International Journal of Innovative Computing, Information and Control
Volume **22**, Number **1**, February **2026**

 ISSN 1349-4198
pp. **95–107**

MONITORING CROWD BEHAVIOR FOR CAMPUS SURVEILLANCE IN INDONESIA USING CONVOLUTIONAL NEURAL NETWORK

Lina Lina[1,*], Arlends Chris[2], Ranny Ranny[3] and Puguh Hiskiawan[3]

[1]Faculty of Information Technology
[2]Faculty of Medicine
Tarumanagara University
Jl. Letjen. S. Parman No.1, Jakarta 11440, Indonesia
arlendsc@fk.untar.ac.id
*Corresponding author: lina@untar.ac.id

[3]Data Science Department
Faculty of Technology and Design
Bunda Mulia University
Kav 7-9 Alam Sutera, Tangerang 15143, Indonesia
{ ranny; phiskiawan }@bundamulia.ac.id

Received June 2025; revised October 2025

Abstract. *Monitoring mass behavior within campus environments is essential for ensuring security, particularly in Indonesia, where universities often have large and dynamic student populations. Typical campus behavior in Indonesia includes informal gatherings, indoor studying, communal dining, and spontaneous group interactions that often occur in public areas. This paper proposes the development of a surveillance system that monitors crowd behaviors to anticipate potential conflicts and criminal activities within campus settings. The system applied a Convolutional Neural Network (CNN) architecture to detecting specific crowd activities and analyzing their behavioral patterns. The CNN model was trained using the ResNet-50 backbone, with various batch sizes and a total of 1,000 training epochs. A total of 8,000 images from the CrowdHuman dataset were used for model training, while 2,000 images collected from an Indonesian university campus were employed for testing. The test images included diverse scenarios featuring varying numbers of individuals engaged in different activities. Experimental results demonstrated that the proposed system achieved high detection accuracy across multiple behavioral scenarios, including standing, fighting, studying, and eating or drinking. The system achieved a crowd-counting accuracy of 95% and an activity recognition accuracy of 85.5%, indicating its potential effectiveness for real-time behavioral monitoring in campus environments.*

Keywords: Monitoring system, Crowd behavior, Activity recognition, Campus surveillance, Convolutional Neural Network

1. **Introduction.** Human overcrowding is one of the well-known cases circulating on social media today. According to the Central Statistics Agency, the population in Indonesia has reached 278.69 million people in mid-2023 [1]. This number increased by around 1.05% from the previous year, 2022 [2]. This of course has an impact on the entire community and the social environment that experiences it. The most visible impact is due to the development of an area which triggers dense population in that area [3].

Crowd counting refers to the process of estimating the number of people in a given space or area [4]. It is an important task in fields like computer vision, surveillance, and public safety [5]. With the advancement of technologies such as deep learning and

DOI: 10.24507/ijicic.22.01.95

artificial intelligence, automated systems are now capable of accurately counting crowds in real time, using tools like cameras, sensors, and other data sources [6-8]. The usefulness of crowd counting spans across various domains such as public safety, urban planning, retail and marketing, event management, and security and surveillance. Using crowd counting and effective crowd management protocols, situations like CSA can be prevented, thus providing a greater sense of public security [9].

Crowd counting can be used to monitor large gatherings or events, such as concerts, sports games, or protests, ensuring that crowd density remains within safe limits. This helps prevent dangerous overcrowding and enables authorities to respond quickly if needed. Accurate crowd data can guide city planners in designing better spaces and public transportation systems, ensuring they can accommodate large crowds during peak times. In shopping malls or retail stores, crowd counting helps businesses analyze customer behavior, optimize store layouts, and manage resources like staff or inventory [9]. Organizers of events use crowd counting to gauge attendance and adjust logistics, such as food distribution or security staffing, to ensure a smooth operation. In critical infrastructure, airports, and public spaces, crowd counting plays a role in monitoring unusual crowd behavior or identifying potentially hazardous situations. Veral, crowd counting is an essential tool for improving safety, enhancing operational efficiency, and making data-driven decisions in a variety of industries. Moreover, while crowd counting provides the necessary context about how many people are in an area, activity recognition helps to understand how people behave in those conditions. Combining both tasks allows for a deeper understanding of human behavior in crowds, improving safety, resource allocation, and situational awareness in various applications [10].

In this paper, a complete crowd counting and activity recognition system is proposed using deep learning algorithm. Deep learning techniques, particularly Convolutional Neural Network (CNN), have become popular for crowd counting because of their ability to extract relevant features from images [4,11-14]. CNN can automatically learn spatial features, such as the presence of faces, body parts, and interactions, which helps in detecting and counting people in crowded settings. By using crowd counting alongside activity recognition, systems can monitor not just how many people are present but also what activities are occurring within the crowd (e.g., a protest, a celebration, or a panic situation). This is especially useful in surveillance for public safety, where knowing both crowd size and activity type is crucial.

2. **Method.**

In this paper, an integrated crowd counting and activity recognition system is developed. Combining both allows for more sophisticated applications, where systems not only count people but also understand crowd dynamics, improving real-time safety, surveillance, and event management. Deep learning plays a central role in both domains, driving advances in accuracy and real-time processing capabilities.

