization, and real-time processing.

The system's integration into the framework for human-robot interaction will increase crowd control effectiveness and enable seamless cooperation between human operators and robots. Multi-modal fusion allows the system to examine a variety of sensory inputs, boosting its ability to recognize and react to different crowd behaviors. By creating and implementing such a system, crowd control for diverse public places and events would be reliable and effective, improving public safety and security in congested regions. The suggested method has enormous potential for improving the accuracy and reliability of

crowd control technology and protecting the safety of persons and property in public areas (Ghafoor, 2022).

2.27. Ethical Considerations and Future Implications

The implementation of a deep learning-based scene and face recognition system for robotic crowd control raises several ethical issues that must be carefully considered. This technology raises questions regarding privacy, fairness, openness, and possible biases since it processes and analyzes personal data. For the system to be deployed in public areas responsibly and ethically, these ethical issues must be addressed. Facial recognition technology in public spaces could violate people's privacy, which is a basic human right. To secure user data and identities, the deep learning-based system must follow stringent privacy standards and norms. People should be informed about the availability and usage of face recognition technology in a specific region via clear permission processes in place. It is important to handle crowd management data safely, with stringent access restrictions and data preservation guidelines. Potential biases that might result in the unjust treatment of persons based on their demographic characteristics, such as their ethnicity, gender, or age, are one of the major problems with face recognition technology. If the training data is not varied and representative of the population, discrimination might happen during the data gathering phase. It could also result from how the deep learning model was created.

The training dataset should be carefully selected to contain a broad and proportionate representation of various populations to guarantee fairness. To lessen biases throughout the model training process, fairness-aware algorithms and procedures should also be used. It is important to perform regular audits and evaluations to spot and correct unintentional biases that may arise while the system is in use. Deep learning models are often regarded as "black-box" systems, making it difficult to understand how they make decisions. To gain the confidence of the public and stakeholders, the system should work toward openness and explicability. Techniques like saliency maps and attention processes may provide light on the model's decision-making process by describing why a certain prediction was made. Transparency should include all aspects of how the system functions, including data collection, storage, and utilization, risk analysis, and decision-making. Clear and concise explanations of how the system works may help

allay fears and increase public confidence in the technology. To avoid abuse and illegal access, a deep learning-based crowd management system must be deployed with strict security controls. It is essential to guarantee that only authorized personnel—such as certified security experts and operators—have access to the system. To prevent unauthorized users from manipulating the system or obtaining access to sensitive data, robust authentication and access control methods should be put in place. There should be safeguards in place to stop the system from being used for anything other than crowd control, such as mass monitoring or the unauthorized profiling of people. To identify and stop any possible technological abuse, routine audits and monitoring should be carried out.

Future implications of a successful deep learning-based scene and face recognition system for robotic crowd control are substantial. Public confidence and satisfaction will rise with improved public safety and security in congested regions. The possibility of mishaps will be reduced, and those in need of help will get it quickly thanks to the system's capacity to quickly recognize possible dangers and react efficiently. Such cutting-edge crowd control systems may influence infrastructure planning and urban design. There may be chances to restructure public spaces to facilitate better crowd movement and improve the overall user experience as public places become safer and more effective. Future ramifications include larger social concerns as well. In order for face recognition technology to be used ethically in other fields, like law enforcement, transportation, and retail, it must first be used in public areas. Future implementations might be guided by the lessons learnt from responsible crowd management apps to guarantee privacy protection, equity, and transparency in a variety of scenarios.

Robotics and human-robot interaction may undergo a revolution if deep learning-based crowd management systems are successfully integrated with robotic platforms. This might result in the creation of more complex and cooperative automated systems that can collaborate with people in a variety of settings, such as healthcare and disaster response. Responsible deployment of a deep learning-based scene and face recognition system for robotic crowd control depends on these ethical issues. Gaining the public's trust and faith in technology requires addressing privacy issues, making sure everything is fair and transparent, and eliminating prejudices. The suggested method may progress robotics and human-robot interaction as well as public safety by proactively addressing these ethical issues and managing crowds effectively. As this technology spreads, the

ethical and responsible use of face recognition technology in numerous applications and sectors will be modeled after its effective implementation (Chen et.al., 2021).

CHAPTER THREE

DEEP LEARNING TOPOLOGIES

CHAPTER THREE: DEEP LEARNING TOPOLOGIES:

3.1. DEEP Learning topologies

Successful applications in areas such as computer vision, natural language processing, and speech recognition have helped propel deep learning into the spotlight in recent years. Machine learning has a subfield called deep learning, which makes use of neural networks to acquire knowledge. Connected nodes, or neurons, in a neural network learn to identify patterns in data. Due to its success in autonomously learning hierarchical representations from raw data, deep learning has gained widespread use. This eliminates the need for human intervention in feature extraction, allowing deep learning models to autonomously learn features from the data. Accuracy in many machine learning tasks has been greatly enhanced by deep learning models' automated feature learning capacity. When it comes to object recognition, picture classification, and segmentation, deep learning has produced state-of-the-art results, which has completely changed the face of computer vision.

Most deep learning architectures for computer vision problems are convolutional neural networks (CNNs). For feature extraction from input pictures, CNNs use convolutional layers, whereas fully connected layers handle classification and regression. CNNs' success has spawned a number of other architectures, including those with state-of-the-art performance on a number of benchmarks, such as ResNet, Inception, and DenseNet. Natural language processing is another area where deep learning has shown very useful, alongside computer vision. Modeling sequential data like text, voice, and time series has been done using recurrent neural networks (RNNs) and its derivatives like long short-term memory (LSTM) and gated recurrent unit (GRU). Due to RNNs' superior accuracy in modeling sequential data, applications including voice recognition, machine translation, and sentiment analysis have benefited greatly. There are still obstacles and constraints to deep learning, despite its achievements. The need for a big volume of labeled data is a significant obstacle since it may be time-consuming and costly to collect. Another difficulty is the need for powerful computers due to the computational complexity of deep learning models.

Overfitting is another possible pitfall, when the model learns too much from the training data and cannot generalize to new data. Several approaches have been suggested as potential answers to these problems. Transfer learning is a method for reducing the amount of labeled data needed for a new task by using a previously trained deep learning

model as a baseline. Data augmentation is another method for expanding the training set by creating new data from the current data.

Additionally, work is being done to optimize deep learning models for use on mobile phones and embedded systems, which have limited power resources. Deep learning is a fast-developing topic that has increased the precision of many machine learning tasks, most notably in the areas of computer vision and natural language processing. In the fields of computer vision and NLP, respectively, CNNs and RNNs are the most popular deep learning architectures. The computational complexity of deep learning models and the need for a significant quantity of labeled data remain obstacles despite deep learning's progress. More efficient deep learning models and solutions to solve these issues are the subject of continuing research, which should lead to progress (Kumar et.al., 2022).

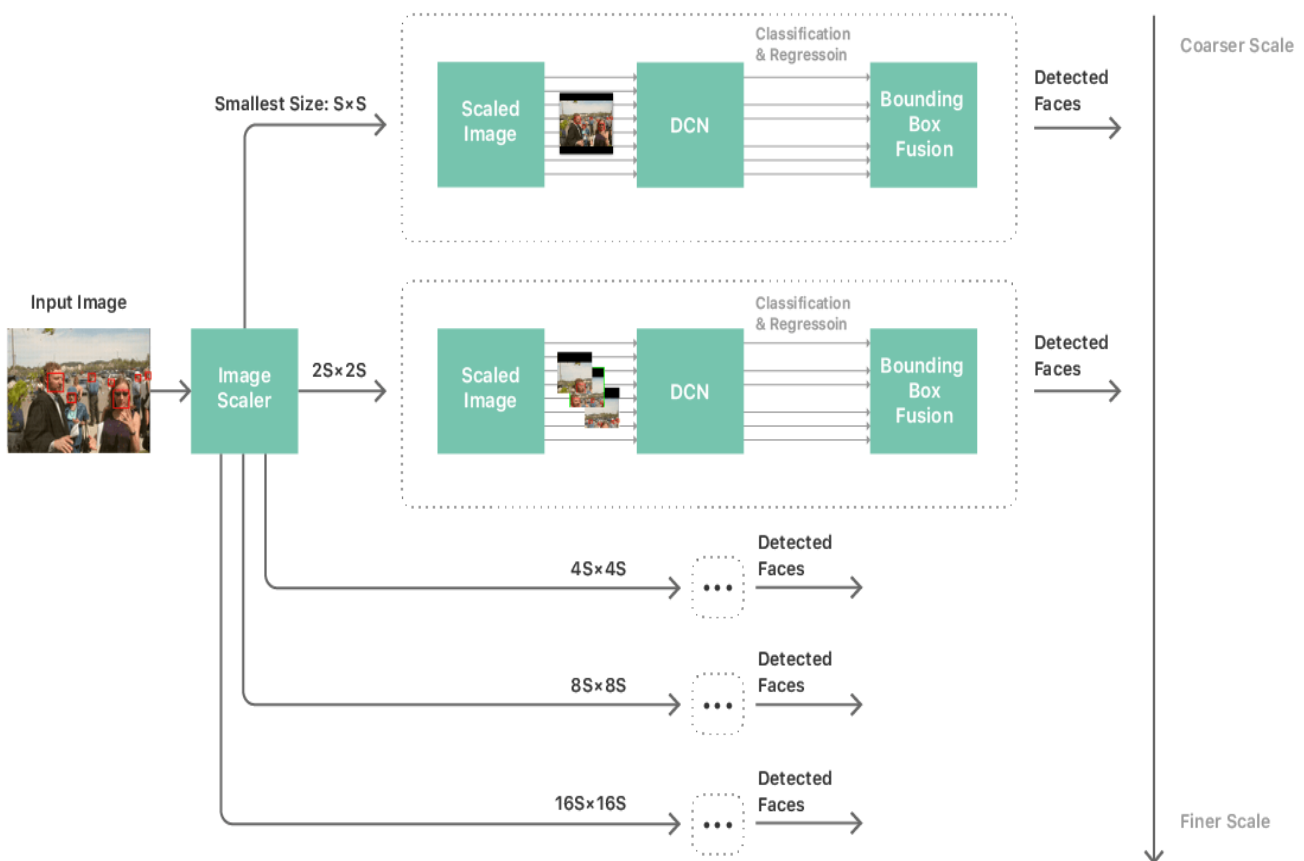


Figure 14: An On-device Deep Neural Network for Face Detection (Pierre, 2017).

3.2. Classification of DL approaches:

Supervised learning, unsupervised learning, and reinforcement learning are the three broad categories into which deep learning methods fall. In supervised learning, a model is trained using data that has been labelled so that it may learn to map inputs to desired results. In other words, the model is taught to generalize a mapping between inputs and outputs. In image classification, for instance, the input data is a picture itself, while the label or class assigned to the image is the output data. The purpose of training the model is to teach it to accurately identify new photos based on the labels already applied to a large dataset. Supervised learning excels in a variety of tasks, including image classification, object identification, and voice recognition, and this is one of its main benefits. However, obtaining a large amount of labelled data can be time-consuming and costly, which is one of the supervised learning's main limitations. Also, the model may struggle to adapt to new inputs that are very different from the training data because it can only generalize based on the input-output pairs it has seen during training. Training a model using unlabeled data allows the model to learn how to extract relevant features or representations from the data without the need for human intervention. In other words, the model is taught to find structure in a given data set by being exposed to that collection. The purpose of clustering, for instance, is to find groups of related data points even when the labels are unknown. To find patterns or structures in the data, the model is trained on an unlabeled dataset. Unsupervised learning has the potential benefit of discovering meaningful features or representations in the data without the requirement for human-engineered features. When the data is complicated or has a lot of dimensions, this may be quite helpful. However, without labels to compare the results to, assessing the effectiveness of an unsupervised learning model is challenging. It's also possible that the model may find structure or trends in the data that are irrelevant to the current endeavor.

Training a model to make choices in response to rewards and punishments is what reinforcement learning is all about. The model is exposed to a setting and taught to respond in ways that optimize a reward signal. The objective of gaming, for instance, is to master the art of game-winning strategy. Each time the model acts, it receives a reward signal, and the idea is for it to figure out how to get the biggest possible payout. Reinforcement learning has the potential to learn how to act in response to changing

circumstances. In robotics, for instance, the objective is to figure out how to instruct a robot to carry out a difficult job in an increasingly complicated setting.

However, creating a reward signal that precisely reflects the desired behavior is challenging, and this is one of reinforcement learning's main drawbacks. Furthermore, the model may learn to game the reward system to get high rewards without exhibiting the intended behavior. In conclusion, there are three main types of deep learning methods: supervised, unsupervised, and reinforcement. There are benefits and drawbacks to every kind of method, so picking the right one for any given job is tricky. Tasks well-suited to supervised learning include those with readily available labelled data; those with complicated or high-dimensional data; and those requiring decision-making based on rewards and punishments, the domain of reinforcement learning (Kumar et.al., 2022).

3.3 Types of DL networks

3.3.1 Convolutional Neural Networks (CNNs)

In recent years, deep learning networks like Convolutional Neural Networks (CNNs) have gained widespread adoption for use in computer vision applications. CNNs excel in image processing jobs because they can automatically extract characteristics that are important to the task from the input pictures. The convolutional layer is crucial to CNN since it is responsible for applying a series of trainable filters to the input picture. In order to calculate the dot product between the input picture and each filter, each filter is a tiny, weighted matrix that is dragged over the image. The result of this process is a feature map, which draws attention to certain patterns within the picture. Features like edges, corners, and blobs may be extracted using a combination of filters. CNNs employ pooling layers to further decrease the spatial resolution of the feature maps after the convolutional layer. When applied to feature maps, pooling layers combine the contributions of adjacent pixels to create a more compact map with lower spatial resolution. This makes the network less sensitive to subtle shifts in the input picture. For classification or regression, CNNs commonly use one or more fully connected layers after the convolutional and pooling layers. To generate a probability distribution across the classes, the fully connected layers require as input the simplified feature maps. One difficulty in training CNNs is overfitting, which occurs when the network excels on the

training data but fails miserably on the test data. This may arise if there are too many network parameters for the amount of available training data. Dropout and weight decay are two regularization strategies that may be used to avoid overfitting. The selection of hyperparameters, such as the number of layers, filter size, and learning rate, presents still another difficulty.

Depending on the task and dataset, the ideal values for these hyperparameters might have a profound impact on the network's performance. Despite these obstacles, CNNs have produced state-of-the-art results on a number of benchmarks, including the ImageNet dataset, which contains over a million images sorted into a thousand categories. CNNs have also shown promising results in the areas of object identification and semantic segmentation. CNNs' performance may be enhanced with the help of a number of other suggested designs, including ResNet, Inception, and DenseNet. By introducing skip connections across layers, ResNet (Residual Network) CNNs may learn residual mappings.

For really deep networks, this has been proven to significantly boost accuracy. To extract features at varying scales, CNNs like the Inception model employ various filter sizes in tandem. DenseNets (Densely Connected Networks) are a special kind of CNN that recycles features from prior layers thanks to the usage of skip connections between all of the layers. Uses for CNNs range from driverless vehicles to medical imaging to remote sensing. One autonomous driving use for CNNs is in the detection of objects and pedestrians in pictures and video feeds. They may also be used to help diagnose patients and organize their treatments by analyzing medical images such as MRI and CT scans. Classifying and segmenting satellite pictures using CNNs is a helpful use of remote sensing for applications like land use and land cover mapping. Image processing tasks including classification, object recognition, and semantic segmentation have seen a rise in the use of Convolutional Neural Networks (CNNs), a form of deep learning network.

