TEKNIKA: JURNAL SAINS DAN TEKNOLOGI

Journal homepage: http://jurnal.untirta.ac.id/index.php/ju-tek/

Real-time detection of fruit ripeness using the YOLOv4 algorithm

Widyawati Widyawati [a,1], ***Reni Febriani*** [a]

[a]*Program Studi Komputerisasi Akuntansi, Fakultas Ilmu Komputer, Universitas Banten Jaya, JL. Ciwaru II no. 73 Kota Serang Banten, 42117, Indonesia*
[1]*E-mail: widyawati@unbaja.ac.id*

ARTICLE INFO

Article history:
Submitted 31 May 2021
Reviewed 3 August 2021
Received 1 October 2021
Accepted 25October 2021
Available online on 1 November 2021

Keywords:
Convolutional neural network (CNN), fruit detection, YOLOv4.

Kata kunci:
Convolutional neural network (CNN), YOLOv4, deteksi buah.

ABSTRACT

Indonesia has an abundance and variety of fruit commodities. Referring to BPS, in 2019 the fruit production reached 22 million tons or increase 5% compared to 2018. However, with such a large production volume, most of the fruit inspection process in Indonesia still performed with human intervention. This process is very labor intensive, time consuming, and prone to inconsistencies and inaccuracies. The automation process using computer vision technology is expected to eliminate manual process therefore reducing costs, increasing efficiency and accuracy. In this study, YOLOv4 algorithm was applied to detect banana ripeness automatically. The training process is carried out using 369 banana images which are divided into two classes and the testing process is carried out on videos that are captured in real-time. Based on the research results, the best average accuracy rate is 87.6% and the video processing speed is 5 FPS (frames per second) using a single-GPU architecture.

ABSTRAK

Indonesia adalah salah satu negara yang memiliki banyak jenis komoditas berbagai macam buah. Merujuk data BPS, pada tahun 2019, produksi buah-buahan mencapai 22 juta ton, meningkat sebesar 5% dibandingkan dengan tahun produksi 2018. Namun dengan jumlah produksi yang begitu besar, mayoritas proses inspeksi buah (*sorting*) di Indonesia masih dilakukan secara manual oleh manusia. Proses manual tersebut membutuhkan sumberdaya waktu dan tenaga yang banyak, rentan terhadap inkonsistensi dan ketidakakuratan. Proses otomasi dengan bantuan *computer vision* diharapkan mampu mengeliminasi proses manual sehingga mengurangi ongkos, meningkatkan efisiensi serta akurasi. Pada penelitian ini, metode yang digunakan adalah algoritma YOLOv4. Algoritma tersebut diterapkan untuk mendeteksi kematangan buah pisang secara otomatis. Proses pelatihan algoritma dilakukan menggunakan 369 citra buah pisang yang dibagi menjadi dua kelas dan proses pengujiannya dilakukan terhadap *video* yang ditangkap secara *real-time*. Berdasarkan hasil penelitian didapatkan tingkat rata-rata akurasi terbaik sebesar 87.6% dan kecepatan pemrosesan video sebesar 5 FPS (*frame per seconds*) menggunakan arsitektur *single*-GPU.

Available online at http://dx.doi.org/10.36055/tjst.v17i2.12254

1. Introduction

Fruits are one of Indonesia's largest horticultural commodity products [1]. Referring to BPS data, in 2019, fruit production reached 22 million tons, an increase of 5% compared to the production year of 2018 [2]. However, with such a large production volume, most fruit inspection (sorting) processes in Indonesia are still carried out manually by experts. Like any manual process, this process is highly resource-intensive of time and effort, prone to inconsistencies and inaccuracies. Mistakes in predicting fruit ripeness will have an economic impact, with one-third of fruit production costs estimated to be left-to-rot or overripe when shipped from the garden [3]. The automation process using computer vision technology is expected to reduce costs, increase efficiency and accuracy.

