OPEN ACCESS
ISSN 2356-5462
https://journals.telkomuniversity.ac.id/ijoict

Intl. Journal on ICT
Vol.11, No. 1, Jun 2025. pp. 64-78
doi:10.21108/ijoict.v11i1.9102

# Ant Colony Optimization for Low-Rank Factorization with DNN on People Counting IoT using Environmental Sensors

Ganendra Zefanya Patty[1], Muhammad Faris Fathoni[2], Aji Gautama Putrada[3]

[1,2,3] *School of Computing Telkom University, Bandung, Indonesia*

ganendrazefanyapatty@student.telkomuniversity.ac.id

**Abstract**

People counting in the Internet of Things (IoT) is essential for smart buildings, enabling efficient control of lighting, air conditioning (AC), and other systems. This study proposes a lightweight, privacy-preserving edge computing-based people counting system using environmental sensors and a Deep Neural Network (DNN) optimized with Low-Rank Factorization (LRF). The system runs in real time on edge devices with low latency and efficient resource use, following a cyclic workflow of data acquisition, model optimization, and performance evaluation. Ant Colony Optimization (ACO) is applied for hyperparameter tuning, yielding optimal parameters: 128 neurons, Adam learning rate of 0.005, and batch size of 8. Results show LRF reduces model size by over 6% without sacrificing accuracy, and the proposed DNN+ACO achieves accuracy, precision, recall, and F1-score of 0.98, 0.99, 0.94, and 0.97, respectively, outperforming a standard DNN and a Random Forest baseline. The model effectively addresses class imbalance, with recall for counts 0, 1, 2, and 3 of 1.00, 1.00, 1.00, and 0.78. This work introduces a novel integration of ACO, DNN, and LRF for privacy-preserving, efficient people counting, demonstrating superior performance and reduced model size for real-world smart building applications.

**Keywords:** Deep Neural Network, Internet of Things, Low-Rank Factorization, People Counting, Ant Colony Optimization

## I. INTRODUCTION

People counting Internet of things (IoT), which plays a role in counting people in a room based on sensor values, is becoming a vital part of smart buildings because it affects other IoT systems that regulate things like lighting and air conditioning (AC), so the impact is on efficiency [1]. Some solutions use CCTV cameras and computer vision algorithms for people counting, which offer high accuracy, but pose serious challenges related to privacy and computing resource consumption [2]. In addition, image solutions are less ideal for implementation on edge devices with limited power and processing capabilities [3]. A solution is needed that can perform people counting without threatening personal privacy and is lightweight.

To address this challenge, environmental sensors such as temperature, humidity, and $CO_2$ have been studied as a non-invasive alternative that can indicate human presence without recording visual identity. Longo et al. [4], who conducted a study on crowdsensing using environmental sensors to prevent privacy breaches, stated that using environmental sensors is more secure in terms of privacy than using cameras. The study used random forest for people counting classification and obtained an accuracy of 0.95. Akhter et al. [5] conducted pedestrian

Received on 14 May 2025. Revised on 19 Jun 2025. Accepted and Published on 21 Jun 2025.

counting involving environmental sensors such as temperature, humidity, pressure, CO2, and total volatile organic compound (TVOC). The use of these sensors makes the product of the study have dual functionality: as people counting and as air quality index (AQI) monitoring. However, this approach still requires a high-complexity predictive model to extract patterns from the sensor data and produce decisions with good performance.

Several studies have used deep learning for people counting using environmental sensors. El Amine et al. [6] stated that deep neural networks (DNN) offer excellent capabilities for modelling nonlinear relationships and complex patterns from sensor data. The study used DNN for people counting using Wi-Fi radar. Kamal *et al.* [7] stated that using DNN for people counting using environmental sensors can improve people counting performance by up to 0.95. The study also noted that the advantage of using environmental sensors is that the sensors are placed out of sight of the user. Unfortunately, traditional DNNs require high computing capacity, which can cause significant delays in edge systems.

Several studies have applied low-rank factorization (LRF) to deep learning to obtain efficient prediction models. Huang *et al.* [8] applied LRF to DNN with a biomedical magnetic resonance case study. Alternating LRF introduced by the study can improve the efficiency of the model as well as its performance. Sainath *et al.* [9] applied LRF to DNN with a large vocabulary continuous speech recognition (LVCSR) case study, where its application can reduce the number of parameters by 30-50%. The study used 50 hours of broadcast news, 400 hours of broadcast news, and 300 hours of switchboard telephony data. Therefore, there is a research opportunity to approach DNN architecture optimization through LRF techniques to reduce the number of parameters without sacrificing accuracy in the IoT people counting case study.

The main objective of this study is to create a high-efficiency and privacy-preserving environmental sensor-based real-time people counting system that can run directly on edge devices without cloud dependency. This study leverages LRF on a DNN model that performs people counting prediction to achieve this goal. Our first step is to use the people counting dataset from Kaggle, containing five sensors: temperature and humidity (DHT22), $CO_2$ (MQ-135), light (LDR), and sound (KY-037). Then we train an optimal DNN model to perform people counting prediction. We benchmark it with a random forest. The last step is to apply LRF to the DNN model. We use a testing methodology to find the optimal rank. We also measure the efficiency of the proposed model. This study also uses ant colony optimization (ACO) for hyperparameter tuning and obtains the optimal hyperparameters.

To the best of our knowledge, no research has applied LRF for efficient DNN on people counting IoT. The following list highlights our research contributions:

1. An ACO hyperparameter tuning algorithm for DNN that can improve the F1-score of the DNN model from 0.93 to 0.97.
2. A people counting IoT using environmental sensors using a novel DNN+ACO model and obtaining a more optimum accuracy than state-of-the-art studies, which is 0.98
3. An LRF for DNN with a case study of people counting IoT, where in LRF with optimum rank, the model size can be reduced from 42.3 KB to 35.2 KB without significant accuracy loss.