2.1. **Crowd counting.**

Crowd counting provides the necessary context about how many people are in an area, which can help in activity recognition by establishing a baseline for interpreting behavior in a crowd. The application of crowd counting is widely applied in monitoring activities in the fields of medical services, urban and strategic planning, defense and public security [15-19]. Crowd counting has been widely applied to images or videos and analyzed to obtain results or goals. The aim of crowd counting is to be able to count the number of objects or subjects in a crowd of image or video data, so that the final results can be analyzed to help other interests [20].

Crowd counting is an essential task in computer vision and has been approached using various techniques. Two primary methods to tackle crowd counting are density map-based counting and object-based counting. Both have distinct advantages and are useful depending on the specific challenges of the dataset or the environment. Density map-based counting is a technique where the goal is to generate a density map that represents the distribution of people across the image. Instead of directly counting individual people, this method estimates the local crowd density for each pixel or region of the image [5,21]. Once the density map is predicted, the total crowd count can be computed by summing all the pixel values in the density map. This model works well when people are occluded, since the density map focuses on regions rather than individual people. However, the disadvantage of this method is lack of fine-grained detail of the individuals locations. The example of a density map counting is represented in Figure 1. Ma [21] introduced a method to estimate the number of people in a crowd from an image with varying crowd density and perspective. For this, the paper introduced a Multi-column Convolutional Neural Network (MCNN) architecture which maps the image to its equivalent crowd density map. Singh [22] tried to solve the same problem using a new CNN architecture compromising of end-to-end cascaded network in order to learn crowd count classification and density map estimation at the same time. The proposed method incorporates a high-level prior into the density estimation network by categorizing the crowd count, thereby providing a coarse estimation of the overall count in the image. This method produced higher-quality density maps than recent state-of-the-art techniques [24].

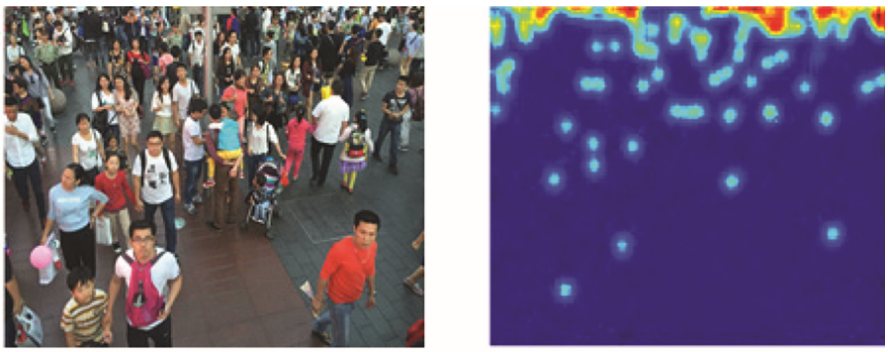

FIGURE 1. Density map-based counting

Object-based counting involves detecting individual people (objects) in the image and directly counting how many are present. This method treats the task as an object detection problem, where the goal is to localize and classify each person as an object. The object-based counting is similar to the traditional object detection. A model is trained to identify individual people as discrete objects within the image. Models like YOLO (You Only Look Once), Faster R-CNN, or SSD (Single Shot MultiBox Detector) are used for this task. They predict bounding boxes around detected people and output the count of people in the image. The advantages of this method are in precise counting and able to localize objects. However, the drawbacks are challenges with occlusions and computationally expensive. Figure 2 shows the example of object-based counting. Thanasutives et al. [24] have a different approach to crowd counting. They suggest a method based on two modified neural networks, SFANet and SegNet, which are dual path multi-scale fusion networks. They named these two networks or models M-SFANet and M-SegNet [25,26]. The encoder for SFANet is connected with Atrous Pyramid Pooling (APP) that contains parallel atrous convolution layers with varying sampling rates, as a result of which it can extract multi-scale features of the target object and incorporate that into a larger context.

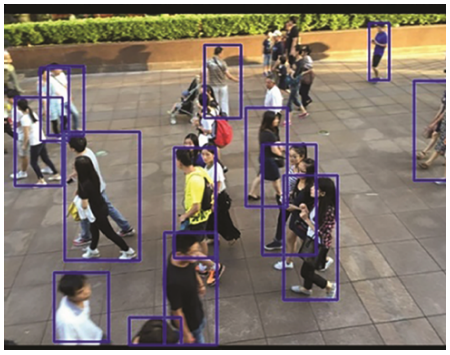

FIGURE 2. Object-based counting

To deal with scale variation in the input image further, to adaptively encode the scale of the contextual information, the authors use the Context-Aware-Module (CAM), which is also coupled to M-SFANet. As a result, the model developed is useful for counting in both dense and sparse crowd scenarios [27].

