







V-DETR: Pure Transformer for End-to-End Object Detection

Dung Nguyen¹ , Van-Dung Hoang²  , and Van-Tuong-Lan Le³ 

¹ Hue University of Sciences, Hue University, Hue City 530000, Vietnam
nguyendung@hueuni.edu.vn

² Faculty of Information Technology, HCMC University of Technology and Education,
Ho Chi Minh City 720000, Vietnam
dunghv@hcmute.edu.vn

³ Department of Academic and Students' Affairs, Hue University, Hue City 530000, Vietnam
lvtlan@husc.edu.vn

Abstract. In the field of computer vision, the task of object detection is one of the important tasks, with many challenges and practical applications. Its task is to classify and determine the location of objects in images or videos. Many machine learning methods, especially deep learning, have been developed to perform this task. This article introduces a model combining DETR (DEtection TRansformer) and ViT (Vision Transformer) as a method to recognize objects in images/videos that only use components of the Transformer model. The DETR model achieves good results in object detection using the Transformer architecture and without the need for complex intermediate steps. The ViT model, a Transformer-based architecture, has brought about a breakthrough in image classification. Combining both architectures opens exciting prospects in computer vision. The input image automatically extracted features using the ViT model previously trained on the ImageNet21K dataset, then the features will be fed into the Transformer model to find the classification and bounding box of the objects. Experimental results on test data sets show that this combined model has better ability in object recognition than DETR and ViT alone. This brings important prospects for the application of the Transformer model not only in the field of natural language processing but also in the field of image classification and object detection. The results of our proposed model have quite high $mAP@0.5 = 0.444$ accuracy, slightly better than the original DETR model. The code is available at <https://github.com/nguyendung622/vitdetr>.

Keywords: Computer Vision · Object Detection · Classification · Convolutional Neural Networks · Deep Learning · DETR · ViT

1 Introduction

In recent years, the computer vision industry has achieved rapid development with the introduction of many models, especially deep learning models, which include models for the task of object detection. The task of object detection is a task in the field of computer vision, in which this task must perform two works: one is to determine the class of the object and the other is to position that object in the image or video. The

position of the object must be defined as a rectangular box, with the center of the shape and its width and height [1]. Figure 1 depicts the general flow of the object detection task. Object detection models are trained and evaluated based on different metrics.

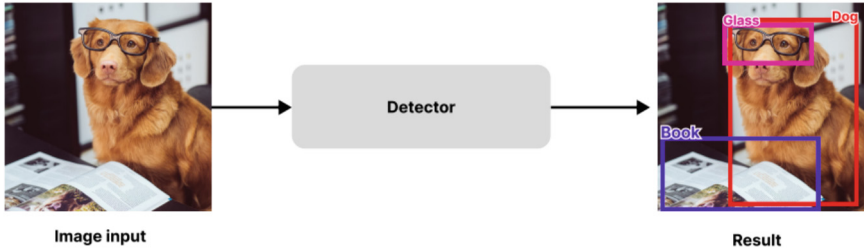


Fig. 1. General object detection task

There are many ways to classify object detection methods, but usually they are divided into 2 groups as follows: One is the one-stage method, the other is the two-stage method. Two-stage models are models in which in the first stage it tries to search and propose potential image regions containing the object, then in the 2nd stage the model conducts classification and finds the bounding box of the object. Proposed regional objects. Because these models consist of two steps, they often take more training and inference time, such as models: R-CNN model family [2–6], SPPNet [7], FPN [8]. The one-stage detection model classifies and locates objects in one pass using anchor boxes of various sizes and scales to define the object’s bounding box. It has a simpler design and has higher real-time performance than two-stage detection models, for example: Yolo model family [9–13], SSD [14], RetinaNet [15], EfficientDet [16, 17], DETR [18], Deformable DETR [19].

