Low-Barrier Object Detection for Mobile Applications

Khoi Nguyen The[1], Tuong Ho Vinh[2], Anh Hoang[3]

[1] Department of Information Technology, FPT University HCMC
[2] Department of Information Technology, TDT University
[3] School of Business Information Technology, College of Technology and Design, University of Economics Ho Chi Minh City

Correspondence: Anh Hoang. Email: anhh@ueh.edu.vn
Communication: received 12 Aug 2024, revised 1 Oct 2024, accepted 25 Oct 2024
DOI: 10.32913/mic-ict-research.v2025.n1.1343

***Abstract*: This paper investigates innovative applications of image processing and object recognition methods aimed at simplifying the creation and deployment of object detection models. By doing so, we seek to expand access to advanced computer vision technologies for small and medium-sized businesses. Our research leverages a combination of modern technologies including Flask for web development, Firebase for database management, and Kotlin and Jetpack Compose for mobile application development. We integrate these with automatic training methods provided by the Autodistill library, utilizing models such as Detic and YOLOv8.**

The results demonstrate that this technological combination significantly enhances the performance of our object detection models, contributing to AI solutions in the digital intelligence era. A notable advancement is Autodistill's capability to bypass the traditional dataset creation step by automatically generating labeled datasets from unlabeled input data. This feature markedly improves the efficiency and effectiveness of both data preparation and model training processes. Overall, our findings underscore the potential of these integrated technologies to democratize access to sophisticated computer vision capabilities for smaller enterprises, fostering greater innovation and competitiveness in the marketplace.

Keywords: *Distillation, object detection, android application*

I. Introduction

This paper presents a novel, user-friendly approach to mobile object detection, offering a low-barrier solution for identifying and locating objects in images and videos. Object detection is increasingly vital across various industries, playing a key role in tasks like autonomous vehicle navigation in complex environments and enhancing medical imaging systems for more accurate diagnoses. The proposed application aims to deliver efficient and precise object detection, meeting the growing demand for accessible solutions in both everyday and specialized use cases.

Traditional object detection methods often involve complex procedures that require specialized technical knowledge. Data preparation, feature engineering, and model training are time-consuming and can be challenging for non-experts. These hurdles have limited the accessibility of object detection to a select group of individuals with advanced technical skills.

In the realm of traffic safety, Computer Vision (CV) is crucial for the recognition of traffic signs and lane markings, significantly enhancing vehicular safety and enabling the development of Advanced Driver-Assistance Systems (ADAS) and autonomous vehicles [1, 2]. Additionally, CV technology is vital in defense and healthcare sectors. In defense, it facilitates the use of drones and robots to access and operate in hazardous areas that are unsafe or unreachable for humans [3]. In healthcare, CV applications include automated diagnosis, medical imaging analysis, and surgical assistance, thereby improving the accuracy and efficiency of medical procedures [4].

Despite the transforming potential of CV, several challenges impede its widespread adoption. Technical barriers, such as the complexity of developing and deploying CV models, remain significant hurdles. Additionally, the high costs associated with implementing and maintaining CV systems can be prohibitive for many businesses, particularly smaller ones. These factors contribute to the slow uptake of CV technology across different industries, despite its proven benefits.

To address these limitations, we propose a user-friendly approach to object detection that eliminates the complexities of data processing and training. Our app allows users to create custom object detection models with minimal input. Users simply need to provide the names of the objects they want to detect, along with a few sample images. The app then automatically handles the data preprocessing, model

training, and evaluation, providing users with a ready-to-use object detection model.

By democratizing object detection, our method enables a wider range of individuals to benefit from this powerful technology. This has the potential to drive innovation in various fields and address real-world challenges.

The following sections delve deeper into our research. Section 2 reviews existing contributions in the field, highlighting their strengths and weaknesses. Section 3 details our methodology, which is the core of our innovation. Section 4 presents the evaluation results, assessing our product on various aspects. If applicable, Section 5 explores experiments and ablation studies we conducted. Finally, Section 6 concludes the report by summarizing our findings.