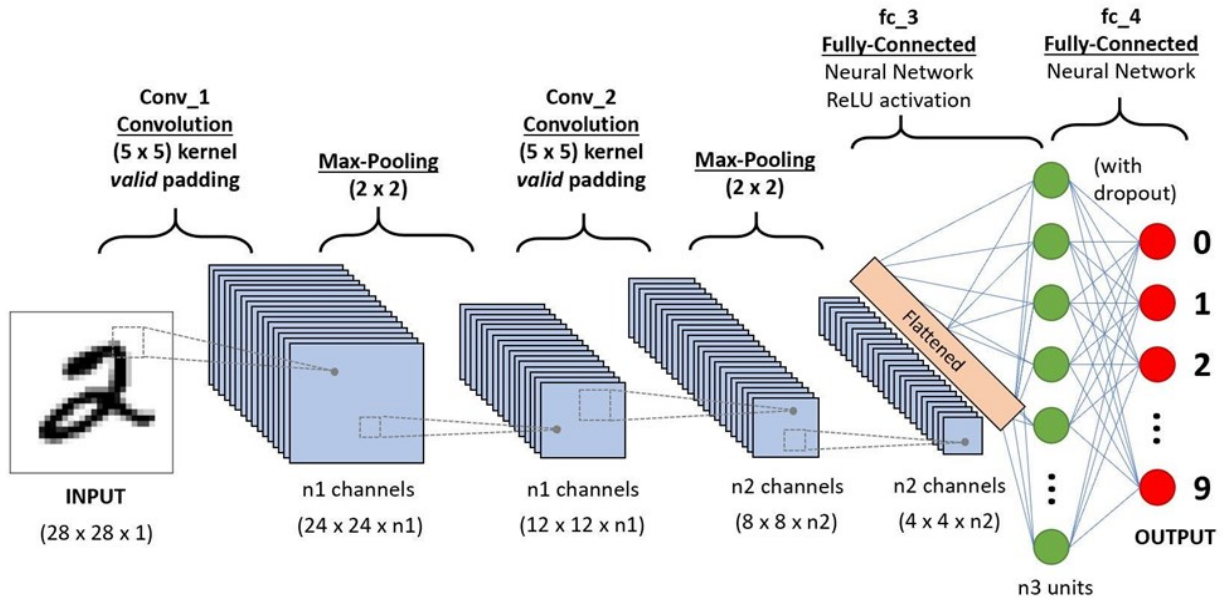


Figure 15: A Comprehensive Guide to Convolutional Neural Networks (Saha, 2018).

Justification:

Convolutional Neural Networks (CNNs) have the capacity to automatically learn hierarchical properties from pictures, making its use for crowd identification a perfect match. CNNs are excellent at collecting local structures and spatial patterns, which are essential for identifying people in crowded areas. CNNs automatically identify pertinent information, such as body forms and motion patterns, by using convolutional and pooling layers. CNNs are a strong tool for accurately recognizing and evaluating crowd dynamics, which is crucial for crowd management, public safety, and urban planning applications. This enables crowd detection models to attain improved accuracy and resilience.

The Convolutional Neural Networks Formula:

CNN is made up of three different kinds of layers: convolutional, pooling, and fully connected (FC). A CNN architecture is created when these layers are stacked (Gurucharan, 2022). The used layers in our model are visualized in Figure (16) and a code example is provided in figure (22). Layers are discussed below with steps:

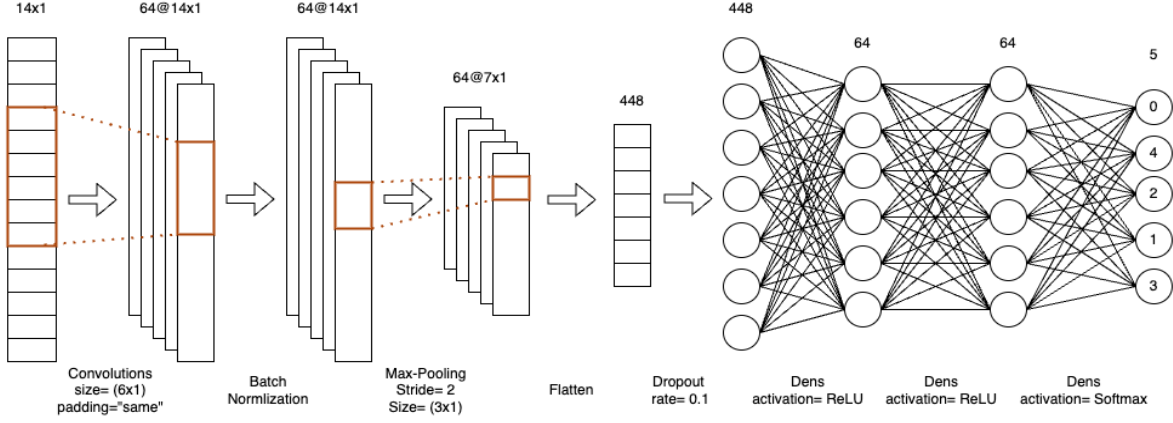


Figure 16: CNN layers visualization.

1. 1D Convolutional layer:

The foundational component of the CNN is the convolution layer. It carries most of the computational load on the network (Mishra, 2020).

The first layer utilized to extract the different features from the input photos is this one. The input image and a filter of a specific size ($M \times M$) are convolution mathematically in this layer; in our scenario, the input image will be of size (14x1). The dot product between the filter and the portions of the input image regarding the size of the filter ($M \times M$), which is in our example (6x1), is taken by swiping the filter over the input image (Gurucharan, 2022):

$$W_{out} = \frac{W-F+P}{S} + 1 \quad (1)$$

From Formula (1), we can calculate the size of the output volume if we have an input of size ($W \times W \times D$), D_{out} number of kernels, spatial/kernel size F (a

single integer or set of single integers specifying the 1D convolution window's length), stride S (the kernel's movable size), and amount of padding P (When you pad, the output size is equal to the input size because of the same convolution) (Nkomo, 2022; Mishra, 2020).

In our case the input shape is $(None, 14, 1)$ so $W = 14$, and the kernel size is 6 so $F = 6$, and padding is 'same' that indicates we want the output to be same size of our input, so $W = W_{out} = 14$, and S is 1 by default.

$$output = (None, 14, 64) \quad (2)$$

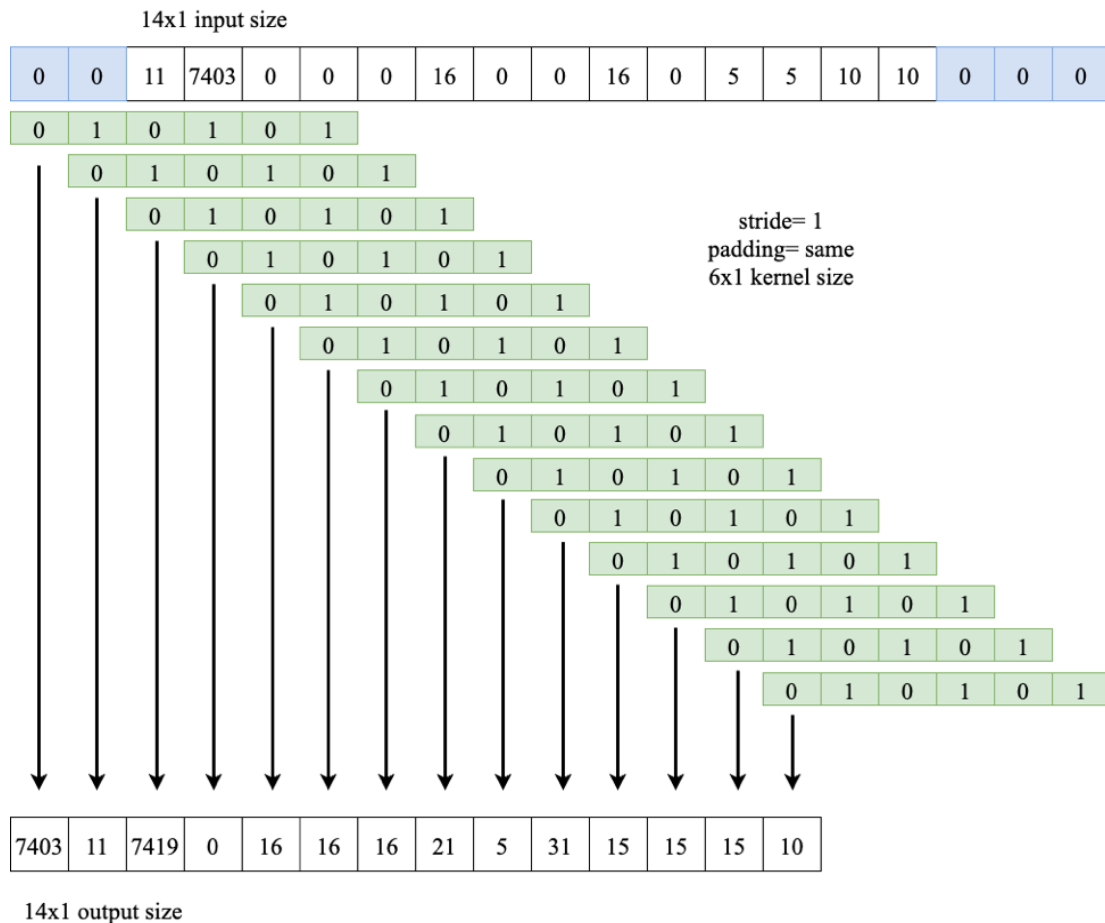


Figure 17: illustrates the steps of Convolution layer.

2. Batch Normalization Layer

Instead of conducting the normalization in the raw data, batch normalization is done between neural network layers. Instead of utilizing the complete data set, mini batches are used. It accelerates training and uses faster learning rates to promote learning (Riva, 2022). The output dimension is the same as the input dimension.

$$output = (None, 14, 64) \quad (3)$$

The theory behind the normalization layer is demonstrated in the proceeding equations, (Peccia, 2019). The batch mean is calculated with the batch variance, Formula (4), and Formula (5):

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (4)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (5)$$

Next step is applying normalization with Formula (6):

$$\bar{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (6)$$

The last step is applying scale and shift to get the output of the layer with Formula (7):

$$y_i = \gamma \bar{x}_i + \beta \quad (7)$$

3. 1D Max Pooling Layer

By obtaining a summary statistic of the surrounding outputs, the pooling layer substitutes for the network's output at specific locations. The representation's spatial size is lowered as a result, which lowers the calculation and weighting requirements. Every slice of the representation that the pooling operation touches is treated separately (Mishra, 2020). A weighted average based on the distance from the central pixel is one of the pooling functions, along with the

average of the rectangular neighborhood and the L2 norm of the rectangle neighborhood. However, max pooling figure(18), which reflects the highest output from the neighborhood, is the most well-liked method (Mishra, 2020).

The greatest component in Max Pooling is derived from a feature map. The elements inside a predetermined sized Picture slice are averaged using the average pooling technique. Sum Pooling calculates the aggregate sum of the components in the predefined section. Convolutional Layer and Fully Connected Layer are typically connected by the Pooling Layer (Gurucharan, 2022).

$$W_{out} = \frac{W-F}{S} + 1 \quad (8)$$

The output volume can be calculated using Formula (8) if we have an activation map of size $(W \times W \times D)$, a pooling kernel of spatial size F, and stride S (Mishra, 2020), in our case here the output of the previous layer is to be used as the input in this layer, so $W=14$, $F=3$, and $S=2$. However, using the "same" padding option produces the following output shape: ***output shape*** = $\frac{\text{input shape}}{\text{strides}}$

$$W_{out} = \frac{14}{2} = 7 \quad (9)$$

$$\text{output} = (\text{None}, 7, 64)$$

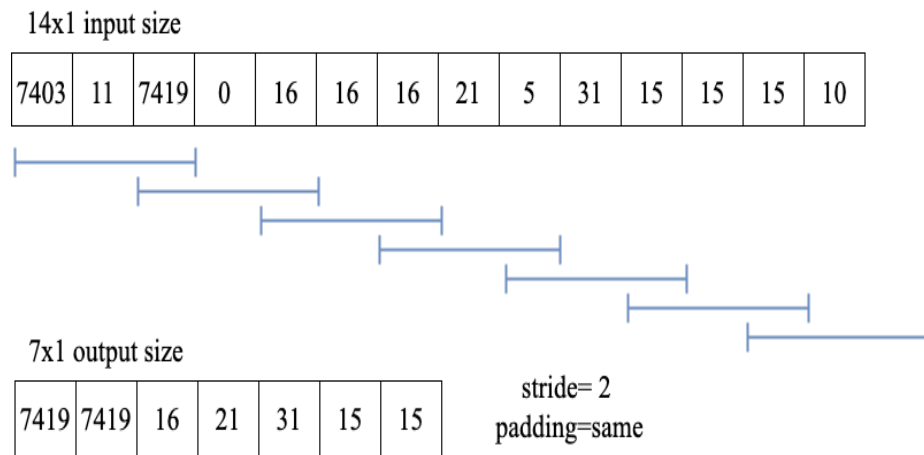


Figure 18: Steps of Maximum Pooling layer.

4. Flatten Layer

We have flattened and completely linked layers at the end of CNN. It must take the form of a one-dimensional linear vector. Direct inputs are not allowed to be rectangular or cubical in form. As a result, we need layers that are fully connected and flattened. When data is converted into a 1-dimensional array for input into the next layer, it is flattened. We flatten the output of the convolutional layer to create a feature vector. It is also linked to the previous classification model, which is known as a fully connected layer, as shown in Figure (21). In other words, we connect the last layer to a single line that contains all the pixel data (Jeong, 2019).

By taking the previous output as an input for this layer, we got the following output:

$$output = (None, 7 \times 64) = (None, 448) \quad (10)$$

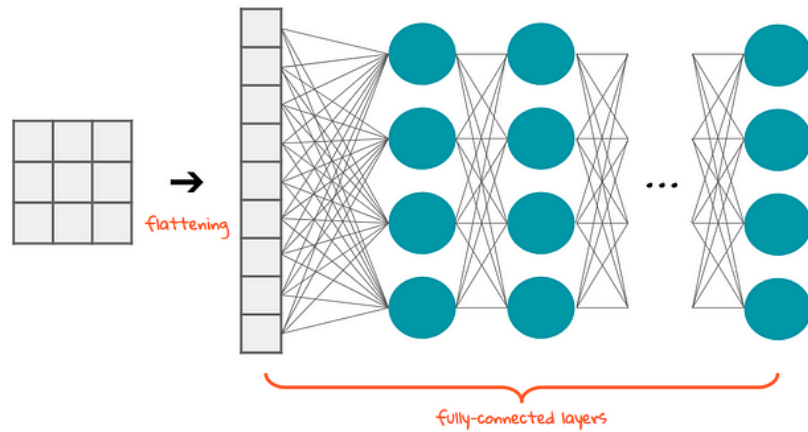


Figure 19: Flatten and fully connected layers, (Jeong, 2019).

