applied sciences

Article

Effect of Compressed Sensing Rates and Video Resolutions on a PoseNet Model in an AIoT System

Hye-Min Kwon and Jeongwook Seo *

Department of IT Transmedia Contents, Hanshin University, Osan-si 18101, Korea
* Correspondence: jwseo@hs.ac.kr

Abstract: To provide an artificial intelligence service such as pose estimation with a PoseNet model in an Artificial Intelligence of Things (AIoT) system, an Internet of Things (IoT) sensing device sends a large amount of data such as images or videos to an AIoT edge server. This causes serious data traffic problems in IoT networks. To mitigate these problems, we can apply compressed sensing (CS) to the IoT sensing device. However, the AIoT edge server may have poor pose estimation accuracy (i.e., pose score), because it has to recover the CS data received from the IoT sensing device and estimate human pose from the imperfectly recovered data according to CS rates. Therefore, in this paper, we analyze the effect of CS rates (from 100% to 10%) and video resolutions (1280 × 720, 640 × 480, 480 × 360) in the IoT sensing device on the pose score of the PoseNet model in the AIoT edge server. When only considering the meaningful range of CS rates from 100% to 50%, we found that the higher the video resolution, the lower the pose score. At the CS rate of 80%, we could reduce data traffic by 20% despite the degradation in pose score of less than about 0.03 for all video resolutions.

Keywords: Internet of Things; data traffic; compressed sensing; deep learning; edge computing; MobileNet v1; PoseNet

Citation: Kwon, H.-M.; Seo, J. Effect of Compressed Sensing Rates and Video Resolutions on a PoseNet Model in an AIoT System. *Appl. Sci.* **2022**, *12*, 9938. https://doi.org/10.3390/app12199938

Academic Editor: Dimitris Mourtzis

Received: 18 August 2022
Accepted: 28 September 2022
Published: 2 October 2022

1. Introduction

The Internet of Things (IoT) denotes a global infrastructure, enabling advanced services by interconnecting physical and virtual things based on existing and evolving interoperable information and communication technologies [1–4]. Artificial intelligence (AI) is defined as an artificial creation of human-like intelligence and designed to imitate human abilities to understand and solve problems [5,6]. Recently, the Artificial Intelligence of Things (AIoT), combining AI and IoT, has attracted widespread attention from researchers, because it enables more effective IoT operations and creates AI services based on IoT data [7]. The AIoT systems generally collect and process a large amount of data (audio, image, video, etc.), and then they analyze and train the data to provide AI services (smart home control, object detection, pose estimation, etc.). In [8], recent developments in AIoT systems and their applications supported by AI (i.e., machine learning/deep learning algorithms) were introduced. In addition, Chen et al. in [9] proposed an AIoT-based smart agricultural system that identifies pests by using YOLOv3 object detection built from the image from drones and mobile devices and predicts the occurrence of pests by using the LSTM built from environmental data from weather stations.

As mentioned before, the AIoT systems have to send and receive a large amount of data between things such as IoT devices and an AIoT edge server through the Internet. Inevitably, this causes serious data traffic problems, which may lead to time delays and unstable channel conditions (congestion, collisions, etc.) due to limited bandwidth [10,11]. Many research studies have been conducted to reduce the problems in the AIoT system [12–17]. A general distributed federated learning framework was presented to avoid the communication bottleneck of most centrally managed AIoT systems in [12]. Baliarsingh et al. in [13] proposed an algorithm using transform techniques for cardiac data compression to reduce the data traffic caused by a large amount of electrocardiogram data in the

Artificial Intelligence of Medical things (AIoMT) system and Makarichev et al. in [14] presented a discrete atomic transformation-based lossless image compression algorithm to reduce the resources required for image processing, storage, and transmission over the network in the IoT system. Djelouat et al. in [15] reviewed compressed sensing (CS)-based IoT applications and identified avenues for future CS-based IoT research. In [16], the authors presented a gradient CS method for image data reduction in unmanned aerial vehicles. Finally, we proposed a oneM2M-compliant AIoT system with the CS to reduce data traffic and analyzed the effects of CS rates on the performance of YOLOv5 object detection in [17].

Pose estimation is a research field in computer vision, which plays a prominent role in various applications, such as healthcare, sports sciences, animations, robotics, intelligent video analytics, etc. [18–21]. This aims to determine the position or spatial location of body keypoints (e.g., parts, joints) of a person from a given image or video in [22]. Siddiq et al. in [23], proposed an IoT technology integration device for a smart home system with a human posture recognition function using the kNN method, and human pose is detected through a PoseNet model. In [24], the authors showed an intelligent collaborative inference approach for a PoseNet model-based service in mobile edge computing networks. Kim et al. in [25] presented a oneM2M-compliant AIoT system with an AIoT edge device and a floating population-monitoring server applicable to commercial area analysis in a smart city, where the AIoT edge device estimates human poses using two PoseNet models built on MobileNet v1 and ResNet-50 backbone architectures. However, to the best of our knowledge, we could not find any literature considering the use of CS for data traffic mitigation in AIoT systems for pose estimation services.