Although this field has made great progress, it still faces some challenges that the field faces in practical applications: (1) Many variations of objects in the same layers: This variation can be due to many different reasons such as obstructed vision, effects of lighting, posture, viewing angle, rotation, scaling or blurring,... (2) Number of object layers large objects: This results in resources spent on training and inference becoming higher than other tasks. (3) Efficiency: Object detection models require high computational resources to produce accurate detection results. Therefore, applying these models to mobile devices and devices with low computational resources will become difficult to balance between accuracy and computational performance.

2 Related Work

In this section, some popular object detection models are investigated with advantages and disadvantages of the methods. R-CNN (Region Convolutional Neural Network) is an object detection model based on a convolutional neural network developed by Ross Girshick and colleagues in 2014 [2]. First, this method uses a selective search algorithm [20] to find about 2000 potential image regions (also called RoI – Region of Interest). These image regions are then adjusted to the same fixed size and passed

through a previously trained CNN network (the author uses ResNet [21] network with the ImageNet1K dataset) to find a feature map. From this feature map, the model uses a feed-forward network (also known as FNN) to classify object and a regression network for bounding box prediction. Although the R-CNN model can perform object detection calculations and achieve efficiency, there are still many shortcomings, typically the model speed. Therefore, a number of improved models have been proposed to solve the problems that the R-CNN model is facing. Fast R-CNN is one of the improved versions of R-CNN developed by Ross Girshick and colleagues in 2015 [3]. This method overcomes the disadvantage of R-CNN: instead of using the CNN network 2000 times for RoIs, Fast R-CNN projects the 2000 found RoIs onto the same found feature map. Therefore, the performance of Fast R-CNN improves significantly. However, this method still does not overcome the major drawback of the previous R-CNN, which is the selective search algorithm to find about 2000 RoIs on the image. To overcome this drawback, the Faster R-CNN model was born, which is a further improved version of Fast R-CNN developed by Ross Girshick and his colleagues in 2015 [4]. With Faster-RCNN, instead of using the 2000 RoIs selective search algorithm, this model designs a subnetwork called RPN (Region Proposal Network) to extract RoIs. Then, the RoIs are projected directly onto the feature map, using RoIs pooling to obtain the feature map of the RoIs. Finally, this model applies a feedforward network to predict the object and a regression network to predict the bounding box of that object. A few years later the Mask R-CNN model was born, it is an object detection model developed by Ross Girshick and his colleagues in 2017 [5]. This model is an extension of Faster R-CNN model, used to predict the object's mask. Instead of using two heads to predict the object's label and bounding box, the model adds an additional head to predict the object's mask. Mask R-CNN has achieved great success in solving the problem of object recognition and localization, especially when it is necessary to determine the mask of the object to create a more detailed representation of the object region.

The models mentioned above have achieved high accuracy in object detection. However, due to going through two steps, including creating a region proposal and detecting objects in that region, the above models have high accuracy, but the speed is still slow, so they cannot be applied in real-time applications. Therefore, one-stage detection models were born, in which these models skip the region proposal step. A typical example is the Yolo family of models. YOLO (You Only Look Once) is one of the first object detection models that uses deep neural networks to perform object detection and object identification in images and videos. The first version of YOLO was called YOLOv1, introduced by Joseph Redmon and Santosh Divvala in 2016 in the paper [9]. YOLOv1 marked a significant advancement in object detection in computer vision. First, the YOLO model (also called YOLOv1) receives the input image and divides it into a grid of image cells of size $S \times S$. The model will check each image cell and if the object's center is in a certain image cell, that cell will be responsible for detecting that object. At each image cell, the model will predict many different bounding boxes. What we want, however, is to have only the bounding box representing one object. To retain only one bounding box among the proposed multiple bounding boxes, the YOLO model uses an important technique called NMS (non-maximum suppression). When predicting, the model will

