information

MDPI

Article

Crowd Counting in Diverse Environments Using a Deep Routing Mechanism Informed by Crowd Density Levels

Abdullah N Alhawsawi [1], Sultan Daud Khan [2,*] and Faizan Ur Rehman [3]

[1] Department of Information and Scientific Services, Custodian of the Two Holy Mosques Institute for Hajj and Umrah Research, Umm Al-Qura University, Makkah 24236, Saudi Arabia; anhawsawi@uqu.edu.sa
[2] Department of Computer Science, National University of Technology, Islamabad 44000, Pakistan
[3] Saudi Data and Artificial Intelligence Authority, Riyadh 11525, Saudi Arabia; faizanurrehman@gmail.com or faizan.rahman@naseej.com
* Correspondence: sultandaud@gmail.com or sultandaud@nutech.edu.pk

Abstract: Automated crowd counting is a crucial aspect of surveillance, especially in the context of mass events attended by large populations. Traditional methods of manually counting the people attending an event are error-prone, necessitating the development of automated methods. Accurately estimating crowd counts across diverse scenes is challenging due to high variations in the sizes of human heads. Regression-based crowd-counting methods often overestimate counts in low-density situations, while detection-based models struggle in high-density scenarios to precisely detect the head. In this work, we propose a unified framework that integrates regression and detection models to estimate the crowd count in diverse scenes. Our approach leverages a routing strategy based on crowd density variations within an image. By classifying image patches into density levels and employing a Patch-Routing Module (PRM) for routing, the framework directs patches to either the Detection or Regression Network to estimate the crowd count. The proposed framework demonstrates superior performance across various datasets, showcasing its effectiveness in handling diverse scenes. By effectively integrating regression and detection models, our approach offers a comprehensive solution for accurate crowd counting in scenarios ranging from low-density to high-density situations.

Keywords: crowd counting; regression models; head detection; crowd surveillance; deep learning

check for updates

Citation: Alhawsawi, A.N.; Khan, S.D.; Ur Rehman, F. Crowd Counting in Diverse Environments Using a Deep Routing Mechanism Informed by Crowd Density Levels. *Information* **2024**, *15*, 275. https://doi.org/10.3390/info15050275

Academic Editor: Francesco Fontanella

Received: 20 March 2024
Revised: 24 April 2024
Accepted: 30 April 2024
Published: 13 May 2024

1. Introduction

Automated crowd analysis is a challenging problem and has received tremendous importance from the research community over the last decade. Due to the increasing population, many people attend mass events, such as religious festivals, marathons, concerts, etc. Although these events are organized for entertainment or fulfillment of religious obligations, sometimes these peaceful events end up with a crowd disaster. To predict and prevent crowd disasters, surveillance cameras are mounted in different locations of venues, where security personnel manually analyze the whole crowd with the naked eye. Studies have proved that such manual analysis is a tedious job and is usually prone to errors [1].

To automatically analyze a crowded scene, researchers have developed different models and methods that automatically analyze a crowd and understand crowd dynamics [2]. Crowd analysis includes various applications, including crowd counting [3,4], congestion detection [5], crowd tracking [6], crowd behavior understanding [7–9], and more. Crowd counting, in particular, has gained significant importance within the research community.

Crowd counting in naturalistic scenes has numerous applications and is significant both from political and geo-political perspectives [10]. The task of crowd counting is to count the number of participants attending an event. Currently, most of the state-of-the-art crowd-counting methods can be divided into two groups: (1) regression-based approaches and (2) detection-based approaches.

Regression-based methods regress density information and estimate the count without localizing people. Zhang et al. [11] proposed a network that simultaneously solves the counting and density estimation problems. This method relies on the generation of perspective maps that enhance the counting accuracy; however, the acquisition of perspective maps for every scene increases the computational cost. Similarly, a Multicolumn Convolutional Neural Network (MCNN) is proposed in [12] that consists of three columns. Each column implements a small network with a different receptive field with the aim of solving multi-scale problems. The Switching Convolutional Neural Network is proposed in [13] that contains multiple Convolutional Neural Network (CNN) regressors with different receptive fields, and a switch classifier is trained to route the patch to one of the CNN regressors that can best estimate the count. While regression-based approaches excel in high-density scenarios, they tend to overestimate crowd counts in low-density situations.

Detection-based crowd-counting methods not only estimate crowd counts but also localize the people in the scenes. Composition loss was introduced in [14] to address the simultaneous challenges of counting, density estimation, and localization. Similarly, Locate, Size and Count Convolutional Neural Network LSC-CNN [15] is proposed that localizes every person in a crowded scene, estimates the bounding box (size) of visible heads and finally counts the number of people. Scale-Driven Convolutional Neural Network (SD-CNN) [16] is proposed to count the number of people in high-density crowds by detecting visible heads. These approaches work well in low-density scenes; however, their performance degrades when applied in high-density situations. Therefore, we need a "one-model method" that can accurately count people in all kinds of scenes.

To address the above problems, we proposed a framework that combines the advantages of both regression-based and detection-based models by exploiting the variations of crowd density within an image to accurately predict the crowd counts. Generally, the proposed framework adopts a routing strategy that routes the image patch to one of two counting modules based on the density level. The framework divides the input image into non-overlapping patches of fixed size. Then, each patch is classified into four classes, i.e., Low, Medium, High, and No Crowd. Then, the patches are provided as input to the Decision Block (DB), where, based on the classification label, the patches are routed to either of two modules, i.e., the Detection Network or Regression Network. The network estimates the count in each patch and then calculates the final count by summing the count from all patches.

The proposed framework offers the following contributions:

1. A unified deep-learning framework is proposed that estimates crowd count in diverse scenes.
2. We introduce a Crowd Classifier (CC) that classifies the patches into four categories, including Low Crowd, Medium Crowd, High Crowd, and No Crowd.
3. A novel Head-Detection (HD) network is introduced for the efficient detection of human heads in complex scenes, leveraging iterative deep aggregation (IDA) to extract multi-scale features from various layers of the network.
4. A novel Crowd-Regression Module (CRM) is introduced, which utilizes an Atrous Convolution Grid (ACG) to densely sample a wide range of scales and contextual information for accurate crowd count estimation.
5. An effective routing strategy is developed that efficiently routes the patches to either a detection network or regression module based on crowd density variations within an image.

The remaining sections of the paper are structured as follows: Section 2 discusses related work, Section 3 outlines the proposed methodology and detailed experiment results along with performance analysis are presented in Section 4. Concluding remarks are provided in Section 7.

2. Related Work

In this section, we briefly overview crowd-counting methods recently proposed in the literature. In general, crowd-counting methods can be classified into two main groups: (1) Regression-based methods and (2) Detection-based methods.

2.1. Regression-Based Methods

These approaches utilize machine-learning models to estimate the count through regression analysis between the image/patch and the count. Regression-based methods can be subdivided into two categories: (1) statistical machine-learning models and (2) deep-learning models.

Statistical machine-learning models extract statistical features, Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradient (HOG), etc., and employ techniques such as support vector regression [17], Gaussian process regression [18], Random forest [19], etc., to estimate the crowd count. Generally, these methods require the computation of complex handcrafted features. Essentially, handcrafted features, such as SIFT, HOG, Bag-of-Words, etc., are extracted from the image. Subsequently, a classifier is trained to categorize the image into different classes. Idrees et al. [20] proposed a crowd-counting framework by the fusion of multiple features. The authors identified that a single feature is not enough to count people in high-density crowds.

Regression-based deep-learning models for crowd counting are designed to predict the count of individuals directly from a given image. Unlike models focused on detecting or localizing individuals, regression models generate a density map. This density map is used to estimate a continuous value representing the crowd count. Wang et al. [21] introduced a method for crowd counting that adopts an adaptive density map generator strategy, which refines existing density maps using a learned refinement network. The refinement process is integrated into an end-to-end framework, allowing joint training with the crowd-counting network. Dong et al. [22] proposed MMNet, which addresses the challenges of crowd counting, such as occlusions and scale variations. The method addresses the scale problem using various filter sizes and integrates features from different layers of the network to handle head scale variations effectively. Li et al. [23] presented CSRNet, aiming to precisely estimate counts and produce high-quality density maps in densely populated scenes. The network comprises a front-end Convolutional Neural Network (CNN) for 2D feature extraction and a dilated CNN for the back end, utilizing dilated kernels to achieve larger reception fields without pooling operations. Xu et al. [24] introduced a method that addresses the challenge of crowd counting with partial annotations. Sindagi et al. [25] introduced the contextual pyramid CNN to estimate the crowd density and count by integrating global and local contextual information in crowd images. Similarly, Zhai et al. [26] presented a novel framework for crowd counting. The framework employs a discriminative feature extractor to extract hierarchical features and utilizes a hierarchical fusion strategy to mine semantic features in a coarse-to-fine manner. Zhang et al. [27] proposed a framework that counts the number of people in multiple views. Zhai et al. [28] introduced a framework consisting of three modules. The initial module extracts multi-scale features using a feature pyramid network. The second module is an attention module designed to suppress less critical information while preserving vital features for crowd counting. Finally, the third module, a multi-scale aggregation module, consolidates features from various layers of the network. Guo et al. [29] introduced a crowd-counting framework employing the Ghost Attention Pyramid Network as a feature extractor (encoder). The extracted features are subsequently fed into the channel attention module to efficiently capture discriminating crowd regions. Moreover, there have been extensive surveys on various crowd-counting algorithms [3,30]. These surveys offer comprehensive insights into state-of-the-art crowd-counting models and delve into the strengths and limitations of different approaches while also discussing the datasets employed in evaluating these models.