Therefore, in this paper, we propose an AIoT system with the CS to efficiently provide pose estimation with a PoseNet model, which consists of an IoT-sensing device and an AIoT edge server. Additionally, we analyze the effect of CS rates and video resolutions in the IoT-sensing device on the pose score of the PoseNet model in the AIoT edge server. The main contributions of this paper are summarized as follows:

- This paper analyzed the effects of CS rates and video resolutions on the PoseNet model in the AIoT system. We used CS among various compression techniques to mitigate data traffic. We chose this because it reduces the amount of data by randomly selecting pixels according to the CS rate and recovers at a rate close to perfection even with a small number of pixels.
- In addition, the pose score of the PoseNet model was used to analyze the effect of CS on pose estimation and the effects of 10 CS rates and 3 video resolutions were analyzed.
- Through this analysis, we can see that CS can effectively mitigate data traffic, but that the pose score does not significantly decrease.

The rest of this paper is organized as follows. Section 2 describes the configuration and operation of the proposed AIoT system and explains the CS for data traffic mitigation and pose estimation with the PoseNet model in detail. Section 3 provides experimental results to analyze the pose scores of the PoseNet model according to CS rates and video resolutions. Finally, concluding remarks are presented in Section 4.

2. Proposed AIoT System with Compressed Sensing and Pose Estimation

2.1. Description of an AIoT System

The proposed AIoT system is shown in Figure 1, where an IoT-sensing device used a random sampling matrix function of the CS for data traffic mitigation, and an AIoT edge server used CS recovery and domain transformation matrix functions of the CS and a PoseNet model for pose estimation.

First, the IoT-sensing device extracts large amounts of data such as images or videos ($\mathbf{O}$) from a high-definition (HD) camera sensor and then uses a random sampling matrix function ($\boldsymbol{\Phi}$) to obtain a compressed image ($\mathbf{c}$) according to the CS rate (r). The compressed image is forwarded to the IoT client application modeled, as an application-dedicated node-application entity (ADN-AE) in oneM2M specifications [26]. Then, the IoT client application transmits the compressed image to an IoT edge middleware (MW) in the AIoT edge server. The IoT edge MW, modeled as an infrastructure node-common service entity (IN-CSE) in oneM2M specifications, forwards the compressed image to a CS recovery function. The CS recovery function generates a sparse transform domain representation ($\hat{x}$) of the original image $\mathbf{O}$ via Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) [27–29]. After a domain transformation matrix function ($\mathbf{\Psi}$) inversely transforms $\hat{x}$ to produce the recovered image ($\hat{\mathbf{O}}$), we estimate human poses from the recovered image $\hat{\mathbf{O}}$ using the PoseNet model with the MobileNet v1 architecture where the human keypoints on the body were used [30–32]. The MobileNet v1 function takes the recovered image with red, green, and blue (RGB) channels and outputs a feature map [25]. In the feature map, the keypoint heatmaps function predicts a heatmap as a binary classification task. The short-range offsets function improves keypoint positioning accuracy by predicting a short-range offset vector pointing from the image location within the keypoint disk to the keypoint of the nearest human instance. The mid-range pairwise offsets function predicts a separate pairwise mid-range 2D offset vector, which is the target regression vector for each directional edge connecting adjacent pair of keypoints in a human tree-structured kinematics graph. The multi-person pose decoding function uses Hough voting to compute heatmaps and short-range offsets into a 2D Hough score map and finally outputs the estimated human poses.

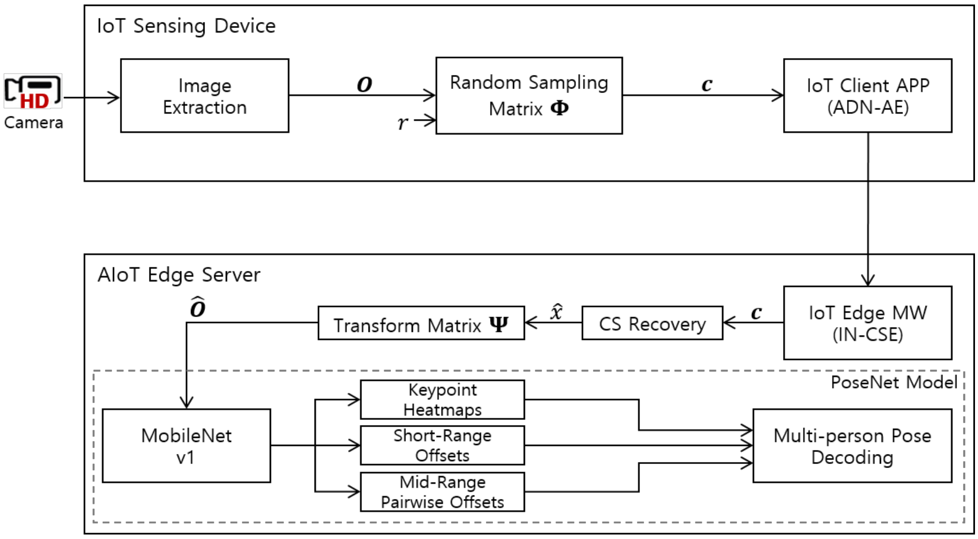


Figure 1. Proposed AIoT system with an IoT sensing device for CS and an AIoT edge server for CS recovery and pose estimation.