often create multiple bounding boxes of various sizes and positions for an object in the image. To retain a unique bounding box for an object, YOLOv1 uses the NMS technique to eliminate bounding boxes with low confidence and low IoU. However, the disadvantage of YOLOv1 is that it cannot detect small, overlapping objects and can only predict at most $S \times S$ objects in the image. YOLOv2 [10], also known as YOLO9000, was introduced in a 2016 paper to improve shortcomings in the YOLOv1 model. And as a result, this model can detect more types of objects, predict faster and more accurately than YOLOv1. Compared to the previous model, this model uses another convolution called DarkNet-19 [22] as a backbone network for feature extraction. Improvements of the YOLOv2 model: one is to use the previously defined anchor boxes. The other is the use of image training strategies with many different scales. Using anchor boxes and training the model with many different image scales helps YOLOv2 be able to predict small objects well. The YOLOv3 model is introduced in the paper [11], published in 2018. The YOLOv3 model uses the Darknet-53 network, consisting of 53 convolutional layers, a variant of the ResNet network, as a backbone network to extract features of objects. Another improvement in the YOLOv3 model compared to YOLOv2 is the use of anchor boxes with different scales, so the model can detect different large and small objects. In addition, in the YOLOv3 model the author also used “feature pyramid network” (FPN) [8]. FPN is a CNN architecture, built as a pyramid of feature maps with many sizes and scales, so it can detect objects in images with many different sizes and shapes. The above improvements, the YOLOv3 model achieves higher accuracy and speed than the previous YOLO family models. The YOLOv7 model is introduced in paper [12], released in 2022. The model uses 9 anchor boxes, allowing it to detect objects with a wider range of shapes and sizes than its predecessors. An important improvement in YOLOv7 is the use loss function called “focal loss” instead of the usual cross-entropy function. Focal loss solves this problem by reducing the loss weight on well-classified samples and focusing on difficult samples—objects that are difficult to detect. The YOLOv7 model’s input has higher resolution than its predecessors, so it can detect smaller objects and has higher overall accuracy. In addition to the YOLO model line, the SSD model (Single Shot Detector) is also a model in this group. SSD is the first single-stage model with accuracy comparable to SOTA two-stage models such as Faster R-CNN. It was introduced by Wei Liu and colleagues in a scientific paper [14] in 2016. The SSD is built on the VGG-16 backbone network, with additional auxiliary structures to improve performance. SSD uses a technique called “multibox” to propose and predict bounding boxes containing objects. Instead of using anchor boxes across the entire image, SSD divides the image space into a grid of image cell, each image cell predicts several bounding boxes and a classification probability. Although the SSD model is faster and more accurate than the YOLOv1 model and the Faster R-CNN model, it still has difficulty detecting small-sized objects.

The above methods have the common feature of using convolutional networks trained on large data sets to extract image features. However, in some recent studies, the authors have used a famous structure in the field of NLP (Natural Language Processing) applying to the field of image processing, which is the Transformer network. One of those models must be followed by the DETR model and the ViT model. The Transformer model is a neural network architecture first introduced in 2017 by Vaswani et al. [23]. It is

mainly applied in the field of NLP. The Transformer model differs from traditional neural network architecture in that it uses a self-attention mechanism to calculate relationships between elements in a sequence. This mechanism allows the Transformer model to understand the relationships between words in a sentence, even when the words are far apart. The Transformer model includes two main components: Encoder and decoder. Both the encoder and decoder consist of a series of transformation layers, which use a self-attention mechanism to calculate the relationship between elements in the input sequence. DETR model: DETR stands for “DEtection TRansformer”, is a model in the field of computer vision introduced to solve the task of detecting objects in images [18]. DETR is a combination of the Transformer architecture and the task of object detection, and it stands out for its ability to perform object recognition without using complex preprocessing such as potential object region proposals. Functions as in many previous architectures. Important characteristics of DETR include:

- End-to-End Object Detection: DETR can perform object detection and classification directly from the input image without the need for intermediate steps such as detecting regions of interest.
- Using Transformer: DETR uses Transformer architecture to represent and process information in images. This allows it to understand the spatial and panoramic relationships between parts in the image.
- Multi-Head Attention: DETR uses multi-head attention mechanism to increase the ability to understand the diverse structure of objects in the image.
- Position Embedding: DETR uses position encoding of each segment in the image, helping it know the position of objects.
- Combining Positioning and Classification: DETR not only predicts the class (classification) for each object but also predicts their coordinates (location) in the image.

