Crowd Density Estimation and Crowd Behaviour Analysis using Deep Learning

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

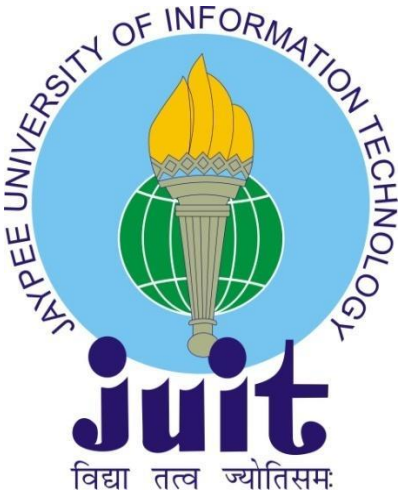
By

Harshit Singh (191348)

Under the supervision of

Dr. Vipul Kumar Sharma

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Crowd Density Estimation and Crowd Behaviour Analysis using Deep Learning"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Vipul Kumar Sharma**, Assistant Professor(SG), Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, Solan.

I also authenticate that I have carried out the above mentioned project work under the proficiency stream **Artificial Intelligence**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Harshit Singh, 191348

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Vipul Kumar Sharma
Assistant Professor (SG)
Computer Science and Engineering

Dated:

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name:_____ **Department:**_____ **Enrolment No** _____

Contact No._____ **E-mail.**_____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

.....

Acknowledgement

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

I am grateful and wish my profound indebtedness to my supervisor **Dr. Vipul Kumar Sharma**, Department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of our supervisor in the field of "Crowd Density Estimation and Crowd Behaviour Analysis using Deep Learning" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartfelt gratitude to **Prof. Dr. Vivek Kumar Sehgal**, Head of Department, Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat.

I am grateful to each and every individual who directly or indirectly helped me in making this project a success.

Finally, I must acknowledge with due respect the constant support and patience of my parents and grandparents.

Harshit Singh
Roll No. 191348

Table of Contents

List of Figures

List of Graphs

List of Abbreviations

CNN	Convolutional Neural Network
M-CNN	Multi-Column Convolutional Neural Network
VGG	Visual Geometry Group
ML	Machine Learning
DL	Deep Learning
KNN	K-Nearest Neighbour
SVM	Support Vector Machine
ReLU	Rectified Linear Unit
IDE	Integrated Development Environment
RNN	Recurrent Neural Network
OpenCV	Open Computer Vision
OS	Operating System
CP-CNN	Contextual Pyramid Convolutional Neural Networks
Sa-CNN	Self-Attention Convolutional Neural Networks
CSRNet	Congested Scene Recognition Networks
CMTL	Cascaded Multi-Task Learning
CC-NN	Counting Convolutional Neural Networks
R-CNN	Region_based Convolutional Neural Networks
HMM	Hidden Markov Models

ABSTRACT

With real time surveillance being the need of the hour, Crowd Density and Crowd Behaviour Analysis can result in better protection and increased quality of services being offered. Urban Planning, crowd estimates, and quick responses to emergencies are some of the applications of this study. Furthermore, this study can be implemented to streamline crowd movements in crowded places and military applications. Many implementations have been presented in the same area, however, different environments might have noise, occlusions and cluttered areas can increase the complexity to analyse crowd density and crowd behaviour accurately. This study proposes a solution, which uses two different models to predict crowd density and crowd behaviour separately. For estimating crowd density, I have implemented M-CNN architecture[9], which takes into account the scale variation and has given accurate results for crowd density estimation. Furthermore, another model is used to analyse crowd behaviour, which uses crowd movement, heat maps and energy graphs[10] to precisely estimate the crowd behaviour.

Keywords: Crowd Density Estimation, Crowd Behaviour Analysis, Deep Learning, M-CNN architecture, Urban Planning

CHAPTER-1 INTRODUCTION

1.1 Introduction

In applications involving video surveillance, people counting is an important topic. Crowd surveillance being the need of the hour, estimating crowd density and crowd behaviour can increase protection and increased quality of life. However, simply counting the number of people manually in a crowd will be difficult and may not be sufficient to fully understand the situation, especially in scenarios where crowd behaviour may play a critical role. For example, in situations such as large protests or sporting events, it is important to not only to estimate number of people attending the event but also to analyse their behaviour in order to detect potential security threats or to improve crowd management strategies.

To address this issue, recent research has focused on developing techniques for estimating crowd density and crowd behavioural analysis using video surveillance systems. These techniques involve extracting features from video streams, such as crowd density, movement patterns, and activity levels, and using these features to to estimate the density and classify different types of behaviours.

While there are many challenges associated with crowd behavioural analysis, including issues related to occlusion, perspective distortion, and the complexity of crowd behaviour itself, however, recent advances in computer vision and machine learning have made it possible. Techniques such as deep learning, optical flow analysis, and trajectory clustering have been used to effectively extract and analyse crowd behaviour data from video streams.

In this project, we will explore the use of such techniques for crowd behavioural analysis, with a focus on developing a CNN-based model for crowd counting and behavioural classification.
And implemented M-CNN Architecture can accurately estimate crowd density. Further, we will also investigate the impact of various factors, such as camera placement, lighting conditions, and crowd density, on the performance of the model, and we will evaluate its effectiveness using standard evaluation metrics and real-world scenarios. By developing and testing a robust crowd behavioural analysis system, we hope to contribute to the development of effective crowd management strategies and enhance public safety in a wide range of applications.

1.2 Problem Statement

1.2.1 Problem Definition

Crowded places are a common thing in India, which has raised serious concerns for emergency management. Timely detection of any mishap can save many lives and infrastructure.
We need to develop a solution which utilises existing infrastructure and technologies to accurately estimate crowd density and crowd behaviour. For which, two separate deep learning models are utilised which can be used to monitor crowd behaviour and crowd densities through camera in real time.

1.2.2 Problem Analysis

The Tsai algorithm is used to calibrate the camera and provides a set of parameters for projection. The map plane created by the blob projection is known as the Ground plane. When an item is far from the camera, its projected size grows and is more obvious as the camera angle narrows. By putting the ground plane at the same height as the person and taking the intersection of those projections and a parallel plane, this can be minimised. The Head aircraft is the

name of the later plane. Blobs are created by filling the empty regions with morphological dilation. Practically speaking, the head plane should be changed in light of the next two details. 1) The twofold projection removes objects below the Head Plane. 2) The first predicted area problems at different distances are only partially managed by HPH[2] that is quite manageable.

It is a simpler approach for keeping track of the number in sizable gatherings. With the use of background removal, the foreground blobs are first and foremost detected during processing. The approach is based on many Gaussians. Algorithm tuning and morphological dilation of blobs are used to fill in the consistent gaps left by the inescapable statistical limitations of algorithms in order to produce homogeneous blobs.

The high level semantics required for crowd counting, which employs an architectural style akin to the VGG-16[4] network design, are captured by deep networks. The filters have applications in object segmentation, saliency prediction, and generic captioning and are very effective at doing so. By removing the fully connected layers used in VGG design[4], pixel level foresights are obtained to get around the issues with image categorization, where only one discontinuous label is given for the entire image.

1.3 Objectives

- Studying tools and techniques for crowd density estimation and crowd behaviour using deep learning.
- Implementing different deep learning techniques to estimate crowd density and crowd behaviour.
- Calculating the concentration of the crowd in witness films and analyse crowd behaviour is the goal of crowd density analysis.
- Creating a model which estimates crowd density and crowd behaviour using deep learning

1.4 Methodology

1.4.1 Basic steps in constructing a model :

1.4.1.1 Data Collection

- Datasets quantity and quality will determine how precise our model is.
- This stage often provides a representation of the information that will be used for training.
- Use pre-collected data, datasets from UCI, Kaggle, etc.

1.4.1.2 Data Preparation

- Accrue data and prepare it for training.
- Try to clean up everything that could need it (remove duplicates, fix errors, handle missing numbers, normalisation, convert data types, etc).
- Create data visualisations to assist in identifying significant correlations between class imbalances, variables or other exploratory analysis.
- Make separate sets for training and evaluating.

1.4.1.3 Choose A Model

- There are various algorithms for various tasks. Pick the best option.

1.4.1.4 Train The Model

- Training goal is to deliver an accurate answer or prediction as frequently as is practical.
- Every period in the process is a training one.

1.4.1.5 Analyse The Model

- Use a measure or group of measures to measure the model's objective performance.
- Although the model is currently being tuned, this unseen data is intended to be reasonably reflective of model performance in the real world.

1.4.1.6 Parameter Tuning

- By modifying the parameters, precision and accuracy of the model can be adjusted.
- Initialization settings, learning rate, training step count , and distribution, among these simple model hyperparameters, may be used.

1.4.1.7 Make Predictions

- In order to better understand how the model will function in the real world, extra data that has previously been withheld from it (and for which class labels are known) is used in its testing.

1.4.2 Proposed Methodology For Crowd Density Estimation

We propose a crowd density estimation model using deep learning and CNN to achieve better performance.

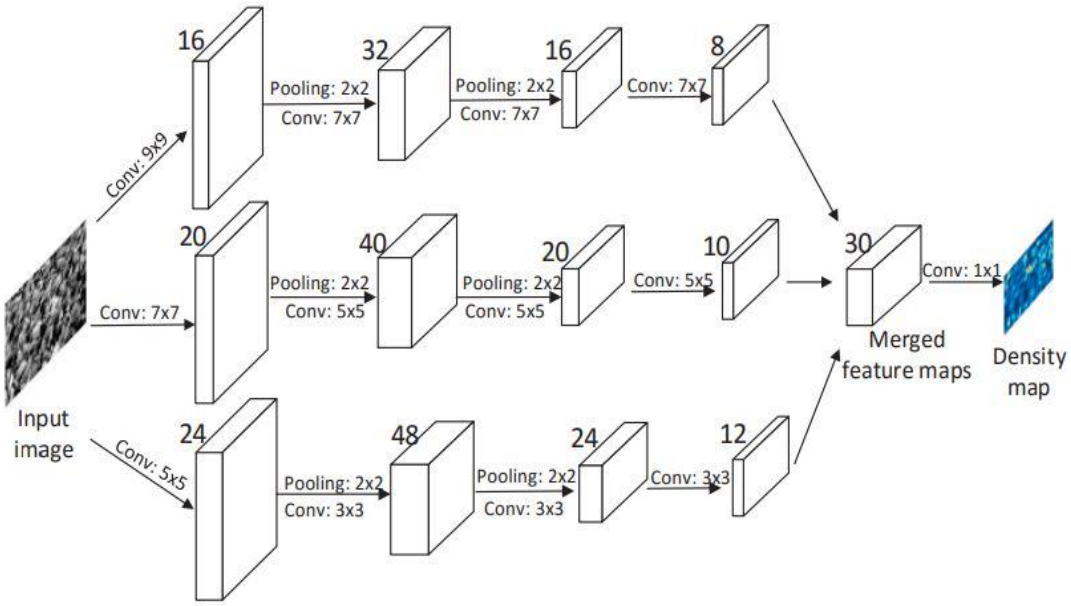


Figure 1. Proposed M-CNN Structure [9]

1.5 Organisation

This project report is divided into five chapters which are as follows:

Chapter 1:

In this chapter, the project is briefly introduced. A brief summary of the study and a description of Crowd Density Estimation using Deep Learning are provided in this chapter. This chapter also covers the project's objectives and the issue statement for the entire project. In this chapter, the project's model for categorising Crowd Density using Keras and transfer learning is described in detail, along with a brief overview of the project's methodology.

