JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CCSA: Crowd Counting with Stability Analysis using Adversarial Network and CNN

[1]B.Ganga, [2]Lata B T, and [3]Venugopal KR Fellow, IEEE
[1]Research Scholar, [2]Associate Professor, [3]Former Vice-Chancellor
[1, 2]CSE, UVCE, Bengaluru, India
[3]BU, Bengaluru, India

Abstract : Crowd analysis, especially the precise counting of individuals, is becoming crucial in fields like urban planning, transportation, and event management. The two main challenges are: accurately counting individuals in varying crowd sizes and monitoring crowd density. To address the above challenges, we have proposed the Crowd Counting Stability Analyzer (CCSA) architecture to enable crowd counting and stability analysis in a density map. The CCSA architecture is designed in two phases (i) Crowd Counting and (ii) Stability Analyzer. The crowd counting uses a Unet Generative Adversarial Network (UGAN), composed of Uskip Connection, Unet Generator, and Discriminator. The crowd stability analyzer uses a Convolutional Neural Network (CNN) to classify the crowd density as very low, low, moderate, high, and very high in a density map. Additionally, to present crowd count details and their corresponding categories, we have constructed a new dataset known as 'DataC' having 48 images with 12 distinct categories. The observation is evaluated using Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics on diverse scenes and densities in the ShanghaiTech (A, B), UCF CC 50, and DataC datasets. The experimental results show the model's effectiveness over MSGAN with the MAE and MSE metrics on the above datasets.

IndexTerms - Unet Generative Adversarial Network (UGAN), CNN, USkip Connection, Density Map, Crowd Stability

I. INTRODUCTION

In intelligent video surveillance, crowd analysis techniques such as crowd counting and crowd stability are crucial for effective crowd management. These techniques are vital in various applications such as public space planning, smart city governance, traffic control, automated transport systems, and real-time disaster prevention. Crowd counting involves estimating the number of people in different scenes and it is conducted using detection, regression, and density estimation methods [1]. In density estimation methods, challenges in crowd counting such as low accuracy are addressed by utilizing the spatial structures of the crowd. This enhanced accuracy provides important insights into crowd dynamics. Crowd stability refers to the ability of the crowd to maintain safe and orderly movement during disasters, such as fire accidents, terror attacks, and earthquakes. It provides valuable insights by monitoring potential changes in the crowd gathering patterns and crowd densities [2]. The survey on crowd counting methods in computer vision, including architectures, learning methods, and evaluation metrics, are presented in [3], and the survey on Convolutional Neural Network (CNN)-based methods for crowd behavior analysis, covering its architectures, optimization methods, datasets, and temporal data approaches, are discussed in [4].

A. Motivations:

Enhanced crowd counting and stability analyzers are essential for crowd management and disaster prevention in applications such as event management, transportation hubs, and public spaces.

B. Contributions:

The main contributions of this work are to design the architecture of Crowd Counting Stability Analyzer (CCSA) to accurately count individuals and to assess Crowd Stability (CS) using density maps in the crowd.

(i) The CCSA is developed in two phases namely (1) Crowd Counting to accurately estimate the number of people and (2) Stability Analyzer to assess the potential risk level in the crowd.

(ii) The Crowd Counting incorporates UGAN, consisting of a Uskip Connection and a Unet Generator, with a discriminator. The Uskip Connection includes two passes namely (i) forward pass and (ii) backward pass which preserves high-resolution and spatial features.

(iii) The Uskip Connection is designed to interpret latent representations, to solve image segmentation challenges while preserving the same dimensions for both input and output.

(iv) The Unet Generator produces high-quality density maps, while the discriminator differentiates real and synthetic images, contributing to efficient crowd management.

C. Organizations of the Work:

The paper is organized as follows. In section II, the Related work is discussed in brief, In section III, the CCSA Architecture is explained with background, problem definition, and architecture. In section IV, the CCSA Algorithm, Flowchart, Evaluation metrics, and Performance metrics are reviewed. In sections V and VI, Experimental Results, and Results and Discussions are analyzed. The conclusion is presented in section VII.

II. RELATED WORK:

In Table 1, a literature survey of related papers is presented. Gaewei et al., [5] have designed a Cross-scale convolution Spatial Generative Adversarial Network (CSGAN) to estimate the density of crowd images. The merits are that it uses loss functions such as adversarial, perceptual, and consistency to enhance counting accuracy. However, challenges like crowd density and path movements still need to be addressed in this work. Xinyue et al., [6] have combined density awareness and background segmentation to explore crowd enumeration with multitask learning methods. The drawback of the method is that dense dataset performance is not explored. Weixing et al., [7] have designed a dilated deep convolutional neural network, incorporating filters that utilize dilated convolution to enhance the accuracy of crowd counting. The dense datasets such as UCF CC 50 are not trained in this work to enhance the accuracy.

Aichun et al., [8] have designed a Cascaded Attentional Generative Adversarial Network (CAGAN) using two models namely an attentional generator and a cascaded attentional discriminator. Dense and sparse datasets demonstrate better accuracy, while baseline methods like Multicolumn Convolutional Neural Network (MCNN) have not been employed for attentional generators in this work. Yuan et al., [9] have addressed challenges in saliency detection with a Linear Feedback Control System (LFCS), a semi-supervised learning framework. The LFCS optimizes saliency maps by integrating multiple cues and iterative convergence. It offers precise saliency estimation, however, achieving effective saliency detection in complex scenes remains a challenge. Xinghao et al., [10] have proposed an encoder-decoder Convolution Neural Network (CNN) to solve the challenges like multifaceted backgrounds, illusory perspective, and occlusion. The cross-scene datasets are accurately estimated using patch absolute error, but this work does not enhance the accuracy of detecting groups in close proximity or in cases of shrinking head size.

Wang et al., [11] have addressed challenges in crowd counting such as data and methodology using GCC dataset and domain adaptation. The challenges in data are addressed using the GCC dataset consisting of crowd images. To enhance crowd analysis, domain adaptation, and supervised data are utilized. Compared to the conventional approach the domain adaptation reduces the tedious labeling efforts. However, this work does not focus on improving domain-adaptive crowd analysis or the generation of related data. Qingg et al., [12] have developed an algorithm Count Forest by combining Convolutional Neural Networks (CNN) and Deep Regression. The merit is that algorithm provides high accuracy with precision and real time analysis. The adaptive scene transformation is not implemented in this work to improve accuracy. Jian et al., [13] have designed a decoupled two-stage method to address challenges in crowd counting, localization, and artificial augmentation. The Decoupled two-stage crowd counting consists of Probability Map Regression (PMR) and Count Map Regression (CMR). The probability map regression needs to be implemented in this work. Reem et al., [14] have presented a crowd counting architecture that incorporates Bayesian Loss (BL). This approach involves point instruction and the BL loss operation, which encompasses the density map, the loss function, and the network structure.

Guoshuai et al., [15] have implemented Span Architecture (SA) for addressing large-scale variations in crowd images, focusing on the construction of inter-range characteristics using deep CNNs. The model exhibits good performance in dense and medium-density scenes, but the accuracy of counting in sparse scenes needs to be improved. Saqib et al., [16] have proposed a Motion-Guided Filter (MGF) to enhance the overall detection accuracy using spatial and temporal information. The merits are that accurate monitoring and evaluation are integrated for enhanced crowd scene interpretation. Wang et al., [17] have examined fine-grained crowd counting to explore low-level behaviour of individuals like standing/sitting, towards or away, and violent or nonviolent behaviour. The overall counting performance across all scenarios has been improved but the classification is limited only to four categories.

