# OBJECT DETECTION AND DISTANCE MEASUREMENT

**Baymatova M.X., Nuratdinova K., Raxmanov M.**
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan

***Abstract.*** *Object detection and distance measurement are fundamental tasks in computer vision, with applications ranging from autonomous vehicles to surveillance systems. This paper provides an overview of the various techniques and technologies used for object detection and distance measurement, including their principles, advantages, and limitations. We discuss the importance of combining these two capabilities to extract valuable information for real-world applications.*

*We used Yolo4 tiny for project.* ***YOLOv4-tiny*** *is the compressed version of* ***YOLOv4.*** *The YOLOv4-tiny model achieves* ***22.0% AP (42.0% AP50)*** *at a speed of 443 FPS on RTX 2080Ti, while by using TensorRT, batch size = 4 and FP16-precision the YOLOv4-tiny achieves 1774 FPS.*

*Moreover, in order to create project, we utilized range of methods such as autoencoder, CNN and DNN.*

***Keywords:*** *YOLOv4-tiny; One-stage methods; Two-stage methods; autoencoder, CNN, DNN.*

## I. INTRODUCTION

Object detection and distance measurement are crucial components of computer vision systems, enabling machines to perceive and interact with their environment in a manner similar to human vision. These capabilities have wide-ranging applications across various industries and domains, including autonomous vehicles, robotics, surveillance systems, augmented reality, and industrial automation. There are some additional insights into the importance of object detection and distance measurement in computer vision applications:

**Safety and Autonomous Systems:**

In autonomous vehicles, accurate object detection and distance measurement are essential for identifying obstacles, pedestrians, and other vehicles on the road. This information is critical for making real-time decisions to ensure the safety of passengers and other road users.

Figure 1. Safety and Autonomous Systems

**Robotics and Industrial Automation:**

- Object detection and distance measurement play a vital role in robotic applications such as pick-and-place operations, navigation in dynamic environments, and collaborative robots (cobots) working alongside humans. These capabilities enable robots to perceive and interact with their surroundings effectively.

**Surveillance and Security:**

- In surveillance systems, object detection and distance measurement are used to detect intruders, monitor crowd movements, and track objects of interest. Accurate distance measurement can help determine the proximity of objects to restricted areas or critical infrastructure.


Figure 2. Understanding object detection

As we said, Object Detection is a computer vision task in which the goal is to detect and locate objects of interest in an image or video. The task involves identifying the position and boundaries of objects in an image, and classifying the

objects into different categories. It forms a crucial part of vision recognition, alongside image classification and retrieval.

The state-of-the-art methods can be categorized into two main types: one-stage methods and two stage-methods:

**One-stage methods** prioritize inference speed, and example models include YOLO, SSD and RetinaNet.

**Two-stage methods** prioritize detection accuracy, and example models include Faster R-CNN, Mask R-CNN and Cascade R-CNN.