Level of accuracy [8]. Algorithms such as Single-shot Multibox Detector (SSD) [4-6] and You Only Look Once (YOLO) [7] are algorithms commonly proposed by researchers to solve object detection (fruit) problems. The R-CNN and Fast R-CNN algorithms have advantages in the high level of accuracy but have weaknesses in processing time due to using a two-stage detection approach, namely region proposal and classification & localization [8]. In contrast to R-CNN and Fast R-CNN, the SSD and YOLO algorithms for the region proposal and classification and localization stages are carried out at the same

stage (called the single-stage detection approach) can significantly reduce processing time even though a reduction follows this in processing time. The process makes the algorithm less suitable for detecting in real-time.

In this study, the YOLO algorithm was chosen because it has the best processing speed, which is crucial when processing real-time images and videos [9]. Specifically, this research uses the 4th version (YOLOv4), the state-of-art of the YOLO algorithm. This research focuses on detecting banana ripeness because banana is one of the main production fruits in Indonesia (in general) and Banten (in particular). The results of this study are expected to provide breakthroughs in the plantation sector, especially in the development of automation technology.

2. Research Methodology

2.1. Convolutional Neural Network (CNN)

According to I Wayan Suartika E.P., convolutional neural network (CNN) is one of the multilayer perceptron (MLP) developments designed to process two-dimensional data. CNN is widely used in data processing, this is because CNN has a high network depth (one type of deep neural network) [10].

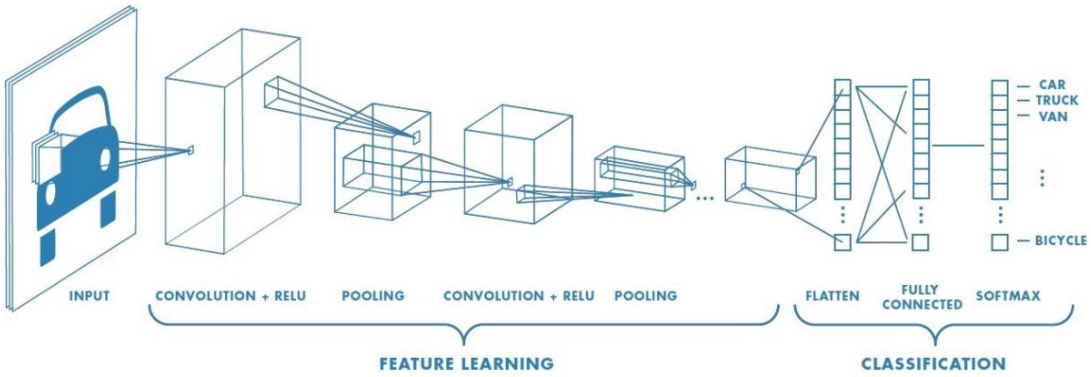


Figure 1. CNN layer.

The following is a layer in the CNN stages contained in Figure 1. An explanation of each of these stages is as follows:

1. *Feature Learning*
   At this stage, there are several layers, including the Convolutional Layer, Rectified Linear Unit (ReLU), Pooling layer. A layer serves to translate input into a feature based on the input characteristics in numbers in the vector. The Convolutional Layer stage performs convolution operations on the output of the previous layer where that layer is the main process in a CNN [11]. The Rectified Linear Unit (ReLU) stage eliminates negative values in the image, where the way ReLu works is to replace the negative value in the image or feature maps with a value of 0 [12]. As for the Pooling Layer, pooling simplifies output by using nonlinear downsampling and reducing the number of parameters used, and pooling is used to eliminate unimportant applications [13]. In this case, two types of pooling are most often used: average pooling and max pooling. In its implementation, max-pooling takes the largest pixels, then compiles them into a new matrix.
2. *Classification*
   This classification stage explains how to classify each neuron extracted in the previous stage, consisting of flattening, fully connected and softmax stages. Flattening is a step that reshapes features, such as reshaping feature maps into a vector that can be used as input from the fully connected (FC) layer. FC is a layer used to calculate the score of a class. Each neuron will be connected to all the numbers in the volume. The next stage is softmax which is the stage to calculate the probability of each possible target class and is used to determine the target class on the given input [10].

