MINSU: Precision Quantity Counter with DNN-based Volume Estimation

Shubhangi S.R. Garnaik*, DongYeop Lee*, Youngho Kim*, Jihoon Ryoo

State University of New York, Korea *Students with equal contribution
{shubhangisaile.garnaik,dongyeop.lee,youngho.kim.1,jihoon.ryoo}@stonybrook.edu

***Abstract*—This paper introduces *MINSU*, the first DNN (Deep Neural Network) based quantity counter specifically designed for smart factories. The innovative DNN design of *MINSU* positions it as a suitable solution for the emerging trend of fully automated, man-less smart factories. By periodically updating inventory status using live images, *MINSU* eliminates the need for manual, repetitive inventory-checking tasks. This system is easily adaptable to existing industrial environments, as it does not require the installation of new racks or modifications to the current factory floor plan. *MINSU* is designed to seamlessly navigate within the existing factory setup, achieving a level of performance that surpasses manual inventory counting methods. We have tested *MINSU* in a lab environment replicating a small-sized factory, using actual bolts as the target inventory. Our experiments demonstrated an accuracy of 84.33%, showcasing the potential for real-world applications in modern smart factories.**

***Index Terms*—Smart Factories, Quantity Counter, Volume estimation, Foreground subtraction.**

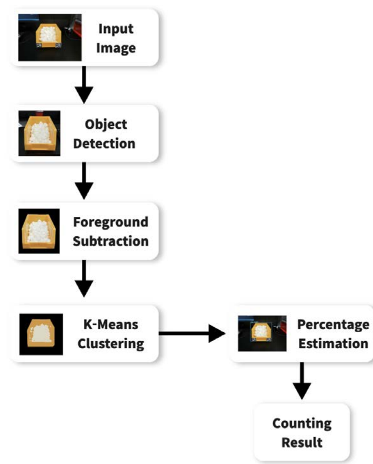

Fig. 1. Work flow of *MINSU*

I. Introduction

Counting items in the factory is indeed a labor-intensive task that can potentially result in inaccuracies, especially in a high-throughput industrial environment. Automating this process through the use of DNN can provide numerous benefits, both in terms of efficiency and reliability [2]. The primary advantage of implementing a DNN-based automated system is the substantial enhancement in operational efficiency. Given the processing capabilities of DNNs, counting tasks can be completed rapidly as compared to manual methods. Importantly, DNNs are immune to fatigue, facilitating continuous operations with sustained accuracy [4]. The secondary advantage is the reduction in potential inaccuracies that may be inherent in human-based counting. Manual counting processes are vulnerable to errors due to fatigue or distraction. In contrast, once a DNN is appropriately trained, it can provide consistent and reliable results, thereby reducing the likelihood of errors in item count [6]. The benefits of such an automation approach extend beyond simple time and accuracy improvements. The application of DNN-based automation can free up human resources to focus on tasks that require higher-order cognitive abilities. Thus, industrial processes can be optimized, with machines handling repetitive tasks and humans focusing on decision-making and problem-solving tasks.

The proposed *MINSU*(Mobile Inventory and Scanning Unit) adopts a computational vision analysis approach to determine the remaining quantity or cabinet fullness. This process unfolds through five stages: object detection, foreground subtraction, K-means clustering, percentage estimation, and counting. In the object detection phase, the specific cabinet location is analyzed in terms of coordinates using the input image. Subsequently, the foreground subtraction method is employed to concentrate on the cabinet image by excluding the background. This might require some manual intervention, such as selecting parts that were not captured by the grab cut algorithm [7]. K-means clustering is then applied to transform the multicolored image into a three-color monotone image, thereby facilitating more rapid and accurate analysis. The final steps are percentage estimation and counting. These methods ascertain the proportion of material inside the cabinet as a percentage, which is subsequently used to approximate the material count. Given a successful implementation, this project could significantly improve residual quantity management, thereby addressing the issues delineated in the introduction.