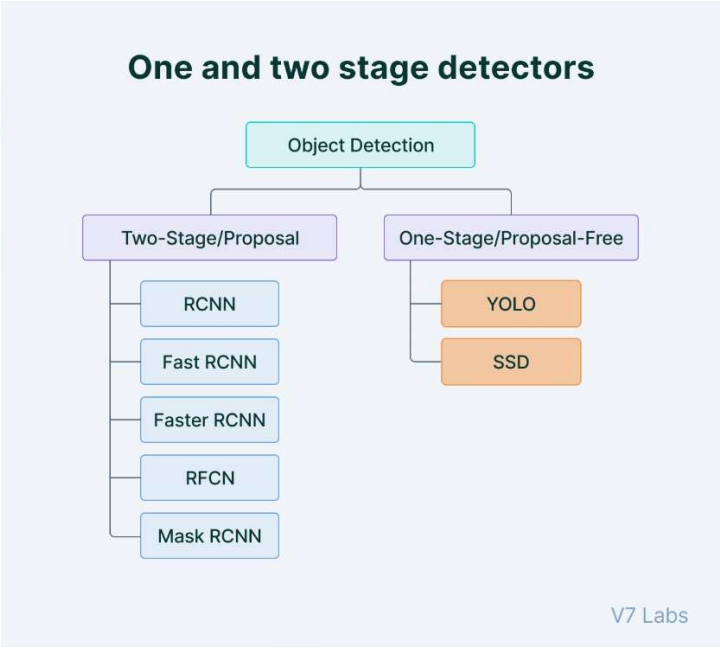



Figure 3. The state-of-the-art methods

## II. METHOD

### 2.1. Architecture

**YOLOv4-Tiny**, introduced by Wang, is a more simplified model of theYOLOv4 network, and the network size is 10% of YOLOv4. While it has a slightly lower detection accuracy, it is used in many studies for real-time object detection due to its fast network learning and detection speed [1-5]. As the performance of this deep learning technique has been verified in various fields, it is also used in agriculture to diagnose and predict diseases and pests, detect fruit and determine ripeness, and predict yield, showing good performance.

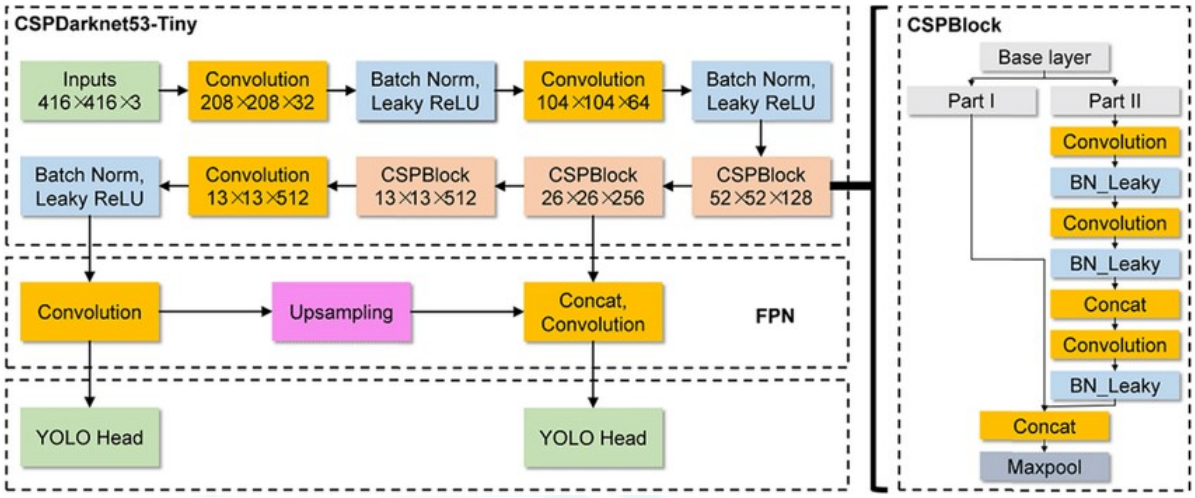



Figure 4. The architecture of YOLOv4-tiny Network

Figure 5 shows a schematic of the YOLOv4-Tiny detector detecting the degree of flowering of chrysanthemum flowers. When an image is an input to the YOLOv4-Tiny network, the location information of the chrysanthemum petal predicted through YOLOv4-Tiny is output, and the object overlap test and final class is output. The YOLOv4-Tiny network extracts object features through the backbone network and FPN (feature pyramid network). The predicted chrysanthemum flower information is output through the YOLO head. In this study, the input image size is (608×608) pxl, and the CSPdarnet53-Tiny model and FPN structure are applied as a network for feature extraction. The output structure of the YOLO head consists of the coordinates of the center point of the bounding box (A, B) representing the location and size of the cBBox, the circle radius (r), the confidence score, and the class probability. Non-maximum suppression (NMS) is applied to the same class to remove redundant detection among the values output by the YOLO head. If different classes are predicted, the class with the more significant confidence score is selected and determined as the final output.