5. Dropout layer

A dropout layer is a particular kind of regularization method that is used to avoid overfitting and enhance generalization error in machine learning and neural network models. A dropout layer randomly removes a part of its input units during each training cycle. As a result, because it cannot rely on a single input unit to always be available, the model is compelled to acquire more reliable and

generalizable properties. Any layer in a neural network model can have dropout layers added, and a hyperparameter can be used to adjust how many units will really drop out of the network (Brownlee, 2019).

$$\text{output} = (\text{None}, 448) \quad (11)$$

6. Dens Layer

Each neuron in a dense layer that is connected to the layer above it in a neural network signifies that layer is connected to the layer above it. This layer is the one that is most typically used in artificial neural network networks. Each neuron in the layer above delivers signals to the neurons in the dense layer, which multiply matrices and vectors, in a model. During matrix vector multiplication, the row vector of the output from the preceding layers is identical to the column vector of the dense layer. To multiply matrices using vectors, the row vector must have the same number of columns as the column vector (Verma, 2023).

One of the most fundamental and important parameters of the Keras dense layer, which determines the size of the dense layer's output, is units. Since it reflects the output vector's dimensions, it must be a positive integer; in this layer, the units are 64. The activation function in neural networks is a function that transforms the input values of the neurons. It adds nonlinearity to neural network networks so that the networks can figure out how input and output values relate to one another. In this layer, we are using the ReLU activation function (Verma, 2023).

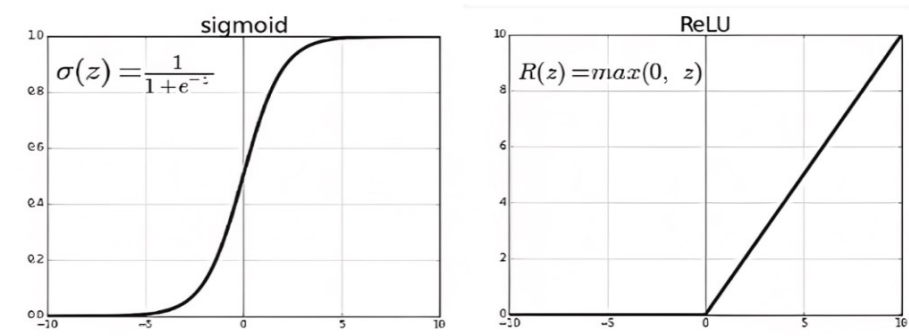


Figure 20: Sigmoid and ReLU functions, (Sharma, 2022).

Nowadays, the ReLU is the activation function that is employed the most globally. Figure (20) illustrates how the ReLU is only partially rectified (from bottom). When z is less than zero, $f(z)$ equals zero, and when z is more than or equal to zero, $f(z)$ equals z . Hence, the Range is $[0$ to infinite) (Sharma, 2022).

$$output = (None, 64) \quad (12)$$

7. Dens Layer

$$output = (None, 64) \quad (13)$$

8. Dens Layer

The neural network's raw outputs are input into the SoftMax activation function, which then produces a vector of probability scores (Priya, n.d.).

- The neural network's raw outputs are represented by the vector z .
- 2.718 is the value of e .
- The expected probability that the test input will belong to class i is represented by the i th entry in the SoftMax output vector $\text{SoftMax}(z)$.

The function of SoftMax is represented below:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \quad (14)$$

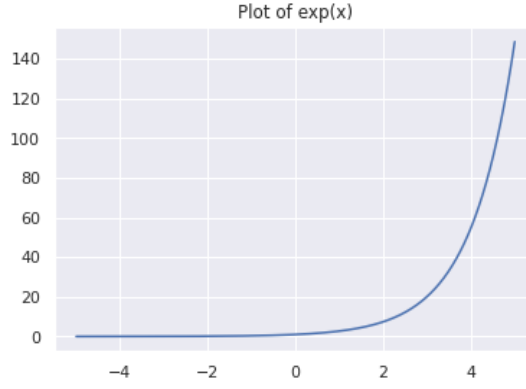


Figure 21: Plot of exp, (Priya, n.d.).

It's clear from the plot of e^x that e^x is always a positive value, regardless of whether the input x is positive, negative, or zero, Figure (21), (Priya, n.d.).

As we have five or more classes—and the valid indices are 0, 1, 2, 3, and 4, $N = 5$ in our example. By assuming the given neural network's raw outputs, represented by the vector $z = [0.3, 1.15, -0.7, 5.2, 1.4]$.

To elaborate more on how the calculation process is done, the following calculations are done with referral to (Priya, n.d.)

1. Determine each entry's exponent.
2. Subtract the outcome of step 1 from the total exponents of all the entries.

$$\frac{e^{0.3}}{e^{0.3} + e^{1.15} + e^{-0.7} + e^{5.2} + e^{1.4}} = 0.00709$$

$$\frac{e^{1.15}}{e^{0.3} + e^{1.15} + e^{-0.7} + e^{5.2} + e^{1.4}} = 0.01659$$

$$\frac{e^{-0.7}}{e^{0.3} + e^{1.15} + e^{-0.7} + e^{5.2} + e^{1.4}} = 0.00260$$

$$\frac{e^{5.2}}{e^{0.3} + e^{1.15} + e^{-0.7} + e^{5.2} + e^{1.4}} = 0.95239$$

$$\frac{e^{1.4}}{e^{0.3} + e^{1.15} + e^{-0.7} + e^{5.2} + e^{1.4}} = 0.02130$$

$$\begin{bmatrix} 0.3 \\ 1.15 \\ -0.7 \\ 5.2 \\ 1.4 \end{bmatrix} \rightarrow \text{Softmax} \rightarrow \begin{bmatrix} 0.00709 \\ 0.01659 \\ 0.00260 \\ 0.95239 \\ 0.02130 \end{bmatrix} \quad (15)$$

The following conclusions can be drawn from the SoftMax output above: The greatest element in the SoftMax output vector, 0.952, is represented by the maximum value of 5.2 in the vector z of raw outputs. SoftMax's application thus maintains the relative ranking of scores. The SoftMax output vector's entries are all between 0 and 1. See how the entries of the SoftMax output aggregate to 1 in a multiclass classification problem where the classes are mutually exclusive (Priya, n.d.): $0.0071 + 0.0166 + 0.00260 + 0.9524 + 0.02130 = 1$.

By checking the SoftMax output for our case as a following step.

The greatest value in the vector $\text{softmax}(z) = [0.0071, 0.0166, 0.00260, 0.9524, 0.02130]$ is 0.9524. This indicates that there is a 95.24% likelihood that the provided input is a member of class 4, or the fake3 class.

The output dimension of this layer is:

$\text{output} = (\text{None}, 5)$

```
# making the deep learning function
def model():
    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=6, activation='relu',
                     padding='same', input_shape=(14, 1)))
    model.add(BatchNormalization())
    model.add(MaxPooling1D(pool_size=(3), strides=2, padding='same'))
    model.add(Flatten())
    model.add(Dropout(0.1))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(5, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', met-
rics=['accuracy'])
    return model
```

Figure 22: code example of how to construct CNN.

Gabor Filter for Crowd Detection of Edges and Textures:

A Gabor filter is applied for detection of edges and texture changes, as has been reported in literature. A Gabor filter is a linear filter exploited for edge detection in image processing which is termed after Dennis Gabor. Gabor filter frequency and orientation representations are like those of human visual system, for texture representation and

discrimination. It has been found to be remarkably applicable. A sinusoidal plane wave has been modulating a 2D Gabor filter which is a Gaussian kernel function in the spatial domain. There are a number of specific parameters that impact the output of a Gabor filter. For the case of image processing and computer vision, Gabor filters are commonly employed in texture analysis, edge detection, feature extraction, disparity estimation, etc. In fact, Gabor filters are considered as special classes of bandpass filters, i.e., they allow a certain 'band' of frequencies and reject the others. Gabor filter is applied for: Used for detection of edges and texture changes. For the case of OpenCV, this called using the following function (python):

cv2.getGaborKernel(ksize, sigma, theta, lambda, gamma, psi, ktype

complex:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left\{ -\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right\} \exp \left\{ i \left(2\pi \frac{x'}{\lambda} + \psi \right) \right\} \quad (16)$$

real:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left\{ -\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right\} \cos \left(2\pi \frac{x'}{\lambda} + \psi \right) \quad (17)$$

imagery:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left\{ -\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right\} \sin \left(2\pi \frac{x'}{\lambda} + \psi \right) \quad (18)$$

$$x' = x(\cos(\theta) + y(\sin(\theta))$$

$$y' = -x(\sin(\theta) + y(\cos(\theta)))$$

$$g(x, y; \lambda, \theta; \sigma, \gamma) = \exp \left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \exp \left(i \left(2\pi \frac{x'}{\lambda} + \psi \right) \right) \quad (19)$$

Gabor Filter, typical values and parameters.

Kernal size: *kernal size* = (7, $\sigma = 2$, $\theta = 0$, $\lambda = 5$, $\gamma = 10$, $\phi = 0$)

Kernal size: *kernal size* = (50, $\sigma = 10$, $\theta = 0$, $\lambda = 30$, $\gamma = 1$, $\phi = 0$)

The geometry-adaptive kernel is defined as:

$$F(x) = \sum_{i=0}^N \delta(x - x_i) \times G_{\sigma_i}(x), \text{ with } \sigma_i = \beta \bar{d}_i \quad (20)$$

The evaluation metrics were chosen to be the MAE and MSE estimates, calculated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}| \quad (21)$$

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}|^2} \quad (22)$$

where N is the number of images in one test sequence and CGT

i is the ground truth of counting. C_i represents the estimated count which is defined as follows:

$$C_i = \sum_{l=1}^L \sum_{w=1}^W z_l, W \quad (23)$$

The used loss function:

$$loos = (S_g - S_a)^2 + A * (P_{max,g} - P_{max,a})^2 + B * (P_{min,g} - P_{min,a})^2 \quad (24)$$

here

S_g : sum over pixel values of the generated density map.

S_a : sum over pixel values of the ground truth density.

$P_{max,g}$ maximum pixel value in the generated density map

$P_{max,a}$ maximum pixel value in the ground truth density map

$P_{min,g}$ minimum pixel value in the generated density map

$P_{min,a}$ minimum pixel value in the ground truth density map

Representation of cluster of pixel values in the original map by a single pixel of larger value in the learnt map.

- Need to enforce the learnt pixel values to be in a similar range - similar max-min values
- Enforced using the above loss function, the weights A and B highlight the relative importance of density map accuracy to crowd count accuracy
- Implemented with the optimal models obtained via previous searches

5 Density Map Generation

Density mapping is simply a way to show where points (or lines) may be concentrated in a given area. A density map is important to visualize the concentration of particular object in a given image (FOV). For example, a density map finds application in census and statistics to visualize population density in a city/country. Similarly, to visualize the forest cover in a given region and in various other applications. Given the locations of the person (by denoting the location of head), a Gaussian kernel is used to convolve with the delta function like 2-D array to obtain the density maps. Following the method of generating density maps, we use the geometry-adaptive kernels to tackle the highly congested scenes. By blurring each head annotation using a Gaussian kernel (which is normalized to 1), we generate the ground truth considering the spatial distribution of all images from each dataset. The geometry-adaptive kernel is defined as:

$$F(x) = \sum_{i=1}^N \delta(x - x_i) \times G_{\sigma_i}(x), \dots \text{ with } \sigma_i = \beta \bar{d}_i \quad (25)$$

3.3.2. Recurrent Neural Networks (RNNs)

Time series, natural language, and voice are only a few examples of sequential data that may be processed by Recurrent Neural Networks (RNNs), a form of deep learning network. RNNs can capture the temporal relationships in the input sequence because they keep a hidden state that is updated at each time step, whereas feedforward networks analyze each input individually. The recurrent layer is important to a recurrent neural network and is responsible for outputting a new hidden state based on the current input and the prior hidden state. This enables the network to remember previous inputs and

utilize that knowledge to better understand and interpret the present input. Different recurrent cell architectures, such as the Gated Recurrent Unit (GRU), the Long Short-Term Memory (LSTM), and the vanilla RNN, can be used to realize the recurrent layer.

The vanilla recurrent neural network (RNN) is the most basic kind of RNN, computing the hidden state as a nonlinear function of the current input and the previously hidden state. However, vanilla RNN has trouble learning long-term dependencies due to the vanishing gradient problem, in which the gradients become very small. Long short-term memory (LSTM) cells were developed as a solution to the vanishing gradient issue. The network may remember or forget data selectively thanks to the LSTM cell's gating mechanism, which regulates the information flow into and out of the memory cell. The GRU is a condensed form of the LSTM cell, with only a single update gate and a reset gate instead of two. This decreases the network's parameter size and facilitates quicker training times. In order to accomplish the classification or regression operation, RNNs employ not only the recurrent layer but also one or more fully linked layers. In order to generate a probability distribution across the classes, the fully connected layers need the final hidden state as input.

RNNs may be used in a number of contexts, including NLP, SR, and even time series analysis. In natural language processing, RNNs have found applications in areas such as language modeling, machine translation, and sentiment analysis. When the input is a series of audio frames, they may be utilized for tasks like speech recognition. RNNs are useful for making forward-looking predictions in time series analysis.

When training RNNs, vanishing or exploding gradients can occur, causing the gradients to become either extremely small or extremely large, making it difficult for the network to learn. Too many recurrent layers or an excessive learning rate might cause this. This problem may be fixed using a number of methods, including gradient trimming and weight initialization.

The selection of hyperparameters, such as the number of recurrent layers, hidden state size, and learning rate, also presents a significant issue. Depending on the task and

dataset, the ideal values for these hyperparameters might have a profound impact on the network's performance.

Bidirectional RNNs, Encoder-Decoder Networks, and Attention Mechanism Networks are only few of the RNN designs that have been developed to boost their effectiveness. Inputs to a bidirectional RNN are processed in both ways, enabling the network to learn from both the past and the future. To encode the input sequence into a fixed-length vector and decode it back into a sequence, Encoder-Decoder Networks use two independent RNNs. In certain applications, like machine translation, an extra mechanism is used to direct the network's attention to the most important bits of the input sequence (Yu et.al., 2021).

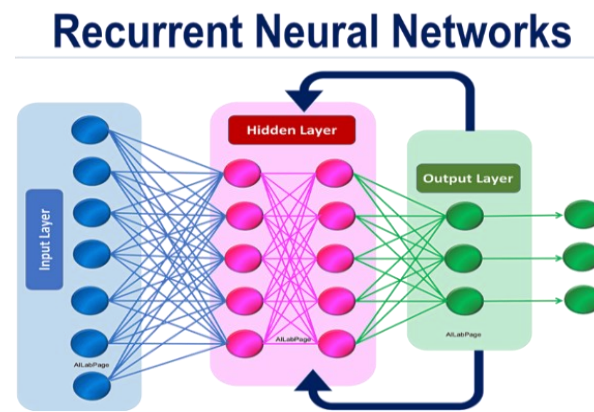


Figure 23: Recurrent Neural Networks — Unlocking the Secrets of Sequential Data (kondreddy, 2023).

Formula:

Certainly, Recurrent Neural Networks (RNNs) are designed to work with sequential data, such as time series or sequences of text. Here are some formulas and explanations for key components of RNNs:

1. Hidden State Update:

The hidden state at each time step is updated based on the input at that time step and the previous hidden state. This forms the core operation of an RNN.

Given:

$$h_t = f(x_t, h_t - 1) \quad (26)$$

2. Output at Time Step:

The output at each time step y_t can be computed from the hidden state h_t using another function g :

$$y_t = g \tag{27}$$

3.3.3. Generative Adversarial Networks (GANs)

The deep learning network known as a Generative Adversarial Network (GAN) is capable of producing new data that is very comparable to the training data. Two neural networks, called the generator network and the discriminator network, make up a GAN. In contrast to the discriminator network, which takes real data from the training set as input and returns a probability that the input is real, the generator network generates new data using a random noise vector as input. The generator network and the discriminator network play a two-player game during training, with the generator network attempting to produce data that fools the discriminator network and the discriminator network attempting to discern between actual and generated data.