Chapter 2:

This chapter contains information about earlier research into the various types of crowd density estimation models and crowd behaviour analysis. In addition, this chapter discusses deep learning, machine learning, and neural networks. There have been numerous periodicals and related articles that detail previous work. The chapter described the numerous models that researchers have tried to use in order to create an efficient crowd density estimation model using deep learning. The methods we should use to train/create our model or project are determined by the tactics and results included in this chapter.

Chapter 3:

This chapter went into great length about the steps we will take to build the full project. Model development as well as system development are discussed. The chapter includes information on the data set that we'll be using. The chapter also contains all the information about the libraries we'll be using. It also gives information on the Convolutional Neural Network (CNN) that will be used. It explains the whole theory underpinning the convolutional neural network by outlining each element of the neural network. It offers information on the different neural network layers. The numerous accuracy measurements and validating tools are also covered. It

also provides information about the system required to start and run the project.

Chapter 4:

This chapter details the work that has been done on the project as a whole and how we have kept track of it at each level. It delivers information on the work done at different levels and also gives the results at different levels. It provides information on the model that we created using a number of modules and packages. It also contains the results of the many performance measures we used during the project. It provides information about the model's accuracy and the forecasts generated by the created model. The entire chapter contains information about the performance of our complete model or project.

Chapter 5:

This chapter contains the whole conclusion of the work that is the subject of this project report. Along with details on each project phase, the future scope of the project is also mentioned. It also offers information on how the project will be used and possible locations where it could be applied to improve computerization in that sector. It offers details on the project's improvements and what we can do moving forward in relation to this project and its improvement.

CHAPTER-2 LITERATURE SURVEY

Single-Image Crowd Counting via Multi-Column Convolutional Neural Network[8].
The purpose of this study is to develop a technique that can accurately determine the number of individuals in a crowd from a single image with arbitrary crowd density and arbitrary perspective. We have proposed a simple yet effective Multi-column Convolutional Neural Network (M-CNN) architecture to map the image to its population density map in order to achieve this.When utilising the recommended M-CNN, the input picture can be of any size or resolution. Due to the perspective effect or picture resolution, the characteristics learned by each column of CNN can be adjusted to account for variations in people's/heads' sizes by using filters with different-sized receptive fields.
Furthermore, the real density map is correctly computed using geometry-adaptive kernels, which do not need to be aware of the perspective map of the input image. As existing crowd counting datasets fall short in this regard, we used a dataset of 1198 photos with approximately 330,000 heads tagged to adequately cover all the problematic conditions taken into account in our work. We conduct extensive testing on this challenging new dataset as well as all current datasets to validate the effectiveness of the proposed model and method. In particular, using the recommended simple MCNN model, their technique outperforms all other methods.

It is impossible for filters with the same-sized receptive fields to capture crowd density features at various scales because of perspective distortion, which causes the heads in the photographs to typically be significantly different sizes. This makes learning the map using filters with different local receptive field widths from raw pixels to density maps more natural. We use filters of various sizes for each column in the proposed MCNN model to simulate the density maps corresponding to heads of varied scales. For instance, filters with larger receptive fields are advantageous when simulating density maps for larger heads.

Fast crowd density estimation with convolutional neural networks[9]
The proposed technique estimates crowd density using an enhanced convolutional neural network (ConvNet). There are two components to the contributions. For

estimating crowd density, convolutional neural networks are first constructed. Some network connections are deleted to significantly speed up estimation based on the fact that identical feature maps exist. Second, a cascade of two ConvNet classifiers has been created to improve accuracy and speed. In this article, a cascade-optimised convolutional neural network is proposed as a real-time technique for crowd density estimation.

The proposed method is built on the multi-stage ConvNet. The critical realisation that some network connections are not necessary leads to the implementation of the idea of deleting similar feature maps and their connections from the second stage.

By making connections simpler, the computational effort is significantly decreased and a lighter network is produced. Since there is no standardised data set for crowd density estimation, the technique is tested on three data sets: PETS 2009 (Ferryman and Ellis, 2010), a subway video (Ma et al., 2008, Ma et al., 2010), and a video of Chunxi Road in Chengdu (Zhou et al., 2012).

For Crowd Behaviour Analysis, several different techniques have been implemented, which can be broadly classified into:

- Tracking crowd movement/flow
- Following trajectories made
- Spatio-temporal Gradients

Based on the study performed by Tal Hassner et al.[9], the model estimates the optical flow of the crowd, by comparing two frames, which creates a violence flow vector that is further used for classification of crowd's behaviour as violent or not.

L Kratz et l in his paper "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models"[10], generates patterns of spatio-temporal motion which help in classifying the behaviour.

Sno.	Author Name	Dataset Used	Approach	Technology Introduced
1	[1] Sindagi Et al	ShanghaiTech	CP-CNN	Contextual pyramid CNN creates high-quality density maps by fusing global and local contextual data.
2	[2] Zhang Et al	UCF_CC_50	Sa-CNN	customises the feature maps that have been scaled down from numerous layers to the same size. The final density map is produced by combining them.
3	[3]Li Et al	ShanghaiTech, UCF_CC_50	CSRNet	uses dilated convolution layers in the backend network to increase the receptive field while preserving resolution
4	[4]Sindagi Et al	ShanghaiTech	CMTL	end-to-end cascaded framework that estimates the crowd density and generates a density map
5	[5] Rubio Et al	TRANCOS	CC-NN	Mostly using a regression function to map the look of the picture patches onto the

				appropriate item density maps
6	[6]Boominathan Et al	UCF_CC_50	CrowdNet	Shallow and Deep Network layers are used, the shallow network gather low-level features and deep network layers process high level semantic data.
7	[7] Sam Et al	UCSD	Switching CNN	Similar to M-CNN structure, this study trains several independent CNN to preserve most of the features.
8	[8]Zhang Et al	ShanghaiTech, UCF_CC_50	M-CNN	three branches of a multi-column architecture using various kernel sizes
9	[9]T. Hassner Et al	Violent Flow Dataset	Linear SVM	Based on crowd movements, violent flow vectors are generated, which are tracked
10	Kratz, Louis & Nishino, Ko. (2009)	Ticket Gate Dataset	HMM	Generates Spatio-temporal motion pattern which identify ‘unusual activities’

CHAPTER-3 SYSTEM DEVELOPMENT

3.1 MODEL DEVELOPMENT

3.1.1 OS

Operating system acts as an interface between software and hardware, in our case Operating system of choice was Microsoft windows 10 , with its easy feature, handy command prompt to run terminal commands and fast and secure development it was an easy choice, also it was the only OS available with us, but it does not changes the fact that even if we had alternatives like linux or mac we would have selected them, by importing system library in python we can easily integrate system file, read and write commands or read system date and time or other features necessary for the development of this prototype for our major project. open() can be utilised to simply open files in the file directory of our computer and os.path command cas be used to change its director. We can also create temporary files for intermediate usage and delete them later once we have obtained final results.

3.1.2 NUMPY

The Python package NumPy is used to manipulate arrays. For usage in the discipline of linear algebra, functions, the Fourier transform, and matrix operations are also given. Numpy, an acronym for "Numerical Python," is a package that contains multidimensional array objects and several methods for handling such arrays. For logical and mathematical operations on arrays, NumPy is commonly used. Additionally, it covers various array methods, indexing strategies, etc. You may use it for nothing because it is an open source project. NumPy is the name given to Python used in math. The goal of NumPy is to offer array objects that are up to fifty times faster than typical Python lists. The NumPy array object is symbolised by the nd array object. Unlike lists, NumPy arrays are stored in a single unified region of memory, making it relatively easy for programs to access and modify them. In computer science, this quality is known as location of reference. This is the primary factor that makes NumPy

faster than lists and is extremely important to us. Additionally, it has been enhanced to support current CPU architectures.

3.1.3 Random

Python's Python Random module is used to generate random numbers. Kindly note that these numbers are not actually random they are pseudo random i.e. generated using a non-random parameter.

3.1.4 Keras

For creating and analysing efficient and user-friendly deep learning models, Keras is open source and simple to use. We introduce Theano and TensorFlow, two frameworks for fast numerical computation, to design and train neural network models. It uses independent machine learning toolkits as well as C#, Python, and C++ libraries.

Although incredibly powerful tools for building neural networks, Theano and TensorFlow are also difficult to comprehend. TensorFlow or Theano-based deep learning models may be easily and quickly constructed using the fundamental Keras framework. With Keras, deep learning models can be defined quickly. Of course, Keras is the best choice for applications that call for deep learning.

3.1.5 GLOB

The glob module is part of the Python standard library. The glob (short for global) function returns all file paths matching a given pattern. If the filename fits a specified pattern, we may search for files that include those files using wildcard characters, which may be more useful than using glob.
Even while glob may be used to search for a file by name, I found that it works best when reading in numerous files with similar names. Then, for further analysis, these files may be found and combined into a single data frame.

3.1.6 TensorFlow

Google created and made accessible TensorFlow, a well-liked Python library for quick numerical applications and calculations. Both novices and experts can benefit from the TensorFlow course. Our lesson covers all fundamental and advanced topics in deep learning and machine learning, such as sentiment analysis, deep neural networks, and image processing. TensorFlow, one of the popular deep learning frameworks, was developed by the Google Team. TensorFlow is free and open source software that was developed in the Python programming language, therefore this guide was made to make it simple to construct a deep learning project using it.