The remainder of this paper uses a systematic approach: Section II discusses state-of-the-art papers and highlights the contribution of our research. Then, Section III contains the research steps we took, and the formulas and theories involved. Furthermore, Section IV discusses two important things: first, this Section shows the test results, and second, it compares the results with the results in the state-of-the-art paper. Finally, Section V highlights the test results and how the results answer the research objectives.

Based on this background, this study proposes a novel real-time people counting system using environmental sensors. This system is optimized through a combination of Ant Colony Optimization (ACO) for hyperparameter tuning and Low-Rank Factorization (LRF) for model compression. Unlike previous studies that tent to focus on conventional optimization approaches or rely on visual data, this study integrates privacy-preserving sensor inputs into a DNN model that is compatible with edge devices. The main contribution of this study are as follows:

1. Presenting a novel combination of ACO, DNN, and LRF specially designed for low-latency people counting system in edge computing enviroments.
2. Providing a practical demonstration that a lightweight and privacy-conscious system using environmental sensors can outperform both visual-based models and large-scale machine learning models.
3. Providing a clear comparative evaluation of baseline methods, by showing improvements in prediction accuracy and resource efficiency in real-world edge testing.

## II. LITERATURE REVIEW

People counting systems are becoming important in developing smart cities and IoT systems. Akhter *et al.* [5] proposed a non-visual solution based on environmental sensors (temperature, humidity) that can monitor pedestrian density in real-time without sacrificing privacy. This system shows potential in efficiency and scalability, although it still has limitations in prediction accuracy in dense environments. Cheng *et al.* [10] said that the complexity of sensor data requires using predictive models that can handle non-linear patterns. DNNs are known to be effective in capturing complex relationships between input variables but have significant limitations regarding resource consumption. DNNs require many computationally intensive parameters, making them less than ideal for direct implementation on edge devices.

LRF is introduced as a method to compress DNN architectures to address this issue. This approach decomposes the weight matrix into a lower-dimensional representation without significantly sacrificing accuracy. Yang *et al.* [11] showed that LRF can drastically reduce the number of parameters, speed up the inference process by 30%, and produce a lightweight and energy-efficient model, making it suitable for edge-based systems. Furthermore, a study by Xie *et al.* [12] emphasized that combining environmental sensors and lightweight learning models can produce a privacy-aware, power-efficient, and accurate people counting system under various conditions. This study strengthens the argument that the Low-Rank DNN on Edge approach can be a reliable solution in the modern IoT ecosystem. Referring to the literature, this study combines environmental sensors, optimized DNN architecture via LRF, and edge-based local processing as a promising approach for an efficient, accurate, and privacy-preserving real-time people counting system.

In addition to the model side, communication and data processing performance in IoT systems are greatly influenced by the network architecture used. Borsatti *et al.* [13] studied the performance of the MQTT protocol in IoT-to-Cloud communication. They found that although MQTT is efficient, cloud-based systems still face bottlenecks, especially for response time-sensitive applications. Viegas *et al.* [14] also evaluated latency on the ThingSpeak platform and concluded that cloud-based response times are not fast enough for real-time system needs. These findings are supported by Zen *et al.* [15], who analyzed cloud infrastructure in the context of time-critical IoT applications. They suggested edge computing architecture design as the main approach to reduce latency and increase reliability.

## III. RESEARCH METHOD

This study aims to develop an edge computing-based people counting system using environmental sensors and a Deep Neural Network (DNN) model optimized using the LRF technique. The system is designed to operate in real-time on edge devices with low latency and efficient resource consumption. In general, the system's work process is divided into three main stages, namely (1) data acquisition and pre-processing, (2) model development and optimization, and (3) overall system performance evaluation. The system runs automatically on edge devices and follows a cyclic workflow to detect the number of people continuously. Fig. 1 shows the overall system architecture, which is divided into three main stages: data acquisition and pre-processing, model development and optimization, and system evaluation. The first stage includes collecting

environmental data such as $CO_2$ levels and temperature, which are then normalized and further processed through feature engineering techniques. In the second stage, the processed data is used to train the DNN model, which is then optimized using the LRF technique to produce a lighter and more efficient model to run on edge devices. The final stage involves evaluating the performance of the compressed model running on edge devices by measuring prediction accuracy and resource usage (CPU and RAM), then comparing it to the performance of a standard DNN model.

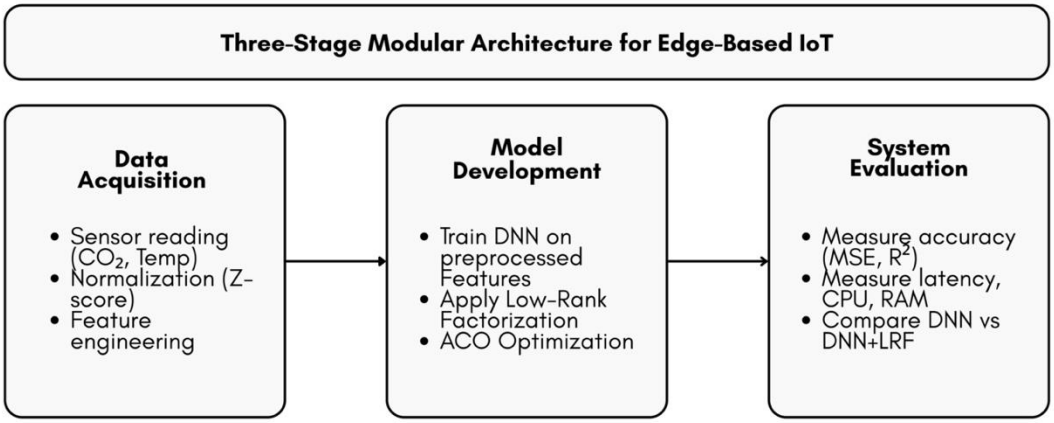


Fig 1. Modular architecture of the proposed edge-based people counting system. The system consists of three stages: (1) data collection and preprocessing using environmental sensors ($CO_2$, temperature, humidity, sound, light); (2) training and fine-tuning of the DNN model using ACO for hyperparameter tuning and LRF for compression; and (3) evaluation on edge devices to test predictions and resource efficiency.

