Journal of Information Systems Engineering and Management
2025, 10(38s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Drone Based Crowd Density Estimation and Localization Using Temporal and Location Sensitive Fused Attention Model on Pyramid Features

*J.Evangelin Deva Sheela, Dr. P. Arockia Jansi Rani, Dr. M. Asha Paul

Research Scholar, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Abhisekapatti, Tirunelveli, Tamilnadu, India.

Mail id: sheela8mca@gmail.com

Associate Professor, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Abhisekapatti, Tirunelveli, India.

Mail id: jansimsuniv@gmail.com

Assistant Professor, Department of Computer Science and Engineering, Francis Xavier Engineering College, Vannarpettai, Tirunelveli, Tamilnadu, India.

Mail id: ashanichelson@gmail.com

*Corresponding Author

ARTICLE INFO

Received: 21 Dec 2024

Revised: 17 Feb 2025

Accepted: 24 Feb 2025

ABSTRACT

Crowd monitoring is essential for security and effective management in public space, and drone imagery offers a powerful tool for this purpose. Though traditional methods often fall short in accuracy and efficiency techniques like manual counting, detection based approaches struggle with challenges like occlusion, low resolution, and high crowd density, leading to unreliable estimates. To address data privacy concerns related to capturing images of individuals without consent, regulatory barriers that restrict flight zones and operational guidelines, and technical limitations such as limited battery life and communication range like limitations, this study introduce a novel approach called the Temporal and Location Sensitive Fused Attention Model on Pyramid Features (TLFA_PF) for crowd density estimation and localization. The method employs scales while minimizing computational complexity. By integrating spatial and temporal attention schemes, the model effectively captures significant information from drone capturing images. A key innovation of this work is the introduction of a Bi Pooling Squeeze and Excitation Block, which enhances the conventional neural network by incorporating two pooling networks. This block selectively emphasizing important features improving the models ability to discern crowd density variation. The TLFA_PF model demonstrates superior performance in estimating crowd density and localizing individual compared to existing methods experimental results highlights the effectiveness of TLFA_PF across various scenario, showcasing its robustness in handling different crowd densities within the fused attention framework allows for more accurate predictions, making it’s a significant advancement in drone based crowd analysis. Overall, this research contributes to the field of computer vision by providing an efficient and effective solution for real time crowd monitoring using aerial imagery.

Keywords: Drone, Spatial, Temporal, Fused Attention models, Feature Extraction, computer vision, localized map, Crowd density

INTRODUCTION

As urbanization accelerates due to population growth, an increasing number of individuals are residing in urban areas, leading to both positive and negative consequences. On the positive side, urban living enriches cultural experiences and maximized the use of urban infrastructure. However, the concentration of people in cities also present significant challenges for urban security and management, especially during large gatherings for events such as political demonstrations or festivals. This has led to a growing interest in automated crowd analysis methods, particularly in crowd counting and density estimation, to enhance safety and management strategies [1, 2]. The drone based crowd tracking has emerged as a vital technology that utilizes Unmanned Aerial Vehicle (UAVs) equipped with cameras for automated surveillance. This technology is essential for identifying and consistently tracking individuals across multiple video frames, even amidst the dynamic movements of both the crowd and the environment, employing Multi Objective Tracking (MOT) techniques to achieve effective monitoring [3]. The crowd gathering may be religious, sports, cultural, or any other public events where, video surveillance are used for crowd control and public safety. Therefore, UAVs used widely and became popular in monitoring mass gathering nowadays, for monitoring the crowd, to protect property, maintaining peace, save human lives and preserve the environment [4]. Crowd density estimation is a critical area of research in computer vision, with applications ranging from urban planning to public safety. Traditional techniques for crowd analysis often rely on stationary cameras, which can limit the potential of drones in overcoming these challenges by providing aerial view that capture dynamic crowd behavior more effectively [5]. For instance, the introduction of Space Time Neighbor Aware Network (STNNet) tailored for drone based crowd analysis, demonstrating significant improvement in density map estimation, localization, and tracking capabilities compared to conventional methods [6, 7].

Traditional methods for drone based crowd density estimation and localization have generally relied on detection based method typically involved identifying individuals in video frames using algorithms that classify and localize objects however, these techniques often struggles with the challenges posed, leading to difficulties in accurate detection and tracking. For instance, the study [8] highlights how aerial views complicates the detection of individuals due to their sizes and proximity, which can result in missed detection or false positive in crowed environments. Regression based method [9] focus on estimating crowd density directly from image features but may fail to account for variations in crowd dynamics and environmental conditions, resulting in inaccuracies. Similarly, density based methods have improved upon some of these limitations by predicting density map that represent the number of individuals, which is critical for effective crowd management . For example, while frameworks like the Space Time Multi Scale Attention Network (STNNet) have shown promise in aggregating multi scale feature to enhance density map perditions, they still face challenges related to object displacement and occlusion due to high crowd density [10]. Moreover, traditional approaches often depends heavily on consecutive frames from tracking which can be problematic in scenarios characterized by significant movements or large inter frame intervals. This dependence is particularly evident in studies that emphasize the need for robust tracking mechanisms capable of handling rapid changes in crowd dynamics [11].

Additionally, many existing dataset used for training these models are primarily designed for static camera environments, limiting their applicability to dynamic aerial setting. As a result, traditional methods frequently encounter issues with accuracy and reliability in real world applications, necessitating the development of more sophisticated techniques that integrations temporal and location-sensitive data to enhance tracking and localization capabilities in drone based crowd monitoring [12, 13]. Recent advancements in drone based crowd density estimation and localization have increasingly incorporated Artificial Intelligence (AI), Machine learning (ML) and Deep Learning (DL) techniques. These techniques are pivotal in enhancing urban management strategies, particularly for monitoring large crowd during events. The study [14, 15], involve the use of deep learning architectures, particularly Conventional Neural Network (CNN), which have shown effective in processing aerial imagery for crowd counting and density estimation. For example, a Deep Neural

Network (DNN) model specifically designed for drone assisted systems, demonstrating improved accuracy in estimating crowd size from drone capturing image. Additionally, attention mechanism have been integrated into these models to focus on relevant features within crowded scenes, future enhancing estimation accuracy [16]. ML also plays a crucial role to improve robust and adaptability to various dataset. Similarly DL are used to improve the accuracy in crowd analysis, although the reliance on specific dataset limits the generalizability of these models across different scenario. Further data augmentation techniques such as color jittering and affine transformations, have been utilized to address variations in lighting and environmental conditions, thus enhancing model performance [17]. Despite these advancements, many existing models struggles with the complexities associated with varying crowd density and perspectives. A significant issue is the limited variety of dataset used for training models, which often rely on similar video sources [18, 19] . This lack of diversity restrict the models ability to adapt to new patterns of crowd behavior. A proper crowd control system and surveillance is highly needed for avoiding risk event situations. For this, real-time images should be utilized to extract information and instructions should be able to be communicated to the crowd at the same time which will be a very crucial tool in smart systems [19, 20].

To overcome such issues the proposed work, introduces a novel approach namely Temporal and Location Sensitive Fused Attention Model on Pyramid Feature (TLFA PF) for crowd estimation and localization. TLFA PF is a multiple scales while minimizing computational complexity through a feature pyramid enhancement model. By incorporating spatial and temporal attention mechanism alongside a Squeeze excitation module with dual pooling network, TLFA PF enhances its ability to focus on prominent information within dense crowds. Experimental results indicates that TLFA PF significantly outperforms many existing method, demonstrating its potential to bridge gaps in precision and reliability that have hindered previous approaches in crowd monitoring and localization. The major contribution of the respective model are signifies in the following:

- The model enhance the urban management strategies by providing accurate crowd estimation and localization, which are critical for effective crowd control during large gathering.
- To employ TLFA PA extract feature at multiple scales while minimizing computational complexity through feature pyramid enhancement model, where the efficiency in processing aerial imagery.

1.1 Objective

The main objective of proposed model, to implement a feature pyramid enhancement model that effectively extracts feature at multiple scales, improving the models ability to capture relevant information from a complex crowd. And to incorporate spatial and temporal attention mechanism that enhances the models focus on significant data point within density crowds thereby improving the accuracy of crowd estimation and localization. To employ a Bi-Pooling Squeeze and Excitation Block that refine feature extraction by emphasizing important feature while reducing noise enhancing the overall classification performance. To validate the efficacy of the proposed model through experimental results that demonstrate its superiority over existing methods in terms of accuracy and reliability in crowd monitoring and localization tasks.

1.2 Paper Organization

The paper is organized based on the effectual approaches applied in the drone based crowd estimation and localization and whereas section 2 explains the related work and identifies key problem in the field. Section 3 details the proposed methodology, including feature extraction techniques and attention mechanism. The result and discussion, include dataset description and performance analysis, are presented in section 4. Finally section 5 conclude the paper and suggest future direction.

RELATED WORK

This section deliberate the analysis of the conventional research in the drone based crowd estimation and localization with temporal and spatial location sensitive fused attention model on pyramid features and other techniques and existing methods on crowd density estimation.

Crowd management is crucial for ensuring safety at events yet effective control remains challenging despite advancements in drone and surveillance technology. The prevailing study [21], introduced an approach for crowd counting using drone data, employed dilated and scaled neural network for feature extraction and density estimation, existing method trained on a new dataset, ViseDrone2020, and compared against ten state of art methods, demonstrating superior accuracy in crowd counting and showed high performance on non-drone dataset like UCF-QNRF and ShanghaiTech, and the model effectively handled a noise, and attained better density and with Gaussian and Salt and pepper noise at a density of 0.02. Similarly, the existing study [22, 23], addresses the critical challenges of crowd density estimation for application like autonomous driving and crowd control, particularly in dynamic scenes with varying object size. The study used parallel multi size receptive field units to leverage features from multiple CNN layers enhancing the models ability to handle different scales, which incorporated asymmetric non local attention and channel weighting to improve prediction accuracy. Experiment done on UFC-CC 50 and ShanghaiTech dataset demonstrate significant improvements in density estimation, effectively managing dense distributions and varying object size. The conventional crowd counting method relied on costly pixel level annotations by the study [24], which used Deep Rank consistent PyrAmid model (DREAM) that utilized rank consistency within latency feature spaces, it enhanced model representation by leveraging pyramid partial orders across coarse-to-fine feature, allowing for effective use of unlabeled images. A new unlabeled dataset, FUDAN-UCC, was collected, comprising 4000 images for training. Investigates on scale datasets namely UCF-QNRF and ShanghaiTech established important enhancements in crowd counting accuracy.

