RES MILITARIS Social Science Journal

PEOPLE COUNTING SYSTEM BASED ON HEAD DETECTION USING FASTER RCNN

Naveen Kumar[1], Sri Varsha Ambati[2], Chilukuri Tulasi[2], Jaligam Sushanth[2]

[2]UG Scholar, [1,2]Department of Computer Science and Engineering

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana.

ABSTRACT

The People Counting System is an advanced computer vision application designed for accurate real-time head detection and counting in crowded areas like shopping malls, transportation hubs, stadiums, and public gatherings. It addresses the growing demand for automated crowd monitoring in security, retail analytics, and crowd management. Traditional methods, such as motion detection or background subtraction, often fail in complex, crowded scenarios due to their lack of fine-grained object recognition and tracking, leading to imprecise and error-prone results. To overcome these limitations, our system employs the Faster R-CNN object detection model, which offers significant improvements in accuracy and efficiency by detecting and counting individual heads in real-time. This focus on head detection minimizes common errors such as double counting, which simpler methods frequently encounter. The system excels in identifying and localizing heads even in densely crowded scenes with occlusions or overlapping objects, ensuring minimal counting errors. Its real-time capability, enabled by Faster R-CNN, allows for continuous and instantaneous counting, providing an immediate response to changing crowd conditions. This makes the People Counting System a valuable tool for effective crowd management and decision-making in various applications, including occupancy management, security, and retail analytics, where precise counting is essential.

Keywords: People Counting, Computer Vision, Real-Time Detection, Faster R-CNN, Head Detection, Crowd Management.

1. INTRODUCTION

Counting people is one of the most important tasks in intelligent video surveillance systems and it has a wide range of applications and business value in many places, such as banks, railway stations, shopping malls, schools, etc. However, counting people in a crowded surveillance environment is a challenge task due to low resolution, occlusion, illumination change, imaging viewpoint variability, background clutter, etc. There are two main types of people detection methods: traditional and deep learning methods. Traditional learning methods extract features by the histogram of oriented gradients (HOG) and classify the features by a linear SVM classifier [1–3]. The advances in deep learning greatly bring single-image people detection to an unprecedented performance level. While many people detection algorithms have been designed for standard side-mounted cameras, the best performance to date has been achieved by deep-learning object detection algorithms [4]. Deep learning methods include single-stage approaches such as YOLO, and SSD. While two-stage approaches reach higher accuracy, single-stage approaches take up fewer computing resources and fast speed.

2. LITERATURE SURVEY

In the past decades, various methods of people counting have been proposed. These methods can be divided into three groups:

Counting by trajectory clustering

This kind of methods count people by adopting tracking techniques. Antonini and Thiran [5] applied a multilayer clustering technique to trajectories which are obtained from a tracking system. This method can reduce the bias between the number of tracks and the real number of persons. Topkaya et al. [6] improved the clustering method using a generic person detector. This method fused different types of features, including color, spatial and temporal features into clustering. For some applications, these approaches can get comparative good performance. However, they are usually time-consuming and computation resource-consuming since they tightly depend on tracking techniques. Besides, the performance would degrade a lot when applying these methods to motion imaging platforms since this situation usually makes the track algorithms unstable.

Counting by regression

This type of methods adopt regression-based techniques to learn a mapping between low-level features and the people count in a scene. Lempitsky and Zisserman [7] introduced an object counting method through pixel-level object density map regression whose integral over any image region gives the count of objects within that region and learning to infer such density can be formulated as a minimization of a regularized risk quadratic cost function. Chan and Vasconcelos [8] extracted a set of holistic low-level features from each segmented region, and learned a function that maps features into the number of people with the Bayesian regression. Morerio et al. [9] proposed a global group-based approach to estimate the count of people relying on accurate camera calibration. Idrees et al. [10] estimated the number of individuals in extremely dense crowds based on multi-source and multi-scale features and regress using SVR from images. Zhang et al. [11] alternatively trained a CNN regression model for crowd counting with two related learning objectives, namely the crowd density and the crowd count. The obvious advantage of these methods is that they can circumvent explicit object segmentation and detection. However, their performance is heavily influenced by the occlusion and perspective effect. Besides, these methods just provide the information of the people count, and fail to locate the individuals. Actually, the location information of individuals is usually very important for video surveillance systems. For example, we can use this information to automatically obtain the spatial distribution of people in a scene.

Counting by detection

The basic idea of this type of methods is to design a detector to detect each individual to obtain people count. The adopted detection methods usually include body detection [12], shoulders detection [12, 13], and head detection [15]. Comparatively speaking, head detection methods usually outperform the other two types of methods in crowded applications with heavy occlusion since only the heads of many persons may be visible in this situation. In order to effectively detect heads, many methods have been presented, such as skeleton graph [16], the head template matching and learning-based methods [17], [18]. These methods can be applied to some specific scenes and would fail for some real surveillance applications.