2.2. Compressed Sensing for Data Traffic Mitigation

Compressed sensing is a framework that samples a high-dimensional sparse signal at much lower rates than the Nyquist rate by sensing or measurement matrices (e.g., random sampling matrix), and this greatly improves the sampling efficiency [33,34]. In a recovery process (e.g., CS recovery), a sparse transform domain representation is produced by solving the optimization problem (i.e., L_1 norm minimization) [17,35–39]. Then, by its inverse transformation (e.g., transform matrix), we can obtain a recovered signal. We will explain the use of CS in data traffic mitigation in our AIoT system.

After the IoT-sensing device extracts a large amount of data $P \times Q$ original image $\mathbf{O}$, the $K \times P$ random sampling matrix $\mathbf{\Phi}$ compresses the original image $\mathbf{O}$ according to the CS rate $r = K/P$ to generate a $K \times Q$ compressed image $\mathbf{c}$, as given below

$$\mathbf{c} = \mathbf{\Phi}\mathbf{O} \tag{1}$$

where $\mathbf{\Phi}$ randomly selects pixels from the original image $\mathbf{O}$. The compressed image $\mathbf{c}$ will be sent to the AIoT edge server. In other words, we can mitigate the data traffic by generating the compressed image according to the CS rate.

In the AIoT edge server, the CS recovery reconstructs the sparse transform domain representation $\hat{x}$ from the compressed image $\mathbf{c}$ by solving the L_1 norm minimization with the above OWL-QN algorithm [27]. In this Algorithm 1, an object function is assumed to minimize $f(x) = \ell(x) + C\|x\|_1$ where $\ell(x)$ is the loss term, and C is a given constant. Additionally, $\diamond f(x)$ denotes the pseudo-gradient of f at x

$$\diamond_i f(x) = \begin{cases} \partial_i^- f(x) & \text{if } \partial_i^- f(x) > 0 \\ \partial_i^+ f(x) & \text{if } \partial_i^+ f(x) < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where right and left partial derivatives are given by $\partial_i^+ f(x)$ and $\partial_i^- f(x)$, $\mathbf{H}_j$ denotes the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) approximation to the inverse Hessian of the loss, and $\pi(x; v)$, denoted as the projection of x onto an orthant defined by v

$$\pi_i(x; v) = \begin{cases} x_i & \text{if } \sigma(x_i) = \sigma(v_i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\sigma(\cdot)$ denotes the sign function to take values in $\{-1, 0, 1\}$. The two-dimensional (2D) inverse discrete cosine transform (IDCT) matrix $\mathbf{\Psi}$ transforms the sparse transform domain representation $\hat{x}$ to obtain the recovered image $\hat{\mathbf{O}}$, which can be represented by

$$\hat{\mathbf{O}} = \mathbf{\Psi}\hat{x} \tag{4}$$

Algorithm 1 Orthant-Wise Limited-memory Quasi-Neton (OWL-QN)

```
choose initial point $x^0$
$M \Leftarrow \{\}, N \Leftarrow \{\}$
for $j = 0, 1, \ldots, MaxIters$ do
    Compute $w^j = -\diamond f(x^j)$
    Compute $e^j \Leftarrow \boldsymbol{H}_j w^j$ using $M$ and $N$
    $t^j \Leftarrow \pi(e^j; w^j)$
    Find $x^{j+1}$ with constrained line search
    if termination condition satisfied then
        Stop and return $x^{j+1}$
    end if
    Update $M$ with $m^j = x^{j+1} - x^j$
    Update $N$ with $n^j = \nabla\ell(x^{j+1}) - \nabla\ell(x^j)$
end for
```

The entries of $\hat{\mathbf{O}}$ are defined as

$$[\hat{\mathbf{O}}]_{pq} = \sum_{b=0}^{P-1} \sum_{d=0}^{Q-1} r_b r_d [\hat{x}]_{bd} \cos\frac{\pi(2p+1)b}{2P} \cos\frac{\pi(2q+1)d}{2Q} \tag{5}$$

where $0 \le p \le P-1$, $0 \le q \le Q-1$, and

$$r_b = \begin{cases} \frac{1}{\sqrt{P}}, & b = 0 \\ \sqrt{\frac{2}{P}}, & 1 \le b \le P-1 \end{cases} \quad \textit{and} \quad r_d = \begin{cases} \frac{1}{\sqrt{Q}}, & d = 0 \\ \sqrt{\frac{2}{Q}}, & 1 \le d \le Q-1 \end{cases} \tag{6}$$

2.3. Pose Estimation with a PoseNet Model

Pose estimation is the process of determining the configuration of the body from images or videos to estimate the positions of human keypoints, typically human joints, and associate them [40]. This is a highly popular research topic in the field of computer vision, which is widely used in human–computer interactions, gaming, virtual reality, video surveillance, etc. [41]. As mentioned above, we used the PoseNet model for pose estimation [25].

Figure 2 illustrates the PoseNet model with the MobileNet v1 backbone architecture proper for our AIoT edge server. When making pose estimations using the PoseNet model, 17 keypoints were detected, including 5 facial points, such as two eyes, two ears, and one nose, and 12 joints of the body, such as shoulders, elbows, wrists, hips, knees, and ankles. Then, the human pose was estimated by associating the closest two keypoints for human skeleton representation.