II. Problem Statement and Related works

1. Problem Statement

Object detection, while a powerful tool, presents significant challenges for non-technical users. Traditional methods often involve complex procedures that require specialized knowledge and skills, limiting accessibility to a select group of individuals. Data preparation, including data collection, annotation, and preprocessing, is time-consuming and error-prone. Moreover, training deep learning models for object detection can be computationally intensive and requires expertise in machine learning.

Existing deep learning frameworks for object detection, while powerful, can be difficult to use for non-technical users. The steep learning curve, the requirement for extensive data preprocessing, and the complexity of model training can be daunting for individuals without specialized knowledge. Additionally, the performance of deep learning models heavily relies on the quality and quantity of training data, which can be a significant limitation for many applications.

To address these challenges, there is a clear need for a user-friendly approach to object detection that eliminates the complexities of data processing and training. Such a solution would enable a wider range of individuals to benefit from this powerful technology, democratizing AI and driving innovation in various fields.

2. Open vocabulary object detection

In the field of object detection, the shift towards Open Vocabulary object Detection (OVD) [5, 6] has presented a significant challenge and opportunity. Traditionally, object detection models have relied on predefined sets of objects, known as closed vocabulary systems, which limit the range of detectable objects to those included in the training data. These closed vocabulary models, while efficient, are constrained by their inability to recognize objects outside their predefined sets, which can be a significant limitation in real-world applications where the variety of objects can be vast and unpredictable.

In response to these limitations, the OVD models have emerged as a critical advancement. Among the pioneers in this field is Detic [7], which stands for Detecting Twenty-thousand Classes using Image-level Supervision. Detic is notable for its ability to identify a large number of objects, up to 21,000 different classes, far surpassing the capabilities of traditional models. This expansive ”vocabulary” allows Detic to recognize a much broader range of objects, including many that were previously difficult or impossible to detect with closed vocabulary systems.

The key innovation of Detic lies in its use of image-level supervision to expand its detection capabilities. By leveraging vast amounts of image data, Detic can learn to identify a wide array of objects without being limited to a predefined set. This method enables the model to generalize better and detect objects that it has not seen explicitly during training.

However, this strength in open vocabulary recognition comes with the trade-off of increased computational time and complexity. The processing power required to handle such a large number of potential object classes is significantly higher compared to closed vocabulary models. This increased computational demand can affect the speed and efficiency of the model, especially in real-time applications where quick detection is crucial.

Despite this trade-off, the capabilities of Detic represent a significant advancement in the field of object detection. Its ability to accurately identify a vast array of different object classes, including those that were previously difficult to detect, opens up new possibilities for applications in various domains. For instance, in dynamic environments such as autonomous driving, surveillance, and healthcare, the ability to recognize a wide variety of objects with high accuracy is invaluable.

Overall, the shift towards open vocabulary object detection, exemplified by Detic, marks a transformative step forward. By overcoming the limitations of closed vocabulary systems, OVD models like Detic are pushing the boundaries of what is possible in object detection, paving the way for more versatile and robust AI systems that can operate effectively in the diverse and unpredictable real world.

The Detic model consists of two stages: The first stage utilizes a specific CNN model, such as RPN (Region Proposal Network) [8], to generate a set of proposals. The second stage processes object features, providing classifi-

cation scores and refined locations for each object (using CLIP) [9].

3. Model distillation

The model distillation approach is a technique designed to leverage a large, bulky pre-trained model to label data, which is then used to train a smaller, more efficient model. This smaller model is optimized for real-time performance while being limited to recognizing a specific number of classes. The primary objective is to transfer the knowledge from the larger model to the smaller model, enabling it to achieve high accuracy without the extensive computational resources required by the larger model.

One of the pioneers in this area is Autodistill [10], developed by Roboflow. Autodistill exemplifies the model distillation approach by utilizing the capabilities of large, pre-trained models to generate high-quality labeled datasets. These datasets are then used to train smaller models that can perform object detection tasks with impressive speed and efficiency. This process significantly reduces the time and effort required to manually label data, while also ensuring that the smaller models retain a high level of accuracy and robustness.

