Smart Footwear Types Detection for Automatic Venue Entry Approval

William Eric Manongga
Department of Information Management
Chaoyang University of Technology
Taichung, Taiwan
s11014907@cyut.edu.tw

Rung-Ching Chen*
Department of Information Management
Chaoyang University of Technology
Taichung, Taiwan
crching@cyut.edu.tw

Janice Vania
Department of Information Management
Chaoyang University of Technology
Taichung, Taiwan
s11114629@cyut.edu.tw

***Abstract*— The Internet of Things (IoT) is one of the new and popular approach for incorporating the internet into our daily life. Using IoT, many smart systems are developed, such as smart homes, smart cities, smart traffic management, and many more. This brings us closer to seeing many implementation of technologies in our daily life. In our research, we employ object detection technology using YOLOv8 to automatically detect the footwear type. By detecting the footwear type, a system can be designed and developed to remind, reject, or give access to certain places based on the footwear they wear. This research poses a challenge because the type of footwear often looks the same and is difficult to differentiate. From our experiments we found that YOLOv8 is able to detect and differentiate the three types of footwear used in this research. YOLOv8m is the best model in this research with the precision of 77%, recall of 76%, and mAP@50 of 76%.**

Keywords—Smart systems, Footwear type detection, YOLOv8, IoT

I. Introduction

As technology improves, it helps humanity more in daily life. In this era, the applications and usage of the internet are expanding daily. The Internet of Things (IoT) is one of the new and popular approach for incorporating the internet into the users private, professional, and societal life[1].

With the rise of advanced technology and awareness towards societal conformance, many people have now understood the act of wearing appropriate attire for specific place and time, in the office people are expected to be professional which requires individual to wear formal shoes while in other places they can wear casual or any other types of shoes. Using object detection helps in identifying and locating things in videos or images especially when most of the shoes looks the same. Industry and educational institutes are insisting employees and students wear the appropriate shoes to improve safety concerns and for professional appearance. As is observed in some cases, when employees and students do not wear the appropriate shoes, it may lead to some serious injuries at the workplace or in the school labs [2]. In some religious places it is required to take of our footwear and go barefooted. When someone neglected this, or forget to take off their footwear, they could potentially offend the members of the specific religion.

In our research, we employ object detection technology using YOLOv8 to automatically detect the footwear type. By detecting the footwear type, a system can be designed and developed to remind, reject, or give access to certain places based on the footwear they wear. From our experiments, we see that all variants of YOLOv8 works well to identify different footwear types with YOLOv8m being the most stable one.

II. Materials and Method

A. Research design

Fig. 1 shows the how we conducted this research. The research started with the data collection. Images are collected from multiple sources in the internet and separated based on the category. Next step is the data preprocessing. Several things are done at this step, starting from removing unused images, rename the files to follow a certain standard, labeling of images, and splitting the dataset for training and testing. When the dataset is ready, it is used to train the object detection model. Lastly, we will test and evaluate the trained model using the testing data.