Furthermore, this process runs iteratively to ensure continuous monitoring. Table 1 summarizes the components, inputs, outputs, and functions of each step in the system to provide a more structured understanding of each stage.

TABLE I

SYSTEM COMPONENT BREAKDOWN

| Step | Component | Input | Output | Description |
|---|---|---|---|---|
| 1 | Initialization | — | Sensor-ready state | Activates environmental sensors and system modules |
| 2 | Activity Detection | Sensor readings | Activity flag (Yes/No) | Detects environmental changes to trigger data flow |
| 3 | Data Collection | Trigger signal | Raw sensor data | Records temperature, humidity, sound, light, $CO_2$ levels |
| 4 | Preprocessing | Raw data | Normalized feature vector | Cleans data, handles missing values, normalizes features |
| 5 | DNN Model | Normalized vector | Feature embedding | Predicts occupancy patterns based on sensor input |
| 6 | LRF | DNN output | Compressed feature space | Reduces parameters to speed up inference on edge |
| 7 | People Count Estimation | Latent representation | Estimated count | Predicts number of people using regression |
| 8 | Output Logging | Estimation result | Display/log update | Stores results and optionally displays them |
| Loop | System Reset | — | — | Repeats process for continuous sensing |

### *A. The People Counting IoT and Dataset*

The system starts with the initialization of the environmental sensors. The device continuously monitors the surrounding environmental conditions. The system remains in a low-power standby mode if there is no significant change. Once a change is detected, the process continues with data acquisition. Data is collected from an enclosed space using five sensors: temperature and humidity (DHT22), $CO_2$ (MQ-135), light (LDR), and sound (KY-037) [16]. Data is recorded every 10 seconds for 30 minutes per session, with the varying number of people (0–3). Ground truth is recorded manually. Each sensor data vector is represented as:

$$x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n \quad (1)$$

Pre-processing includes data cleaning (removing outliers), imputation of missing values (with interpolation), and normalization using Z-score [17]. The formula for calculating the Z-score is as follows:

$$x'_i = \frac{x_i - \mu_i}{\sigma_i} \quad (2)$$

Another pre-processing stage is engineering temporal features, such as moving averages and delta features across time. The output of this stage is a feature vector x′ that is ready for use by the predictive model.

### *B. DNN and LRF*

DNN is an extension of an artificial neural network (ANN), namely an ANN that has more than one hidden layer [18]. One or more of these hidden layers function as feature learning, extracting high-level information from features. One of the remaining hidden layers functions as a decision maker [19]. Like ANN, neurons in DNN consist of weights and biases, where the values of the weights and biases are adjusted during the training process [20]. The adjustment reduces loss, namely the error between the actual and predicted values [21]. Optimization methods such as Adam can adjust weights with high performance within the epoch limit set by the programmer [22].

Continuing from pre-processing, the pre-processed data is then fed into a lightweight DNN model. The formula is as follows:

$$h = f(x'; \theta) \quad (3)$$

The model consists of 3 main layers: input layer (8 neurons), hidden layer (16 neurons, ReLU), and a single output layer. Training uses Adam optimizer and categorical cross-entropy loss function, with an early stopping mechanism [23]. Table II provides a summary of the DNN model. The number of neurons in the input layer depends on F because its value depends on the result of feature selection. We use feature importance from extra trees classification as the feature selection method [24].

TABLE II

THE ORIGINAL DNN ARCHITECTURE

| Layer | Type | Number of Neurons | Activation Function |
|---|---|---|---|
| Input Layer | Dense | F | Linear |
| Hidden Layer 1 | Dense | 16 | ReLU |
| Output Layer | Dense | 4 | SoftMax |

LRF is one of the model compression methods in DNN, where like other model compression methods, its goal is to shrink the DNN model while maintaining its accuracy [25]. LRF works by changing the parameter matrix of DNN into a multiplication of two or more matrices [26]. In this way, the number of parameters in DNN can be reduced without changing the weights and bias operations of the original DNN. The fully connected layer is compressed using LRF to adjust the model to edge devices' limitations. The model transforms the basic parameter matrix (H) into two new parameter matrices, U and V [27]. The formula is as follows:

$$H \approx U \cdot V^{\mathrm{T}} \tag{4}$$

$$U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}, \qquad r \ll min(m, n) \tag{5}$$

where r is the rank, the programmer determines the optimum model size, which is the smallest one without significant accuracy loss. The factorization result is used in the linear regression function to estimate the number of people:

$$\hat{y} = w^{\mathrm{T}} \cdot vec(U \cdot V^{\mathrm{T}}) + b \tag{6}$$

This model produces a lighter and faster estimation without sacrificing significant accuracy. Fig. 2 shows the architecture of the people counting IoT, which contains classification by a DNN model and compression by an LRF.

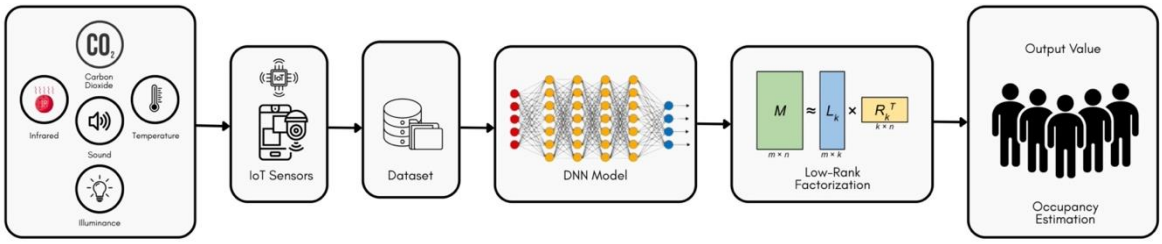


Fig. 2. The DNN-based people counting IoT with LRF architecture.