2.2. **Human activity recognition.** Human Activity Recognition (HAR) is the process of identifying and classifying human activities based on data, often collected from sensors. HAR is a term used to describe the process of identifying and recognizing various human activities, for example, walking, running, sitting, lying, standing, bathing, cooking, driving, opening doors, and other activities [28]. The application of human activity recognition is also widely applied in monitoring activities, such as for diagnosis in the medical field, tracking parental activities, tracking crime levels, home and driving security, military activities and others. Through this application, HAR can be applied with data based on sensors, accelerometers, images, or videos [29].

Human Activity Recognition (HAR) can be approached using vision-based methods and sensor-based methods. Vision-based HAR is more suited for environments where observing the entirety of a space or activity is essential, such as smart homes and public safety. However, it faces privacy and environmental challenges. Vision-based HAR uses computer vision techniques to analyze video data which were captured by cameras or other visual sensors. In addition, for sensor-based ones, this is done by using sensor data on certain axes, and identifying activity from these axes [30].

Sensor-based HAR excels in personal, continuous monitoring and is more suitable for applications where the user wears a device, such as fitness tracking and personal healthcare. Sensor-based HAR utilizes sensor data to measure physical properties. Thus, it uses wearable sensors, such as accelerometers, gyroscopes, and magnetometers.

2.3. **Convolutional neural network.** Convolutional Neural Network (CNN) has become very popular for various image and video classification tasks, but their application to Human Activity Recognition (HAR) has also gained traction in recent years. When applying CNN to vision-based HAR, the CNN architecture processes image or video frames to identify the activities of individuals in those frames [31]. Convolutional Neural Networks (CNN) consists of several layers that work together to extract features and make predictions. The basic structure of a CNN usually consists of the input layer, the convolutional layer, the activation layer, the pooling layer, and the fully connected layer. These layers can be varied depending on the complexity of the task, such as adding more convolutional layers, different activation functions, or advanced techniques like residual connections [32]. The illustration of the CNN architecture is depicted in Figure 3.

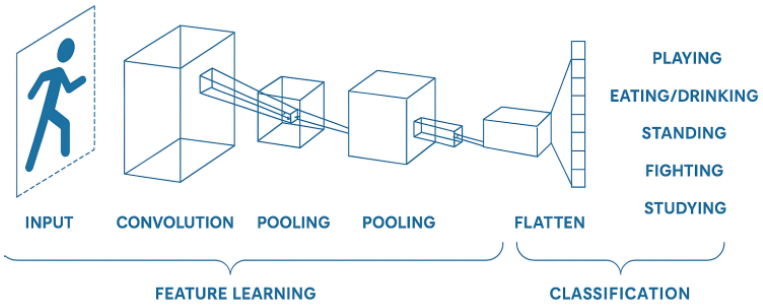

FIGURE 3. The illustration of the CNN architecture

The input layer of the CNN is where the data is fed into the network. For an image, this would be a 2D grid of pixel values with RGB channels [33]. The core building block of a CNN is the convolutional layer. This layer applies a set of filters to the input data in order to detect various features such as edges, textures, or patterns. Each filter convolves across the input data, performing a dot product between the filter and the local region of input, producing a feature or activation map [34].

The convolutional layer is calculated with the following equation [35]:

$$x(i,j) = \Sigma_m \Sigma_n w^l_{m,n} * o^{l-1}_{i+m,j+n} + b \tag{1}$$

where x is result of convolutional process, m and n are numbers of data, o is data, and b is bias. An example of convolutional layer calculation can be seen in Figure 4.

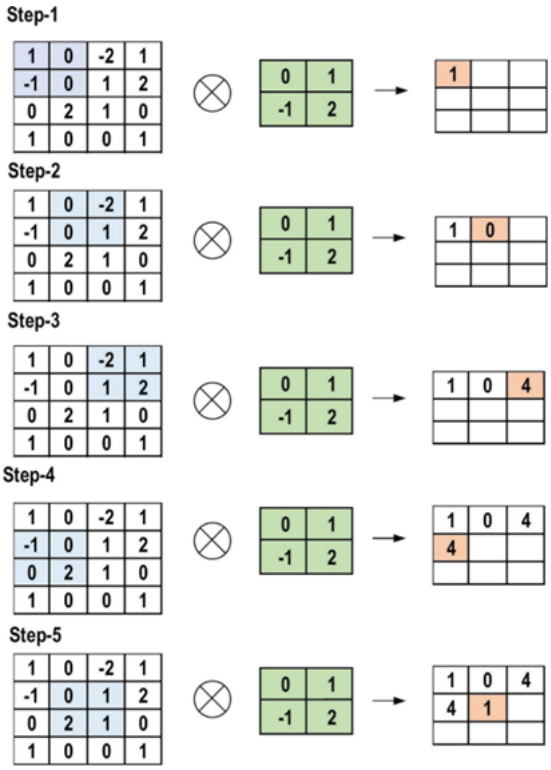

FIGURE 4. The examples of convolutional layer processes

The next process is applying the activation function to each convolution operation to introducing non-linearity into the model. Rectified Linear Unit (ReLU) is the most common activation function used in CNN. ReLU replaces all negative values in the feature

map with zero which allow the network to learn more complex patterns [36]. The process follows Equation (2) below.

$$ReLU(x) = \max(0, x) \tag{2}$$

with x as the value in convolution map.

