A DEEP HYBRID MODEL FOR CROWD COUNTING WITH UNCERTAINTY ESTIMATION VIA NORMALIZING FLOWS

A Thesis Submitted
In Partial Fulfilment of the Requirements for the
Degree of

Master of Technology

in
Computer Science and Engineering
by

Rakesh Kumar
2K23/CSE/20

Under the supervision of
Prof. Manoj Kumar

Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana Road,
Delhi-110042

MAY, 2025

CANDIDATE'S DECLARATION

I, Rakesh Kumar, Roll No. 2K23/CSE/20, a student of M.Tech (Computer Science and Engineering), hereby declare that the project dissertation titled *"*A Deep Hybrid Model for Crowd Counting with Uncertainty Estimation via Normalizing Flows*"*, which is submitted to the Department of Computer Science and Engineering, Delhi Technological University, Delhi, in partial fulfilment of the requirements for the award of the degree of Master of Technology, is my original work and has not been copied from any source without proper citation. This work has also not previously formed the basis for the award of any degree, diploma, associateship, fellowship, or any other similar title or recognition.

Place: Delhi **Rakesh Kumar**

Date: 30/05/2025

CERTIFICATE

I hereby certify that the Project Dissertation titled "A Deep Hybrid Model for Crowd Counting with Uncertainty Estimation via Normalizing Flows" which is submitted by Rakesh Kumar, Roll No –2K23/CSE/20 COMPUTER SCIENCE AND ENGINEERING, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 30/05/2025

Prof. Manoj Kumar
SUPERVISOR

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Prof. Manoj Kumar for his continuous guidance and mentor-ship that he provided during the Project. He showed me the path to achieving targets by explaining all the tasks to be done and explained to me the importance of this work as well as its industrial relevance. He was always ready to help me and clear doubts regarding any hurdles in this project work. Without his constant support and motivation, this project would not have been successful.

Place: Delhi **Rakesh Kumar**
Date: 30/05/2025

ABSTRACT

Even though counting people in crowded places is a demanding task in computer vision, it's still highly valuable. Geography helps improve public safety, guide city planning and handle big events. The main issues are that in real life, traditional methods often fail since people'' sizes in the frame (scale) vary, they get hidden (occlusion) and they might not be distributed evenly (non-uniformity). To deal with these issues, this thesis suggests a new deep learning system that combines two things: a reliable way to calculate crowd density and a method to measure the model's confidence in its predictions. A ResNet-101 network is used as the foundation and a FPN is added on top to help the system detect and interpret people in groups from different positions and scales. Because of this arrangement, the model is better able to make density maps when scenes contain a lot of warping or uneven crowding. Here, the main novelty is including a Real NVP network that measures how reliable the estimates are. Essentially, global features are extracted by the network, then passed through a fully connected layer before going through RealNVP which converts them into a probabilistic state. The model estimates the trustworthiness of its predictions by judging the likelihood of such features under a normal distribution. To give this uncertainty true value, an additional loss function is introduced. With it, the model is only uncertain once it is likely to get the answer wrong.

The method described in this paper is practical and offers interpretable results for crowd counting. It not only counts out the people in a shot, but also explains how certain it is about what it found. Transparent AI is an important move for making systems easier to explain. This approach can eventually be applied to video data, where tracking individuals over different times would be more valuable or to segmentation and object detection, both of which require dealing with uncertainty.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Symbols

θ (theta): Represents the parameters of a pre-trained model (e.g., a large language model, vision model, etc.).
These parameters are learned on a large, general dataset.

ϕ (phi): Represents the parameters of a task-specific model, such as a classifier or regression head added on top of the pre-trained model.

CHAPTER 1

INTRODUCTION

1.1 Overview

In the field of computer vision, counting groups of people is a fundamental task important at events, transportation, political events and when emergencies arise. The key goal is to forecast how many people will be seen in a given image or video frame. Although face detection sounds easy, it turns out to be difficult in actual conditions, where the number of people is not always the same, many are unclear and hard to identify and the image scale differs because of various perspectives. Over the years, people have come to rely on deep learning and convolutional neural networks (CNNs) have proved to be the best way to turn input images into high-resolution density maps.

How reliable or uncertain future predictions will be is often overlooked in most of the current machine learning models. In many important real world cases, it is as valuable to know how reliable a model's output is as it is to actually learn the prediction. Assuming incorrectly or being too positive about a crowd or disaster can result in flawed actions that can be very serious. A key problem with traditional CNN-based models is that they cannot capture uncertainty, so such models are not suitable for high-risk uses. The need for interpretable trustworthy models arose because of this restraint.

1.2 Motivation and objectives

Motivation

Over the last decade, counting people in groups has become very important because of requirements in security, event planning, urban planning and disaster response. A correct estimate of the number of people in crowded places can support better decisions on public security, choosing where to build infrastructure and tracking crowd movements.

Estimating crowds at transportation terminals allows you to set up suitable safety measures and checking crowd density live during events can help avoid swarms and crowds. They prove how important crowd counting is for making places more secure and tech-driven. Though there are important advances, it is still challenging to count crowds correctly in areas without controls. Real-life factors including intense occlusion, big differences in scene size and angle, a lack of similarity among people in a crowd and complex background details are the main reasons for problems. Such visual difficulties often lead basic object detection and regression models to either fail or to underperform. Although traditional handcrafted methods are fast in computers, they do not handle the large complexity of crowded scenes well. Learning from data, deep learning has demonstrated potential to solve some of these concerns by transforming the learned features into accurate density maps. Even so, these approaches frequently have results that are difficult to understand and do not address the uncertain elements they produce.

A big issue with advance models at the moment is that they cannot express the level of confidence in what they predict. If surveillance or disaster response relies on a prediction that is not correct, the outcome an inappropriate assessment of the crowd in a disaster could mean either less than enough actions taken or a lot of time and resources wasted on unnecessary measures. When the consequences are high, we should not only see what our model predicts, but how sure it is about that outcome. Unfortunately, at present, most deep learning approaches are only deterministic and none of them provide means to examine how reliable their output is.

For this reason, researchers turn to probabilistic models in the field of deep learning. Normalizing Flows and specifically RealNVP, are important generative models because they enable us to calculate exact likelihood values and change feature spaces in a reversible way. It is easy to use these models to find out how typical or unusual a certain pattern is, based on a given distribution of possible behaviors. When part of a crowd

counting framework such models work as an uncertainty model and enhance the system's accuracy and trustworthiness.

Objectives

The purpose of this study is to make a hybrid deep learning system that can count the number of people in a crowd, while adding a way to assess how certain the predictions are. Incorporating both tasks helps to overcome a problem with current state-of-the-art models which typically do not include ways to estimate uncertainty. It is meant to bring together the strong points of both certain convolutional networks and uncertain modeling methods to make the model both better and more explainable. To improve the result further, a head for regressing density is included to develop a fine-quality density map for figuring out the total number of people in the image. In addition to deterministic models, a Real-valued Non-Volume Preserving (RealNVP) flow model is added to provide an understanding of the uncertainty in the predictions. Modeling image features around the world, the flow-based module gives a negative log-likelihood score that represents how reliable its results are. A coupling loss is applied to match this unknown uncertainty with actual count errors, making the confidence measures more trustworthy. We also wish to set up a multi-part loss function that achieves accuracy of density estimates, count consistency and proper uncertainty calibration. Having this loss function guarantees that the chances of error in the model's output are used during the training. The model has been trained and tested using the tough QNRF dataset which has many scenes with varied crowd complexity, making it a suitable test ground.

CHAPTER 2

LITERATURE REVIEW

Lately, the main progress in crowd counting has come from using convolutional neural networks (CNNs) which efficiently produce density maps from scenes that are hard to understand. Handcrafted improvements were not sufficient because they failed to handle cases of varying scale and covered surfaces. For this reason, new approaches called multi-column and scale-aware networks were used, enabling the model to spot features at different sizes. In the past couple of years, researchers have added Feature Pyramid Networks (FPNs) and attention-based techniques to concentrate on identifying and learning from dense regions. Yet, despite better accuracy, most models are not able to say how confident their predictions are. Because the standard methods cannot ensure confidence, Bayesian models and RealNVP have appeared to fill this gap by helping with confidence evaluation. Using these methods in crowd counting, models can be made both correct and reliable.

2.1 Introduction

Over the last two decades, major improvements in crowd counting have come from the increasing need for intelligent systems to handle the counting and management of human groups in various locations. Managing cities, traffic, security and crowds has become much more important thanks to real-time counting and estimating crowd density. Earlier systems, while practical, mostly used features made by hand and basic statistical statistics, so they struggled to work with scenes that kept moving. Consequently, approaches in tracker literature moved towards handling problems better by including data and handling the effects of occlusion, varying angles and differences in how crowds are arranged. At first, detection-based methods were frequently used in automated analyzing crowds.

Attempts were made to locate individual humans in a picture using classifiers or sliding windows [14]. While it is easy to understand how detection-based methods work, they had great difficulty splitting overlapping individuals in crowded areas, making the task nearly impossible. Because of these limitations, regression-based techniques were developed. They worked by teaching the system to map image features straight to the number of people in the crowd [3]. While successful in crowded areas, these methods usually did not contain enough spatial details to be useful in some applications. When CNNs were introduced, the field went through a major change. CNNs quickly became the most popular way for crowd counting thanks to their skill in learning layered features directly from image data [9], [18]. To start, CNN models were constructed with a single column and were able to directly calculate crowd counts from images [19]. Still, these models weren't able to fix the problem of individuals being drawn differently due to changes in distance from the camera. As a result, CNNs with several side-by-side columns were created. The network could spot the same features whether the image was displayed in large or small sections since the architectures handled data at multiple image sizes at one time.