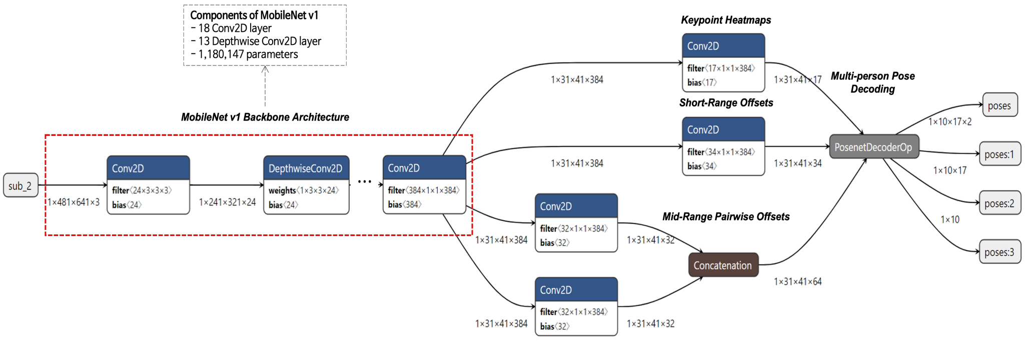


Figure 2. The PoseNet model with the MobileNet v1 backbone architecture.

The MobileNet v1 consists of 18 Conv2D layers and 13 Depthwise Conv2D layers. The first Conv2D layer receives an input image of (1 × 481 × 641 × 3) as a tensor of (batch × height × width × channels). Then, it outputs a feature map of (1 × 241 × 321 × 24) with a filter of (24 × 3 × 3 × 3) and a bias of (24). The second DepthwiseConv2D layer receives the feature map and outputs a new feature map with a weight of (1 × 3 × 3 × 24) and a bias of (24). The above process is repeated for other Conv2D and DepthwiseConv2D layers. The last Conv2D layer feeds the same feature map of (1 × 31 × 41 × 384) to keypoint heatmaps, short-range offsets, and mid-range pairwise offsets functions. The keypoint heatmaps function with a Conv2D layer predicts 17 heatmaps of (1 × 31 × 41 × 17) with a filter of (17 × 1 × 1 × 384) and a bias of (17). During training, the average logistic loss was considered for predicted heatmaps. To improve the keypoint position accuracy, the short-range offsets function with a Conv2D layer predicts short-range offset vectors pointing from the position within the keypoint disks to the keypoint of the closest human instance [25]. During training, the short-range offset prediction errors were penalized with the L_1 loss, and only the errors at the position in the keypoint disks were averaged and back-propagated. To connect pairs of keypoints belonging to each human instance, the mid-range pairwise offset function with two Conv2D layers and concatenation predicts separate pairwise mid-range 2D offset vectors, which are target regression vectors for each directed edge, connecting pairs of keypoints that are adjacent to each other in a tree-structured kinematic graph of the person. During training, the average L_1 loss of the regression prediction errors over the source keypoint disks was computed and back-propagated through the network. The multi-person pose decoded function group keypoints into detected human instances using Hough score maps and mid-range offset vectors. The Hough score maps can be obtained by aggregating heatmaps and short-range offsets through Hough voting.

The pose score, which is the average of keypoint scores, is calculated using Expected-Object Keypoint Similarity (OKS), keypoint heatmaps $p_k(x)$, and the Hough score maps [42]. The detected human instance with key point coordinates $y_{pj,pk}$ is called pj, and λpj is the square root of the bounding box area, which tightly contains all detected keypoints of the pj-th person instance. We define the Expected-OKS for the pk-th keypoint as:

$$s_{psj,psk} = E\left\{OKS_{psj,psk}\right\} = p_k\left(y_{psj,psk}\right) \int_{x\in\mathcal{D}_R\left(y_{psj,psk}\right)} \hat{h}_{psk}(x) \exp\left(-\frac{\left(x - y_{psj,psk}\right)^2}{2\lambda^2_{psj}\kappa^2_{psk}}\right) dx \tag{7}$$

where $\hat{h}_{psk}(x)$ is the Hough score maps normalized to $\mathcal{D}_R\left(y_{psj,psk}\right)$. The Expected-OKS keypoint level score is the confidence that a keypoint exists multiplied by the OKS localization accuracy confidence. The pose score is calculated as the instance-level score $s^h_{psj} = (1/K)\sum_k s_{psj,psk}$, followed by non-maximum suppression (NMS).

$$s_{psj} = (1/K) \sum_{k=1:K} s_{psj,psk} \left[\left\|y_{psj,psk} - y_{psj',psk}\right\| > r, \text{ for every } psj' < psj\right] \tag{8}$$

where $r = 10$ is the NMS radius.

The PoseNet model has 1,180,147 parameters calculated by $N_{\text{Conv } 2D} = \{(f_w \times f_h \times f_n) + 1\} \times f_i$ where f_w is the width of a filter, f_h is the height of a filter, f_n is the number of filters in the previous layer, and f_i is the number of filters.