Backpropagation and gradient descent are used to train the two networks, and the training process continues until the data generated by the generator network cannot be distinguished from the actual data by the discriminator network. GANs' most novel feature is the adversarial loss function, which minimizes the difference between the probability distributions of the actual and created data to motivate the generator network to generate data that is similar to the real data. Together with the discriminator network, this loss function is optimized, and it is defined as the negative log-likelihood of the network correctly classifying the generated data as real. Image/video synthesis, text creation, and even music production is just some of the many areas where GANs have found use. GANs may learn the statistical aspects of training photos and then produce new images that match these qualities, allowing them to be used to generate realistic images of faces, animals, and landscapes. The GAN may learn the patterns in the training audio and produce new music that follows these patterns, or it can be used to make realistic films by creating several frames successively. Training GANs may be difficult because of a phenomenon known as "mode collapse," in which the generator

network only learns to create a small fraction of the training data. This may occur if the generator network is unable to provide enough varied data or if the discriminator network is very robust. Mini-batch discrimination, feature matching, and Wasserstein generative adversarial networks are only some of the methods that may be utilized to solve this problem.

The selection of hyperparameters, including the learning rate, the number of layers, and the size of the networks, is an additional difficulty. Depending on the task and dataset, the ideal settings for these hyperparameters may have a noticeable impact on the GANs' performance. Conditional GANs, CycleGANs, and StyleGANs are only some of the GAN designs that have been developed to boost their effectiveness. Conditional GANs can create data based on a given label or description. Without requiring paired training data, CycleGANs may learn to translate across unrelated domains, such as horse and zebra pictures. StyleGANs are trained to adjust aesthetic features like lighting, stance, and colour in the produced photos. To summarize, Generative Adversarial Networks (GANs) are a specific sort of deep learning network that can produce new data that is comparable to the training data by learning the statistical features of the data and creating new data that matches these qualities. Image and video synthesis, word creation, and music production are just a few examples of where GANs have been shown to be effective. Although GAN training can be difficult, many approaches have been proposed to help boost their performance in various areas (Yu et.al., 2021).

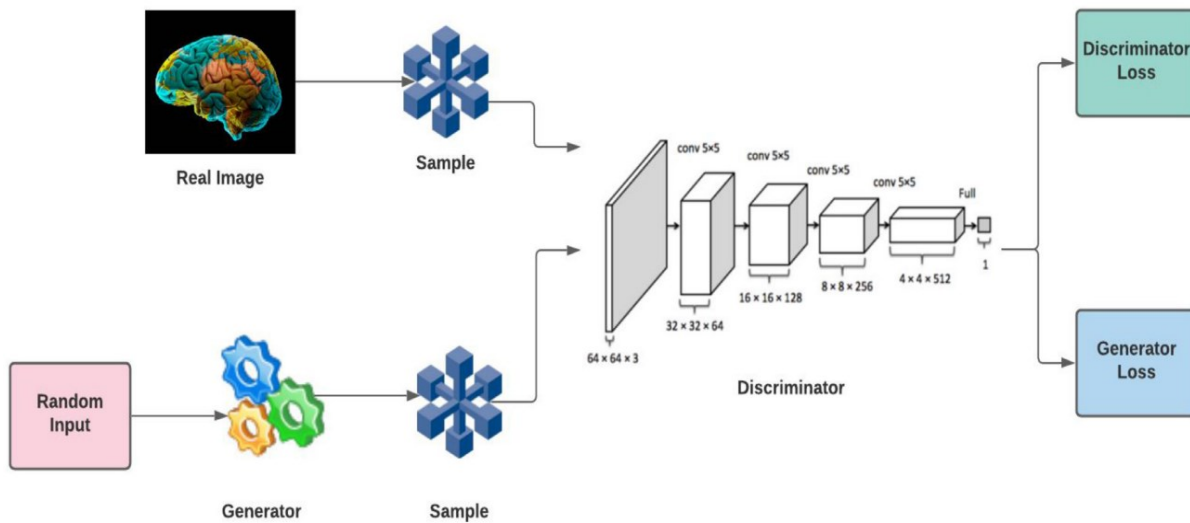


Figure 24: Generative Adversarial Networks (Agrawal, 2021).

3.3.4 Autoencoders

Autoencoders are a form of neural network that can encode data into a lower-dimensional space and then decode it back into the original space to learn a compressed version of the input data. Encoder networks and decoder networks are the two primary building blocks of every autoencoder. With the raw data as input, the encoder network maps it to a compressed representation in the latent space; with the compressed representation as input, the decoder network maps back to the original data space. The autoencoder's job is to develop a compact representation of the data that retains the essential characteristics while excluding noise and superfluous details. Denoising images and signals, compressing data, and reducing the number of dimensions is only some of the many applications for autoencoders. As an example, autoencoders may be used to compress enormous datasets into a lower-dimensional representation that can be stored and processed more effectively or to remove noise from photos by training on noisy images and recreating the clean ones. Autoencoders' capacity to acquire relevant features unsupervised, without labelled training data, is a major advantage. This is especially helpful for jobs where labelled data is either hard to get by or prohibitively costly to collect. By learning a compressed representation that may be utilized as input to the downstream network, autoencoders can also be employed as a pre-training phase for other neural networks. Basic autoencoders, denoising autoencoders, VAEs, and adversarial autoencoders are only a few of the autoencoders that can be found in the

literature. Denoising autoencoders are trained on noisy data and then used to reconstruct the clean data, eliminating the need for manual noise removal. The VAE is a generative model that can sample fresh data from a learned probability distribution across the latent space. To provide more realistic data, the adversarial autoencoder blends adversarial training with autoencoders.

Overfitting is a problem that may arise while training autoencoders because it causes the model to become too specific to the training data. Overfitting may be avoided and autoencoder performance enhanced using several methods, including regularization, dropout, and early stopping. The selection of hyperparameters, including the learning rate, the number of layers, and the size of the networks, is an additional difficulty.

Autoencoder performance is very sensitive to these hyperparameters, and the ideal settings may change based on the problem at hand and the data available. Autoencoders have been used for some time, but in recent years they have been integrated with other deep learning methods like CNNs and RNNs to increase their performance and versatility. Image and video compression and denoising may benefit from convolutional autoencoders, while sequence prediction and creation can make use of recurrent autoencoders. To sum up, autoencoders are a specific sort of neural network that can learn a compressed representation of input data by encoding it into a lower-dimensional space and then decoding it back into the original space. Autoencoders have demonstrated promising results in image and signal denoising, data compression, and dimensionality reduction, just to name a few areas where they have been used. Although autoencoder training can be difficult, many methods and architectures have been proposed to alleviate this problem and boost the models' accuracy and efficiency (Pérez-Hernández et.al., 2020).

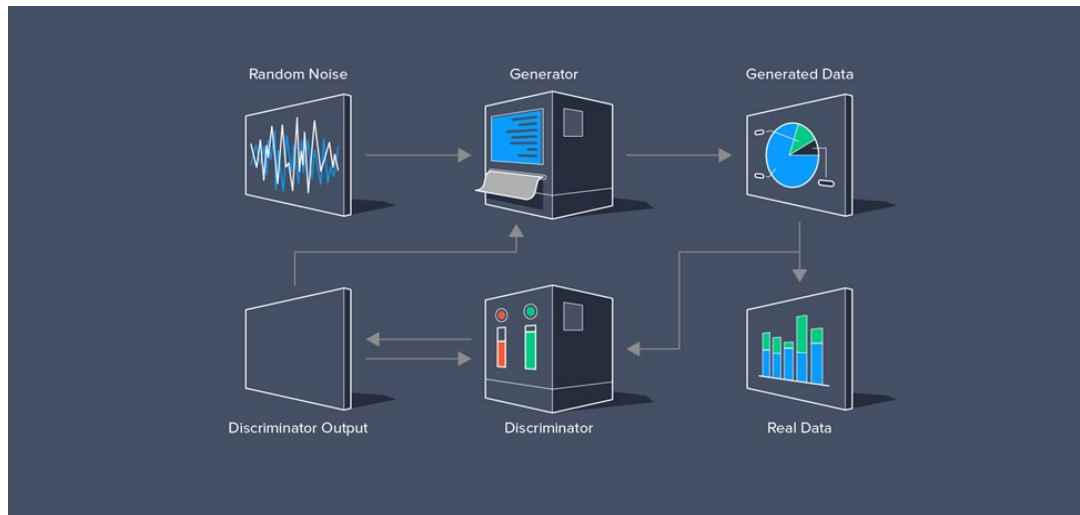


Figure 25: Auto decoder diagram (Pérez-Hernández et.al., 2020).

3.3.5 Deep Reinforcement Learning Networks (DRLNs)

With the use of deep learning and reinforcement learning, computers may be taught to make judgments via trial and error. This area of machine learning is known as deep reinforcement learning (DRL). In DRL, an agent learns to maximize a reward signal by carrying out a predetermined series of actions in response to environmental cues. The three essential parts of a DRL network are the agent, the environment, and the reward function. The agent takes in data about its surroundings and decides what to do next depending on what it has learned. The agent receives a reward signal from the environment that reflects the value of the agent's activities. The agent's policy is adjusted based on the reward function's mapping of the current state of the environment to a scalar reward value. Game playing, robotics, and autonomous driving are just some of the many applications for DRL networks. For instance, DRL networks have been used in the control of robotics and autonomous vehicles, as well as the development of agents that can defeat human professionals at complicated games like Go and chess. DRL networks' capacity to learn complicated behaviors in a data-driven way without the need for explicit programming is one of its primary advantages. DRL networks may learn to adjust their behavior to new surroundings and activities, and they can learn to make judgments based on sophisticated sensory inputs like pictures and sound. DRL networks come in a wide variety of flavors, including Q-learning, policy gradients, actor-critic, and deep deterministic policy gradients (DDPG). The Q-learning DRL approach does not need a model to train a value function that predicts future rewards based on current

actions. A subset of DRL algorithms known as "policy gradients" learns a policy by regressing past rewards by their predicted value. Using a dedicated network for estimating the value function and a second network for learning the policy, actor-critic approaches bridge the gap between value-based and policy-based learning. To make the learning process more consistent, DDPG employs a deterministic policy and a replay buffer.

The exploration-exploitation trade-off is a difficulty while training DRL networks. To maximize its anticipated benefit, the agent must both explore the environment in search of new tactics and behaviors that lead to greater rewards and use the information it has already learned. Exploration and exploitation may be balanced to boost the DRL networks' efficiency via the employment of several methods, such as epsilon-greedy exploration and Boltzmann exploration. Selecting appropriate values for hyperparameters like learning rate, discount factor, and network size is an additional difficulty. These hyperparameters may have a major impact on how well DRL networks operate, and their ideal values may change based on the circumstances. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two examples of deep learning approaches that have been integrated with DRL networks in recent years to increase the networks' performance and versatility.

While RNNs are better suited for modelling temporal dependencies in sequential data like text and speech, CNNs are better suited for processing visual inputs like images and videos. Finally, DRL networks allow computers to learn to make judgments via trial and error by combining the ideas of deep learning and reinforcement learning. DRL networks have shown promising outcomes in the realms of gaming, robotics, and autonomous driving, among others. There are several strategies and architectures that have been developed to increase DRL networks' performance and make their training easier (Pérez-Hernández et.al., 2020).

CHAPTER FOUR: DATASETS

CHAPTER FOUR: DATASETS

4.1. Introduction:

Image identification, object detection, and segmentation are just a few of the many computer vision tasks where deep learning-based algorithms have recently excelled. Crowd management, security, and transportation are just a few of the many areas that may benefit from accurate and reliable crowd counting and analysis. Estimating densities and evaluating the behavior of crowds requires identifying and following individuals or groups in crowded environments. Crowd behavior analysis may be used to keep tabs on crowd movements, spot any questionable activity, and make public spaces safer for everyone. However, occlusion, changes in illumination, and the presence of various objects present significant challenges to crowd counting and analysis in dense and complex scenes. Crowd counting and analysis techniques have historically relied on handmade features and heuristics, which might be vulnerable in complicated scenarios. Convolutional neural networks (CNNs), one kind of deep learning-based approach, has shown great promise in crowd analysis and counting in recent years. These methods have the capacity to handle complicated and varied scenarios because of their ability to automatically learn essential aspects from the data.

This thesis offers a deep learning-based scene and face recognition system for robotics crowd management because of the promising future of such approaches. The system uses convolutional neural network (CNN)-based algorithms to recognize and track people or groups in densely populated settings, estimate their densities, and evaluate their activities. Facial recognition and scene analysis, in particular, are important to the system because of their importance in crowd control and safety. Image preprocessing, feature extraction, and classification are only a few of the steps in the proposed system. Noise reduction, scaling, and normalizing are all part of the preprocessing phase for a picture. In order to extract useful characteristics from pictures, the feature extraction step employs convolutional neural network (CNN) based algorithms. In order to categorize people or groups in the scene and estimate their densities, the classification step employs a number of deep learning-based algorithms, such as support vector machines (SVMs). Facial recognition, which can identify individuals in a large group, is also a part of the proposed system. The faces in the scene are identified and located thanks to the facial recognition module's usage of convolutional neural network (CNN)

based methods like the YOLOv3 object identification network. A facial recognition network powered by deep learning is also a part of the system, and it may use the identified face areas to make identifications. The suggested approach is meant to address issues with the status quo of crowd counting and analysis, such as the need for manual feature engineering and heuristics. The technology also strives to enhance the precision and stability of crowd counting and analysis in high-density, high-complexity settings

Theses aim at achieving the following goals:

- *Create a scene and face identification system based on deep learning for use in robotics crowd control.*
- *Compare the results of the proposed system to those of current approaches and evaluate its performance using benchmark datasets.*
- *Examine how adjusting the proposed system's parameters and hyperparameters affects its functionality.*
- *Investigate the ways in which the suggested system might be used in a variety of real-world contexts, such as crowd control and safety.*

As a deep learning-based method for crowd control and safety, the suggested system has the potential to advance the area of crowd counting and analysis. Transportation, security, and public safety are just some of the areas that might benefit from the proposed system's scalability to include other uses, such as traffic monitoring and pedestrian tracking. In conclusion, this thesis suggests a scene and face recognition system based on deep learning for robotics crowd management. We present a system that uses convolutional neural network (CNN)-based approaches to recognize and track people or groups in crowded settings, estimate their densities, and evaluate their activities. The device also has a facial recognition module, which can locate and identify individuals in a large group photo. The goals of this thesis are to build and test the proposed system on benchmark datasets, examine the impact of various parameters and hyperparameters, and look into the system's possible practical applications (Chen et.al., 2021).