We implemented YOLOv4-tiny in our project. YOLOv4-tiny is a condensed version of YOLOv4 designed to simplify the network structure and reduce parameters, making it suitable for use on mobile and embedded devices. It offers quicker training and detection compared to YOLOv4, with only two YOLO heads instead of three and training from 29 pre-trained convolutional layers, as opposed to YOLOv4's 137 pre-trained convolutional layers. YOLOv4-tiny achieves approximately eight times the frames per second (FPS) of YOLOv4, but its accuracy on the MS COCO dataset is around two-thirds that of YOLOv4.

On an RTX 2080Ti, the YOLOv4-tiny model achieves a speed of 443 FPS with an average precision (AP) of 22.0% (42.0% AP50). When utilizing TensorRT, a batch size of 4, and FP16-precision, the YOLOv4-tiny model achieves a remarkable 1774 FPS.

For real-time object detection, YOLOv4-tiny is preferred over YOLOv4 due to its faster inference time, which is more crucial than precision or accuracy in a real-time object detection environment.

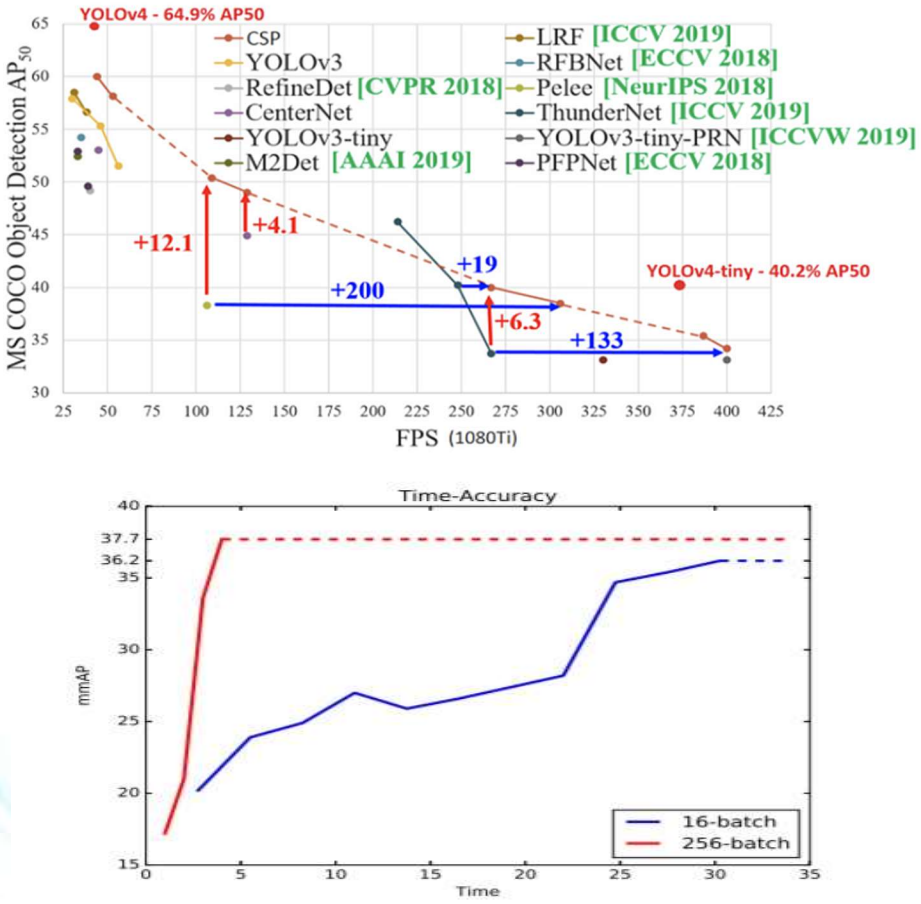
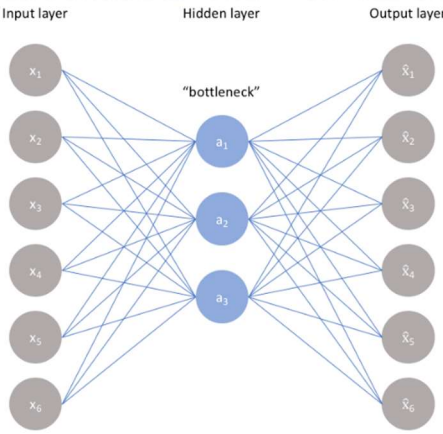