From an economic viewpoint, the utilization of manpower for resource monitoring may constitute an inefficient resource allocation, which could lead to a decline in welfare. Moreover, the substantial height of the inventory (2 to 5 meters) exposes personnel to potential fall hazards as they ascend ladders for monitoring. Such falls could result in industrial accidents, thus leading to significant financial losses for the company. Consequently, the solution developed herein emerges as an advantageous alternative for circumventing these scenarios.

II. Related Works

Some previous techniques for determining the number of items in a rack are discussed in this section. These methods range from the most traditional form, where human are involved, to some enhanced processes that rely on sensors. While these improved systems eased the process, they also lead to additional cost to the factory and required prior pre-processing for the system to work effectively.

A. Physical count of inventory

This is the most primitive and accurate way to get the count of items in a cabinet. The process is very time consuming and taxing for workers, and there is a possibility of errors [9]. However, counting the items in a cabinet once and updating the count when some of them are taken out is much easier and more effective. The work described in this paper is to avoid the manual counting process and instead use an automated process to do the same.

B. Weight Sensors

To ease the process of keeping track of the count of objects in the inventory, the weight sensors are considered to be very useful [11]. Here, the weight sensors are made aware of the weight of one item in the cabinet. With reference to this data, the number of items in the cabinet could be retrieved by dividing the weight of the cabinet filled with items by the weight of one item of the same type. However, there is a drawback in that the cabinet could be filled with items other than the original type. This would result in a faulty estimated count of items in a cabinet. Moreover, installing weight sensors to the existing racks are expensive and require periodic calibration for the maintenance.

C. Volume of the item and cabinet

Our approach requires one to find the volume of the cabinet with items and the volume of one of these items. The approximate count of items in a cabinet is equal to the total volume of a cabinet containing multiple items divided by the volume of one item.

III. Challenge and Target Scenario

In this section, the drawbacks of traditional methods for inventory management are discussed. Among many different techniques, counting with hands, electronic scales, and IR sensors are some of the most well known ways to count items in inventories.

A. Counting with hands

Companies use an ERP (Enterprise Resource Planning) system to integrate and manage their main business processes. The system receives several essential pieces of information from different parts of the business, one of which is inventory information. To update inventory information, workers depend on receiving and forwarding based management and manual inventory counting. In other words, workers record the information such as type of stock type, quantity, and date whenever stock enters or exits the warehouse. In order to ensure accurate inventory, workers also have to manually count the number of inventories regularly. For small warehouses where only a few types of inventories are concerned, manually counting the number of items can be effective. In most cases, however, managing inventories physically can be very tedious and inefficient. Especially for small objects, such as bolts and nuts, it is nearly impossible to accurately count. Also, as workers get easily tired, human errors are bound to occur frequently. This makes the information in the ERP system and the real inventory count different.

B. Electronic scale

Some factories adopt electronic scales to avoid the burden of manual inventory counting. The scales are installed under each inventory box to measure its weight. Since the weight information of a single inventory is stored in the database, the inventory count can be calculated from the total weight. The table below shows some examples of calculations .

TABLE I
EXAMPLE OF INVENTORY CALCULATION WITH ELECTRONIC SCALE

Scale	Product Type	Single product weight	Total weight	Inventory count
A	Bolt	3g	543g	543 / 3 = 181
B	Nut	5g	325g	2540 / 5 = 508
C	Pin	13g	1157g	1157 / 13 = 89

There are a few advantages to using electronic scales for inventory management. First, as we can see from the Table I, exact and precise amount of inventory can be calculated unless scales are dis-functional. The concerns regarding differences between the count information on the ERP system and real inventory can be avoided. Second, workers do not have to check the inventory regularly. Usually, in cases of physical counting, workers from other sections of the factory have to spare their time for the inventory counting. Electronic scales, however, save their energy and time which leads to better production efficiency. Third, since electronic scales can be connected to the server directly, they can save the information on the ERP system without any human intervention. Hence, it reduces the burden on workers.