## C. *ACO for optimum hyperparameter tuning*

ACO is a swarm intelligence inspired by ants colonizing and foraging [28]. In ACO for hyperparameter tuning for DNN, each artificial ant represents a candidate set of hyperparameters, namely the number of neurons, the learning rate of the Adam optimizer, and the batch size [29]. ACO explores the hyperparameter value space based on a probabilistic decision rule that uses a pheromone trail guided by the encoded quality of the previous results [30]. We use F1-score as the fitness function. ACO uses the value to update the pheromone value and make the artificial ants aware of the more profitable position in the feature space [31]. With this method, ACO becomes a more effective method than methods such as grid search on problems that have infinite fitness boundaries [32].

The probability of an ant $k$ walking from $i$ to $j$ ($P_{i,j}^{k}$) is as follows:

$$P_{i,j}^{k} = \frac{[\tau_{i,j}]^{\alpha}}{\sum_{l \in \aleph}[\tau_{i,l}]^{\alpha}} \tag{7}$$

where $\tau_{i,j}$ is the pheromone level at edge $i, j$, $\alpha$ is the importance level of the pheromone, and $\aleph$ is the allowed neighborhood.

The two main aspects of evaluation are model accuracy and model efficiency comparison. The model's accuracy in estimating the number of people is evaluated using two main metrics: accuracy, precision, recall, and F1-score. Accuracy shows the ratio of data correctly predicted by the machine learning model to all available data. Precision shows the model's predictive ability, namely, seeing how much the ratio of true positives (TP) in all data is considered positive by the machine learning model. Recall shows the ratio of TP in actual positives. F1-score is the harmonic average between precision and recall. Accuracy loss is the ratio between the decrease in accuracy due to model compression and the original model's accuracy.

Regarding efficiency, the metrics that measure system performance are model size and compression ratio (CR). Model size is the size of the DNN on the hard drive, where the unit is kilobytes (KB). CR is the ratio between the size of the DNN model after applying LRF and the size of the original DNN model. This evaluation also compares the performance of the regular DNN model and the DNN model optimized using the LRF technique. Then, resource consumption. This evaluation includes memory usage (KB) during the inference process. This evaluation aims to determine how well the system can run sustainably on edge devices without overheating or performance degradation.

To assess the benefits of the LRF technique, a comparative test was conducted between the baseline model (without optimization) and the compressed model. The aspects compared include model size (number of parameters), inference time, accuracy, and resource consumption. The goal is to prove that LRF optimization can improve efficiency without significantly decreasing prediction performance. Then, each experiment was repeated 10 times, and the results were averaged to obtain stable values. All evaluations were conducted in a closed environment with controlled network conditions and room temperature to ensure consistent results.

## IV. RESULTS AND DISCUSSION

### *A. Results*

The first testing step is to obtain the “Room Occupancy Estimation Data Set” dataset uploaded by ANANTH R on Kaggle. The dataset contains 10,129 data items, 16 features, and four labels. The labels are numeric values 0, 1, 2, and 3, indicating the number of people in the room. The features contain values from four temperature sensors, four light sensors, four sound sensors, one CO2 sensor, one slope of the CO2 sensor value, and two passive infrared (PIR) sensors. We apply normalization to the features and then apply feature selection using feature importance from extra trees classification. The result of feature selection leaves four out of 16 features, namely three light sensors and one PIR sensor. Next, we divide the dataset, 80% for training and 20% for testing.

We first train the benchmark model, which is the random forest model. The accuracy of the benchmark model is 0.995. The next step is to train the DNN model, where the loss uses categorical cross-entropy, the optimizer uses Adam, the metric uses F1-score, the epoch is 30, the batch size is 16, and then we use 20% of the training data for validation data. The accuracy result of the original DNN is 0.986. Next, we perform hyperparameter tuning of the DNN model using ACO. The number of artificial ants from ACO is 5, the iteration is 10, and the evaporation rate is 0.5. Fig. 3 shows the fitness curve for 10 iterations, where in the 4th iteration, the ACO process has found a plateau at the F1-score value of 0.96.

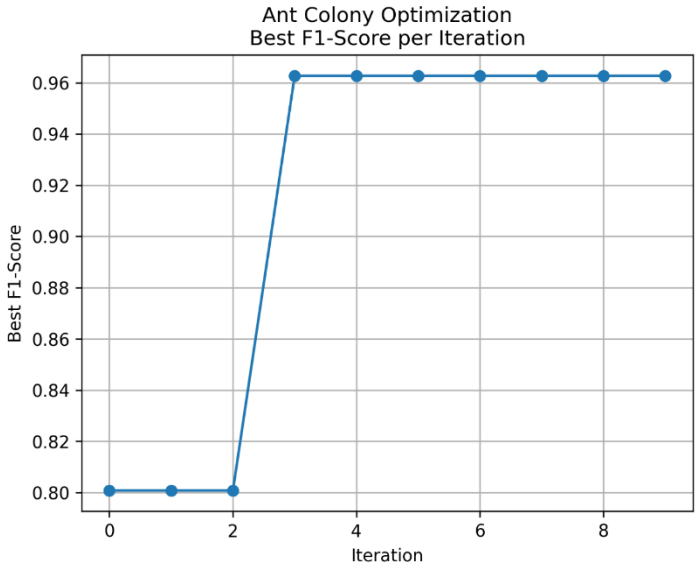


Fig 3. Fitness curve of the hyperparameter tuning process using ACO for 10 iterations. The curve shows that the process converges at the 4th iteration, with the F1-score reaching a stable value of 0.96. This indicates that ACO works effectively in optimizing parameters in the DNN model.