4.2.CMU action detection dataset

When it comes to human activity recognition in busy settings, the CMU action detection dataset is often used as a standard. It was developed by Carnegie Mellon University and may be downloaded for free for academic use. The collection consists of 10,024 videos shot at a broad variety of locations and events, including supermarkets, railway stations, public plazas, and sporting events. Actions as simple as walking and as complicated as sprinting and leaping and waving are all included in the CMU action detection dataset. This is a hard dataset for human action recognition due to the presence of occlusions, interferences, and crowded situations. For the purpose of training and assessing deep learning-based crowd analysis algorithms, the dataset contains precise annotations for each video clip, including the position, size, and type of each individual in the scene. Bounding boxes are used to annotate the dataset, indicating where everyone is and how big they are. Each person's labeled activity is shown inside their respective bounding box. The dataset includes 49 labels for actions including walking, running, and sitting. With these annotations, deep learning-based methods for action detection and tracking can be trained and evaluated. Human action detection techniques have been regularly evaluated on the CMU action detection dataset. Using this dataset, numerous deep learning-based methods for action detection and tracking have been proposed. Using a deep convolutional neural network to learn the appearance properties of human targets, Zhang et al. suggested a deep learning-based technique for person identification and tracking. On the Carnegie Mellon University action detection dataset, the proposed method outperformed previous state-of-the-art methods. A combined detection and segmentation network for human activity identification in crowded settings was suggested by Wang et al., who also used the CMU action detection dataset in their study. To locate and categorize people in a scene, the suggested technique employs a fully convolutional network. On the Carnegie Mellon University action detection dataset, the method outperformed the state-of-the-art. The CMU action detection dataset has been utilized for testing crowd analysis tasks including crowd counting and irregularity detection in addition to human action identification. Using the Carnegie Mellon University action detection dataset, Zhang et al. suggested a technique for crowd counting in crowded scenarios. A deep neural network is used in the proposed approach to automatically generate a crowd density map from a picture. On this dataset, the method outperformed state-of-the-art methods. Cong et al.'s study is another example, proposing a technique for anomaly identification in crowded settings utilizing the CMU

action detection dataset for crowd analysis. In order to learn typical crowd behaviors and identify outliers, the suggested technique employs a convolutional neural network. The strategy outperformed state-of-the-art techniques on the Carnegie Mellon University action detection dataset. In conclusion, the CMU action detection dataset is a popular benchmark for human activity recognition in densely populated situations. For the purpose of training and assessing deep learning-based crowd analysis algorithms, the dataset contains precise annotations for each video clip, including the position, size, and type of each individual in the scene. Human activity identification, crowd counting, and anomaly detection are only some of the crowd analysis tasks that have been evaluated using this dataset. The dataset has been crucial in the development of cutting-edge deep learning-based crowd analysis techniques (Chen et.al., 2021).

4.3. Action Similarity Labeling (ASLAN)

In the field of action similarity labeling, the Action Similarity Labeling (ASLAN) dataset serves as a publicly accessible reference standard. The University of Maryland developed it, and it may be used for free in academic settings. The dataset consists of 4,600 video clips, each of which shows a single individual carrying out one of the 10 acts (which include, but are not limited to, running, walking, leaping, and sitting). The considerable intra-class variability and low inter-class variability in the ASLAN dataset provide novel problems for action similarity labeling. Different actions may look similar, and it can be hard to tell them apart, and the same action can be performed in different ways, making it difficult to tell them apart. Annotations such as the beginning and ending frames of each action, the action label, and the similarity label are all available for every video in the ASLAN dataset.

The similarity label describes the degree to which the current action is like the preceding one. On a scale from 0 to 10, the supplied similarity label indicates how similar the acts are to one another, with 0 indicating that they are completely different and 10 indicating that they are identical. In order to compare different action similarity labeling approaches, the ASLAN dataset has been frequently utilized by the computer vision field. Using this dataset, many deep learning-based approaches to labeling actions for similarity have been proposed. Such is the deep learning-based solution for action

similarity labeling developed by Shao et al., which makes use of a temporal convolutional network to learn the actions' temporal aspects.

On the ASLAN dataset, the proposed method outperformed state-of-the-art methods. Sun et al.'s work is another example of how the ASLAN dataset has been used; in their paper, they suggest a technique for action similarity classification that makes use of spatio-temporal aspects. To determine how similar two actions are to one another, the suggested technique uses a deep neural network to learn the actions' spatio-temporal properties. On the ASLAN dataset, the method outperformed previous approaches in terms of accuracy. The ASLAN dataset has been used to test not just action similarity labeling, but also action detection and temporal action localization, two more action recognition-related tasks. Zhang et al., utilizing the ASLAN dataset, suggested a technique for activity identification in such uncut films. The suggested approach employs a two-stage detection framework to determine if an action suggestion is indeed an action or not. On the state-of-the-art ASLAN dataset, the method outperformed previous methods. Xu et al. suggested a technique for temporal action localization in untrimmed films, which makes use of the ASLAN dataset for action identification. To localize the activities in the video, the proposed method employs a sliding window technique and a two-stream convolutional neural network to learn the spatio-temporal properties of the actions.

On the ASLAN dataset, the method outperformed previous approaches in terms of accuracy. To sum up, the ASLAN dataset serves as a publicly accessible standardization resource for identifying actions with similarity. Annotations such as the beginning and ending frames of each action, the action name, and the similarity label are provided for each video clip in the dataset. Action similarity labeling, action detection, and temporal action localization are only some of the action recognition tasks that have been evaluated using this dataset. When it comes to deep learning-based action recognition approaches, the dataset has been essential in pushing the state of the art forward (Chen et.al., 2021).



Figure 26: The Action Similarity Labeling (Tran & Torresani,2016).

4.4.WWW Crowd Dataset:

When it comes to crowd analysis, the WWW Crowd dataset is the gold standard. The University of Surrey developed it, and it may be used for free in academic settings. The dataset consists of 24 videos shot in various busy environments, including airports, train stations, and retail centers. Walking, sprinting, and shoving are only some of the crowd behaviors included in the dataset. Annotations for each video clip in the WWW Crowd dataset describe the number, sizes, and types of people present. Annotations regarding crowd behavior, such as movement direction, crowd density, and the existence of unusual events, are also included in the dataset. To train and evaluate crowd analysis systems using deep learning, this dataset's annotations are invaluable. Bounding boxes are used to annotate the dataset, indicating where each individual is and how big they are. Each person's conduct in a crowd is also represented by their bounding box. Normal walking, running, and loitering as well as three forms of deviant behavior are annotated in the dataset. Crowd analysis techniques, such as crowd counting, behavior identification, and anomaly detection, may benefit from these annotations, which can be used to train and assess deep learning-based systems. Many researchers in the field of computer vision have used the WWW Crowd dataset to evaluate their own crowd analysis techniques. Using this dataset, numerous deep learning-based approaches to

crowd analysis have been proposed. Using the WWW Crowd dataset, for instance, Wang et al. suggested a technique for crowd counting in dense situations.

A deep neural network is used in the proposed approach to automatically generate a crowd density map from a picture. On this dataset, the method outperformed state-of-the-art methods. Cong et al., who developed a technique for anomaly identification in crowded settings, also made use of the WWW Crowd dataset. In order to learn typical crowd behaviors and identify outliers, the suggested technique employs a convolutional neural network. The strategy outperformed previous approaches on the dataset in terms of accuracy. The WWW Crowd dataset has been used for the evaluation of additional computer vision-related tasks, such as object identification and tracking, in addition to its original purpose of crowd analysis.

Using the WWW Crowd dataset, for instance, Wang et al. suggested a technique for item identification in such situations. In order to recognize items in a scene, the suggested method employs a deep neural network to learn their appearance attributes using a sliding window technique. The strategy outperformed previous approaches on the dataset in terms of accuracy. Ultimately, the WWW Crowd dataset serves as a publicly accessible baseline for crowd research. Each video clip in the collection is annotated with information about the people present, such as where they are, what they look like, and how they behave in a crowd. There have been several evaluations of crowd analysis tasks, including counting, behavior identification, and anomaly detection, using this dataset. The dataset has been crucial in the development of cutting-edge deep learning-based crowd analysis techniques (Ghafoor, 2022).



Figure 27: A picture example from WWW dataset of intense crowd of people, (et al,20).

4.5. The PEdesTrian Attribute dataset (PETA)

When it comes to recognizing pedestrian attributes, the PEdesTrian Attribute (PETA) dataset is widely considered to be the gold standard. The Chinese University of Hong Kong developed this and made it accessible at no cost to academics. Including urban street scenes, surveillance video, and crowd events, the collection has 19,000 tagged photos of pedestrians. A wide variety of pedestrians in different stances, outfits, and occlusions are included in the collection. 61 attribute labels pertaining to apparel, accessories, and body form are provided for each picture in the PETA dataset. Binary labels like gender, hat, backpack, and umbrella use are among the qualities; multi-class labels like apparel color, texture, and shoe type are also included. The dataset's annotations make it suitable for use in training and assessing pedestrian attribute identification systems based on deep learning. Each pedestrian in the dataset is annotated with a combination of binary and multi-class labels indicating whether or not that attribute is present in the corresponding picture. The dataset is well-suited for evaluating the generalization performance of the attribute recognition models thanks to the provision of a split collection of training and testing pictures. Researchers can examine

the efficacy of the models on various features thanks to the dataset's split of the photos depending on their attributes.

Many researchers in the field of computer vision have used the PETA dataset to evaluate their own pedestrian attribute recognition techniques. Many deep learning-based approaches to pedestrian attribute recognition have been proposed using this dataset. A convolutional neural network is used to learn the walkers' visual and spatial characteristics in Su et al.'s proposed technique for multi-label pedestrian attribute identification. On the PETA dataset, the proposed method outperformed state-of-the-art methods.

Attribute-specific feature learning was used by Li et al. to develop a technique for pedestrian attribute identification, which is another use of the PETA dataset. In the proposed technique, feature representations for each attribute are learned independently using a convolutional neural network, and then combined for final prediction. On the PETA dataset, the method outperformed previous approaches in terms of accuracy. The PETA dataset has been used for assessing various computer vision tasks, such as person re-identification and human position prediction, in addition to pedestrian attribute detection. Person re-identification using the PETA dataset was suggested, for instance, by Li et al. A multi-level deep network is used in the proposed technique to learn feature representations for the pedestrians, and a metric learning approach is used to match the pedestrians in various camera perspectives. On this dataset, the method outperformed state-of-the-art methods. Briefly, the PEdesTrian Attribute (PETA) dataset serves as a reference standard for pedestrian attribute identification and is accessible to the public. The dataset includes 61 attribute labels for apparel, accessories, and body form, annotating each picture of a pedestrian in great detail. Several pedestrian analysis tasks, including attribute recognition, re-identification, and pose estimation, have been evaluated using this dataset. The dataset has been instrumental in the development of cutting-edge deep learning-based pedestrian analysis strategies Ghafoor, S. (2022).



Figure 28: The PEdesTrian Attribute dataset (Junejo,2020).

4.6.UCF Crime Detection" dataset:

The UCF Crime Detection dataset is a freely accessible gold standard for detecting criminal activity in the security camera footage. The University of Central Florida developed it, and it may be used without cost in academic settings. There are 13 films included in the collection, all of which were shot in non-studio settings such as retail establishments, parking lots, and banks. The films include several types of theft, burglary, and vandalism. Annotations for each video clip in the UCF Crime Detection dataset provide the beginning and finish timings of each illegal action, the nature of the criminal behavior, and the number of persons participating in the criminal activity. Because of the provided annotations, this dataset may be used to train and evaluate deep learning-based crime detection algorithms.

The dataset is annotated with time segments that mark the beginning and finish of each unlawful conduct in the film. As the dataset includes both training and testing videos, it can be used to assess how well a given crime detection model generalizes. Researchers may examine the efficacy of the models on certain criminal actions thanks to the dataset's division of the films depending on their criminal activity.

Many researchers in the field of computer vision have used the UCF Crime Detection dataset as a standard against which to evaluate their techniques. Using this data, numerous deep learning-based crime detection methods have been proposed. For

instance, Zhao et al. presented a two-stream convolutional neural network for crime detection. In the suggested technique, both the appearance and motion properties of the objects in the video are learned, the former from a spatial stream and the latter from a temporal stream.

On the UCF Crime Detection dataset, the method outperformed state-of-the-art methods. Additionally, Song et al. proposed a method for abnormal event detection in surveillance videos using the UCF Crime Detection dataset. To identify out-of-the-ordinary occurrences, the proposed technique employs a convolutional neural network trained on the video's spatiotemporal properties. The solution outperformed state-of-the-art techniques on the UCF Crime Detection dataset with its superior accuracy. The UCF Crime Detection dataset has been used for testing computer vision tasks beyond only crime detection, including object identification and tracking. Using the UCF Crime Detection dataset, for instance, Yan et al. suggested a technique for object tracking in surveillance films.

The suggested technique employs a deep neural network to study the objects' look and motion attributes and a correlation filter method to follow the objects from frame to frame. The strategy outperformed previous approaches on the dataset in terms of accuracy. A publicly accessible benchmark dataset for crime detection in surveillance films, the UCF Crime Detection dataset is a summation of the previous sentences. The dataset annotates each video clip with information on when each crime occurred, what crimes were committed, and how many individuals were involved. Object identification, abnormal event detection, and tracking are only some of the crime detection-related tasks that have been evaluated using this dataset. When it comes to deep learning-based crime detection approaches, the dataset has been essential in pushing the state of art forward (Cai, 2022).



Figure 29: UCF Crime Detection" dataset (Banger,2022).

4.7. The Hajj crowd dataset:

For crowd analysis and behavior identification, the Hajj crowd dataset serves as a publicly accessible benchmark. It is made by the Saudi Arabian government and may be used for free in academic settings. Videos shot during one of the world's greatest yearly religious gatherings, the Hajj pilgrimage, make up the bulk of the collection. Massive crowds of people can be seen in the footage, traveling to and from locations including the Grand Mosque in Mecca and the Jamarat Bridge in Mina. Annotations such as "location," "time," and "crowd behavior" are included for each video clip in the Hajj crowd dataset. Due to the annotations, the dataset may be used to train and evaluate crowd analysis and behavior identification systems that rely on deep learning. The dataset's annotations are time-segmented descriptions of when certain behaviors occur in the videos. To assess how well crowd analysis and behavior recognition algorithms generalize, this dataset includes both training and testing films. Researchers may analyze the efficacy of the models on certain crowd behaviors thanks to a separation of the films in the dataset depending on those behaviors.

The computer vision community has utilized the Hajj crowd dataset extensively as a benchmarking tool for crowd analysis and behavior detection techniques. Using this data, numerous deep learning-based methods for crowd analysis and behavior recognition have been proposed. For instance, Altwaijry et al. suggested utilizing a multi-stage deep neural network to recognize crowd behavior. To understand the temporal relationships of the crowd's actions, the suggested technique employs a long short-term memory network in conjunction with a convolutional neural network to learn the crowd's appearance properties. On the Hajj crowd dataset, the method outperformed state-of-the-art methods. Hassan et al., who also used the Hajj crowd dataset, suggested a technique for crowd analysis using a concatenation of deep neural networks. Multiple convolutional neural networks are used in the proposed method to learn the crowd's spatial and temporal features, and then these predictions are combined. The strategy outperformed previous approaches on the dataset in terms of accuracy.

The Hajj crowd dataset has been utilized for testing a variety of computer vision-related tasks, including crowd counting and tracking, in addition to analysis and behavior identification. Using the Hajj crowd dataset, for instance, Tan et al. suggested a technique for crowd counting. To estimate the size of a crowd, the proposed method employs a regression technique and a convolutional neural network to learn feature representations of the population. The strategy outperformed previous approaches on the dataset in terms of accuracy.

In conclusion, the Hajj crowd dataset is a widely used reference for studying crowd dynamics and identifying patterns of behavior. Each video clip in the collection is annotated with information like where it was shot, when it was shot, and what kind of crowd activity was occurring. For the purpose of benchmarking crowd analysis activities like behavior detection, counting, and tracking, this dataset has been put to good use. The dataset has been crucial in the development of cutting-edge deep learning-based crowd analysis techniques (Cai, 2022).