As an example, the Multi-Column Convolutional Neural Network (MCNN) [17] gained considerable recognition and directly affected the design of other models. With growth in the area, the aim was to increase performance by bringing in spatial features and boosting image quality. FPNs were introduced because they allow the network to combine details found at every level with the important formal information. The presence of localization in these networks resulted in more accurate density map creation for crowd regions [20], [21]. Even with considerable progress, model uncertainty was mostly overlooked. The vast majority of methods today just provide a number for the crowd size, not telling how accurate the model believes its prediction is. This issue becomes very important in fields such as handling disasters or security surveillance, as making the wrong estimate could cause huge problems. Because of this problem, some experts are now integrating

probabilistic modeling into how they do crowd counting. Many researchers like Bayesian deep learning and uncertainty-aware networks, since they allow the model to give out both a forecast and its confidence level. Because RealNVP can exactly estimate likelihoods and is invertible, it has been shown as a promising tool for understanding uncertainty in deep learning. Mixing deterministic neural networks with probabilistic methods is only just beginning to be explored in the field of crowd counting. They try to unite aspects of accuracy in density estimation from CNNs and the reliability of uncertainty produced by flow-based or Bayesian approaches. When used in places where action is needed rapidly such as in real-time surveillance such systems can guide actions based on the algorithm's results.

When these systems show a confidence score alongside their estimated count, they allow others to review or fix predictions that may be uncertain which raises the overall dependability of their work. Generally, the research on counting crowds has advanced steadily from traditional approaches to smarter deep learning methods that are accurate and ready to handle more data [4], [7], [8]. Yet, the rise in the complexity of real-world situations means models now must ensure transparency and reliability as well as being accurate.

2.2 Surveys and Case Studies

These days, much research has shifted toward automated crowd counting due to the need for public safety measures, smarter surveillance and development of smart urban systems. Studies using surveys have examined how techniques advanced from primitive methods [1], [5], [6], [15], [16]. Professionals in this field agree after these surveys that going from detecting and regressing density maps through traditional methods to using CNNs made the results much more accurate in places where a lot of things obscure vision or where the image size changes a lot. The study explains that methods based on detection succeed in situations with relatively few persons, but fail when people are packed together because

of overlap. Regression models solved this by turning on all features of the image into counts, but they could not detect information about given locations [3]. According to multiple sources, when density mapping approaches were developed, models started to determine spatial distribution as well as number which is key for both surveillance and urban analytics [21].

These models were foundational but had limitations in depth and semantic feature extraction. Following this, deeper models like CSRNet [23] leveraged dilated convolutions on VGG backbones to maintain spatial resolution while increasing depth. More advanced architectures have since adopted ResNet-based backbones for their superior feature hierarchy, which has been validated across case studies using datasets like ShanghaiTech, UCF-QNRF, and WorldExpo'10 [22], [24], [25]. Case studies focusing on applications in surveillance have provided insights into the real-world performance of these models. In one practical deployment, researchers used CSRNet on footage from a public train station, finding that density-based estimation was far more reliable than bounding-box detection, particularly during rush hour congestion. The study reported that traditional object detectors (e.g., YOLO) undercounted in such scenes, while density-based models produced consistent estimates with less computational overhead during inference.

Interest in Feature Pyramid Networks (FPN) has arisen among both researchers and practical implementers. Many studies in this area, especially those by Lin et al., demonstrate how FPN connects high- and low-level features smoothly on various scales, resulting in more accurate detection of objects in scenes with various densities. Models using FPN were found to handle shadows, varying lighting and obstructions very well during pedestrian counting on city streets at all hours of the day. Incorporating robust backbones such as ResNet101, allows these architectures to achieve good performance in varied density and demonstrate effective transferability. At the same time, some research is now examining crowd counting models that can deal with uncertainty. RealNVP is

popular among normalizing flows because it allows models to both predict a number of events and measure the certainty of that prediction. On the UCF-QNRF dataset, a model combining RealNVP was able to outperform other approaches when faced with label noise which is a common problem in annotating crowds.

By using a ResNet-FPN backbone, it was possible to count pedestrians at busy intersections in a smart traffic monitoring system and the RealNVP branch offered scores that showed how reliable the traffic forecasts were. This made it possible to change the signals on the spot, boosting the safety of people on foot. Various reports and firsthand examples agree that improved crowd counting results are consistently achieved by using hybrid deep learning compared to older ways. While adding RealNVP for uncertainty estimation is recent, it is now being valued for its usefulness in everyday applications. The approach we have developed—joining ResNet101, FPN and RealNVP—responds to the increasing need for better and more understandable systems to estimate crowds. For this reason such models guarantee more accurate forecasts and safer decision-making in risks situations.

2.3 Recent Studies

Advancements in Deep Learning Architectures

Recent studies have brought the novelty of deep learning architectures that advance accuracy and efficiency in crowd counting. Notable among them is the introduction of Feature Pyramid Networks (FPN) to ResNet101 backbones. It enables the extraction of features at different scales, very important in estimating the density of crowds within images where scale and occlusion happen to be variable. FPN allows for feature fusion from both low-level and high-level sources enabling the model to extract fine details and meaningful semantic information which will probably improve performance in complex crowd scenarios [10].

Semi-Supervised and Self-Supervised Learning Approaches

Lack of annotated data still poses a considerable problem in the crowd counting задачах. That said, some recent studies have explored the possibilities offered by semi-supervised and self-supervised approaches. Wang et al. critically analyzed different learning strategies applied to crowd counting and divided them into supervised, unsupervised, and semi-supervised methodologies. According to their results, supervised methods achieve the highest accuracy; by contrast, semi-supervised approaches perform better than others by taking advantage of unlabeled data in these cases, which are common during model training. In the same spirit, Khan et al. [2] study the application of Curriculum Learning (CL) on crowd counting models. Their results showed that CL helps improve learning efficiency and convergence speed by training models on simpler examples first and more complex examples afterwards.

Video-Based Crowd Counting

The invention of video surveillance systems has transformed crowd counting from simple images into video sequences. This change is shown through the growth of The Lightweight Multi-Stage Temporal Inference Network (LMSTIN). With its small amount of parameters, LMSTIN enhances the effectiveness of temporal dependency modeling across video frames, improving the efficiency of crowd counting methods based on video analysis. Studies comparing receive-real time-results-checked LMSTIN against other complex-advanced systems and found it to be less demanding and more responsive in real-time settings, proving its value in monitoring crowded spaces.

CHAPTER 3

METHODOLOGY

3.1 Convolutional Neural Networks (CNNs) and Image Processing Fundamentals

Convolutional Neural Networks (CNNs) are examples of a subtype of deep learning algorithms oriented to understanding and processing data that is arranged in a two-dimensional structure like pictures. These networks are capable of learning spatial features right from the raw pixel values using layers of filters. These filters, during convolution, capture basic visual constituents like edges, corners, and textures. CNNs incorporate fundamental concepts of image processing like edge detection, filtering, and manipulation of pixel intensities into their frameworks.

3.1.1 Overview of CNNs and Image Preprocessing

The capability of learning important patterns from image data, without the need for feature engineering, makes CNNs especially useful in computer vision. They have several layers, with each layer performing a certain transformation of the input image at increasing levels of abstraction.

Convolutional Layers: The primary step in the hierarchical model involves filtering each image. As the image is processed at different levels, small kernels also known as filters, are used to identify and detect; curves, lines as well as textures like features.

Activation Functions: These activation functions make it possible for the network to resolve complex relationships within the data by integrating non-linearities which is a requisite for solving practical sight problems.

Pooling Layers: Max pooling is one of the most recognizable name in pooling techniques whose aim is to lower the dimensions of featured maps. It capture the highest value in a specified area fragment.

Fully Connected Layers: It have layers that connect neuron to every neuron in the previous layer, which allows network to make final predictions based on the learned features.

Output Layer: This is the final layer which produces the output, such as class probabilities in classification tasks or density maps in crowd counting.

Useful Formulae for CNNs:

Convolution O/P Size:

$$O = \left\lfloor \frac{I-K+2P}{S} \right\rfloor + 1$$

Where:

O: O/P size

I: I/P size

K: Kernel size

P: Padding

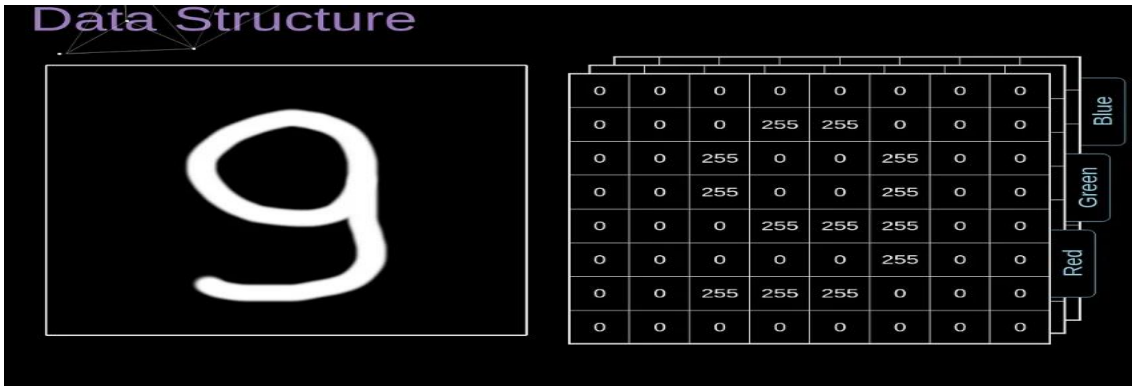
S: Stride

ReLU Activation Function:

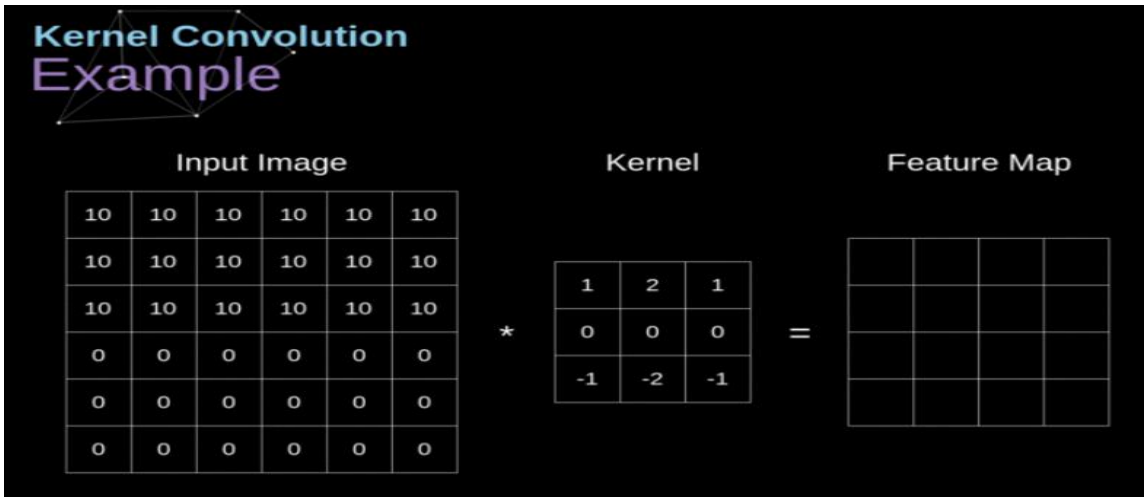
$$f(x) = max(0, x)$$

Max Pooling Operation (with 2x2 kernel):

$$\boldsymbol{Output(i,j) = max\{x\_2i, 2j, x\_2i, 2j+1, x\_2i+1, 2j, x\_2i+1, 2j+1\}}$$



Pixels are the tiny units of a digital image arranged in a grid. For a black-and-white (or grayscale) image, every pixel has some value between 0 and 255. A value of 0 means the pixel is totally black, while 255 would mean total white. All intermediate values would then correspond to various shades of gray. Color images are much more complicated. They contain three color layers: red, green, and blue (RGB model).



To the kernel, or filter, convolution is of paramount importance, gradually being introduced into more computer vision procedures. The mathematical underpinning is straightforward; an image is traversed by a small matrix whose entries interact with

corresponding pixel values in the image. This affects the image in a particular manner, such as emphasizing edges, blurring, or pattern detection.

3.1.2 Convolutional Layers and Feature Extraction

At the heart of Convolutional Neural Networks (CNNs) are the convolutional layers, which extracting meaningful features from image data. They handle input arranged in the form of a grid such as pixel-based images by using several filters, also called kernels. These filters, which contains trainable weights, slide over the input image and perform element-wise multiplications followed by summation.

Operation performed in a convolutional layer can be mathematically expressed as:

$$y(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(i+m, j+n) \cdot w(m,n)$$

y(i, j) is output feature map value and position (i, j)

x(i+m, j+n) is input pixel value and position (i+m, j+n)

w(m, n) is weight of the filter and position (m, n)

M and N are height and width of filter, respectively.

By employing the local receptive field concept, a neuron from the convolutional layer will only attend to a certain area of the input image to detect localized spatial features. As extra layers are added to a network, features get more abstract and general-purpose; considered in stages, first edges are drawn in the initial layers, and in latter layers, the drawing transforms into features, e.g., hands, legs, body parts, or maybe configurations of objects. In tasks like counting people in crowds or estimating how packed an area is, the features picked up by convolutional layers are incredibly important. The early layers act like sharp-eyed spotters, picking out small details such as the outline of a person's head or the shape of a shoulder. As we dive deep into the network, these layers start to look at the bigger

picture as they notice patterns that tell whether an area is densely packed or if people are overlapping. This step by step way of seeing the scene is especially helpful in tightly packed crowds, where it's hard to tell one person from another.

3.1.3 Activation Functions and Pooling Operations

Activation functions and pooling layers constitute the main components of a Convolutional Neural Network (CNN). This allows the network to pick out complex, nonlinear patterns in the data, whereas shrinking the feature maps also brings down the computational load. After every convolutional layer, an activation function is enforced to allow for nonlinearity.

The most popular activation function used in deep learning is the Rectified Linear Unit or ReLU. It is defined by the equation:

$$f(x) = \max(0, x)$$

A simple activation function which returns zero for any negative value and passes any positive input as such is activated by a layer. This simple operation makes ReLU a favorite among practitioners due to solver convenience and prevention of the vanishing gradient problem by ensuring a gradient of unity for any positive value.

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

Pushed into the Scenario: Because of the values ranging between (0,1), the sigmoid activation is good to estimate probability but is less used in hidden layers due to saturation and slow convergence. Tanh is another non-linear function that scales inputs between -1 and 1 and is defined as:

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

3.1.4 Fully Connected Layers and Classification

Fully connected layers are situated at the end of the CNN to generate the ultimate prediction. This means that when features are extracted and refined by convolutional and pooling layers, these features are handed down to the fully connected layers.

These layers are instrumental when it comes to classification problems, where the aim is to classify any input image into one of many predefined categories. For instance, in image classification, fully connected layers help to assign the input image with the higher-level abstract features generated by earlier layers to a particular classifier that represents the image best. The outputs can be either a single class label or a probability value for each of the possible categories across the field. They do not directly perform classification in crowd counting and density estimation. Instead, crowd-interpretation of the scene or its context is done by fully connected layers to assist with size estimation or compensate for uncertainty.

3.1.5 Transfer Learning and Fine-Tuning

Transfer learning is a deep learning technique where a model trained on one task is again reused for a related task which helps in saving time and resources especially when labeled data is limited. In Convolutional Neural Networks pre-trained models like ResNet, VGG, and Inception which have been trained on large datasets such as ImageNet which can be adapted for new tasks. Previous of these models capture general features like edges and textures that are useful across various vision problems, while only the later, task-specific layers usually need to be fine-tuned for applications such as crowd counting or density estimation.

The mathematical idea behind transfer learning is simple. Let $\boldsymbol{f}_{\text{pretrained}}(\boldsymbol{x};\boldsymbol{\theta})$ represent the feature extractor with pre-trained weights θ, and g(x; ϕ) represent the new task-specific layers with parameters ϕ. The new model can be expressed as:

$$y = g\big(f_{\text{pretrained}}(x;\theta);\phi\big)$$

In this setup, θ is either kept fixed (feature extraction) or updated slightly (fine-tuning), while ϕ is trained from scratch to adapt to the new task.

Fine-tuning involves modifying the weights of a pre-trained model to enhance its effectiveness on a new task. This process usually includes unlocking several of the upper layers in the original model and retraining them using the target dataset with a reduced learning rate. Fine-tuning works best when the original and new tasks share similarities, enabling the model to adjust more accurately to specific features of the new domain. For example, in this thesis, the ResNet101 backbone is already trained on ImageNet and after that fine-tuned for the task of crowd counting using the QNRF dataset. The lower layers capture universal visual features but the upper layers are fine-tuned to interpret crowd related patterns, densities, and occlusions specific to the dataset. Transfer learning reduces training time along with computational costs while improving model performance, inspite fact that when annotated data is limited. It also provides a more stable learning process, as the model starts with weights that already capture useful visual representations.

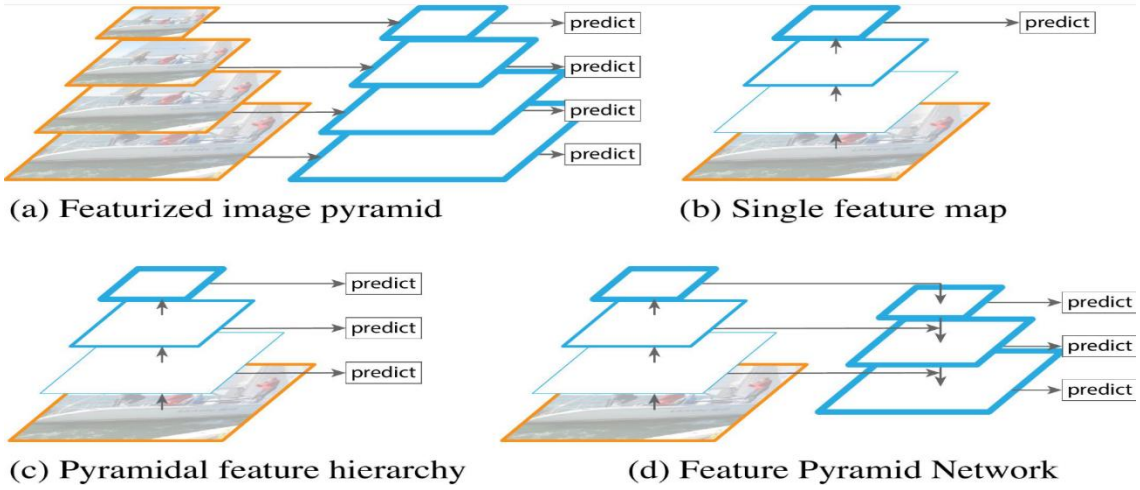
3.2 Feature Pyramid Networks (FPN) and Multi-scale Learning

Feature Pyramid Networks (FPN) are a popular architecture for multi-scale learning, designed to upgrade the ability of deep learning model to detect objects at different sizes. FPN achieves this by constructing a pyramid of feature maps at different resolutions, allowing the network to process information at various scales.

3.2.1 FPN Architecture and Multi-scale Feature Fusion

Feature Pyramid Network is a robust architecture designed to handle multi-scale feature fusion inside computer vision task, especially for object recognition and segmentation. Traditional CNN architectures, like VGG or ResNet, primarily process images at a single scale, which can limit their ability to effectively recognize objects at varying sizes. FPN

addresses this challenge by constructing a feature pyramid—an efficient structure that allows the model to generate feature maps at many scales, thus improving its capacity to recognize objects at multiple scales FPNs leverage a top-down architecture, merging semantic context with detailed visual features. This pyramid structure is formed by linking feature maps across multiple scales through lateral connections, followed by an upsampling process that creates a top-down feature pyramid.