In the best fitness ACO results, the optimum hyperparameters are 128 neuron units, an Adam learning rate of 0.005, and a batch size of 8. Fig. 4 shows a bar chart comparing random forest, DNN, and DNN+ACO. DNN+ACO performs best with accuracy, precision, recall, and f1-score of 0.98, 0.99, 0.94, and 0.97. The recall value of 0.83 is obtained from the macro average of four classes, namely labels "0," "1," "2," and "3," which represent the number of people in the room. The recall values for each label are 1.0, 1.0, 1.0, and 0.8. The support values are 78, 3, 5, and 4, respectively. This shows two things: People counting performs better when fewer people are in the room. Second, the number of labels in the dataset affects the model's decision-making performance. We have tested each model several times and in each test, the model performance is consistent. This shows that the functions involved are deterministic and not probabilistic, there is no standard deviation and the difference in each performance is significant.

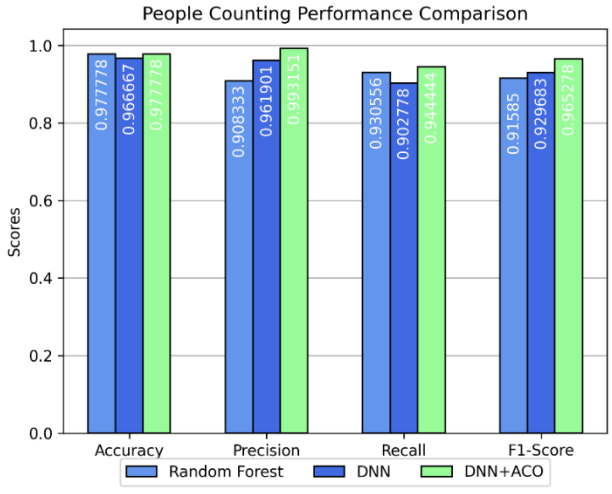


Fig 4. Comparison of prediction performance of three models: Random Forest, standard DNN, and DNN optimized using ACO. Evaluation is done based on Accuracy, Precision, Recall, and F1-score metrics. The results show that the DNN+ACO model excels in all evaluation metrics, with the most significant improvement seen in the F1-score value.

Next, we conducted a test to find the optimum rank for applying LRF to DNN. Fig. 5 shows the test on seven ranks: 1, 2, 4, 8, 16, 32, and 64. We conducted each test five times. For ranks 1 to 32, the test showed no different results, also according to the t-test, the average of the results did not have a significant difference. In other words, the accuracy loss from rank 32 to rank one is not significant. While the accuracy loss from ranks 32 and 64 to rank one is significant. On the other hand, by using rank 32, the model size can be reduced from 28.6 KB to 26.6 KB. This means the rank 32 is optimum because it can minimize model size without significant accuracy loss.

The dataset used in this study is the Room Occupancy Estimation Dataset from Kaggle, which is a fairly reliable basis for training and testing indoor people counting systems based on environmental data. However, there are several limitations that need to be considered. First, the number of people recorded in each session is limited to a maximum of three individuals. This may be less representative of real conditions in dense environments such as classrooms or shopping centers. Second, there is a class imbalance, where data tends to be more for those with a small number of occupants. This imbalance can affect the model training process, as seen from the lower recall score for class "3" (0.8), compared to other classes that achieve a perfect score (1.0).

Finally, although the environmental sensors used are privacy-preserving because they do not record visual data, their accuracy can vary depending on the layout of the room and the surrounding environmental conditions. Therefore, the application of this model in other environments may require adjustments or retraining according to the needs of a particular domain.

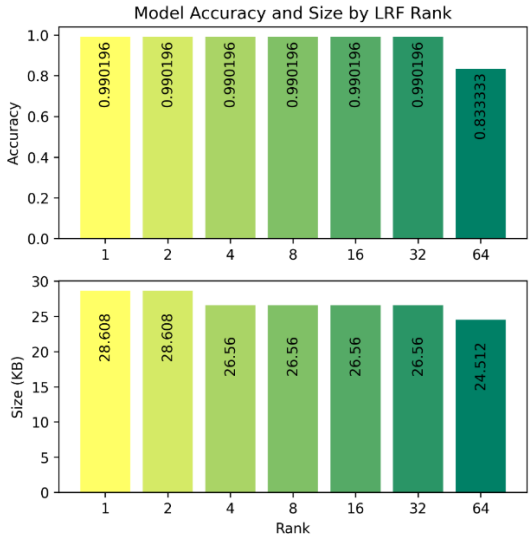


Fig 5. Trade-off analysis between model rank and accuracy in applying Low-Rank Factorization (LRF). Rank 32 is identified as the most optimal point because it provides the best balance between model size efficiency and maintaining accuracy. At this point, the model size is successfully reduced from 28.6 KB to 26.6 KB without significant performance degradation.

### *D. Discussion*

Several studies have applied ACO to optimize DNN models in various case studies. Zhang *et al.* [33] applied ACO to DNN in a case study of capital cost optimization in mining projects, where the R-squared value increased to 0.992. Samriya *et al.* [34]applied ACO to DNN models with an intrusion detection system (IDS) case study, where the role of ACO in the case study was for feature selection. Sathya *et al.* [35] also applied ACO as a feature selection in a DNN model for cardiovascular disease (CVD) detection, where the feature selection could improve the accuracy and efficiency of the model. Our study showed that ACO on DNN in the IoT people counting case study could improve model performance, especially its F1-score value from 0.93 to

0.97. The contribution of our study is a DNN+ACO model that can improve the performance of an IoT people counting system. Table III compares these studies with our study and highlights the contributions of each study.