Autodistill's approach addresses several critical challenges in the field of computer vision. Firstly, it mitigates the issue of computational resource constraints by enabling the deployment of lightweight models capable of real-time performance on devices with limited processing power. Secondly, it enhances the accessibility of advanced object detection capabilities, making them available to a wider range of applications, from mobile devices to embedded systems.

By streamlining the model training process and improving the efficiency of object detection tasks, Autodistill and similar model distillation techniques are driving significant advancements in the field of computer vision. These innovations are paving the way for more practical and scalable AI solutions, enabling broader adoption and integration of sophisticated computer vision technologies across various industries.

4. Real-time object detection

Various versions of the You Only Look Once (YOLO) [11–13] object detection model prioritize real-time processing speed with high accuracy, making them ideal for applications that demand fast processing. These single-stage models process the entire image at once, achieving high speed with minimal latency. However, this emphasis on speed comes with a trade-off: traditional YOLO models can only detect objects they have been trained on.

YOLOv8 [14], the current stable version of their YOLO, focuses on real-time object detection capabilities. Similar to other single-stage models, YOLOv8 excels in processing speed, achieving high frame rates for quick processing efficiency. While the recent release of YOLOv9 [15] marks further advancements in their YOLO lineage, YOLOv8 remains a robust and well-tested choice for applications that require real-time processing. Thanks to its balance of speed and accuracy, YOLOv8 emerges as an ideal choice for scenarios that demand immediate results, while YOLOv9 promises further improvements in the future. YOLOv8 is one of the most-endorsed real-time object detection models for the sake of application purposes.

5. Object Detection for everyone

Fit-Q [16] is one of the pioneers in bringing AI to Android app, this app implements the pose detection to locate key-joints of a person, then counts the number of reps that is performed, this app can detect various types of exercise including push-up, sit-up, burpee, etc.

Vmake [17] is a fashion app, this application takes in a normal image of a clothes item, then generates a high-quality advertizable image of that item tried on by a model, who can be customized with various types of body shape and facial expression.

Moreover, RepDetect[18] is also a fitness app that is not yet published onto Google Play, offers a wide number of exercise rep counting, similar to Fit-Q.

However, the approach of making the training stages accessible to everyday users with no technical expertise has not been adequately addressed by any publicly available work. While there have been significant advancements in user-friendly interfaces and simplified tools, the process of training models, especially in the context of machine learning and computer vision remains largely confined to those with specialized knowledge. Existing platforms and applications often assume a level of technical proficiency that excludes a significant portion of potential users. This gap highlights a critical area for development, as empowering non-experts to engage in the training process could democratize access to AI and significantly expand its practical applications. Despite the growing demand for more accessible AI solutions, there is a noticeable lack of public resources or tools that fully address this need.

III. Methodology

Here we will describe the core pipeline of the model creation process, from user's input to the result model. Then, in Section V, we will explain our application design as well as user flow.

1. Model pipeline

The overall concept of our model creation pipeline is inspired by Autodistill's distillation approach. This method uses a large pre-trained model to train a lightweight model. Our model pipeline, illustrated in Figure 1, follows this approach. We implemented knowledge distillation by using the annotated dataset from the Detic model to train the smaller YOLOv8 model. This process involved transferring the learned knowledge from the larger Detic model to the smaller one, following the distillation concept of extracting a portion of a large model's knowledge into a smaller model. Specifically, an open-vocabulary object detection model (the base model) is employed to label raw images of the desired objects, creating a dataset for training the target model. The setup of our pipeline is as follows:

a) Dataset preparation: images preparation

Since user-uploaded images may not provide sufficient coverage of all object angles and the diverse environments in which the objects may appear, we enhance the dataset by incorporating additional images. These supplementary images are sourced from public databases through APIs, broadening the dataset's scope. This approach ensures better representation of various object perspectives and environmental contexts, improving the model's performance and overall accuracy in object detection tasks.

b) Dataset preparation: image labeling

The image set is then labeled automatically by a base model named by Autodistill [19], which is pre-trained on gigantic image datasets that has the ability to detect any object in the vocabulary.