Although the regression-based deep-learning models achieve good performance in high-density crowds, this can be attributed to the fact that images in high-density crowds often exhibit regular and repetitive crowd structures. As a result, regression-based methods excel in such scenarios, as they effectively capture generalized density information. However, these methods suffer from the following limitations. (1) These methods do not localize the people in the environment, and (2) these methods overestimate the crowd counts when applied in low-density crowds.

2.2. Detection-Based Methods

Detection-based approaches involve training object detectors to identify the position of each person in the crowd, with the total number of detections indicating the overall crowd count.

Most of the existing methods approach the crowd-counting problem by treating it as the detection of faces, heads, or pedestrians in images. Traditional methods, for example, Viola and Jones [31] use Haar-like features and learn a haar-cascade classifier. More recently, Ren et al. [32] has incorporated temporal information using the conditional random field (CRF) to further improve the accuracy of the Viola and Jones method. The deformable parts model (DPM) [33] is a part-based model that uses a histogram of oriented gradient features to detect different parts of the human body. These models work well in low-density crowds; however, the performance degrades when applied in high-density crowded scenes. This is because these rely on complex handcrafted features that do not have discriminating power.

Recently, deep-learning models have achieved superior performance in object detection, segmentation, and classification tasks. In recent years, several methods [34,35] have been proposed to detect human faces in videos and images. Other methods have exploited contextual information of the scene to detect human faces in complex scenes [36,37]. Hao et al. [38] proposed a sophisticated method that detects tiny human faces in high-density crowds by leveraging contextual information about the scene. Khan et al. [39] proposed a novel method for crowd counting in high-density crowds by introducing an end-to-end scale-invariant Head-Detection framework. The proposed framework employs specialized scale-specific convolutional neural networks (CNNs) with different receptive fields to handle diverse scales effectively. Shami et al. [40] introduced a head detector based on CNN for crowd counting. Lian et al. [41] introduced a method that simultaneously estimates headcount and localizes the humans through a detection-based strategy. Zhou et al. [42] introduced a framework that combines multiple kernel learning (MKL)-based fast head detection and shape-aware matching. Saqib et al. [43] introduced a crowd-counting method that improves the performance of pedestrian detectors via motion-guided filters (MGF).

While the aforementioned methods excel in low-density crowds, where either all or part of the human body is visible, they face challenges when applied to high-density crowded scenes. This limitation is attributed to the fact that detection methods relying on facial features encounter difficulties, particularly when the person is far away from the camera or turns their back to it. Furthermore, facial features become hardly visible in high-crowded situations due to severe occlusions.

Conversely, pedestrian detection relies on identifying the entire pedestrian, which becomes challenging in high-density crowded situations where the full body of the pedestrian is not visible. In such cases, pedestrian detectors typically struggle to perform effectively.

3. Proposed Methodology

In this section, we will delve into the various components of the proposed framework. The pipeline of the framework work is illustrated in Figure 1. Generally, the framework comprises four major modules: the Crowd Classifier (CC), Patch-Routing Module (PRM), Head Detector (HD), and the Crowd-Regression Module (CR). The primary objective of this framework is to estimate the number of people within a given image. The initial step involves dividing the input image into non-overlapping patches. Subsequently, these patches serve as input to the CC, which classifies them into four distinct categories: No Crowd (NC), Low Crowd (LC), Medium Crowd (MC), and High Crowd (HC). Based on the classification outcomes, the PRM directs the patches towards the Head-Detector Module and the Crowd-Regression Module. The counting modules estimate the count in the input patch, and then the count accumulator provides the final count by accumulating the count of all patches of the input image.

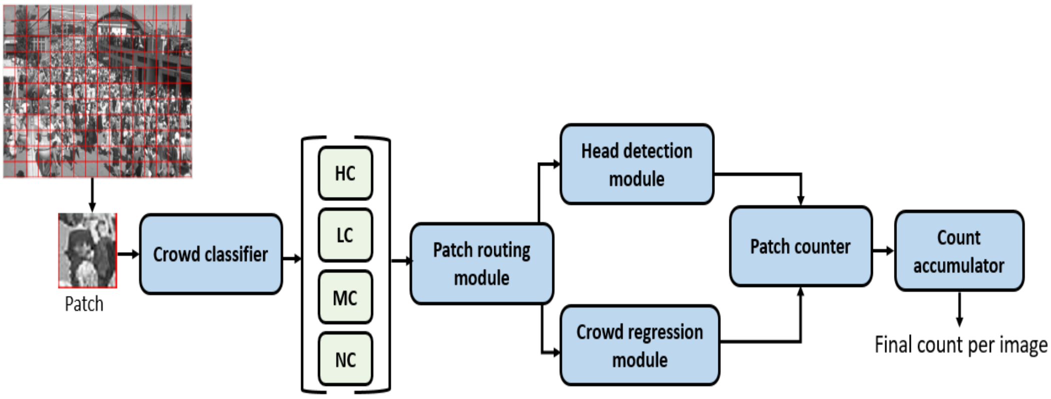

Figure 1. Overall pipeline of the proposed crowd-counting framework.

The Head-Detection Module is responsible for processing Low-Density Crowd and Medium-Density Crowd patches. It employs a deep-learning model to detect the number of heads in each patch. On the other hand, the Crowd-Regression Module handles high-density crowd patches and estimates the count within each patch. Detailed information on each module is provided as follows:

3.1. Crowd Classifier

Crowd classification plays a vital role and serves as a preprocessing step in crowd analysis. Most of the existing crowd-counting models are based on regression techniques [20,44,45], where the models learn repetitive structures within the whole image to estimate the count. However, these regression-based models tend to produce false positives when applied to images with No Crowd, which causes significant inaccuracies in crowd counting. Since these regression models are blind and learn from the patterns present in images, they may not distinguish the difference between individual heads and the background. This fact is illustrated in Figure 2.

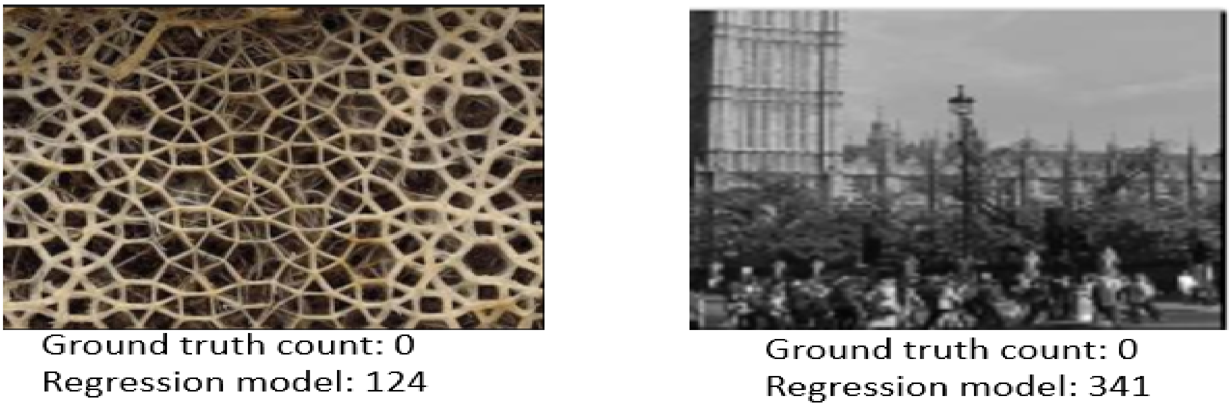

Figure 2. Samples of non-crowded scenarios. The ground truth is 0, while the regression model [44] still predicts the count of people.

To remedy this problem, we introduce a crowd-classification model to characterize the crowd image from non-crowd images. However, it has been observed that classifying the whole image usually leads to inaccuracies in crowd counting, as some areas within the image may be significantly denser than others [20]. Therefore, instead of considering the whole image, we divide the image into smaller, non-overlapping patches by assuming that the crowd density is uniform across the whole image. In this manner, the model assigns significance to each region of the image, therefore identifying non-crowded areas, which enhances the efficiency and precision of the counting process. With this approach, the framework avoids routing non-crowded patches to the other two counting modules and accommodates the variations in crowd density across different areas of the scene, which results in improved counting accuracy and processing speed.

Let I be the input image of arbitrary size. We divide the input image I into N number of non-overlapping patches. Let $\{p_1, p_2, \ldots, p_N\}$ be the patches extracted from the input image. All patches have identical dimensions, with each patch comprising S × S pixels. After the extraction of patches, we then preprocess the patches to make them suitable for training the deep-learning model. During the preprocessing step, the patches are resized to 224 × 224 pixels to make them fit for the input of the deep-learning model. Subsequently, the normalization step is employed to reduce the impact of brightness and contrast among different images. The patches are then converted to RGB color space as the deep-learning model utilizes RGB format during training.

For the patch-classification task, we employ the ResNet-152 deep-learning model; however, any deep-learning model may be employed. The selection of ResNet-152 is based on its specialized architecture, characterized by the presence of residual connections, which enables the network to learn complex features while effectively mitigating the issue of vanishing gradients. These attributes greatly enhance ResNet-152's generalization capabilities, which is crucial for achieving accurate patch classification across a diverse range of patches.