2.2. You Only Look Once (YOLO).

The YOLO algorithm is developed in [7] to solve object detection problems. This algorithm is a variant of CNN whose architecture adopts the pattern of neuron connectivity in the human brain. This algorithm is proposed to improve similar algorithms based on deep learning or neural networks such as AlexNet, SSD & R-FCN, which have processing time problems. In this algorithm, the classification model for detecting objects in the image is applied several times to a single image, several areas, and several sizes, thereby increasing processing time. While the approach to the YOLO algorithm only involves a one-time application of the neural network for all images (global input), it can significantly reduce processing time. It has been developed into YOLO version 4, namely YOLOv4 [14] and in other studies conducted [15]. The YOLO algorithm divides the input image into a grid consisting of cells S x S, where each cell in the grid will produce bounding boxes B and confidence values. The confidence value is described in Equation (1).

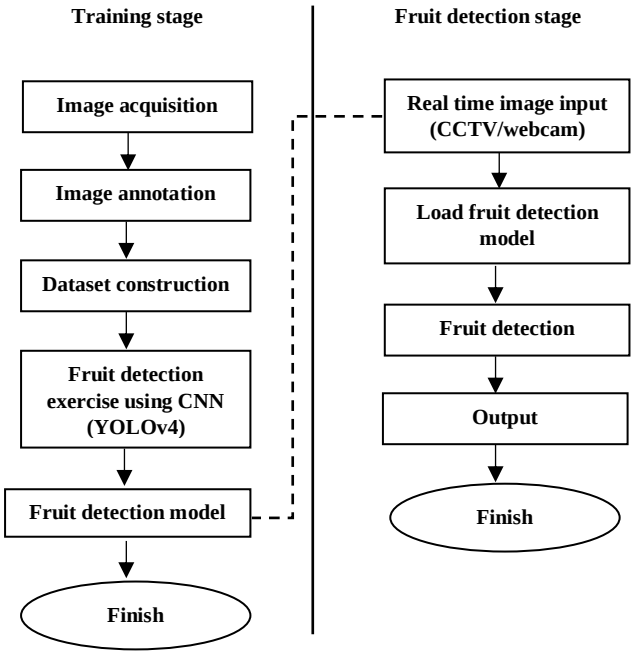
$$\mathrm{P}rob(Object) \times IUO_{predict}^{truth} \tag{1}$$

IOU (intersection over union) is the process of calculating the overlapping area of two unions. In the case of YOLO, it is the intersection between ground truth and predicts, while Prob (Object) refers to the probability of the presence or absence of objects in cells in the grid. The confidence value in the class is described in Equation (2).

$$\mathrm{P}rob(\mathrm{Class}) \times IUO_{predict}^{truth} = \mathrm{P}rob(\mathrm{Class}|\mathrm{Object}) \times \mathrm{P}rob(\mathrm{Object}) \times IUO_{predict}^{truth} \tag{2}$$

2.3. Research Method

The data processing method or technique used is the YOLOv4 (you only look once) algorithm. Figure 2 shows an overview of the method used.



Gambar 2. Metode yang diusulkan.

1. Image acquisition and annotation
   The image used is data are taken directly (primary data) obtained by observing the object of research directly. The researcher took the initial image was taken (acquisition) by the researcher using a smartphone. After the acquisition stage, the images in the dataset are manually annotated by the researcher based on their maturity level. In addition, as an additional requirement for the testing process, some images are also downloaded via google images. In this dataset, one image can consist of fruit with different maturity levels to simulate real-world situations.
2. Dataset construction
   To form the dataset used in this study, the images that have been collected and annotated are divided randomly into two parts, namely training images and testing images. The percentages for both parts are 90% of the images used as training images, and the remaining 10% are used as testing images.
3. The training and testing process
   In the training process, images categorized as training images are used as input to the YOLO algorithm to form a fruit ripeness detection model. The model is tested on a test dataset (testing) to see the level of accuracy and will also be tested against a data stream (webcam) to measure processing time in real-time (real-time¬). The meaning of the word real-time is the time interval that runs according to the current situation because the time calculation uses video directly, not through photos used as object detection. The device used is a webcam, so the algorithm is run in line with the time that is currently happening. If a photo is used, then the photo is an activity history captured and stored in the form of an image (.jpg or .png) which makes the photo an object of detection, not real-time.