Figure 5. Validation accuracy of the same FPN object detector trained on COCO dataset, with mini-batch size 16 (on 8 GPUs) and mini-batch size 256 (on 128 GPUs). The large mini-batch detector is more accurate and its training is nearly an order-of-magnitude faster.

Table 1: Comparison of speed and accuracy of different object detectors on the MS COCO dataset (test-dev 2017). (Real-time detectors with FSP 30 or higher are highlighted here. We compare the results with batch=1 without using tensorRT).

| Method | Backbone | Size | FPS | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|
| YOLOv4: Optimal Speed and Accuracy of Object Detection | | | | | | | | | |
| **YOLOv4** | CSPDarknet-53 | 416 | 38 (M) | 41.2% | 62.8% | 44.3% | 20.4% | 44.4% | 56.0% |
| **YOLOv4** | CSPDarknet-53 | 512 | 31 (M) | **43.0%** | **64.9%** | **46.5%** | **24.3%** | **46.1%** | **55.2%** |
| **YOLOv4** | CSPDarknet-53 | 608 | 23 (M) | 43.5% | 65.7% | 47.3% | 26.7% | 46.7% | 53.3% |
| Learning Rich Features at High-Speed for Single-Shot Object Detection [84] | | | | | | | | | |
| LRF | VGG-16 | 300 | 76.9 (M) | 32.0% | 51.5% | 33.8% | 12.6% | 34.9% | 47.0% |
| LRF | ResNet-101 | 300 | 52.6 (M) | 34.3% | 54.1% | 36.6% | 13.2% | 38.2% | 50.7% |
| LRF | VGG-16 | 512 | 38.5 (M) | 36.2% | 56.6% | 38.7% | 19.0% | 39.9% | 48.8% |
| LRF | ResNet-101 | 512 | 31.3 (M) | 37.3% | 58.5% | 39.7% | 19.7% | 42.8% | 50.1% |
| Receptive Field Block Net for Accurate and Fast Object Detection [47] | | | | | | | | | |
| RFBNet | VGG-16 | 300 | 66.7 (M) | 30.3% | 49.3% | 31.8% | 11.8% | 31.9% | 45.9% |
| RFBNet | VGG-16 | 512 | 33.3 (M) | 33.8% | 54.2% | 35.9% | 16.2% | 37.1% | 47.4% |
| RFBNet-E | VGG-16 | 512 | 30.3 (M) | 34.4% | 55.7% | 36.4% | 17.6% | 37.0% | 47.6% |

**Autoencoder.** Autoencoders are an unsupervised learning technique in which we leverage neural networks for the task of representation learning. Specifically, we'll design a neural network architecture such that we impose a bottleneck in the network which forces a compressed knowledge representation of the original input. If the input features were each independent of one another, this compression and subsequent reconstruction would be a very difficult task. However, if some sort of structure exists in the data (ie. correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.




**CNN**s are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

CNN has high accuracy, and because of the same, it is useful in image recognition. Image recognition has a wide range of uses in various industries such as medical image analysis, phone, security, recommendation systems.

There are two main parts to a CNN architecture:

-A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction.

-The network of feature extraction consists of many pairs of convolutional or pooling layers.

-A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

-This CNN model of feature extraction aims to reduce the number of features present in a dataset. It creates new features which summarises the existing features contained in an original set of features. There are many CNN layers as shown in the CNN architecture diagram.