Wan et al., [18] have proposed SGANet, combining the Inception module, Inception-v3, and curriculum loss to address the scale variance in crowd images, while achieving a high counting accuracy. Zhou et al., [19] have designed a DPDNet, an RGB-D crowd counting employing density map guided detection to increase detection. Kong et al., [20] have presented a cascaded crowd counting that employs an attention mechanism and automatic scale-adaptive approach, to achieve efficient crowd counting. Sindagi et al., [21] have proposed a crowd counting technique by using a residual error prediction and a confusion judgment grading system. Wang et al., [22] have employed cross-dimensional urban forecasts and domain adaptation, in designing ST-DAAN, a transfer learning framework, which is used to estimate non-rural crowd counts. Gao et al., [23] have employed GAN with multilevel feature aware transformation and structured density map alignment for domain-invariant crowd counting. Tian et al., [24] have proposed PaDNet, a pan-density crowd counting integrating Density-Aware Network, Feature Enhancement Layer, and Feature Fusion Network, to achieve robust detection in varying density crowd scenarios.

TABLE I: LITERATURE SURVEY

Authors/ Year/ Model/ Con-Cept	Algorithm/ Implementa-tion/ Dataset	Performance/ Advantages	Future Directions/ Disadvantages
YuanZhou *et al.,* [1], Multi Scale Generative Adversarial Network(MS-GAN), 2020	Generative adversarial network	The density map is refined by Adversarial training	Crowd estimation networks like MCNN, and CSRNet are not experimented as Generator
Rongyong *et al.,* [2], Crowd stability analysis model (CSAM), 2021	Improved MCNN	Precise crowd stability detec-Tion	Monitoring factors like velocity and panic is not explored in pedestrian
Gaewei *et al.,* [5], Cross-scale convolution spatial genera-tive adversarial network (CS-GAN), 2020	GAN-based architecture	Enhance counting accuracy	Crowd density and crowd path move-ments still need to be addressed.
Xinyue *et al.,* [6], Density awareness and background segmentation, 2021	Background segmentation and density classification	Results are better with multi-task learning methods	Dense dataset performance is not ex-plored.
Weixing *et al.,* [7], Pyramid dilated deep convolutional neural network(PDD CNN), 2022	Dot-Annotation for den-sity map	Prediction of crowd count is Better	UCFF CC 50 are not trained for accu-rate crowd count prediction.
Jian *et al.*,[13], Decoupled two-stage crowd counting 2021	The Probability Map Regression (PMR) and Count Map Regression (CMR) are sub- modules.	Decoupled two-stage count-ing (D2CNet) is enhanced to provide a better output model	The probability map regression needs to be implemented.
Wan *et al.,* [17], Fine-Grained Counting, 2021	Density and segmenta-tion module	The overall counting perfor-mance across all scenarios has been improved.	The crowd is classified into four cate-gories only in this work
Proposed Model, CCSA, 2023	Unet Generative adver-sarial network	Preserved data during density map generation and build ac-curate maps	The model to be experimented with video datasets

Zhou et al., [25] have developed a linear feedback control system by incorporating a semi-supervised classifier for saliency detection. The model iteratively optimizes saliency maps using multiple cues and image features. Wang et al., [26] have proposed a congestion detection technique using a deep network in Vehicular Management, with multiple fusion networks. Xu et al., [27] have introduced the Depth Information Guided Crowd Counting (DigCrowd) method. This approach focuses on analyzing Extended Depth of Field (EDOF) scenes by effectively mapping crowd density and counting individuals in areas close to the viewer. Alashban et al., [28] have employed Single-Convolutional Neural Network (S-CNN3) with three Layers, for estimating crowd density, with high accuracy and efficiency. Zhang et al., [29] have achieved crowd counting accuracy in dense scenarios by utilizing a fully Convolutional Neural Network (CNN) and a Peak Confidence Map (PCM). Sharma et al., [30] have explored CNN-based architecture by integrating a scale-aware attention module and motion map-based features for analyzing crowd density and behavior. Jing et al., [31] have improved the YOLOX-nano model by integrating a Ghost Module to enhance lightweight vehicle detection. Cao et al., [32] have proposed YOLOv4 for enhanced traffic sign detection which surpasses existing object detection algorithms. Qingrong et al., [33] have presented the TCN-LSTM model to capture the patterns in traffic flow. The advantage of the model lies in its forecasting abilities, however, its effectiveness in the context of traffic collisions has not been explored in this work. Bai et al., [34] have designed the DUCAF-Net for drone image object detection with enhanced accuracy but have not explored its real-time processing capabilities.

III. ARCHITECTURE

A. Background

The background of the work, problem definition, and CCSA architectures are explained in this section.

Zhou et al., [1] have designed the architecture Multi Scale Generative Adversarial Network (MSGAN) to address the challenges of occlusion, perspective distortion, and visual similarity between pedestrian and background elements. MSGAN is designed in two phases namely (i) Generator and (ii) Discriminator. The generator is designed to build density maps for crowd images. It exhibits scale variations by combining, global and local features from multiple receptive fields and hierarchical convolution layers. Multicolumn CNN (MCNN) serves as a baseline for the generator to extract multiscale features with large-scale variation. The merit is that the generator detects individuals at different scales within the crowd.

The adversarial network serves as the baseline for the discriminator which obtains the density maps and its corresponding crowd images from the generator. The generator and discriminator in a GAN engage in a min-max game, as outlined in game theory. Additionally, the discriminator employs an adversarial network to distinguish between real and fake images, as demonstrated in Equation (1, 2, 3).

$$E_{(I_C,I_M)\sim P_{\text{data}}(I_C,I_M)}\left[\log D(I_C,I_M)\right] = A \quad (1)$$
$$E_{I_C\sim P_{\text{data}}(I_C)}\left[\log\left(1-D(I_C,G(I_C))\right)\right] = B \quad (2)$$
$$\boldsymbol{min}_G\ \boldsymbol{max}_D\ V(D,G) = A + B \quad (3)$$

In this min and max game, the generator (G) is trained with crowd images (IC) to produce the high-quality crowd density maps. On the other hand, the discriminator (D) utilizes the ground truth information to distinguish between real density maps (IM) and the density maps generated by G (G(IC). The main objective is to train G to produce realistic and accurate density representations so that D cannot differentiate between the generated and the actual density maps.

B. Problem Definition

The proposed Crowd Counting Stability Analysis (CCSA) architecture aims to accurately estimate crowd counts and detect potential instabilities by utilizing spatial data, such as density maps.

C. Objectives

(i) To develop the Crowd Counting Stability Analyzer (CCSA) that combines crowd counting and crowd stability in density maps.
(ii) To design the crowd-counting by integrating Uskip Connection, Unet Generator and a Discriminator, where Uskip Connections preserve spatial information during image segmentation.
(iii) To construct a crowd stability analyzer to classify crowd count into five categories namely very low, low, moderate, high, and very high based on density values.

D. CCSA Architecture

The CCSA architecture is composed of UGAN and CNN. The UGAN is developed into three phases namely (i) Uskip Connection, (ii) Unet Generator, (iii) Discriminator, and CNN is used in Crowd Stability (CS) as shown in Fig. 1. The crowd counting generally focuses on generating density maps and estimating the number of individuals in a crowd. In this proposed architecture additionally, a new model named Crowd Stability (CS) has been integrated to enhance this process. The CS model utilizes a density map to categorize them into five categories very low, low, moderate, high, and very high based on density values.

1) **Uskip Connection**: The Uskip Connection is designed with two passes viz., forward and backward pass. These passes, comprising three distinct blocks, enable the flow of information between the downsampling and upsampling operations. The forward pass is organized as three blocks the down block, the submodule block, and the up block. The down block performs a downsampling operation on the input. The submodule block forms the inner blocks which recursively contain its forward pass blocks. The up block implements an upsampling operation on the concatenated input. The forward pass achieves an upsampling operation on the concatenated input. It captures the context, and the spatial information and also enables the learned features that are specific to the segmentation task.

The backward pass is also organized as three blocks : up block, the submodule block, and the down block. The up block achieves a backward upsampling operation on the input. The submodule block recursively contains its backward pass blocks. The down block implements a backward downsampling operation on the concatenated input.