Correspondingly, the study [25], light weighted on board crowd pattern identification method using H.264 video compression standard, achieved real time recognition in as low as 2 milliseconds on NVIDIA TX2, with a 45* execution time reduction compared to previous methods, the technique featured a temporally aware system that adapted to changing crowd movement patterns as the drone's point of view varied. Evaluations against public dataset demonstrated significant performance and computational advantages, enhancing drone centric crowd management solutions. Similar study [19, 26] focused on leveraging drone technology for crowd detection during COVID-19 pandemic recognizing the importance of effective crowd management in mitigating virus spread. The novelty of the research lay in its application of microguadroctor drones equipped with camera for real time air surveillance and crowd pattern identification. The method involved processing aerial images transmitted to mobile applications for enhanced data analysis and monitoring, successfully collecting data. The space time multi scale attention network for joint density estimation and localization. The study [27], method emphasize attention layers to capture temporal spatial crowd information from drone perspectives, using the Drone Crowd dataset for validation, showing high accuracy in dense crowds. Similarly the prevailing study [28], discusses advancements in drone based crowd counting and localization, emphasizing the need for temporal spatial attention models. Existing dataset has been utilized, it evaluate real time crowd localization approaches, highlighting improvement in crowd detection efficiency under varied conditions. The paper [29], introduced a large scale dataset of crowd tracking and counting. The result utilized temporal and spatial features to address crowd estimation challenges and the benchmark includes results on detection accuracy.

The Drone Net introduced self-organizing neural network to improve crowd density estimation, incorporated pyramid feature to address scale variations. Tested on the drone crowd dataset, the model demonstrate efficiency in handling high density scenes [30]. Similarly the use of attention mechanism to improve feature extraction in drone images, applying it to drone collected dataset. ARCN [31, 32], achieves a Mean Absolute Error (MAE) of 19.9 and a Mean Squared Error (MSE) of 27.7 on the Drone Crowd dataset, processing at 48 FPS on an NVIDIA GTX 2080 Ti GPU, marking it as a novel real-time solution. This study [33], presents JMFEEL-Net, a novel approach that enhances crowd counting accuracy by integrating joint multi-scale feature enhancement with a lightweight transformer. The method employs a high resolution CNN supports and multi-scale feature enhancement module, validated on datasets including ShanghaiTech Part A/B, JHU-Crowd++, and UCF-QNRF, achieving competitive counting performance across these challenging datasets. The existing study [34, 35], addresses the challenges of crowd counting from drone-captured data, including small object inference and background clutter, by collecting a large-scale dataset of 3,360 images for the vision meets Drone Crowd Counting Challenge in VisDrone-CC2020 at ECCV 2020. The dataset includes 2,460 training images and 900 testing images, all manually annotated to support advancements in this field.

1.3 Problem Identification

Several conventional research has been limited by crow detection viewpoint and accuracy in crowd density estimation.

- The crowd counting and density estimation in a static setting, neglecting the challenges posed by dynamic environments where crowd behaviour can change rapidly [25].
- The significant computational resources, making them unsuitable for real-time applications, particularly on drones with limited processing capabilities are limited [33].
- Although attention mechanisms have shown promise in other areas of computer vision, their full potential in crowd monitoring applications using drones has not been thoroughly explored [28].

PROPOSED METHODOLOGY

The crowd management has emerged as a critical challenge. Traditional methods of crowd monitoring often struggle with accuracy, and real time processing, particularly in dynamic setting where crowd density can fluctuate rapidly. To overcome this challenges, the proposed work used novelty to improve precision.

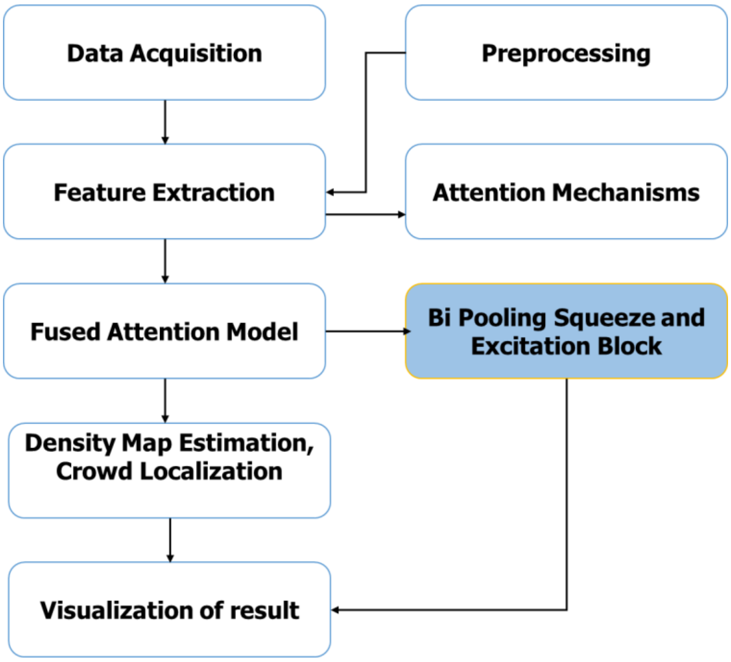


Figure 1. Flow of proposed model

The figure 1 depicts the structured workflow for the data processing starting with data acquisition, where the data is collected from drone crowd dataset. This data is then subjected to pre-processing to clean and organize for quality. Next feature extraction identifies and generate relevant features for potential pyramid feature with attention mechanism to emphasizing critical elements. Following this the fused attention model combined with multiple attention strategies to improve feature extraction, the Bi Pooling Squeeze and Excitation block novel step refine the feature map by enhancing significant features and suppressing less important one. The density localization and crowd counting is also done with density map estimation. The following metrics helps to improvise the visualization result.

1.4 Feature Pyramid Enhancement for Feature Extraction

The study use Feature pyramid Network (FPNs) to extract image features at different scales. However the processing the same image at multiple scales, occlusion, variation and viewpoints can lead to increase computational complexity. To mitigate this issues, the FPEM was introduced, maintaining the advantages of FPNs while reducing complexity. The FPEM operates in two phase: the scale enhancement phase and the down scale enhancement phase. During the up sampling process, input features are mapped using strides of 4, 8, 16, and 32 pixels are applied. The method further enhances accuracy through element- wise multiplication of the input with the down samples output, effectively depending the network and expanding the receptive fields.

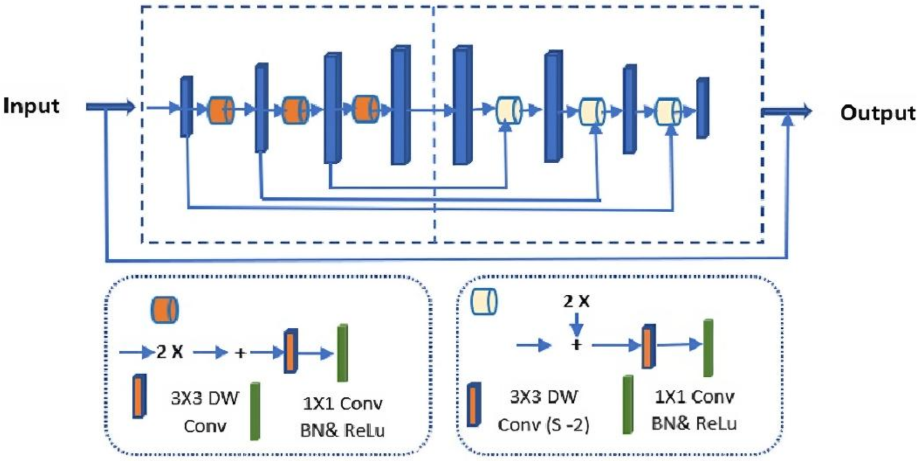


Figure 2. Feature Pyramid Enhancement Module (FPEM)

The figure 2. Illustrate the architecture of a neural network, highlighting its main flow and two specific block configurations. The network processes input on the left and produces output on the right utilizing residual connections to enhance information retention and improving gradient flow. The enlarged block details shows a left block featuring a 3*# depth wise conventional followed by a 1*1 convention with batch normalization and ReLU activation.

1.5 Spatial and Temporal Attention Schemes

Attention Model

Attention model are employed in various application to highlight significant features. By utilizing these models, distinct features can be extracted, which can significantly benefit the applications. There are two main types of attention models namely, channel attention models and spatial attention models.

Channel attention model, in this mechanism the prominent feature are extracted by leveraging the inter channel relationship of features. The accompanying figures outline the steps involved into channel attention. This approach spatial examine that statistics of pixel information. By integrating two pooling methods max pooling and average pooling, the feature representation is fine tuned in the model.

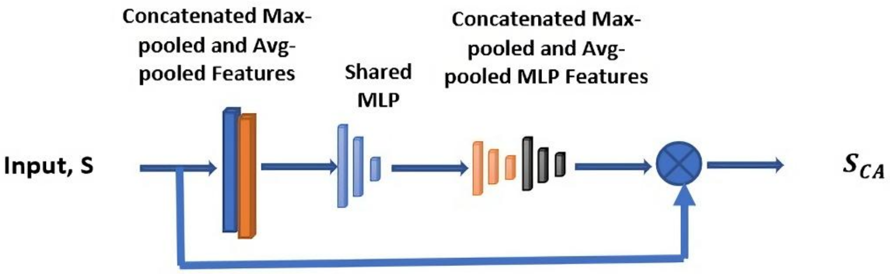

Concatenated Max-pooled and Avg-pooled Features
Shared MLP
Concatenated Max-pooled and Avg-pooled MLP Features
Input, S
S_{CA}

Figure 3. Channel Attention Model

Figure 3 illustrates the channel attention model. The output of the channel attention, S_{CA}can be calculated feeding the average and max pooled features into a shared multi layered perceptron (MLP) whose output is again applied with pooling layers. The input is multiplies with this output to get theS_{CA}.

Spatial Attention Model

The spatial attention model examines the spatial relationship among pixels. The accompanying figure depicts the steps involved in the spatial attention process. Average pooling and max pooling are performed on the input feature S, and their concatenated outputs are passed through a conventional layer. These pooling operations emphasizing the information's present at particular locations. The output from this layer multiplied by the input layer to produce the spatial attention output.