3. PROPOSED SYSTEM

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

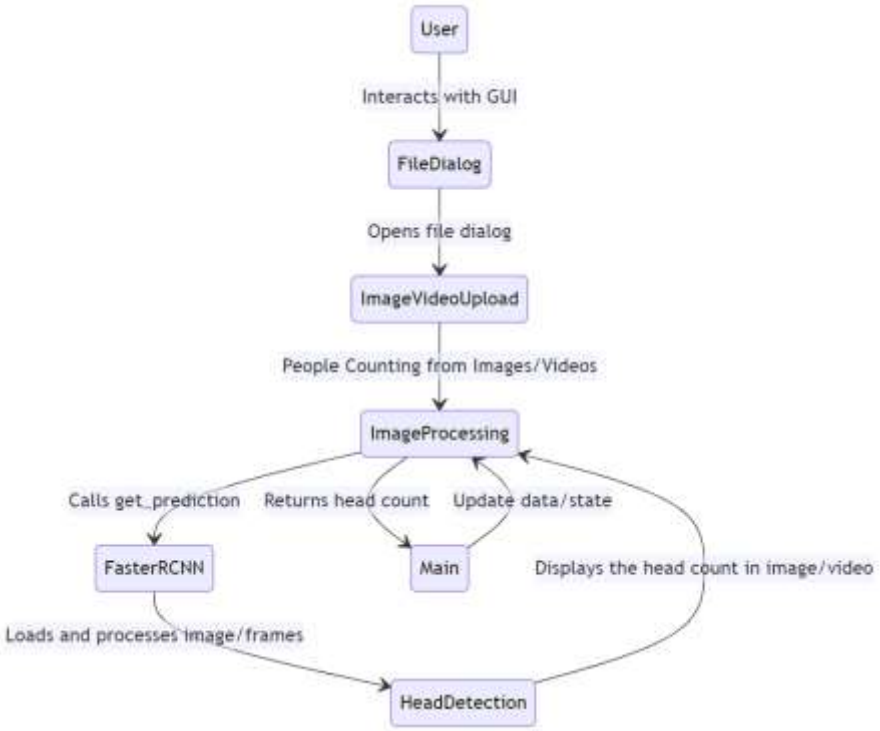

Figure 1. Block Diagram of Proposed System.

Convolutional Neural Network

Convolutional Neural Networks (CNNs) represent a pivotal advancement in deep learning, particularly well-suited for tasks involving structured grid-like data, such as images. Their architecture and design draw inspiration from biological processes, specifically how the visual cortex processes visual information. CNNs have revolutionized fields like computer vision, enabling breakthroughs in image classification, object detection, and segmentation, among others.

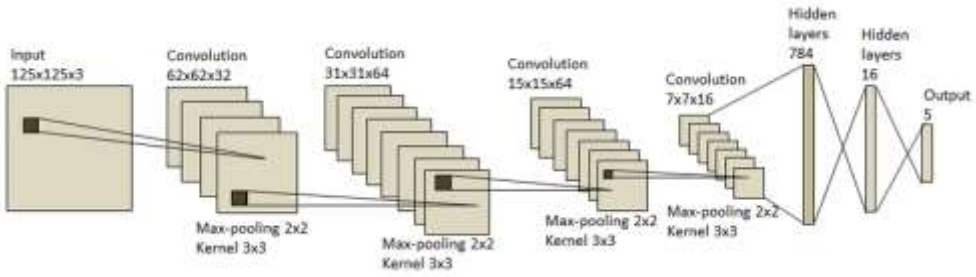

Figure. 2: Architectural Diagram of RCNN Model.

Components of a CNN

1. Input Layer

The input to a CNN is typically an image represented as a grid of pixel values. For a color image, this might be a 3D array (height x width x 3) where 3 represents the RGB channels.

2. Convolutional Layer

The convolutional layer applies filters (kernels) to the input image. Each filter performs convolution operation, which involves sliding the filter matrix over the input image and computing dot products to produce a feature map. Mathematically, for a 2D input image I and a filter K, the convolution operation at a specific position (i,j) is given by:

$$(I*K)(i,j) = \sum_m \sum_n I(m,n) \cdot K(i-m, j-n)$$

where I(m,n) represents the pixel intensity at position (m,n) in the input image.

3. Activation Function

After convolution, an activation function like ReLU (Rectified Linear Unit) is applied element-wise to introduce non-linearity into the model:

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

ReLU helps CNNs learn complex patterns and improves convergence during training.