3.1.7 Matplotlib

Library used for visualisation of the data and can further be used for static and interactive visualisations.For 2D displays of arrays, Matplotlib is a fantastic Python visualisation package. A multi-platform data visualisation package called Matplotlib was created to deal with the larger SciPy stack and is based on NumPy arrays. In the year 2002, John Hunter first presented it.
One of visualisation's biggest advantages is that it gives us visual access to vast volumes of data in forms that are simple to understand. There are numerous plots in Matplotlib, including line, bar, scatter, histogram, etc.

3.1.8 Pytorch

A popular open-source machine learning package called PyTorch is based on the Torch library. It is written in Python, which makes it simple to use, interact with other Python libraries, and is created and maintained by Facebook's AI Research team. For study and experimentation, PyTorch provides a dynamic computational graph that enables developers to create and alter neural networks on-the-fly. Additionally, it has comprehensive GPU acceleration support, which greatly accelerates the training and inference of deep neural networks. Overall, PyTorch is a strong tool for creating machine learning models and is increasingly used by academics and industry professionals.

3.1.9 Scikit-learn

A well-liked machine learning toolkit for Python called Scikit-learn offers a variety of tools for preprocessing data, classifying objects, predicting future outcomes, grouping data, and choosing models. It offers a user-friendly interface and is built on top of other scientific computing libraries like NumPy, SciPy, and matplotlib, making it simple to apply machine learning techniques to practical issues. Support vector machines, decision trees, random forests, k-means clustering, and principal component analysis are just a few of the supervised and unsupervised learning methods that are supported by Scikit-learn. Additionally, it offers performance indicators and cross-validation as model evaluation tools. Generally speaking, scikit-learn is a robust and adaptable Python machine learning package that has grown in popularity among data scientists and machine learning professionals.

3.1.10 YOLOV5

Deep learning object detection model YOLOv5 (You Only Look Once version 5) has become well-known for its high accuracy and quick inference speed. Without the aid of area proposal networks or anchor boxes, it is a single-stage detector that can predict item bounding boxes and class probabilities from input photos alone. With a lightweight design and few parameters, YOLOv5 is simple to train and deploy on devices with limited resources. It is especially helpful for detecting the presence of individuals in crowds because it can do it rapidly and precisely. Additionally, YOLOv5 supports a number of backbone topologies, including S, M, L, and X, which present various trade-offs between accuracy and speed.

Overall, YOLOv5 is a strong and adaptable object detection technique that has gained popularity for real-time computer vision applications including crowd density detection.

3.1.11 PyCUDA

Users can quickly use the CUDA parallel computation API from Python using the PyCUDA package. NVIDIA created CUDA, a parallel computing framework and

application programming interface (API), for multipurpose GPU computing. PyCUDA offers a straightforward interface for Python programmers to compile and run CUDA code, making it simple to use GPU power for computationally demanding tasks like deep learning and scientific simulations. PyCUDA allows users to build GPU-accelerated Python code without having to be an expert in CUDA or low-level programming languages like C or C++. PyCUDA is a well-liked option for individuals looking to speed their Python-based calculations using GPUs because it is well-documented, constantly developed, and has a sizable user community.

3.1.12 HOG+SVM

A well-known machine learning method for classifying and detecting objects is HOG+SVM. HOG, which stands for "Histogram of Oriented Gradients," is a feature descriptor that records an image's local gradient data. The machine learning technique known as "Support Vector Machine" (SVM) learns to classify data points by locating the ideal hyperplane that divides the classes.

In the HOG+SVM method, an SVM classifier uses the feature vector produced by computing the HOG feature descriptor from the input image to determine whether an item is present in the image. The detection of pedestrians, faces, and other objects in photos has been accomplished with success using this method.
HOG+SVM does have some restrictions, though. It could not be very good at recognising objects of various sizes or orientations, and it might be sensitive to changes in lighting conditions. Furthermore, it could not be as accurate as methods based on deep learning, such as YOLOv5 or Mask R-CNN. However, HOG+SVM can be an excellent option for applications where real-time efficiency and simplicity are critical for object detection in some situations.

3.1.13 Mask R-CNN

An object recognition, segmentation, and instance segmentation deep learning model are called Mask R-CNN (Region-based Convolutional Neural Network with Masking). It is a modification of the well-known Faster R-CNN architecture that gives the network

a mask branch so it can forecast pixel-level segmentation masks for each object found in an image.

Mask R-CNN is very helpful for locating and following individual objects in a cluttered setting, such individuals in a crowd. Even when partially obscured or close to other objects, it can reliably detect and segment objects of different sizes and orientations.

Mask R-CNN needs a lot of labelled training data and substantial computational power for both training and inference, just as other deep learning models. On a range of object detection and segmentation tasks, it can obtain state-of-the-art results after being trained.

Mask R-CNN is an effective tool for object detection and segmentation in crowded situations and is frequently used in computer vision applications including autonomous driving, surveillance, and robotics.

3.2 Datasets Used

3.2.1 ShanghaiTech Dataset

A sizable crowd counting dataset called the ShanghaiTech dataset was published in 2016. There are 1198 annotated crowd photos in total, split into Part-A and Part-B. There are 482 photos in Part-A and 716 images in Part-B. While the photos in Part-B were taken on Shanghai's crowded streets, the photos in Part-A were taken from the Internet. A location in the centre of the head is labelled on each individual in a crowd shot. The collection has 330,165 annotated individuals in total.

For scientists researching crowd counting methods, the ShanghaiTech dataset is a useful tool. It covers a variety of crowd settings and is one of the most frequently used dataset due to the many scenarios offered by this dataset and most difficult crowd counting datasets available. Several cutting-edge crowd counting techniques have been trained and evaluated using the dataset.

The following are some advantages of using the ShanghaiTech dataset:

- It is a huge and difficult dataset.
- Numerous different crowd scenarios are covered.
- It is simple to use and well-organised.
- Many modern crowd counting algorithms have been trained and tested using it.

3.2.2 UCSD Dataset

Videos of congested pedestrian pathways make up the UCSD Anomaly Detection Dataset. In 2010, scientists at the University of California, San Diego (UCSD) developed it. 50 video segments totaling 30 seconds each make up the dataset. A stationary camera set at a height and looking down on pedestrian pathways was used to record the recordings. The walkways can have varying densities of people, from few to many. Only pedestrians are shown in the video at its regular setting. Odd things happen for one of two reasons:

the movement of non-pedestrians (such as bikes, skaters, and small carts) through the walkways.
abnormal pedestrian movement patterns, such as those involving running, jumping, or collisions.
Annotations at the pixel and frame levels make up the dataset's ground truth. Whether an anomaly is present at a certain frame is indicated by the annotations at the frame level. The regions with abnormalities are identified using the pixel-level annotations.

Researchers studying anomaly identification in crowded environments can benefit greatly from the UCSD Anomaly identification Dataset. One of the first datasets to offer frame- and pixel-level annotations for anomaly identification in congested environments, it offers both types of information. A number of cutting-edge anomaly detection algorithms have been trained and evaluated using the dataset.

The following are some advantages of utilising the UCSD Anomaly Detection Dataset:

- It is a huge and difficult dataset.

- Numerous different crowd scenarios are covered.
- It is simple to use and well-organised.
- Several cutting-edge anomaly detection algorithms have been trained and tested using it.

3.2.3 Violent Flow Dataset

Database is a collection of more than 240 video footage of crowd violence, with each video of dimension 224 X 224. Dataset is designed to train models which classify crowd's behaviour as violent and non-violent. Violent Flow Database include videos of length ranging from 1 second to 7 seconds, while average duration of a video in this dataset is 3.6 seconds.

3.2.1 Examining Dataset

A Data Quality Assessment, a distinct phase in the data collection process, is used to confirm the origin, amount, and effect of any data items that do not adhere to accepted data quality standards. quality of data lifecycle. To assure the quality of the data, the Data Quality Assessment can be performed once after a certain amount of time as part of an ongoing project or only once.

The quality of your data might rapidly decrease with time, even with stringent data collection protocols that clean the data as it enters your database. Due to factors like people moving homes, changing phone numbers, and passing away, the information you have might soon become outdated.

Finding records that have become erroneous, as well as the source of the data and any possible implications that inaccuracy may have had, is made easier with the use of a data quality evaluation. With the help of this evaluation, it may be rectified and potential new issues discovered.

3.2.2 Preprocessing and Outlier deletion

Data cleansing is one of the core elements of Machine Learning. It is required for the development of models. It is hardly the most sophisticated part of machine learning, but there are no secrets or puzzles either. There are also no unknown twists or mysteries.

But if done correctly, data cleaning might make or kill your project. This stage frequently takes up a considerable percentage of the time that professional data scientists spend on it. And because the adage "Better data trumps sophisticated algorithms" is true.

Even a simple process can get the desired results given a clean dataset, which can occasionally be quite helpful. It should be obvious that various data kinds would call for various cleaning methods. But generally speaking, this methodical approach is a great place to start.

3.2.3 Data Transformation and normalisation

Actually, the data has already been cleansed and smoothed. On the other side, data transformation defines the steps required to convert the data into a machine-usable learning format. Data transformation is the process of merging data from its initial, indistinct, segmented, and normal source condition, modelling it dimensionally, de-normalizing it, and making it available for analysis.

Without the right technical stack in place, data transformation may be expensive, time-consuming, and arduous. However, conversion will ensure the best data quality, which is necessary for accurate analysis and the generation of valuable insights that will eventually allow for data-driven choices.

Making and training models to analyse data is a brilliant concept, and more companies are utilising or planning to utilise machine learning to handle a range of real-world applications. However, for models to learn from the data and generate valuable predictions, the data must be organised in a way that its analysis yields meaningful insights.

3.3 Learning Algorithms

3.3.1 Supervised Learning

Supervised learning consists of a pair that includes the supervisory signal (also known as the input item), which is frequently a vector, and the desired output value. An inferred function from the training data is produced by a supervised learning technique and may

be used to map new samples. The algorithm will be able to precisely identify the class labels for samples that are not yet obvious in an ideal setting.
Finding the mathematical connection between the input, X and output, Y variable is a challenge for machine learning. To learn how to predict the outcome from the input, a model is built using these X, Y pairings of labelled data. supervised learning challenges include issues with regression and classification.