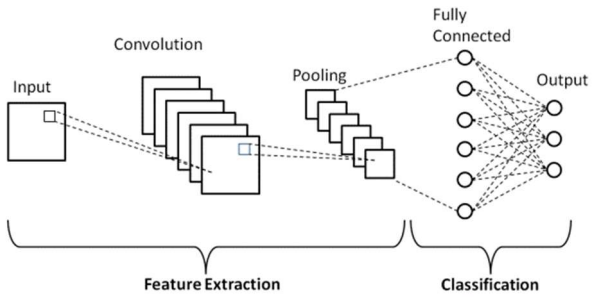



Figure 6. Basic Architecture CNN

**The architecture of social distancing**

In this section, we will discuss the essential steps that are required to build a sequence design to determine and check if the social distancing rules are respected or not among the individuals on the thermal videos as seen in Fig. 7:

1. Streaming the thermal videos, which contains the individuals.
2. Extracting the thermal video into frames.
3. Applying YOLOv4-Tiny architecture to detect only the individuals in thermal videos.
4. Verify the number of the individuals that are in the thermal videos.
5. Calculate the distance between the center point of the bounding boxes that contains the individuals in the thermal videos.
6. Lastly, the algorithm will make the decision for violation or safe conditions for the individuals based on the number of individuals in the thermal videos, and the measured distance between the centroid of bounding boxes. This is to note that we made two different levels for violation with two different threshold set points

for the measured distance between the center points of the bounding boxes. First violation level is called Alert, which is marked with a yellow color for the bounding

box, and the second violation level is defined as Risk, which is marked with a red color for the bounding box. We marked the safe condition with a green color for the bounding box.

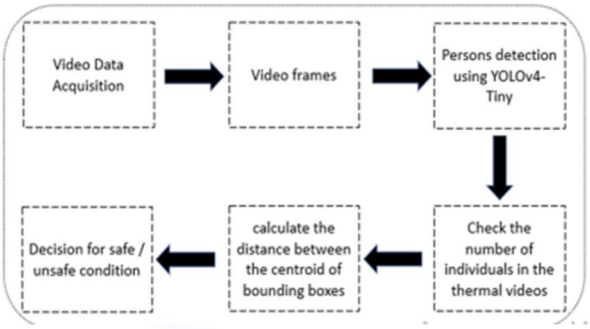



Figure 7. Sequence design for social distancing architecture

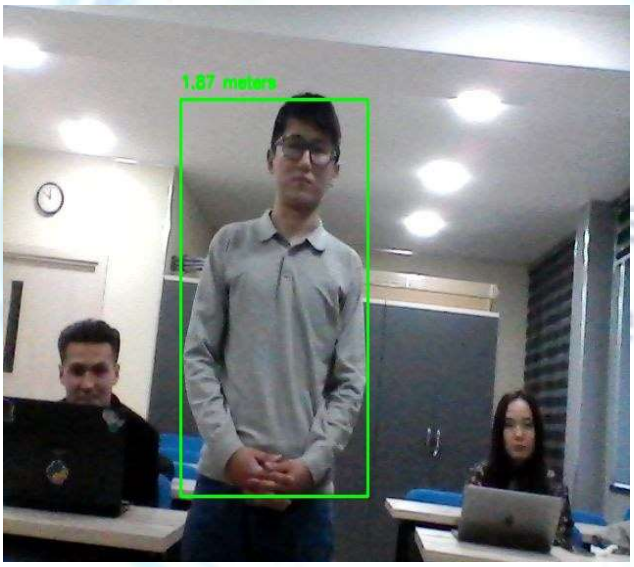
**III. Experiment results and discussion.**