Furthermore, dataset augmentation is added to improve the dataset's coverage of environments. Augmentation methods that are used include: rotation, shear, brightness, saturation, exposure, etc. These augment operations are added at a fixed ratio of the dataset for everytime a new dataset is created.

c) Target model training

With the dataset prepared, we train a relatively small model on the dataset, with the expectation that it will be able to find user-defined objects at real-time speed.

We train a YOLOv8 model on the dataset with hyperparameters mostly kept as default, with some modifications to match our resources such as batch size, amp (turn off to prevent nan loss, as advised by YOLOv8 owner).

d) Model deployment

We will run the model on a cloud server, the customer's camera frames will be sent to the server for processing, and the annotated frames with bounding box indicating objects will be sent to customer's devices to show onto screen.

As a result, our pipeline is able to produce real-time object detection models with easy input of only object classes and sample images for each class.

IV. Results and Discussions

We evaluate the two most crucial aspects of our apps: the target model produced by the pipeline, and the intermediate dataset created from user input and the base model used to train the target model.

1. Evaluation of the target model

We run a full pipeline and evaluate the resulting model's detection on mAP50-95 score. We evaluate on two sets of object, *common objects* and *rare objects*.

a) Common Objects

The common objects were randomly selected from the classes in COCO dataset, all 80 classes of COCO dataset can be found in Table I.

Here we only randomly select 10 classes to evaluate, 10 classes are shown in Table I, including *keyboard*, *scissors*, etc.

Table I
Object classes used to evaluate our pipeline

Common Objects	Uncommon Objects
Keyboard	Snail
Scissors	Mango
Cow	Durian
Motorcycle	Dragon Fruit
Giraffe	Mangosteen
Laptop	Frangipani
Mouse	Lotus
Potted Plant	Mini Crispy Pancakes
Banana	Vietnamese Banana Cake
Carrot	Chung Cake

b) Rare Objects

In addition to common objects, which frequently appear in object recognition datasets, we selected a number of special object categories, such as certain dishes or flowers. The 10 selected objects are shown in Table I, including *snail*, *mango*, *durian*, etc..

The quantitative recognition results are presented in Table II. The Detic model, designed for labeling and creating datasets, exhibits acceptable processing time given its complexity. However, the overall performance remains relatively low compared to the practical expectations for an object recognition model.

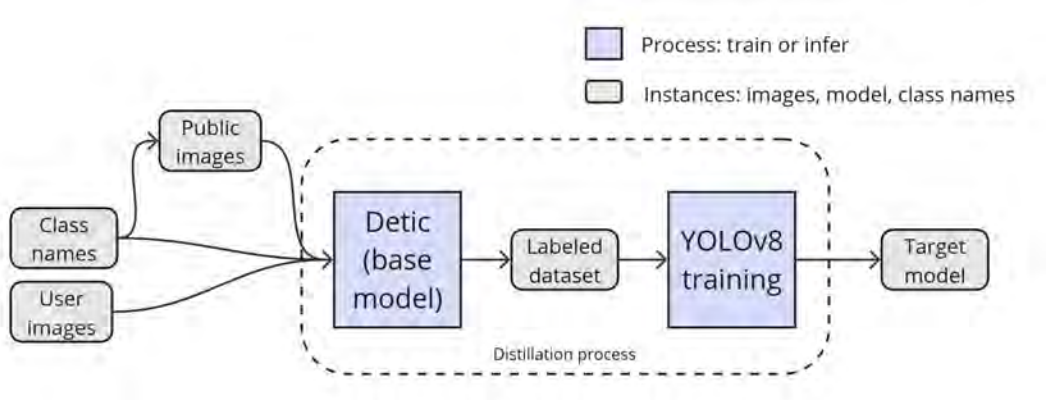


Figure 1. The model training pipeline

Table II
RESULT OF TARGET MODEL (YOLOV8) AFTER TRAINING

Results	Common objects	Rare objects
mAP50 – 95	0.305	0.055
Time (hh:mm)	1:28	1:32

2. Quality of the created dataset

Since the target model's mAP score is calculated on the base-model-labeled dataset, the quality of this dataset should also be evaluated.