TABLE III

RESEARCH CONTRIBUTION IN DNN+ACO FOR PEOPLE COUNTING IOT

| Reference | DNN+ACO | People Counting IoT | Research Contribution |
|---|---|---|---|
| Zhang et al. [33] | √ | x | Capital cost prediction in mining using ACO-tuned DNN |
| Samriya et al. [34] | √ | x | Feature selection using ACO in intrusion detection systems (IDS) with DNN |
| Sathya et al. [35] | √ | x | Improved accuracy in cardiovascular disease (CVD) detection using ACO with DNN |
| Proposed Model | √ | √ | Improves F1-score in people counting IoT from 0.93 to 0.97 using integrated ACO with DNN |

Several studies use computer vision for people counting, and several studies use environmental sensors. Then, among these studies, different machine learning models are used. El Amine *et al.* [6]used environmental sensors for people counting, namely Wi-Fi radar, where, using 3D-CNN, the study obtained an accuracy of 0.89. Kalita *et al.* [36] used computer vision-based people counting with several deep learning models. The best model from the study was You Only Look Once (YOLO) v8 with an accuracy of 0.9. Padmashini *et al.* [37] also used computer vision-based people counting with the DNN model. Like the previous study, the study obtained an accuracy of 0.9. Kamal *et al.* [38] used several environmental sensors, namely $CO_2$, LPG, $NO_2$, and $SO_2$ sensors, and obtained an accuracy of 0.95. The method compares several models, such as random forest and bagging, and then DNN is the model with the best performance. Our study uses four environmental sensors: $CO_2$, temperature, infrared, and illumination sensors, whereby using DNN+ACO, the prediction accuracy of our model is 0.98. The contribution of our study is an IoT people counting using environmental sensors using a novel DNN+ACO model and obtaining a more optimal accuracy than state-of-the-art studies, which is 0.98. Table IV compares these studies with our study and highlights the contributions of each study.

TABLE IV

RESEARCH CONTRIBUTION IN PEOPLE COUNTING IOT USING COMPUTER VISION-BASED AND ENVIRONMENTAL SENSORS AND VARIOUS MACHINE LEARNING MODELS

| Reference | Sensor Type | Model | Sensor Names | Accuracy |
|---|---|---|---|---|
| El Amine et al. [6] | Environmental | 3D-CNN | Wi-Fi Radar | 0.89 |
| Kalita et al. [36] | Computer Vision | YOLOv8 | Camera | 0.90 |
| Padmashini et al. [37] | Computer Vision | DNN | Camera | 0.90 |
| Kamal et al. [38] | Environmental | DNN | $CO_2$, LPG, $NO_2$, $SO_2$ | 0.95 |
| Proposed Model | Environmental | DNN+ACO | $CO_2$, temperature, infrared, light | 0.98 |

Several studies have applied LRF to DNN in several IoT case studies. Huang et al. [8] applied LRF to DNN with a biomedical magnetic resonance case study. Alternating LRF introduced by the study can improve the efficiency of the model as well as its performance. Sainath *et al.* [9] applied LRF to DNN with a speech recognition case study, where LRF can reduce the number of parameters in the model by 30–50%. Chen *et al.* [39] applied a new type of LRF that minimizes the number of parameters in the DNN model without reducing the model performance. The study stated that LRF can make a DNN model more compact for smartphone deployment. Pan *et al.* [40] introduced LRF to reduce the number of parameters in the DNN model with a multi-view data processing case study, namely, data collection and execution from several different perspectives or methods. This method can improve performance by 5–10%. Our study is the first to introduce LRF to reduce the number of parameters in DNN for people counting IoT, where the optimum rank is 32. The contribution of our study is an LRF for DNN with a case study of people counting IoT, where in LRF with optimum rank, the model size can be reduced from 28.6 KB to 26.6 KB without significant accuracy loss. Table V shows the comparison of state-of-the-art LRF on DNN, including our research and all the research contributions.

Although Random Forest (RF) performs well with an accuracy of 0.995 on this dataset, it is less flexible for application on edge devices due to its large model size and computational complexity. Ensemble models like RF require many decision trees, which, while fast for inference, require more memory and are difficult to compress.

TABLE V

RESEARCH CONTRIBUTION IN LRF ON DNN FOR PEOPLE COUNTING IOT

| Reference | LRF on DNN | People Counting IoT | Research Contribution |
|---|---|---|---|
| Huang et al. [8] | √ | x | Biomedical application: improved efficiency & accuracy through alternating LRF |
| Sainath et al. [9] | √ | x | Speech recognition: 30–50% parameter reduction |
| Chen et al. [39] | √ | x | Efficient smartphone deployment using adaptive LRF without sacrificing performance |
| Pan et al. [40] | √ | x | Improved multi-view data processing by 5–10% via LRF |
| Proposed Model | √ | √ | Reduced model size from 42.3 KB to 35.2 KB with no significant accuracy loss |

In contrast, the DNN+ACO model — despite its slightly lower accuracy (0.98) — provides a much better balance between performance, memory efficiency, and adaptability. ACO helps improve DNN performance by optimally adjusting hyperparameters based on real-world evaluation feedback (such as F1-score), resulting in a lightweight and efficient inference pipeline. Additionally, when combined with compression using LRF, the model size and memory requirements can be further reduced, making it well-suited for real-time IoT applications that require fast response time and low energy consumption.

## V. CONCLUSION

This study aims to demonstrate the effectiveness of applying LRF to Deep Neural Networks (DNN) for a low-latency edge-based people counting system using environmental sensors. We use ACO for hyperparameter tuning to obtain the optimum hyperparameters. Experimental results support the claim that LRF significantly

reduces the model size, while maintaining a high level of prediction accuracy. ACO in hyperparameter tuning obtains the optimum hyperparameters: the number of neurons as many as 128 units, Adam learning rate of 0.005, and batch size of 8. Then DNN + ACO is proven to perform better than DNN without ACO and the state-of-the-art random forest model with accuracy, precision, recall, and F1-score of 0.98, 0.99, 0.94, and 0.97. This is while overcoming the imbalance problem in the dataset with recall for counts 0, 1, 2, and 3, of 1.00, 1.00, 1.00, and 0.78, respectively. Finally, we found that the optimum rank of LRF to reduce the number of parameters in DNN is 32, at which rank the model size is reduced from 28.6 KB to 26.6 KB without significant accuracy loss.