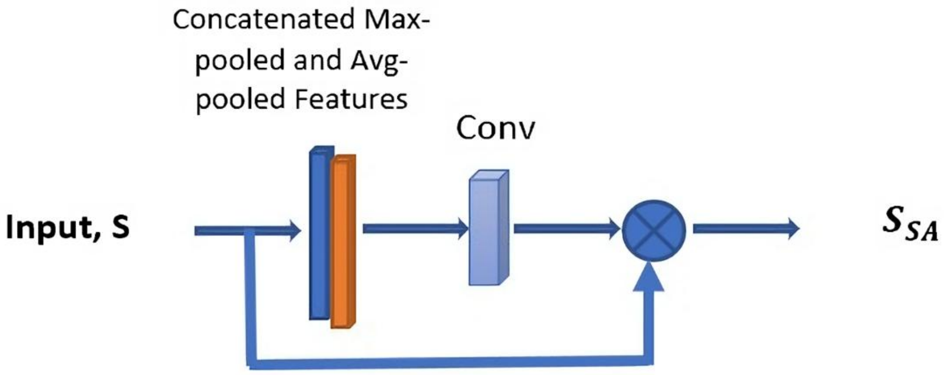


Figure 4. Spatial Attention Model (SAM)

Figure 4. Illustrate the SAM, the output of S_{SA}, can be calculated through feeding a concatenated max pooled and average pooled features which is passed from input spatial to conventional layer whose output layer is again with output spatial layer S_{SA}.

1.6 Temporal an Location Sensitive Fused Attention model on Pyramid Features

The proposed method leverage attention mechanisms and VGGNet to create an innovative framework for crowd management. This framework processed video sequence as input. Two parallel streams are provided with video sequences such as I_i *and* $I_{i-\text{r}}$ and these inputs are fed into the foundational layer of VGGNet simultaneously. Figure 1 illustrate the flow architecture of TLFA_PF.

Journal of Information Systems Engineering and Management
2025, 10(38s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

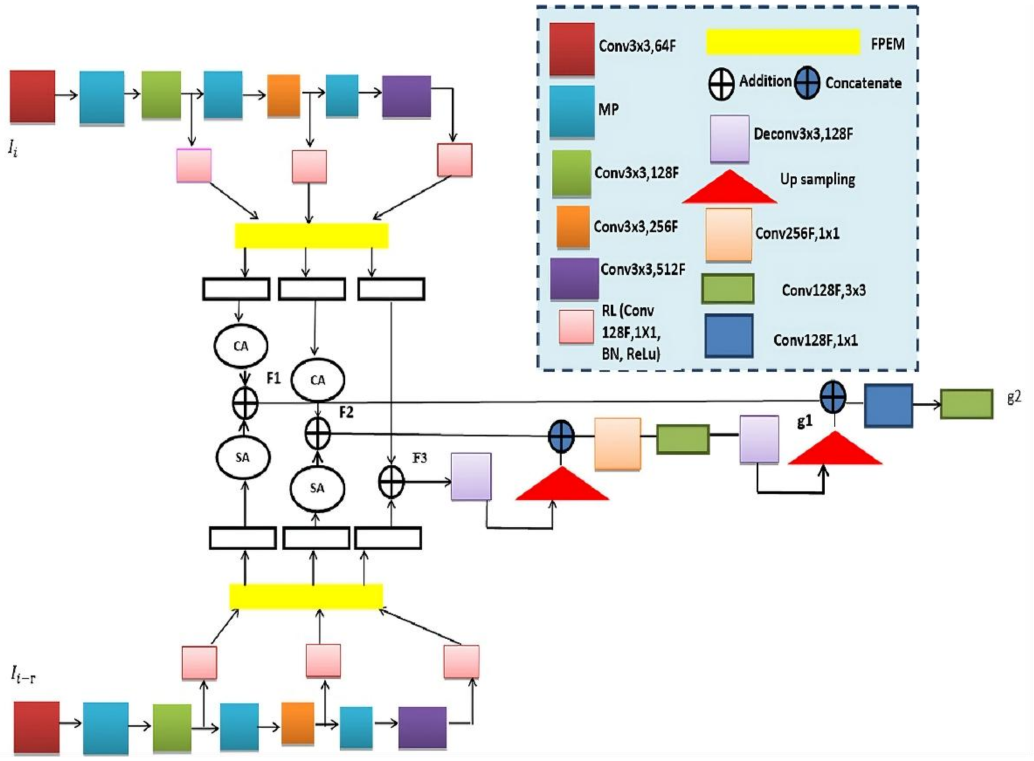


Figure 5. Proposed Architecture of TLFA_PF

Figure 5. depict the two input are passed through a conventional layer with a 3*3 kernel and 64 filters, and the resulting output is then routed through a max pooling layer the max pooling layer selects the prominent features, which are subsequently fed into another conventional layer with 3*3 kernel and 128 filters. The output from the top branch of this layer is labelled asB_1^1, while the output from the top branch is labelled asB_2^1. Following this, both outputs are sent through another max pooling layer before being routed to their respective conventional layers, each equipped with 256 filters. The output from the top branch is designed asB_1^2, and the output from the bottom branch is designated asB_2^2. This process continues with another max pooling layer, followed by a conventional layer that employs 512 filters and a 3*3 kernel size. The output from the top branch at this stage s denoted asB_1^3, while the outline from the bottom branch is denoted asB_2^3.

The outputs $B_l^s, where\ l = 1, 2\ and\ s = 1, 2, 3$ are separately processed through a reduction layer. This layer is applied using a convolutional layer with a kernel size of 1 and 128 filters. The output from this layer undergoes batch normalization followed by the application of the ReLU activation function, resulting in the output described in equation (1).

$$F_l^s = RL(B_l^s)|_{\ l=1,2\ ands=1,2,3} \qquad 1)$$

Thus, the top branch produces the outputs F1, F2, and F3, while the bottom branch yields$F_2^1, F_2^2, and\ F_2^3$. The inputs to this layer have sizes of 128, 256, and 512. The reduction layer transforms these inputs into outputs of 128 bits. The outputs from the reduction layer ($F_1^1, F_1^2, F_1^3, F_2^1, F_2^2, and\ F_2^3$) are then fed into a Feature Pyramid Enhancement Module (FPEM), resulting in distinct features for each input: ($FE_1^1, FE_1^2, FE_1^3, FE_2^1, FE_2^2, and\ FE_2^3$)are then the feature FE_1^1 is input into a channel attention model to generate the outputFE_{1CA}^1. Similarly, the feature FE_2^1undergoes spatial attention to produce the outputFE_{2SA}^1. This process is repeated for the second set of features, where channel attention and spatial attention are applied to obtain outputs for $FE_{1CA}^2$and FE_{2SA}^2 from features FE_1^2andFE_2^2. The combined features are calculated using the following equations (2) to (4):

$$F_1 = FE^1_{1CA} + FE^1_{2SA} \quad (2)$$

$$F_2 = FE^2_{1CA} + FE^2_{2SA} \quad (3)$$

$$F_3 = FE^3_1 + FE^3_2 \quad (4)$$

The de-convoluted and up-sampled output of $F_3$is concatenated withF_2, as illustrated in the figure. This combined feature is then convolved twice, followed by a single de-convolution and up-sampling to obtaing_1, which is subsequently concatenated withF_1. To achieve the final outputg_2, this result is passed through two convolutional layers with 128 filters and kernel sizes of 3 and 1, respectively. Thus, the proposed framework integrates attention mechanisms within a novel architecture to extract prominent features. In the first convolutional layer, features are extracted using a 3x3 filter with 64 filters. The output from this layer, referred to as convolutional layer 1, is then processed by a maximum pooling layer (Maxpool 1) with a filter size of 2x2 and a stride of 2. This process is detailed in the following

$$Convlayer1 = 2Dconv64{,}3 \times 3(input\ image1) \quad (4)$$
$$MP1 = maxpool2 \times 2 \times 2S(convlayer1) \quad (5)$$

Max pooling layers with a filter size of 3 and 128 filters are employed to extract features in Convolutional Layer 2. The first reduction layer, Reduced Layer 1, processes the output from Convolutional Layer 2. This output is then directed to a max pooling layer, Maxpool 2, which has a filter size of 2×2 and a stride of 2. This process is signified in the following equations

$$Conv\ layer2 = 2Dconv\ 128{,}3 \times 3(MP1) \quad (6)$$

$$Reduced\ layer1 = RL1(conv\ layer2) \quad (7)$$

$$MP2 = maxpool2 \times 2 \times 2S(convlayer2) \quad (8)$$

In Convolutional Layer 3, max pooling layers with a filter size of 3 and 256 filters are used for feature extraction. The second reduction layer, Reduced Layer 2, processes the output from Convolutional Layer 3, which is then passed to the max pooling layer Maxpool 3 with a filter size of 2×2 and a stride of 2. This is illustrated in the following equations:

$$Conv\ layer3 = 2Dconv\ 256{,}3 \times 3(MP2) \quad (9)$$

$$Reduced\ layer2 = RL2(conv\ layer3) \quad (10)$$

$$MP3 = maxpool2 \times 2 \times 2S(convlayer3) \quad (11)$$

In Convolutional Layer 4, max pooling layers are again utilized for feature extraction, this time with a filter size of 3 and 512 filters. The third reduction layer, Reduced Layer 3, processes the output from Convolutional Layer 4, as shown in the following equations:

$$Conv\ layer4 = 2Dconv\ 512{,}3 \times 3(MP3) \quad (12)$$

$$Reduced\ layer3 = RL3(conv\ layer4) \quad (13)$$

The outputs from Reduced Layers 1, 2, and 3 are then fed into the Feature Pyramid Enhancement Module (FPEM), as represented by:

$$FPEM1 = FPEM(Reduced\ layer1, Reduced\ layer2, Reduced\ layer3) \quad (14)$$

Features are extracted in the first convolutional layer using a filter size of 3x3 with 64 filters. The output from this layer, referred to as Convolutional Layer 1, is subsequently processed through a maximum pooling layer (Maxpool1) with a filter size of 2x2 and a stride of 2. This process is depicted in equations (4) and (5):

$FPEM2 = FPEM(Reduced\ layer4, Reduced\ layer5, Reduced\ layer6)$ (15)

The channel weight layer from $CA1$ and $SA1$ corresponds to$FPEM1$, while the spatial weight layer from $CA2$ and $SA2$ corresponds to$FPEM2$. The concatenation of $CA1$ and $SA1$ yields the result $F1$, while the concatenation of $CA2$ and $SA2$ yields$F2$. The concatenation of $FPEM03$ and $FPEM06$ results in$F3$. This procedure is applied to both the top and bottom layers, as shown in the following equations:

$CA1 = FPEM01(FPEM1)$ (16)

$SA1 = FPEM04(FPEM2)$ (17)

$F1 = add\ (CA1, SA1)$ (18)

$CA2 = FPEM02(FPEM1)$ (19)

$SA2 = FPEM05(FPEM2)$ (20)

$F2\ = add(CA2, SA2)$ (21)

$F3 = add(FPEM03, FPEM06)$ (22)