The process continues with the pooling layer. Pooling reduces the spatial dimensions of the feature maps which help to reduce the number of parameters and computational complexity. There are two most popular types of pooling methods, i.e., the average pooling and the maximum pooling [37]. In average pooling, the average or mean of the feature map is determined. Meanwhile, in max pooling, the highest value of the matrix value is selected. That way, the resulting output is a matrix with a smaller pool size [34].

After several layers of convolutions and pooling, the CNN typically has one or more fully connected layers, which help to make the final classification decision. The output of the final pooling layer is flattened into a 1D vector and passed through one or more fully connected layers. These layers perform a dot product and pass the result through an activation function. The SoftMax function is commonly used in the final layer for multi-class classification tasks. Given a vector of raw outputs, SoftMax applies the following formula to each element z_i of the vector. The output from the fully-connected layer is the output of CNN [37].

$$softmax(z_i) = \frac{e^{z_i}}{\Sigma_j e^{z_j}} \tag{3}$$

2.4. **ResNet-50.** ResNet or Residual Network is a type of Convolutional Neural Network (CNN) architecture which has advantages in optimizing neural networks and can obtain greater accuracy compared to other architectures [38]. It consists of 50 layers, hence the name ResNet-50. The main innovation in ResNet (Residual Networks) is the introduction of residual connections, or skip connections, that allow the model to bypass certain layers. Figure 5 shows the architecture and layers of ResNet with various number of layers. ResNet has advantages in optimizing neural networks due to the problem of missing gradients when training very deep neural networks so that the application of ResNet is a solution in training deep networks so that neurons can remember and store values optimally. In order to be more optimal, in ResNet there is a function called skip network or skip connection. The skip network contained in the building block is called the residual function or residual block. ResNet-50 has 16 residual blocks, each residual block consists of 1 (one) convolutional block and the rest are identity blocks. The convolutional block of ResNet-50 architecture along with each block can be seen in Figure 6.

3. **Results and Discussion.** To determine the system's performance, several videos with various scenarios were created and tested in the experiments. The experiments conducted in this paper used an Intel i9 CPU, 32GB of memory, and a Nvidia GeForce RTX 3070 graphics card. The public dataset used to test the proposed method was the CrowdHuman dataset which was collected through the Internet. This dataset contains 8,000 images for training and 2,000 images for testing. In the preprocessing stage, images were resized to 315×315 pixels. The layers used to construct the model follow the original ResNet-50 architecture. The CrowdHuman dataset automatically provides detailed annotations for each image through two main components: "ID" and "gtboxes". The ID denotes the image filename, while gtboxes include attributes such as "tag", "vbox", "fbox", "hbox", "extra", and "head_attr". These fields contain automatically generated bounding boxes for the visible body (vbox), full body (fbox), and head region (hbox), along

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

FIGURE 5. The architecture and layers of ResNet

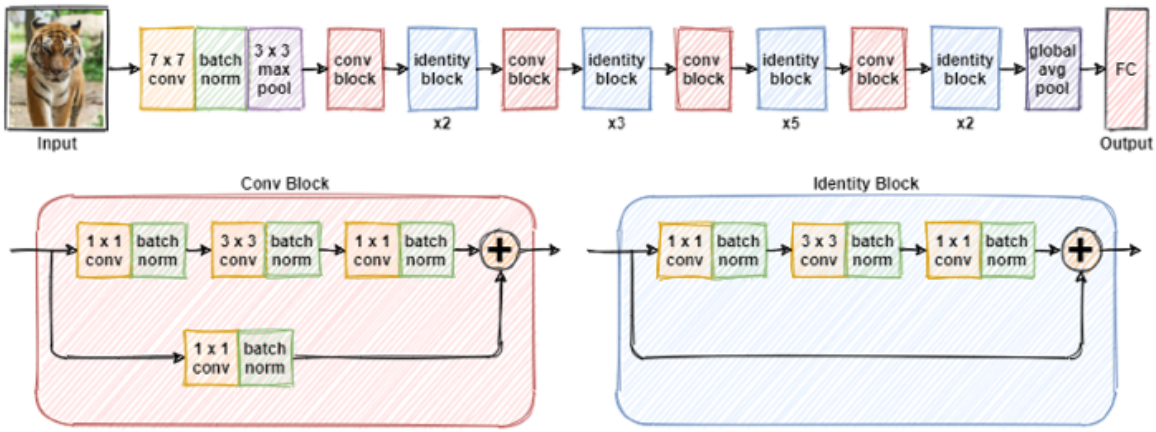

FIGURE 6. The architecture, residual block, convolutional block, and identity block in ResNet-50 [26-28]

with metadata describing occlusion, uncertainty, and ignore flags. Since these annotations are provided directly by the dataset, no additional manual labeling is required. In this study, only the "tag" and "hbox" annotations are utilized for model development and analysis. In the current study, we intentionally limit occlusion handling to basic bounding-box filtering to maintain focus on evaluating spatial feature transfer rather than developing a specialized occlusion-robust detector. Different scenarios were conducted in the experiments using four activities, namely standing, fighting, eating and drinking, and studying. Figure 7 shows the samples of images that are used in the experiments.