3.3.2 Unsupervised Learning

Unsupervised learning takes unlabelled data as input and X input variables. The learning algorithm uses unlabeled data from such X variables to predict the underlying structure of the data. Unsupervised learning problems include clustering and association concerns.

3.3.3 Reinforcement learning

A subset of machine learning is reinforcement learning. It comprises taking prudent action to maximise benefit in a certain circumstance. Many computer programmes and machines utilise it to decide what action to take in a given circumstance.
Reinforcement learning depends on the reinforcement agent to decide what to do in order to finish the job at hand as opposed to supervised learning, which incorporates an answer key in the training data. In supervised learning, the model is provided with labelled data whereas in reinforcement learning, no labelled data is provided, but instead feedback from the user is obtained. It is necessary for it to draw lessons from its past experiences in the absence of a training dataset.

3.4 Models Utilised

3.4.1 CNN (Convolution neural Networks)

An artificial neural network known as CNN excels at identifying and interpreting patterns. CNN has so far proven to be the most useful for categorising images.

A CNN model may employ various numbers and sizes of filters. The key element that enables us to find the pattern is these filters. Convolutional neural networks, or CNNs for short, are specialised neural network models which are created for use with two-dimensional visual input, however they may also be used for one-dimensional and three-dimensional data.

The convolutional layer, which gives the network its name, is a key part of a convolutional neural network. The most often utilised layers in CNN models are convolutional and pooling layers. CNN is a useful solution for problems involving photo categorization since it performs better with data that are represented as grid structures. By deactivating a portion of the neurons during training, the dropout layer reduces the likelihood of the model overfitting.

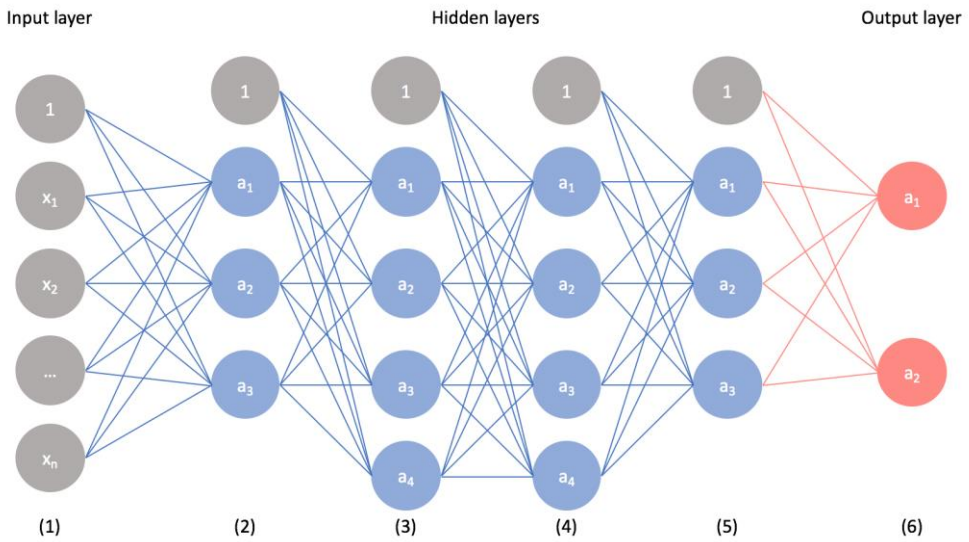


Figure 2. Basic CNN architecture [9]

3.4.1 Functioning of CNN

Convolutional neural networks are made up of many layers of artificial neurons. Artificial neurons are mathematical operations that compute the weighted sum of a collection of inputs and output an activation value, much like their biological counterparts.

The behaviour of each neuron is governed by its weight. When given pixel values, CNN's artificial neurons can recognise a variety of visual properties.

Each layer in a ConvNet produces a number of activation maps when you feed it an image. Activation maps draw attention to the image's most important components. Each neuron receives the colour values of a pixel patch as input and multiplies them by its weights before adding them all up and sending the sum via the activation function

3.4.1.1 InceptionV3

This model was developed as a combination of several ideas and has proven to be more than 78% accurate on the ImageNet dataset.

Over time, researchers have evolved. According to Figure below, this model contains fewer than 25 million parameters. Convolutions, average pooling, max pooling, concatenations, dropouts, and entirely linked layers are the structure that make up a model itself. The model heavily employs batch normalisation, which is also applied to the inputs for activation. Softmax is used to compute the loss. We imported this pre-trained model for the project using Keras and transfer learning.

Additionally, you chose the imagnet weights, provided a unique 299 x 299 input shape for the model, and changed the include top value to false. A dropout layer, a flatten layer, and a thick layer were the next layers we added using sequential.

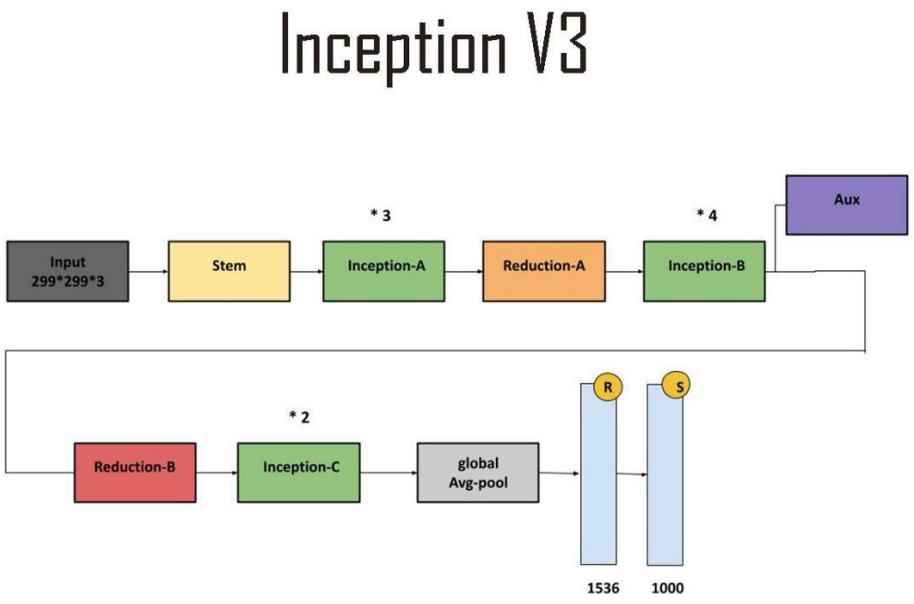


Figure 3. Inception V3 [10]

For our model, we have used AdamOptimizer and categorical cross entropy as loss function.

3.4.1.2 DenseNet-201

The Figure illustrates the 201 layers of this CNN model. Its trained model is able to categorise photos into 1000 different object categories. DenseNet was developed especially to improve the vanishing gradient error introduced by high-level neural networks.

Similar to that, we imported this pre-built model using Keras and transfer learning. Additionally, give the model a 299 x 299 input shape, utilise the imagnet weights, and set the include top value to false. Next, we added a dropout layer, a flatten layer, a repeat vector layer, another flatten layer, and lastly a dense layer using sequential. The loss function in our model, category cross entropy, was then used to build our model. Adam serves as the optimizer.

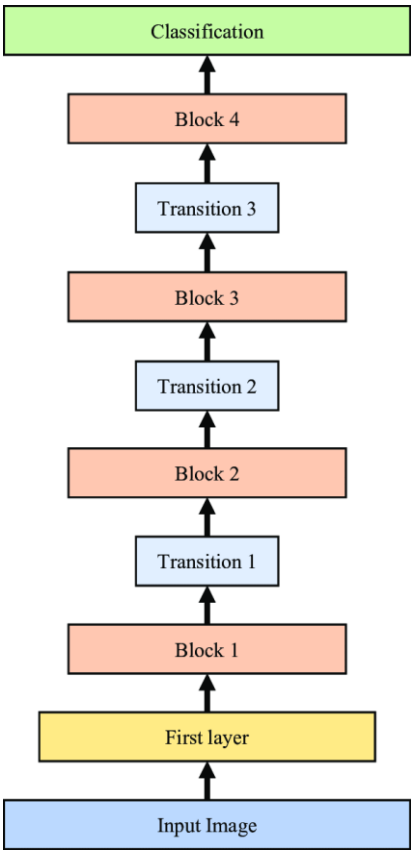


Figure 4. DenseNet-201 [11]

3.4.1.3 VGG 16

The Figure depicts this model, which has 16 layers total, and it was utilised to triumph in the 2014 ILSVRC ImageNet competition. There are over 138 million parameters in this model. Many deep learning image classification issues employ the VGG16 approach, however smaller network designs are often preferred. But because it's so simple to use, it's a terrific foundation for educational activities.

Using Keras and transfer learning, we imported the pre-trained VGG 16 model for the project. Additionally, give the model a 299 x 299 input shape, utilise the imagnet weights, and set the include top value to false. We added other layers, such as GlobalAveragePooling2D, dropout layer, two dense layers, a dropout layer, and then a dense layer after freezing part of its layers. Our model uses AdamOptimizer and categorical cross entropy as the loss function.

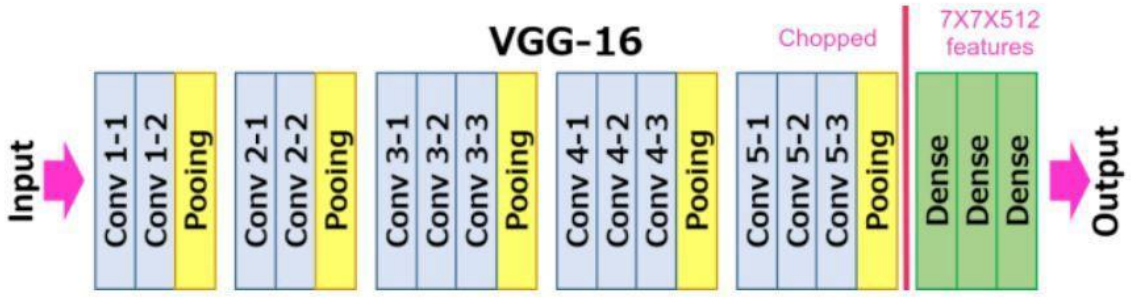


Figure 5. VGG 16 architecture [12]

3.4.1.4 VGG 19

A pre-trained version of the network that has been trained on more than a million photos is available in the ImageNet database. The pretrained network can distinguish a range of animal species among the 1000 objects, as well as other object kinds including a keyboard, mouse, and pencil.