2) **Unet Generator**: The architecture of the Unet Generator, utilized for image segmentation tasks is illustrated in Table II. The components of the Unet Generator are skip connection blocks, forward pass, and weights functions. The Uskip Connection blocks construct the unet structure [31] by combining the outputs of each block with the corresponding upsampling block. The forward pass describes the flow of data that converts the input tensor into the output tensor. The weight function initializes the weights of the convolutional and batch normalization layers. It is implemented with Pytorch and is used in many applications. The Unet Generator model enhances the density map and crowd counting accuracy. The Unet Generator Loss (gl) measures generator's effectiveness in fooling the discriminator through realistic images as shown in equation (4).

$$g_l = \frac{1}{m}\sum_{i=1}^{m}\left(\log\left(1-D\left(G\left(Z^i\right)\right)\right)\right) \quad (4)$$

where m denotes the number of samples, Z i denotes the random noise vector sampled from a prior distribution D(G(Zi)) represents the discriminator output, provided with a density map constructed by the generator.

3) **Discriminator**: The discriminator model consists of four convolutional layers, one fully connected layer, 4*4 kernel, stride 2, leaky or sigmoid activations, and batch normalization as shown in Table III. In the convolution layer, LeakyReLU is used as the activation function, while the sigmoid function is employed in the fully connected layer. The two data instances, namely real and fake are used to train the discriminator. In fake data instances, the density map is obtained from the generator, and in real data instances, a density map is obtained from real crowd images. The discriminator loss (dl) is used to classify the density map as real or fake. It is evaluated as given in equation (5) where Xi represents real crowd samples from the training dataset. The logarithm of the discriminator's prediction for real data D(X(i)), enables the discriminator to correctly classify real data as real. The logarithm of 1 - D(G(Zi)), enables the discriminator to classify generated data as fake (close to 0).

$$d_l = \frac{1}{m}\sum_{i=1}^{m}\log(D\left(X^i\right) + \log\left(1 - D\left(G\left(Z^i\right)\right)\right) \quad (5)$$

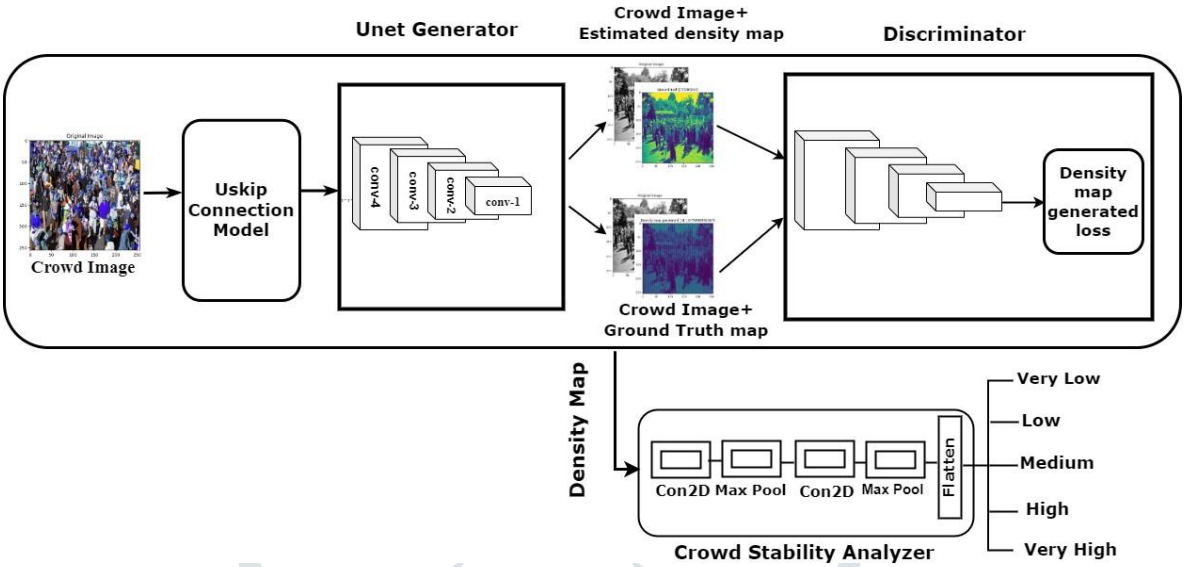

FIG. 1: ARCHITECTURE OF CROWD COUNTING AND STABILITY ANALYZER

TABLE II: Unet Generative Architecture Details

Layer	Input Size	Output Size	Details
Downconv1	$H \times W \times C_{\text{in}}$	$\frac{H}{2} \times \frac{W}{2} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Downconv2	$\frac{H}{2} \times \frac{W}{2} \times 64$	$\frac{H}{4} \times \frac{W}{4} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Downconv3	$\frac{H}{4} \times \frac{W}{4} \times 64$	$\frac{H}{8} \times \frac{W}{8} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Downconv4	$\frac{H}{8} \times \frac{W}{8} \times 64$	$\frac{H}{16} \times \frac{W}{16} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Downconv5	$\frac{H}{16} \times \frac{W}{16} \times 64$	$\frac{H}{32} \times \frac{W}{32} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Downconv6	$\frac{H}{32} \times \frac{W}{32} \times 64$	$\frac{H}{64} \times \frac{W}{64} \times 64$	Conv2d (4x4, stride 2), LeakyReLU, Batch-Norm2d
Upconv5	$\frac{H}{32} \times \frac{W}{32} \times 64$	$\frac{H}{16} \times \frac{W}{16} \times 64$	ConvTranspose2d (4x4, stride 2), ReLU, Batch-Norm2d
Skip Connection	$\frac{H}{16} \times \frac{W}{16} \times 128$	$\frac{H}{16} \times \frac{W}{16} \times 128$	Concatenate features from Upconv5 and Down-conv5
Upconv4	$\frac{H}{16} \times \frac{W}{16} \times 128$	$\frac{H}{8} \times \frac{W}{8} \times 64$	ConvTranspose2d (4x4, stride 2), ReLU, Batch-Norm2d
Skip Connection	$\frac{H}{8} \times \frac{W}{8} \times 128$	$\frac{H}{8} \times \frac{W}{8} \times 128$	Concatenate features from Upconv4 and Down-conv4
Upconv3	$\frac{H}{8} \times \frac{W}{8} \times 128$	$\frac{H}{4} \times \frac{W}{4} \times 64$	ConvTranspose2d (4x4, stride 2), ReLU, Batch-Norm2d
Skip Connection	$\frac{H}{4} \times \frac{W}{4} \times 128$	$\frac{H}{4} \times \frac{W}{4} \times 128$	Concatenate features from Upconv3 and Down-conv3
Upconv2	$\frac{H}{4} \times \frac{W}{4} \times 128$	$\frac{H}{2} \times \frac{W}{2} \times 64$	ConvTranspose2d (4x4, stride 2), ReLU, Batch-Norm2d
Skip Connection	$\frac{H}{2} \times \frac{W}{2} \times 128$	$\frac{H}{2} \times \frac{W}{2} \times 128$	Concatenate features from Upconv2 and Down-conv2
Upconv1	$\frac{H}{2} \times \frac{W}{2} \times 128$	$H \times W \times C_{\text{out}}$	ConvTranspose2d (4x4, stride 2), ReLU, Batch-Norm2d
Skip Connection	$H \times W \times 128$	$H \times W \times C_{\text{out}}$	Concatenate features from Upconv1 and Down-conv1

4) **CCN: Crowd Stability (CS):** A Convolutional Neural Network (CNN) is defined as a deep learning model designed with specialized layers, such as convolution and pooling, to 4) CCN: Crowd Stability (CS): A Convolutional Neural Network (CNN) is defined as a deep learning model designed with specialized layers, such as convolution and pooling, to learn automatically and extract essential spatial patterns from the images. CS enables accurate estimation of crowd density without

the dependency on hand-crafted features. Designed using the sequential API from Keras, CS comprises two convolutional layers followed by a max-pooling layer, a flattened layer, and a dense layer with 5 neurons and a softmax activation function. In Table IV, the CS layers summary of a density map to predict the crowd category is shown.