On the other hand, electronic scales have some critical disadvantages. The price of the hardware and installation is quite high. The cost of one commercialized electronic scale is about 100 dollars, which leads to a total expense that will grow exponentially based on the size of the warehouse. Installing scales under each cabinet requires a reconfiguration of the factory layout which leads to factory down time and additional labor. In addition, different sizes of electronic scales are needed for different sizes of inventory cabinets. Since workers have to make contact with cabinets frequently, the possibility of failure of electronic scales are high.

IV. Volume Measurement Algorithm

A. Object Detection Method: YOLO

The YOLO (DNN framework) object detection model used in our *MINSU* makes it possible to find the specific position (x, y, width, and height) of a cabinet. These specific coordinates and sizes are considered an input to our system. In a factory

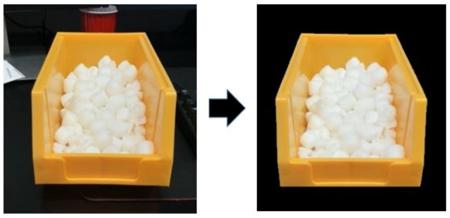

Fig. 2. Result of foreground subtraction on a cabinet image by the *GrabCut*

Fig. 3. Result image of K-Means Clustering

where huge amounts of inventory are stored, the speed of estimating a set of inventories is as important as reducing labor costs. The algorithm used in *MINSU*, helps to tremendously reduce the time of detecting the cabinets in the factories, and hence it was possible to easily trade off between speed and accuracy by tuning the size of the model.

B. Foreground Subtraction

After the coordinates and positions resulting from the execution of the object detection model, foreground subtraction is performed on them. The purpose of this step is to focus on the actual region of interest in an image containing various irrelevant background elements. This region of interest is retrieved using the *GrabCut* algorithm from *OpenCV* which helps to extract images that correspond to the foreground from its background, thus facilitating foreground subtraction. The process of extracting foreground from an image is carried out with numerous iterations of the *GrabCut* algorithm, which needs some user interaction for a better precision mask. This interaction occurs largely in two stages, the first of which specifies the area of the image that contains the foreground approximately as a rectangle where the specified rectangular area must contain all the objects that need to be focused. And secondly, if a user marks the background part of the foreground image and the missing foreground part with a mouse cursor, the foreground image is extracted more efficiently.

To implement this algorithm with the object detection model discussed above, parameters like the X-coordinate, Y-coordinate, width, and height, along with the mask attribute and number of iterations, should be provided as input to the *GrabCut* algorithm. The parameters in the algorithm assist in removing unnecessary objects from the background of a cabinet.

The *iterCount* parameter of the *GrabCut* algorithm, increases the probability of correctly subtracting the foreground by reiterating through the algorithm for the specified number of times. This parameter accepts an integer number as input to repeat the foreground subtraction with the same image and coordinates.

C. K-Means Clustering on foreground subtracted image

After implementing the foreground subtraction algorithm on an image, the K-Means Clustering algorithm is implemented on the resulting region of interest image to separate the cabinet from the items within it with two different colors. Figure 3 depicts two different colors: white and yellow assigned for each item and cabinet respectively as the actual color of the cabinet is yellow and the items in it are white . This algorithm helps to approximately distinguish the items in a cabinet from the actual cabinet based on the colors they represent. The prime limitation of the K-Means Clustering algorithm in this implementation is that it detects all the similar color pixels on the image, which might include the reflected image of the items in a cabinet. In Figure 3, the reflected image of white items due to the light are also considered as set of items, which leads to a partial error in the final result.

D. Percentage Estimation

The final image contains three different colors: black, yellow, and white. The color of the region of interest changes depending on the objects in the cabinet and the cabinet itself. With reference to Figure 3, our program will count the number of pixels for both yellow and white sections in the image. Because of the empty spaces in the cabinet, the number of yellow pixels (the cabinet color) is excessive in some scenarios. The frame of the cabinet and the front side of the cabinet cannot be covered by items contained in the cabinet. Considering this proportion, 20 percent of the unused yellow color pixels are subtracted for the calibration purpose.