3. Experimental Results

We describe experimental results to analyze the effect of CS rates and video resolutions on the PoseNet model in the proposed AIoT system. First, the IoT-sensing device was developed using a Raspberry Pi 4 Model B equipped with a C270 HD webcam to extract and compress real-time videos at 30 frames per second (fps). Its IoT client application was implemented by modifying an open-source IoT device platform, called &Cube-Thyme. For the convenience of the experiment, we used three 50-second video clips [43] with 1280 × 720, 640 × 480, and 480 × 320 resolutions pre-recorded by the IoT-sensing device. We used the same video extracted at three video resolutions, the network input size of the PoseNet model. Next, the AIoT edge server was developed by using a laptop equipped with a Google Coral USB Accelerator, where we installed PyCoral, TensorFlow Lite Runtime, and libedgetpu-max [44]. Its IoT edge middleware was implemented with an open source IoT server platform, called Mobius.

Figures 3–5 show the outputs of the PoseNet model for original and recovered images according to CS rates and video resolutions. It should be noted that the recovered images are typically contaminated by recovery errors, measured for various values of the CS rate. In Figure 3, the outputs of the PoseNet model are represented according to the CS rates when the input video resolution is 1280 × 720. Compared with Figure 3a, Figure 3b,d indicate that the PoseNet model similarly detects keypoints of human instances but misconnects some keypoints due to the recovery errors. The number of estimated human poses is slightly reduced in Figure 3e, whereas it is significantly reduced in Figure 3f, where only a few keypoints are detected. In Figure 4, the outputs of the PoseNet model are represented according to the CS rates when the input video resolution is 640 × 480. Comparing Figure 4a with Figure 3a, the PoseNet model can estimate the human pose of the largest human instance as well as that of the human instance climbing stairs. As shown in Figure 4d,e, it can detect and connect some keypoints of the largest human instance. In Figure 4f, it can not detect any keypoint of human instances due to severe recovery errors. In Figure 5, the outputs of the PoseNet model are represented according to the CS rates when the input video resolution is 480 × 360. Figure 5b,c show that most keypoints of the largest human instance are detected, but a few detected keypoints are misconnected. The number of estimated human poses in Figure 5e is lower than the number of those in Figure 4e. As in Figure 4f, no keypoint of human instances is detected in Figure 5f.

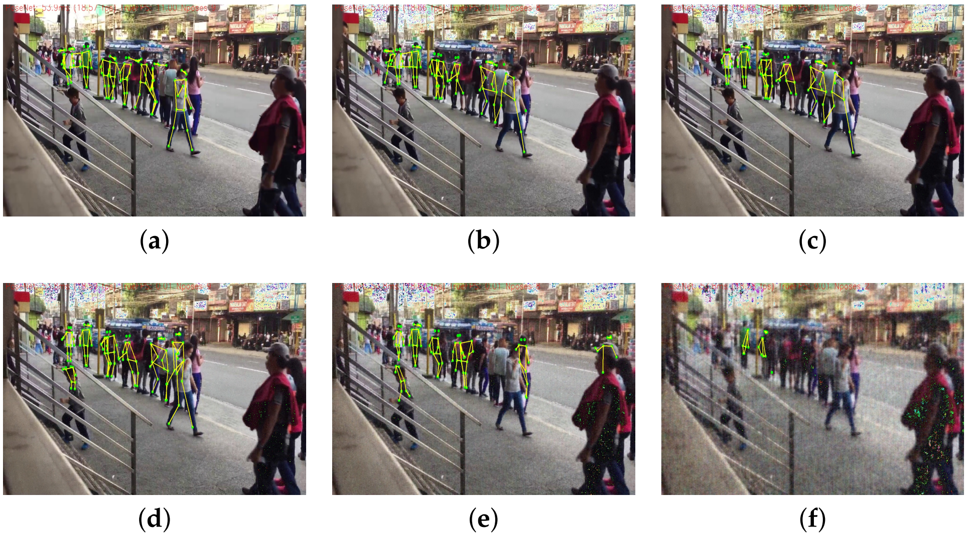

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)

Figure 3. Outputs of the PoseNet model for video resolution of 1280 × 720: (**a**) original image, (**b**) CS rate of 90%, (**c**) CS rate of 70%, (**d**) CS rate of 50%, (**e**) CS rate of 30% and (**f**) CS rate of 10%.