DETR is a powerful object detection model that can be used in many applications. It is still under development but has shown promising results in object detection tasks.

3 Pure Transformer-Based Detection Solution

With the idea of completely using the components of the transformer model for the object detection problem, in this study we propose to replace the ResNet backbone [21] in the original model of DETR with a model that uses that transformer is the ViT model.

3.1 Backbones Based on ViT Architecture

In object detection models, it is extremely important to use an object classification model that has been pre-trained on a large dataset. This model will act as an automatic feature extraction tool. Thanks to the features, the object detection model will use appropriate techniques to detect objects based on the extracted features. In this paper we use the ViT model as a model for automatic feature extraction. ViT is a model in the field of computer vision introduced to solve the task of image classification [13, 24]. The special thing is that ViT does not use traditional CNN networks such as ResNet or

VGG to extract features from images, but instead, it converts the image into an ordered sequence representation by dividing the image into regions small space called segment before entering the Transformer architecture as depicted in Fig. 2.

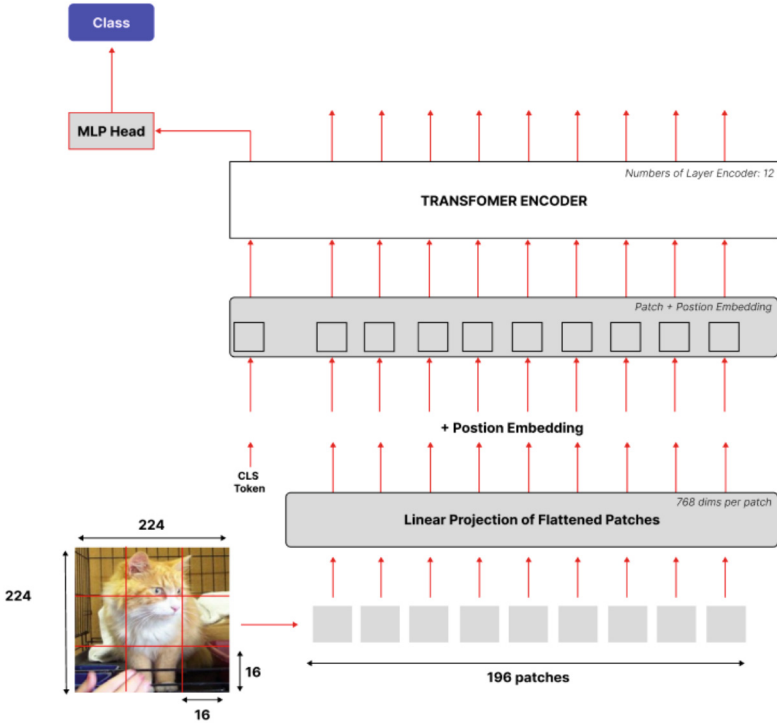


Fig. 2. General object detection based on ViT architecture.

Characteristics of the ViT model:

- The image is divided into patches and each patch is then converted into an Embedding Vector. This set of vectors will be used as input to the Transformer architecture.
- Embedding vectors before putting them into the Transformer network will add a special vector, called Position Encoding. This is the vector that encodes the positions of the Patches, to determine the order of the Patches before entering the model. This allows the model to understand the spatial relationships between segments and learn how to represent the image.
- Finally, the output of the ViT model is a string representing the image, and it can be used for many tasks, such as image classification. Through classification layers, ViT can predict the label of an image.