The final image often contain shadow of items in the cabinet, which leads to an increased number of pixels considered as the objects. This error can be further adjusted by adding an adjustment ratio for the shadow as a parameter. For example, if the result of the Figure 7 gives 88 percent accuracy without post processing, for the post processing, *MINSU* fine tunes parameters to overcome shadow effect. This parameter is between 0 and 1, then multiplied to the number of white pixels(objects pixels) in next process.

Eventually, given the yellow and white pixels, the percentage of the white pixels are calculated. The adjusted yellow and white pixels are added up together to get a total pixel number, which serve as the denominator of the final equation, and the adjusted white pixel number serve as the numerator. Figure 4 represents the estimated volume of items other than the white balls in the cabinet.

V. Test-bed

A. Data set

To test the volume measurement algorithm, we used eight different cabinets and three different types of objects to

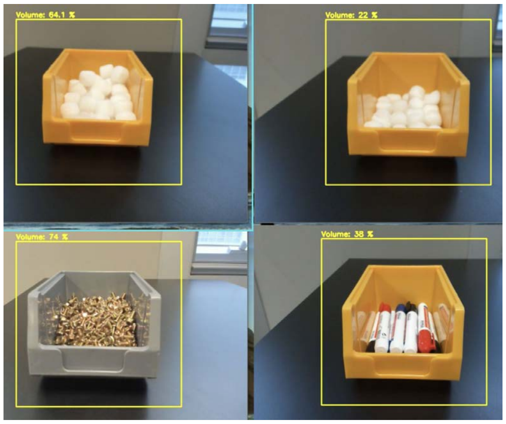

Fig. 4. Result in images of Percentage Estimation from different objects

(a) Cabinets

(b) Samples

Fig. 5. Cabines and Samples used for experiment

measure. Three color for cabinets were considered namely; red, yellow, and gray, with eight combinations of size in total.

The objects considered to occupy the cabinet, differ by size, color, and shape as shown in Figure 5. They are three types of fasteners used in any factory.

B. Test scenarios

To evaluate the volume measurement algorithm in a realistic environment, we have tested multiple scenarios. Diverse variables, such as the size of the cabinet, the color of the cabinet, various types of sample fasteners, and the shape of piled up items in the cabinet, were considered. Each of these variables broadened the evaluation aspects of the algorithm presented in this paper. Particularly, the effect of the shape of the pile of items in the cabinet was of great interest. This is because it affects the result of the volume measurement algorithm. In practice, different curves with the same number of objects should all have the same count. According to our hypothesis, however, it would be difficult to produce the desired results in some edge cases. For instance, in Figure 6, the images have the same amount of objects with different curves, but they seem to be different when observed by a human. This is because the peak of the curve could obstruct the view of the objects behind it [14]. As a result, three distinct edge cases were considered: a pile with peaks formed at the front, center and back.

a. Top, side and front view of the cabinet when peak is at the back.

b. Top, side and front view of the cabinet when peak is at the front.

c. Top, side and front view of the cabinet when peak is at the center/middle.

Fig. 6. Top, side, face view of the cabinet when items form peaks at back, front and center section

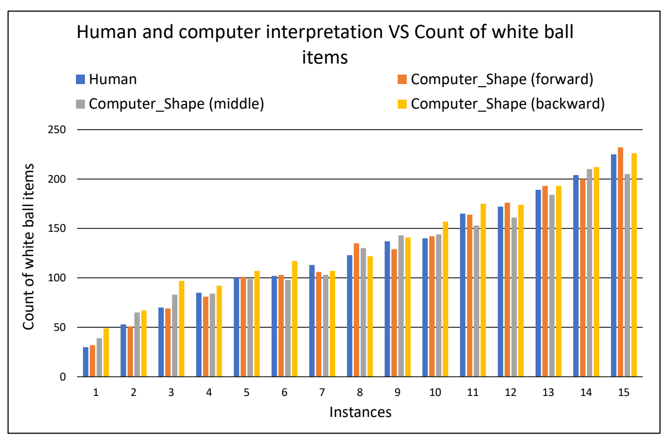

Fig. 7. Human/Computer count interpretation for items in cabinet with peak located at front, middle and back.