The output $F3$ is then sent into a de-convolution layer (Deconvlayer1) with a filter size of 128 and a kernel size of 3×3, followed by an up-sampling step. This output is combined with F2. The result of this concatenation (Concat1) is processed through Convolutional Layer 9 with 256 filters and a kernel size of 1×1, followed by Convolutional Layer 10 with a filter size of 128 and a kernel size of 3×3. The output from this process is then sent into another de-convolution layer (Deconvlayer2) followed by an up-sampling step. The resulting output g1 is concatenated with$F1$, producing another concatenated result (Concat2) that is fed into Convolutional Layer 11 with a kernel size of 128 and a kernel size of 1×1, followed by Convolutional Layer 12 with a filter size of 128 and a kernel size of 3×3. This process is summarized in the following equations:

$Deconvlayer1\ = 2DDeconv\ 128{,}3 \times 3(F3)$ (23)

$Upsample1\ = upsample(Deconvlayer1)$ (24)

$Concat1\ \ = concatenate(upsample1, F2)$ (25)

$Convlayer9\ = 2Dconv256{,}1 \times 1(concat1)$
(26)

$Convlayer10\ = 2Dconv128{,}3 \times 3(convlayer9)$ (27)

$Deconvlayer2\ = 2DDeconv\ 128{,}3 \times 3(conv\ layer10)$ (28)

$G1\ \ = upsample(Deconvlayer2)$ (29)

$Concat2\ \ = concatenate(g1, f1)$ (30)

$Convlayer11\ = 2Dconv128{,}1 \times 1(concat2)$ (31)

$Convlayer12\ = 2Dconv128{,}3 \times 3(convlayer11)$ (32)

$G2\ =\ Convlayer12$ (33)

The final output G2 is then sent for further processing.

Density Map Estimation

Extracting crowd features is crucial for evaluating crowd density. We proposed a method that utilizes local feature points to characterize the crowd based on the premise that areas with low population density exhibits sparser local features compared to region with high density. The process begins with density by assessing the proximity of these features. Visual surveillance systems have investigated various techniques for crowd management and monitoring, including crowd density analysis. From this perspective, generating region-specific crowd density maps is more advantageous than calculating a single overall density or simply counting the total number of individuals in a frame. Our approach transitions from global per-frame information to detailed local pixel-level analysis. Below is an outline of our method for estimating density maps. To assess each analysed frame, local features are first extracted.

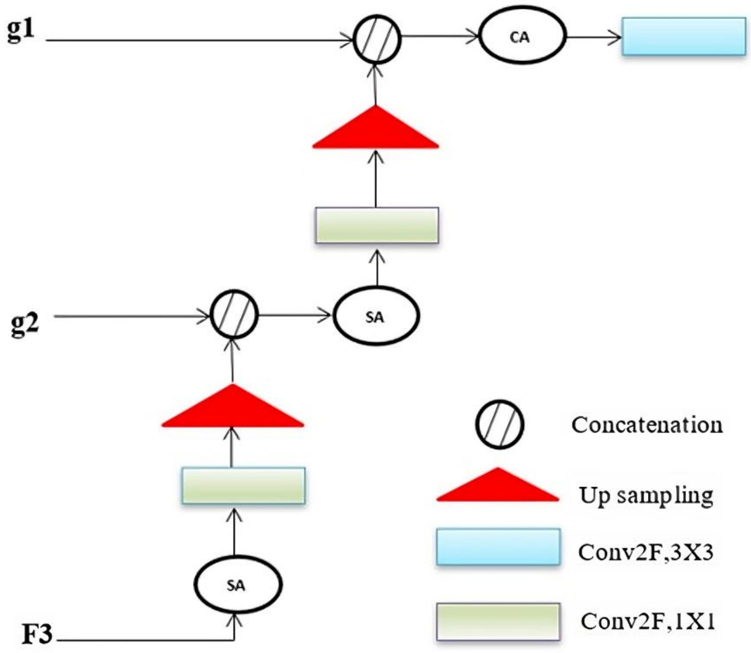


Figure 6. Density Map Estimation

The proposed methods for density map estimation is illustrated in Figure 6. The task consists of three processing stage and three distinct inputs. The initial input frame $F3$ undergoes spatial attention processing $Sa1$ followed by a convolution layer $CONVF, 1 \times 1$ kernel and up sampling$Up1$. Next, the second input $G2$ is concatenate with the output from the initial up sampling$Up1$. Subsequently, the second input $g2$ is concatenated with the output from the initial up-sampling$Up1$. This concatenated output is then fed into another spatial attention layer$Sa2$. After processing, it passes through the convolution layer CONVF$CONVF$again, followed by another up-sampling layer$Up2$. The third input $g1$ is combined with the output from the second input$g2$. The result of this concatenation is then directed into channel attention $Ca1$ for further processing through the third convolution layer $CONVF, 3 \times 3$ kernel. The entire procedure is detailed in the following equations.

$$Concat1output = Concat1(Up1(ConvF1 \times 1(Sa1(F3))), g2) \tag{34}$$

$$Concat2output = Concat2\ (Up2\ (ConvF1 \times 1(Sa2(Concat1output))), g1) \tag{35}$$

$$Final\ output = \ (ConvF3 \times 3\ (Ca1(Concat2output)) \tag{36}$$

Localized Map Estimation

The proposed localized estimation approach employs a local temporal estimator to highlight the overall shape of prominent objects in each current frame. This method effectively leverages the temporal consistency and strong correlations between adjacent frames. To capitalize on these temporal correlations, we introduce a novel localized estimation method, as shown in figure 7.

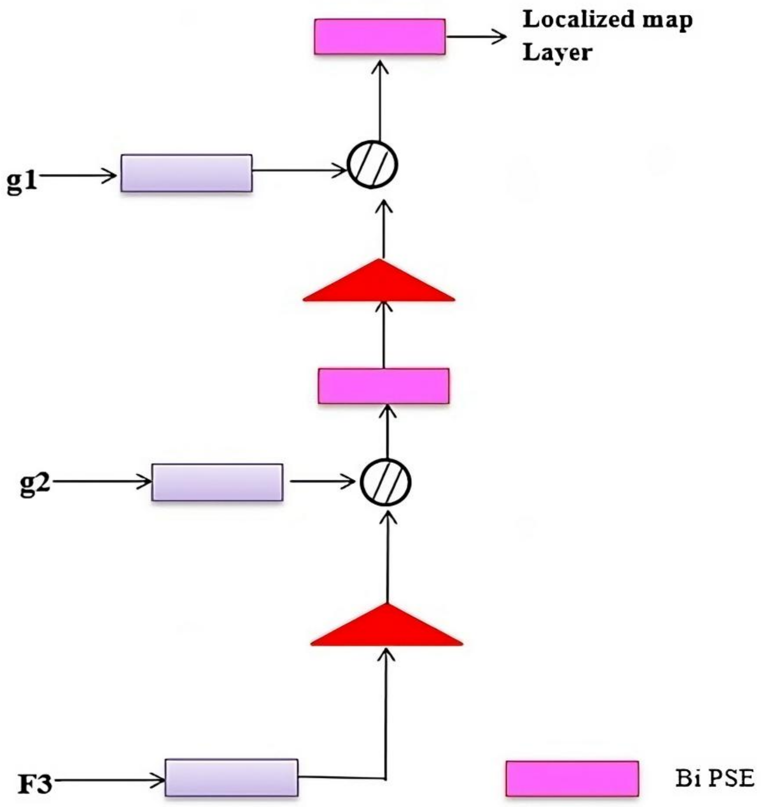


Figure 7. Localized Map Estimations

To capitalize on these temporal correlations, we introduce a novel localized estimation method, as shown in figure 7. The conventional layer $CONVF$ processes the initial inputs frame$F3$, which is then directed to an up sampling layer. Meanwhile, the conventional layer also received input. The output from $F3$ and $g2$ are sub sequential combined. This concatenated output is then sent to the Bi Pooling Squeeze Excitation block (Bi PSE layer) for further processing, followed by another up-sampling step. On the other hand, the convolution layer $CONVF$ receives input$g1$, and the outputs from $g2$ and $g1$ are merged. The resulting concatenated output is then passed to the Bi Pooling Squeeze Excitation block (Bi PSE layer), which serves as a localized map layer. The explanation of this process is illustrated in the following equations.

$$Concate1output = Concate1(Up1(ConvF1 \times 1(F3)), (g2) ConvF1 \times 1)) \tag{34}$$

$$Concate2output = Concate2\ (Up2(ConvF1 \times 1(Concate1output), (g1)\ ConvF1 \times 1)) \tag{35}$$

$$Final\ output = Concate2output(ConvF1 \times 1) \tag{36}$$

1.6.1 Bi Pooling Squeeze and Excitation Block

The computing unit known as the Squeeze and excitation block can be integrated into any transformation. This block enhances the representation capacity of CNN by allowing dynamic recalibration of channel wise features with minimal computational overhead. It not only fits seamlessly into existing architectures but also significantly improves performance. Essentially, the squeeze-and-excitation block enhances neural networks' ability to accurately map global information and channel dependencies, resulting in performance improvements through better calibration of filter outputs.

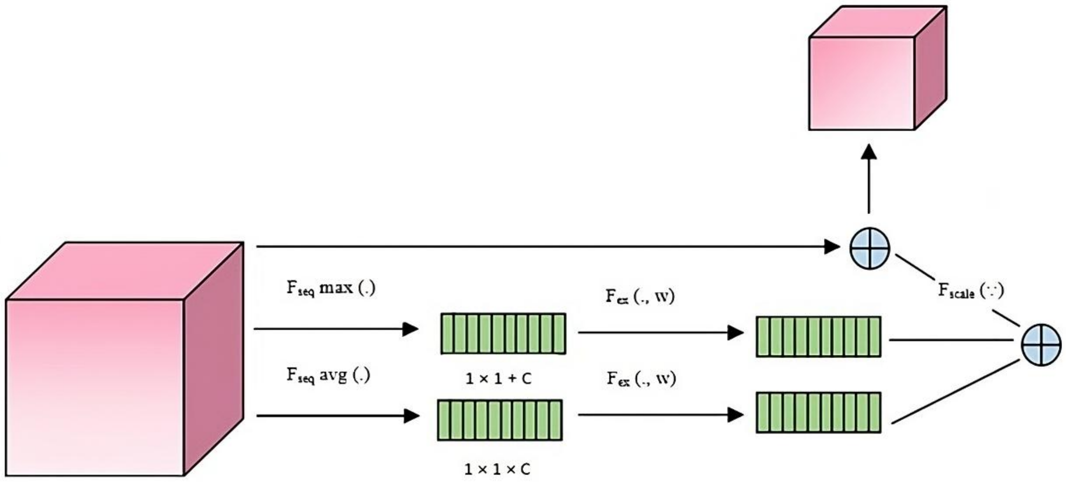


Figure. Bi-pooling Squeeze and Excitation

The proposed novel Bi-pooling Squeeze-and-Excitation Block employs both max pooling and average pooling techniques. Max pooling provides a more abstract representation, reducing over fitting and offering fundamental translation invariance to the internal representation, while also decreasing the learning burden by lowering the number of parameters. In contrast, average pooling generates a down-sampled feature map by calculating the average value of patches within a feature map, typically following a convolutional layer. Consequently, while max pooling captures the most prominent feature in a specific patch, average pooling yields the average of all features within that patch. The block receives a convolutional input and applies both average and max pooling to condense each channel into a single numerical value. Following this, non-linearity is introduced after a dense layer, which reduces the output channel complexity. Each channel is assigned a smooth gating function through a second dense layer followed by a sigmoid activation. Ultimately, this process assigns weights to each feature map from the convolution block on the "excitation" side of the network.