The CNN model was trained using the ResNet-50 whose backbone design is depicted in Figure 8. For obtaining the best model for the system, we conducted several experiments using various training parameters for the ResNet-50 architecture. In the experimental setup, the batch size limit manageable by the machine was first examined to validate code stability. The number of epochs was then varied to evaluate model performance and select the optimal configuration. Initially, the batch size was determined with the smallest value, namely 32 with 1,000 epoch. Then, larger batch sizes were used, namely 64 and 128 with the same epoch size. The best result of these three models will then

(a) (b)

(c) (d)

FIGURE 7. The samples of images used in the experiments: (a) Standing, (b) fighting, (c) eating and drinking, and (d) studying

Layer (Type)	Output Shape	Param #
ResNet (Backbone)		
stem.Conv2d	[1, 64, 112, 112]	9408
res2.BottleneckBlock-0	[1, 256, 56, 56]	72192
res2.BottleneckBlock-1	[1, 256, 56, 56]	72192
res2.BottleneckBlock-2	[1, 256, 56, 56]	72192
res3.BottleneckBlock-0	[1, 512, 28, 28]	295680
res3.BottleneckBlock-1	[1, 512, 28, 28]	295680
res3.BottleneckBlock-2	[1, 512, 28, 28]	295680
res3.BottleneckBlock-3	[1, 512, 28, 28]	295680
res4.BottleneckBlock-0	[1, 1024, 14, 14]	1182720
res4.BottleneckBlock-1	[1, 1024, 14, 14]	1182720
res4.BottleneckBlock-2	[1, 1024, 14, 14]	1182720
res4.BottleneckBlock-3	[1, 1024, 14, 14]	1182720
res4.BottleneckBlock-4	[1, 1024, 14, 14]	1182720
res4.BottleneckBlock-5	[1, 1024, 14, 14]	1182720
RPN (Proposal Generator)		
rpn_head.Conv2d	[1, 1024, 14, 14]	9438208
rpn_head.objectness_logits	[1, 15, 14, 14]	15375
rpn_head.anchor_deltas	[1, 60, 14, 14]	61500
ROIHeads (Region of Interest)		
roi_heads.pooler	[1, 1024, 7, 7]	0
roi_heads.res5.BottleneckBlock	[1, 2048, 7, 7]	14811136
roi_heads.box_predictor	[1, 4]	8196
Total params: 32839439		

FIGURE 8. The ResNet-50 backbone model used in the experiments

TABLE 1. Various training parameters for ResNet-50 model

Model	Batch size	Epoch	Training	
			Accuracy (%)	Loss
1	32	1,000	88.67	1.21
2	64	1,000	89.26	1.12
3	128	1,000	89.34	1.21
4	64	10,000	89.65	1.15

TABLE 2. Testing results of the system

Scenario		Accuracy (%)	
# subject in each frame	Activity type	# of subjects correctly detected	Activity correctly recognized
2 subjects	Standing	98	74
	Fighting		
	Studying		
	Eating/Drinking		
3 subjects	Standing	94	84
	Fighting		
	Studying		
	Eating/Drinking		
4 subjects	Standing	96	88
	Fighting		
	Studying		
	Eating/Drinking		
5 subjects	Standing	92	96
	Fighting		
	Studying		
	Eating/Drinking		

be re-trained with higher epoch values. The accuracy and loss of the trained four models can be seen in Table 1. The values (batch sizes of 32, 64, and 128; epochs ranging from 1,000-10,000) were determined through a series of preliminary experiments designed to balance model convergence, generalization performance, and computational feasibility. Specifically, we conducted exploratory trials similar to a coarse grid search to evaluate the impact of different batch sizes on stability and accuracy. Smaller batches (e.g., 32) yielded smoother convergence but required longer training times, whereas larger batches (e.g., 128) improved computational efficiency at the cost of slight fluctuations in validation performance. Therefore, intermediate values were retained to ensure robustness under varying computational constraints. The epoch range was selected to accommodate models with different convergence rates, with early stopping applied once performance plateaued. The highest validation accuracy was achieved by Model 4 with 89.65%. Thus, Model 4 will be used as a model for the proposed system.

Moreover, to determine the system's performance in real situations, several video scenarios were created in the testing phase. From the test results with 200 videos, it shows that for the 2 human subject scenarios, the accuracy was 98% for the number of subjects and 74% for activity recognition. Then, for the scenario of 3 human subjects, the accuracy was 94% for the number of subjects and 84% for activity recognition. For the 4 human subject scenarios, the accuracy was 96% for the number of subjects and 88% for activity

recognition. Meanwhile, for the 5 human subject scenarios, the accuracy was 92% for the number of subjects and 96% for activity recognition.