To overcome the problem with a peak, for some cases the image was captured from higher angle so that it minimizes the amount of unseen objects behind the peak.

For each test case, we put random amount of objects in a cabinet and executed the program to estimate its accuracy and precision.

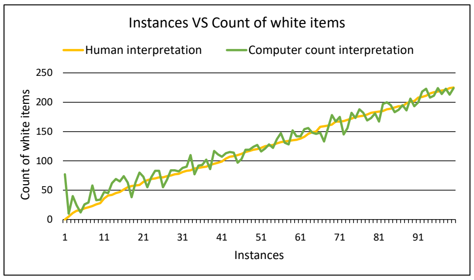

Fig. 8. Graph for actual and estimated count of white balls in a cabinet

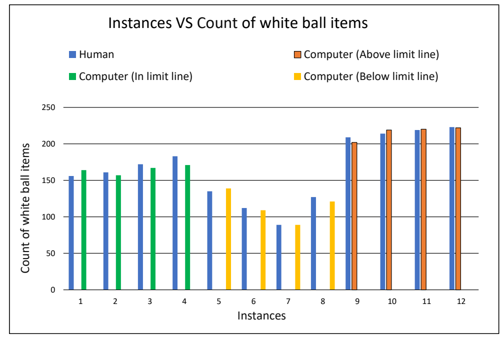

Fig. 9. Graph for human/computer interpreted count of items in, above or below the marked level of a cabinet

C. Experimental Setup

The experiment was performed in the demonstration lab at the University. The lab resembles an actual factory storage location. The position of the camera was fixed on the desk so that the tests would not be affected by varying camera locations. The experimental set-up used for testing the system consists of a Logitech Brio 4K Pro webcam, which has a resolution of 4096 x 2160 pixels, and a desktop with an Intel i7 processor and 2.0 GB of RAM. The procedure for the experiment consists of three steps. First, the camera takes a picture of the cabinet with its contents. Second, the image is processed by the volume measurement program. Third, the result of the program is compared with the ground truth(actual value) to find out the accuracy of the system.

D. Analysis

In this section the collected experimental data is analysed. To evaluate the systems accuracy, hundred instances of the actual count of items and the system estimated values were noted. The values have been plotted on to a graph Figure 8 as follows:

It is observed that the count, that is, the actual and estimated values, are almost similar.

At times, items in a cabinet are filled unevenly, which leads to peaks on the front, back, or middle portion of the cabinet. In such cases, the estimated count may differ from the actual count as the camera might not capture the presence of items behind the heap. Thus, to find the most relevant count of items in the cabinet with peaks in any of the positions, the program interpreted data was compared to the actual data. This data was plotted on a graph as shown in Figure 7 to represent varying interpretations of item count based on the positions of the peak of items in the cabinet.

Another study was carried out with respect to the marked level of a cabinet. The aim was to check the actual and estimated count of items in the cabinet to see if the set of items were visibly below, above, or at the marked level. The data on the graph Figure 9 suggests that the estimated count is almost similar to the actual count of items if the cabinet is filled above the marked level.

In previously observed cases, the items considered for the experiment were white balls. Thus, to overcome the skewness of the algorithm's performance on specific items, its performance was evaluated on cabinets filled with other fasteners like nuts and bolts. It was evident that the algorithm performed partially well, up to some extent. With the data collected after the study, specific graphs for nuts Figure 10 and bolts Figure 11 were plotted to display the performance.