Using Keras and transfer learning, we imported the pre-trained VGG 19 model for the project. Additionally, give the model a 299 x 299 input shape, utilise the imagnet weights, and set the include top value to false. And then we added 27 more layers—GlobalAveragePooling2D, a dropout layer, two dense layers, a dropout layer, and then a dense layer—freezing some of its layers in the process.

Then, using AdamOptimizer and categorical cross entropy as the loss function, we constructed our model.

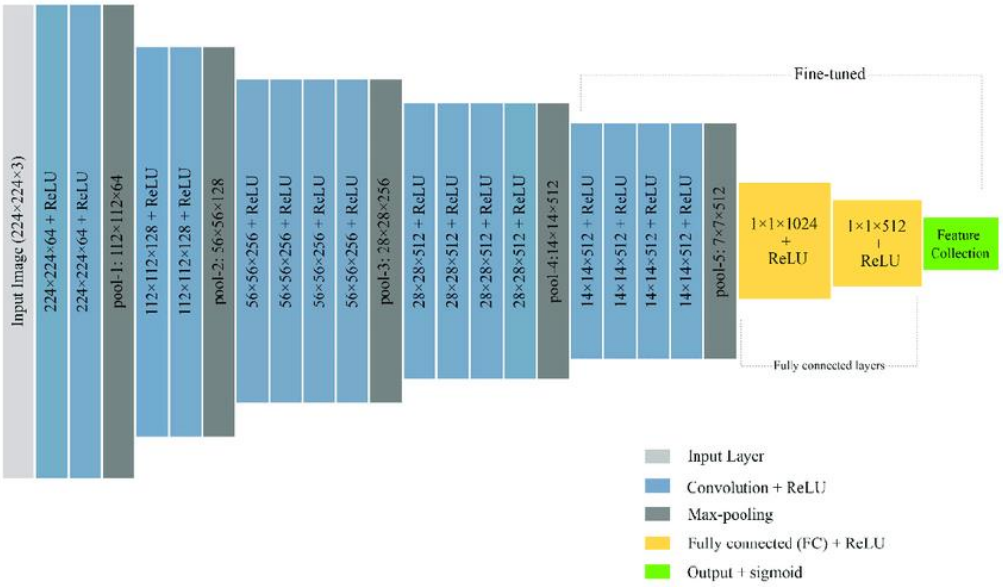


Figure 6. VGG 19 architecture [13]

3.4.1.5 ResNet-152

Understanding the residual representation functions, as illustrated in below figure, is preferable to examining the signal representation directly. ResNet might have a network that is up to 152 layers thick. This model contains over 60M parameters. We imported the pre-trained ResNet 152 model for the project using Keras and transfer learning. Additionally, you utilised the imagnet weights, provided a 229 x 229 input shape, and changed the include top value to false. It already has layers built in, so after partially freezing its layers, we added additional layers—one flatten layer and two thick layers.

Then, we built our model using categorical cross entropy as the loss function and Adam as the optimizer.

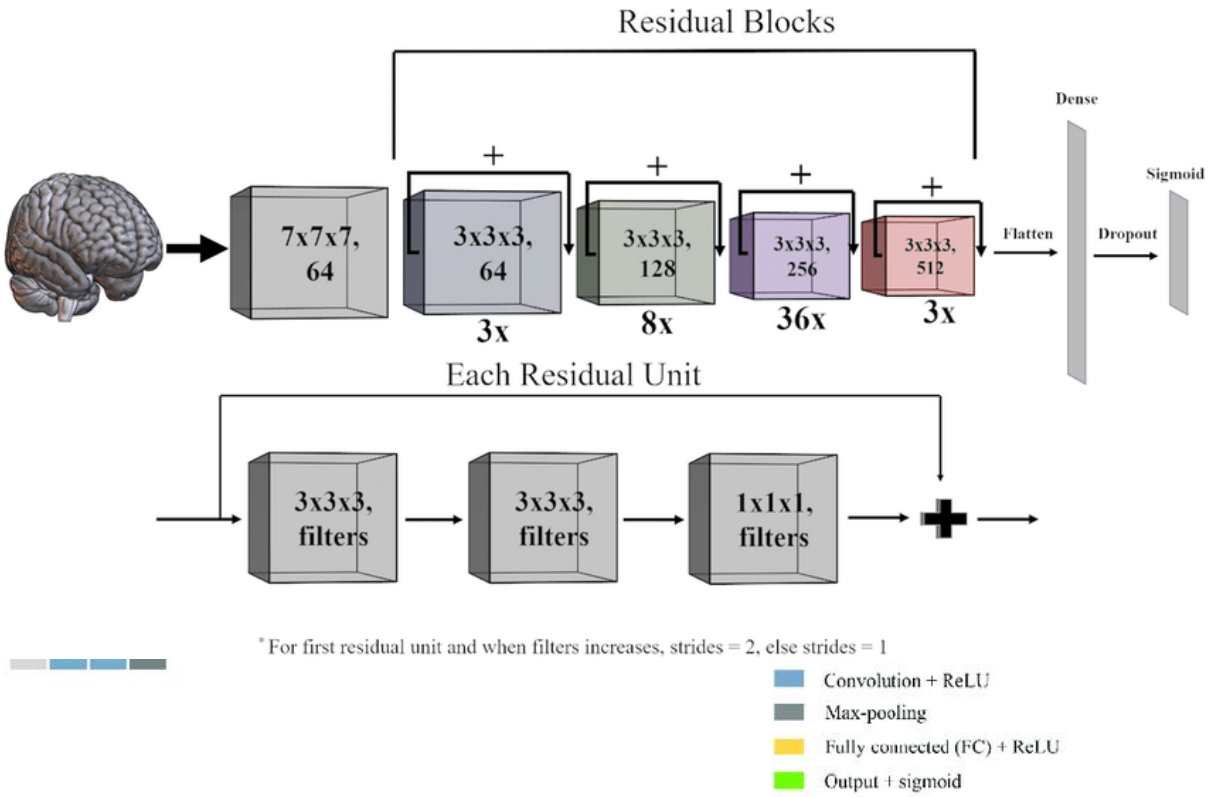


Figure 7. RasNet 152 architecture [14]

3.4.2 CNN architecture

The processing of visual images is a common application for CNNs, a subtype of Deep Neural Networks. Certain elements in photographs may be recognised and categorised by CNNs. They are employed in computer vision, image classification, image analysis for therapeutic purposes, and image and video recognition.

In mathematics, convolution is the act of multiplying two functions to produce a third function that signifies the change in shape of one function that is altered by the other. This mathematical process is referred to as a convolution by CNN. In order to extract features from a picture, two images that could be represented as matrices are multiplied to create an output.

Each input image will technically be processed through a sequence of convolution layers with filters (Kernels), Pooling, fully connected layers (FC), and apply SoftMax function in order to categorise an item with probabilistic values between 0 and 1. This is done in order to develop and evaluate CNN deep learning models. The flowchart that follows demonstrates how CNN examines an input picture and categorises the elements based on values

3.4.2.1 Basic Architecture Used

An architecture for a convolutional neural network consists of two primary components.

- Utilising the feature extraction method, a convolution tool may separate and identify the distinctive features of a picture for study.
- A fully connected layer that establishes the picture class using the features that were previously extracted and uses the output of the convolutional process.

3.4.3 CNN Layers

Since these layers recur often, our neural networks are deep, so, these types of structures are known as deep neural networks.

- Input given are raw pixel values.
- Convolutional layer: The input layers transform the results given by the neuron layer as output.
- The kernel size that will be applied to the input must be specified. Only 5x5 window that moves across the input raw pixel values and preserve pixels with maximum intensities are allowed for each filter.
- Rectified linear unit [ReLU] layer: this layer provided the opportunity to activate the gathered picture data. The ReLU function is used in back propagation scenario to prevent any changes that occur in pixel value.
- A volume down sampling procedure is used for the length and breadth of the dimensions by the pooling layer.

- FC layer: The maximum score generated of the input digits may be found after concentrating on the scoring class.

As we delve further and deeper into the levels, there is a notable increase in complexity. The trip would be valuable, though, because while precision could advance, time consumption, regrettably, does as well.

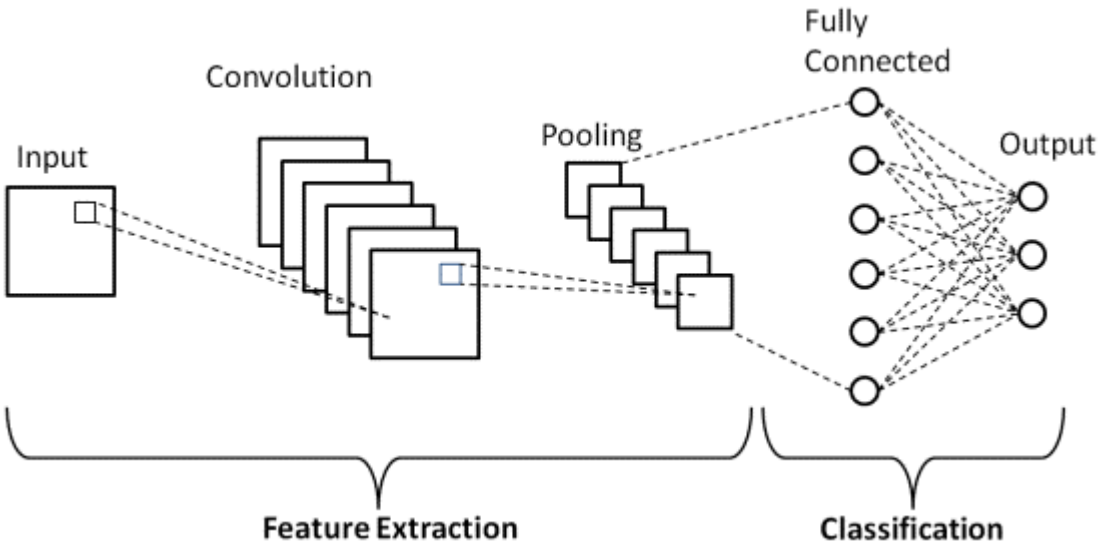


Figure 8. CNN layers [15]

1. Convolutional Layer

This layer's initial action was to extract the various attributes from the incoming photos. A MxM filter is utilised as the input at this layer, and mathematical convolution is done between the two. The dot product between the filter and the elements of the input image is formed (MxM) by swiping the filter over the input image in accordance with the filter's size.