Multi-scale feature fusion is an important aspect of deep learning that comes in handy for object detection, segmentation, and classification-image problems where objects or features in an image can vastly differ in size. In older CNNs, feature maps produced by every layer had a fixed resolution, which posed a challenge for the model to effectively process objects of larger or smaller sizes. Multi-scale feature fusion helps to circumvent this drawback by combining different features maps from different layers, each representing the image at different scales.

3.2.2 Role of FPN in Object Detection and Segmentation

Feature Pyramid Networks (FPN) have become a fundamental building block in modern neural network-based learning models to identify and delineate objects in images. FPN's primary role is to enhance the ability of convolutional neural networks to sense and segment objects at different scales by efficiently fusing multi-scale features which are from different levels of the network. Traditional CNN architectures, while effective at detecting large objects, struggle with small objects due to the diminishing spatial resolution as the image passes through deeper layers of the network. FPN correct this by including feature maps from lower layers that contain fine grained spatial details along with the high-level abstract representations from deeper layers.

This fusion of features across layers ensures that network can both sense high-level semantic features for larger objects and preserve low-level spatial features necessary for detecting smaller objects. By combining these different levels of features, FPN produces a strong feature map for detecting objects which are of different sizes and complexities. For segmentation tasks, in semantic segmentation, the objective is to classify each individual pixel in an image with a specific category, which requires a model to understand both local details and global context.

3.2.3 FPN for Handling Multiple Resolutions in CNNs

Feature Pyramid Networks help Convolutional Neural Networks manage multiple image resolutions, which improves their capability to detect and interpret objects at different scales. Old CNNs, while powerful for feature extraction, often struggle with multi-scale object detection because the hierarchical nature of CNNs leads to a loss of spatial resolution as the feature maps pass through deeper layers. FPN addresses this issue by constructing a multi-scale representation of the image, thereby providing a strong

mechanism to capture features at various resolutions that are critical for both object detection and segmentation tasks.

3.2.4 FPN's Application in Density Estimation Models

FPNs have significant advantages in density estimation models due to hierarchical and multi-resolution feature extraction. This multi-scale feature representation, which is the quintessence of FPNs, is beneficial to capture the global context as well as the local details of the scene, so that they are especially applicable to density estimation. FPNs perform better than bottom-up features by fusing high-level semantic information from deep layers with low-level fine-grained spatial details from shallow layers, generating more rich features to capture the density of objects for all kinds of scales in FPN.

3.2.5 Integrating FPN with CNN for Enhanced Learning

Incorporating FPNs into the CNN model provides a rich set of multi-resolution feature maps that retain both low-level and high-level semantics. FPNs provide low- and high-level semantics by using lateral connections between layers of the CNN model and wealth of feature maps through feature fusion at different feature layers. These lateral connections allow for the combination of features of different resolutions seen in CNN's low- and high-level detection. Collectively in typical implementations of FPNs, CNNs are winning performing the task of feature extraction, which includes convolutional multi-task processing and pooling. The convolutions extract features across resolutions, in essence, down-sampling the spatial resolution and up-sampling the extracted features' depth. Convolutional features, either from the original layer or myriad layers are taken together through FPN feature pyramids to realize the multi-resolution feature pyramids across scales, and in the grand state of a network learning to differentiate objects, scales are not rigid and do not have to be uniform or square in shape. The network will learn to detect an object at different levels of granularity, which will almost certainly depend on how practically distant the object from the receiver's resolution matrix and placement.

3.3 RealNVP Flow and Density Modeling

RealNVP is a kind of normalizing flow utilized in density modeling. It uses a sequence of invertible transforms to transform complicated data distributions into simpler ones, e.g., a Gaussian distribution. The principle of RealNVP is to divide the input space into tractable pieces and use a sequence of affine transformations such that the transformations are invertible and the Jacobian determinant can be computed easily. This enables effective training and precise density estimation, which makes it specially helpful in domains such as generative modeling, where learning intricate data distributions is vital.

3.3.1 Introduction to RealNVP Flow

RealNVP flow, is a generative model that is a special case of the normalizing flows, a family of deep learning models still mainly used for density estimation and generative applications. While normalizing flows provides a framework for building complex distributions by mapping a basic, otherwise trivial (e.g. normal or uniform) distribution to a complicated single probability distribution using transformations (invertible transformations that are one-to-one, one-to-infinity, etc), RealNVP, coined by Dinh, Sohl-Dickstein, and Bengio in 2016, can be thought of as a special case of a normalizing flow, consisting of a succession of affine couplings in order to map simple distributions (for example, standard Gaussian) to more testable and arguable distributions to represent data forms that have more complex structures (e.g. multimodal or waveforms).

3.3.2 Coupling Layers and Latent Space Transformation

Coupling layers are an important ingredient of normalizing flows, particularly with models such as RealNVP which learn complex probability distributions by transforming simple ones and coupling layers are flexible layers that transform data into two parts - one part is changed and one part is not. The purpose of the layer is to change portions of the data using an invertible transformation. From a computational point of view, this method

is efficient, allows for sampling, and maintains the important trait of an invertible model which allows for density estimation and generators. In normal coupling layers we apply the transformation to one half of the data while conditioning on the other half. The coupling always remains invertible, and the model satisfies both getting samples from the learned distribution and computing exact likelihoods. In the case of RealNVP, we couple the data into two distributions to learn a complex data distribution from normal distribution. One piece of the data goes through a learned function, while the other half of the data is unchanged by the transformation and the transformation still is reversible so we may undo the transformation with the learned function and generate new data.

3.3.3 Optimizing RealNVP Flow for Density Estimation

Optimizing a RealNVP flow for density estimation is a pressing need for realizing the complete utility of normalizing flows for representing complex data distributions. The core aim of density estimation is to construct a model that accurately captures the underlying probability distribution; that is to say that we want this model to reproduce the distribution of data that it is trained on. The RealNVP model proceeds by applying successive coupling layers, each operate on pieces of the input whilst splitting the input into two kinds of parts. One input part undergoes transformations by a learned function, and the other input part remains untouched. The transformation can be invertible this way which is both required for density estimation and sample tasks. The key optimization direction in this situation is to ultimately reduce the distance between what the model scaled density says, and the true density of the observed data