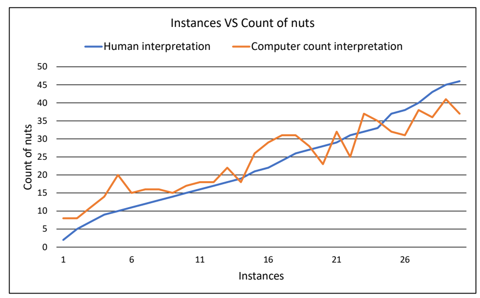

Fig. 10. Actual and estimated count of nuts

VI. Future Work and Discussion

This section discusses *MINSU* future works for various conditions in an inventory management system. The system introduced in this paper is improved and more effective than the traditional method. It could, however, be improved further to meet any additional requirements in a factory or to improve the existing functions for an inventory process.

A. Limitation of 2-dimensional images

The volume of residual quantity of a cabinet differ when it is observed from five different positions(top, front, left, right). The algorithm mentioned in this paper uses the frontal part image of a target cabinet. Even though the image was taken by a high resolution camera, the limitation of the dimension still exists. At this point, the computer can observe the front side of the given image, while 3-dimensional images can give multiple possibilities for estimating the volume of the given image. To reduce the error rate and improve counting algorithms, it is necessary to use various sides of images of a cabinet to get the best output out of them.

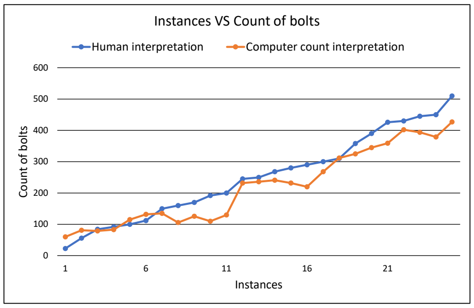

Fig. 11. Actual and estimated count of bolts

Using 3D scanning tools such as 3D Lidar and stereo cameras may bring better results when our algorithm is used, but using these tool needs careful rotation of single cabinet and require repeated captures. Therefore not fitted for robot based automated system. In better image reconstruction algorithm, *MINSU* could perform equal or better with simpler and economical configuration.

B. Detecting wrong items and errors

So far, this algorithm can approximately count the numbers of items in a cabinet and send the raw data to the main server, but it does not detect errors or incorrectly positioned items in a target cabinet. When this model can detect incorrectly positioned fasteners or cabinet errors, it will reduce manpower in the smart factory and increase time efficiency. In fact, it would be very challenging since detecting the wrong parts would require this model to learn new mechanisms for detecting the items in the cabinet. Though, this is one of the plans to improve the model.

C. Identify specific items in a cabinet containing mixed items

The program currently developed over the YOLO algorithm returns the count of items in a cabinet. This facilitates counting similar items in a cabinet. However, for the program to count all different items in a cabinet, it could adapt to future coming YOLO version, which probably has an ability to efficiently identify an element on the pixel level.

VII. Conclusion

We introduced a DNN-based method to estimate material volume in storage units using 2D image analysis. The method integrates object detection, grab-cutting, and subsequent image techniques to categorize object proportions into volume categories, and then estimates material counts. Implemented in factories, this can boost efficiency and safety by automating inventory tasks and minimizing hazards.

Acknowledgement

This work is supported by the Korean National Research Foundation grant NRF-2021R1F1A1052489, the MSIT, ICT CC Program (IITP-2019-2011-1-00783)/ICT start-up fund (RS-2022-00156555), the MOLIT start-up fund (RS-2022-00144171), and the Korea Radio Promotion Association (RAPA) XR-LAB grant.

References

[1] Li, Chenglong, Emmeric Tanghe, David Plets, Pieter Suanet, Jeroen Hoebeke, Eli De Poorter, and Wout Joseph. *"ReLoc: Hybrid RSSI-and phase-based relative UHF-RFID tag localization with COTS devices."* IEEE Transactions on Instrumentation and Measurement 69, no. 10 (2020): 8613-8627.