To train ResNet-152 on the obtained patches, we first accumulate the patches from all training images of the dataset and then label each patch as one of the four categories. To label the patches, we employ an automated approach that utilizes dot annotations available for each image. These dot annotations are the markers indicating the presence of human heads within the image. For every patch extracted from the image, we perform a straightforward procedure by counting the number of dots within the patch area. The cumulative count of these dots is then used to determine the label for the patch. This method simplifies the patch-labeling process, as the presence of dots offers a straightforward and automated means of associating each patch with its corresponding crowd density label, making it a practical and efficient approach for crowd analysis tasks. We generate class-wise patches according to the density levels defined in [46]. An "LC" class label is assigned to a patch if the count is greater than 0 and less than 10. An "HC" is assigned to a patch if the count is greater than 15. An "MC" is assigned to patches if the count is greater than 10 but less than 15, and an "NC" label is assigned to patches where the count is 0. The overall pipeline of the patch-labeling process is illustrated in Figure 3.

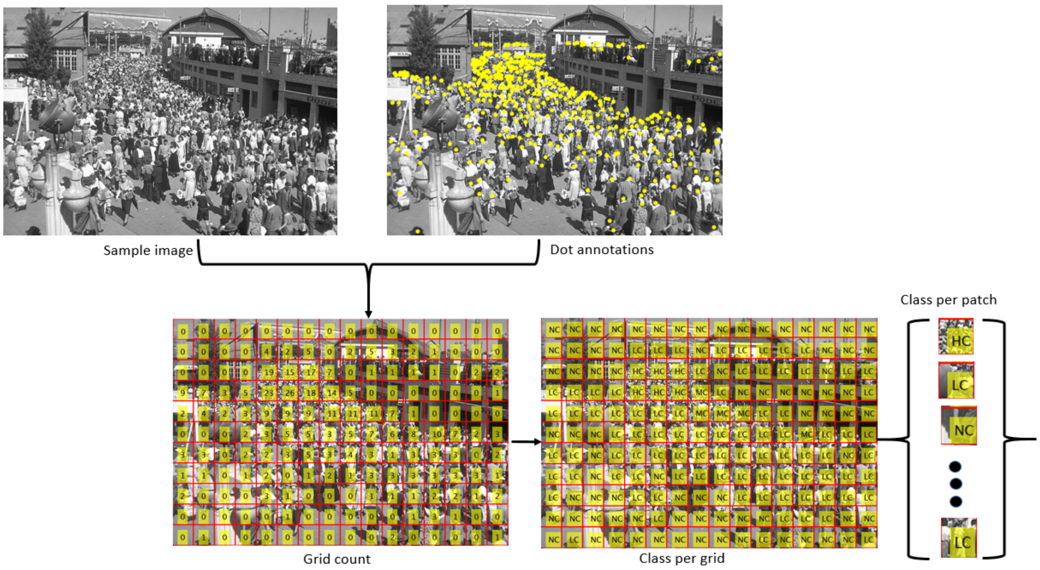

Figure 3. Pipeline of patch-wise labeling process. The algorithm utilizes the input image and its corresponding dot annotations to generate a grid-wise count. The grid-wise count is then transformed into class-wise patches for network training.

3.2. Patch-Routing Module

The Patch-Routing Module (PRM) plays a crucial role in deciding whether a specific image patch should be directed to the regression network or the detection network for further processing. As previously mentioned, when dealing with crowd counting in diverse and complex scenes with varying densities, it becomes imperative to judiciously determine the destination of each image patch to attain the best possible results. The PRM accomplishes this task through a heuristics-based approach, which involves applying a set of predefined rules to determine the most appropriate network for processing the patch. These rules are typically established based on prior experience and an understanding of the problem.

It is worth mentioning that in high-density crowd scenarios, achieving accurate counting through the detection of human heads is very challenging. Conversely, in low-density crowd scenarios where the head of each individual is readily visible, employing a regression-based model for counting is not efficient since the regression-based models tend to overestimate the crowd counts in such situations. To address the problem, we formulate some rules that intelligently direct image patches to the appropriate network modules. In this context, high-density patches are directed towards the regression module, while low-density and medium-density patches are routed to the detection module. This strategic routing of patches serves to optimize the efficiency of crowd counting by aligning the computational capabilities of each module with the specific characteristics of the input patches.

Algorithm 1 provides a detailed illustration of the routing approach and the counting process during the inference stage. The algorithm takes an image, I, as input, and it produces an output represented by the grid count CG, the size of which is equal to the size of the input image. The algorithm begins by overlaying a gird G of size N × M over the input image and initialize a count grid (CG) with the same dimension as the grid. The algorithm then iterates through each cell of the grid, where each cell of the grid represents a patch of the image. After extracting the patch, the patch is normalized and resized to 224 × 224 to make it fit the input of the crowd-classification network. The classifier feed forwards the input patch and predicts its class label. Based on the class label, the routing algorithm decides where to direct the patch. If the predicted label of the patch is "HC", the algorithm routes the patch to the regression network; otherwise, the algorithm directs the patch to the detection network. The count contained within each patch is then cumulatively aggregated within the count grid, denoted as CG.

Algorithm 1 Routing patches and counting during inference stage

```
Input: Image I, N, M
Output: Count Grid CG
  Overlay N × M grid G over the input image.
  Initialize count grid CG equal to the size of G.
  for each $i$ in $N$ do
      for each $j$ in $M$ do
          Normalize and re-size patch $p_{i,j}$
          Re-size patch $p_{i,j}$ to 224 × 224 pixels
          Classify $p_{i,j}$ in categories: LC, NC, HC, MC
          if $p_{i,j}$ is HC then
              $CG_{i,j}$ = CountRegressor($p_{i,j}$)
          else if $p_{i,j}$ is NC then
              $CG_{i,j} = 0$
          else
              $CG_{i,j}$ = HeadDetector($p_{i,j}$)
          end if
      end for
  end for
```

3.3. Head-Detection Module

Head detection in images and videos has a wide range of applications in crowd analysis and large-scale surveillance. Head detection is a special case of object detection. Although object detection in images has achieved significant progress, head detection presents a distinctive set of challenges. These challenges arise from the substantial variations in head sizes, complex background clutter, and the relatively small size of human heads within images.

The current generic object detectors face the following challenges while detecting human heads in images for counting tasks: (1) Current deep-learning-based object detectors represent the objects through bounding boxes that tightly encompass the objects. This approach is highly effective when precise ground-truth bounding box annotations are available for training. However, such annotations are not available in the crowd-counting dataset. The crowd-counting datasets usually contain dot annotations (2-D points), which represent the position of a human head in the image. This difference in annotation methodology complicates the training process for Head-Detection models, as these models are primarily designed for bounding box annotations. (2) Current deep-learning models, such as Faster R-CNN, extract deep hierarchical features from the input image by passing the input image through subsequent convolution and pooling layers. These pooling operations typically downsample the input image, leading to the loss of crucial information regarding small objects.

For precisely detecting human heads in complex scenes, we propose a simple yet effective approach by addressing the above-mentioned problems. To tackle the problem related to bounding box representation, we employ CenterNet. CenterNet adopts a keypoint-centric approach, which demonstrates exceptional performance in situations where bounding boxes are not available or in cases that involve small and densely clustered human heads. The network efficiently identifies the location of heads by predicting the central of each human head, even in crowded scenes or cases of occlusions.

Although the adoption of CenterNet solves the bounding box representation problem, and we directly use the dot annotation provided by the dataset, CenterNet in its original form may suffer from a loss of fine-grained information and may be unable to address the second problem. This is because CenterNet employs subsequent pooling operations, which leads to the downsampling of input images. This downsampling potentially results in the loss of crucial information about small objects and may result in many false positives. To address this problem, we modified the original CenterNet by incorporating the iterative deep layer aggregation strategy, which combines features from both shallow and deep layers of the network. This strategy allows for better context understanding while retaining the spatial details of tiny heads. The integration of shallow- and deep-feature layers helps the network address the downsampling problem by providing the network with more comprehensive and precise information about the small heads.

As in high-density crowds, the distance between the human heads is a few pixels. To accurately detect each head, the Head-Detection network produces a high-resolution heatmap. In this heatmap, dark pixels indicate the likelihood of a human head's presence, while bluish pixels represent the background or other objects. The overall architecture of the proposed Head-Detection framework is illustrated in Figure 4. We use ResNet-18 as the backbone of the framework. Resnet-18 consists of four blocks, namely *ResNet-1*, *ResNet-2*, *ResNet-3*, and *ResNet-4*. The network accepts the input image and applies a convolutional layer of size 7 × 7 with stride 2 followed by a max-pooling layer of size 3 × 3 and stride 2. The resultant feature map is then passed through *Resnet-1*, which employs a stack of two convolutional layers of size 3 × 3 and reduces the size of the original feature map to half. The reduced feature map is then passed through *ResNet-2*. The output of *ResNet-2* is then up-sampled by employing a deconvolutional layer and then integrated with the feature map of the *Resnet-1* using an iterative deep-aggregation module, *IDA-1*. The feature maps of the subsequent ResNet blocks are integrated through iterative deep-aggregation function Ψ, which captures the deep semantic information formulated in Equation (1).

$$\Psi = \begin{cases} F_1, & \text{if } n = 1 \\ \Psi\{N(F_1, F_2), \ldots, N(F_{n-2}, F_{N-1}), F_n\} & \text{otherwise} \end{cases} \tag{1}$$

where F_i is the feature map of ResNet block i, and N represents aggregation node.