3. Results and Discussion

3.1. Image Collection

The image data collection is done with the help of a smartphone during direct observation of the research object. The total image collected is 369, with the groupings presented in Table 1. The original resolution of the smartphone image is 4032 x 3024 pixels which are resized to 416 x 416 pixels. It aims to make the training process lighter and faster [9]. The resizing process is done using the Pillow library in Python 3.7.

Table 1. Image data collection.

No	Image group	Number of images
1	Ripe bananas	185
2	Unripe bananas	184

3.2. Image Annotation

The collected images are manually annotated one by one to create a dataset that can be used in the fruit detection model. Each image is labelled in the annotation process according to the class or classification required by the fruit detection model, which is the level of fruit maturity in this study. Yolo Mark software is used for this procedure. As illustrated in Figure 3, each item identified in the picture is assigned an anchor box and named based on its categorization during the annotation process.

Figure 3. Annotation process using Yolo Mark.

For each annotated image, Yolo Mark provides output in the form of a text (.txt) file containing the class and coordinates (x-y) of the anchor box used as input for the training process. The following is the result of image annotation using Yolo Mark shown in Figure 4 below:

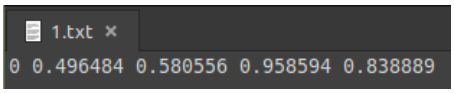


Figure 4. Image annotation results using Yolo Mark.

3.3. Dataset COnstruction

After all, images are annotated, the next step is to form a dataset. This process is done by allocating images that have been randomly collected into two groups, namely training (332 images ~ 90%) and testing (37 images ~ 10%). To avoid imbalanced datasets, each group contains images with almost the same proportion of ripe and unripe banana images [16].

3.4. Fruit Detection Model Training Using YOLOv4

The training stage for the ripe fruit detector was carried out using the Darknet.conv.137 framework [17] with Intel Core i3-9100F 4-core CPU (3.60 GHz), NVIDIA GeForce GTX 1050 Ti (4 GiB, 768 Cuda Cores) hardware and 16GiB RAM running on Ubuntu operating system. The training stages for 332 images are carried out up to the 6000th iteration, wherein in this iteration, the average loss value is 0.9584. The graph of the average loss from the beginning of the iteration to the 6000th can be seen in Figure 5. Each training result (weights) of the 1000th iteration is saved throughout the YOLOv4 training stage setup procedure to compare the performance of each weight. In this study, the performance of the training results in the 4000, 5000, and 6000 iterations was tested.

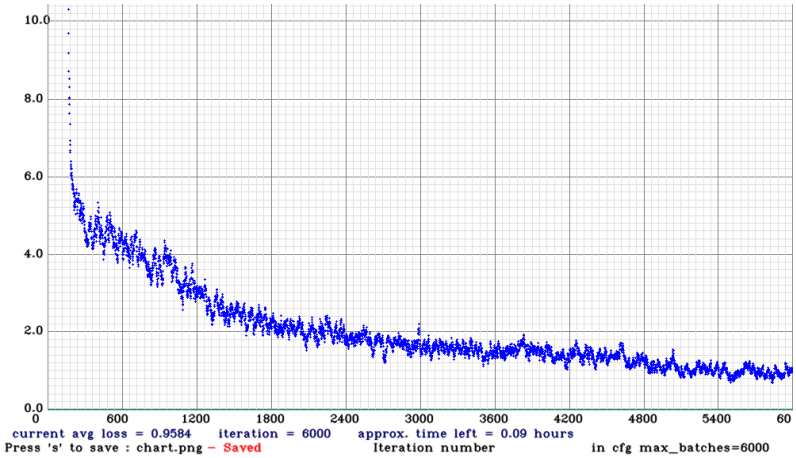


Figure 5. The YOLOv4 training process.