TABLE III: Discriminator Architecture Details

Block	Kernel	Num	Stride	Activation
Conv-1	8*8	64	2	LeakyReLU
Conv-2	4*8	128	2	LeakyReLU
Conv-3	2*4	256	2	LeakyReLU
Conv-4	1*2	512	2	LeakyReLU
FC-5	-	1	-	Sigmoid

TABLE IV: CNN Model

Layer (Type)	Output Shape	Param #
Conv2D	(None, 222, 222, 64)	640
MaxPooling2D	(None, 111, 111, 64)	0
Conv2D	(None, 109, 109, 32)	18464
MaxPooling2D	(None, 54, 54, 32)	0
Flatten	(None, 93312)	0
Dense	(None, 5)	466565
Total params:		485,669
Trainable params:		485,669
Non-trainable params:		0

Algorithm 1: UGAN: Unet Generative Adversarial Network

```
function Uskip Connection(low_features, high_features)
    Input : Lower-level feature map low_features,
      higher-level feature map high_features
    Output: Fused feature map fused_features
    Concatenate(low_features, high_features)
    Perform Convolution, Batch Normalization, ReLU
    return fused_features (X) of the Image
function Unet Generator(X)
    Input: Fused_features X
    Output: Generated data Y_fake
    Downsampling Path(fused_features (X))
    Y_downsampled = DownsamplingOperation(X)
    Y_up1 =
     UpsamplingOperation1(Y_downsampled,
     Y_downsampled)
    Y_up2 = UpsamplingOperation2(Y_up1,
     Y_up1)
    Y_fake = FinalGenerator
     Output(Y_downsampled, Y_up1, Y_up2)
    return Y_fake;
function Discriminator(Y, Y_fake)
    Input: Real data Y, fake data Y_fake
    Output: Probability scores scores_real,
     scores_fake
    Apply Convolution, Batch Normalization,
     LeakyReLU
    Flatten, FullyConnected
    return scores_real, scores_fake
function TrainingLoop()
    Initialize Unet Generator G, Discriminator D, loss
     functions, optimizers, hyperparameters
    Set Unet Generator G to training mode
    Set Discriminator D to training mode
    for epoch in range(epochs) do
        for X, Y in train_loader do
            Y_fake = Unet Generator(X)
            scores_real, scores_fake =
             Discriminator(Y, Y_fake)
            Update discriminator using scores_real,
             scores_fake
            Update generator using Y_fake, Y
        Save model checkpoints, perform evaluations if
         needed
    Training completed
```

Algorithm 2: CNN: Crowd Stability using Convolutional Neural Network

```
begin Crowd Stability(Input shape:
  img_height = 224, img_width = 224, channels = 1);
Input: Density map image
Output: Crowd category based on count
Input data to match the required shape for the CNN
  model
Retrieve the crowd count category from the CNN model
Classify the crowd category based on the count of the
  density map:
if (crowd count (25, 25 to 50, 50 to 75, 75 to 100)) then
    Assign the crowd category as (Very Low, Low,
      Moderate, High);
end
else
    Assign the crowd category as Very High
    Return the crowd category
end
```

IV. CROWD COUNT STABILITY ANALYZER

A. Algorithm 1: UGAN: Unet Generative Adversarial Network

Algorithm 1 and CCSA flowchart in Fig. 2 explains a Unet Generative Adversarial Network (UGAN). The UGAN consists of (i) Uskip Connection, (ii) Unet Generator, (iii) Discriminator. The Uskip Connection concatenates lower-level and higher-level feature maps with convolution, batch normalization, and ReLU activation. The functions generate the combined features of a given image to aid in feature extraction and integration.

The Unet Generator constructs an image with a downsampling to capture features and an upsampling to build the generated image. It also obtains the combined image feature from Uskip Connection which is used in training to determine a real or fake output image after comparing it with its database. The Discriminator evaluates the authenticity of real and generated images using convolution, batch normalization, leaky ReLU activation, flattening, and fully connected layers. The training function iteratively updates the Unet generator and discriminator using adversarial training.

B. Algorithm 2: CNN: Crowd Stability using Convolutional Neural Network

Algorithm 2 explains a Convolutional Neural Network (CNN) for crowd stability. The crowd stability constructs the sequential CNN model by including two Conv2D layers with relu activation, two MaxPooling2D layers with a stride of 2, and one Dense layer with softmax activation. The CS module predicts crowd count categories using CNN for every image in the dataset. The crowd category is calculated from the predicted count where a value up to 25 corresponds to very low, 25 to 50 corresponds to low, 50 to 75 corresponds to moderate, 75 to 100 corresponds to high, and other higher value indicates a very high crowd count.

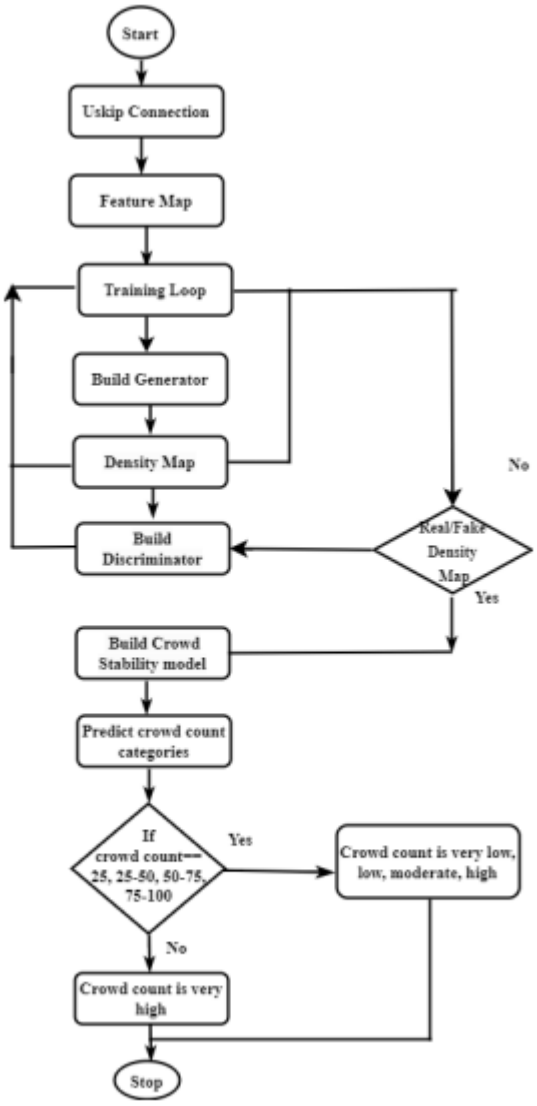

Fig. 2: Crowd Count Stability Analyzer (CCSA) Flowchart

Fig. 3: Crowd Street

For example, crowd counting can be implemented to analyze crowd density in busy urban areas such as crowded streets as shown in Fig. 3.

(i) Input: The CCSA algorithm constructs a density map to find the count of crowd images which is explained in Algorithm 1 and flowchart (Fig. 2).

(ii) Crowd Count Prediction: The CCSA algorithm utilizes UGAN to predict crowd counts on density maps. The UGAN consists of a generative network that builds a density map while the discriminator determines the authenticity of the generated density map.

(iii) Output: From the generated density map, the determined crowd count is categorized into very low, low, moderate, high, and very high.

C. Evaluation Metrics

The Mean Squared Error: The Mean Squared Error is defined as the mean of the squared differences among the predicted values, denoted as yˆ, and the actual values, denoted as y as shown in equation (6).

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \qquad (6)$$

The Mean Absolute Error: The Mean Absolute Error is defined as the average of the absolute differences among the predicted values, denoted as yˆ, and the actual values, denoted as y as shown in equation (7).

$$\mathbf{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \qquad (7)$$

The CCSA architecture is evaluated with MSE and MAE parameters.