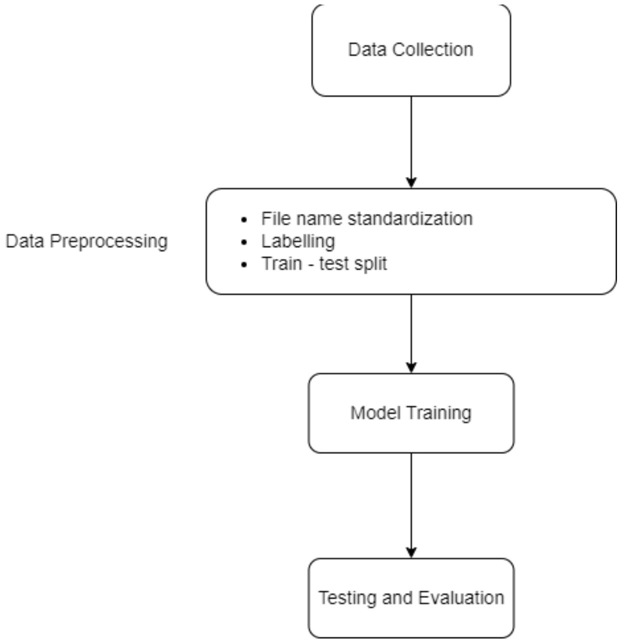


Fig. 1. Research flow

B. Dataset

For this research we created our own dataset by collecting images from multiple sources in the internet. These images are mostly images of people wearing shoes, or people barefooting. Only few images are shoes images without the person wearing it. Every image in our dataset has a minimal dimension of 640 pixels either on the height or the width to make sure that the image is not too small. There are total of 733 images in our dataset, split into the training, validation, and testing set. To label the image, we use the LabelImg [3] tool that provides image annotation in YOLO format. To split the data, we first separate the images into training and validation set with the ratio of 7:3. Then for the testing set, we split the validation set with the ration of 2:1. The total number of images, instances and the details of the split is shown in Table I.

TABLE I. DATASET DETAILS

Class	Images				Instances			
	Train	Val	Test	Total	Train	Val	Test	Total
C1	513	132	88	733	493	110	78	681
C2					381	85	76	542
C3					275	97	42	414

Three classes are defined in our dataset, formal shoes (C1), casual shoes (C2), and barefoot (C3). Formal shoes consist of leather shoes and high heel shoes. Casual shoes consist of sneakers and sport shoes. Barefoot consists of pictures of people's feet wearing nothing. Fig.2 shows some sample of the images in our dataset.

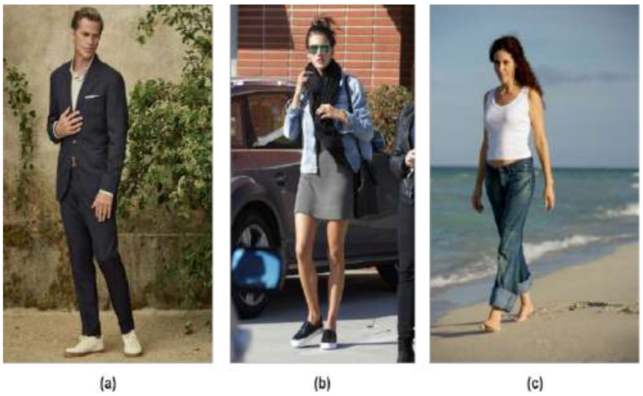


Fig. 2. Sample images from the dataset. (a) Formal shoes; (b) Casual shoes; (c) Barefoot

C. YOLOv8

YOLO (You Only Look Once) is a popular object detection algorithm and framework that enables computers to efficiently detect and classify objects like a human vision by recognizing each object in images or video frames and is often praised for its speed and accuracy [4]. YOLO was presented initially in 2016 by Joseph Redmon et al. and has since undergone several updates, with the most recent version being YOLOv8. Due to its speed and accuracy, YOLO has been found useful in various applications, including autonomous driving, surveillance systems, and object recognition in real-time scenarios [5].

YOLOv8, the latest state-of-the-art version in the YOLO series, offers advanced capabilities for object detection, image classification, and instance segmentation tasks. It comes in five different versions, nano (n), small (s), medium (m), large (l), and extra-large (x). Table II shows the comparison of the YOLOv8 variants on the object detection task when trained and evaluated using the COCO (Common Objects in Context) dataset with the image size of 640 pixels. Using different dataset and image sizes may result on different results.

TABLE II. COMPARISON OF YOLOV8 VARIANTS

Model	mAP@50-95	Parameters (M)	FLOPS (B)
YOLOv8n	37.3	3.2	8.7
YOLOv8s	44.9	11.2	28.6
YOLOv8m	50.2	25.9	78.9
YOLOv8l	52.9	43.7	165.2
YOLOv8x	53.9	68.2	257.8

Ultralytics developed and introduces YOLOv8 with substantial architectural amplifications and improvements, enhancing the overall developer involvement compared to its predecessor [6]. The YOLOv8 architecture follows a similar structure as its previous versions, which consist of backbone network and subsequent detection layers for bounding box prediction for feature extraction and object classification which is based on a deep convolutional neural network (CNN) [7].