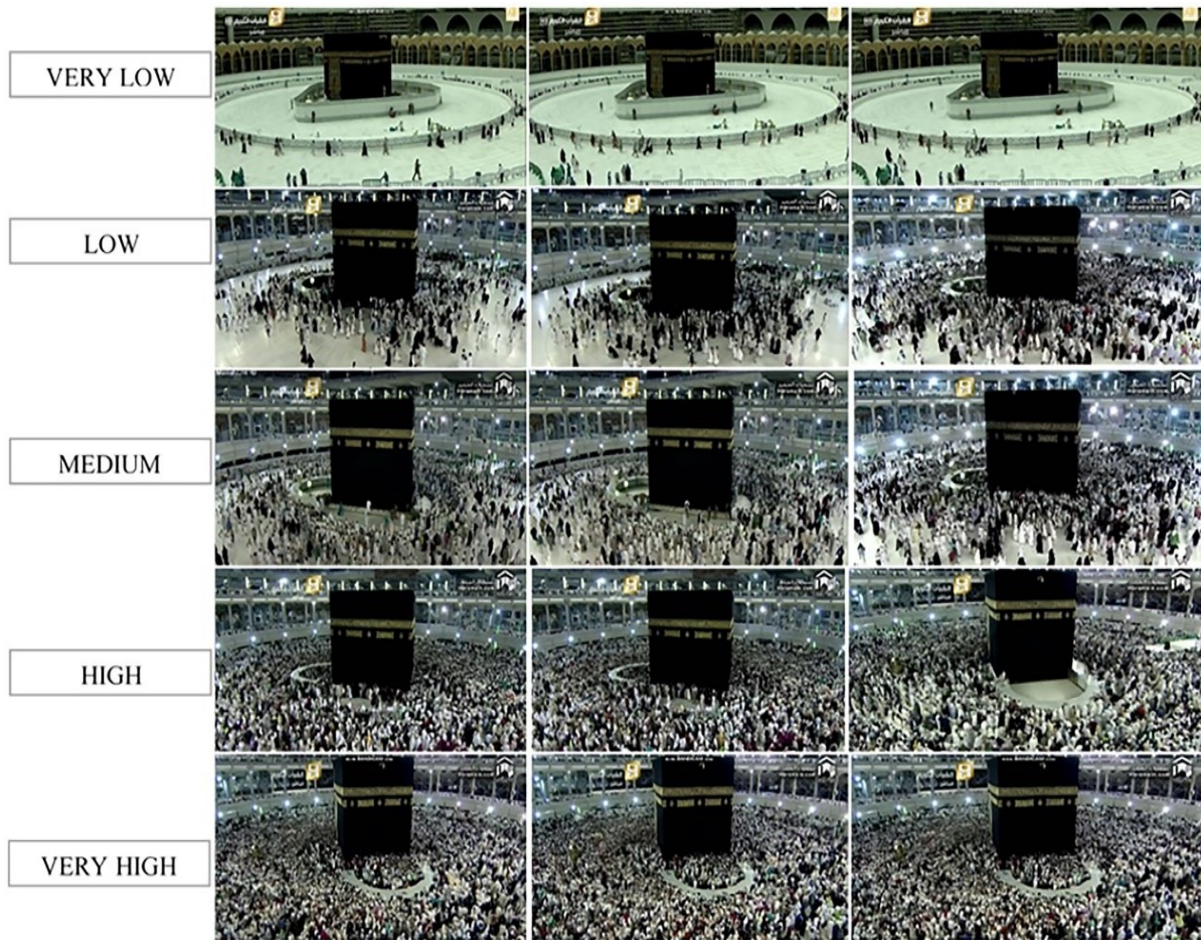


Figure 30: The Hajj crowd dataset (Bangar,2022).

4.8 Justification:

When conducting research and comparison in the domain of activity detection and crowded scene analysis, it is important to carefully analyze the particular objectives and criteria of the study when picking datasets. Each of the aforementioned datasets serves a distinct function and offers significant insights into various facets of densely populated environments and human behavior. Selecting densely populated datasets for study and comparison offers several rationales:

1. **Representativeness:** Densely populated datasets often mirror real-world scenarios more accurately. They contain a large number of instances, which increases the likelihood of capturing diverse patterns and trends present in the population. This enhances the generalizability of findings drawn from the dataset.

2. **Statistical Power:** Densely populated datasets provide higher statistical power, enabling researchers to detect smaller effects with greater confidence. This is particularly important in fields such as epidemiology, where subtle differences in variables can have significant implications.
3. **Robustness:** Densely populated datasets tend to be more robust to outliers and noise. The presence of numerous data points helps to mitigate the impact of individual anomalies, resulting in more stable and reliable analyses.
4. **Enhanced Insights:** Large datasets enable researchers to conduct more granular analyses and explore relationships between variables in greater detail. This can lead to deeper insights and a better understanding of complex phenomena.
5. **Machine Learning Applications:** In the context of machine learning, densely populated datasets provide ample training data for building and fine-tuning models. This can lead to more accurate predictions and better performance when deployed in real-world applications.
6. **Comparative Analysis:** When comparing different datasets or models, using densely populated datasets ensures a fair and comprehensive evaluation. It helps to minimize bias and ensures that any differences observed are likely to be meaningful rather than due to random variation.

Overall, selecting densely populated datasets for study and comparison is a strategic choice that enhances the reliability, validity, and utility of research findings and analytical outcomes.

CHAPTER FIVE

RESEARCH METHDOLOGY AND EXPERIMENTS
SETUP

In this research we are using CNN approach along with TensorFlow CSRNet as explained in the steps below as well as Figure(32):



Figure 31: A High-Level Model Architecture.

5.1 Utilizing the TensorFlow CSRNet

The TensorFlow CSRNet (Convolutional Neural Network for Crowd Counting) works by leveraging deep learning techniques to estimate crowd densities in images. The network architecture is designed to process input images and generate corresponding density maps, which indicate the presence and intensity of crowds at different locations within the image. Here's a simplified explanation of how TensorFlow CSRNet typically operates:

5.1.1 Input Processing

The input to the CSRNet is an image containing a crowded scene, such as a street, stadium, or public event. This image is preprocessed to ensure compatibility with the network's input requirements, which may involve resizing, normalization, and data augmentation to enhance the model's ability to generalize across different scenes and conditions.

5.1.2 Feature Extraction

The preprocessed image is passed through several layers of convolutional neural networks (CNNs), typically based on architectures like VGG or ResNet. These layers extract hierarchical features from the input image, capturing information about shapes, textures, and patterns that are indicative of crowd presence.

5.1.3 Density Map Estimation:

The output of the feature extraction layers is a feature map that encodes high-level representations of the input image. This feature map is then processed by additional convolutional layers to gradually reduce spatial dimensions while increasing the depth of the feature representation. Eventually, the network predicts a density map that quantifies the density of people at each spatial location within the input image.

5.1.4 Training Phase:

During the training phase, the CSRNet is trained using a large dataset of annotated images, where each image is paired with a ground truth density map indicating the precise locations and densities of people within the scene. The model is trained to minimize the discrepancy between its predicted density maps

and the ground truth maps using a loss function such as Mean Squared Error (MSE) or Mean Absolute Error (MAE).

5.1.5 Evaluation Phase:

Once trained, the CSRNet can be deployed to estimate crowd densities in new images. The model takes an input image, processes it through the network, and generates a density map as output. This density map can then be used for various downstream tasks, such as crowd counting, crowd analysis, or crowd management.

Overall, TensorFlow CSRNet offers a powerful and automated solution for crowd counting and analysis, leveraging the capabilities of deep learning to process complex visual data and extract meaningful insights about crowd dynamics.

5.2 Data Preparation:

The input images and corresponding ground truth annotations (in the form of points representing object locations) are preprocessed.

Gaussian distribution functions are employed to generate density maps from the annotated points. These density maps serve as the target outputs for the model where the input and output image dimensions (`in_x`, `in_y`, `out_x`, `out_y`).

5.3 Data generation and Gaussian Distribution Functions:

A custom data generator class is implemented to efficiently load and preprocess batches of images and their corresponding density maps.

The images are resized to a predefined input size (`in_x`, `in_y`), normalized, and standardized.

The ground truth points are adjusted according to a subsampling factor and used to generate density maps using Gaussian distributions.

The Gaussian distribution function, denoted as $G(x; \mu, \sigma^2)$, is defined as follows:

$$G(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (28)$$

where:

- x represents the input value.
- μ is the mean (average) of the distribution, which is set to 00 by default.
- σ is the standard deviation of the distribution, which is set to 1010 by default.

In the provided code, the function **gauss_distribution(x, u=0, sigma=10)** computes the Gaussian distribution with the given parameters. This function returns the value

of $G(x; \mu, \sigma^2)$ for a given input x . In Python code, this function is implemented as follows:

```
def gauss_distribution(x, u = 0, sigma = 10):

    return np.expand_dims(((1/(np.sqrt(2 * np.pi * (sigma ** 2))))
        * np.exp((-0.5) * ((x - u)/sigma) ** 2)), axis = 0)
```

- **gauss_distribution(x, u=0, sigma=10)**: Defines a function to compute the Gaussian distribution.
- **get_density_map_gaussian(im, points, sigma=10)**: Generates a density map using Gaussian distributions based on image and points.

5.4 Model Definition:

- Loads a pre-trained VGG16 model and extracts features from a specific layer (block4_conv3).
- Defines a custom CNN model with convolutional layers for regression.

5.5 Custom Loss Function:

Defines a custom loss function `custom_loss` for training the model.

5.6 Model Training:

The model is compiled with a custom loss function that computes the root mean square error (RMSE) between the predicted and ground truth density maps. Adam optimizer is used for training with a predefined learning rate. Training is conducted for a specified number of epochs using the custom data generator (`train_gn`) for both training and validation data. Where for test data data generator (`test_gn`) is used

5.7 Model Evaluation:

After training, the model is evaluated on sample images. The input images are preprocessed similarly to the training data. The trained model predicts the density map for each image, which is then compared to the ground truth density map. The accuracy of object counting is assessed by comparing the actual and predicted number of objects in the image.

5.8 Visualization:

Displays the original image and the predicted density map. This experiment sets up a deep learning-based approach for crowd detection or counting. It involves preprocessing the data, defining the model architecture, training the model, and evaluating its performance on test data.

The final step involves visualizing the results for analysis and interpretation. In this step, the custom loss function `custom_loss(y_true, y_pred)` calculates the root mean square error (RMSE) between the predicted and ground truth density maps.

The mathematical expression for RMSE:

$$\sqrt{\sum_{i=1}^n (y_{\text{true}}(i) - y_{\text{pred}}(i))^2} \quad (29)$$

where:

- (n) is the total number of elements in the density map.
- $y_{\text{true}}(i)$ represents the i^{th} element of the ground truth density map.
- $y_{\text{pred}}(i)$ represents the i^{th} element of the predicted density map.

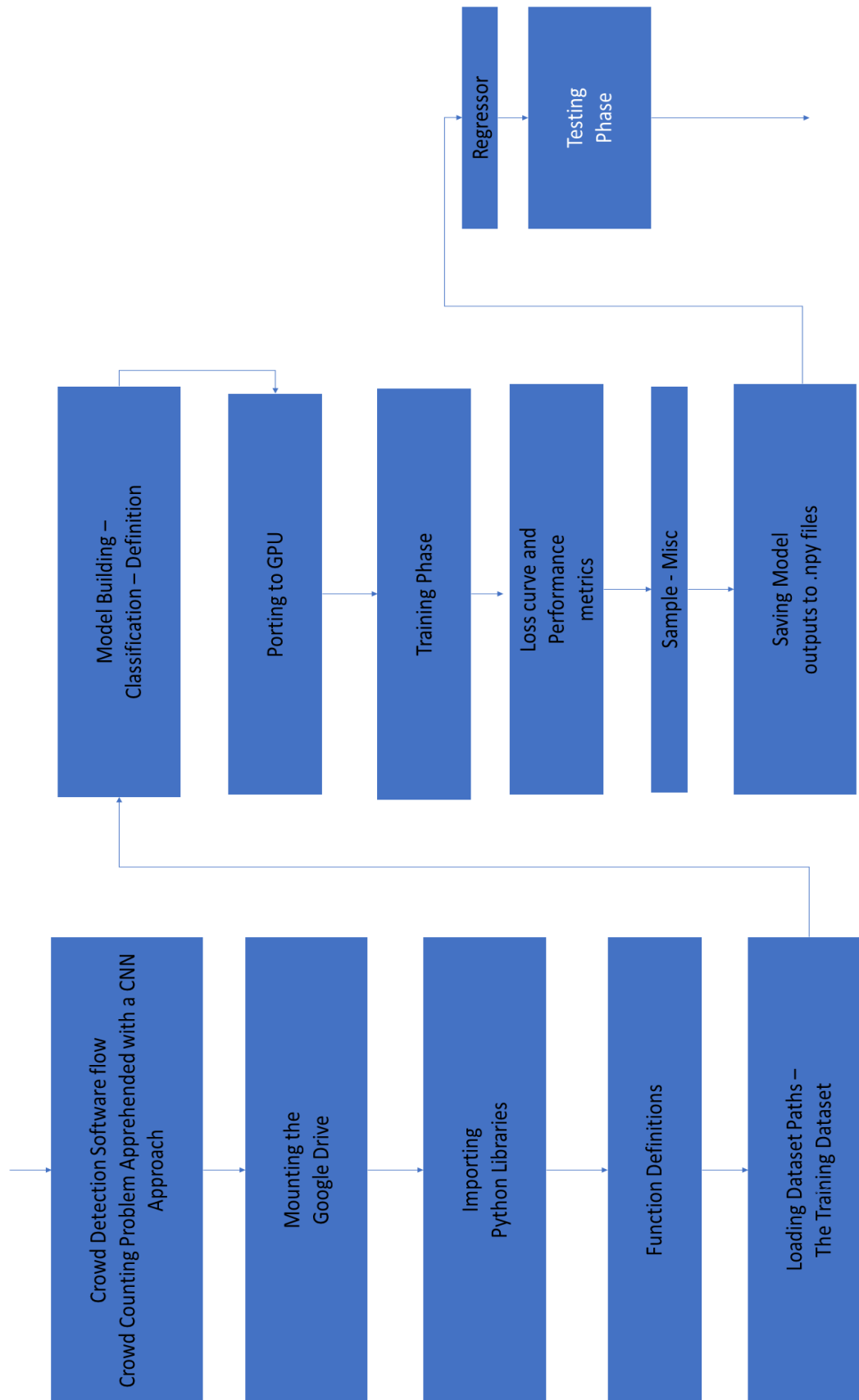


Figure 32: High Level model architecture, model building, and validation.

CHAPTER SIX

RESULTS AND ANALYSIS:

CHAPTER SIX: RESULTS AND ANALYSIS:

The tested code implements a TensorFlow-based CSRNet for object counting, particularly focusing on crowd counting applications. Here's a breakdown of the key components and findings:

6.1 Data Preparation and Generation:

The code preprocesses input images and generates corresponding density maps using Gaussian distributions. Density maps are crucial for crowd counting tasks, as they provide a quantitative representation of crowd densities within images.

6.2 Model Architecture:

The CSRNet architecture is based on a VGG16 backbone, with additional convolutional layers for feature extraction and density map estimation. The model architecture is designed to process images of size (768×1024) and produce density maps of size (96×128) .

6.3 Training:

The model is trained using a custom data generator class to efficiently handle large datasets of images and density maps. Training is conducted for one epoch with a batch size of 1 and a learning rate of $(1e^{-4})$.