Fig. 4: Sample of Four Datasets

Dataset	Scenes	Images	FPS
Shanghaitech-B	716	716	Images
Shanghaitech-A	1198	1198	Images
UCF_CC_50	Diverse	50	varies
DataC	48	48	varies

Fig. 5: Summarization of Four Datasets

Fig. 6: ShanghaiTech, UCF CC 50, DataC Dataset Samples and Details

D. Performance Metrics: Datasets

The experiments are conducted on four datasets namely (i) ShanghaiTech-A Dataset, ii) ShanghaiTech-B Dataset, iii) UCF CC 50 datasets, and iv) DataC.

(i) ShanghaiTech-A Dataset The ShanghaiTech is a sizable crowd-counting dataset, having 1198 marked-up pictures of 330,165 individuals. The dataset is partitioned into two parts, ShanghaiTech A and ShanghaiTech B. ShanghaiTech A has 482 photos of which 182 are used for testing, while the remaining 300 are used for training. The advantage of this dataset is that it has the most annotated people of varied sizes.

(ii) ShanghaiTech-B Dataset The ShanghaiTech-B, a part B component of the Shanghaitech has been used to evaluate the experiment model. The 716 pictures are depicting 716 different scenarios, each representing a single scenario. In ShanghaiTech B, 400 images are used for training and three hundred sixteen images are for testing. The advantage of this dataset is that the coordinates of the head's center are shown in each image.

(iii) UCF CC 50 Dataset UCF CC 50 crowd dataset is built of extremely dense images. It has 50 images with 63,974 head annotations in total. The head counts vary from 94 to 4,543 per image. The advantage of the dataset is that the images are dense with large variances.

(iv) DataC Dataset The dataset DataC is constructed with 48 images of 12 distinct categories such as Games, Street, Pedestrian, Protest, Cafeteria, School, Transport, Urban Design, Worship Places, Market Place, Playground, and Malls from the internet. The merit of the dataset is to analyze diverse crowd categories, which is not explored in other datasets.

Fig. 4 shows a sample of four datasets and Fig. 5 displays a summary of four datasets where FPS denotes the number of frames. In these datasets, there are considerable differences in crowd densities, population distributions, and perspective distortions, and hence these datasets are used to evaluate the crowd counting technique. Fig. 6 displays ShanghaiTech, UCF CC 50, DataC Dataset samples and details.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

The CCSA architecture is implemented on NVDIS Geforce TitanX GPU equipped with 12 GB of memory. The Unet generator and discriminator are trained in an adversarial manner. In training the discriminator's parameters such as loss functions and optimizers are updated based on the comparison of scores between real and fake images, enhancing its ability to differentiate them. Simultaneously, the generator's parameters are also updated, driving it to generate fake images that successfully deceive the discriminator. The training concludes after specified epochs, yielding a generator capable of producing realistic images.

The CCSA architecture can be applied in various examples such as Transportation Hubs, Urban Planning, and Public Spaces Designing. The transportation hubs like airports, train stations, and bus terminals are shown in Fig. 7 where crowd counting can be used to assess passenger movement and density flow. The knowledge can be used to optimize transportation operations such as adding more gates, upgrading ticket systems, and reducing waiting times. It can also be used to develop more effective crowd management strategies, such as enhancing security, and safety measures. The Urban Planning and Public Spaces Design like malls, theatres, stadiums, and playgrounds are shown in Fig. 8, where crowd counting can be implemented to analyze crowd density. Understanding crowd density with patterns in Fig. 9 can provide better design decisions, for public spaces such as Community Parks, and Urban Green space designs to meet the diverse needs of the population.

VI. RESULTS AND DISCUSSIONS

Fig. 7: Transportation Hubs

Fig. 8: Urban Planning

Fig. 9: Crowd Analysis Examples

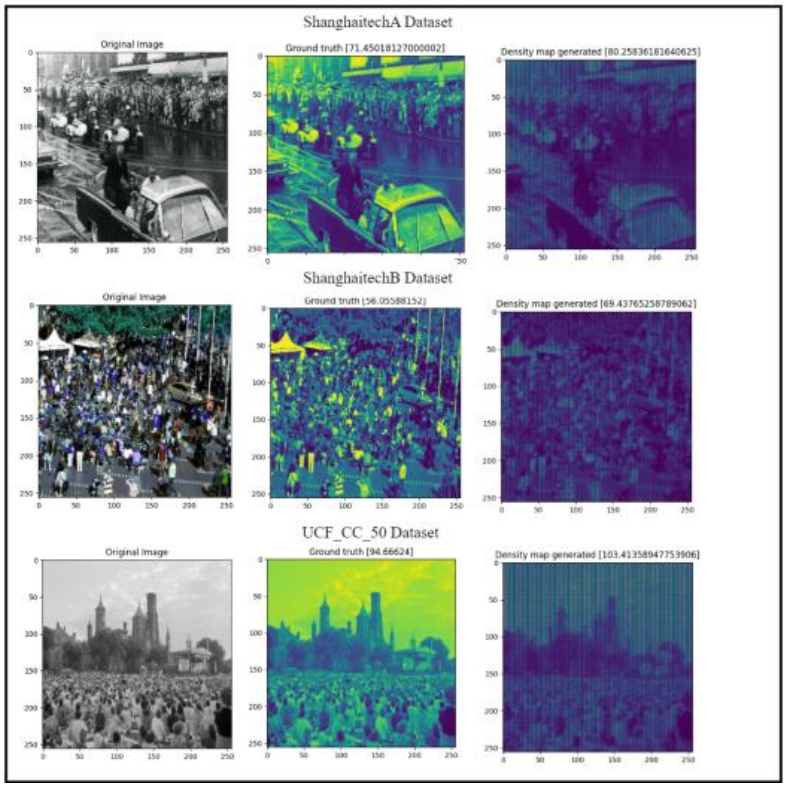

Fig. 10: Experimental results on ShanghaiTech A , B & UCF CC 50 Datasets

A. Discussion

The proposed model is examined on four crowd-counting datasets namely ShanghaiTech A, B, UCF CC 50, and DataC. In Fig. 10, the experimental results of Shanghaitech A, B, and UCF CC 50 datasets, displaying the original image, density map, and corresponding ground truth are shown. In Table V, the results of CCSA without CNN in Shanghaitech A with MAE 2.57, and MSE 11.57 and Shanghaitech B with MAE 4.73, MSE 24.87 and UCF CC 50 with MAE 9.3, and MSE 106.97 are illustrated. The results of the UCF CC 50 dataset present significant challenges in accurate crowd counting due to its dense nature. One potential solution is to generate more synthetic data, which can be used to enhance the training of the model. In Table VI, the results of CCSA with CNN for random images in Shanghaitech B with a predicted count of 171, ground truth of 179, and the classification as very high (above 100), Shanghai tech A with a predicted count of 334, ground truth of 336, and the classification asvery high, and UCF CC 50 with a predicted count of 73, ground truth of 69, and the classification as moderate is illustrated.

TABLE V: Comparison performance of Proposed with CNN model CCSA without CNN with MSGAN

Model	Dataset	MAE	MSE
CCSA without CNN	ShanghaiTech-B	4.73	24.87
CCSA without CNN	ShanghaiTech-A	2.57	11.57
CCSA without CNN	UCF_CC_50	9.3	106.97
MS-GAN *et al.*,[1]	ShanghaiTech-B	18.7	30.5
MS-GAN *et al.*,[1]	UCF_CC_50	345.7	418.3

TABLE VI: Comparison performance of Proposed model CCSA

Model	Dataset	Predicted_count	Ground Truth Count	Category
CCSA with CNN	ShanghaiTech-B	171	179	Very High
CCSA with CNN	ShanghaiTech-A	334	336	Very High
CCSA with CNN	UCF_CC_50	73	69	Moderate
CCSA with CNN	DataC	103	92	Very High
MS-GAN *et al.*,[1]	UCF_CC_50	483	469	nil

The model is evaluated by using two metrics MAE and MSE. The MSE is more appropriate to avoid large errors, whereas MAE is used to avoid bias towards errors. Compared with MS-GAN et al., [1] the experimental results presented in Tables V and VI demonstrate that the proposed model effectively estimates the crowd density in ShanghaiTech- A, B, and UCF CC 50 datasets. The model achieves greater accuracy, as UGAN, captures intricate details and maintains spatial information, which potentially enhances the generation of images.