YOLOv8 applies anchor-free approach with a decoupled head to independently predict the objectness, classification, and box regression task. Each branch will focus on its task and improves the model accuracy[8]. Early version of YOLO models which uses anchor boxes are posed with significant challenge, since the predefined anchor boxes might not align well on a custom dataset. Using the anchor–free approach, YOLOv8 now directly predicts the center of the object instead. This approach also reduces the number of box prediction and speeds up the Non-Maximum Suppression (NMS) process that sifts out redundant bounding boxes. For the loss function YOLOv8 still uses BCE loss for the classification loss while using both DFL[9] loss and CIoU[10] loss for the box regression loss. The three loss functions are given a specific weight ratio before added together to get the final loss value.

D. Evaluation metrics

Similar to other object detection methods, YOLOv8 is also evaluated based on three evaluation metrics, precision, recall, and mAP (mean average precision).

Precision is the proportion of positive predictions that are actually correct while taking account the false positives. The formula for precision is exhibited by

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall is the proportion of positive predictions that are actually correct while taking account of the false negatives. The formula for recall is exhibited by

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The mAP is calculated by finding the average precision (AP) for each class. It incorporates the trade-off between precision and recall that makes it suitable evaluation for most detection applications. The formula of mAP is exhibited by

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

III. EXPERIMENT AND RESULT

A. Experimental settings

The experiments in this research is run using a PC running Windows 10 operating system, using Python 3.9.15, PyTorch 1.13.1, CUDA version 11.7.1. The hardware is using Intel Core i7-8700 CPU, Nvidia RTX2080 Ti GPU with 11GB memory, and 32GB of RAM.

We use the same training parameters to train all the YOLOv8 variants to ensure the objectivity when comparing the results of our proposed methods. We start the training using the pre-trained weights provided by Ultralytics [11]. The training parameters settings can be seen in Table III.

TABLE III. YOLOV8 TRAINING PARAMETERS

Parameter	Value
Image size	320
Epochs	200
Learning rate	0.01
Batch size	8

B. *Results*

1) Training results

In the training phase, the model will be training using the data from the training set, while validated using the data from the validation set. Table IV shows the training results of all five basic YOLOv8 variants. For our assessment, we focus on the value of precision, recall, and mAP@50. The higher the value, the better the result. In terms of precision, YOLOv8x and YOLOv8s have the highest score with 78%, followed by YOLOv8l and YOLOv8n with 77%, and YOLOv8m has the lowest score with 76%. There is not much difference in the result in terms of precision for all five models. For the recall score, YOLOv8s have the best score with 75%, while YOLOv8n scores the worst with only 68%. For the mAP@50, there is not much difference in the result between each model. YOLOv8l have the highest score with 77%, and the lowest score is YOLOv8n with 73%. From the training result, we can see that there is a trend where larger model gives a more stable result. This can be seen from the example of YOLOv8n which get the lowest score in recall and mAP@50. In the training phase, YOLOv8l is the best model with stable performance.

TABLE IV. YOLOV8 TRAINING RESULTS

Class	v8n			v8s			v8m			v8l			v8x		
	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50
C1	0.73	0.85	0.82	0.84	0.87	0.85	0.77	0.85	0.86	0.79	0.87	0.86	0.76	0.85	0.84
C2	0.86	0.59	0.70	0.79	0.69	0.70	0.76	0.72	0.73	0.75	0.68	0.70	0.73	0.64	0.70
C3	0.71	0.62	0.67	0.71	0.68	0.71	0.75	0.60	0.70	0.77	0.68	0.75	0.84	0.65	0.72
all	0.77	0.68	0.73	0.78	0.75	0.75	0.76	0.72	0.76	0.77	0.74	0.77	0.78	0.71	0.75

2) Testing results

While the training result doesn't show that much difference, the real performance of the model will be assessed when using data that are not introduced during the training stage. A well trained model should be able to perform in almost the same performance as in the training phase. In this stage, we will use the data from the testing set to evaluate the performance of the trained YOLOv8 model in detecting footwear type. Table V shows the testing result of the trained YOLOv8 models.

From Table IV it is shown that YOLOv8 s and YOLOv8m have the highest precision with 77%. YOLOv8l and YOLOv8n have the lowest precision with 69%. For the recall, YOLov8m have the highest score with 76% and YOLOv8n have the lowest score with 68%. In terms of mAP@50 score, YOLOv8s have the highest score with 77% and the lowest score is by YOLOv8n with 69%. From the testing result, we can see that YOLOv8m has the best performance, while YOLOv8n is the worst.