[2] Edvards, J. *Building a Smart Factory with AI and Robotics.* Robotics Business Review, Dec 21, 2022

[3] Nyalala I, Okinda C, Kunjie C, Korohou T, Nyalala L, Chao Q. *Weight and volume estimation of poultry and products based on computer vision systems: a review.*

[4] Sadiku, M. N., Tolulope J. Ashaolu, Abayomi Ajayi-Majebi, and Sarhan M. Musa. *"Emerging technologies in manufacturing."* Int. J. Sci. Adv 1 (2020): 105-108.

[5] Chen, Pengchang, and Vinayak Elangovan. *"Object sorting using faster r-cnn."*

[6] Gamal, Mohamed, Ahmed Donkol, Ahmed Shaban, Francesco Costantino, G. Di, and Riccardo Patriarca. *"Anomalies detection in smart manufacturing using machine learning and deep learning algorithms."* In Proceedings of the International Conference on Industrial Engineering and Operations Management, Rome, Italy, pp. 1611-1622, 2021.

[7] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. *"GrabCut": interactive foreground extraction using iterated graph cuts.* ACM Trans. Graph. 23, 3 (August 2004), 309–314.

[8] He, Kaiming, et al. *"Deep residual learning for image recognition."* Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[9] Chandren, Sitraselvi, Santhirasegaran Nadarajan, and Zaimah Abdullah. *"Inventory Physical Count Process: A Best Practice Discourse."* International Journal of Supply Chain Management 4 (2015): n. pag.

[10] O'Shea, Keiron, and Ryan Nash. *"An introduction to convolutional neural networks."* arXiv preprint arXiv:1511.08458 (2015).

[11] Singh, Kunal Kumar. *"Smart Inventory Management Using Electronic Sensor Based Computational Intelligence."* Recent Advances in Computational Intelligence (2019).

[12] Batra, Isha, Sahil Verma, and Mamoun Alazab. *"A lightweight IoT-based security framework for inventory automation using wireless sensor network."* International Journal of Communication Systems 33, no. 4 (2020): e4228.

[13] Pandian, A. Pasumpon. *"Artificial intelligence application in smart warehousing environment for automated logistics."* Journal of Artificial Intelligence 1, no. 02 (2019): 63-72.

[14] Nyalala, Innocent, Cedric Okinda, Qi Chao, Peter Mecha, Tchalla Korohou, Zuo Yi, Samuel Nyalala, Zhang Jiayu, Liu Chao, and Chen Kunjie. *"Weight and volume estimation of single and occluded tomatoes using machine vision."* International Journal of Food Properties 24, no. 1 (2021): 818-832.

[15] Samarasinghe, Sandhya. *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition.* Auerbach publications, 2016.

[16] Redmon, Joseph, and Ali Farhadi. *Yolov3: An incremental improvement.* arXiv preprint arXiv:1804.02767 (2018).

[17] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, Xiaojun Bi, and Samir R Das. 2020. *Modeling User-Centered Page Load Time for Smartphones.* In MobileHCI '20. ACM, 1–12.

[18] Duin Baek, Pratik Musale, and Jihoon Ryoo. 2019. *Walk to Show Your Identity: Gait-based Seamless User Authentication Framework Using Deep Neural Network.* In WearSys '19. ACM, 53–58.

[19] Shahroz Tariq, Hoyoung Kim, and Jihoon Ryoo. 2019. *AuthGPS: Lightweight GPS Authentication against GPS and LTE Spoofing (poster).* In MobiSys '19. ACM, 547–548.

[20] Conor Kelton and Jihoon Ryoo and Aruna Balasubramanian and Samir R. Das. 2017. *Improving User Perceived Page Load Times Using Gaze.* In NSDI 17. 545-559. USENIX Association

[21] Hoyoung Kim, Junghun Park, Seonghoon Park, and Jihoon Ryoo. 2022. *uGPS: design and field-tested seamless GNSS infrastructure in metro city.* In MobiCom 22. ACM, 636-647.