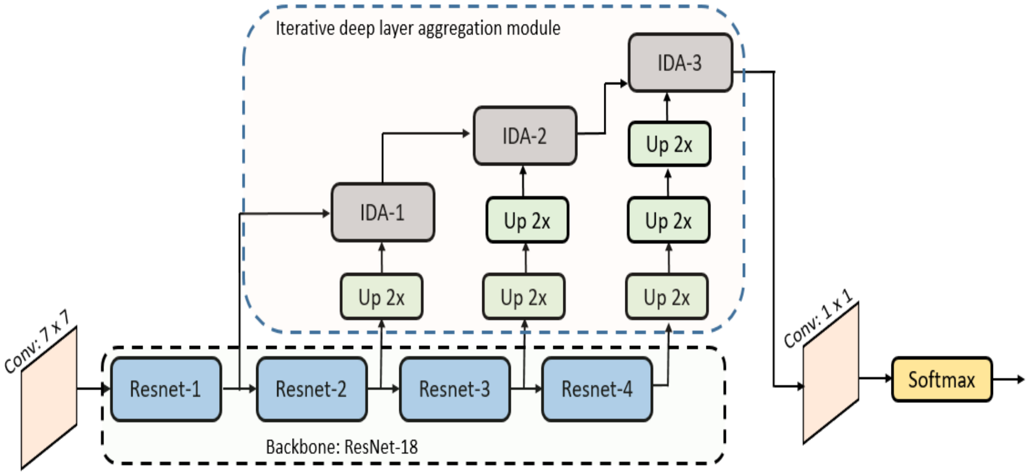

Figure 4. Detailed architecture of Head-Detection network. The input is the image, and the output is the detections, where the number of detections represents the crowd count in the image.

The final feature map is subsequently subjected to a 1 × 1 convolution layer followed by a SoftMax operation to estimate the probability of human heads. Next, a 3 × 3 filter is applied to mitigate noise and detect peaks based on a specified threshold. In this study, we employ a threshold value of 0.5. Any pixel with a value lower than 0.5 is considered noise and is suppressed, while pixels with values greater than 0.5 are set to 1. We then utilize the coordinates of these peaks to derive the location of human heads.

For training the Head-Detection network, we utilize dot-level annotations, where 1 represents the presence of the human head, and 0 represents the background. To supervise the Head-Detection network, we need to generate a ground-truth heatmap. For this purpose, we place a 2D-Gaussian kernel at the location of the head. After generating ground-truth heatmaps from dot-level annotations, we train the Head-Detection network employing the focal cross-entropy loss function formulated in Equation (2).

$$L = -\frac{1}{P_s} \sum_{k \in G} \begin{cases} (1 - \hat{\omega_k})^{\tau} log(\hat{\omega_k}), & if \omega = 1 \\ \Omega(1 - \omega_k)^{\delta} (\hat{\omega}_k)^{\tau} log(1 - \hat{\omega}_k), & \text{otherwise} \end{cases} \tag{2}$$

where P_s is the number of positive samples (heads) in the image G, $\hat{\omega}$ represents the predicted probability of the pixel, and ω is the ground truth, where 1 is for head and 0 for background, τ is the hyper-parameter of focal loss [47] and we set its value to 2 in all experiments as also adopted in [48], δ is also hyper-parameter that controls the penalty of negative samples and we fix its value to 4 in all experiments. Ω is the balancing parameter that balances the positive and negative points, and its value is fixed as $\frac{1}{16}$.

3.4. Crowd-Regression Module

In this section, we discuss the specifics of the proposed crowd-counting model based on regression. As mentioned earlier, in densely populated scenarios, head detection faces challenges in identifying heads due to occlusions. Consequently, in such situations, we leverage regression techniques to estimate the people count. The patches categorized as "HC" will be directed to the Crowd-Regression Module.

Counting crowds in high-density situations poses a challenge due to significant scale variations induced by perspective distortions. To address the scale issue, several methods have proposed using a single CNN with multiple branches [13,49] or employing multiple-column CNNs [12,50] to capture significant variations in object scale. However, these models can cover limited scales due to a fixed number of branches or columns and cannot capture wide-scale variations [51]. Chen et al. [52] introduced Atrous convolutions to capture large capture context without the loss of spatial information. However, these models have wide gaps between the scales due to large dilation rates, which is not suitable for crowded situations where the scales are continuous or have narrow gaps.

To densely sample a wide range of scales, we introduce the Atrous Convolution Grid (ACG). The overall architecture of the proposed Crowd-Regression Module (CRM) is illustrated in Figure 5. Generally, the pipeline of CRM consists of three parts: (1) Backbone network, (2) ACG, and a fusion module that fuses the feature maps of the last layer of the backbone network and feature maps with different dilated rates obtained from the G-ASPP module.

We use VGG-16 as a feature extractor (backbone), which enables the Crowd-Regression Module to extract deep hierarchical features to understand complex crowd dynamics. It is to be noted that we use the first four convolution layers of the VGG-16 and then connect the 3 × 3 Atrous Convolution Grid (ACG) to the fourth convolution layer. The first row of the grid consists of four Atrous Convolution Layers with dilation rates of 1, 2, 4, and 6. The second row of the grid consists of four Atrous Convolution Layers with dilation rates of 5, 7, 9, and 11. The third row of the grid consists of four Atrous Convolution Layers with dilation rates of 8, 10, 11, and 13. From the experiment results, we observe that the choice of Atrous Convolution Layers and their dilation rates plays a critical role in the network's effectiveness. We observe that using a single Atrous Convolution Layer as originally employed in [52] with a dilation rate of 6, 12, 18 is not able to capture the full spectrum of object sizes. We then fuse the feature maps extracted from the fourth convolutional layer of the VGG-16 backbone with the feature maps generated by the Atrous Convolution Grid (ACG), specifically the output from each row of the grid. This fusion is aimed at combining both the high-level hierarchical features learned by VGG-16 and the multi-scale information captured by the ACG. This fusion enables the network to understand complex patterns and details in crowd scenes while accommodating various object sizes and scales. After merging these feature maps, we utilize a convolutional layer with a filter size of 1 × 1, followed by a SoftMax layer to estimate the crowd density.

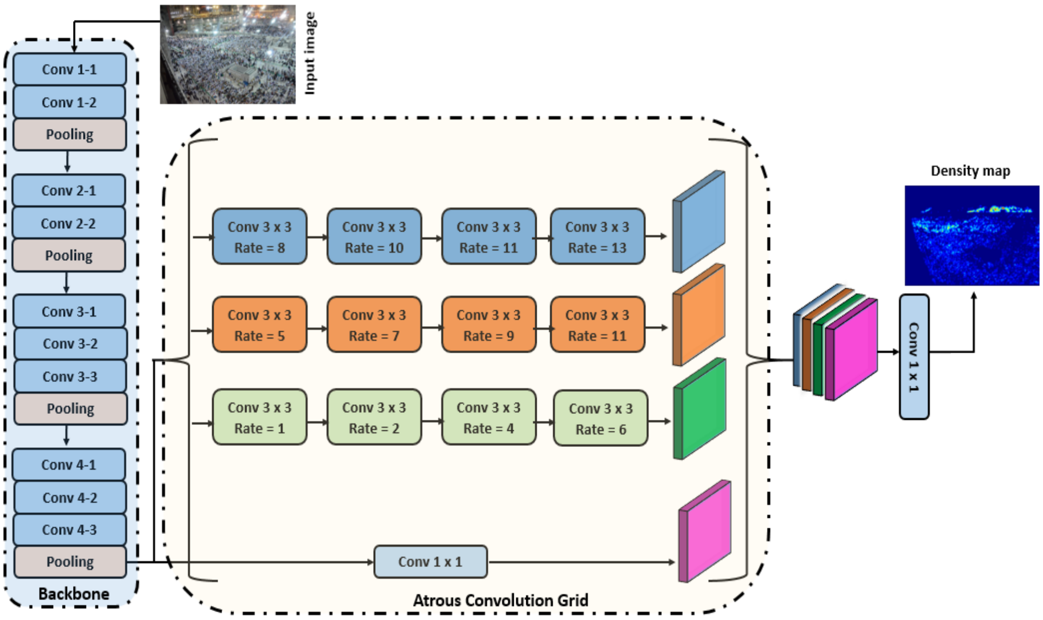

Figure 5. Detailed architecture of Crowd-Regression Module.

We use the Pytorch library [53] for implementing both Head-Detection and crowd-regression modules. To train the models, we employ the Adam optimizer [54], initialized with a learning rate of 3.0×10^{-4} and weight decay set to 1.0×10^{-5}. All models are trained for 60 epochs, using a mini-batch size of 20, with batch normalization updated across the entire mini-batch.

4. Experiment Results

In this section, we evaluate the effectiveness of the proposed framework through both quantitative and qualitative analysis. Additionally, we present a detailed comparative analysis of our framework in relation to other relevant methods. To carry out this assessment and draw comparisons with reference methods, we rely on three publicly available and challenging datasets: UCF_CC_50, UCF_QNRF, and ShanghaiTech. We provide the details of each dataset as follows:

4.1. Datasets

UCF_CC_50 is the first and most widely adopted dataset for crowd-counting tasks and was proposed by Idrees et al. [20]. The dataset contains 50 images, each representing a different real-world scene with varying crowd densities. The dataset covers both low-density and high-density situations, where the number of people ranges from 94 to 4543. Each image in the dataset is accompanied by dot annotations, which represent the position/location of each individual within the image. The dataset covers complex crowd scenes, including outdoor events, public gatherings, and urban environments, with challenging elements like occlusions, scale variations, and perspective distortions, which make the dataset challenging for crowd-counting models.

UCF_QNRF is a significant and challenging dataset for crowd counting and analysis and was proposed by [14]. The dataset contains a total of 1535 images, with varying resolutions, spanning from high resolution to low resolution, with an average image size of 2013 × 2902 pixels. The density within the images also exhibits a significant variation, with the maximum count reaching 12,865, while the minimum count is 65, and the average is 815 per image. The dataset provides 1,251,642 dot annotations, ensuring an abundant resource for training crowd-counting models. In addition to varying resolution and densities, the dataset also covers real-world scenarios, including outdoor events, public gatherings, and urban environments, which offer several complexities, including occlusions, scale variations, and perspective distortions.