The DataC dataset is built of 48 images from the internet consisting of 12 different categories such as Games, Street, Pedestrian, Protest, Cafeteria, School, Transport, Urban Design, Worship Places, Market Place, Playground, and Malls. Table VII shows the result of 48 images of DataC with the image number, type of crowd, predicted count, ground truth count, the difference between the both, loss in %, MAE, MSE, and finally the crowd category. Fig. 11 shows the density map of 48 images in DataC.

In Fig. 12 shows that the CCSA (MAE of 4.73 and MSE of 24.87) achieved lower errors compared to MS-GAN (MAE of 18.7 and MSE of 30.5) on the ShanghaiTech-B dataset. Fig. 13 compares MAE and MSE between CCSA without CNN and MS-GAN on UCF CC 50. CCSA demonstrates superior performance with MAE 9.3, and MSE 106.97, while MS-GAN shows higher errors with MAE 345.7, and MSE 418.3, highlighting CCSA's accuracy.

In Fig. 14 graph compares the predicted and ground truth counts for crowd data across different datasets using CCSA and MS-GAN methods. CCSA generally predicts counts ranging from 73 to 334 closer to the ground truth compared to MS-GAN, (only for UCF CC 50 dataset, ranging from 0 to 483, with ground truth counts ranging from 69 to 336). In Fig. 15 chart depicts the distribution of density maps across various types of crowds. It visually represents the percentage of total density contributed by each type of crowd, with the transport displaying a major distribution of 40%. In Fig. 16 chart displays the distribution of categories based on the type of crowd. It visually represents the proportion of each category (High, Moderate, Low, and Very High) within the dataset. Fig. 17 compares predicted and ground truth density values for various crowd types. Predicted densities range from 44 to 103, while ground truth densities range from 38 to 92, showing accuracy between predicted and actual values across different crowd types. The model accuracy graph as illustrated in Fig. 18 shows the performance of the crowd counting model on a set of 48 random images of DataC. To draw the model accuracy graph, the density map count and ground truth crowd count of 48 images are obtained. The matplotlib is utilized to draw the graph among the 48 images of DataC which shows the difference between the prediction count (green color) and ground truth count (black color) which are less than 5 % in 9 images, between 5-10 % in 31 images and above 10% in 8 images of the crowd.

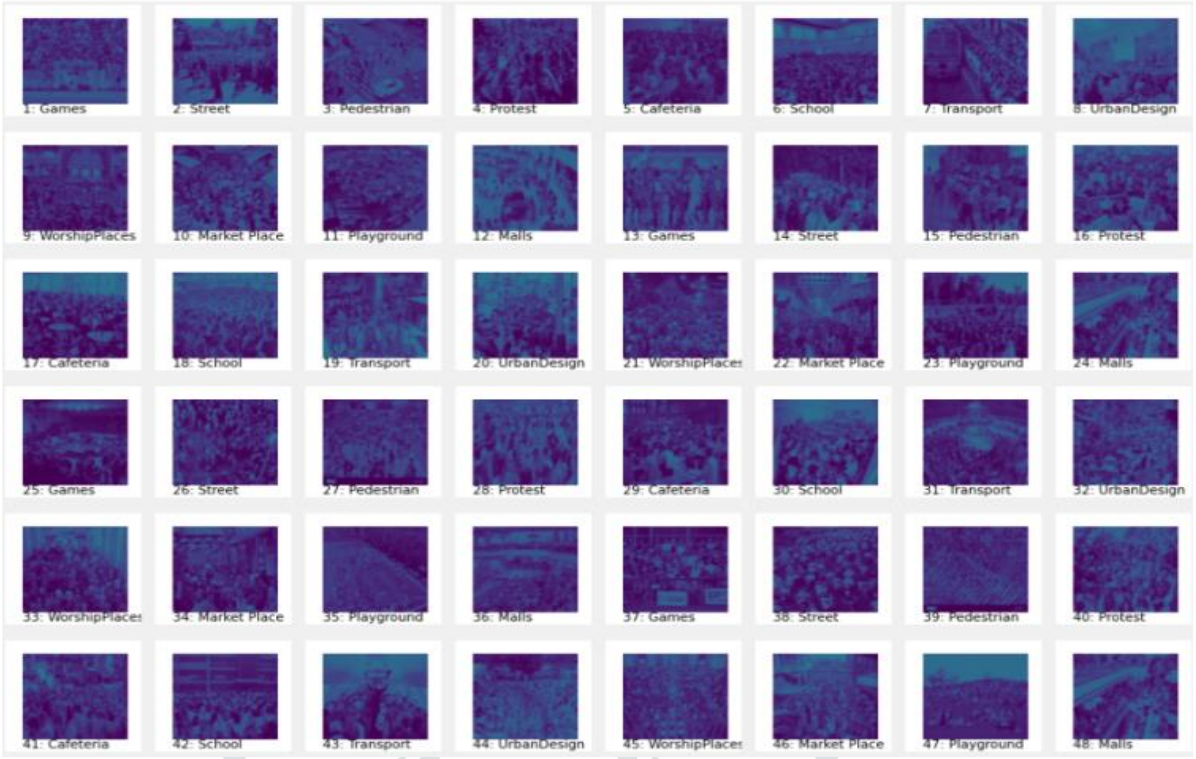

Fig. 11: Density Map of DataC 48 Images

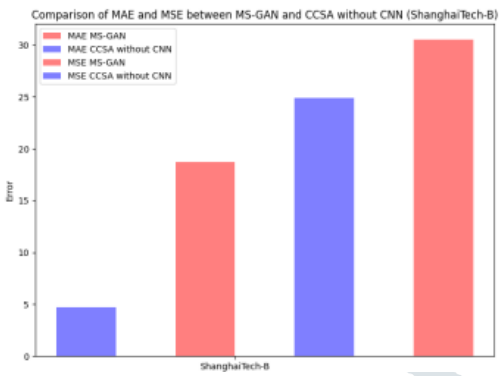

Fig. 12: MAE & MSE (between MSGAN CCSA without CNN on UCF CC 50)

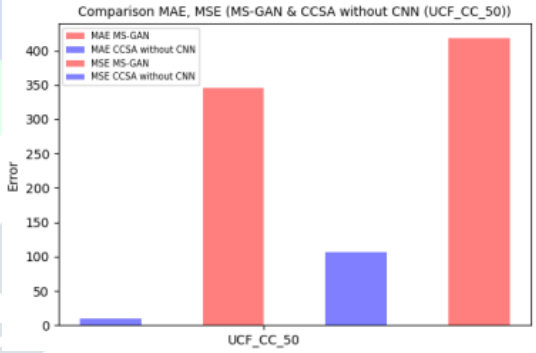

Fig. 13: MAE & MSE between (MSGAN & & CCSA without CNN on ShanghaiTech-B)

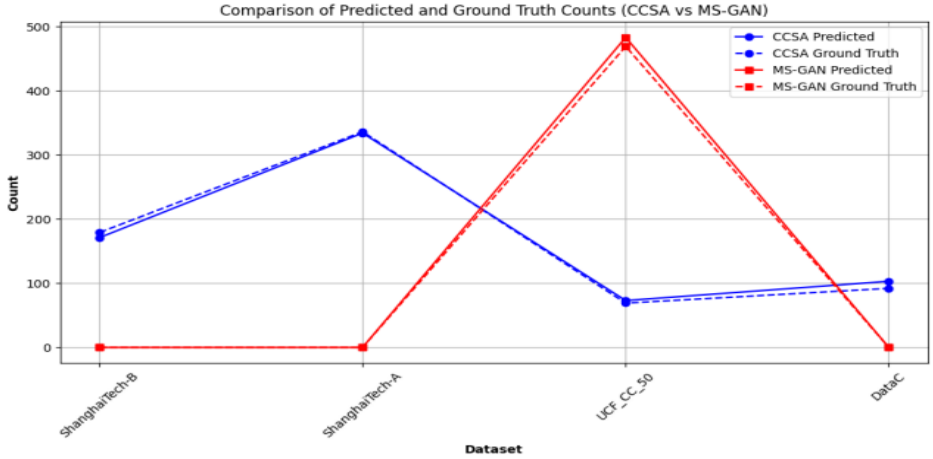

Fig. 14: Comparision of Predicted and GroundTruth Counts (CCSA vs MS-GAN)