The feature map provides a detailed description of the corners and edges of the image. This feature map will later be made accessible to other layers so they can add additional features from the source image.

2. Pooling Layer

After a convolutional layer, a pooling layer is typically used. This layer's goal is to minimise the size of the generated feature map in order to lower the computation cost.

Less linkages between layers and separate changes to each feature map are used to achieve this.

There are numerous different pooling procedures, depending on the technology used. The most important factor in Max Pooling is minimizing the size of the feature map. Segments of an input image segment of a particular size are averaged out by using average pooling.

Sum pooling is used to determine the cumulative sums of the raw values inside the selected segment of the image. The Pooling Layer frequently functions as a bridge between the FC Layer and the Convolutional Layer and help in reducing the dimensions.

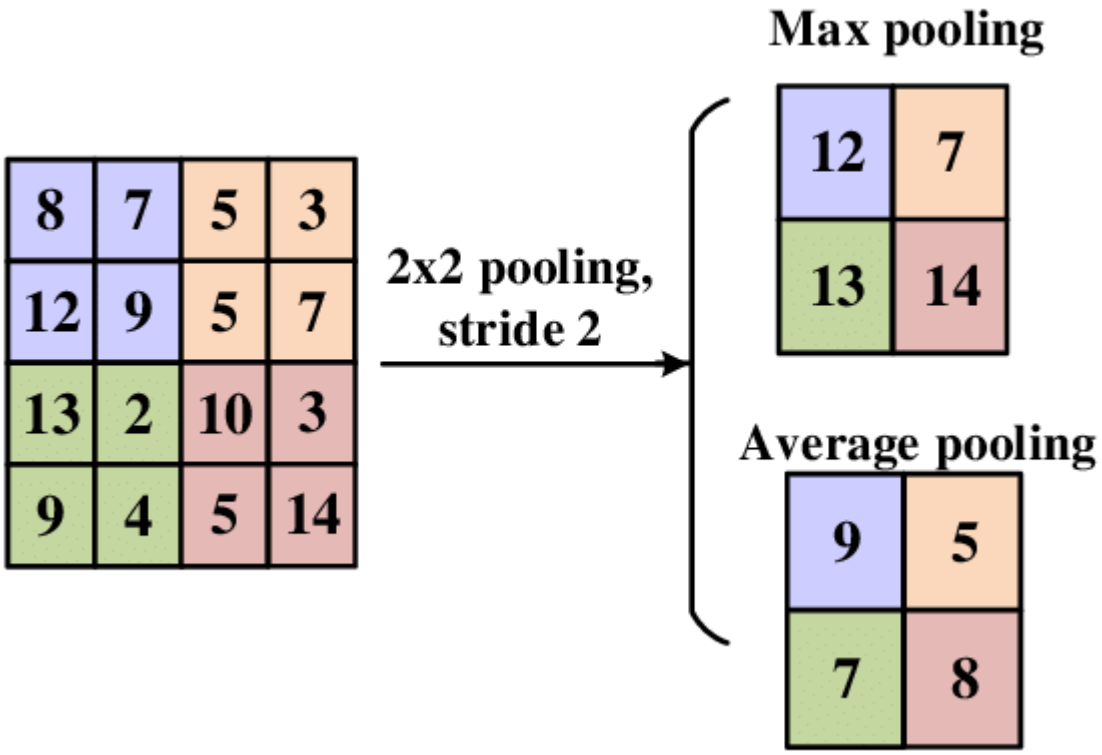


Figure 9. Pooling Layer [16]

3. Fully Connected Layers

The Fully Connected (FC) layer, which also has weights and biases, connects the neurons between two layers. In a CNN architecture, these layers are frequently positioned before the output layer.

The input picture from the layers below is given to the FC layer in a flattened form as a result. The vector is flattened before going through many FC stages of processing, where high-quality calculations on mathematical functions are carried out. At this point, the classification process is crucial.

4. Dropout

Normally, overfitting in the training dataset may be caused by all features related to the FC layer. When a model performs so well on training data that it has a detrimental impact on how well it works on fresh data, this is referred to as overfitting.

A dropout layer is employed to address this problem, which shrinks the size of the model by removing a few neurons from the neural network during training. 30% of the nodes in the neural network are arbitrarily eliminated at a dropout threshold of 0.3.

5. Activation Functions

A neural network's activation function demonstrates how a node or nodes in a layer of the network convert the input's weighted sum into an output. Numerous activation functions in the layer or network architecture exhibit nonlinear behaviour, also called "nonlinearity." It may also be referred to as a "squashing function" if the output range of the activation function is restricted.

The neural network's capacity and performance are significantly impacted by the choice of activation functions. Different model components may use various activation mechanisms.

Although networks are supposed to employ the same activation function for all nodes in a layer, in practise, the activation function is performed either before or after each network node's internal processing.

A network can have three different types of layers: input layers, which take in domain-specific raw data, hidden layers, which transfer data from one layer to another, and output layers, which provide predictions.

In many cases, the activation function is the same for all buried layers. The sort of prediction the model needs to make will determine the activation function used in the output layer, which is frequently different from the hidden layers.

One can discover the first-order derivative for activation functions that are typically differentiable from an input value. This is crucial because neural network training

techniques that involve backpropagation of errors call for the derivative of the prediction error in order to modify the model's weights.

One of the most important components of the CNN model is the activation function. To find an approximation, any kind of complex continuous relationship between network variables is used. Simply said, it decides which model information at the network's end should be advanced and which should not.

The network then departs from linearity after that. The ReLU, SoftMax, tanH, and Sigmoid functions are a few of the frequently utilised activation functions. Each of these acts serves a certain objective. A CNN model containing the sigmoid and SoftMax functions is used for binary classification, whereas SoftMax is commonly utilised for multi-class classification.

- A convolution tool is used in the feature extraction process to identify and isolate the distinctive features of a picture for analysis.
- A fully connected layer that classifies the picture using the previously recovered features and the results of the convolutional process.

Selected Applications:

1. Currently, we have sophisticated CNN models, such as Fast R-CNN and Faster R-CNN, for object detection. R-CNN is frequently used as the main pipeline for object detection models in 34 automated cars, as well as for face recognition and other purposes.
2. Semantic Segmentation: In order to better segment images using rich data, a group of Hong Kong researchers developed the Deep Parsing Network in 2015. In addition, researchers at UC Berkeley developed fully convolutional networks that advanced semantic segmentation to the cutting edge.

3.5 Building Ensemble Models

We employed 5 CNN models that were pre-trained with the aid of the Keras Library and Transfer Learning, as shown in model development and selection (3.3.2). Inception V3[10], DenseNet-201[11], VGG-16[12], VGG-19[13], and ResNet-152[14] are the models we employed. These models had completed weight training for ImageNet. Upon further examination of each model's performance, we discover that the VGG-16 [12]

and VGG-19[13] models have the greatest performance or accuracy levels exceeding 95%. So, for our ensemble model, we choose to employ these two models.
We combined the selected models and averaged them using a straightforward ensemble approach. Our performance was improved by the resulting model.

CHAPTER-4 PERFORMANCE ANALYSIS

4.1 Analysis of system developed

4.1.1 MCNN network for Crowd Density Estimation

Filters with the same-sized receptive fields are ineffective at capturing crowd density features at different scales because perspective distortion commonly results in images with heads that are somewhat different sizes.

As a result, it makes more sense to generate the map from the raw pixels to the density maps using filters with various local receptive field widths.

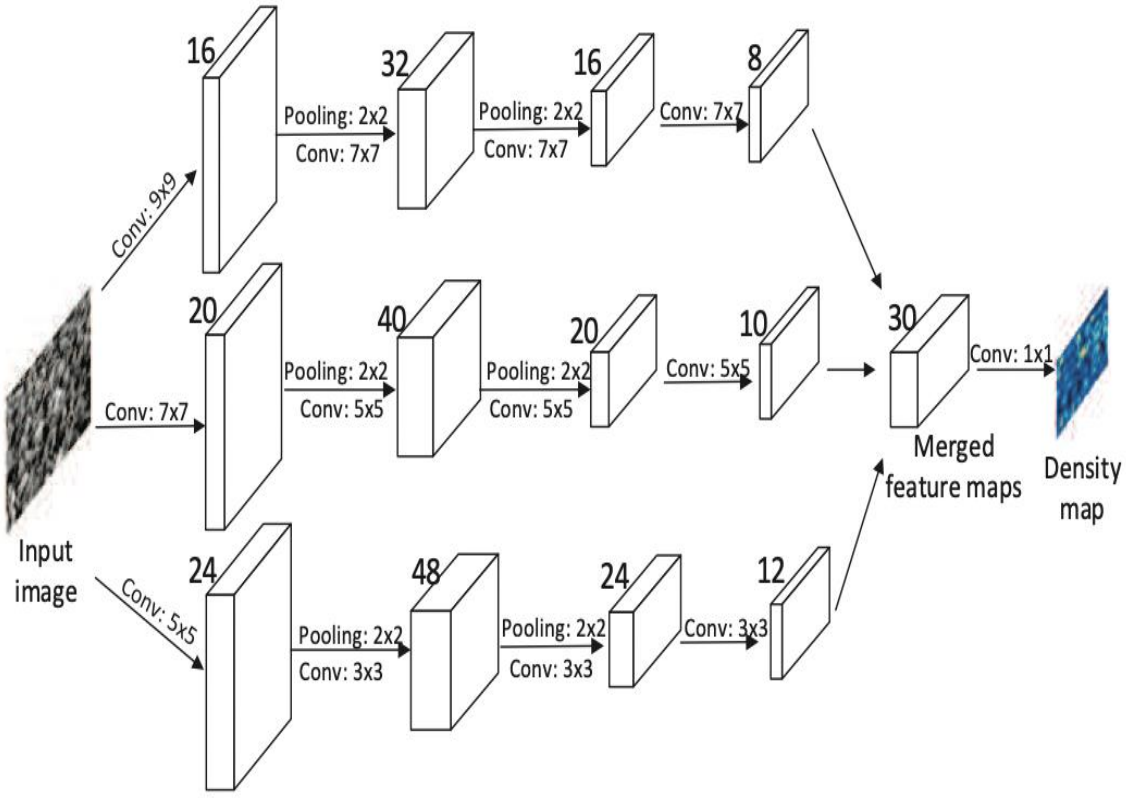


Figure 10. M-CNN Structure [9]

4.1.1.1 Performance benefits of MCNN over other models:

Benefit of using a MCNN model for estimation is the ability to train the filters to imitate the density maps of different-sized heads. Therefore, if the model has been trained on a large dataset with a range of crowd head sizes, it can easily be applied to a fresh dataset with crowd heads.