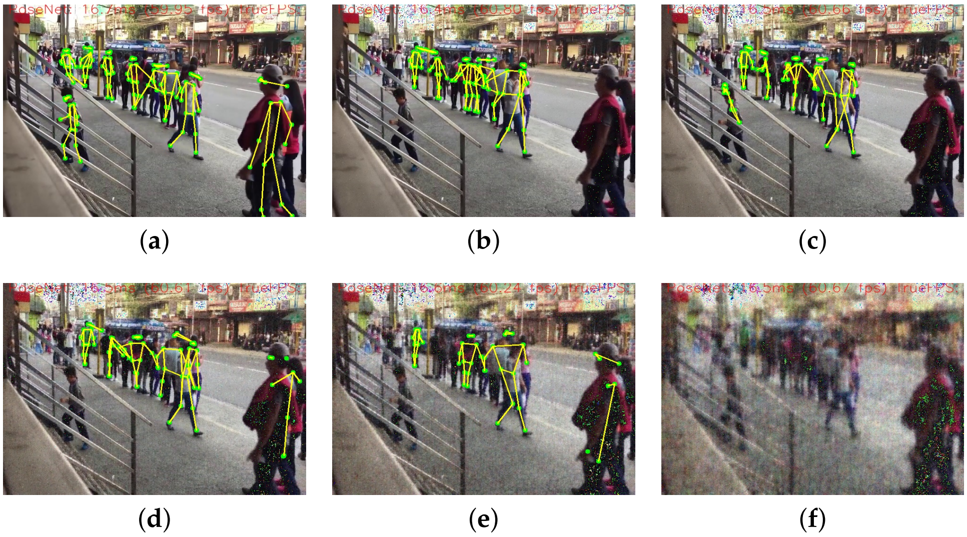
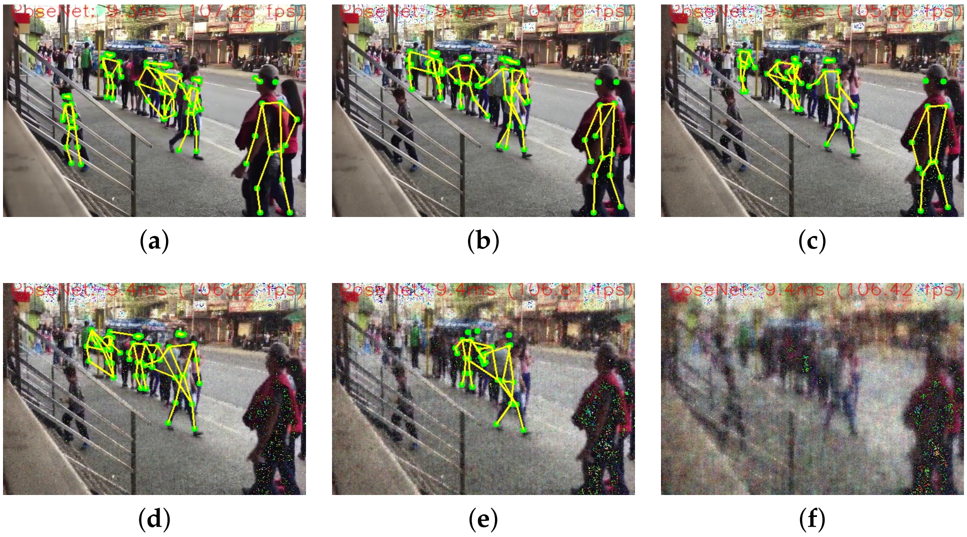

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)

Figure 4. Outputs of the PoseNet model for video resolution of 640 × 480: (**a**) original image, (**b**) CS rate of 90%, (**c**) CS rate of 70%, (**d**) CS rate of 50%, (**e**) CS rate of 30% and (**f**) CS rate of 10%

In Figure 6, pose scores of the PoseNet model are shown according to CS rates and video resolutions, where the pose score denotes the overall confidence in pose estimation and ranges between 0.0 and 1.0. The pose score is the average pose score for the entire frame of each video. Regardless of video resolution, the pose score gradually decreased as the CS rate decreased from 100% to 50%, but sharply decreased as the CS rate decreased from 50% to 10%. In addition, when the CS rate ranged from 100% to 50%, the PoseNet model showed better pose scores at lower video resolutions. At the CS rate of 80%, we could reduce the data traffic by 20%, but the pose score slightly decreased at all video resolutions. At the CS rate of 70%, the data traffic was reduced by 30%, even though each pose score for video resolutions of 1280 × 720, 640 × 480, and 480 × 360 decreased by 0.0208, 0.0467, and 0.0539, respectively. At the CS rate of 60%, we could reduce the data traffic by 40%, but the pose scores for video resolutions of 640 × 480, and 480 × 360 decreased by more than 0.05. Note that when the CS rate was 80% and the video resolution was 1280 × 720, we could reduce the data traffic by 20%, and the pose score only decreased by about 0.01.

Therefore, we think that the CS rate of 80% and the video resolution of 1280 × 720 are the most suitable for the PoseNet model in the proposed AIoT system.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 5. Outputs of the PoseNet model for video resolution of 480 × 360: (**a**) original image, (**b**) CS rate of 90%, (**c**) CS rate of 70%, (**d**) CS rate of 50%, (**e**) CS rate of 30% and (**f**) CS rate of 10%

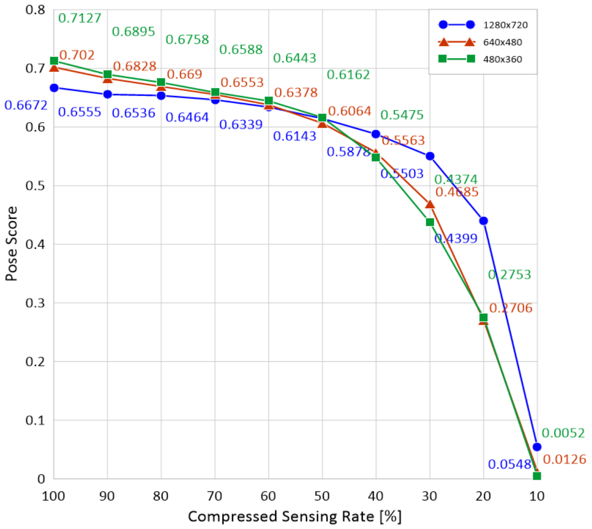


Figure 6. Pose scores of the PoseNet model according to CS rates and video resolutions.