ShanghaiTech The ShanghaiTech Part A is also a prominent dataset for evaluating the performance of crowd-counting models. The dataset was proposed by [12] and comprises a total of 482 images with an average resolution of 589 × 868 pixels. The data contains 241,677 dot annotations, with the highest count per image being 3139 individuals, while the average count per image is 501. This reflects that the dataset contains a diverse set of crowd densities.

4.2. Evaluation Metrics

To evaluate the performance of the proposed crowd-counting framework, we use prominent evaluation metrics, Mean Absolute Error (MAE) and Mean Squared Error (MSE). These metrics are commonly used metrics in quantifying the accuracy of crowd-counting algorithms.

Mean Absolute Error (MAE) measures the average absolute difference between the predicted crowd count and the ground-truth count. The lower value of MAE represents a more accurate prediction, while the higher value of MAE represents the error. Since MAE computes the absolute difference between predicted and ground-truth count, it sometimes overestimates or underestimates the predictions. The purpose of using MAE is to know how closely the model's predictions match the actual crowd counts in the images. MAE is formulated as: $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, where n represents the number of samples, y_i is the ground-truth count while $\hat{y}_i$ is the predicted count.

Mean Squared Error (MSE), on the other hand, computes the squared differences between predicted and ground-truth counts. While MSE is a widely used metric, it tends to penalize larger errors more heavily due to the squaring operation. It is essential to consider both MAE and MSE together to gain a comprehensive understanding of the model's performance. MSE highlights the robustness of the models. The lower the value of MSE, the better the performance of the model, while the higher the value of MSE, the lower the performance. MSE is formulated as: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$.

4.3. Performance Evaluation

To gauge the effectiveness of our proposed framework comprehensively, we first evaluate the performance of the Crowd Classifier, which is a critical component that significantly impacts the overall framework performance. For this experiment, we utilize UCF_CC_50 dataset. We train the Crowd Classifier over 60 epochs and monitor the training loss and validation loss trends, as illustrated in the figure. From Figure 6, it is observed that the training loss consistently remains lower than the validation loss. This divergence between the training and validation loss may be attributed to the learning process. During the training process, the model minimizes the training loss by adapting to the training data and optimizing its parameters to fit the specific examples provided during training. However, during the validation stage, the network is evaluated on the unseen data during training.

We present the results of our evaluation using two key performance metrics, which are depicted in Figure 7. From Figure 7, it is observed that our Crowd Classifier achieves an impressive top-1 accuracy of 97%. This implies that, in most cases, the classifier accurately assigns the single most probable crowd class to an image.

Additionally, we present the class-wise performance comparisons of the different classifiers, including AlexNet [55], VGG-16 [56], ResNet-50 [57], and ResNet-152 [57] in Table 1. Precision (P), recall (R), and F1-score (F1) metrics are employed to evaluate the performance of crowd classifiers across all classes in the dataset. From Table 1, it is obvious that ResNet-152 demonstrates superior performance across all classes compared to other methods, achieving high precision, recall, and F1 scores consistently. Based on the performance, ResNet-152 is selected as the preferred model for crowd classification within the proposed framework.

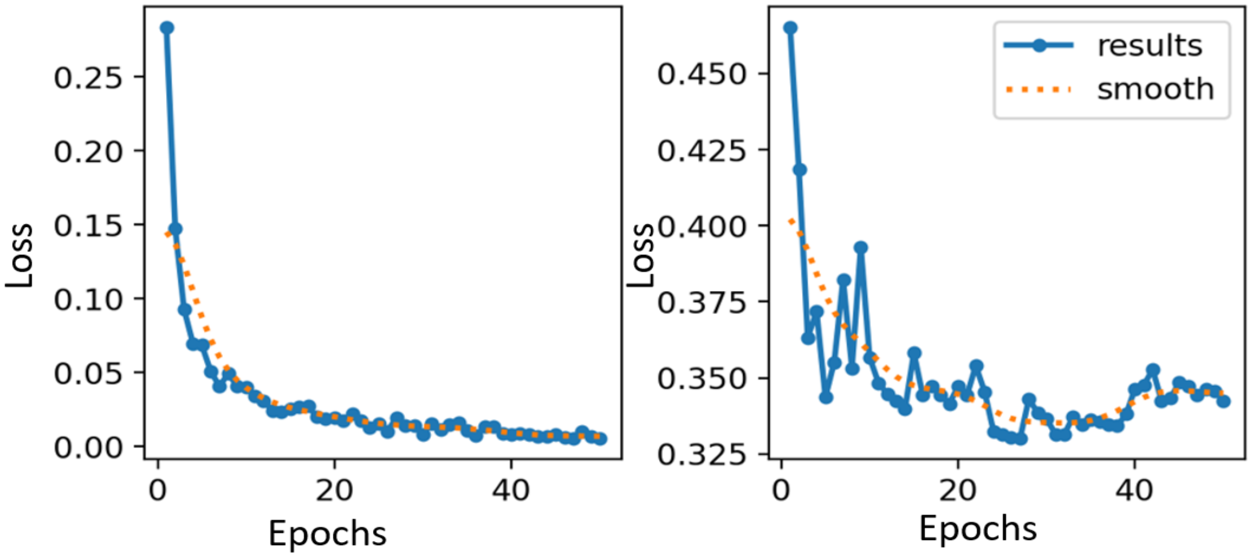

Figure 6. Illustrates training loss and validation loss during training.

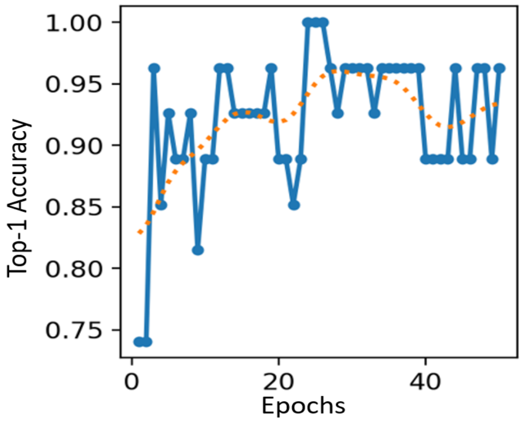

Figure 7. Performance of Crowd Classifier in terms of top-1 accuracy metrics.

Table 1. Class-wise performance comparison of different methods.

Class	**AlexNet** [55]			**VGG-16** [56]			**ResNet-50** [57]			**ResNet-152** [57]		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	F1	**P**	**R**	**F1**
High Crowd	0.92	0.9	0.91	0.95	0.96	0.95	0.97	0.94	0.95	0.98	0.98	0.98
Low Crowd	0.94	0.92	0.93	0.94	0.95	0.94	0.95	0.96	0.95	0.98	0.97	0.98
Medium Crowd	0.92	0.94	0.93	0.96	0.94	0.95	0.96	0.95	0.95	0.99	0.97	0.98
No Crowd	0.92	0.9	0.91	0.95	0.96	0.95	0.96	0.95	0.95	0.98	0.97	0.98

We further investigate the performance of the Crowd Classifier using a confusion matrix, and the results are reported in Figure 8. From Figure 8, it is illustrated that, in most instances, the classifier effectively identifies and categorizes most samples correctly. However, there are cases where the classifier faces challenges and makes misclassifications.

For example, the classifier occasionally misclassifies some samples of 'High Crowd' as 'Medium Crowd'. This misclassification may be attributed to the inherent challenges associated with distinguishing high-density crowds from those with a slightly lower density. Factors such as occlusions, overlapping individuals, or variations in head sizes within the 'High Crowd' class may lead to misclassifications as 'Medium Crowd'. Similarly, the classifier may misclassify 'Low Crowd' samples as 'No Crowd'. This misclassification might occur when the crowd density is very sparse, making it challenging for the classifier to detect the presence of a crowd. In such cases, the lack of obvious crowd patterns or the presence of large empty spaces within the images could lead to misclassifications as 'No Crowd'.

We present the visualization of the output generated by our proposed framework in Figure 9, where we select random samples from the dataset to assess the framework's performance against the corresponding ground truth. Each sampled frame is divided into a 40 × 40 grid, and the counting framework is applied to predict the count within each grid cell. The results demonstrate that the proposed framework produces outputs closely aligned with the ground truth. Notably, certain grid cells exhibit an overestimation of the count, while others show an underestimation. The overestimation may be attributed to instances where the Crowd Classifier module erroneously classifies medium crowds as high-density crowds, leading the router to assign the corresponding patch to the Crowd-Regression Module instead of the Head-Detection Module. Conversely, misclassifications may occur when the Crowd Classifier identifies High Crowd patches as a medium crowd, assigning them to the Head-Detection Module rather than the Crowd-Regression Module. We also note that the framework achieves exceptional performance in accurately classifying "No Crowd" patches, as demonstrated by the precise alignment between the predicted and the corresponding ground truth. This experiment highlights the pivotal role of the crowd-classification module in shaping the overall performance of our proposed crowd-counting framework.