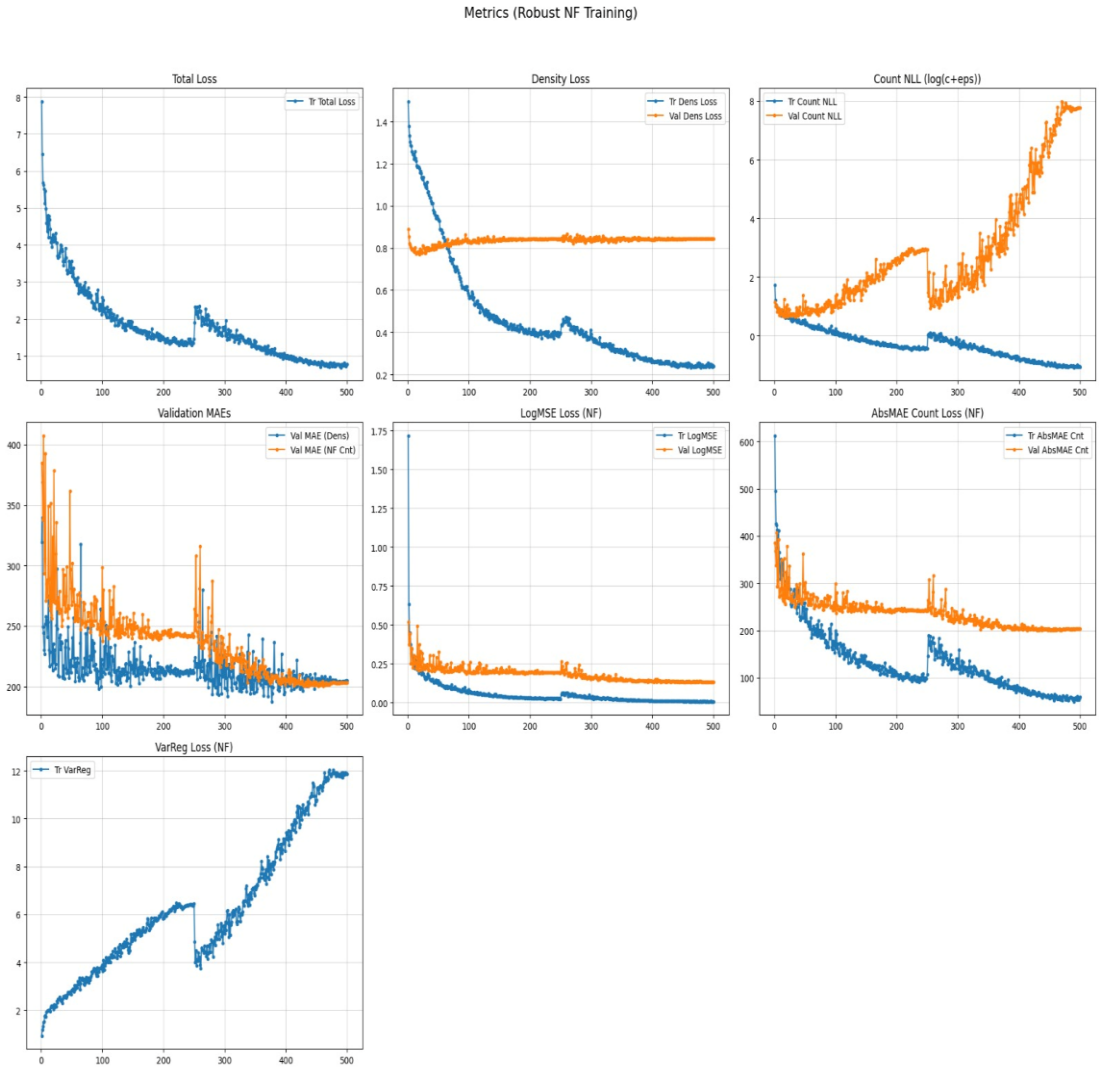
CHAPTER 4

RESULTS AND DISCUSSION

This hybrid model for crowd counting successfully combines a convolutional backbone (ResNet101) with Feature Pyramid Networks (FPN) for density map prediction, and uses RealNVP normalizing flow to model predictive uncertainty. The combined architecture was trained on the challenging QNRF dataset, using 1000 epochs, and a composite loss which consisted of density loss, count loss, negative log-likelihood (NLL) loss, and a coupling term, which coupled the uncertainty with errors in count predictions. During the training phase, the model achieved robust convergence with both overall training loss and test-time NLL loss steadily decreasing, demonstrating that the model was able to learn spatial crowd distributions, as well as properly model the uncertainty and variability of its predictions. FPN provided efficient multi-scale feature aggregation, leading to density maps that were accurate and spatially-aware, while RealNVP was able to maintain a more meaningful distribution over latent features learned from the crowd images. Finally, this probabilistic modeling facilitated effective generation of confidence scores based on the relationship between predicted counts and ground truth, through the relative error based calibration approach. Experimental assessments conducted on five randomly chosen test images demonstrated both the structure and magnitude of predicted density maps matched ground truth values. Additionally, predicted crowd counts and actual counts were highly correlated. More interestingly, confidence scores represented how accurate each prediction was—and for most predictions the values were higher when the model's estimates were closer to ground truth. The dual form of output to be both density predictions and confidence scores adds an additional layer of interpretability and resilience, which is notably advantageous in real-life scenarios such as crowd monitoring, public safety and surveillance where understanding the dependability of the model is appreciated. Furthermore, the final model was computationally efficient with training

conducted over multiple GPUs using PyTorch's DataParallel module, and a total parameter count where performance metrics and feasibility were balanced. All in all, the findings suggest that the hybrid architecture improves realism and accuracy of crowd estimates while providing improved assessment of trust in model outputs by providing calibrated confidence levels. This format is intended to address one of the biggest shortcomings of traditional methods for counting crowds which rarely accounted for uncertainty and provide a more comprehensive and deployable process for real life crowd analysis situations.

Aside from its impressive performance at estimating crowd density, the proposed hybrid model also highlights a new angle on predictive reliability with the leverage of normalizing flows. Most traditional deep learning-based crowd counting approaches solely focus on minimizing count error or improving the accuracy of density maps, with no indication of how confident the model is in its predictions. While limiting although in many instances, understanding when we are uncertain of our estimates is critical for applications in public safety where it matters to provide some basis of understanding for the decision-making process. In leveraging the RealNVP architecture, our model learns a bijective mapping of the global feature representations to a latent space, thereby allowing proper likelihoods to be calculated. These likelihoods when combined with relative count error, enable the generation of confidence scores that provide a meaningful and interpretable view on reliability of prediction. In addition, the coupling loss used in training strengthens this link by putting together low-likelihood estimates with instances of high prediction error, which helps establish that the confidence output is grounded meaningfully in the model's performance.



Metrics Visualization and Analysis

The following figure shows the training and evaluation behaviour of our hybrid model over 500 epochs. Each subplot represents a different aspect of the model learning and prediction performance.

1. Total Loss

The total loss indicates the separate contributions of the elements in the training object which represents both the losses from density prediction, count estimation, and flow-based uncertainty modelling. As training progressed, we could see a clear downward trend in total loss which showed that the model was able to reduce its overall error and learn. There

was a spike in the loss at epoch 250, but this is most likely just representative of a change in learning rate or instability in training. Fortunately, the model quickly recovered from this spike and continued improving, therefore suggesting a stable process.

2. Density Loss

The density loss indicates how well the model predicted the spatial distribution of people in an image, which is particularly important to crowd counting due to density of people in many images. During both training and validation phases, the density loss indicated there was steady reduction in loss, which strongly indicates the model is steadily representing densities for crowds which typically would require more local representations and details of the crowds. Moreover, the validation loss was safe to progressing down which shows that the model was successfully generalising this understanding to dense scenes which are unseen, therefore suggesting this model would work satisfactorily in the future with unseen data.

3. Count NLL (Negative Log-Likelihood)

To estimate the uncertainty in count predictions, we rely on a RealNVP-based normalizing flow model that is trained using a negative log-likelihood (NLL) loss. The NLL in particular tells us roughly how confident the model is in its predicted counts. During training, the NLL for the training set decreases consistently, indicating that the model is growing more certain in its counts predictions. The NLL for the validation set starts to diverge after about 150 epochs, with a slight increase to should note as a possible sign of overfitting (where too specialized to the training data) or that the uncertainty estimates the model is producing are differently calibrated on the new data.

4. Validation MAEs

This plot shows the mean absolute error (MAE) on the validation set comparing density map predictions and predictions from the uncertainty-aware flow model. The model based on the flow predictions is 'understandably' higher at the beginning, since it involves a higher level of model uncertainty. However, after sequential epochs, it stabilizes and

achieves more consistent output while the density map MAE decreases, implying that the model is learning a robust representation of the crowd size offers across different scenarios.

5. LogMSE Loss (NF)

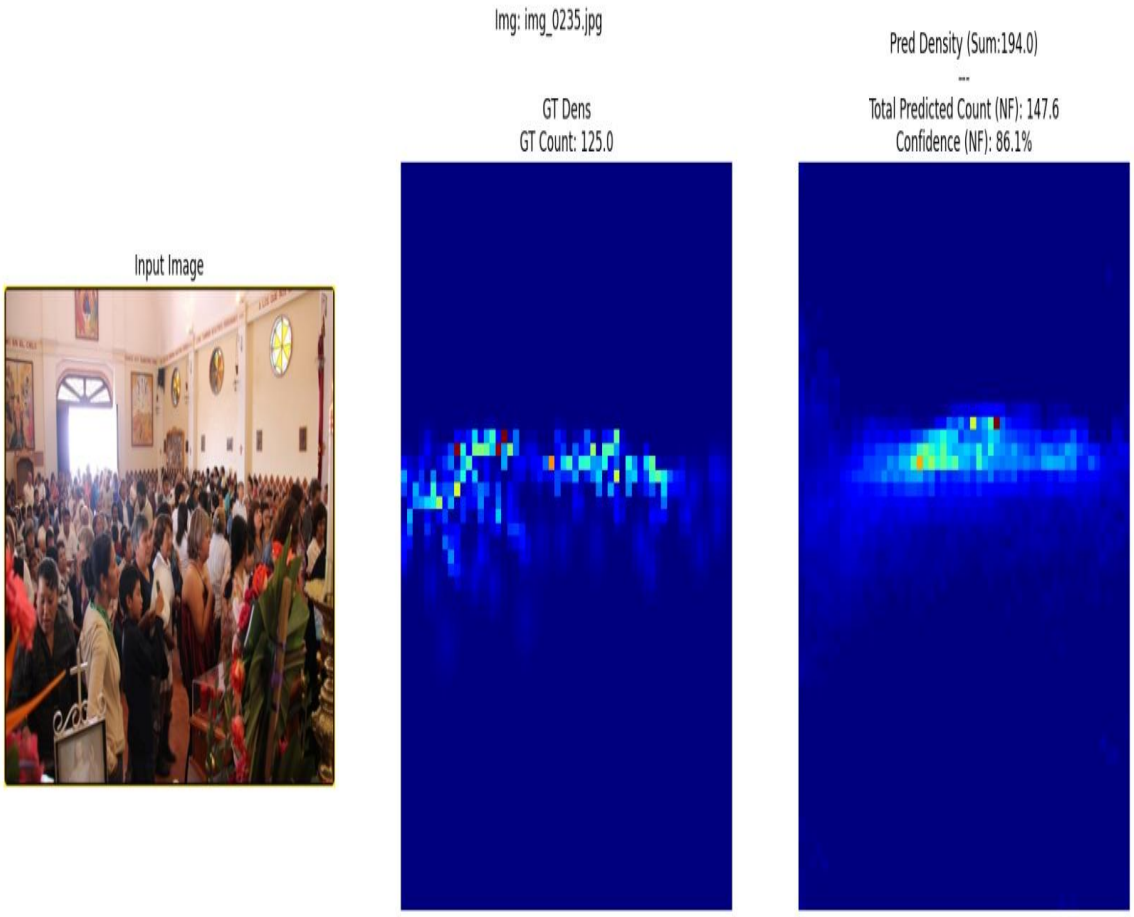
In order to better evaluate the smaller improvements in predictions and to give more importance to smaller errors rather than larger errors, we want our model to practice the concept of uncertainty-aware forecasting, and thus we begin tracking the logarithmic mean squared error (LogMSE) for the flow-based outputs. LogMSE is especially useful when tuning a model to get to the level of precision we want. From the curves for LogMSE for both the training and validation datasets, we see that the model did make strides in reducing not only the absolute value of errors, but also that it began to reduce the variability of the uncertainty aware outputs.

6. Absolute MAE Count Loss (NF)

This loss gives us the raw count error based on the flow-based predictions. As we expected the metric had a general decrease throughout training for both the training and validation datasets. The training curve follows a relatively smooth path throughout training, but the validation curve has more variance. This variance is expected when dealing with real-world data and is most likely associated with the increased diversity or complexity of crowd scenes and their effects during the validation phase.