4. Conclusions

In this paper, we proposed an AIoT system consisting of an IoT-sensing device and an AIoT edge server. The IoT-sensing device exploited a random sampling function of compressed sensing for data traffic mitigation, and the AIoT edge server fulfilled the CS recovery and domain transform functions of compressed sensing and a PoseNet model based on the MobileNet v1 architecture for pose estimation. To investigate the trade-off between data traffic mitigation and pose estimation accuracy (i.e. pose score), we analyzed the effects of CS rates and video resolutions on pose estimation in the AIoT edge servers in terms of pose score. In the meaningful range of CS rates from 100% to 50%, the higher the video resolution, the lower the pose score. At the CS rate of 80%, we could reduce data traffic by 20%, but the pose score decreased less than about 0.03 for all video resolutions. When the range of CS rates from 50% to 10%, the quality of the recovered image is very low, so it is difficult to properly estimate human pose and the pose score decreases sharply.

Therefore, when the CS rate is 80% and the video resolution is 1280 × 720, we can achieve the most effective data traffic mitigation and pose score.

Author Contributions: Writing—original draft, H.-M.K.; Writing—review & editing, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00959, Fast Intelligence Analysis HW/SW Engine Exploiting IoT Platform for Boosting On-device AI in 5G Environment).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository The data presented in this study are openly available in reference number [43].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ITU. ITU-T, Overview of the Internet of Things. 2019. Available online: https://www.itu.int/rec/T-REC-Y.2060-201206-I (accessed on 15 June 2012).
2. Weber, R. H.; Weber, R. *Internet of Things*; Springer: New York, NY, USA, 2010.
3. Chen, W.; Jeong, S.; Jung, H. WiFi-Based home IoT communication system. *J. Inf. Commun. Converg. Eng.* **2020**, *18*, 8–15.
4. Remesh Babu, K.R.; Preetha, K.G.; Saritha, S.; Rinil, K.R. An Energy efficient intelligent method for sensor node selection to improve the data reliability in Internet of Things networks. *KSII Trans. Internet Inf. Syst.* **2021**, *15*, 3151–3168.
5. ITU. ITU-T, Regulatory Responses to Evolving Technologies. 2020. Available online: https://digitalregulation.org/regulatory-responses-to-evolving-technologies/ (accessed on 28 December 2020).
6. Zhang, J.; Tao, D. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet Things J.* **2020**, *8*, 7789–7817. [CrossRef]
7. Revathy, R.; Raj, M.G.; Selvi, M.; Periasamy, J.K. Analysis of artificial intelligence of things. *Int. J. Electr. Eng. Technol.* **2020**, *11*, 275–280. [CrossRef]
8. Haroun, A.; Le, X.; Gao, S.; Dong, B.; He, T.; Zhang, Z.; Wen, F.; Xu, S.; Lee, C. Progress in micro/nano sensors and nanoenergy for future AIoT-based smart home applications. *Nano Express.* **2021**, *2*, 022005. [CrossRef]
9. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chang, C.Y.; Huang, Y.M. An AIoT based smart agricultural system for pests detection. *IEEE Access* **2020**, *2*, 180750–180761. [CrossRef]
10. Kang, J.; Eom, D. Offloading and transmission strategies for IoT edge devices and networks. *Sensors* **2019**, *19*, 835. [CrossRef]
11. Oh, A.S. A study on intelligent edge computing network technology for road danger context aware and notification. *J. Inf. Commun. Converg. Eng.* **2020**, *18*, 183–187.
12. Pinyoanuntapong, P.; Huff, W.H.; Lee, M.; Chen, C.; Wang, P. Toward scalable and robust AIoT via decentralized federated learning. *IEEE Internet Things Mag.* **2022**, *5*, 30–35. [CrossRef]
13. Baliarsingh, S.; Mohapatra, S.K.; Panda, P.K.; Mohanty, M.N. Cardiac data compression for reduced traffic on application of IoMT. In Proceedings of the 2022 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 9–11 March 2022, pp. 1–5.
14. Makarichev, V.; Lukin, V.; Illiashenko, O.; Kharchenko, V. Digital image representation by atomic functions: The compression and protection of data for edge computing in IoT systems. *J. Sens.* **2022**, *22*, 3751. [CrossRef]
15. Djelouat, H.; Amira, A.; Bensaali, F. Compressive sensing-based IoT applications: A review. *J. Sens. Actuator Netw.* **2018**, *7*, 337–348. [CrossRef]
16. Musić, J.; Orović, I.; Marasović, T.; Papić, V.; Stanković, S. Gradient compressive sensing for image data reduction in UAV based search and rescue in the wild. *Math. Probl. Eng.* **2016**, *2016*, 6827414. [CrossRef]
17. Kwon, H.; Hong, S.; Kang, M.; Seo, J. Data traffic reduction with compressed sensing in an AIoT system. *Comput. Mater. Contin.* **2022**, *70*, 1769–1780. [CrossRef]
18. Davoodnia, V.; Ghorbani, S.; Etemad, A. In-bed pressure-based pose estimation using image space representation learning. *arXiv* **2019**, arXiv:1908.08919.
19. Desmarais, Y.; Mottet, D.; Slangen, P.; Montesinos, P. A review of 3D human pose estimation algorithms for markerless motion capture. *Comput. Vis. Image Underst.* **2021**, *212*, 103275. [CrossRef]
20. Ma, X.; Su, J.; Wang, C.; Ci, H.; Wang, Y. Context modeling in 3d human pose estimation: A unified perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6238–6247.
21. Zhao, W.; Wang, W.; Tian, Y. GraFormer: Graph-oriented transformer for 3D pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 20438–20447.