3.5. Testing the Fruit Detection Model Using YOLOv4

The first test is to measure the mean average precision (MAP) for each weight of the testing phase results (training). This value compares the ground-truth bounding box and the detected box [15], where the higher the MAP value indicates, the more accurate the detection model is. Based on the performance in Table 2, the training results (weights) of the 5000th iteration were chosen because they provide a better MAP value between the 4000th and 6000th iterations. The weights will be used at the next testing stage to evaluate the classification accuracy level of each class being tested. Evaluation of accuracy is carried out by running the command in Figure 6. Table 3 shows the results displayed in evaluating accuracy using YOLOv4.

Table 2. Evaluation of YOLOv4 MAP.

Model iteration	MAP	Average IoU
4000	82.6%	58.5%
5000	87.6%	64.7%
6000	87.5%	65.1%

Table 3. YOLOv4 accuracy.

Image class	Accuracy (%)
Buah pisang matang	93.33%
Buah pisang mentah	81.9%
Rata–rata	87.6%

```
./darknet detector map data/obj.data yolo-obj.cfg yolo-obj_5000.weights
```

Figure 6. The command source code generates an accuracy evaluation.

The results in Table 3 show that the proposed model can detect ripe bananas better than unripe bananas, where the accuracy of ripe bananas is 93.33% compared to 81.9% unripe bananas. The average accuracy value of 37 images used as test data is 87.6%. Figure 7 illustrates an example of the output from the testing process on a static image. As shown in Figure 7 above, the proposed system generates a bounding box for each object (banana) with a label according to the classification of its maturity level.

Figure 7. The output of the testing process on a static image.

3.6. Testing with Real-Time Data

The next testing process is carried out using a webcam device that takes live and continuous fruit image streams to measure whether the built detector can process the image close to real-time. As shown in Figure 8, this system can detect banana ripeness in real-time with an average accuracy of above 90% with frames per second (FPS) of 5 FPS. Figure 8 shows a pink box with the caption "ripe bananas, (98%)" contained in bananas is the output of banana ripeness detection by implementing the YOLOv4 algorithm using a webcam.

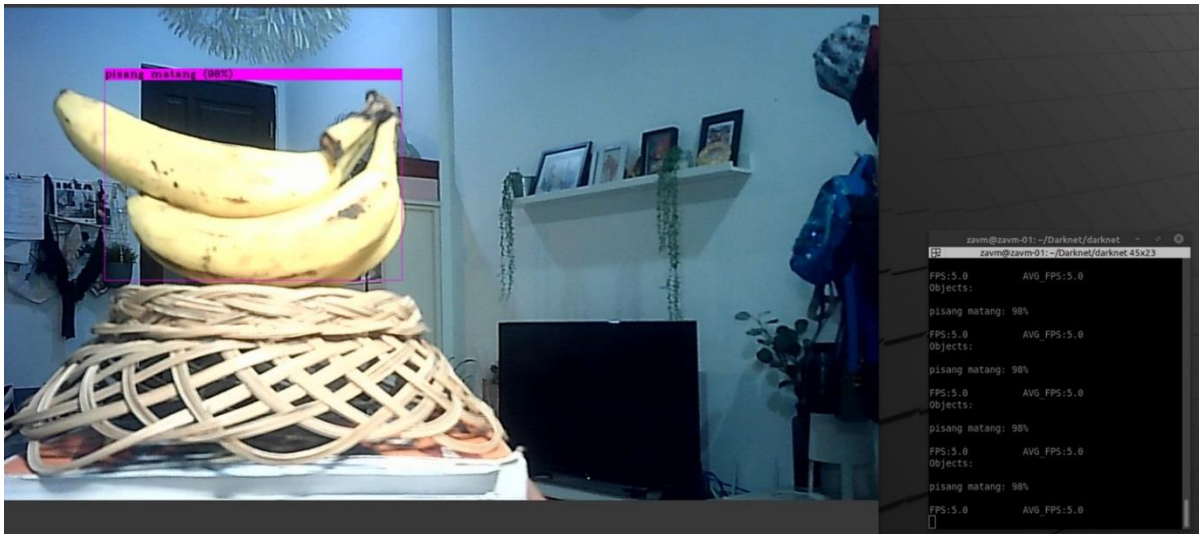


Figure 8. Output stages of banana ripeness detection in real time.