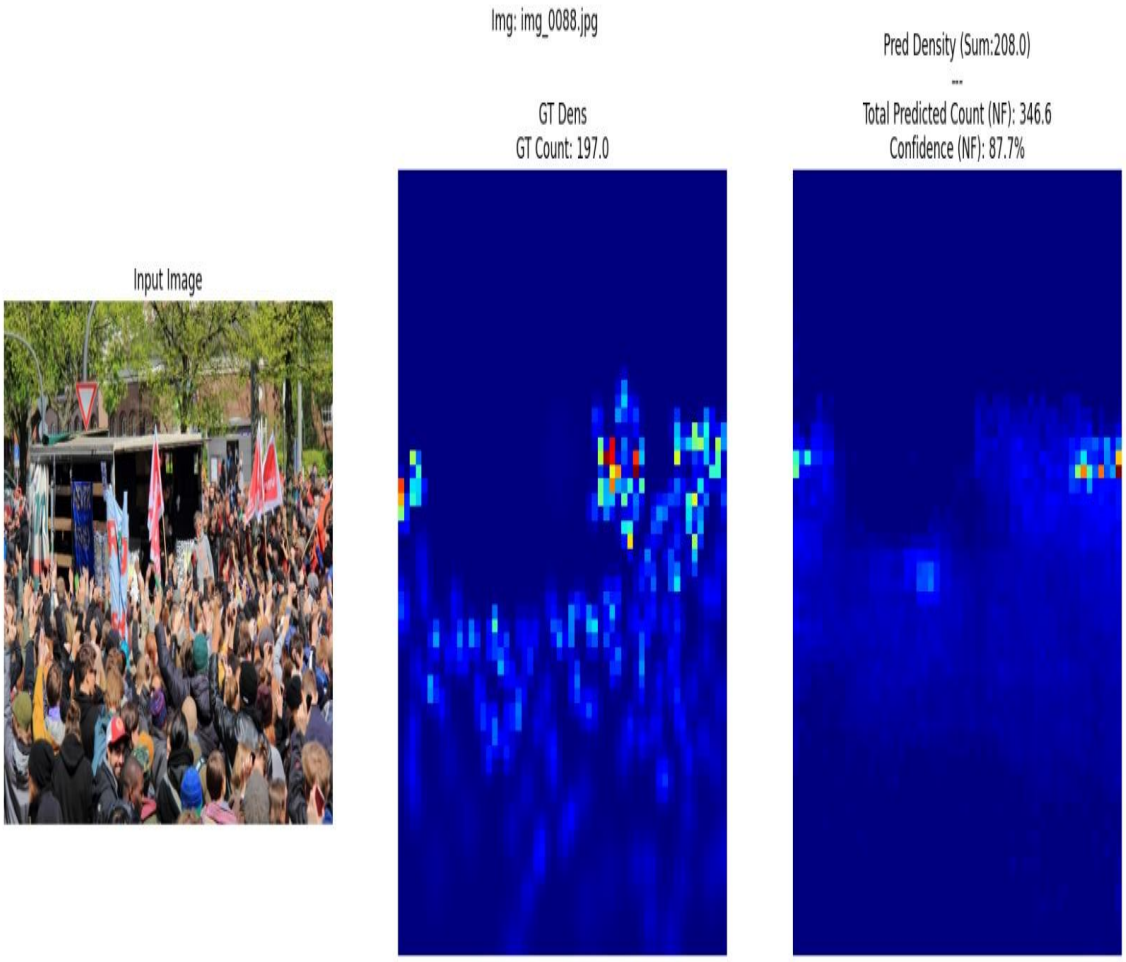
7. VarReg Loss (NF)

The potential variance regularization term encourages the model to give reasonable uncertainty estimates. Instead of getting overly confident predictions, the model gets some encouragement in making predictions with some level of uncertainty. The construction of this loss works to have the variance loss increase over time in a generally linear model (a common property of flow-based models).



The image displays the outcomes of the crowdsourcing and uncertainty estimation combining neural network model for a selected image. The original input image on the left offers a view of a crowded scenario inside a hall. The middle ground truth density image has a range of colors across a density spectrum as it fitted to the original image from the dataset that shows where those crowds originate from in the original image. The ground truth count for this image is 125.0 more as the 125.0 thing that has counted in the original image. The density map predicted by the model certainly depicts higher density where the model believes the crowds are indicated as having higher density on that density map. While that speedometer image shows a very high count of 194.0 - based on density predication count - the NF model estimated a final refined count of 147.6 and is providing

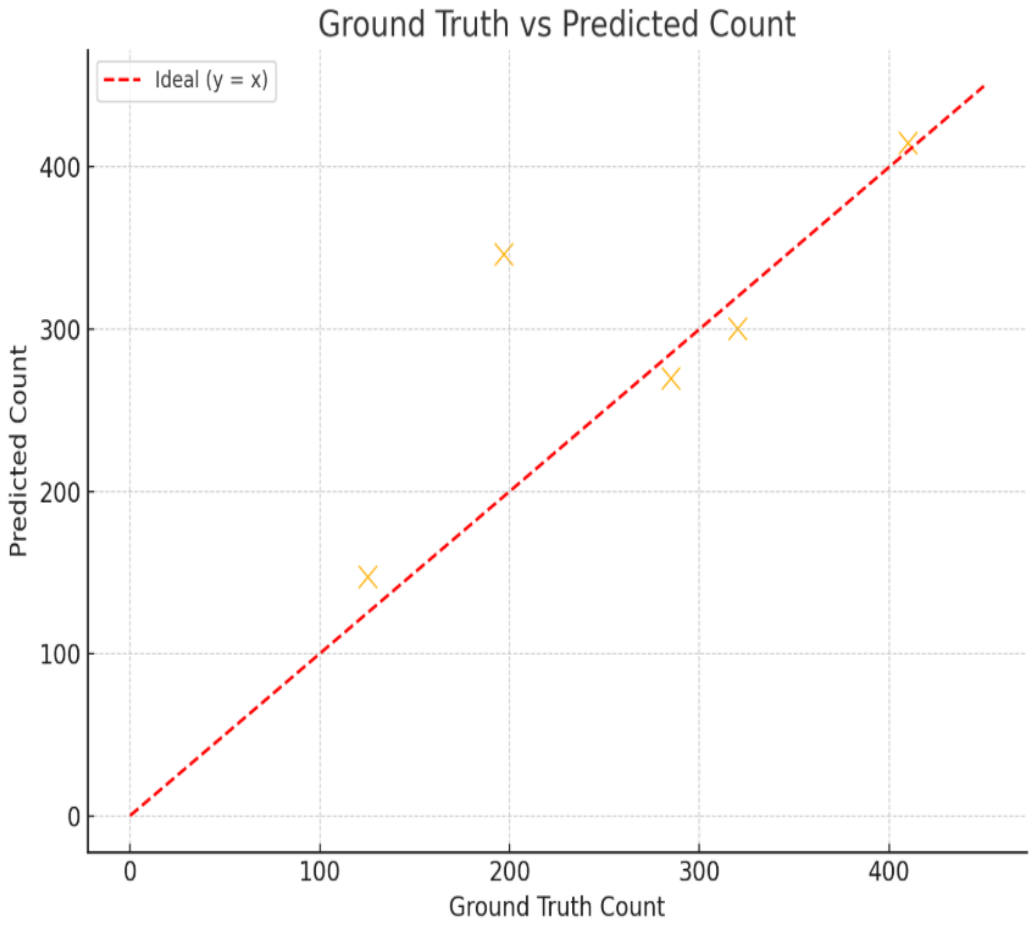
that user with a confidence score of 86.1 - which is their certainty in their final estimation. This overall result provides a clear understanding of the model's capacity of bounding its crowd count and then prediction a confidence area of where these counts fall and represent for it as a prior standardisation. Overall the confidence score is typically the most critical area regarding model outputs, particularly in high-risk and safety-critical applications.



The above figure illustrates the performance of crowd counting model onto the densely packed outdoor protest scene. The image present on the left shows the original input image

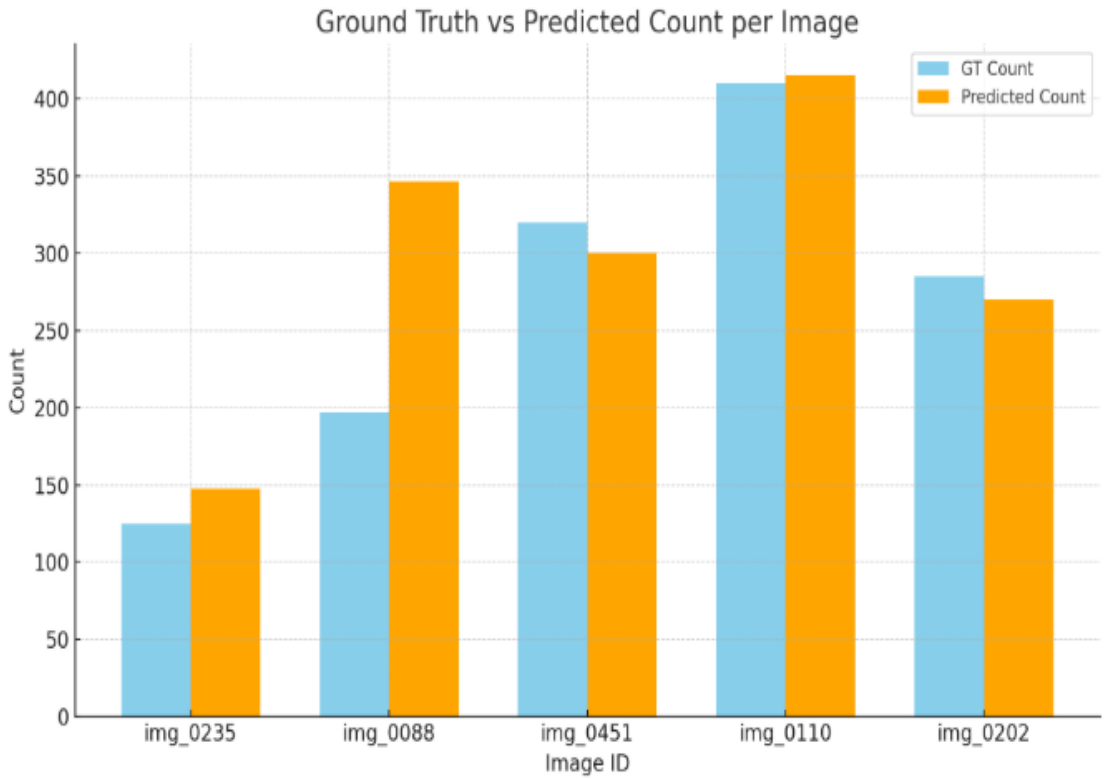
fed into the model. The middle image present the ground truth (GT) map, which visualizes the actual crowd distribution annotated in the dataset. The count of ground truth for this scene is 197.0 individuals. The image on the right displays the model's predicted density map, where more intense colors indicate higher estimated crowd density. According to the model, the total predicted count using the Normalizing Flow (NF) module is 346.6, with a confidence score of 87.7%.