The application system for detection, calculation and activity recognition received a final score for test accuracy of 95% for the number of subjects and 85.5% for activity recognition, where the details per scenario are 98% for 2 subjects and 74% for activity recognition, 94% for 3 subjects and 84% for activity recognition, 96% for 4 subjects and 88% for activity recognition, and 92% for 5 subjects and 96% for the introduction of the activity. The activity classes tested are standing, fighting, playing, and eating and drinking.

The confusion matrix in Figure 9 shows that the model performs well in recognizing standing class, with the highest correct predictions, indicating effective posture-based feature learning. However, considerable confusion occurs among visually similar activities such as studying and eating/drinking, likely due to shared spatial contexts and limited object cues. Misclassifications between those two classes further suggest that static spatial features alone are insufficient to capture motion dynamics. These results highlight that while the model effectively distinguishes clear posture-based actions, it struggles with contextual and occluded scenes.

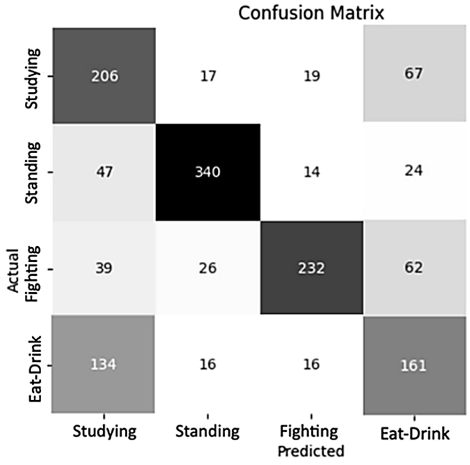

FIGURE 9. Confusion matrix

For the human subject calculation system, the number of bounding box heads detected from the Faster R-CNN model using the ResNet-50 backbone is taken. This model is quite good at detecting human subjects, especially the head. Meanwhile, for the activity recognition system for human subjects, the CNN model uses the ResNet-50 architecture. This model is quite good at detecting human subjects, especially the head. Meanwhile, for the activity recognition system for human subjects, the CNN model uses the ResNet-50 architecture. This model is quite good in recognizing human activities with a predetermined 5 classes of classification.

From the tests that have been carried out, higher accuracy is still needed so that the system is more precise in detecting and recognizing, especially for human subjects in the head, because the system can detect certain parts that resemble a head, such as a bag that is detected as a head. Meanwhile, for the recognition of human subjects, it is considered accurate but the classification for several similar activities is still often confused, so that in testing, subjects need to carry out activities very clearly so that they can be classified correctly. Things that can be done are increasing the amount of data with more specific data, changing and trying the configuration of several layers and trying to use different architectures that are suitable for subject detection and activity recognition.

4. **Conclusion.** Based on the conducted experiments, the proposed system could give high detection accuracy for images with various scenarios. The application successfully performed human detection, counting, and activity recognition tasks. The number of human subjects was determined based on the detection of head regions, while activity recognition involved classifying detected behaviors into predefined categories. The system was capable of categorizing activities into one of four classes: standing, fighting, studying, and eating or drinking. The evaluation included testing images containing varying numbers of individuals. In scenarios involving two human subjects, the system achieved a subject count accuracy of 98% and an activity recognition accuracy of 74%. For three subjects, the corresponding accuracies were 94% and 84%, respectively. In the case of four subjects, the system attained 96% accuracy for subject counting and 88% for activity recognition. With five human subjects, the accuracies were 92% and 96%, respectively. Overall, the system achieved an average accuracy of 95% for subject counting and 85.5% for activity recognition. Future work will focus on enhancing the system's performance by incorporating more specific classification schemes, including facial recognition and detailed analysis of human activities and interactions, with the goal of achieving higher overall accuracy.

REFERENCES

[1] A. Zaqiah, M. Triani and I. Yeni, The influence of education level, unemployment and population on poverty levels in Indonesia, *J. Econ. Dev. Stud.*, vol.5, no.4, pp.33-42, 2023.

[2] B. T. Diniati and D. A. Permana, The influence of population size, economic growth, human development index, unemployment on poverty in Indonesia from 1998 to 2022, *ICHES Int. Conf. Humanit. Educ. Sos.*, vol.3, no.1, 11, 2024.

[3] E. Rustiadi, A. E. Pravitasari, Y. Setiawan, S. P. Mulya, D. O. Pribadi and N. Tsutsumida, Impact of continuous Jakarta megacity urban expansion on the formation of the Jakarta-Bandung conurbation over the rice farm regions, *Cities*, vol.111, 103000, DOI: 10.1016/j.cities.2020.103000, 2021.

[4] K. B. A. Hassen, J. J. M. Machado and J. M. R. S. Tavares, Convolutional neural networks and heuristic methods for crowd counting: A systematic review, *Sensors*, vol.22, no.14, pp.1-17, DOI: 10.3390/s22145286, 2022.

[5] M. Fiandeiro, T. T. Nguyen, H. Wong and E. B. Hsu, Modernized crowd counting strategies for mass gatherings – A review, *J. Acute Med.*, vol.13, no.1, pp.4-11, DOI: 10.6705/j.jacme.202303_13(1).0002, 2023.