This finding is consistent with and strengthens previous results showing that low-rank techniques can optimize deep learning models for resource-constrained environments. Furthermore, future work from this study can extend the existing knowledge by demonstrating that people counting tasks, which traditionally rely on visual sensors, can be effectively performed using non-visual environmental data. This opens new opportunities for privacy-friendly and energy-efficient IoT applications in smart buildings and public spaces.

## DATA AND COMPUTER PROGRAM AVAILABILITY

The dataset used in this study is public and taken from the Kaggle site, specifically from the Room Occupancy Estimation Data Set provided by Ananthanarayanan Ramanathan. This dataset contains real environmental sensor data, such as temperature, humidity, $CO_2$, light, and noise levels, collected from indoor environments with varying numbers of occupants. It is used for the training and evaluation process of the deep learning model developed in this study.

All pre-processing scripts, model training, and evaluation notebooks used in this study are available on GitHub and can be accessed upon request or through the repository link provided. This aims to ensure the reproducibility and transparency of the experiments conducted.

Dataset source: Kaggle: https://www.kaggle.com/datasets/ananthr1/room-occupancy-estimation-data-set

## ACKNOWLEDGMENT

Sincere gratitude is expressed to the Directorate of Research and Community Service (PPM) of Telkom University Bandung, which has provided moral and material support and facilities that greatly support the implementation of this research. The assistance provided is an important foundation in developing ideas and implementing the system studied in this study. The author would also like to express his deep appreciation to Ananth Ramanathan *et. al.*, the compiler and provider of the Room Occupancy Estimation Dataset. The dataset has become a fundamental component in experimentation, model training, and evaluation of the performance of the developed system. Their contribution to sharing the dataset openly has enriched this research and opened wider opportunities for the development of data-based systems in the fields of people counting and the Internet of Things (IoT).

## REFERENCES

[1] E. P. Myint and M. M. Sein, "People detecting and counting system," in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, IEEE, 2021, pp. 289–290. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9391951/

[2] Y. Yoo, J. Suh, Y. Lee, S. Kang, B. Kim, and S. Bahk, "Privacy-preserving people counting with channel state information," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2020, pp. 753–755. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9289512/

[3] Z. Mo, S. Xia, S. Shen, S. Du, and Q. Liu, "A Cloud-Edge-Terminal Collaborative System for Image-Based Crowd Counting," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, IEEE, 2023, pp. 642–647. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10271914/

[4] S. Longo and B. Cheng, "Real-time privacy preserving crowd estimation based on sensor data," in *2016 IEEE International Conference on Mobile Services (MS)*, IEEE, 2016, pp. 95–102. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7787060/

[5] F. Akhter, S. Khadivizand, J. Lodyga, H. R. Siddiquei, M. E. E. Alahi, and S. C. Mukhopadhyay, "Design and development of an IoT enabled pedestrian counting and environmental monitoring system for a smart city," in *2019 13th International Conference on Sensing Technology (ICST)*, IEEE, 2019, pp. 1–6. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9047695.

[6] A. El Amine and V. Guillet, "Device-free people counting using 5 ghz wi-fi radar in indoor environment with deep learning," in *2020 IEEE Globecom Workshops (GC Wkshps*, IEEE, 2020, pp. 1–6. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9367393/

[7] U. Kamal, S. Ahmed, T. R. Toha, N. Islam, and A. B. M. Alim Al Islam, "Intelligent human counting through environmental sensing in closed indoor settings," *Mobile Networks and Applications*, vol. 25, pp. 474–490, 2020.

[8] Y. Huang *et al.*, "Alternating Deep Low-Rank Approach for Exponential Function Reconstruction and Its Biomedical Magnetic Resonance Applications," Aug. 14, 2024, *arXiv*: arXiv:2211.13479. doi: 10.48550/arXiv.2211.13479.

[9] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6655–6659. Accessed: May 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6638949/

[10] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A Survey of Model Compression and Acceleration for Deep Neural Networks," Jun. 14, 2020, *arXiv*: arXiv:1710.09282. doi: 10.48550/arXiv.1710.09282.

[11] H. Yang *et al.*, "Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 678–679. Accessed: May 13, 2025. [Online]. Available: http://openaccess.thecvf.com/content_CVPRW_2020/html/w40/Yang_Learning_Low-Rank_Deep_Neural_Networks_via_Singular_Vector_Orthogonality_Regularization_CVPRW_2020_paper.html

[12] C. Xie *et al.*, "Efficient deep learning models for privacy-preserving people counting on low-resolution infrared arrays," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13895–13907, 2023.

[13] D. Borsatti, W. Cerroni, F. Tonini, and C. Raffaelli, "From IoT to cloud: applications and performance of the MQTT protocol," in *2020 22nd international conference on transparent optical networks (ICTON)*, IEEE, 2020, pp. 1–4. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9203167/

[14] V. Viegas, J. M. Dias Pereira, P. M. Girão, and O. Postolache, "Study of latencies in ThingSpeak," *Study of latencies in ThingSpeak*, no. 1, pp. 342–348, 2021.

[15] K. Zen, S. Mohanan, S. Tarmizi, N. Annuar, and N. U. Sama, "Latency analysis of cloud infrastructure for time-critical iot use cases," in *2022 Applied Informatics International Conference (AiIC)*, IEEE, 2022, pp. 111–116. Accessed: May 13, 2025. [Online]. Available:

https://ieeexplore.ieee.org/abstract/document/9914601/

[16] M. A. R. Tarigan, A. G. Putrada, and R. L. Wicaksono, "ElGamal Homomorphic Encryption with SMOTE for PET in Occupancy Monitoring by XGBoost," in *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*, IEEE, 2025, pp. 1–6. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10933291/

[17] E. Gaudet, F. Persson, and M. Saidel, "Z-score-based posttest risk as an alternative risk metric to positive predictive value following positive noninvasive prenatal screening," *American Journal of Obstetrics and Gynecology*, vol. 232, no. 4, pp. 367–372, 2025.