To evaluate the base model's accuracy of label results, we make Detic [19] detect on labeled dataset to see how much Detic matches expected results. We also evaluate on common objects and rare objects.

a) Common Objects

We run Detic in inference mode on the valid set of COCO 2017 dataset [20], which contains about five thousand images, the results of Detic compared to labels from COCO is as below:

- mAP50-95: 0.55
- Inference time: 0.70s/image (total time 58:34)

Speaking of accuracy, Detic performs pretty well, comparable to state-of-the-art models in object detection. However, inference time of Detic is relatively high due to the size of the model, but as we use Detic as a base model, processing times does not pose a concern in our work.

b) Rare Objects

To measure rare object performance of the Detic model, we select a public dataset from Roboflow [21], which contains 100 labeled images of snails. The inference result of Detic is:

- mAP50-95: 0.144
- Inference time: 3.94s/image (total time 06:34)

Compared to common objects, rare objects cause Detic more difficulties, with the mAP score achieved at only 0.144. The inference time of 3.94s per image is signifi-canyly high, possibly due to the small number of images to infer, and Detic run from cold-start.

V. DEPLOYMENT

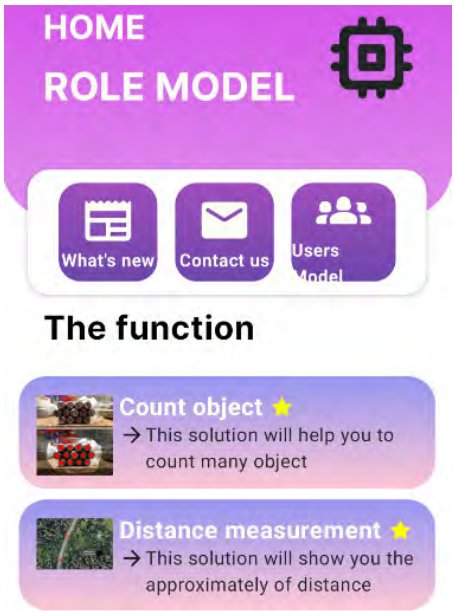
1. Front-end: user interface

Our app design targets making advanced technology accessible to everyday users, regardless of their technical background. The primary goal is to create an intuitive and user-friendly interface that allows individuals with no prior technical knowledge to easily navigate and utilize the app's features. By prioritizing simplicity and ease of use, we aim to democratize access to powerful tools and functionalities that were previously available only to tech-savvy users or professionals.

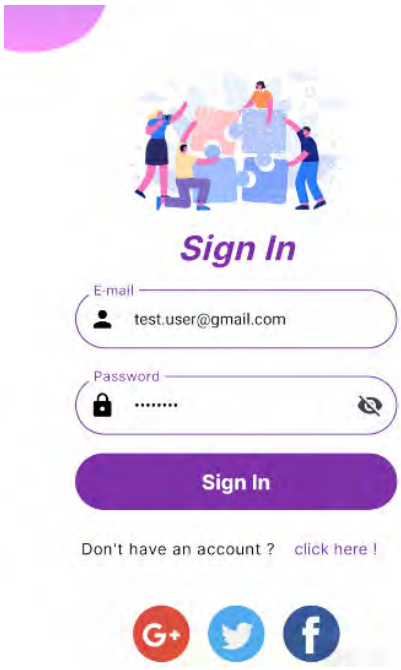
To achieve this, our design incorporates clear instructions, visual aids, and interactive tutorials that guide users through each step of the process. The interface is streamlined to minimize complexity and avoid technical jargon, ensuring that users can quickly grasp how to operate the app, as demonstrated in Figure 2, other snapshots of the application are reported in Figure 2.