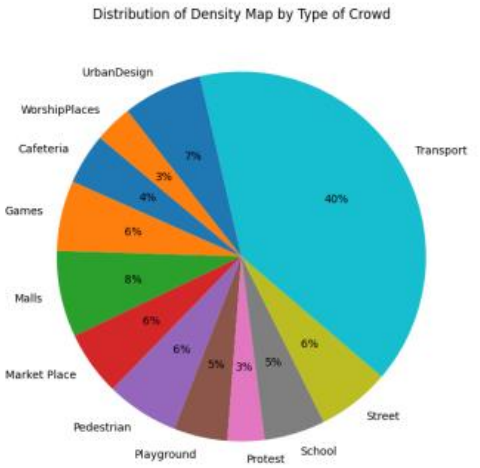

Fig. 15: Density Map Distribution with Crowd Type

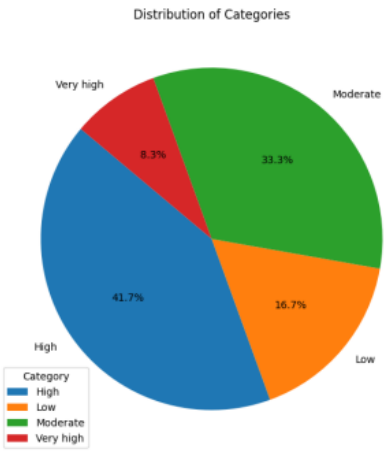

Fig. 16: Distribution of Categories

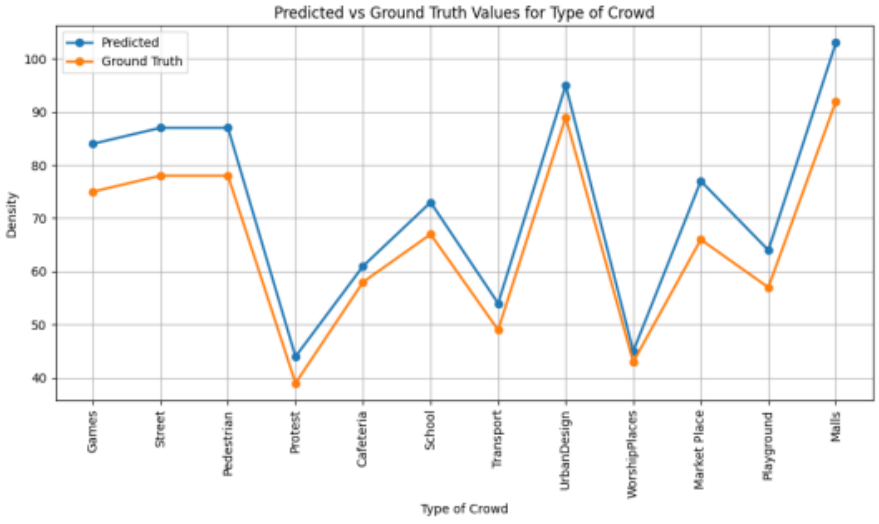

Fig. 17: Predicted vs Ground Truth Values for Crowd Type

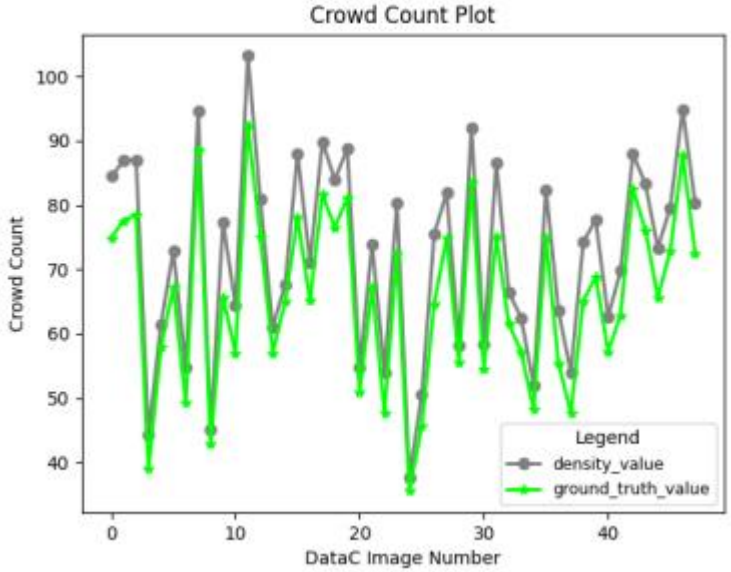

Fig. 18: Model Accuracy Graph for DataC

VII. CONCLUSION

The architecture crowd stability analyzer is composed of two models (i) UGAN (crowd counting) and (ii) CNN (crowd stability). The UGAN is composed of three phases namely (i) Uskip Connection, (ii) Unet Generator, (iii) Discriminator, and CNN are used in Crowd Stability (CS). The Unet Generator designed with Uskip GAN constructs the density maps while the discriminator refines using loss functions such as binary cross entropy. In crowd stability, the analyzer counts the sum of the density map and classifies them as very low, low, moderate, high, and very high. The experimental results prove that the proposed model CCSA compared to MSGAN et al.,[1] effectively generates the crowd density map of varied scenes in ShanghaiTech- A, B, UCF CC 50, and DataC datasets. It evaluates more precisely on MAE and MSE metrics due to the UGAN and CNN which derive finer details of the image with minimal loss. It also exhibits the best performance on varied screen image datasets. In addition to the above result, the 12 distinct crowd categories along with its crowd count details are explicitly shown with our newly built dataset DataC, a feature not explored in other datasets. For future enhancements, analyzing unstable crowd behavior and alerting security personnel for necessary actions could be considered.