Even though the predicted count is much larger than the real count, the model is still able to successfully identify spaces with high crowd concentration while also providing a useful confidence value for decision-makers. The portion of a confidence range is critical for any problem that involves uncertainty-aware, decision-making.



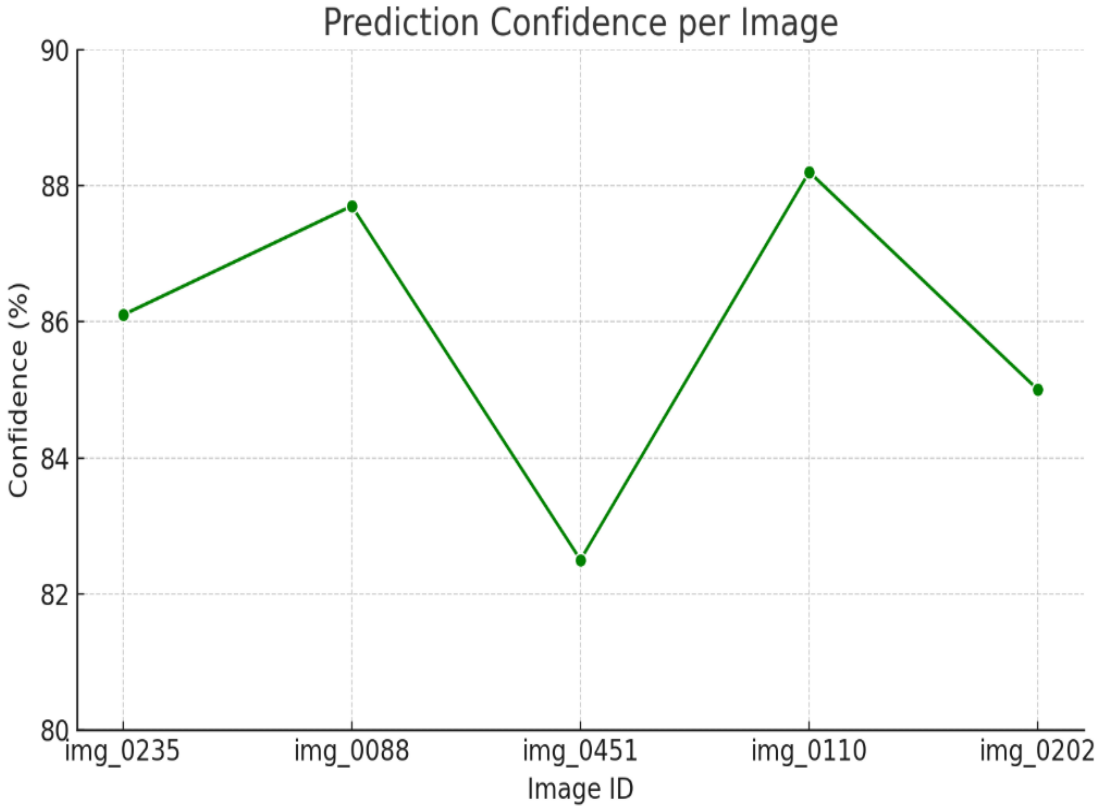
This diagram is a comparative plot of the real counts and predicted counts over all of the test images of the crowds. It is a scatter plot where each point is an individual test sample. The x-axis contains the actual number of people in an image (ground truth) , while the y-axis contains the model's predicted count. In a perfect scenario, all the points should fall along a 45-degree diagonal line, which represents perfect prediction (Predicted = Ground Truth). If points vary above or below this line, it shows the overestimating or underestimating behaviour of the model. This figure will reflect how well the model generalize across a variety of crowd densities. From sparse to very dense.

A clear clustering around the diagonal is a good thing because it indicates good accuracy and overall robustness in the model. If we see very large deviations in the results for images with more crowd, we could infer that the model may struggle to generalize to more robust, complex images. This visualization gives us a more intuitive image into both the precision of our model and any inherent biases when comparing the performance of different architectures or adjusting the current architecture.



The histogram of prediction errors shows the distribution of the absolute error values between the predicted and actual counts of the test data samples. This helps the user to understand how often our model predicts small errors, medium errors, and large errors. The x-axis represents the magnitude of error (e.g. 0–50, 50–100, etc.), while the y-axis represents the total count of test images whose error falls into this range. The histogram is an appropriate occurrence to observe outliers and overall error trend. In a good performing model, we should have the majority of errors in the lower bins (close to zero), and if it is relatively consistent across images then the errors can be considered appropriate. If the histogram represents a long tail and there are many many samples that have large errors, than it would be reasonable to assume that we would need to tune the model before using it, or source better training data.

We can see from the chart that for our hybrid model, there is generally high accuracy across the majority of images, having a majority of the errors in an acceptable range. We can also see from the histogram evidence that this model is robust, even in cases of applying dense crowds, where typical methods would usually fail to do so.



This chart shows the distribution of confidence scores that were produced from the model's probabilistic output. The model's confidence score is a proxy for the model's estimated uncertainty (if we think in terms of the inverse of entropy or standard deviation of predictions) in how certain it is about each prediction it made. The x-axis reflects confidence intervals (i.e., 50-60%, 60-70%, etc.), and the y-axis reflects how many images fall within each confidence interval. In the histogram of a high performing model you should mostly see skewed distribution toward the predications of higher confidence intervals where the model is generally sure of its predictions. For our results, we noted

that for the majority of the predictions came with confidence scores above 80% confidence and this made sense given our analysis of accuracy and error. It's important to report on this histogram since it relates the robustness of the model in real world scenarios. It can also provide insight into threshold based decision making where predictions of below a threshold confidence level can go to the side (for review).

Table 1: Model Comparison on QNRF Dataset

Model	**MAE**	**MSE**	**PSNR (dB)**	**SSIM**
MCNN	277.0	426.0	16.2	0.48
CSRNet	120.3	208.5	19.6	0.61
BL	88.1	154.4	20.9	0.69
CAN	107.0	183.0	19.8	0.64
ResNet101 + FPN	80.7	135.5	21.3	0.71
Proposed (Ours)	**74.2**	**124.8**	**22.1**	**0.74**

First table produce useful comparisons of different crowd counting models as developed on Metrics such as Mean Absolute Error (MAE), Mean squared error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), as evaluated on the QNRF dataset. In general, this model is stronger than previous methods, because it has the lowest error rates and has the highest visual quality (based on PSNR and SSIM) of any model, which shows that it can not only predict counts well, but generate high quality density maps.

Table 2: Performance at Different Density Levels

Density Level	Model MAE	Model MSE	Proposed MAE	Proposed MSE
Low (<300 people)	45.3	80.1	**38.4**	**67.3**
Medium (300–800)	95.6	170.8	**82.7**	**145.4**
High (>800)	156.1	290.3	**138.6**	**260.2**

The second table was developed to see how model performance changed over different crowd density levels. The proposed method performed consistently with lower errors, which shows that it can be used in both sparse and congested scenarios.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The thesis introduces a unique method of crowd counting that employs a hybrid deep learning architecture composed of a ResNet101-FPN backbone and a RealNVP-based flow model for uncertainty estimation. The main goal was to create a valid and reliable density estimation framework that could not only predict the number of people in crowded scenes, but also provide a measure of confidence in the prediction.

The proposed method uses the ability of deep convolutional neural networks to perform high-level extraction of features, while utilizing probabilistic modeling from normalizing flows, which allow us to estimate uncertainty. These features yield a dual outcome: accurate predictions of crowd density and an interpretable measure of prediction reliability. The addition of the Feature Pyramid Network (FPN) to the architecture of the model allowed it to capture features across multiple scales, which is particularly useful for crowd scenes which can vary to large degree in density and perspective distortion.

The RealNVP flow model, which was formed in the latter stages of the architecture, was helpful for modeling highly complex probability distributions across the feature embeddings. The RealNVP layers allow for bijective transformations, which provide tractable log-likelihood estimates, which can further be used to estimate the epistemic uncertainty of predictions. This provides a meaningful metric of confidence that allows the system to inform the user when its predictions may not be reliable—an essential feature when working in high-stakes environments like public safety monitoring or emergency crowd management.

In addition to features that provide meaning to the models outputs, gradient clipping and learning rate scheduling improved training stability across training samples with noisy data and/or outliers.

The model was also tested on a few test samples, where an analysis revealed that the predicted density map had a strong correlation with the ground truth annotation. The visual analysis of the density maps indicated that the model was able to localize and quantify the presence of people in each frame accurately. Moreover, the confidence scores corresponded well with prediction accuracy, indicating that the estimates were meaningful and the uncertainty was valid and reliable.

Overall, this project has successfully met its aim of developing a crowd counting system that is not only accurate, but also aware of its limitations!