TABLE V. YOLOV8 TESTING RESULTS

Class	v8n			v8s			v8m			v8l			v8x		
	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50	P	R	mAP @50
C1	0.66	0.68	0.63	0.80	0.74	0.78	0.79	0.74	0.74	0.66	0.65	0.66	0.74	0.69	0.67
C2	0.64	0.81	0.78	0.75	0.85	0.85	0.79	0.86	0.84	0.74	0.83	0.83	0.71	0.81	0.81
C3	0.78	0.55	0.66	0.75	0.63	0.68	0.73	0.67	0.69	0.68	0.66	0.64	0.71	0.73	0.69
all	0.69	0.68	0.69	0.77	0.74	0.77	0.77	0.76	0.76	0.69	0.72	0.71	0.72	0.74	0.73

C. *Discussions*

In this section we will discuss about the performance gap between the training and testing. Table VI compares the precision, recall and mAP@50 for each model during the training and testing, then calculate the difference between them. This information can be used to see which model have the better and more stable performance. For each evaluation metrics, the best model in terms of difference is shown in bold. YOLOv8m have a slight increase in terms of precision and recall in the training and testing, while YOLOv8s is the best one in terms of mAP@50 with a slight increase. YOLOv8m maintains the mAP@50 score with 76% on training and testing. From this information, we can conclude that YOLOv8m is the best model in detecting footwear type. It has a stable performance when evaluated with the testing set and also maintains a high score in precision, recall, and mAP@50.