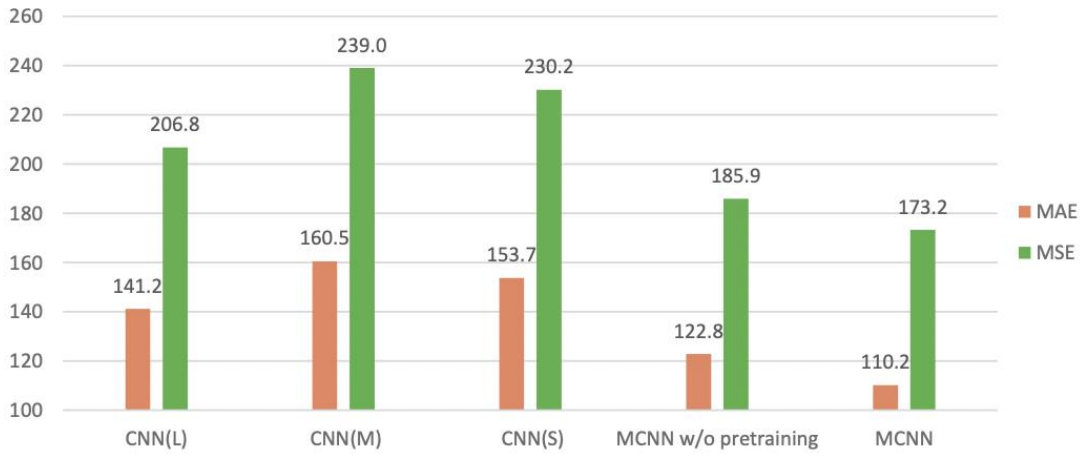
If the target domain only contains a small number of training samples, we may simply fix the first few layers in each column of our MCNN and concentrate on optimising the last few convolutional layers. In this case, optimising the last few layers offers two advantages.

The models can be adjusted to the target domain by fine-tuning the final few layers after the knowledge gained in the source domain is first preserved by fixing the first few layers. Therefore, accuracy may be improved by combining information from the source domain and the target domain. Second, in addition to fine-tuning the whole network, doing so also considerably decreases the computational complexity in the last few layers.

It uses three parallel CNNs as filters, each with a local receptive field of a different size. To make things simpler, we use the same network architecture (i.e., conv-pooling-conv-pooling) for all columns, with the exception of the widths and number of filters. We use fewer filters for CNNs with larger filters to reduce the computational complexity (the number of parameters to be tuned). We stack all of the output feature maps from the CNNs and map the density map to them.

4.1.1.2 Performance benefits of our model (MCNN) over CNN:

As can be shown, MCNNs perform much better for MAE and MSE than everyone single column CNN. This demonstrates the MCNN architecture's accuracy.



Graph 1. Performance of different architectures

4.1.1.3 Performance evaluation on transfer learning:

In order to demonstrate the applicability of the learned model in our method, we evaluate our technique in a transfer learning situation using the Part A of ShanghaiTech dataset as the source domain and the UCF_CC_50 dataset as the target domain.
To create an MCNNs model, we explicitly utilise data from the source domain. In the target domain, we examine two conditions for the crowd counting task: Condition I, which includes no training examples, and Condition II, which includes a small number of samples. We use our assessment model, which was developed using Part A of the Shanghaitech dataset, directly in scenario I. Using training data from the target domain, we modify the network.

The last two layers of the MCNN can be changed to considerably improve accuracy using training data from the UCF_CC_50. Performance suffers greatly if the entire network is modified as opposed to simply the top two levels. The UCF_CC_50 dataset's insufficient training sample size may be the cause of the performance discrepancy between tuning the entire network and the final few layers. By keeping the first several levels of the model and tweaking the last two layers, the output of the model is appropriate for the target domain and the useful features/filters that have been learned from sufficient data in the source domain are kept.
The learnt model, however, resembles that created using only the training data in the target domain if the entire network is fine-tuned using insufficient data from the target

domain. As a result, the performance deteriorates to fit the model found in the second scenario.

METHOD	MAE	MSE
MCNN w/o transfer	377.7	509.1
MCNN trained on Part_A	397.7	624.1
Finetune the whole MCNN	378.3	594.6
FineTuning the model	295.1	490.23

Table 1. Performance of proposed model

4.1.4 Performance analysis using loss functions:

We test our approach on a variety of loss functions. We can directly map the photographs to the total headcounts in the image in addition to mapping the photos to their density maps.

Theta is the total number of heads for the input image Xi (i = 1,...,N), and F (Xi) stands for the estimated density map and MCNN parameters. Objective Function is as follows:

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \|F(X_i; \Theta) - F_i\|_2^2,$$

In this instance, the estimated density map's spatial area, S, is used instead of the ground truth density map. We additionally pretrain CNNs in each column separately to compensate for this loss. Such a baseline is referred to as crowd count regression using MCNN (MCNN-CCR).

We can observe that the outcomes of the crowd count regression are not very good. The learning density map can, in a sense, save more of the image's finer details, which improves count precision.

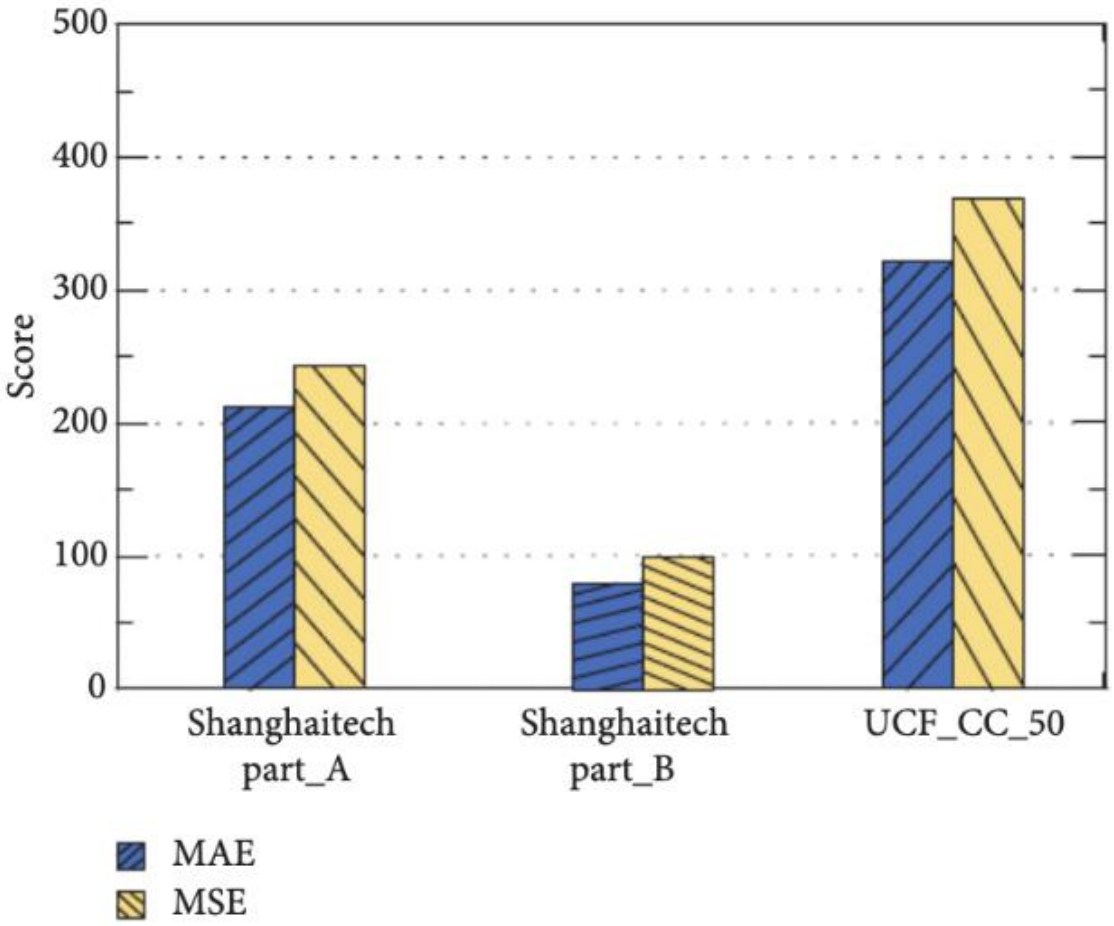
4.1.1.5 Evaluation matrix:

To evaluate the proposed model, we have used the mean absolute error (MAE) and the mean squared error (MSE).

$$MAE = \frac{1}{N}\sum_{1}^{N}\left|Z_i - \hat{Z}_i\right|, \; MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}\left(Z_i - \hat{Z}_i\right)^2}.$$

4.1.1.6 Testing model on various datasets:

The UCF_CC_50 and ShanghaiTech datasets were used to test the CNN counting model, and the results are displayed:



Graph 2. Performance on different datasets [11]

4.1.2 The C3D network for Crowd Behaviour Analysis

C3D Model or the 3D Convolutional Neural Network is used to classify the crowd's behaviour as violent or not. Model extracts the features from each frame of the video. Working of the C3D model is similar to the 2D CNN, however, C3D model works on more than one frame at a time.