[18] A. Pant and A. Kumar, "Design and implementation of deep neural network hardware chip and its performance analysis", Accessed: May 13, 2025. [Online]. Available: https://www.academia.edu/download/119133250/13_20709.pdf

[19] M. T. Islam and L. Xing, "Deciphering the feature representation of deep neural networks for high-performance AI," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024, Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10439639/?casa_token=TC8v3yz5f4kAAAAA:dyltdlwEUURq2PV467fUOYYo2lSOmqOcdxb8HnjeIza3eZKn48LYX4RitS1ggYmAlwAfz72PXBtz

[20] P. Cui and K. C. Wiese, "EvoDNN-Evolving Weights, Biases, and Activation Functions in a Deep Neural Network," in *2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, IEEE, 2022, pp. 1–9. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9863054/

[21] S. Jandial, A. Chopra, M. Sarkar, P. Gupta, B. Krishnamurthy, and V. Balasubramanian, "Retrospective Loss: Looking Back to Improve Training of Deep Neural Networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event CA USA: ACM, Aug. 2020, pp. 1123–1131. doi: 10.1145/3394486.3403165.

[22] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How do adam and training strategies help bnns optimization," in *International conference on machine learning*, PMLR, 2021, pp. 6936–6946. Accessed: May 13, 2025. [Online]. Available: https://proceedings.mlr.press/v139/liu21t.html

[23] Á. D. Reguero, S. Martínez-Fernández, and R. Verdecchia, "Energy-efficient neural network training through runtime layer freezing, model quantization, and early stopping," *Computer Standards & Interfaces*, vol. 92, p. 103906, 2025.

[24] G. Alfian *et al.*, "Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method," *Computers*, vol. 11, no. 9, p. 136, 2022.

[25] M. Kokhazadeh, G. Keramidas, V. Kelefouras, and I. Stamoulis, "Denseflex: A Low Rank Factorization Methodology for Adaptable Dense Layers in DNNs," in *Proceedings of the 21st ACM International Conference on Computing Frontiers*, Ischia Italy: ACM, May 2024, pp. 21–31. doi: 10.1145/3649153.3649183.

[26] K. Wu, Y. Guo, and C. Zhang, "Compressing deep neural networks with sparse matrix factorization," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 10, pp. 3828–3838, 2019.

[27] G. N. Iswara, A. G. Putrada, and H. H. Nuha, "Low-Rank Factorization for Edge Computing in Fall Detection with Wearable Sensors," in *2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, IEEE, 2024, pp. 549–554. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10913100/

[28] A. Mohammadi, S. Hajiaghajani, and M. Bahrani, "ACO-tagger: A Novel Method for Part-of-Speech Tagging using Ant Colony Optimization," Mar. 27, 2023, *arXiv*: arXiv:2303.16760. doi: 10.48550/arXiv.2303.16760.

[29] M. Lohvithee, W. Sun, S. Chretien, and M. Soleimani, "Ant colony-based hyperparameter optimisation in total variation reconstruction in X-ray computed tomography," *Sensors*, vol. 21, no. 2, p. 591, 2021.

[30] S. Mori, S. Nakamura, K. Nakayama, and M. Hisakado, "Phase transition in ant colony optimization," *Physics*, vol. 6, no. 1, pp. 123–137, 2024.

[31] Y.-H. Qiu *et al.*, "Non-Linearly Weighted Pheromone Updating for Ant Colony Optimization," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2024, pp. 653–658. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10831580/

[32] X. Huang, Y. Han, and Z. Hu, "An optimization ant colony algorithm based on fixed point theory," in *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, IEEE, 2022, pp. 912–916. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9873586/

[33] H. Zhang *et al.*, "Developing a novel artificial intelligence model to estimate the capital cost of mining projects using deep neural network-based ant colony optimization algorithm," *Resources Policy*, vol. 66, p. 101604, Jun. 2020, doi: 10.1016/j.resourpol.2020.101604.

[34] J. K. Samriya, R. Tiwari, X. Cheng, R. K. Singh, A. Shankar, and M. Kumar, "Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100746, Sep. 2022, doi: 10.1016/j.suscom.2022.100746.

[35] K. Sathya, R. Shanthi, J. Esther, C. Selvi, and M. Pandiyan, "Automated Cardiovascular Disease Prediction Using Ant Colony Optimization with a Deep Learning Improved Neural Network," in *Advances in Computational Intelligence for Health Informatics and Computer-Aided Diagnosis*, CRC Press, 2025.

[36] D. Kalita, A. K. Talukdar, S. Deka, and K. K. Sarma, "Vision-Based People Counting System for Indoor and Outdoor Environments using different Deep Learning Models," in *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, IEEE, 2024, pp. 1–6. Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10782123/

[37] M. Padmashini, R. Manjusha, and L. Parameswaran, "Vision based algorithm for people counting using deep learning," *International Journal of Engineering & Technology*, vol. 7, no. 3.6, pp. 74–80, 2018.

[38] U. Kamal, S. Ahmed, T. R. Toha, N. Islam, and A. B. M. Alim Al Islam, "Intelligent human counting through environmental sensing in closed indoor settings," *Mobile Networks and Applications*, vol. 25, pp. 474–490, 2020.

[39] T. Chen, J. Lin, T. Lin, S. Han, C. Wang, and D. Zhou, "Adaptive mixture of low-rank factorizations for compact neural modeling," 2018, Accessed: May 13, 2025. [Online]. Available: https://openreview.net/forum?id=r1xFE3Rqt7

[40] B. Pan, H. Che, C. Li, and H. Li, "Robust Deep Matrix Factorization with Low-rank and Hypergraph Learning for Multi-view Data Processing," *IEEE Transactions on Consumer Electronics*, 2025, Accessed: May 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10820964/?casa_token=QQnCvo2MY30AAAAA:Ri83zt0Rf6-upqEYcxB8eU6APR_m6hZsj-0f9Zq1v6xxfThZuhrr-kcAj-HzrLz9uoTiFgXvj5Mv