[6] H. Mokayed, T. Z. Quan, L. Alkhaled and V. Sivakumar, Real-time human detection and counting system using deep learning computer vision techniques, *Artif. Intell. Appl.*, vol.1, no.4, pp.221-229, DOI: 10.47852/bonviewaia2202391, 2022.

[7] A. Tomar, R. Nijhawan and D. Koundal, EDCCN: A benchmark encoder-decoder framework for accurate crowd counting, *Neurocomputing*, vol.640, 130304, DOI: 10.1016/j.neucom.2025.130304, 2025.

[8] M. Ş. Gündüz and G. Işık, A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models, *J. Real-Time Image Process.*, vol.20, no.1, pp.1-12, DOI: 10.1007/s11554-023-01276-w, 2023.

[9] S. Peng, B. Yin, Q. Yang, Q. He and L. Wang, Exploring density rectification and domain adaption method for crowd counting, *Neural Comput. Appl.*, vol.35, no.4, pp.3551-3569, DOI: 10.1007/s00521-022-07917-8, 2023.

[10] J. Dong, Z. Zhao and T. Wang, Crowd counting by multi-scale dilated convolution networks, *Electron.*, vol.12, no.12, DOI: 10.3390/electronics12122624, 2023.

[11] Z. Yuan, SPCANet: Congested crowd counting via strip pooling combined attention network, DOI: 10.7717/peerj-cs.2273, 2024.

[12] Z. Zhao, P. Ma, M. Jia, X. Wang and X. Hei, A dilated convolutional neural network for cross-layers of contextual information for congested crowd counting, *Sensors*, vol.24, no.6, pp.1-19, DOI: 10.3390/s24061816, 2024.

[13] P. Zhang, W. Lei, X. Zhao, L. Dong and Z. Lin, An adaptive multi-scale network based on depth information for crowd counting, *Sensors*, vol.23, no.18, DOI: 10.3390/s23187805, 2023.

[14] U. Sajid, H. Sajid, H. Wang and G. Wang, ZoomCount: A zooming mechanism for crowd counting in static images, *IEEE Trans. Circuits Syst. Video Technol.*, vol.30, no.10, pp.3499-3512, DOI: 10.1109/TCSVT.2020.2978717, 2020.
[15] A. Samuel, D. Arisandi and L. Lina, Web and mobile-based information systems for monitoring children activities in Kindergarten, *AIP Conf. Proc.*, vol.2680, no.1, DOI: 10.1063/5.0127837, 2023.
[16] J. Andrean and L. Lina, Human activity recognition of video recording using convolutional neural network, *AIP Conf. Proc.*, vol.2680, no.1, DOI: 10.1063/5.0127839, 2023.
[17] J. Su and L. Lina, Human activity recognition of surveillance videos using multilayer perceptron, *AIP Conf. Proc.*, DOI: 020121.10.1063/5.0127838, 2023.
[18] K. Prastika and L. Lina, Application of individual activity recognition in the room using CNN Alexnet method, *IOP Conf. Ser. Mater. Sci. Eng.*, vol.1007, no.1, DOI: 10.1088/1757-899X/1007/1/012162, 2020.
[19] M. Fazli, K. Kowsari, E. Gharavi, L. Barnes and A. Doryab, HHAR-net: hierarchical human activity recognition using neural networks, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol.12615, pp.48-58, DOI: 10.1007/978-3-030-68449-5_6, 2021.
[20] V. A. Sindagi and V. M. Patel, A survey of recent advances in CNN-based single image crowd counting and density estimation, *Pattern Recognit. Lett.*, vol.107, pp.3-16, DOI: 10.1016/j.patrec.2017.07.007, 2018.
[21] Y. Ma, Single-image crowd counting via multi-column convolutional neural network, *CVPR Pap.*, vol.2, no.35, pp.589-597, DOI: 10.1002/slct.201701956, 2016.
[22] D. K. Singh, S. Paroothi, M. K. Rusia and M. A. Ansari, Human crowd detection for city wide surveillance, *Procedia Comput. Sci.*, vol.171, pp.350-359, DOI: 10.1016/j.procs.2020.04.036, 2020.
[23] N. H. Shabrina, D. Gunawan and A. S. Harahap, Convolutional neural networks for identifying papillary thyroid cancer histopathological image, *International Journal of Innovative Computing, Information and Control*, vol.21, no.2, pp.565-576, DOI: 10.24507/ijicic.21.02.565, 2025.
[24] P. Thanasutives, K. I. Fukui, M. Numao and B. Kijsirikul, Encoder-decoder based convolutional neural networks with multi-scale-aware modules for crowd counting, *Proc. – Int. Conf. Pattern Recognit.*, pp.2382-2389, DOI: 10.1109/ICPR48806.2021.9413286, 2020.
[25] P. Hiskiawan, C. Chih, C. Zheng and K. Ye, Processing of electrical resistivity tomography data using convolutional neural network in ERT-NET architectures, *Arab. J. Geosci.*, pp.1-14, DOI: 10.1007/s12517-023-11690-w, 2023.
[26] T. Lähivaara, L. Kärkkäinen, J. M. J. Huttunen and J. S. Hesthaven, Deep convolutional neural networks for estimating porous material parameters with ultrasound tomography, *J. Acoust. Soc. Am.*, vol.143, no.2, pp.1148-1158, DOI: 10.1121/1.5024341, 2018.
[27] B. Li, H. Huang, A. Zhang, P. Liu and C. Liu, Approaches on crowd counting and density estimation: A review, *Pattern Anal. Appl.*, vol.24, no.3, pp.853-874, DOI: 10.1007/s10044-021-00959-z, 2021.
[28] M. A. Khan, H. Menouar and R. Hamila, Revisiting crowd counting: State-of-the-art, trends, and future perspectives, *Image Vis. Comput.*, vol.129, pp.1-17, DOI: 10.1016/j.imavis.2022.104597, 2023.
[29] Y. Gu, M. Wu, Q. Wang, S. Chen and L. Yang, A deep learning-based crowd counting method and system implementation on neural processing unit platform, *Comput. Mater. Contin.*, vol.75, no.1, pp.493-512, DOI: 10.32604/cmc.2023.035974, 2023.
[30] A. R. Kamila, J. F. Andry, A. Wahyu, C. Kusuma, W. Prasetyo and G. H. Derhass, Analysis comparison of K-Nearest Neighbor, multi-layer perceptron, and decision tree algorithms in diamond price prediction, *Cogito Smart Journal*, vol.10, no.2, pp.298-311, 2024.
[31] Q. Zhao and Z. Shang, Deep learning and its development, *J. Phys. Conf. Ser.*, vol.1948, no.1, DOI: 10.1088/1742-6596/1948/1/012023, 2021.
[32] W. Wang, J. Yu, Y. Ma, Z. Pan and T. Chen, Bearing fault diagnosis based on deep learning and array stochastic resonance under strong noise background, *International Journal of Innovative Computing, Information and Control*, vol.21, no.2, pp.549-563, DOI: 10.24507/ijicic.21.02.549, 2025.
[33] A. Anton, N. F. Nissa, A. Janiati, N. Cahya and P. Astuti, Application of deep learning using convolutional neural network (CNN) method for women's skin classification, *Sci. J. Informatics*, vol.8, no.1, pp.144-153, DOI: 10.15294/sji.v8i1.26888, 2021.
[34] V. Moshnyaga, T. Osamu, T. Ryu and K. Hashimoto, Identification of basic behavioral activities by heterogeneous sensors of in-home monitoring system, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol.9277, pp.160-174, DOI: 10.1007/978-3-319-24195-1_12, 2015.
[35] A. L. Maas, A. Y. Hannun and A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, *ICML Work. Deep Learn. Audio, Speech Lang. Process.*, vol.28, 2013.