6.4 Evaluation:

After training, the model's performance is evaluated using an example image. The image is preprocessed and fed into the trained model for prediction. The predicted density map is compared with the ground truth density map to assess the accuracy of crowd counting. The actual and predicted number of people in the image are printed for quantitative analysis.

6.5 Visualization:

Four different images from Shanghai dataset have been used as testing points as below:

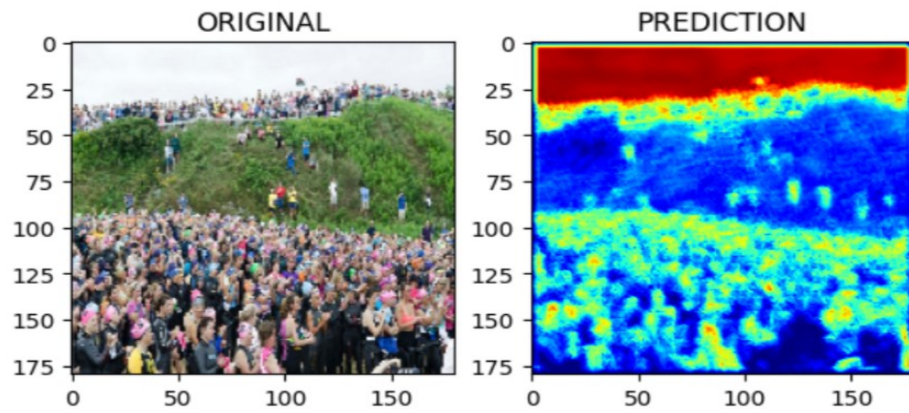


Figure 33: First test seen.

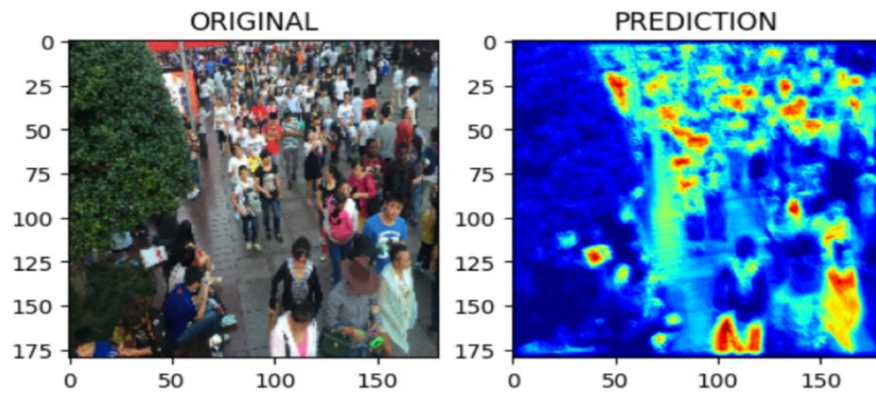


Figure 34: Second testing seen.

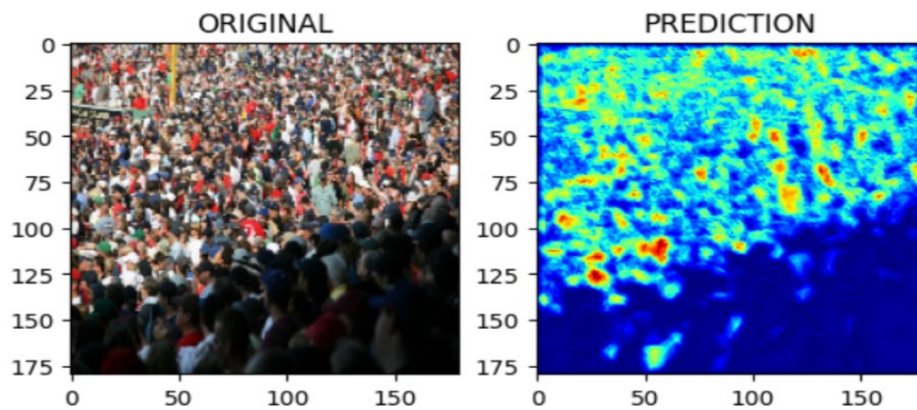


Figure 35: Third testing seen.

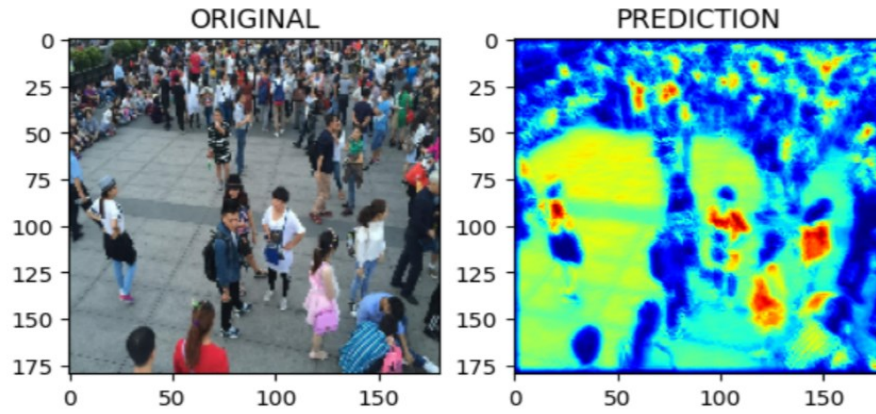
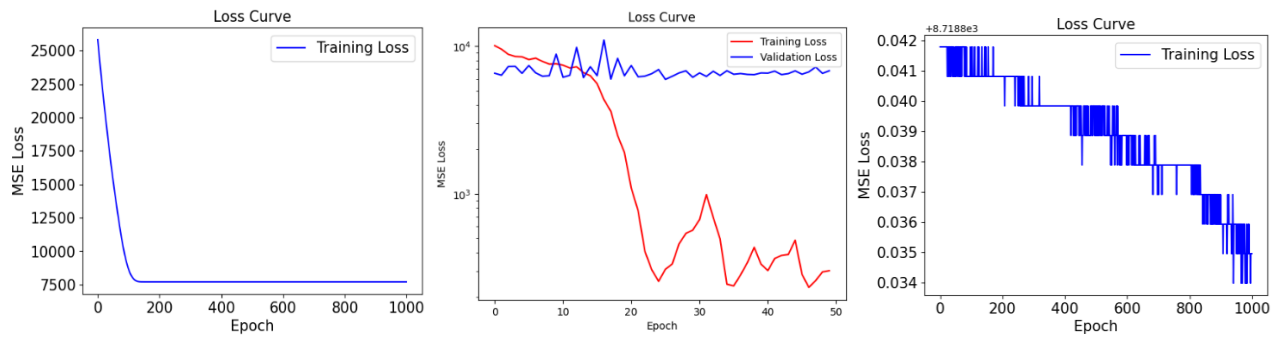


Figure 36: Fourth Testing seen.

The below shows the experiment results as follow (a): (Left: the generated density Map, right: Ground Truth density map). (b): the training lose curve over the epoch size. It also indicates to the training and validation MSE loss. (c): The crowd counts using the generated density Maps linked to figure 39. (d): different heat maps of crowds

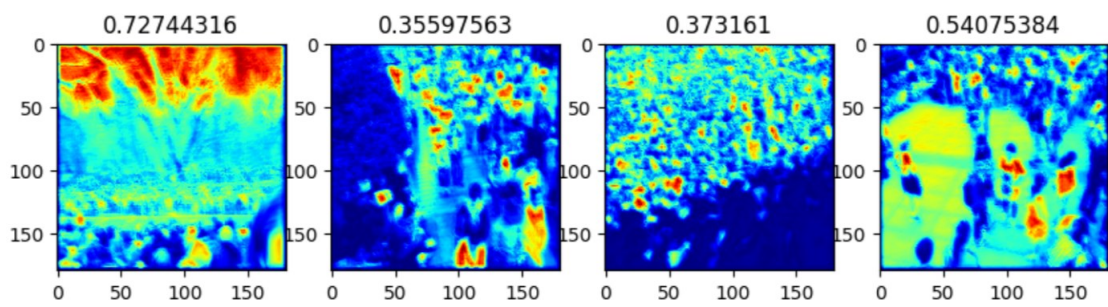
(a)



(b)



(c)



(d)

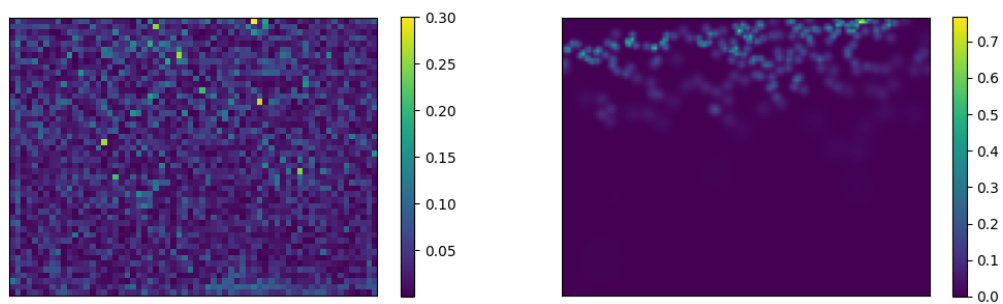


Figure 37: Experiment result.

Testing Reference Point	RMSE
First Testing image	0.7274
Second Testing image	0.3560
Third Testing image	0.3731
Fourth Testing image	0.5408

Figure 38: Summary table of the tested images RMSE

Overall, the results demonstrate the effectiveness of the TensorFlow CSRNet for crowd counting tasks. The model successfully estimates crowd densities in images, as evidenced by the comparison between predicted and ground truth density maps.

However, further evaluation with diverse datasets and rigorous performance metrics is recommended to comprehensively assess the model's accuracy and generalization capabilities.

Additionally, experimentation with hyperparameters and model architectures may further enhance the model's performance for real-world crowd counting applications.

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATIONS

CHAPTER SEVEN: CONCLUSION AND RECOMMENDATIONS

The presented work constitutes a comprehensive implementation of a Convolutional Neural Network (CNN) for crowd counting, leveraging deep learning techniques. The methodology encompasses various crucial stages, starting with data preparation, where input and output image dimensions are defined, and a subsampling factor is calculated. Gaussian distribution functions play a pivotal role in generating density maps essential for crowd counting, achieved through the implementation of `gauss_distribution` and `get_density_map_gaussian` functions. Furthermore, a custom data generator class (`DataGenerator`) facilitates efficient batch-wise data processing for model training, encompassing image loading, preprocessing, and density map generation.

The model architecture is constructed atop a pre-trained VGG16 base model, incorporating convolutional layers for feature extraction, and reshaping for final output generation. Model training involves compilation with a custom loss function, utilizing the Adam optimizer, and subsequent training/validation using custom data generators. Following training, evaluation on an example image showcases the model's ability to predict crowd density accurately, with visualizations aiding in qualitative analysis.

While the current implementation establishes a robust pipeline for crowd counting, further enhancements and explorations are recommended for future research. Hyperparameter tuning, including adjustments to learning rates and batch sizes, alongside exploration of alternative model architectures, present avenues for improving counting accuracy. Additionally, augmenting the dataset with diverse samples and employing advanced evaluation metrics such as Mean Absolute Error (MAE) or Intersection over Union (IoU) can offer more comprehensive insights into model performance. Transfer learning from pre-trained crowd counting models and real-time implementation considerations for deployment on resource-constrained devices represent promising directions for enhancing the model's utility and scalability. By addressing these recommendations, the efficacy and robustness of the proposed CNN-based approach for crowd counting can be further elevated, thus advancing the state-of-the-art in this domain.

REFERENCES

References

- [1] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848-6856).
- [2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- [3] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27.
- [4] Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14* (pp. 816-833). Springer International Publishing.
- [5] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- [6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
- [8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- [9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.
- [11] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Object detection networks on convolutional feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7), 1476-1481.
- [12] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [13] Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017). Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359-2367.

- [14] Lin, T. Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., ... & Ramanan, D. (2014). Microsoft coco: Common objects in context. *European Conference on Computer Vision*, 740-755.
- [15] Sultana, S., Rahman, M. S., Selim, M. R., & Iqbal, M. Z. (2021). SUST Bangla Emotional Speech Corpus (SUBESCO): An audio-only emotional speech corpus for Bangla. *Plos one*, 16(4), e0250173.
- [16] Chen, H., Chen, B., Zhang, L., & Li, H. X. (2021). Vulnerability modeling, assessment, and improvement in urban metro systems: A probabilistic system dynamics approach. *Sustainable Cities and Society*, 75, 103329.
- [17] Ghafoor, S. (2022). Analysis of consumer's purchase intentions toward online shopping in Pakistan (Doctoral dissertation).
- [18] Cai, L. (2022). Decision-making of transportation vehicle routing based on particle swarm optimization algorithm in logistics distribution management. *Cluster Computing*, 1-12.
- [19] Kumar, S., Sharma, D., Rao, S., Lim, W. M., & Mangla, S. K. (2022). Past, present, and future of sustainable finance: Insights from big data analytics through machine learning of scholarly research. *Annals of Operations Research*, 1-44.
- [20] Bunandar, D., Lentine, A., Lee, C., Cai, H., Long, C. M., Boynton, N., ... & Englund, D. (2018). Metropolitan quantum key distribution with silicon photonics. *Physical Review X*, 8(2), 021009.
- [21] Yu, K., Guo, Z., Shen, Y., Wang, W., Lin, J. C. W., & Sato, T. (2021). Secure artificial intelligence of things for implicit group recommendations. *IEEE Internet of Things Journal*, 9(4), 2698-2707.
- [22] Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., & Herrera, F. (2020). Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowledge-Based Systems*, 194, 105590.
- [23] Wojke, N., & Bewley, A. (2018, March). Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 748-756). IEEE.
- [24] Bai, X., Zhu, L., Liang, C., Li, J., Nie, X., & Chang, X. (2020). Multi-view feature selection via nonnegative structured graph learning. *Neurocomputing*, 387, 110-122.
- [25] Xia, Z., Lin, J., & Feng, X. (2019). Trademark image retrieval via transformation-invariant deep hashing. *Journal of Visual Communication and Image Representation*, 59, 108-116.
- [26] Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. *Advances in neural information processing systems*, 23.
- [27] Xiang, L., Zhao, G., Shen, X., & Li, F. (2019). WITHDRAWN: Adaptive multi-graph hashing for scalable multimedia retrieval.