4. Pooling Layer

Pooling layers (e.g., max pooling or average pooling) follow convolutional layers to reduce spatial dimensions of the feature maps, thereby decreasing computational complexity and controlling overfitting.

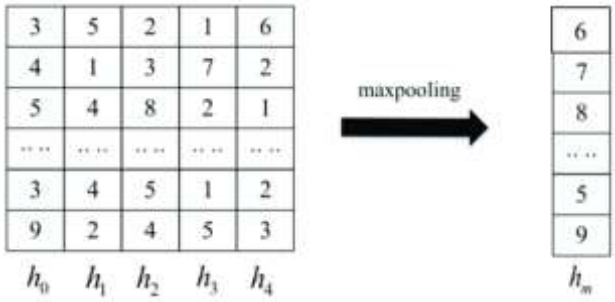

Figure. 3: MaxPolling layer.

For max pooling, the operation extracts the maximum value from each patch of the feature map:

$$MaxPooling(I)(i,j)=max_{m,n} I(m,n)$$

5. Flattening

After several convolutional and pooling layers, the resulting feature maps are flattened into a vector. This step converts the 2D or 3D matrix representation into a 1D vector that can be input to fully connected layers.

6. Fully Connected (Dense) Layers

These layers process the flattened features. Each neuron in a fully connected layer is connected to every neuron in the previous layer. Mathematically, for a layer l with output vector $h^{(l)}$ and weights $W^{(l)}$ the output $z^{(l+1)}$ of the next layer is computed as:

$$z^{(l+1)} = W^{(l)} h^{(l)} + b^{(l)}$$

where $b^{(l)}$ is the bias vector.

7. Output Layer

The final layer of the CNN produces the network's output. For classification tasks, this typically involves applying a softmax activation function to the output of the last fully connected layer. Softmax normalizes the outputs into probabilities across multiple classes, enabling the model to make predictions.

8. Loss Function

A loss function quantifies how well the network's predictions match the actual target values during training. For classification tasks, cross-entropy loss is commonly used.

9. Optimization

To minimize the loss function and improve the network's performance, optimization algorithms like stochastic gradient descent (SGD) or its variants (e.g., Adam, RMSProp) are employed. These algorithms adjust the weights and biases of the network iteratively based on the gradients computed during backpropagation.

10. Backpropagation

Backpropagation is a key mechanism in CNNs for computing the gradients of the loss function with respect to each parameter in the network. It propagates these gradients backward through the network, allowing efficient updates of the model's weights and biases.

4. RESULTS AND DESCRIPTION

Figure 4 depicts a visual representation of the GUI of the people counting system described in the project. It showcases how the application's main window looks to the user. This includes buttons, labels, and other graphical elements that allow users to interact with the system. It provides an overview of the user interface's design and layout. Figure 5 consists of sample images that serve as test cases for the people counting system. These images are used to demonstrate how the system works when processing static images. The sample test images contain scenes with people, and it helps illustrate the variety of images that the system can handle. Figure 6 shows the results of applying the Faster R-CNN model to the sample test images. It illustrates how the system detects and counts human heads within these test images. It displays the corresponding head count based on the model's predictions. Figure 7 demonstrates the application of the proposed model to sample videos. It displays a sample frame from a video with human subjects and show the real-time head detection and counting as the video plays. This illustrates the dynamic nature of the system when processing videos and how it provides continuous head count updates.

Figure 4: Illustrating the main GUI application of proposed people counting system based on head detection.

Figure 5: Sample test images.