The simple stages that a user will experience while creating a model include:

- Keyword input: User needs to provide the system with names of objects expected to be found in images/videos.
- Submit and wait for the model to be trained.
- Once model training is done, the user can load the model for detection on live camera.
- Nevertheless, our app allows users to create multiple models, and call one model to run at a time.



(a) Homepage



(b) Sign in

Figure 2. UI samples

2. Backend

Behind the scenes, the app is composed of several key components that work together seamlessly to deliver real-time object detection and user-friendly functionality. At the core of the system is a robust server infrastructure, equipped with powerful computational units capable of running complex detection algorithms in real time. This server processes the visual data, executes the detection tasks, and ensures that the results are delivered swiftly to the user.

User and model-related data, including user profiles, preferences, and trained models, are securely stored and managed within Firebase, a cloud-based platform that provides scalable and reliable data management services. Firebase serves as the backbone for data synchronization, enabling real-time updates and interactions between the app's various components.

The user interface (UI) application acts as the crucial intermediary that connects the server, Firebase, and the end user. This UI is designed to be intuitive and responsive, allowing users to interact with the app effortlessly. It facilitates communication between the user's device and the server, ensuring that detection requests are processed efficiently and that results are displayed promptly. Additionally, the UI provides access to various features and settings, enabling users to customize their experience and manage their data with ease.

The entire system is designed to be both scalable and efficient, capable of handling multiple simultaneous users while maintaining high performance. This architecture ensures that the app can deliver advanced object detection capabilities in a way that is both accessible and practical for everyday use.

The system's overall design, including the interaction between the server, Firebase, and the UI application, is illustrated in Figure 3. This diagram provides a visual representation of how the components are integrated, highlighting the flow of data and the processes that drive the app's functionality.

VI. Conclusion

This paper has presented a novel approach to democratizing object detection model creation. By introducing a streamlined pipeline for dataset creation and model training, wrapped in a user-friendly application, we have lowered the barrier of entry for individuals and SMEs to leverage the power of computer vision. Our proposed solution emphasizes simplicity and accessibility, enabling users to build effective object detection models with minimal technical expertise.

By making computer vision more accessible, we aim to contribute to the broader application of AI in everyday life and business operations. Our work has the potential to reduce the cost and complexity associated with implementing AI solutions, thereby fostering innovation and driving economic growth.

Looking ahead, we envision further enhancing our platform by expanding dataset capabilities and refining the base

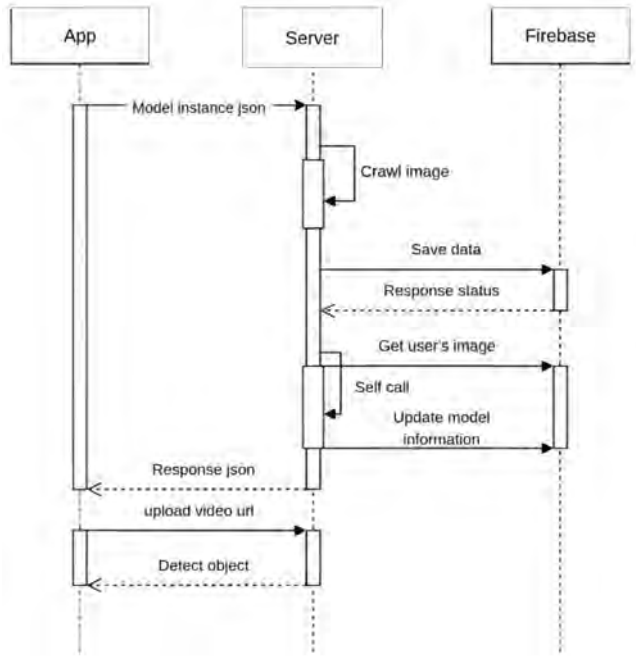


Figure 3. Backend components of the application

model to achieve even higher detection accuracy. These advancements will solidify our solution's position as a leading tool for rapid and cost-effective object detection model development.