To extract features from an input image, a feature transformation (such as convolution) is first applied. The squeeze operation then condenses each output channel into a single value. Next, an excitation operation is performed on the output from the squeeze step to derive per-channel weights. The final output of the block is obtained by rescaling the feature map using these activations after determining the per-channel weights. The study illustrates how these building blocks can be stacked to create Squeeze-and-Excitation structures that generalize effectively across various datasets. Additionally, proposed model demonstrates that SE blocks significantly enhance performance for existing state-of-the-art CNNs with minimal additional computational cost.

RESULT AND DISCUSSION

1.7 Dataset Description

The drone crowd dataset comprises 112 videos segments, amounting to a total of 33,600 high definition frame (1920*1080 Pixels), captured under 70 distinct lighting conditions. It features 28800 person trajectories and over 4.8 million hand annotations along with various video leveling sequence elements. This extensive dataset was compiled using drones equipped with mounted cameras in four Chinese cities namely Daqing, Hong Kong, Tianjin, and Guangzhou. Currently, the Drone Crowd dataset is recognized as the most comprehensive resource available for crowd density estimation and localization tasks.

The complete workflow was developed using Python 3.7 and PyTorch 1.5, running on an Intel Core i7 processor with 16GB of RAM and an RTX-2060 6GB NVidia graphics card. For analysis, Tensor Flow-GPU 2.1.0 libraries were utilized, with a learning rate set at 0.01 and RMSprop as the chosen optimizer. The evaluation process was carried out over 200 epochs.

1.8 Performance Metrics

Density map estimation, as highlighted in previous studies, involves calculating the per pixel density while preserving spatial information about the number of individuals present at each location in the image. To evaluate performance, we utilize the Mean Absolute Error (MAE) [36] and Mean Square Error (MSE) [5].

$$MAE = \frac{1}{\sum_{i=1}^{k} N_i} \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left| z_{i,j} - \hat{z}_{i,j} \right|, \tag{37}$$

$$MSE = \sqrt{\frac{1}{\sum_{i=1}^{k} N_i \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left| z_{i,j} - \hat{z}_{i,j} \right|^2}} \tag{38}$$

Where Ni represent the count of frames in the i^{th} video and K denotes he overall number of video clips. The actual count of individuals in the j^{th} frame of thee i^{th} video clip is denoted by $Z_{i \cdot j}$ while the projected count is indicating by$Z_{i \cdot j}$. MAE and MSE as noted indicates the accuracy and dependability of the estimation.

Crowd Localization

As noted in, the optimal technique for assessing the number of people in a crowd involves recognizing each individual in a photo a counting these identifications, which is essential for uses like security and monitoring.

Every assesse method must deliver a list of recognized points along with confidences scores for ach test image. Ground truth localization are paired with estimated localization through a greedy algorithm that relies on confidence thresholds. To assess the localization through a greedy algorithm that relies on confidence thresholds. To assess the localization outcomes, we calculate the mean average precision (L-mAP) at different distance threshold (1, 2, 3, ..., 25 Pixel) distance limits. These metrics consider both duplicate detection and those that were missed for example, several detection of the same people.

Experimental Results

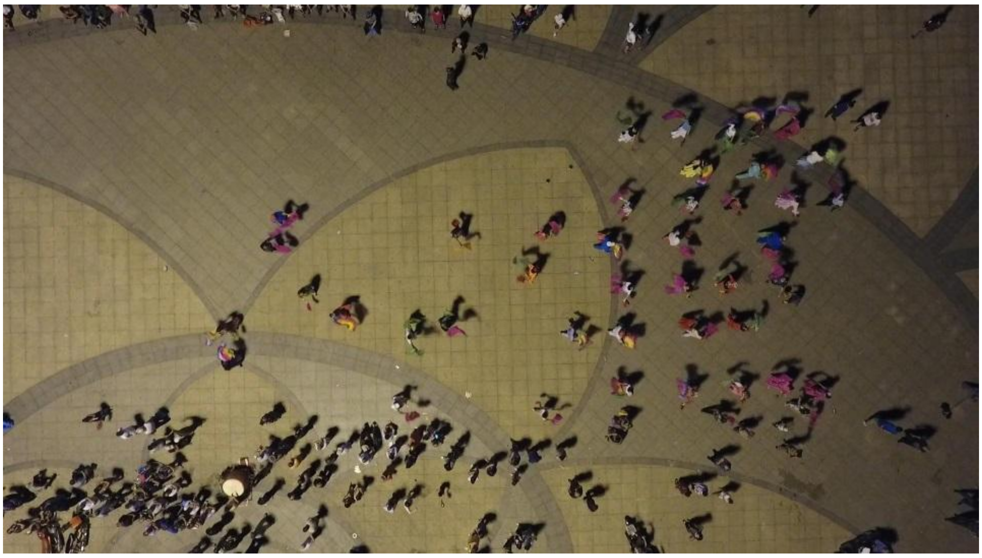
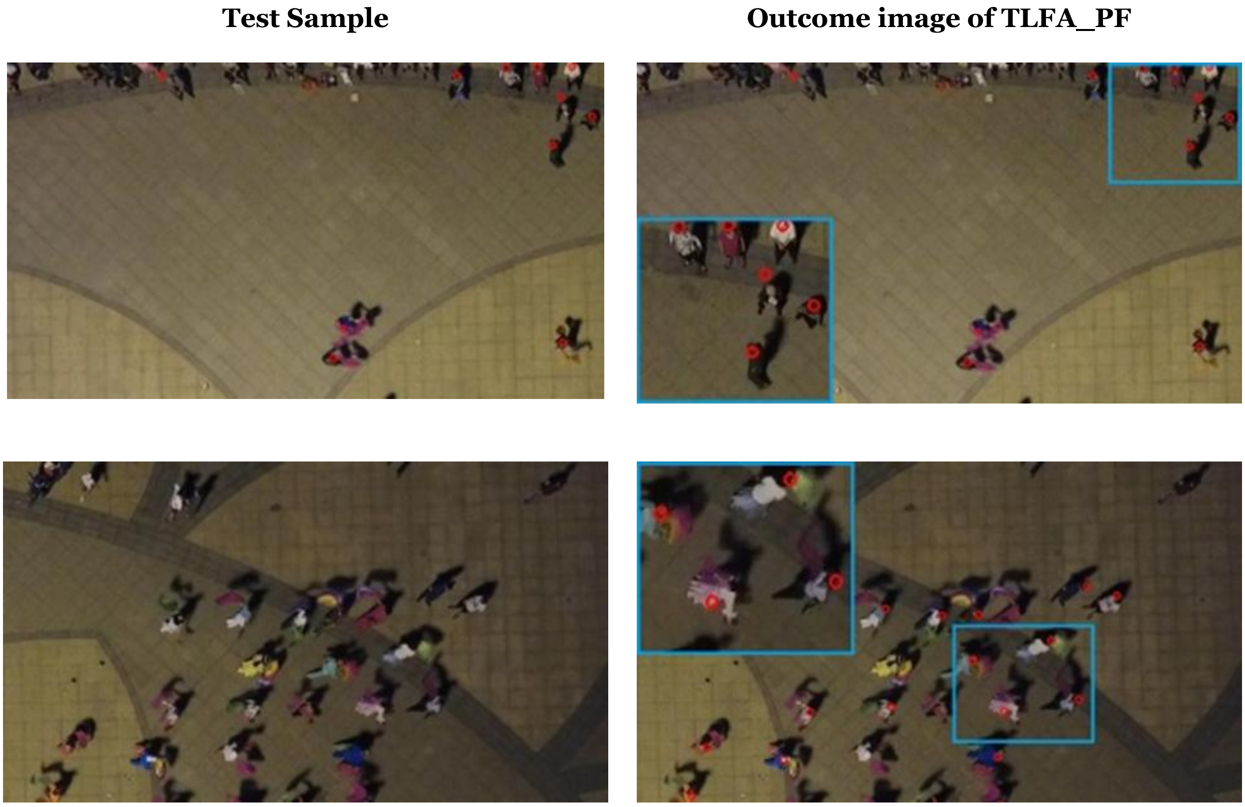