REFERENCES

[1] Zhou, Y., Yang, J., Li, H., Cao, T. and Kung, S, "Adversarial Learning for Multiscale Crowd Counting under Complex Scenes," IEEE Transactions on Cybernetics, vol. 51, no. 11, pp. 5423-5432, 2020.
[2] Zhao, R., Dong, D., Wang, Y., Li, C., Ma, Y. and Enrıquez, V "Image-Based Crowd Stability Analysis using Improved MultiColumn Convolutional Neural Network," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 6, pp. 5480-5489, 2021.
[3] Khan, M., Menouar, H. and Hamila, R, "Revisiting Crowd Counting: State-of-the-art, Trends, and Future Perspectives," Image and Vision Computing, vol. 129, pp. 104597, 2022.
[4] Tripathi, G., Singh, K. and Vishwakarma, D, "Convolutional Neural Networks for Crowd Behavior Analysis: a Survey," The Visual Computer, vol. 35, no. 5, pp. 753-776, 2019.
[5] Zhang, G., Pan, Y., Zhang, L. and Tiong, R, "Cross-Scale Generative Adversarial Network for Crowd Density Estimation from Images," Engineering Applications of Artificial Intelligence, vol. 94, pp. 103777, 2020.
[6] Liu, X., Sang, J., Wu, W., Liu, K., Liu, Q. and Xia, X, "DensityAware and Background-Aware Network for Crowd Counting via Multi-Task Learning," Pattern Recognition Letters, vol. 150, pp. 221- 227, 2021.
[7] Wang, W., Liu, Q. and Wang, W, "Pyramid-Dilated Deep Convolutional Neural Network for Crowd Counting," Applied Intelligence, vol. 52, no. 2, pp. 1825-1837, 2022.
[8] Zhu, A., Zheng, Z., Huang, Y., Wang, T., Jin, J., Hu, F., Hua, G. and Snoussi, H, "CACrowdGAN: Cascaded Attentional Generative Adversarial Network for Crowd Counting," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 8090-8102, 2021.
[9] Zhou, Y., Huo, S., Xiang, W., Hou, C. and Kung, S, "Semi Supervised Salient Object Detection using a Linear Feedback Control System Model," IEEE Transactions on Cybernetics, vol. 49, no. 4, pp. 1173-1185, 2019.
[10] Ding, X., He, F., Lin, Z., Wang, Y., Guo, H. and Huang, Y, "Crowd Density Estimation using Fusion of Multi-Layer Features," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 4776-4787, 2020.
[11] Wang, Q., Gao, J., Lin, W. and Yuan, Y, "Pixel-Wise Crowd Understanding via Synthetic Data," International Journal of Computer Vision, vol. 129, no. 1, pp. 225-245, 2021.
[12] Ji, Q., Zhu, T. and Bao, D, "A Hybrid Model of Convolutional Neural Networks and Deep Regression Forests for Crowd Counting," Applied Intelligence, vol. 50, no. 9, pp. 2818-2832, 2020.
[13] Cheng, J., Xiong, H., Cao, Z. and Lu, H, "Decoupled Two Stage Crowd Counting and Beyond," IEEE Transactions on Image Processing, vol. 30, pp. 2862-2875, 2021.
[14] Alotaibi, R., Alzahrani, B., Wang, R., Alafif, T., Barnawi, A. and Hu, L, "Performance Comparison and Analysis for Large-Scale Crowd Counting based on Convolutional Neural Networks," IEEE Access, vol. 8, pp. 204425-204432, 2020.
[15] Wang, G., Zou, Y., Li, Z. and Yang, D, "SMCA-CNN: Learning a Semantic Mask and Cross-Scale Adaptive Feature for Robust Crowd Counting," IEEE Access, pp. 168495-168506, 2019.
[16] Saqib, M., Khan, S., Sharma, N., and Blumenstein, M, "Crowd Counting in Low-Resolution Crowded Scenes using Region-based Deep Convolutional Neural Networks," IEEE Access, vol. 7, pp. 35317-35329, 2019.
[17] Wan, J., Kumar, N. and Chan, A, "Fine Grained Crowd Counting," IEEE Transactions on Image Processing, vol. 30, pp. 2114- 2126, 2021.
[18] Wang, Q. and Breckon, T, "Crowd Counting via Segmentation Guided Attention Networks and Curriculum Loss," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 15233-15243, 2022.
[19] Zhou, J., Zhang, L., Du, J., Peng, X., Fang, Z., Xiao, Z. and Zhu, H, "Locality-Aware Crowd Counting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3602-3613, 2021.
[20] Kong, W., Li, H., Xing, G. and Zhao, F, "An Automatic Scale Adaptive Approach with Attention Mechanism-based Crowd Spatial Information for Crowd Counting," IEEE Access, vol. 7, pp. 66215-66225, 2019.
[21] Sindagi, V., Yasarla, R. and Patel, V, "JHU-CROWD++: Large Scale Crowd Counting Dataset and A Benchmark Method," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 5, pp. 2594-2609, 2022.
[22] Wang, S., Miao, H., Li, J. and Cao, J, "Spatio-Temporal knowledge Transfer for Urban Crowd Flow Prediction via Deep Attentive Adaptation Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pp. 4695-4705, 2021.
[23] Gao, J., Yuan, Y. and Wang, Q, "Feature-Aware Adaptation and Density Alignment for Crowd Counting in Video Surveillance," IEEE Transactions on Cybernetics, vol. 51, no. 10, pp. 4822-4833, 2020.
[24] Tian, Y., Lei, Y., Zhang, J. and Wang, J, "Padnet: Pan-density Crowd Counting," IEEE Transactions on Image Processing, vol. 29, pp. 2714-2727, 2019.
[25] Zhou, Y., Huo, S., Xiang, W., Hou, C. and Kung, S, "Semi Supervised Salient Object Detection using a Linear Feedback

Control System Model," IEEE Transactions on Cybernetics, vol. 49, no. 4, pp. 1173-1185, 2018.
[26] Wang, Q., Wan, J. and Li, X, "Robust Hierarchical Deep Learning for Vehicular Management," IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4148-4156, 2018.
[27] Xu, M., Ge, Z., Jiang, X., Cui, G., Lv, P., Zhou, B. and Xu, C, "Depth Information Guided Crowd Counting for Complex Crowd Scenes," Pattern Recognition Letters, vol. 125, issue. C, pp. 563-569, 2019.
[28] Alashban, A., Alsadan, A., Alhussainan, N. and Ouni, R, "Single Convolutional Neural Network with Three Layers Model for Crowd Density Estimation," IEEE Access, vol. 10, pp. 63823-63833, 2022.
[29] Zhang, J., Chen, S., Tian, S., Gong, W., Cai, G. and Wang, Y, "A Crowd Counting Framework Combining with Crowd Location," Journal of Advanced Transportation, pp. 1-14, 2021.
[30] Sharma, V., Mir, R. and Singh, C, "Scale-aware CNN for Crowd Density Estimation and Crowd Behavior Analysis," Computers and Electrical Engineering, vol. 106, pp. 108569, 2023.
[31] Jing X, and Tian Y, "Lightweight Vehicle Detection Based on Improved Yolox-Nano," IAENG International Journal of Computer Science, vol. 50, no. 1, 2023.
[32] Cao J, Li P, Zhang H, and Su G, "An Improved YOLOv4 Lightweight Traffic Sign Detection Algorithm," International Journal of Computer Science, vol. 50, no. 3, pp. 825-31, 2023.
[33] Wang Q, Chen X, Zhu C, Zhang K, He R and Fang J, "Short Term Traffic Flow Prediction Based on Spatiotemporal and Periodic Feature Fusion," IAENG Engineering Letters, vol. 32, no. 1, 2024.
[34] Bai Y, Li Z, Wu J, and Yu X, "DUCAF-Net: An Object Detection Method for UAV Imagery," Engineering Letters vol. 31, no. 4, 2023.

VIII. AUTHORS

B.Ganga is a research scholar in the Department of Computer Science and Engineering at the University of Visvesvaraya College of Engineering (UVCE), Bangalore University, Bengaluru, India. She received a B.E degree in Computer Science and Engineering from Karunya Institute of Technology, Tamilnadu, and an M.Tech in Software Engineering from M.S.Ramaiah Institute of Technology, Bangalore. She has 10 years of teaching experience. Her area of interest is Artificial intelligence, Deep learning, and data mining.

Dr. Lata B T is an Associate Professor in the Department of Computer Science and Engineering at the University of Visvesvaraya College of Engineering (UVCE), Bangalore University, Bengaluru, India. She obtained her B.E in Computer Science and Engineering from Karnataka University, Dharwad, and M.Tech degree in Computer Network Engineering from Visvesvaraya Technological University, Belgaum. Ph.D. degree in the area of Wireless Sensor Networks from Bangalore University. She is having 2 decades of teaching experience. Her research interest is in the area of Sensor Networks, IoT, Deep learning, Artificial intelligence and Image processing.

Dr. Venugopal K R, Former Vice-Chancellor, Bangalore University has served Bangalore University and UVCE for over the last five decades. He has 11 degrees with two Ph.Ds., one in Economics from Bangalore University and another in Computer Science Engineering from IIT Madras. He received his ME degree in Computer Science Engineering from the Indian Institute of Science, Bangalore. He has authored and edited 84 books, he has published more than 1200 Research Papers, he has a Google Scholar citations H-index of 39, and holds 40 patents. He has awarded Ph.D to 30 students, guided informally 150 Research Scholars, and supervised more than 800 Post Graduate dissertations in Computer Science and Engineering. He received the IEEE Fellow and ACM Distinguished Educator award from the USA for his outstanding contributions to the field of Computer Science Engineering. He was a Post Doctoral Research Scholar and visiting Professor at the University of Southern California, USA