4. Conclusion

This study proposes a system to detect banana ripeness using YOLOv4. Based on the results obtained at the testing stage, the best average level of accuracy obtained is 87.6%. These results are obtained using weight in the 5000th iteration (from a maximum of 6000 iterations) in the training phase. For testing maturity detection with real-time (real-time) using a webcam, the device produces frames per second (FPS) of 5 FPS. As input for further research, a multi-GPU architecture needs to be implemented to increase the average frame per second (FPS) during real-time detection [18-19].

Acknowledgement

The author expresses gratitude for the financial support of this research for the grant provided by the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia, under the beginner lecturer scheme with master contract no. 065/SP2H/LT/DRPM/2021 with contract No. 033/SP2H/RDPKR-MONO/LL4/2021, 004.9/LP3M-UNBAJA/MOU/VII/2021.

REFERENCE

[1] Kaido, B., & Katsuhito, F. (2020). Supply chain and value-added distribution of pineapple fruit in Muaro Jambi Regency, Jambi province, Indonesia. *International Journal of Research in Economics and Social Sciences (IJRESS)*, vol. 10, no. 2, pp. 50-57.

[2] Badan Pusat Statistik. (2020). *Produksi Tanaman Buah-buahan 2020.* Accesssed at 1 April 2021. Available online on https://www.bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.html.

[3] Food and Agriculture Organization of the United Nations. (2011). *Global Food Losses and Food Waste: Extent, Causes, and Prevention*. Rome: Food and Agriculture Organization UN.

[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Amsterdam: Springer.

[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587.

[6] Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448.

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.

[8] Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, vol. 6, no. 100134, pp. 1-13.

[9] Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, vol. 8, no. 1, pp. 1-27.

[10] Bagas N, H. R. W., Mailoa, E., & Purnomo, H. D. (2020). Fruit detection for classification by type with YNOv3-based CNN algorithm. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 4, no. 3, pp. 476 - 481.

[11] Putra, I. W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi citra menggunakan convolutional neural network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, vol. 5, no. 1, pp. A65-A69.

[12] Pangestu, R. A., Rahmat, B., & Anggraeny, F. T. (2020). Implementasi algoritma CNN untuk klasifikasi citra lahan dan perhitungan luas. *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 1, no. 1, pp. 166-174.

[13] Miranda, N. D., Novamizanti, L. & Rizal, S. (2020). *Convolutional Neural Network* pada klasifikasi sidik jari menggunakan RESNET-50. *Jurnal Teknik Informatika (JUTIF)*, vol. 1, no. 2, pp. 61-68.

[14] Bochkovskiy, A., Wang, C.-Y. & Mark Liao, M. L. (2020). *YOLOv4: Optimal speed and accuracy of object detection*. Accessed at 1 April 2021. Available online on https://arxiv.org/pdf/2004.10934.pdf.

[15] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2021). Scaled-yolov4: Scaling cross stage partial network. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13029-13038.

[16] Japkowicz, N. (2000, June). The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, vol. 56, pp. 111-117.

[17] Redmond, J. (2016). Darknet: open source neural networks in C. Accessed at 1 April 2021. Available online on https://pjreddie.com/darknet/.

[18] Mansoub, S. K., Abri, R., & Yarıcı, A. H. (2019). Concurrent real-time object detection on multiple live streams using optimization CPU and GPU resources in YOLOv3. *SIGNAL:* The Fourth International Conference on Advances in Signal, Image and Video Processing, pp. 23-28.

[19] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329.