Figure 9. Full Image of the Location



Test Sample

Outcome image of TLFA_PF

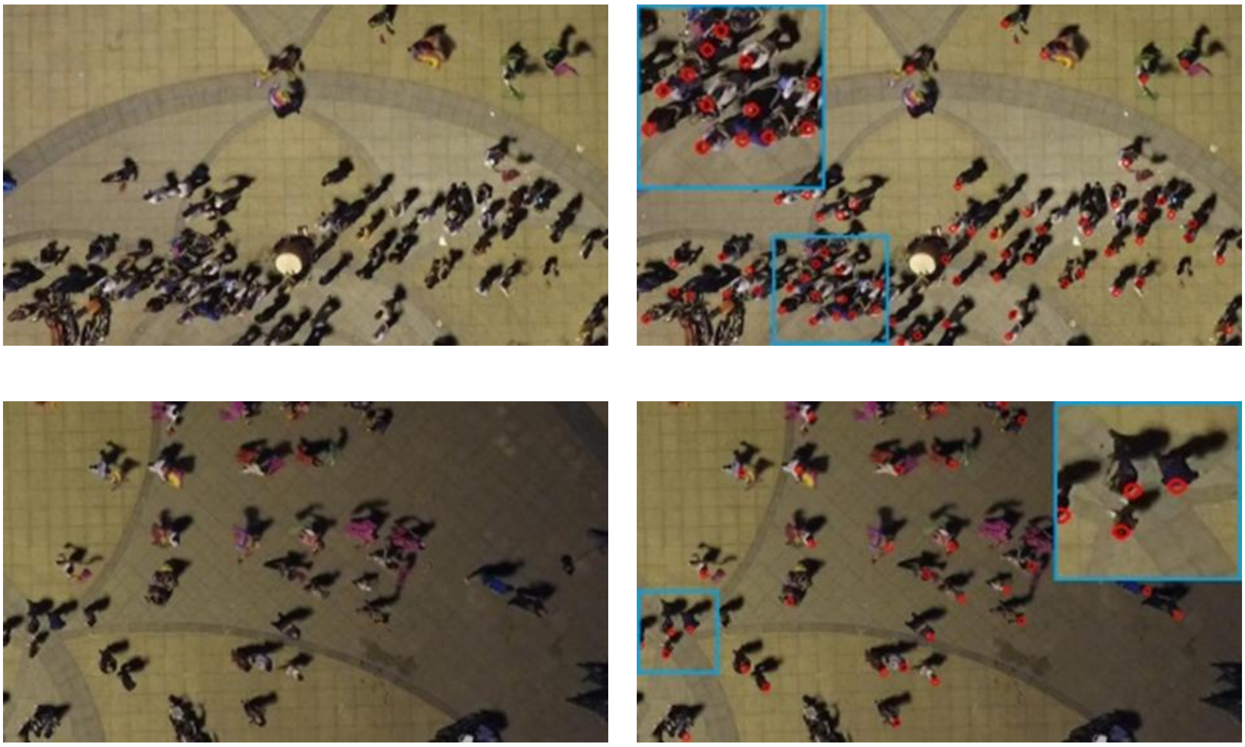


Figure 10. Output Images of the Proposed Work (A, B, C, and D) Correspond to the Input Images with Indication the Presence of Crowd to Assist In Localization

1.9 Performance Analysis

The performance analysis of the TLFA_PF model shown an overall MAE and MSE that reflects similar trends across environmental conditions. The model achieves its best accuracy in all methods. While both MAE and MSE values surge in sunny conditions, indicating potential challenges in maintaining accuracy under varying lighting conditions.

Table 1. MAE outperforms on TLFA_PF

Method	Speed FPS	Overall MAE	High MAE	Low MAE	Cloudy MAE	Sunny MAE	Night MAE	Crowded MAE	Sparse MAE
TLFA_PF	2.94	*15.4*	*15.7*	*15.9*	*13.8*	*18.7*	*14.8*	*18.1*	*13.5*

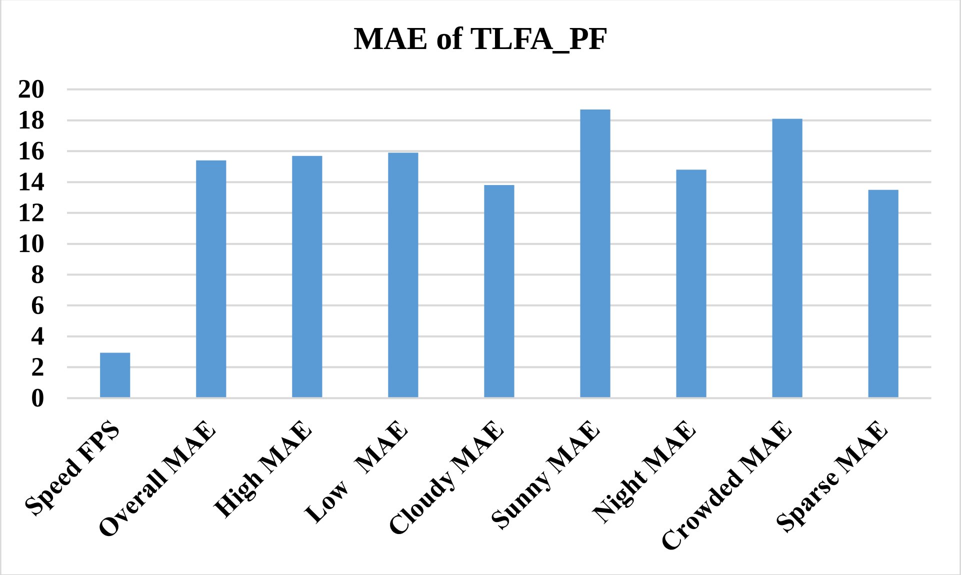


Figure 9. Graphical MAE in TLFA_PF

The table 1 and figure 9 highlights the performance of the TLFA_PF, which operates at a speed of 2.94 FPS and achieves an overall MAE of 15.4 across various conditions. Notably, TLFA_PF excels in cloudy conditions with the lowest MAE of 13.8, while its highest error occurs in sunny conditions at 18.7. Overall, the model demonstrates strong accuracy and consistent performance in crowd density estimation, making it a valuable tool for real-time monitoring applications

Table 2. MSE outperforms on TLFA_PF

Method	Speed FPS	Overall MSE	High MSE	Low MSE	Cloudy MSE	Sunny MSE	Night MSE	Crowded MSE	Sparse MSE
TLFA_PF	2.94	18.2	17.7	19.3	16.5	20.4	18.6	21.2	16.3

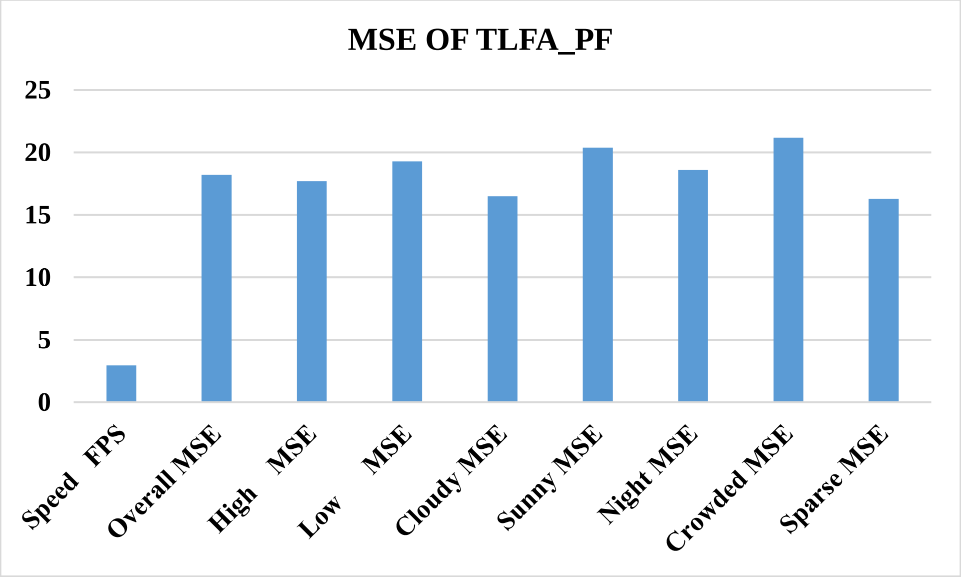


Figure 10. Graphical MSE in TLFA_PF

Similar to MAE Figure 10 and table 2 highlights the TLFA_PF, which operates at a speed of 2.94 FPS and achieves an overall MAE of 18.2. The model performs best in cloudy conditions with an MAE of 16.5, while the highest error occurs in sunny conditions at 20.4.

1.10 Comparison Analysis

The density map estimation is performed by analyzing pixel values across various environment, including cloudy environments, including cloudy, sunny, night, crowded, and sparse conditions. MAE is calculated at both high and low levels to derive an overall MAR. The performance evaluations includes algorithm such as MCNN [37], MSCNN [38], C-MTL [5], ACSSCP[39], LCFCN [27], SwitchCNN, CSRNet, AMCN, StackPoolig, STANet (w/o ms), DA-Net and STNNet [37]. Table 1 and 2 present the MAE and MSE results for the density map on the Drone Crowd dataset, respectively.

Table 3. MAE of the density map on Drone Crowd dataset

Method	Speed FPS	Overall MAE	High MAE	Low MAE	Cloudy MAE	Sunny MAE	Night MAE	Crowded MAE	Sparse MAE
MCNN	28.98	34.7	36.8	31.7	21	39	67.2	29.5	37.7
C-MTL	2.31	56.7	53.5	61.5	59.5	56.6	48.2	81.6	42.2
MSCNN	1.76	58	58.4	57.5	64.5	53.8	46.8	91.4	38.7
LCFCN	3.08	136.9	126.3	152.8	147.1	137.1	105.6	208.5	95.4
SwitchCNN	0.014	66.5	61.5	74	56	69	92.8	67.7	65.7
ACSCP	1.58	48.1	57	34.8	42.5	37.3	86.6	36	55.1
AMDCN	0.16	165.6	166.7	163.8	160.5	174.8	162.3	165.5	165.6
CSRNet	3.92	19.8	17.8	22.9	12.8	19.1	42.3	20.2	19.6
Stack Pooling	0.73	68.8	68.7	68.8	66.5	74	65.2	95.7	53.1
DA-Net	2.52	36.5	41.5	28.9	45.4	26.5	29.5	56.5	24.9
STANet (w/o ms)	9.49	26.3	27.3	24.7	21.3	29.5	34.7	22.4	28.5
STNNet	3.41	15.8	16	15.6	14.1	19.9	12.9	18.5	14.3
TLFA_PF	2.94	***15.4***	***15.7***	***15.9***	***13.8***	***18.7***	***14.8***	***18.1***	***13.5***

From Table 3. Present the performance metrics of various crowd counting methods, highlighting their processing speed I frame per second (FPS) alongside their MAE across different conditions. For instance, MCNN operates at 28.98 FPS with an overall MAE of 34.7, showing high MAE of 36.8 and low MAE of 31.7. In contrast, C-MTL has a slower speed of 2.31 FPS but achieves a higher overall MAE of 56.7. The MSCNN model runs at 1.76 FPS, maintaining an overall MAE of 58, indicating consistent performance across various conditions. Notably, LCFCN demonstrates the highest speed at 3.08 FPS with an impressive overall MAE of 136.9, while SwitchCNN operates at a very low speed of 0.014 FPS with an overall MAE of 66.5. The lightweight model AMDCN excels with a high speed of 165.6 FPS and an overall MAE of 15.4, showcasing its efficiency in real-time applications. Other models like CSRNet and DA-Net perform at speeds of 3.92 FPS and 2.52 FPS, respectively, with overall MAEs of 19.8 and 36.5, reflecting competitive accuracy in various environments. Overall, the results illustrate a trade-off between speed and accuracy, as evidenced by the varying MAE values across different environmental conditions such as cloudy, sunny, night, crowded, and sparse scenarios.