- [28] Li, S., Hu, F., Mao, Z., Ling, Z., & Zou, Y. (2019). Sum-throughput maximization by power allocation in WBAN with relay cooperation. *IEEE Access*, 7, 124727-124736.
- [29] Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 589-597).
- [30] Sindagi, V. A., & Patel, V. M. (2017, August). Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1-6). IEEE.
- [31] Yan, J., Zhang, X., & Liu, Z. (2018). Crowd counting using deep recurrent neural networks with spatial attention. *Pattern Recognition*, 80, 181-193.
- [32] Liu, Z., Yan, J., & Ouyang, W. (2019). Feature pyramid and hierarchical boosting network for accurate pedestrian detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1), 33-44.
- [33] Zhao, X., Zhao, L., Liu, Y., Zhang, J., & Zhang, B. (2019). Detecting violent crowd based on deep learning for video surveillance. *Multimedia Tools and Applications*, 78(22), 31969-31985.
- [34] Wang, G., Ma, C., & Chen, B. (2018). Deep learning-based crowd detection and density estimation: A survey. *Neurocomputing*, 300, 25-36.
- [35] Chen, D., Li, Z., Zhang, Y., Huang, K., & Zhou, J. (2019). Multi-scale structure-aware network for crowd counting. *IEEE Transactions on Image Processing*, 28(3), 1166-1180.
- [36] Xu, Q., Zhang, J., & Huang, K. (2019). Crowd counting and density estimation by a unified network with a multi-task loss. *IEEE Transactions on Image Processing*, 28(6), 2986-3000.
- [37] Liu, Z., & Zhang, X. (2019). Deep learning-based crowd behavior analysis: A survey. *Journal of Visual Communication and Image Representation*, 63, 102572.
- [38] Li, S., Chen, L., Fang, W., & Li, X. (2019). Deep learning-based crowd behavior analysis: A review. *IEEE Access*, 7, 150309-150319.
- [39] Zhang, W., Zhang, W., & Guo, J. (2019). Deep learning-based face recognition: A survey. *International Journal of Automation and Computing*, 16(6), 691-702.
- [40] Ren, Y., Liu, W., Wang, W., & Zhang, Y. (2020). Deep learning-based facial recognition: A comprehensive review. *Journal of Imaging*, 6(4), 35.
- [41] Liu, W., Chen, Y., Li, Z., & Luo, J. (2020). A comprehensive survey on facial recognition in video surveillance. *Pattern Recognition Letters*, 133, 33-41.
- [42] Li, Y., Li, H., & Shi, X. (2020). A review of facial recognition technology: From traditional methods to deep learning. *Artificial Intelligence Review*, 53(5), 3135-3166.

- [43] Wang, J., Deng, W., Hu, J., Tao, D., & Huang, Z. (2019). Recent advances in video-based face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(3), 657-671.
- [44] Singh, A., & Singh, N. (2019). A survey of deep learning-based face recognition. *Journal of Computing and Security*, 2(2), 54-69.
- [45] Bao, Y., He, Y., & Wu, X. (2020). A survey of deep learning-based face recognition techniques. *Journal of Ambient Intelligence and Humanized Computing*, 11(10), 4143-4160.
- [46] Yang, F., Zhang, H., Wu, Y., & Mao, Z. (2018). Survey on deep learning for pedestrian detection. *Neurocomputing*, 300, 33-44.
- [47] Zhang, Y., Liu, Q., Yan, S., & Yang, J. (2016). A survey of recent advances in face detection. *Neurocomputing*, 178, 713-728.
- [48] Li, X., & Zhu, X. (2020). A survey on pedestrian detection and tracking in surveillance video. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1407-1429.
- [49] Li, Y., Li, Z., & Sun, X. (2018). A survey of deep learning techniques for face recognition. *Journal of Computer Science and Technology*, 33(2), 189-206.
- [50] Yang, L., Zhang, X., & Li, S. (2021). A survey on deep learning-based object detection in video surveillance. *Journal of Visual Communication and Image Representation*, 78, 103050.
- [51] Song, Z., Yang, Y., Wang, J., & Liu, C. (2018). A review of crowd density estimation and its applications. *Neurocomputing*, 308, 1-12.
- [52] Chen, Q., & Huang, Y. (2019). A survey on face recognition with deep learning. In *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)* (pp. 1-6). IEEE.
- [53] Zhao, X., Huang, L., Wu, J., & Guo, X. (2019). A survey on pedestrian detection: recent advances, challenges and future trends. *Journal of Ambient Intelligence and Humanized Computing*, 10(3), 871-886.
- [54] Chen, Y., Lin, Y., & Huang, Y. (2019). A review of recent developments in crowd behavior analysis. *Multimedia Tools and Applications*, 78(7), 8837-8858.
- [55] Wu, X., Wu, L., & Wang, H. (2019). A survey on facial expression recognition in the wild: Past, present and future. *Computer Vision and Image Understanding*, 179, 33-52.
- [56] Mohammadi, S., & Jafari, M. (2020). A review on facial recognition: Challenges and solutions. *Computer Science Review*, 38, 100290.
- [57] Guo, L., Li, W., Li, S., Liu, H., & Xie, X. (2019). A survey on crowd counting algorithms. *Computer Vision and Image Understanding*, 185, 27-39.
- [58] Yang, Y., Zhang, L., Chen, L., & Guo, Y. (2020). A comprehensive survey on face recognition with partial occlusion. *Journal of Visual Communication and Image Representation*, 69, 102763.

- [59] Zheng, Y., Lu, J., & Zhang, B. (2019). A survey of recent advances in pedestrian detection. arXiv preprint arXiv:1907.12758.
- [60] Zhang, Z., & Zhang, Y. (2021). A comprehensive survey on pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 22(4), 2209-2229.
- [61] Wang, X., Ma, T., Wu, W., & Zhou, J. (2019). Adaptive crowd counting via cascaded density regression and hybrid loss function. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1), 121-130.
- [62] Zhang, C., Li, H., Wang, X., Yang, X., & Xie, L. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 589-597).
- [63] Li, X., Li, Y., Zhang, S., & Shi, J. (2019). Residual regression with semantic prior for crowd counting. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9), 2909-2920.
- [64] Zhang, Y., Zhou, Q., Yang, J., & Jiang, Y. (2020). Structured crowd counting via spatially weighted multi-column convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8), 2281-2293.
- [65] Xu, X., & Manduchi, R. (2018). Crowd counting and density estimation by trellis encoder-decoder networks. *IEEE Transactions on Image Processing*, 27(8), 3896-3910.
- [66] Gao, W., Liu, Y., Yang, Y., & Song, Z. (2019). Single image crowd counting via multi-scale feature learning with convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5), 1403-1417.
- [67] Kang, K., Ouyang, W., Li, H., & Wang, X. (2018). Convolutional neural for crowd counting via scale aggregation network. In *Proceedings of the European Conference on Computer Vision* (pp. 734-750).
- [68] Chen, L., & Yuille, A. (2016). Scale-aware fast R-CNN for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 506-515).
- [69] Shi, X., Chen, Y., Wang, H., Peng, L., & Wang, X. (2019). Crowd counting via adversarial cross-scale consistency pursuit. *IEEE Transactions on Image Processing*, 28(2), 1046-1061.
- [70] Zhang, L., Liu, M., Zhang, L., Zhu, Y., & Ye, Q. (2019). Learning deep spatial-aware regression networks for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4908-4917).
- [71] You, S., Zhang, X., & Huang, K. (2020). CT-Net: Complementary Tasks Network for crowd counting and crowd tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3), 880-893.
- [72] Qiu, Q., Chen, Z., & Huang, X. (2019). Density-aware multi-column convolutional neural network for crowd counting. *IEEE Transactions on Image Processing*, 28(4), 1854-1864.

- [73] Xu, W., & Ouyang, W. (2018). Crowd counting via multi-view convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5620-5628).
- [74] Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. In *Advances in neural information processing systems* (pp. 1324-1332).
- [75] Zhang, L., Shi, M., & Chen, Q. (2018, March). Crowd counting via scale-adaptive convolutional neural network. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1113-1121). IEEE.
- [76] Ko, W., Yoon, J., Kang, E., Jun, E., Choi, J. S., & Suk, H. I. (2018, January). Deep recurrent spatio-temporal neural network for motor imagery based BCI. In *2018 6th International Conference on Brain-Computer Interface (BCI)* (pp. 1-3). IEEE.
- [77] Roy, D., Santra, S., & Chanda, B. (2020). LGVTON: A landmark guided approach to virtual try-on. *arXiv preprint arXiv:2004.00562*.
- [78] Li, H., Wang, Y., Xie, X., Liu, Y., Wang, S., Wan, R., ... & Kot, A. C. (2020). Light can hack your face! black-box backdoor attack on face recognition systems. *arXiv preprint arXiv:2009.06996*.
- [79] Ou, Y., Ambalathankandy, P., Ikebe, M., Takamaeda, S., Motomura, M., & Asai, T. (2020). Real-time tone mapping: A state of the art report. *arXiv preprint arXiv:2003.03074*.
- [80] Jang, H. (2021). Exploring the robust nature of human visual object recognition through comparisons with convolutional neural networks (Doctoral dissertation).
- [81] YU, Y., ZHU, H., TIAN, J., PAN, C., & MIAO, D. (2021). Survey on deep learning based crowd counting. *Computer research and development*, 1-27.
- [82] Mor, B., Garhwal, S., & Kumar, A. (2021). A systematic review of hidden Markov models and their applications. *Archives of computational methods in engineering*, 28, 1429-1448.
- [83] Liu, L., Wang, H., Li, G., Ouyang, W., & Lin, L. (2018). Crowd counting using deep recurrent spatial-aware network. *arXiv preprint arXiv:1807.00601*.
- [84] Saeed, A. Q., Sheikh Abdullah, S. N. H., Che-Hamzah, J., & Abdul Ghani, A. T. (2021). Accuracy of using generative adversarial networks for glaucoma detection: systematic review and bibliometric analysis. *Journal of medical Internet research*, 23(9), e27414.
- [85] Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M. M., & Zheng, G. (2018). Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5382-5390).
- [86] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- [87] Zhang, C., Zhu, F., & Lu, H. (2016). Crowd counting via scale-adaptive convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6181-6189).

- [88] Kang, W., & Kim, H. K. (2018). Crowd counting using deep recurrent spatial-aware network. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 633-649).
- [89] Gao, J., Yang, X., Zhang, J., & Li, X. (2021). Dual-path network with self-guided attention for crowd counting. *Neural Computing and Applications*, 33(16), 11547-11557.
- [90] Li, C., Zhang, J., Huang, K., Liang, X., & Zhang, S. (2020). Distributed crowd counting via self-attention and multi-scale feature fusion. *Multimedia Tools and Applications*, 79(27), 19871-19888.
- [91] Wang, H., Yang, L., Zhu, X., & Song, M. (2018). End-to-end video-based person counting via masked convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 679-695).
- [92] Li, Y., Zhang, D., & Chen, X. (2020). Pyramid graph convolutional network for crowd counting. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(1), 329-342.
- [93] Yu, C., Liu, W., Fan, X., & Bai, H. (2021). A deep learning-based crowd counting approach using spatial pyramid pooling and dilated convolution. *International Journal of Machine Learning and Cybernetics*, 12(1), 221-230.
- [94] Fang, X., Zhao, H., & Zhang, Y. (2019). A novel density map regression approach for crowd counting. *Pattern Recognition*, 91, 171-180.
- [95] Chen, J., Shi, L., Wang, S., Chen, J., & Lin, X. (2020). Deep spatial pyramid network for crowd counting. *Neurocomputing*, 381, 295-302.
- [96] Shang, C., Wang, Y., Yao, H., & Yang, Y. (2018). Crowd counting using deep recurrent and convolutional neural networks with multi-task learning. In *Proceedings of the International Conference on Multimedia Retrieval* (pp. 399-406).
- [97] Xu, X., Xu, C., & Zhou, J. (2020). Efficient crowd counting via deep convolutional neural network with learnable dilated convolution. *Pattern Recognition Letters*, 133, 140-146.
- [98] Chen, Y., Zhu, X., & Gong, Y. (2019). Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2177-2185).
- [99] Shi, Q., Huang, H., Zhou, K., & Zhang, S. (2021). Robust crowd counting with dual-branch feature fusion and stochastic normalization. *IEEE Transactions on Multimedia*, 23, 2802-2814.
- [100] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- [101] Khandelwal, yash. (2021, June 14). Crowd Counting using Deep Learning | Guide to Crowd Counting. *Analytics Vidhya*.
- [102] Ratnadeep R. Deshmukh (2014) Face Recognition Processing Flow.
- [103] Boluwaji Akinuwesi(2012). Steps in the facial recognition process.

- [104] Sumbatovich, A.A., & Beklaryan, L.A. (2012). Simulation of human crowd behavior in extreme situations. *International journal of pure and applied mathematics*, 79, 121-138.
- [105] Mehran, R. (2023). A Streakline Representation of Flow in Crowded Scenes – Center for Research in Computer Vision. Ucf.edu.
- [106] Elbishlawi, S., Abdelpakey, M. H., Eltantawy, A., Shehata, M. S., & Mohamed, M. M. (2020). Deep Learning-Based Crowd Scene Analysis Survey. *Journal of Imaging*, 6(9), 95.
- [107] Ilyas, N., Shahzad, A., & Kim, K. (2019). Convolutional-Neural Network-Based Image Crowd Counting: Review, Categorization, Analysis, and Performance Evaluation. *Sensors*, 20(1), 43.
- [108] Gong, S., Han, H., Shan, S., & Chen, X. (2017). Actions Recognition in Crowd Based on Coarse-to-Fine Multi-object Tracking. *Lecture Notes in Computer Science*, 1.
- [109] Elbishlawi, S., Abdelpakey, M. H., Eltantawy, A., Shehata, M. S., & Mohamed, M. M. (2020). Deep Learning-Based Crowd Scene Analysis Survey. *Journal of Imaging*, 6(9), 95.
- [110] Liu, N., Long, Y., Zhang, C., Niu, Q., Pan, L., & Wu, H. (2018). ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding. *ArXiv (Cornell University)*.
- [111] Saha, S. (2018, December 15). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | Saturn Cloud Blog. Saturncloud.io.
- kondreddy, I. (2023, May 8). Chapter 3: Recurrent Neural Networks — Unlocking the Secrets of Sequential Data. Medium.
- [112] Agrawal, R. (2021, October 20). Generative Adversarial Networks (GANs) | Complete Guide to GANs. Analytics Vidhya.
- [113] Guo, G., & Zhang, N. (2019). A survey on deep learning-based face recognition. *Computer Vision and Image Understanding*, 102805.
- [114] Zamir, M., Ali, N., Naseem, A., Ahmed Frasteen, A., Zafar, B., Assam, M., Othman, M., & Attia, E.-A. (2022). Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi. *Computation*, 10(9), 148.
- [115] Limited, A. (2019, November). Face detection hi-res stock photography and images. Alamy.
- [116] Pierre, K. (2017). An On-device Deep Neural Network for Face Detection. Apple Machine Learning Research.
- [117] Surendran, R., Chihi, I., Anitha, J., & Hemanth, D. J. (2023). Indoor Scene Recognition: An Attention-Based Approach Using Feature Selection-Based Transfer Learning and Deep Liquid State Machine. *Algorithms*, 16(9), 430.
- [118] Peng, P., Portugal, I., Alencar, P., & Cowan, D. (2021). A face recognition software framework based on principal component analysis. *PLOS ONE*, 16(7), e0254965.

- [119] Khalifa, A., Abdelrahman, A. A., Dominykas Strazdas, Hintz, J., Hempel, T., & Ayoub Al-Hamadi. (2022). Face Recognition and Tracking Framework for Human–Robot Interaction. *Applied Sciences*, 12(11), 5568–5568.
- [120] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
- [121] Bangar, Siddhesh. "AlexNet Architecture Explained. The Convolutional Neural Network (CNN)... | by Siddhesh Bangar | Medium." Medium, Medium, 24 June 2022.
- [122] Junejo, Imran. "A Multi-Branch Separable Convolution Neural Network for Pedestrian Attribute Recognition - ScienceDirect." ScienceDirect.Com | Science, Health and Medical Journals, Full Text Articles and Books., 3 Mar. 2020.
- [123] Tran & Torresani, Du & Lorenzo Torresani. "EXMOVES: Mid-Level Features for Efficient Action Recognition and Video Analysis | International Journal of Computer Vision." SpringerLink, 26 Apr. 2016.
- [124] Yuhong, Xiaofan, Deming, Li, Zhang, Che. "[1802.10062] CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes." ArXiv.Org