In this section, all experiments details and comparison are described. We illustrated the experiments from different perspectives. To evaluate the performance of the proposed approach, we run the algorithm over the testing thermal images of both datasets. The thermal images were composed from realistic situation by different thermal cameras in the indoor environments. Thermal cameras can perform the measurement for radiated emitted energy from the skin of human in a safe and fast manner. With this in mind, we chose these datasets for our experiments. We also applied the YOLOv4-tiny and the technique proposed for social distancing measurement with basic function on large scale of thermal videos. These videos are scalable of screening individuals' movement while these thermal cameras were measuring their skin temperature. Further to our exploration, we carried out other experiment by examining (CNN) and you only look once (YOLOv4) detectors for people detection, using the same thermal images of the two training datasets of thermal images. The metrics that have been chosen to analyze the goodness for the algorithm are recall, accuracy, and precision; see Eq. (1):

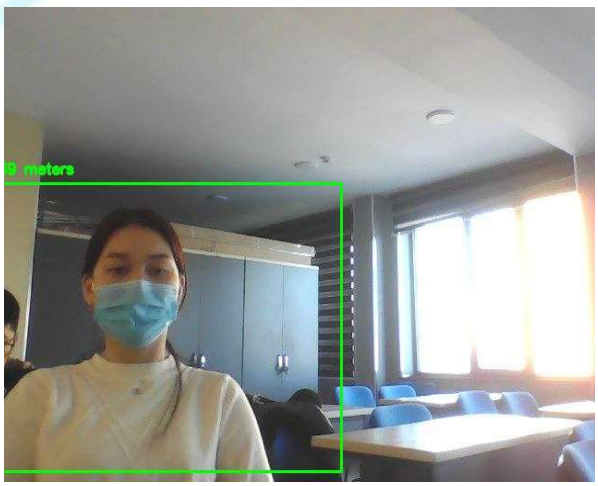
$$\text{Precision} = \frac{\text{TP}}{\text{TP+FT}}, \quad \text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FN+TN+FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP+F}}, \quad (1)$$

Table 2 Evaluation

| № | Real distance (m) | Model result (m) | Model efficiency(%) | Class name | Comp model and camera | True positive rate (TPR) |
|---|---|---|---|---|---|---|
| 1 | 2,93 | 2,62 | 69 | chair | usb 2.0 vga | 0,904320988 |
| 2 | 0,45 | 0,42 | 97 | person | usb 2.0 vga | 0,9375 |
| 3 | 0,6 | 0,62 | 98 | person | usb 2.0 vga | 0,967741935 |
| 4 | 0,63 | 0,75 | 88 | cell phone | usb 2.0 vga | 0,84 |
| 5 | 5,97 | 5,71 | 74 | clock | usb 2.0 vga | 0,958266453 |
| 6 | 0,53 | 0,56 | 97 | person | usb\VID_322E | 0,946428571 |
| 7 | 0,45 | 0,5 | 95 | person | usb\VID_322E | 0,9 |
| 8 | 0,59 | 0,68 | 91 | person | usb\VID_322E | 0,867647059 |
| 9 | 0,68 | 0,45 | 77 | cup | usb\VID_322E | 0,747252747 |
| 10 | 1,78 | 1,87 | 91 | person | usb 2.0 vga | 0,951871658 |





## IV. Conclusion

In summary, this article discusses the significance of object detection and distance measurement in computer vision and their wide-ranging applications. It provides an overview of the techniques and technologies involved, highlighting the need to combine these capabilities for practical use. The paper specifically utilized YOLOv4-tiny for the project, achieving high accuracy and speed, and also employed various methods including autoencoder, CNN, and DNN.

## References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5561–5569, 2017.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018.

[3] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9705–9714, 2019.

[4] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. HarDNet: A low memory traffic network. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017.

[7]. Zheng, Q., Zhao, P., Zhang, D., Wang, H.: MR-DCAE: Manifold regularization-based deep convolutional autoencoder for unauthorized broadcasting identification. Int. J. Intell. Syst. (2021).

[8] Yadav, S.: Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence. Int. J. Res. Appl. Sci. Eng. Technol. **8**, 1–10 (2020)

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr

Doll´ar, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000– 16009, 2022. 1, 2

[10] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2

[11] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 1

[12] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 1

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll´ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2

[14] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2

[15] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 1,