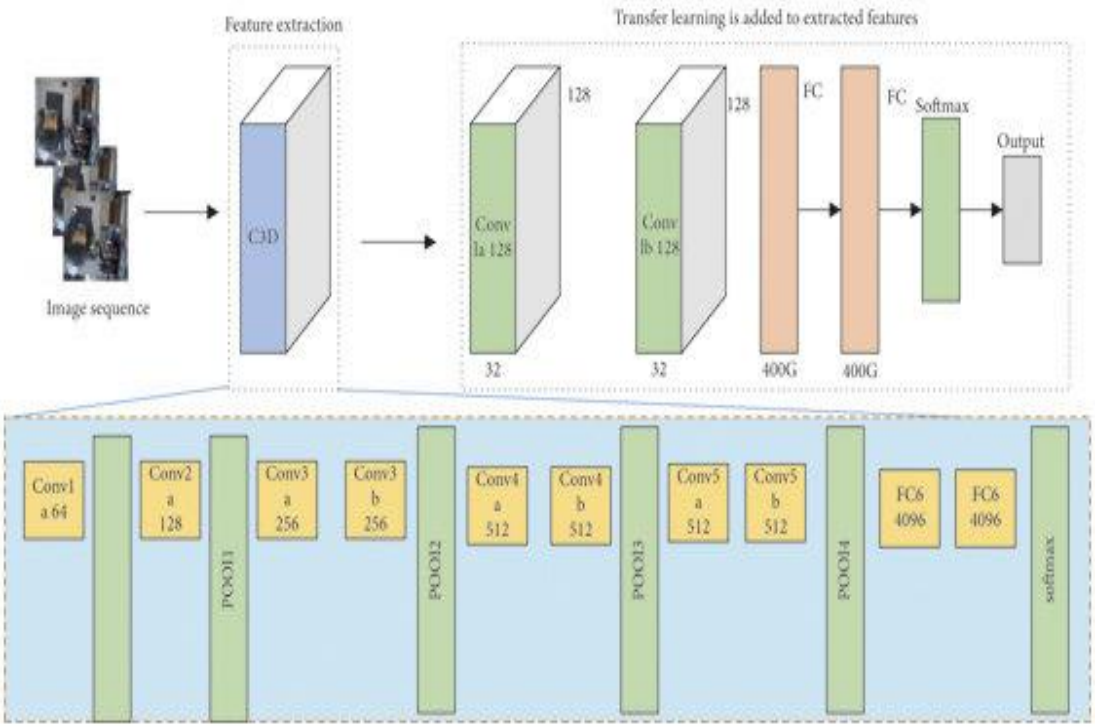


Figure 11: C3D Model architecture

4.1.2.1 Performance Analysis of C3D Model

3D Convolutional Neural Network delivers better performance on large crowd gatherings, where individual movements cannot be tracked. 3D Convolutional Neural Networks solves the problem of scalability and upon being tested, C3D model has proven to be effective on large crowds and has 61.2% accuracy.

4.2 Problems with existing literature

4.2.1 Foreground segmentation

Foreground segmentation is essential in the majority of recent work. Foreground segmentation is a difficult process in and of itself, and incorrect segmentation will have a permanent negative impact on the final count.
The vantage point from which a picture is taken might vary in our assignment. Accurately separating the crowd from its background is nearly impossible without knowledge of the scene's geometry or motion. As a result, we must estimate the size of the throng without previously

4.2.2 Variation of scale

To accurately estimate crowd sizes for different images, we must combine attributes at several scales because the size of the people in the photographs may vary greatly. We are forced to use hand-crafted features because we lack tracked features and it is challenging to hand-craft features for all scales.

4.2.3 Our solution to the problem

In this work, we propose a novel framework for crowd counting in an arbitrary still age based on convolutional neural networks (CNN). More specifically, we implemented an image classification model called a multi-column convolutional neural network (MCNN), which was inspired by the work of [8]. Their model allows for the training of any number of columns on inputs that have undergone various pre-processing steps. Then, final predictions are obtained by averaging each deep neural network's individual predictions. Three columns of convolutional neural networks with variously sized filters make up our MCNN.

The three columns are filters with different parameters, more specifically, receptive fields of varied widths, therefore using a multi-column architecture in this situation makes sense (large, medium, and small), making each column understand its features

CNN is resilient to significant differences in head or person size resulting from perspective effects or when seen at different picture resolutions.

In our proposed MCNN model, the fully linked layer is swapped out for a convolution layer with a 1 X 1 filter size. In order to prevent distortion, the input picture for our model may be of a different size. An estimate of the crowd's density, from which we calculate the total count, is the network's immediate output.

CHAPTER-5 CONCLUSIONS

5.1 Conclusions

I have implemented a Multi-Column Convolutional Neural Network (M-CNN) to precisely calculate the crowd density in a single image, from different perspective and scenarios. I have used ShanghaiTech dataset, which contains a total of 330,165 participants annotated under two section, in order to compare my result with other studies and techniques. As of right now, this dataset contains the most marked heads for crowd counting. On all test datasets, our approach surpasses cutting-edge crowd counting techniques.

Additionally, the excellent generalizability of the suggested model is demonstrated by the fact, that just by tuning parameters and final few layers, this model can be used for a number of applications across many sectors.

Implementation of 3D Convolutional Neural Network can predict the crowd's behaviour with an accuracy of 61.2%, and the architecture has proven to be effective in large gatherings.

5.2 Future Scope

Before building a convolutional neural network in this project, a thorough study is necessary to select the optimal solutions for the various parameters (CNN). Numerous choices may be made about the number of hidden layers and nodes in each layer while investigating neural networks. In this project, the Softmax activation function is used in the neural network's output layer. However, additional activation mechanisms for those layers can also be studied. The effectiveness of our deep neural network is unquestionably impacted by that. And if we can find the more precise activation function and parameters, the model's accuracy can be increased.

The number of scenarios in the datasets should be increased, and there should be more than 500,000 pedestrians. These improvements are necessary to better the evaluation

phase results. Reduce the difference between the actual crowd size and the projected count to make improvements to approaches. Additional approaches would be utilised to make the camera mobile and to integrate data from multiple sources. The Real-Time limitation must also be considered in terms of this domain's future research.

Crowd’s Behaviour can further be categorised in more than the two categories,
Which can include panic state, medical emergencies and similar categories. Accuracy and precision can be improved by tuning parameters and improving datasets.

5.3 Application

The system is able to find faces in a variety of lighting situations. There are some situations in which perfect accuracy is not possible. One of the causes is the size and form of the structuring element. When we vary the size and shape of the structuring element, the detection of the number of faces may also change. The skin colour model's restriction, which identifies both faces and background objects, is another factor. Making a distinction between face and non-facial regions can help to further reduce false positives.

References

[1] V. A. Sindagi and V. M. Patel, “Generating high-quality crowd density maps using contextual pyramid cnns,” in ICCV, 2017, pp. 1861–1870.

[2] L. Zhang, M. Shi, and Q. Chen, “Crowd counting via scale-adaptive convolutional neural network,” in WACV. IEEE, 2018, pp. 1113–1121.

[3] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,”

[4] V. A. Sindagi and V. M. Patel, “Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting,” in AVSS. IEEE, 2017, pp. 1–6.

[5] D. Onoro-Rubio and R. J. Lopez-Sastre, “Towards perspective-free object counting with deep learning,” in ECCV. Springer, 2016, pp. 615–629.

[6] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, “Crowdnet: A deep convolutional network for dense crowd counting,” in ACM MM. ACM, 2016, pp. 640–644.

[7] D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional neural network for crowd counting,” in CVPR. IEEE, 2017, pp. 4031–4039.

[8] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in CVPR, 2016, pp. 589–597.

[9] T. Hassner, Y. Itcher and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behaviour," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, RI, USA, 2012,

[10] Kratz, Louis & Nishino, Ko. (2009). Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009. 1446-1453. 10.1109/CVPRW.2009.5206771.

Major

ORIGINALITY REPORT

13%	10%	12%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	people.eecs.berkeley.edu Internet Source	2%
2	ijircce.com Internet Source	2%
3	arxiv.org Internet Source	1%
4	Fu, Min, Pei Xu, Xudong Li, Qihe Liu, Mao Ye, and Ce Zhu. "Fast crowd density estimation with convolutional neural networks", Engineering Applications of Artificial Intelligence, 2015. Publication	1%
5	Ashwani Kumar, Sunil Kumar, Premananda Sahu, Sachin Kumar Yadav. "Theoretical Guidance in the Field of Health and Healing Using NLP", 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), 2023 Publication	1%
6	ebin.pub Internet Source	<1%

7	blog.actorsfit.com Internet Source	<1 %
8	ieeexplore.ieee.org Internet Source	<1 %
9	Khorshed Alam, Nishargo Nigar, Heidy Erler, Anonnya Banerjee. "Speech Emotion Recognition from Audio Files Using Feedforward Neural Network", 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2023 Publication	<1 %
10	Addike Ajay Kumar, Madana Vamshi Krishna Reddy, V Manikanta Sanjay, Vivek Kothuru, Kunjal Prashant Shah, Akshay Kalucha. "Comparative Analysis of Skin cancer Prediction Using Neural Networks and Transfer Learning", 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022 Publication	<1 %
11	www.analyticsvidhya.com Internet Source	<1 %
12	Neetu Gupta, Gunjan Sardana. "Anomaly detection in video frames: hybrid gain optimized Kalman filter", Multimedia Tools and Applications, 2023	<1 %

Publication

13	Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, Yi Ma. "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016 Publication	<1 %
14	www.tjprc.org Internet Source	<1 %
15	keep.lib.asu.edu Internet Source	<1 %
16	sibgrapi.sid.inpe.br Internet Source	<1 %
17	Li Dong, Haijun Zhang, Yuzhu Ji, Yuxin Ding. "Crowd Counting by Using Multi-level Density-based Spatial Information: A Multi-scale CNN Framework", Information Sciences, 2020 Publication	<1 %
18	Nurzarinah Zakaria, Yana Mazwin Mohmad Hassim. "Improved VGG Architecture in CNNs for Image Classification", 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2022 Publication	<1 %

19	Syed Inthiyaz, Baraa Riyadh Altahan, Sk Hasane Ahammad, V Rajesh et al. "Skin disease detection using deep learning", Advances in Engineering Software, 2023 Publication	<1 %
20	"Pattern Recognition and Computer Vision", Springer Nature America, Inc, 2018 Publication	<1 %
21	Ying Chen, Chengying Gao, Zhuo Su, Xiangjian He, Ning Liu. "Scale-Aware Rolling Fusion Network for Crowd Counting", 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020 Publication	<1 %
22	MD Khadimul Islam Zim. "OpenCV and Python for Emotion Analysis of Face Expressions", 2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2023 Publication	<1 %
23	kalaharijournals.com Internet Source	<1 %
24	Mohammad Farukh Hashmi, Aashish Kumar, Avinash G. Keskar. "chapter 6 Subjective and Objective Assessment for Variation of Plant Nitrogen Content to Air Pollutants Using Machine Intelligence", IGI Global, 2020 Publication	<1 %

25	www.mdpi.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	< 14 words
Exclude bibliography	Off		