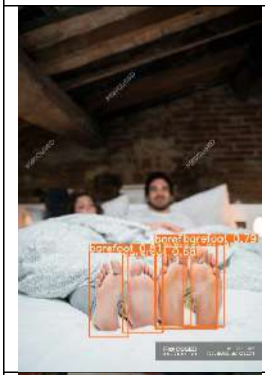
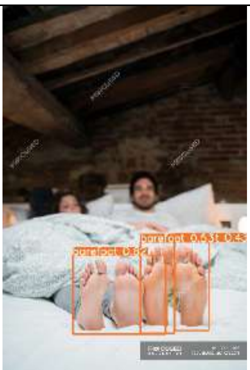
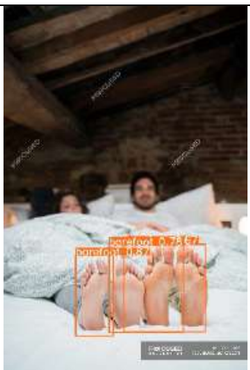
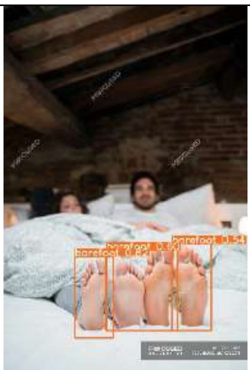
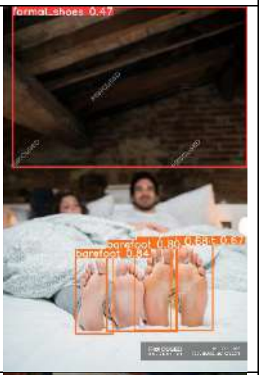
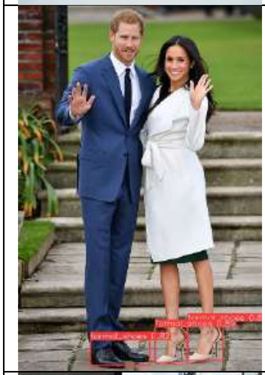
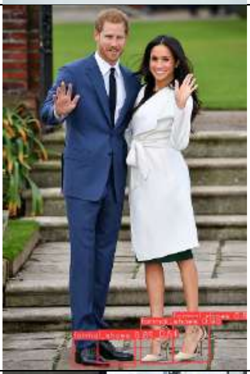
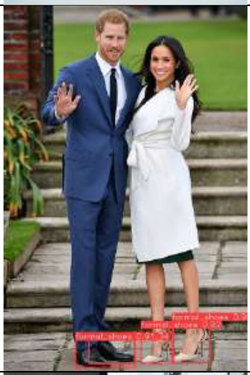
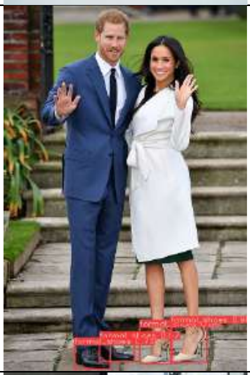
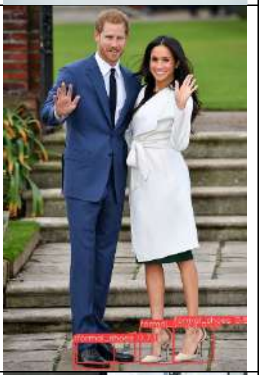
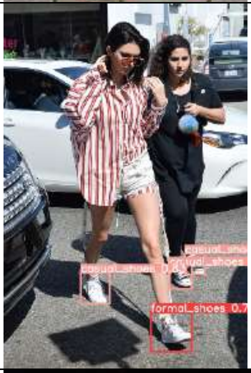
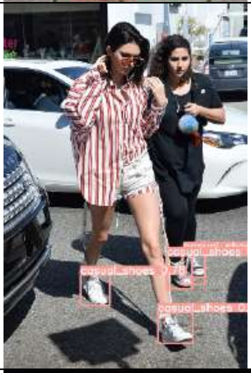
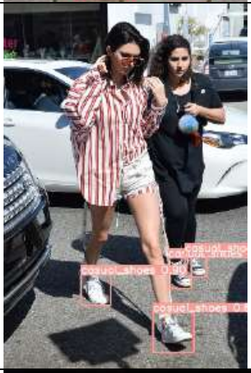
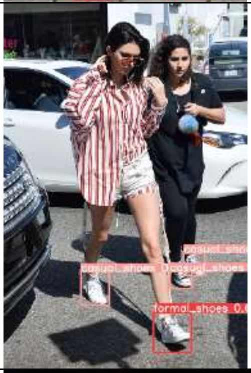
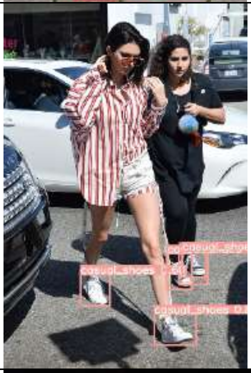
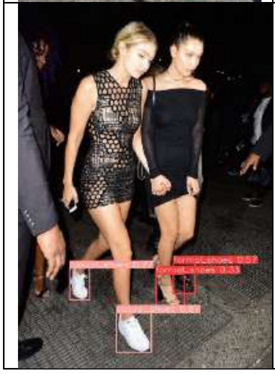
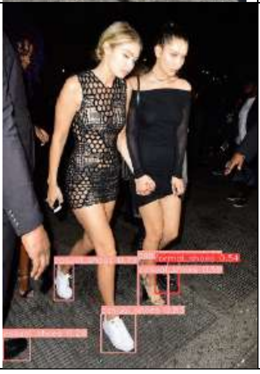
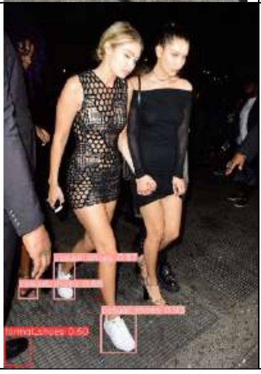
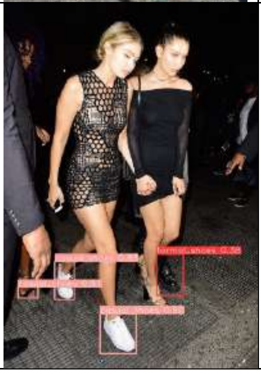
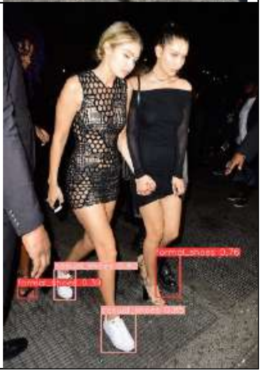
TABLE VI. COMPARISON BETWEEN TRAINING AND TESTING RESULT

		v8n	v8s	v8m	v8l	v8x
P	Train	0.77	0.78	**0.76**	0.77	0.78
	Test	0.69	0.77	**0.77**	0.69	0.72
	Diff	-0.08	-0.01	**+0.01**	-0.08	-0.06