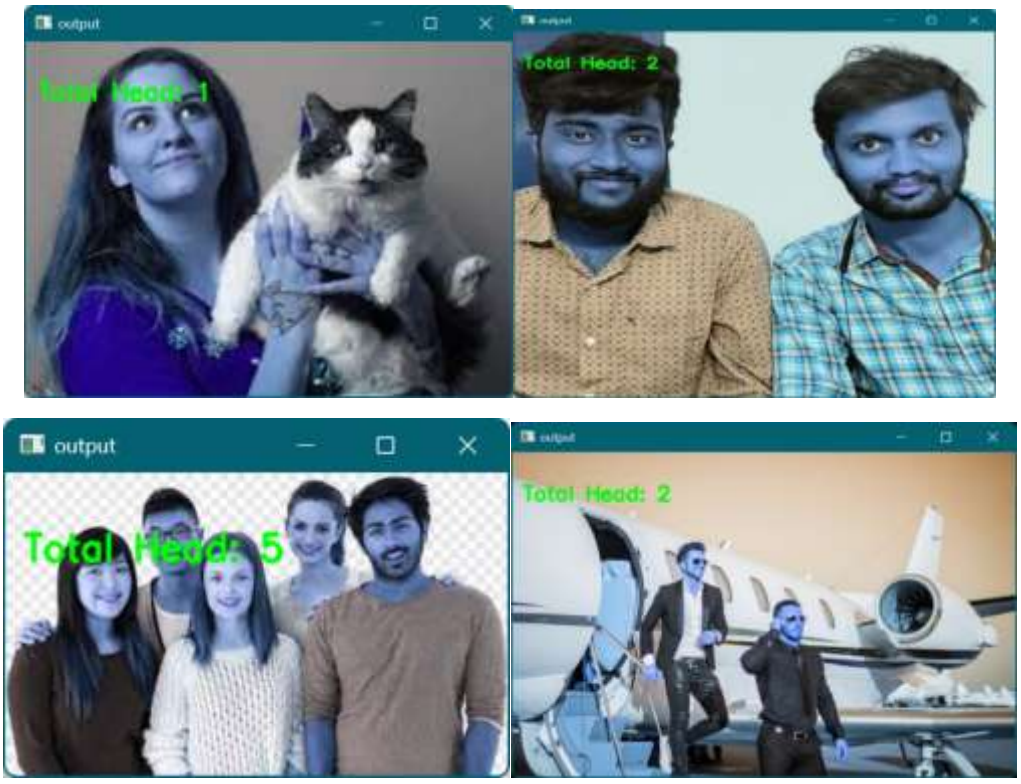

Figure 6: Sample prediction on test images using proposed RCNN model.

Figure 7: Sample prediction on test video using proposed RCNN model.

5. CONCLUSION

The People Counting System presented in this research successfully uses a pre-trained Faster R-CNN model for detecting and counting human heads in both images and videos. The system incorporates a user-friendly graphical interface, making it accessible to individuals with varying levels of technical expertise. Real-time processing capabilities enhance its utility, particularly for applications like crowd management, security surveillance, and occupancy analysis.

References

[1] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: People detection and articulated pose estimation, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 886–893.

[3] D.T. Nguyen, W. Li, P.O. Ogunbona, Human detection from images and videos: A survey, Pattern Recognit. 51 (2016) 148–175.

[4] S. Li, M.O. Tezcan, P. Ishwar, J. Konrad, Supervised people counting using an overhead fisheye camera, in: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8.

[5] G. Antonini, J. P. Thiran, Counting pedestrians in video sequences using 330 trajectory clustering, Circuits and Systems for Video Technology, IEEE Transactions on 16 (8) (2006) 1008–1020.

[6] I. S. Topkaya, H. Erdogan, F. Porikli, Counting people by clustering person detector outputs, in: Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, IEEE, 2014, pp. 313–318.

[7] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Advances in Neural Information Processing Systems, 2010, pp. 1324–1332.

[8] A. B. Chan, N. Vasconcelos, Counting people with low-level features and bayesian regression, Image Processing, IEEE Transactions on 21 (4) (2012) 2160–2177.

[9] P. Morerio, L. Marcenaro, C. S. Regazzoni, People count estimation in small crowds, in: Advanced video and signal-based surveillance (AVSS), 2012 IEEE Ninth International Conference on, IEEE, 2012, pp. 476–480.

[10] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 2547– 2554.

[11] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proc. CVPR, 2015.

[12] W. Ouyang, X. Wang, Joint deep learning for pedestrian detection, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2056–2063.

[13] D. Conte, P. Foggia, G. Percannella, F. Tufano, M. Vento, A method for counting people in crowded scenes, in: Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on, IEEE, 2010, pp. 225–232.

[14] S. Wang, J. Zhang, Z. Miao, A new edge feature for head-shoulder detection, in: Image Processing (ICIP), 2013 20th IEEE International Conference on, IEEE, 2013, pp. 2822–2826.

[15] T. Van Oosterhout, S. Bakkes, B. J. Kr¨ose, Head detection in stereo data for people counting and segmentation., in: VISAPP, 2011, pp. 620–625.

[16] K.-E. Aziz, D. Merad, B. Fertil, N. Thome, Pedestrian head detection and tracking using skeleton graph for people counting in crowded environments., in: MVA, 2011, pp. 516–519.

[17] V. B. Subburaman, A. Descamps, C. Carincotte, Counting people in the crowd using a generic head detector, in: Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on, IEEE, 2012, pp. 470–475.

[18] F. Conti, A. Pullini, L. Benini, Brain-inspired classroom occupancy monitoring on a low-power mobile platform, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, IEEE, 2014, pp. 624–629.