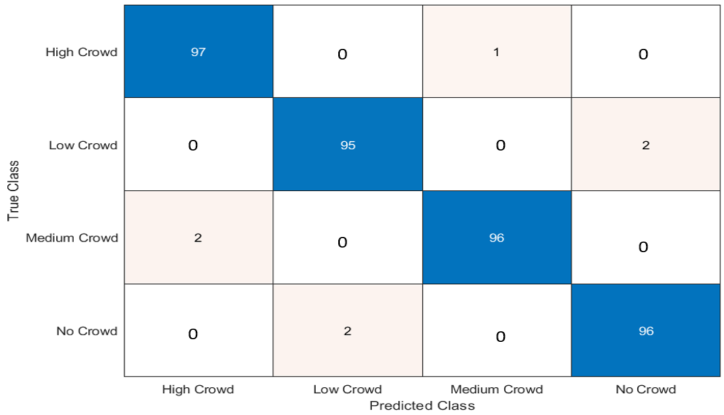

Figure 8. Performance of Crowd Classifier using the Confusion matrix.

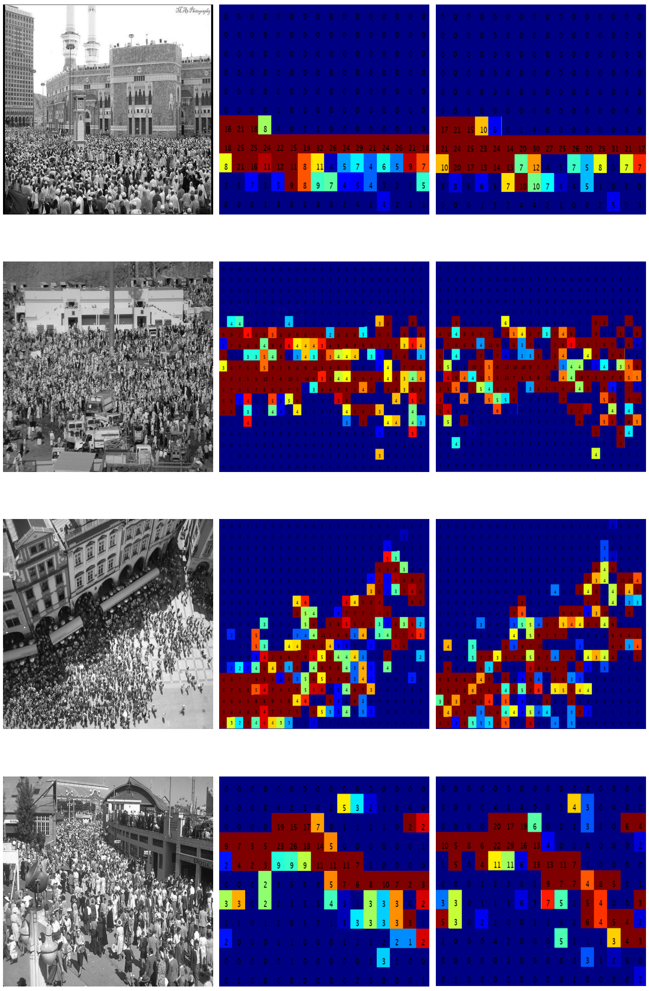

Figure 9. Visualization of random samples from the UCF_CC_50 dataset and their corresponding ground-truth count and predicted count. Each sample is divided into a grid of size 40 × 40. Each number in the cell of the grid represents the people count. The first column shows random samples. The second column represents the corresponding ground-truth count. The third column represents the predicted count.

4.4. Comparisons and Discussion

We compare the proposed framework with other relevant methods. To ensure fair comparisons, we carefully select methods to validate the effectiveness of the proposed framework. These methods include MCNN [12], Idrees et al. [14], CSRNet [23], GauNet (MCNN) [58], URC [24], SRNet [59], Switching CNN [13], DSPNet [60], and Khan et al. [39]. All of these approaches utilize regression techniques, except for the method proposed by Khan et al. [39], which adopts a detection-based approach.

Table 2 displays the performance of various methods in terms of MAE and MSE on the UCF-QNRF dataset.

Table 2. Performance comparisons of different models on the UCF-QNRF dataset.

Method	**MAE**	**MSE**
MCNN [12]	277.0	426.0
Idrees et al. [14]	132.0	191.0
CSRNet [23]	119.2	211.4
GauNet (MCNN) [58]	204.2	280.4
URC [24]	128.1	218.1
SCLNet [61]	109.6	182.5
SRNet [59]	108.2	177.5
Switching CNN [13]	228.0	445.0
DSPNet [60]	107.5	182.7
Khan et al. [39]	112.0	173.0
Proposed	97.20	156.4

From Table 2, it is obvious that the proposed framework demonstrates notable superiority over other related methods. The related methods, including MCNN, GauNet (MCNN), and Switching CNN, achieve higher Mean Absolute Error (MAE) and Mean Squared Error (MSE) values, indicating that these models could not handle diverse crowd densities, occlusions, and perspective variations. Among the competing methods, MCNN and Switching CNN exhibit relatively lower performance compared to other methods. This could be attributed to the fact that MCNN employs multiple CNN columns (three columns) to adapt to variations in people/head size within the scene. However, the reality is that there is a significant amount of variation in the size of human heads, posing a challenge for models with limited scales. As a result, MCNN encounters difficulties in accurately estimating crowd count when confronted with highly dynamic and diverse crowd scenes. Switching CNN utilizes multiple independent CNN regressors with different receptive fields to address the diverse range of scales in human head size. Nevertheless, the network encounters a similar challenge as MCNN, struggling to cover a broad spectrum of scale. Therefore, it faces difficulties in accurately counting people in complex scenes. The limitations in handling the extensive variations in crowd density and scale within diverse scenes may hinder the overall performance of Switching CNN, especially in highly dynamic and challenging crowd scenarios. SRNet [59] achieves comparable performance by producing lower MAE and MSE values. This is because the network introduces a Scale-aware Feature Learning Module (SAM) that captures multi-scale features at different levels, adjusting to various receptive field sizes, which contributes to improved counting accuracy.

The performance of reference crowd-counting methods on the UCF_CC_50 dataset is provided in Table 3. Among the competing models, Idrees et al. [14] exhibit relatively high errors with an MAE of 419.5 and MSE of 541.6. This is because the model could not robustly handle such scale variations and encounters difficulties in accurately counting and localizing individuals in scenes with diverse crowd densities. In contrast, CSRNet [23] and SCLNet [61] show improved performance, achieving an MAE of 266.1 and 258.92, and MSE of 397.5 and 326.24, respectively. However, the proposed framework achieves better performance by exhibiting a lower MAE of 201.6 and MSE of 286.4.

Table 3. Performance comparisons of different models on the UCF_CC_50 dataset.

Method	**MAE**	**MSE**
MCNN [12]	377.6	509.1
Idrees et al. [14]	419.5	541.6
CSRNet [23]	266.1	397.5
GauNet (MCNN) [58]	282.6	387.2
URC [24]	294.0	443.1
SCLNet [61]	258.92	326.24
Switching CNN [13]	318.1	439.2
Cascaded-MTL [62]	322.8	397.9
DSPNet [60]	243.3	307.6
Proposed	201.6	286.4

The performance of different methods on the ShanghaiTech Part A dataset is presented in Table 4. Among the compared methods, DSPNet [60] achieves a relatively low MAE of 68.2 and MSE of 107.8, demonstrating its effectiveness in estimating crowd counts in complex scenes. However, the proposed method outperforms the other models, achieving a notable improvement with an MAE of 57.7 and MSE of 97.5. This suggests that the proposed approach is highly effective in accurately estimating crowd counts in the ShanghaiTech Part A dataset, showcasing its potential for robust performance across diverse crowd scenarios.

Table 4. Performance comparisons of different models on the ShanghaiTech Part A dataset.

Method	**MAE**	**MSE**
MCNN [12]	110.2	173.2
CSRNet [23]	68.2	115.0
GauNet (MCNN) [58]	94.2	141.8
URC [24]	72.8	111.6
SCLNet [61]	67.89	102.94
Switching CNN [13]	90.4	135.0
Cascaded-MTL [62]	101.3	152.4
DSPNet [60]	68.2	107.8
CP-CNN [25]	73.6	106.4
PCC Net [63]	73.5	124
U-ASD Net [64]	64.6	106.1
Proposed	57.7	97.5

It is to be noted that in Tables 2–4, we directly assess the performance of the methods using the dataset we employed. Some entries are missing for certain methods, as they did not evaluate their performance on these datasets. Hence, the corresponding performance values are absent.

From the above Tables 2–4, we observe an interesting finding that the crowd-counting models exhibit notably better performance on the ShanghaiTech Part A dataset compared to their performance on the UCF-QNRF and UCF_CC_50 datasets. This is because the scenes in ShanghaiTech Part A exhibit clearer and more organized crowd structures, allowing models to learn and generalize patterns effectively. Furthermore, the crowd density in ShanghaiTech Part A is relatively less and uniform, making it easier for models to estimate counts accurately.

On the other hand, the UCF-QNRF and UCF_CC_50 datasets present more diverse and challenging crowd scenarios. UCF-QNRF is characterized by a wide range of crowd densities, including both high-density and sparse crowds, which poses a challenge for models to adapt to varying scales and levels of congestion. Similarly, UCF_CC_50 features complex scenes with diverse crowd compositions and occlusions, which pose challenges to the crowd-counting model to precisely estimate the crowd count.

5. Ablation Study

The proposed crowd-counting framework contains two important modules, namely the Head-Detection Module and the Crowd-Regression Module. These modules play a crucial role in accurately estimating crowd counts across diverse scenarios. To understand the impact of these modules and evaluate the effectiveness of various configurations within the Crowd-Regression Module, a series of methods with different settings and configurations are generated. Table 5 presents a comprehensive ablation study of the proposed crowd-counting framework using the UCF-QNRF dataset. Each method represents a distinct configuration, allowing for a systematic exploration of how variations in these configurations influence the overall performance of the crowd-counting framework. We first provide the details of each method as follows:

Table 5. Effect of different configurations on the performance of crowd-counting framework using the UCF-QNRF dataset.