[36] O. Ozturk, B. Sariturk and D. Z. Seker, Comparison of fully convolutional networks (FCN) and U-Net for road segmentation from high resolution imageries, *Int. J. Environ. Geoinformatics*, vol.7, no.3, pp.272-279, DOI: 10.30897/ijegeo.737993, 2020.
[37] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *Proc. of IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol.2016, pp.770-778, DOI: 10.1109/CVPR.2016.90, 2016.
[38] S. Li, J. Lin, Y. Lv and T. Li, Deep learning-based algorithm for complex small target detection in UAV aerial images, *International Journal of Innovative Computing, Information and Control*, vol.21, no.1, pp.135-152, DOI: 10.24507/ijicic.21.01.135, 2025.

Author Biography

Lina Lina is a Professor at the Faculty of Information Technology, Tarumanagara University, Indonesia. She received her Doctoral degree in 2009 from Nagoya University, Japan. Her research interests include computer vision, image recognition, and intelligent systems. She is a member of the Institute of Electrical and Electronic Engineers (IEEE) and the Institute of Electronics, Information and Communication Engineers (IEICE).

Arlends Chris is an Assistant Professor at Faculty of Medicine, Tarumanagara University, Indonesia. He received his Ph.D. degree from Universitas Negeri Jakarta (UNJ), Indonesia, in 2018. His main teaching and research interests include human anatomy, histology and physiology, mental health and primary care medicine. He has published several research articles in international journals of medicine.

Ranny Ranny obtained her Bachelor degree from Tarumanagara University, Indonesia, in 2010, Master degree from the University of Indonesia, Indonesia, in 2013, and Doctorate degree from the Bandung Institute of Technology, Indonesia, in 2024. She is a faculty member at the Data Science Department, Bunda Mulia University, Indonesia. Her academic and research interests are focused on machine learning, artificial intelligence, and sound processing. She has been involved in various research projects and has published several papers in the fields of audio signal processing and machine learning.

Puguh Hiskiawan received his Ph.D. degree in 2024 from National Central University, Taiwan. He obtained his B.Sc. and M.Sc. degrees from Institut Teknologi Sepuluh Nopember, Indonesia, in 2000 and 2006. His research interests include computational physics, data science, digital image processing, and computer vision. He is currently a Lecturer in Data Science and Artificial Intelligence at Bunda Mulia University, Indonesia.