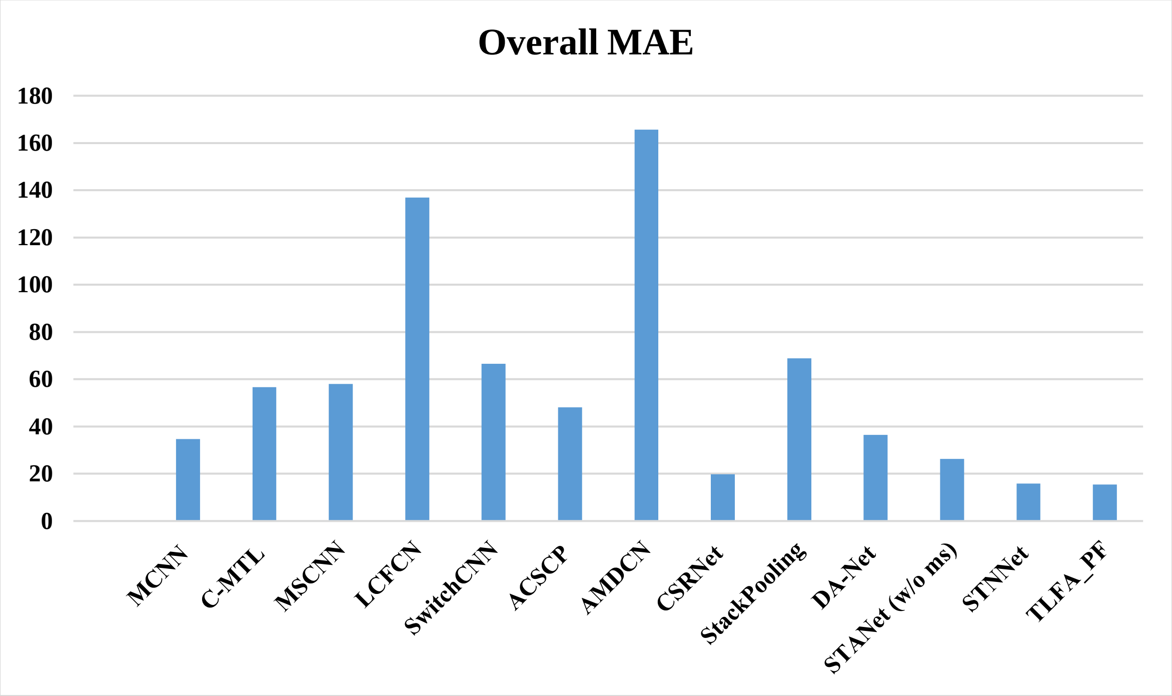


Figure 9. Overall MAE

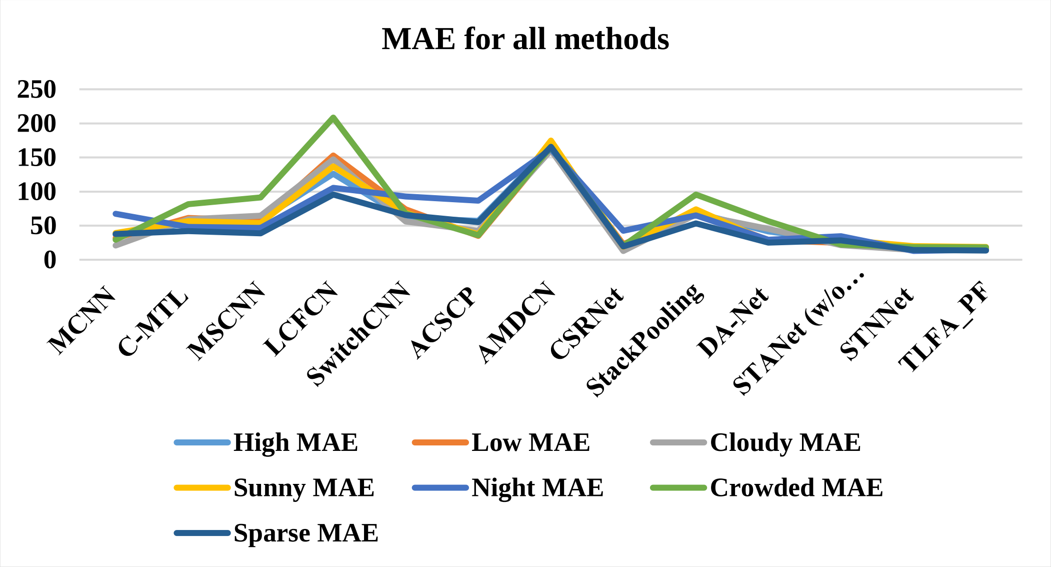


Figure 10. MAE estimation for all methods

Table 4. MSE of the Density Map on the Drown Crowd Dataset

Method	Speed FPS	Overall	High	Low	Cloudy	Sunny	Night	Crowded	Sparse
		MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE
MCNN	28.98	42.5	44.1	40.1	27.5	43.9	68.7	35.3	46.2
C-MTL	2.31	65.9	63.2	69.7	66.9	67.8	58.3	88.7	47.9
MSCNN	1.76	75.2	77.9	71.1	85.8	65.5	57.3	106.4	48.8
LCFCN	3.08	150.6	140.3	164.8	160.3	151.7	113.8	211.1	110
SwitchCNN	0.014	77.8	74.2	83	63.4	80.9	105.8	79.8	76.7
ACSCP	1.58	60.2	70.6	39.7	46.4	44.3	106.6	41.9	68.5
AMDCN	0.16	167.7	168.9	165.9	162.3	177.1	164.3	167.7	167.8
CSRNet	3.92	25.6	25.4	25.8	16.6	22.5	45.8	24	26.5
Stack Pooling	0.73	77.2	77.1	77.3	75.9	83.4	67.4	101.1	59.1
DA-Net	2.52	47.3	54.7	33.1	58.6	31.3	34	68.3	28.7
STANet (w/o ms)	9.49	31.4	33.9	27.1	23.2	37.7	38	25	34.5
STNNet	3.41	18.7	18.4	19.2	17.2	22.5	14.4	21.6	16.9
TLFA_PF	**2.94**	**18.2**	**17.7**	**19.3**	**16.5**	**20.4**	**18.6**	**21.2**	**16.3**

The table 4, compares various neural network method based on their performance metrics, specifically MSE and processing speed in FPS as in MAE, across different environment conditions. Notably, CSRNet emerges as the best performer with the lowest overall MSE of 25.6, indicating high accuracy, while also maintaining a moderate speed of 3.92 FPS. In contrast, AMDCN and LCFCN exhibit high processing demands with speeds of 0.16 FPS and 3.08 FPS, respectively, but suffer from higher MSE values, suggesting lower accuracy. Environmental conditions significantly affect performance; for example, MCNN performs best at night with an MSE of 27.5 but struggles in other scenarios. C-MTL excels in sunny conditions with an MSE of 66.9, highlighting the variability in model effectiveness based on context. Overall, this analysis underscores the trade-offs between speed and accuracy among the different architectures, providing insights for their application in tasks such as crowd counting and image recognition.

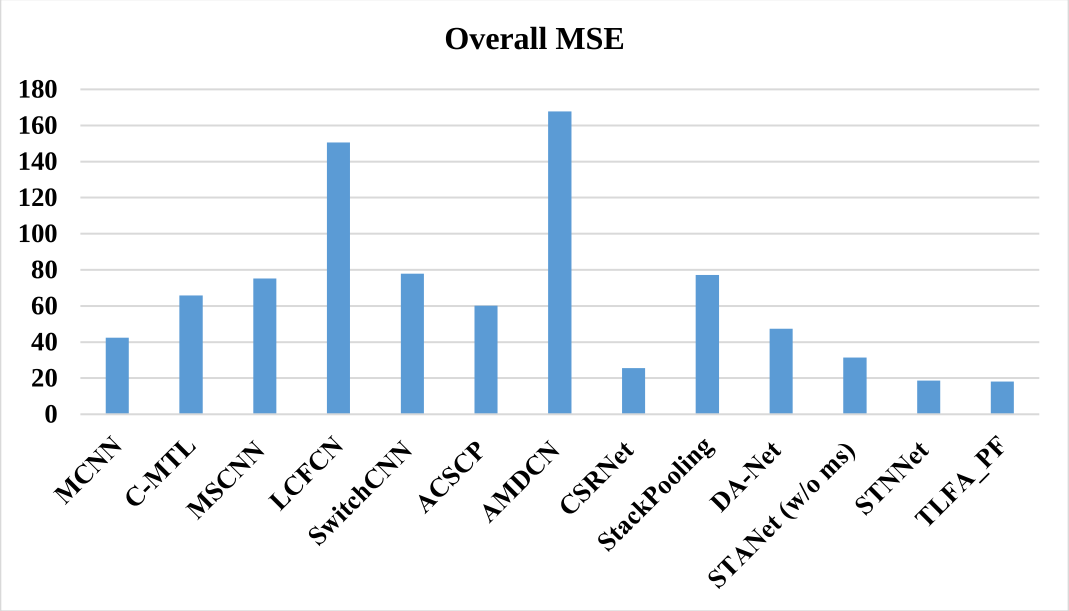


Figure 11. Overall MSE

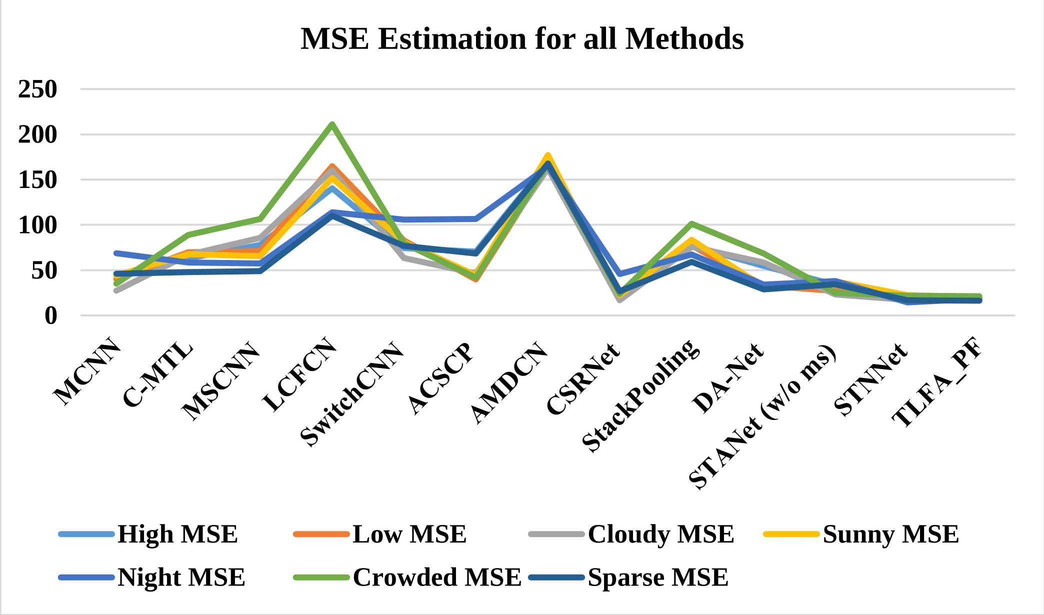


Figure 12. MSE estimation for all methods

From the above figures 10, 11,12 shows the proposed work performs better than other existing methods. The crowd localization on Drone Crowd dataset is analyzed using average precision.