Method	Head Detection	Crowd Regression						MAE	MSE
		ACG-Row-1		ACG-Row-2		ACG-Row-3			
		No. of Layers	Dilation Rate	No. of Layers	Dilation Rate	No. of Layers	Dilation Rate		
M1	Yes	1 × Conv	6	1 × Conv	12	1 × Conv	18	127.03	194.36
M2	Yes	2 × Conv	3,6	2 × Conv	8,12	2 × Conv	12,18	117.54	186.28
M3	Yes	3 × Conv	2,3,4	3 × Conv	5,7,11	3 × Conv	8,12,18	105.72	172.10
M4	Yes	3 × Conv	6,12,18	No				125.20	192.72
M4	No	4 × Conv	1,2,3,4	4 × Conv	5,7,9,11	4 × Conv	8,10,11,13	132.42	195.37
M5	Yes	5 × Conv	2,4,6,8,9	5 × Conv	4,7,8,10,11	5 × Conv	8,12,16,18,20	107.82	178.75
M6	Yes	No						187.23	221.14
M7 (Proposed)	Yes	4 × Conv	1,2,3,4	4 × Conv	5,7,9,11	4 × Conv	8,10,11,13	97.20	156.4

1. **Method M1**: This method comprises the Head-Detection and crowd-regression modules. However, the Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches, each containing one convolutional layer, resulting in a total of three convolutional layers with dilation rates of (6,12,18).
2. **Method M2**: This method comprises Head-Detection and crowd-regression modules. Similar to M1, the Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches. Each branch contains two convolutional layers, resulting in a total of six convolutional layers with dilation rates of (3,6) in the first branch, (8,12) in the second branch, and (12,18) in the third branch.
3. **Method M3**: Similar to previous methods, the M3 method comprises Head-Detection and a Crowd-Regression Module. The Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches. Each branch contains three convolutional layers, resulting in a total of nine (9) convolutional layers with dilation rates of (2,3,4) in the first branch, (5,7,11) in the second branch and (8,12,18) in the third branch.
4. **Method M4**: Similar to previous methods, the M4 method comprises a Head-Detection and Crowd-Regression Module. However, the Atrous Convolution Grid (ACG) of the

Crowd-Regression Module consists of only one branch. The branch contains three convolutional layers with dilation rates of (6,12,18).

5. **Method M5**: The M5 method comprises only a Crowd-Regression Module and does not have a Head-Detection Module. The Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches. Each branch contains four convolutional layers, resulting in a total of twelve (12) convolutional layers with dilation rates of (1,2,3,4) in the first branch, (5,7,9,11) in the second branch and (8,10,11,13) in the third branch.
6. **Method M6**: Similar to previous methods, the M6 method comprises Head-Detection and a Crowd-Regression Module. The Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches. Each branch contains five convolutional layers, resulting in a total of fifteen (15) convolutional layers with dilation rates of (2,4,6,8,9) in the first branch, (4,7,8,10,11) in the second branch and (8,12,16,18,20) in the third branch.
7. **Method M7**: The M7 method is comprised of only head detection and does not have a Crowd-Regression Module.
8. **Method M8**: The M8 method comprises Head-Detection and crowd-regression modules. The Atrous Convolution Grid (ACG) of the Crowd-Regression Module consists of three branches. Each branch contains four convolutional layers, resulting in a total of twelve (12) convolutional layers with dilation rates of (1,2,3,4) in the first branch, (5,7,9,11) in the second branch and (8,10,11,13) in the third branch.

Each method is evaluated based on its Mean Absolute Error (MAE) and Mean Squared Error (MSE) metrics. From Table 5, it is obvious that Method M7 achieves higher MAE and MSE values compared to other methods. This is because M7 does not utilize a crowd-regression model. Without the Crowd-Regression Module, Method M7 solely relies on head detection for crowd counting. As a result, it cannot accurately estimate the crowd count by capturing the spatial distribution and density variations within the crowd.

Method M1 achieves moderate performance due to simpler architecture (limited depth of the ACG) with one convolutional layer in each branch of the Atrous Convolution Grid (ACG). In contrast, M2 improves upon M1 by doubling the depth of the ACG, leading to better feature extraction and, consequently, lower MAE and MSE values. Method M3 further enhances performance by adding a third convolutional layer in each branch, capturing more intricate spatial relationships within the crowd. It is noted that as the number of convolutional layers increases in the ACG, such as in Methods M1, M2, and M3, there is a notable improvement in performance. It is further observed that increasing the depth of the ACG may initially lead to improved feature extraction and representation; however, beyond a certain point, additional layers may introduce redundancy or overfitting, resulting in a degradation of performance as illustrated by Method M5.

This experiment shows that even deeper architectures could help capture more complex features within the crowd; the reduced improvements seen in Method M5 necessitate the need to find the right balance between the complexity of the model and its performance.

The proposed method, M8, achieves the lowest Mean Absolute Error (MAE) and Mean Squared Error (MSE) values compared to other methods in Table 5. Despite incorporating a crowd-regression model with four convolutional layers in each branch of the Atrous Convolution Grid (ACG), M8 maintains a balance between complexity and performance. From the experiments, we observe that by carefully selecting the number of layers and their dilation rates in ACG, M8 effectively captures the complex spatial relationships within the crowd while avoiding excessive model complexity.

6. Computational Complexity

To investigate the effectiveness of the proposed framework further, we compared it with other methods in terms of computational complexity. The computational complexity of different methods, including the proposed one on the ShanghaiTech dataset, is reported in Table 6. The computational complexity during the inference time is denoted in milliseconds (ms) and frames per second (fps). From Table 6, it is evident that PCC Net and U-ASD Net achieve relatively faster inference times, with 89 milliseconds and 94 milliseconds, respectively. Among the evaluated methods, CP-CNN is relatively slower, achieving the longest inference time. While the proposed framework's inference time of 146 milliseconds lags behind PCC Net and U-ASD Net, it still outperforms the compared methods. Although it does not surpass all methods in computational efficiency, the proposed framework outperforms them in terms of Mean Absolute Error (MAE) and Mean Squared Error (MSE). This suggests that the proposed framework strikes a balance between computational complexity and performance, making it a promising solution for crowd-counting tasks.

Table 6. Comparisons of different crowd-counting methods in terms of computational complexity.

Method	Inference Time (Milliseconds)	Frames per Second	MAE	MSE
Switching CNN [13]	153	6.54	90.4	135
CSRNet [23]	330	3.0	68.2	115.0
CP-CNN [25]	5113	0.195	73.6	106.4
PCC Net [63]	89	11.24	73.5	124.0
U-ASD Net [64]	94	10.63	64.6	106.1
Proposed	146	6.84	57.7	97.5

7. Conclusions

In this work, we proposed a framework that effectively leverages the strengths of both regression and detection models for estimating crowd counts in diverse scenes. The performance of the framework is evaluated on challenging datasets. From the experiment results, we draw the following conclusions:

1. The proposed framework demonstrates superior performance across all datasets, demonstrating its effectiveness and versatility in addressing the challenges posed by various complex scenes.
2. The proposed framework employs a unique way of handling the scale problem in crowd counting by adopting a routing strategy that directs image patches to one of two counting modules based on their density levels. In this way, based on the complexity of the crowd, the network can effectively handle the scale problem and achieve high performance across all datasets.

For future work, we will refine the proposed framework to enhance its adaptability to even more diverse and complex scenes. Additionally, we will focus on investigating ways to extend the framework's scalability and efficiency for real-time crowd-counting applications, potentially through optimization techniques or architectural enhancements.

Author Contributions: Conceptualization, S.D.K. and F.U.R.; methodology, S.D.K.; software, S.D.K.; validation, F.U.R., A.N.A. and S.D.K.; formal analysis, S.D.K.; investigation, S.D.K.; resources, A.N.A.; data curation, A.N.A.; writing—original draft preparation, S.D.K.; writing—review and editing, A.N.A.; visualization, F.U.R.; supervision, A.N.A.; project administration, A.N.A.; funding acquisition, A.N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Custodian of the Two Holy Mosques Institute for Hajj and Umrah Research, Umm Al-Qura, Makkah, Saudi Arabia under project No. 23/113.

Data Availability Statement: The data used in this research is publicly available.