R	Train	0.68	0.75	**0.72**	0.74	0.71
	Test	0.68	0.74	**0.76**	0.72	0.74
	Diff	0	-0.01	**+0.04**	-0.02	+0.03
mAP@50	Train	0.73	**0.75**	0.76	0.77	0.75
	Test	0.69	**0.77**	0.76	0.71	0.73
	Diff	-0.04	**+0.02**	0	-0.06	-0.02

Table VII shows some sample result from the footwear type prediction using all five models of YOLOv8. The sample images show the prediction of the three classes used in this research, formal shoes, casual shoes, and barefoot. In general, all the YOLOv8 models are able to detect the footwear type correctly. Occasionally, there will be some mistakes when differentiating between formal shoes and casual shoes. This is caused by the similar shapes between shoes types. This can be seen in the third row of Table VII where YOLOv8n and YOLOv8l did a mistake by detecting casual shoe as a formal shoe. Another mistake is seen on the first row, where YOLOv8x did a wrong detection of formal shoes when there is nothing.

TABLE VII. PREDICTION RESULT OF FOOTWEAR TYPE DETECTION

YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x
				
				
				
				

IV. Conclusion

In this research, we did a footwear type detection using YOLOv8. Three footwear types are used in this research, formal shoes, casual shoes, and barefoot. We used all five variants of YOLOv8 in this research, and all of them works well in detecting and differentiating the footwear type. From the experiments, we concluded that YOLOv8m is the model with the best and stable performance compared to the other variants. While some mistakes are found during the testing, it was still in the acceptable range. In the future, we will expand the dataset by adding more images and also adding more labels instead of only using the current three labels. This is because some use cases might need a more specific footwear type differentiation, not only, formal shoes, casual shoes, or barefoot.

The output from this research can be utilized as basis for doing automatic venue entry approval system by integrating it into an IoT system. Depending on the use case, this system can help to prevent injuries caused by improper footwear, blocking some guest that doesn't follow the dress code of a venue or vent, and many more. This could help to reduce the workforce needed to do manual checking done by human. By changing the dataset or adding more labels into the dataset, we can implement the system to do many different things automatically, which is the goal of IoT.

Acknowledgment

This paper is supported by the Ministry of Science and Technology, Taiwan. The Nos are NSTC-112-2221-E-324 -003 –MY3 and NSTC-112-2221-E-324 -011-MY2, Taiwan.

References

[1] K. R. Prasanna Kumar, D. Pravin, N. Rokith Dhayal, and S. Sathya, "Automatic Shoe Detection Using Image Processing," in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-3-030-96299-9_26.

[2] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "Internet of Nano-Things, things and everything: Future growth trends," *Future Internet*, vol. 10, no. 8. 2018. doi: 10.3390/fi10080068.

[3] Tzutalin, "LabelImg." 2015.

[4] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed. Tools Appl.*, vol. 81, no. 25, 2022, doi: 10.1007/s11042-021-11480-0.

[5] M. Kaushal, "Rapid -YOLO: A novel YOLO based architecture for shadow detection," *Optik (Stuttg).*, vol. 260, 2022, doi: 10.1016/j.ijleo.2022.169084.

[6] Ultralytics, "Ultralytics YOLOv8." https://docs.ultralytics.com/ (accessed Jul. 14, 2023).

[7] N. Zarei, P. Moallem, and M. Shams, "Fast-Yolo-Rec: Incorporating Yolo-Base Detection and Recurrent-Base Prediction Networks for Fast Vehicle Detection in Consecutive Images," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3221942.

[8] J. R. Terven and D. M. Cordova-Esparaza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," Apr. 2023, Accessed: Jul. 15, 2023. [Online]. Available: https://arxiv.org/abs/2304.00501v3

[9] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Advances in Neural Information Processing Systems*, 2020.

[10] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020. doi: 10.1609/aaai.v34i07.6999.

[11] Ultralytics, "YOLOv8." https://github.com/ultralytics/ultralytics (accessed Jul. 01, 2023).