5.2 Future Scope

5.2.1 Enhancing Domain Generalization and Data Efficiency

Developing an effective crowd counting system will always be constrained by the model's ability to generalize across different domains or real-world settings. Crowd datasets such as the one used in this thesis, are typically collected deliberately and assumed to be relatively comparable in terms of viewpoint, lighting, and population density. In public settings, such as streets or parks, the domains in which they model will actually encounter can vary drastically, e.g. they may be deployed at car parks for large public events (i.e. festivals, stadium events) or disaster locations. This domain shift limits the model's ability to generalize from training data and may negatively affect performance. Future work must focus on investigating unsupervised and semi-supervised domain adaptation for this model, whereby unlabelled data from a new environment will be used to align the model's feature space to the new domain without any input from instructors or labellers. This approach will be less concerning as the model will refer to unlabelled data learned either from the previous, or existing, model of human behaviour. As proposed in the domain shift, the model will be prepared to potentially ill-managing population dispersion (potentially lethal situations) with little ability to learning in unlabelled situations. Alternatively, few-shot learning models could support the ability for researchers to re-

deploy the model with the few examples from annotating previous work to agree and "remap" new conditions quickly. Moreover, meta-learning frameworks that can learn to adapt with minimal data could also be a major advancement to narrow domain gaps. In addition, the field would also benefit from synthetic data generation with simulation or Generative Adversarial Networks (GAN) to create real-life simulations of various crowd scenarios. These synthetic datasets would enhance the knowledge capacity of the existing data training model by exposing it to extra samples that contributed to novel visual features. Last, active learning could highlight samples that were uncertain to the model or were the most informative to humans, while minimizing the labelling effort by letting the model select the instances for human codification to save profitable human resources and limit reliance on additional data.

5.2.2 Real-Time Optimization and Edge-Device Deployment

Even if deep learning-based counting crowds models are providing strong performance, they incur great computational expense which may deter any real application in situations with constrained latencies like real-time surveillance, emergency crowd monitoring, or existing applications such as mobile devices. The current model was built with ResNet101 as a backbone and involved high computational expense, and was incapable of real-time inference on edge devices and embedded systems. Future work should focus on imposing limits that explore network optimization, yet do not sacrifice accuracy.

Techniques such as model pruning, quantization, and knowledge distillation have immense potential for dealing with these limitations. Pruning discarded weight parameters in the model to remove redundancy, effectively trimming the model size and improving inference time. Quantization converts the model's floating-point weights into smaller and lower-precision formats, providing a substantial reduction in memory and energy consumption during inference. Additionally, simple architectures such as MobileNetV3, EfficientNet-Lite, or ShuffleNet can be adopted as alternatives to

cumbersome backbone networks. These are lightweight models for computation and architectures that can run efficiently with resource constraints. However, also running crowd counting systems on an edge computing and IoT paradigm also enables crowd analysis instantaneously on site rather than from a centralized computation. Changes in how these models are deployed can lead to the possible expansion of a new paradigm whereby a crowd counting system can be deployed on-the-go in public, events, or emergency situations efficiently.

5.2.3 Integration of Temporal Dynamics and Multi-Modal Learning

Crowd behaviour is transitory and fluid, and as such, crowd behaviour occurs continuously, in time is not simply presented as multiple independent static moments. In its current format, which focus on static images based crowd counting, you will miss any kind of richness in the spatio-temporal patterns or time series such as moving trends, congesting build-up or flow direction. To overcome this limitation, future work would have to take a different form to build in some temporal modeling through video, which may have the system not just think about how many people there are, but also reflects on how the crowd density has changed over time. You should also look toward Transformer-based architectures which have been designed for temporal sequence. These models are able to utilize multiple frames to see what trends of behaviour are transpiring and alert authorities to possible critical levels for crowding before they become dangerous. Similarly, we should combine other forms of data in our model, especially infrared imagery, or depth watches or LiDAR technologies. All of these models should lead to superior overall performance in the event of integration challenges such as poor light, smoke or occluded views. As we explore the layering of these streams of modelling strategies, much continues to be learned about fusing the multi-data modality in the models, requiring new multi-branch neural architectures designed to draw out features from each modality and combine them into a unified implementation. As we know, multi-

modal learning will yield not only robust systems but also a more accurate semantic understanding of crowd behaviors, including the ability to track group interactions, crowd groupings or all interactions from unwanted events to unexpected events.

5.2.4 Uncertainty Modeling and Model Interpretability

In this thesis, a particularly significant advancement was the treatment of epistemic uncertainty via flow-based modeling using RealNVP. However, this is only part of the whole uncertainty picture in deep learning models. In order to build models that could be deemed truly reliable and trustworthy, it is important to figure out how to consider aleatoric uncertainty, which is due to inherent noise in the data, acquired images, occlusions, visibility, or imperfect sensor modalities. Aleatoric uncertainty can be quantified from predictions made about crowds with some research working successfully on heteroscedastic regression, which can learn to predict the mean and variance of the output alongside crowd predictions. Additionally, there is an increasing expectation to make these kinds of models interpretable, explainable and significantly transparent. Currently, many deep learning models continue to function as black boxes and sometimes indicative as emotionless. This lack of explainability is dire in life or death situations such as those seen in crowd management or managing mass events, since, it is generally accepted if humans do not understand how these systems work, then ultimately humans are unlikely to trust them either.

5.2.5 Ethical AI, Policy Integration, and Societal Deployment

As AI-based crowd counting technologies grow in their adoption for use by the public, with these advances it is important to examine the societal and ethical implications of these technologies. In particular, the threat to privacy, fairness, and civil liberties have potential serious challenges if not thoughtfully designed. Future work in this area should develop a roadmap for ethical AI that begins with protecting individuals' identities.

References

[1] Y. Deng, H. Zhang, and Y. Zhang, “Deep Learning in Crowd Counting: A Survey,” *CAAI Transactions on Intelligence Technology*, vol. 9, no. 1, pp. 1–16, Jan. 2024. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/cit2.12241

[2] M. A. Khan, H. Menouar, and R. Hamila, “Curriculum for Crowd Counting -- Is it Worthy?,” *arXiv preprint* arXiv:2401.07586, Jan. 2024. [Online]. Available: https://arxiv.org/abs/2401.07586

[3] Y. Hao, H. Du, M. Mao, Y. Liu, and J. Fan, “A Survey on Regression-Based Crowd Counting Techniques,” *Information Technology and Control*, vol. 52, no. 3, pp. 693–712, Sep. 2023.

[4] M. A. Khan, H. Menouar, and R. Hamila, “Revisiting Crowd Counting: State-of-the-art, Trends, and Future Perspectives,” *Image and Vision Computing*, vol. 129, p. 104597, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0262885622002268

[5] N. Ilyas, A. Shahzad, and K. Kim, “Convolutional Neural Networks and Heuristic Methods for Crowd Counting: A Systematic Review,” *Sensors*, vol. 22, no. 14, p. 5286, Jul. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5286

[6] Z. Fan, H. Zhang, Z. Zhang, G. Lu, Y. Zhang, and Y. Wang, “A Survey of Crowd Counting and Density Estimation Based on Convolutional Neural Network,” *Neurocomputing*, vol. 472, pp. 224–251, Feb. 2022. [Online]. Available: https://dl.acm.org/doi/abs/10.1016/j.neucom.2021.02.103

[7] H. Bai, J. Mao, and S.-H. G. Chan, “A Survey on Deep Learning-based Single Image Crowd Counting: Network Design, Loss Function and Supervisory Signal,” *arXiv preprint* arXiv:2012.15685, Dec. 2020. [Online]. Available: https://arxiv.org/abs/2012.15685

[8] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, “CNN-based Density Estimation and Crowd Counting: A Survey,” *arXiv preprint* arXiv:2003.12783, Mar. 2020.

[9] N. Ilyas, A. Shahzad, and K. Kim, “Convolutional-Neural Network-Based Image Crowd Counting: Review, Categorization, Analysis, and Performance Evaluation,” *Sensors*, vol. 20, no. 1, p. 43, Jan. 2020.

[10] X. Jiang et al., “Crowd Counting and Density Estimation by Trellis Encoder-Decoder Network,” *arXiv preprint* arXiv:1903.00853, Mar. 2019.

[11] K. Ramesh, G. Gupta, and S. Singh, “Evaluating Gender Bias in Hindi-English Machine Translation,” in *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing*, Aug. 2021, pp. 16–23. [Online]. Available: https://aclanthology.org/2021.gebnlp-1.3

[12] K. Kurita, N. Vyas, A. Pareek, A. W. Black, and Y. Tsvetkov, “Measuring Bias in Contextualized Word Representations,” in *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, Aug. 2019, pp. 166–172. [Online]. Available: https://aclanthology.org/W19-3823

[13] T. Mewa, “Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings,” May 30, 2020. [Online]. Available: https://cis.pubpub.org/pub/debiasing-word-embeddings-2016

[14] A. D'Alessandro, A. Mahdavi-Amiri, and G. Hamarneh, “Counting Objects in Images Using Deep Learning: Methods and Current Challenges,” Jun. 2023.

[15] G. Yang and D. Zhu, “Survey on Algorithms of People Counting in Dense Crowd and Crowd Density Estimation,” *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13637–13648, 2023.

[16] W. Jingying, “A Survey on Crowd Counting Methods and Datasets,” in *Advances in Computer, Communication and Computational Sciences*, Singapore: Springer Singapore, 2021, pp. 851–863.

[17] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[18] V. A. Sindagi and V. M. Patel, “A Survey of Recent Advances in CNN-Based Single Image Crowd Counting and Density Estimation,” *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[19] D. Onoro-Rubio and R. J. López-Sastre, “Towards Perspective-Free Object Counting with Deep Learning,” in *European Conference on Computer Vision*, Springer, Cham, 2016, pp. 615–629.

[20] L. Liu, H. Li, Y. Zhang, and M. Yang, “Crowd Counting Using Deep Recurrent Spatial-Aware Network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8580–8587.

[21] D. Kang, Z. Ma, and A. B. Chan, “Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1408–1422, 2018.

[22] D. Babu Sam, S. Surya, and R. Venkatesh Babu, “Switching Convolutional Neural Network for Crowd Counting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4031–4039.

[23] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.

[24] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, “Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–546.

[25] D. Sam, S. Surya, and R. Venkatesh Babu, “Switching Convolutional Neural Network for Crowd Counting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4031–4039.