Acknowledgments: The researchers extend their sincere thanks to the Custodian of the Two Holy Mosques Institute for Hajj and Umrah Research for supporting and financing this project No. 23/113, which significantly contributed to the completion of the project phases.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, S.D.; Tayyab, M.; Amin, M.K.; Nour, A.; Basalamah, A.; Basalamah, S.; Khan, S.A. Towards a crowd analytic framework for crowd management in Majid-al-Haram. *arXiv* **2017**, arXiv:1709.05952.
2. Gayathri, H.; Aparna, P.; Verma, A. A review of studies on understanding crowd dynamics in the context of crowd safety in mass religious gatherings. *Int. J. Disaster Risk Reduct.* **2017**, *25*, 82–91. [CrossRef]
3. Khan, M.A.; Menouar, H.; Hamila, R. Revisiting crowd counting: State-of-the-art, trends, and future perspectives. *Image Vis. Comput.* **2023**, *129*, 104597. [CrossRef]
4. Wang, M.; Cai, H.; Dai, Y.; Gong, M. Dynamic Mixture of Counter Network for Location-Agnostic Crowd Counting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 167–177.
5. Basalamah, S.; Khan, S.D.; Felemban, E.; Naseer, A.; Rehman, F.U. Deep learning framework for congestion detection at public places via learning from synthetic data. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 102–114. [CrossRef]
6. Stadler, D.; Beyerer, J. Modelling ambiguous assignments for multi-person tracking in crowds. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 133–142.
7. Li, Y. A deep spatiotemporal perspective for understanding crowd behavior. *IEEE Trans. Multimed.* **2018**, *20*, 3289–3297. [CrossRef]
8. Grant, J.M.; Flynn, P.J. Crowd scene understanding from video: A survey. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2017**, *13*, 1–23. [CrossRef]
9. Khan, S.D.; Bandini, S.; Basalamah, S.; Vizzari, G. Analyzing crowd behavior in naturalistic conditions: Identifying sources and sinks and characterizing main flows. *Neurocomputing* **2016**, *177*, 543–563. [CrossRef]
10. Gao, G.; Gao, J.; Liu, Q.; Wang, Q.; Wang, Y. Cnn-based density estimation and crowd counting: A survey. *arXiv* **2020**, arXiv:2003.12783.
11. Zhang, C.; Li, H.; Wang, X.; Yang, X. Cross-scene crowd counting via deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 833–841.
12. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 589–597.
13. Babu Sam, D.; Surya, S.; Venkatesh Babu, R. Switching convolutional neural network for crowd counting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5744–5752.
14. Idrees, H.; Tayyab, M.; Athrey, K.; Zhang, D.; Al-Maadeed, S.; Rajpoot, N.; Shah, M. Composition loss for counting, density map estimation and localization in dense crowds. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 532–546.
15. Sam, D.B.; Peri, S.V.; Sundararaman, M.N.; Kamath, A.; Babu, R.V. Locate, size and count: Accurately resolving people in dense crowds via detection. *arXiv* **2019**, arXiv:1906.07538.
16. Basalamah, S.; Khan, S.D.; Ullah, H. Scale driven convolutional neural network model for people counting and localization in crowd scenes. *IEEE Access* **2019**, *7*, 71576–71584. [CrossRef]
17. Wang, Y.; Lian, H.; Chen, P.; Lu, Z. Counting people with support vector regression. In Proceedings of the 2014 10th International Conference on Natural Computation (ICNC), Xiamen, China, 19–21 August 2014; pp. 139–143.
18. Chan, A.B.; Liang, Z.S.J.; Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7.
19. Pham, V.Q.; Kozakaya, T.; Yamaguchi, O.; Okada, R. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3253–3261.
20. Idrees, H.; Saleemi, I.; Seibert, C.; Shah, M. Multi-source multi-scale counting in extremely dense crowd images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2547–2554.
21. Wan, J.; Chan, A. Adaptive density map generation for crowd counting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1130–1139.
22. Dong, L.; Zhang, H.; Ji, Y.; Ding, Y. Crowd counting by using multi-level density-based spatial information: A Multi-scale CNN framework. *Inf. Sci.* **2020**, *528*, 79–91. [CrossRef]
23. Li, Y.; Zhang, X.; Chen, D. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1091–1100.
24. Xu, Y.; Zhong, Z.; Lian, D.; Li, J.; Li, Z.; Xu, X.; Gao, S. Crowd counting with partial annotations in an image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15570–15579.
25. Sindagi, V.A.; Patel, V.M. Generating high-quality crowd density maps using contextual pyramid cnns. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1861–1870.
26. Zhai, W.; Gao, M.; Souri, A.; Li, Q.; Guo, X.; Shang, J.; Zou, G. An attentive hierarchy ConvNet for crowd counting in smart city. *Clust. Comput.* **2023**, *26*, 1099–1111. [CrossRef]

27. Zhang, J.; Ye, L.; Wu, J.; Sun, D.; Wu, C. A Fusion-Based Dense Crowd Counting Method for Multi-Imaging Systems. *Int. J. Intell. Syst.* **2023**, *2023*, 6677622. [CrossRef]
28. Zhai, W.; Gao, M.; Li, Q.; Jeon, G.; Anisetti, M. FPANet: Feature pyramid attention network for crowd counting. *Appl. Intell.* **2023**, *53*, 19199–19216. [CrossRef]
29. Guo, X.; Song, K.; Gao, M.; Zhai, W.; Li, Q.; Jeon, G. Crowd counting in smart city via lightweight ghost attention pyramid network. *Future Gener. Comput. Syst.* **2023**, *147*, 328–338. [CrossRef]
30. Gao, M.; Souri, A.; Zaker, M.; Zhai, W.; Guo, X.; Li, Q. A comprehensive analysis for crowd counting methodologies and algorithms in Internet of Things. *Clust. Comput.* **2024**, *27*, 859–873. [CrossRef]
31. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
32. Ren, X. Finding people in archive films through tracking. In Proceedings of the Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
33. Yan, J.; Lei, Z.; Wen, L.; Li, S.Z. The fastest deformable part model for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2497–2504.
34. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
35. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. From facial parts responses to face detection: A deep learning approach. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3676–3684.
36. Zhang, K.; Zhang, Z.; Wang, H.; Li, Z.; Qiao, Y.; Liu, W. Detecting faces using inside cascaded contextual cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3171–3179.
37. Zhu, C.; Zheng, Y.; Luu, K.; Savvides, M. Cms-rcnn: Contextual multi-scale region-based cnn for unconstrained face detection. In *Deep Learning for Biometrics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 57–79.
38. Hu, P.; Ramanan, D. Finding tiny faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 951–959.
39. Khan, S.D.; Basalamah, S. Scale and density invariant head detection deep model for crowd counting in pedestrian crowds. *Vis. Comput.* **2021**, *37*, 2127–2137. [CrossRef]
40. Shami, M.B.; Maqbool, S.; Sajid, H.; Ayaz, Y.; Cheung, S.C.S. People counting in dense crowd images using sparse head detections. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2627–2636. [CrossRef]
41. Lian, D.; Chen, X.; Li, J.; Luo, W.; Gao, S. Locating and counting heads in crowds with a depth prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 9056–9072. [CrossRef]
42. Zhou, T.; Yang, J.; Loza, A.; Bhaskar, H.; Al-Mualla, M. Crowd modeling framework using fast head detection and shape-aware matching. *J. Electron. Imaging* **2015**, *24*, 023019. [CrossRef]
43. Saqib, M.; Khan, S.D.; Sharma, N.; Blumenstein, M. Crowd counting in low-resolution crowded scenes using region-based deep convolutional neural networks. *IEEE Access* **2019**, *7*, 35317–35329. [CrossRef]
44. Arandjelovic, O. Crowd detection from still images 2008. In Proceedings of the British Machine Vision Conference, Leeds, UK, September 2008.
45. Sirmacek, B.; Reinartz, P. Automatic crowd analysis from airborne images. In Proceedings of the 5th International Conference on Recent Advances in Space Technologies-RAST2011, Istanbul, Turkey, 9–11 June 2011; pp. 116–120.
46. Saqib, M.; Khan, S.D.; Blumenstein, M. Texture-based feature mining for crowd density estimation: A study. In Proceedings of the 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, New Zealand, 21–22 November 2016; pp. 1–6.
47. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
48. Wang, Y.; Hou, J.; Hou, X.; Chau, L.P. A self-training approach for point-supervised object detection and counting in crowds. *IEEE Trans. Image Process.* **2021**, *30*, 2876–2887. [CrossRef]
49. Wang, Y.; Zhang, W.; Liu, Y.; Zhu, J. Two-branch fusion network with attention map for crowd counting. *Neurocomputing* **2020**, *411*, 1–8. [CrossRef]
50. Yang, Y.; Li, G.; Du, D.; Huang, Q.; Sebe, N. Embedding perspective analysis into multi-column convolutional neural network for crowd counting. *IEEE Trans. Image Process.* **2020**, *30*, 1395–1407. [CrossRef]
51. Dai, F.; Liu, H.; Ma, Y.; Zhang, X.; Zhao, Q. Dense scale network for crowd counting. In Proceedings of the 2021 International Conference on Multimedia Retrieval, Taipei, Taiwan, 21–24 August 2021; pp. 64–72.
52. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]
53. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 18 March 2024).
54. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
55. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. [CrossRef]
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
58. Cheng, Z.Q.; Dai, Q.; Li, H.; Song, J.; Wu, X.; Hauptmann, A.G. Rethinking spatial invariance of convolutional networks for object counting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–27 June 2022; pp. 19638–19648.
59. Huang, L.; Zhu, L.; Shen, S.; Zhang, Q.; Zhang, J. SRNet: Scale-aware representation learning network for dense crowd counting. *IEEE Access* **2021**, *9*, 136032–136044. [CrossRef]
60. Zeng, X.; Wu, Y.; Hu, S.; Wang, R.; Ye, Y. DSPNet: Deep scale purifier network for dense crowd counting. *Expert Syst. Appl.* **2020**, *141*, 112977. [CrossRef]
61. Wang, S.; Lu, Y.; Zhou, T.; Di, H.; Lu, L.; Zhang, L. SCLNet: Spatial context learning network for congested crowd counting. *Neurocomputing* **2020**, *404*, 227–239. [CrossRef]
62. Sindagi, V.A.; Patel, V.M. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
63. Gao, J.; Wang, Q.; Li, X. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 3486–3498. [CrossRef]
64. Hafeezallah, A.; Al-Dhamari, A.; Abu-Bakar, S.A.R. U-ASD net: Supervised crowd counting based on semantic segmentation and adaptive scenario discovery. *IEEE Access* **2021**, *9*, 127444–127459. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.