REFERENCES

[1] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced driver-assistance systems: A path toward autonomous vehicles," *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 18–25, 2018.

[2] M. M. Antony and R. Whenish, "Advanced driver assistance systems (adas)," in *Automotive Embedded Systems: Key Technologies, Innovations, and Applications*. Springer, 2021, pp. 165–181.

[3] M. Payal, P. Dixit, T. Sairam, and N. Goyal, "Robotics, ai, and the iot in defense systems," *AI and IoT-Based Intelligent Automation in Robotics*, pp. 109–128, 2021.

[4] A. Khang, V. Abdullayev, E. Litvinova, S. Chumachenko, A. V. Alyar, and P. Anh, "Application of computer vision (cv) in the healthcare ecosystem," in *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem*. CRC Press, 2024, pp. 1–16.

[5] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, "Open-vocabulary object detection using captions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 393–14 402.

[6] H. Bangalath, M. Maaz, M. U. Khattak, S. H. Khan, and F. Shahbaz Khan, "Bridging the gap between object and image-level representations for open-vocabulary detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 781–33 794, 2022.

[7] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision*. Springer, 2022, pp. 350–368.

[8] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *CoRR*, vol. abs/2103.00020, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[10] Roboflow, "autodistill." [Online]. Available: \url{https://github.com/autodistill/autodistill}

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

[12] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[13] J. Du, "Understanding of object detection based on cnn family and yolo," in *Journal of Physics: Conference Series*, vol. 1004. IOP Publishing, 2018, p. 012029.

[14] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," jan 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, 2024.

[16] real BI GmbH, "Fit-q: Ai fitness + gaming." [Online]. Available: https://play.google.com/store/apps/details?id=com.statletics.bodyweightconnect

[17] Pixocial technology PTE. LTD., "Vmake ai fashion model studio." [Online]. Available: https://play.google.com/store/apps/details?id=com.airbrush.vmake

[18] G. Ngo, "Repdetect," https://github.com/giaongo/RepDetect, 2024-07-19.

[19] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," *CoRR*, vol. abs/2201.02605, 2022. [Online]. Available: https://arxiv.org/abs/2201.02605

[20] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[21] NBDuy, "Snail dataset," https://universe.roboflow.com/nbduy/snail-edpnm, dec 2021, visited on 2024-07-21. [Online]. Available: https://universe.roboflow.com/nbduy/snail-edpnm

Khoi Nguyen The , received the bachelor's degree in artificial intelligence from FPT University. His current research interests include computer vision, object detection, deep learning, and self-supervised learning approaches.
Email: nguyenthekhoig7@gmail.com

Tuong Ho Vinh is a final-year Software Engineering student at Ton Duc Thang University, specializing in computer vision with a focus on self-supervised learning and multimodal approaches for open-vocabulary detection tasks. His research direction is to explore the integration of textual and visual modalities to improve the adaptability and generalization of computer vision models.
Email: viktorho210@gmail.com

Anh HOANG received the B.S. degree in Telecommunication engineer from the Department of Electrical, Electronic, and Information Engineering, Hanoi University of Transport and Communication, in 2007, and the M.S. degree in Computer Science (major in Wireless Networks Security) from the National Taiwan University of Science and Technology (NTUST), Taiwan, in 2010. He completed Ph.D. program at the Graduate School of Advanced Science and Technology (major in Knowledge Science), Japan Advanced Institute of Science and Technology (JAIST), Japan, in September 2021. His research interests are related to AI/Machine Learning, Data Science/Data Mining/Data Analytics, CyberSecurity, and Business Intelligence/ Business Analytics. He is currently teaching at School of Business Information Technology, College of Technology and Design, University of Economics Ho Chi Minh City (UEH), Vietnam.
Email: anhh@ueh.edu.vn