CONCLUSION

The TLFA_PF marks a substantial advancement in the field of crowd density estimation and localization utilizing drone imagery. This innovative model effectively overcomes the limitations associated with traditional crowd monitoring methods, which often struggle with challenges such as occlusion, low resolution, and high crowd density. By achieving an overall Mean Absolute Error (MAE) of 15.4 and a Mean Squared Error (MSE) of 18.2, TLFA_PF demonstrates its capability to accurately predict crowd densities across various environmental conditions. The model's architecture incorporates advanced techniques such as spatial and temporal attention mechanisms, which allow it to extract critical features from drone-captured images at multiple scales while minimizing computational complexity. A key innovation is the introduction of the Bi Pooling Squeeze and Excitation Block, which enhances the model's ability to emphasize important features selectively, thereby improving its performance in discerning variations in crowd density. Furthermore, TLFA_PF exhibits superior Average Precision values for crowd localization compared to existing methods, highlighting its effectiveness in accurately identifying individuals within a crowd. The experimental results underscore the robustness and reliability of TLFA_PF, making it a valuable tool for real-time crowd monitoring applications. Ultimately, this research contributes significantly to the field of computer vision by providing an efficient and effective solution for enhancing safety and management in public spaces. The successful implementation of TLFA_PF paves the way for future advancements in aerial surveillance technologies, further improving our ability to monitor and manage crowded environments effectively.

DECLARATION

Competing Interests – There is no competing interest for this study.

Funding Information – There is no funding for this study

Author contribution - All the author contribute equally to this paper.

Data Availability Statement – Not applicable

Research Involving Human and /or Animals – Not applicable

Informed Consent – Not application

REFERENCE

[1] G. Castellano, E. Cotardo, C. Mencar, and G. Vessio, "Density-based clustering with fully-convolutional networks for crowd flow detection from drones," *Neurocomputing,* vol. 526, pp. 169-179, 2023.

[2] A. S. Saif and Z. R. Mahayuddin, "Crowd density estimation from autonomous drones using deep learning: challenges and applications," *Journal of Engineering and Science Research,* vol. 5, no. 6, pp. 1-6, 2021.

[3] Y. Lei, H. Zhu, J. Yuan, G. Xiang, X. Zhong, and S. He, "DenseTrack: Drone-based Crowd Tracking via Density-aware Motion-appearance Synergy," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 2050-2058.

[4] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Communications Magazine,* vol. 55, no. 2, pp. 128-134, 2017.

[5] M. Nazeer, K. Sharma, S. Sathappan, P. Srilatha, and A. A. K. Mohammed, "Improved STNNet, A benchmark for detection, tracking, and counting crowds using Drones," *MethodsX,* vol. 13, p. 102820, 2024.

[6] S. Rallabandi, V. Madhan, A. Telaprolu, and M. Nazeer, "Improved STNNet: A Benchmark from Detection, Tracking and Counting Crowds using Drones," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, 2024: IEEE, pp. 1-6.

[7] M. Zhao, C. Zhang, J. Zhang, F. Porikli, B. Ni, and W. Zhang, "Scale-aware crowd counting via depth-embedded convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 30, no. 10, pp. 3651-3662, 2019.

[8] L. Wen *et al.*, "Detection, Tracking, and Counting Meets Drones in Crowds: A Benchmark," *arXiv preprint arXiv:2105.02440,* 2021.

[9] L. Mei, M. Yu, L. Jia, and M. Fu, "Crowd Density Estimation via Global Crowd Collectiveness Metric," *Drones,* vol. 8, no. 11, p. 616, 2024.

[10] H. Li *et al.*, "Video crowd localization with multifocus gaussian neighborhood attention and a large-scale benchmark," *IEEE Transactions on Image Processing,* vol. 31, pp. 6032-6047, 2022.

[11] F. Zhu, H. Yan, X. Chen, T. Li, and Z. Zhang, "A multi-scale and multi-level feature aggregation network for crowd counting," *Neurocomputing,* vol. 423, pp. 46-56, 2021.

[12] X. Zhang, Y. Sun, Q. Li, X. Li, and X. Shi, "Crowd density estimation and mapping method based on surveillance video and GIS," *ISPRS International Journal of Geo-Information,* vol. 12, no. 2, p. 56, 2023.

[13] Y. Hu *et al.*, "CLDE-Net: crowd localization and density estimation based on CNN and transformer network," *Multimedia Systems,* vol. 30, no. 3, p. 120, 2024.

[14] M. Woźniak, J. Siłka, and M. Wieczorek, "Deep learning based crowd counting model for drone assisted systems," in *Proceedings of the 4th ACM MobiCom workshop on drone assisted wireless communications for 5G and beyond*, 2021, pp. 31-36.

[15] Q. Liu, Y. Zhong, and J. Fang, "Crowd counting network based on attention feature fusion and multi-column feature enhancement," *Journal of Visual Communication and Image Representation,* p. 104323, 2024.

[16] Z. Zhang, "Drone-YOLO: an efficient neural network method for target detection in drone images," *Drones,* vol. 7, no. 8, p. 526, 2023.

[17] B. Ptak, D. Pieczyński, M. Piechocki, and M. Kraft, "On-board crowd counting and density estimation using low altitude unmanned aerial vehicles—looking beyond beating the benchmark," *Remote Sensing,* vol. 14, no. 10, p. 2288, 2022.

[18] M. R. Bhuiyan, J. Abdullah, N. Hashim, and F. Al Farid, "Video analytics using deep learning for crowd analysis: a review," *Multimedia Tools and Applications,* vol. 81, no. 19, pp. 27895-27922, 2022.

[19] L. M. Wastupranata and R. Munir, "Convolutional neural network-based crowd detection for COVID-19 social distancing protocol from unmanned aerial vehicles onboard camera," *Journal of Applied Remote Sensing,* vol. 17, no. 4, pp. 044502-044502, 2023.

[20] A. Ilyas and N. Bawany, "Crowd dynamics analysis and behavior recognition in surveillance videos based on deep learning," *Multimedia Tools and Applications,* pp. 1-35, 2024.

[21] O. Elharrouss *et al.*, "Drone-SCNet: Scaled cascade network for crowd counting on drone images," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 57, no. 6, pp. 3988-4001, 2021.

[22] A. A. Assefa, W. Tian, N. W. Hundera, and M. U. Aftab, "Crowd Density Estimation in Spatial and Temporal Distortion Environment Using Parallel Multi-Size Receptive Fields and Stack Ensemble Meta-Learning," *Symmetry,* vol. 14, no. 10, p. 2159, 2022.

[23] 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, 2023.

[24] J. Gao *et al.*, "Deep rank-consistent pyramid model for enhanced crowd counting," *IEEE Transactions on Neural Networks and Learning Systems,* 2023.

[25] P. Serrano, M. Gramaglia, F. Mancini, L. Chiaraviglio, and G. Bianchi, "Balloons in the sky: unveiling the characteristics and trade-offs of the Google loon service," *IEEE Transactions on Mobile Computing,* vol. 22, no. 6, pp. 3165-3178, 2021.

[26] M. D. B. Pranoto, M. I. Sani, and M. I. Sari, "Aerial Object Tracking System on Micro Quadrotor Drone for Crowd Detection in Small-Scale Area," in *The 6th International Conference on Vocational Education Applied Science and Technology (ICVEAST 2023)*, 2023: Atlantis Press, pp. 992-1006.

[27] L. Wen *et al.*, "Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network," *arXiv preprint arXiv:1912.01811,* 2019.

[28] A. N. Alhawsawi, S. D. Khan, and F. U. Rehman, "Enhanced YOLOv8-Based Model with Context Enrichment Module for Crowd Counting in Complex Drone Imagery," *Remote Sensing,* vol. 16, no. 22, p. 4175, 2024.

[29] L. Wen *et al.*, "Detection, Tracking, and Counting Meets Drones in Crowds: A Benchmark," *arXiv e-prints,* p. arXiv: 2105.02440, 2021.

[30] M. A. Khan, H. Menouar, and R. Hamila, "DroneNet: Crowd Density Estimation using Self-ONNs for Drones," *arXiv preprint arXiv:2211.07137,* 2022.

[31] S. Nag, Y. Khandelwal, S. Mittal, C. K. Mohan, and A. K. Qin, "ARCN: A Real-time Attention-based Network for Crowd Counting from Drone Images."

[32] J. Chen, S. Xiu, X. Chen, H. Guo, and X. Xie, "Flounder-Net: An efficient CNN for crowd counting by aerial photography," *Neurocomputing,* vol. 420, pp. 82-89, 2021.

[33] M. Wang, X. Zhou, and Y. Chen, "JMFEEL-Net: a joint multi-scale feature enhancement and lightweight transformer network for crowd counting," *Knowledge and Information Systems,* pp. 1-21, 2024.

[34] D. Du *et al.*, "VisDrone-CC2020: The Vision Meets Drone Crowd Counting Challenge Results," *arXiv preprint arXiv:2107.08766,* 2021.

[35] Y. Wang, S. Hu, G. Wang, C. Chen, and Z. Pan, "Multi-scale dilated convolution of convolutional neural network for crowd counting," *Multimedia Tools and Applications,* vol. 79, pp. 1057-1073, 2020.

[36] O. Elharrouss, H. H. Mohammed, S. Al-Maadeed, K. Abualsaud, A. Mohamed, and T. Khattab, "Crowd density estimation with a block-based density map generation."

[37] R. Gouiaa, M. A. Akhloufi, and M. Shahbazi, "Advances in convolution neural networks based crowd counting and density estimation," *Big Data and Cognitive Computing,* vol. 5, no. 4, p. 50, 2021.

[38] V. K. Sharma, R. N. Mir, and C. Singh, "Scale-aware CNN for crowd density estimation and crowd behavior analysis," *Computers and Electrical Engineering,* vol. 106, p. 108569, 2023.

[39] L. Xiong, Y. Zeng, X. Huang, Z. Li, and P. Huang, "MLANet: multi-level attention network with multi-scale feature fusion for crowd counting," *Cluster Computing,* pp. 1-18, 2024.