22. Munea, T. L.; Jembre, Y. Z.; Weldegebriel, H. T.; Chen, L.; Huang, C.; Yang, C. The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation. *Int. J. Electr. Eng. Technol.* **2020**, *8*, 133330–133348. [CrossRef]
23. Li, X.; Qin, Y.; Zhou, H.; Zhang, Z. An intelligent collaborative inference approach of service partitioning and task offloading for deep learning based service in mobile edge computing networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *8*, 133330–133348. [CrossRef]
24. Siddiq, M.I.; Wibawa, I.P.D.; Kallista, M. Integrated Internet of Things (IoT) technology device on smart home system with human posture recognition using kNN method. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Sanya, China, 12–14 November 2021; Volume 1098, p. 42065.
25. Kim, M.; Hong, S.; Kang, M.; Seo, J. Performance comparison of PoseNet models on an AIoT edge server. *Intell. Autom. Soft Comput.* **2021**, *30*, 743–753. [CrossRef]
26. *oneM2M, Release 2 TS-0004 v3.11.2, Service Layer Core Protocol Specification*; oneM2M: Sophia Antipolis, France, 2019.
27. Andrew, G.; Gao, J. Scalable training of L^1-regularized log-linear models. In Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, USA, 20–24 June 2007; pp. 32–58.
28. Fathi, M.F.; Bakhshinejad, A.; Baghaie, A.; D'Souza, R.M. Dynamic denoising and gappy data reconstruction based on dynamic mode decomposition and discrete cosine transform. *Appl. Sci.* **2018**, *8*, 1515. [CrossRef]
29. Dai, M.X.; Chen, J.B.; Cao, J. L1-Regularized full-waveform inversion with prior model information based on orthant-wise limited memory quasi-Newton method. *J. Appl. Geophys.* **2017**, *142*, 49–57. [CrossRef]
30. Ghorai, A.; Gawde, S.; Kalbande, D. Digital solution for enforcing social distancing. In Proceedings of the International Conference on Innovative Computing & Communications, New Delhi, India, 31 May 2020.
31. Rishan, F.; De Silva, B.; Alawathugoda, S.; Nijabdeen, S.; Rupasinghe L.; Liyanapathirana, C. Infinity yoga tutor: Yoga posture detection and correction system. In Proceedings of the 2020 5th International Conference on Information Technology Research, Moratuwa, Sri Lanka, 2–4 December 2020; pp. 1–6.
32. Ha, T.Y.; Lee, H. Analysis on the mobile healthcare behavior using an artificial intelligence based pose estimation. *J. IEIE* **2020**, *57*, 63–69. [CrossRef]
33. Lu, C.; Chen, W.; Xu, H. Deterministic bipolar compressed sensing matrices from binary sequence family. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 2497–2517.
34. Lu, C.; Chen, W.; Xu, H. Binary sequence family for chaotic compressed sensing. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 4645–4664.
35. Mahdaoui, A.E.; Ouahabi, A.; Moulay, M.S. Image denoising using a compressive sensing approach based on regularization constraints. *Sensors* **2022**, *22*, 2199. [CrossRef]
36. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering: Machine Learning, Dynamical System, and Control*; Cambridge University Press: Cambridge, UK, 2019.
37. Hayashi, K.; Nagahara, M.; Tanaka, T. A user's guide to compressed sensing for communications systems. *IEICE Trans. Commun.* **2013**, *96*, 685–712. [CrossRef]
38. Kwon, H.; Ahn, H.; Lee, Y.; Sung, N.; Kang, M.; Seo, J. Real-time recovery and object detection of compressed sensed data. In Proceedings of the International Conference on Future Information & Communication Engineering, Seattle, WA, USA, 27–29 June 2022; Volume 13, pp. 90–92.
39. Compressed Sensing in Python. Available online: http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python (accessed on 20 July 2016).
40. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. Openpose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef] [PubMed]
41. Dang, Q.; Yin, J.; Wang, B.; Zheng, W. Deep learning based 2d human pose estimation: A survey. *Tsinghua. Sci. Technol.* **2019**, *24*, 663–676. [CrossRef]
42. Papandreou, G.; Zhu, T.; Chen, L.C.; Gidaris, S.; Tompson, J.; Murphy, K. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In Proceedings of the European conference on computer vision, Munich, Germany, 8–14 September 2018.
43. Pose-Estimation-Dataset. Available online: https://github.com/hyemin4845/Pose-Estimation-Dataset (accessed on 1 October 2022).
44. Coral. Available online: https://coral.ai/software (accessed on 9 April 2021).