3.2 Improved Transformer for Object Detection

General DETR Pipeline: First with the input image, the DETR model uses the Backbone ResNet network (after removing fully connected layers) to extract features automatically. The obtained feature is combined with the position encoding vector before feeding it into the Transformer network because the transformer model does not keep information about the relative positions between objects in the image. The Encoder Transformer network uses multi-head attention to transform the input into a series of feature vectors, also called Feature Map. Object queries and feature maps are then passed into a Transformer decoder to predict the location and class of the objects. DETR uses a special loss function called “Hungarian matching loss” to compare the model’s predictions with the ground truth. This function helps determine the most suitable pair of prediction and ground truth objects to calculate the loss.

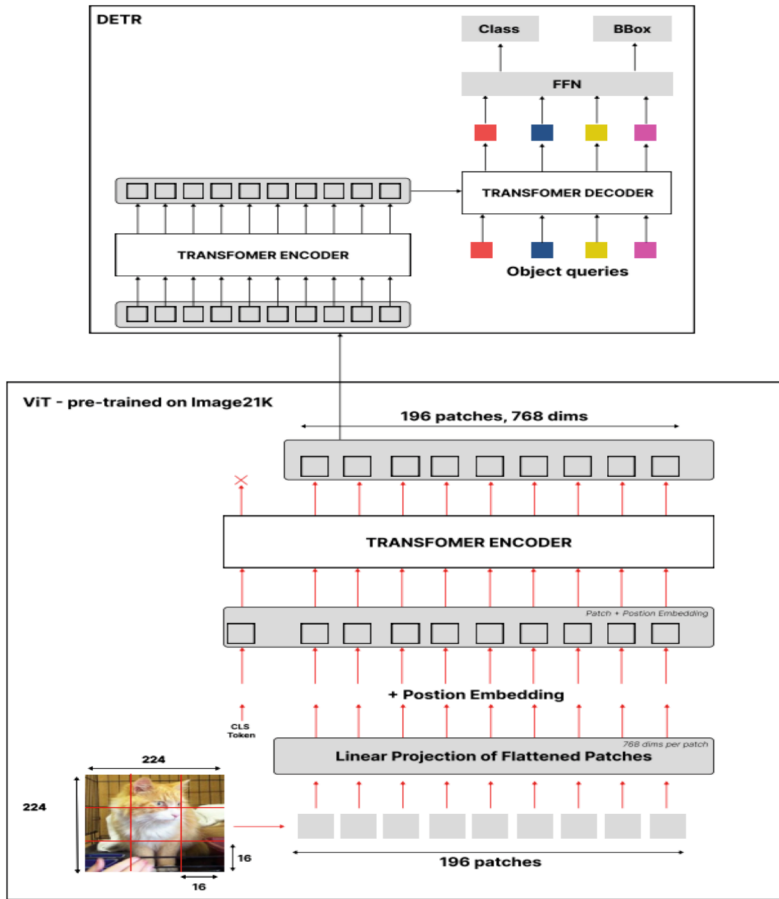


Fig. 3. Model combining DETR with ViT

V-DETR Pipeline: The model we propose to use ViT network as a backbone network for automatic feature extraction. The ViT network was previously trained with the ImageNet21K dataset, which is a very large dataset, with 21,000 object classes. The feature extraction result of the ViT architecture is a tensor of size [batch_size, 197, 768], which includes the CLS Token. To obtain features, we remove the CLS token and only keep the feature vectors obtained from the input fragments. Then, we use linear project technique to obtain the input form of the Encoder Transformer network with the following size [batch_size, 768, 14, 14]. We then convert the obtained features into the input form of the Transformer network. Of course, these features must also be combined with a position encoding vector to store relative position information between objects in the image. We let the feature map obtained through 6 Encoder Transformer blocks and 6 Decoder Transformer blocks stacked on top of each other. For each block in Transformer, we use 8 Multi-head Attention layers. The output vector from the above combination is transferred to the Encoder and decoder network of the Transformer network to perform the object detection task. Details of the model are shown in Fig. 3.

4 Experimental Results

Dataset. We conduct experiments on the PASCAL VOC 2012 dataset [25]. The Pascal Visual Object Class (VOC) Challenge is a multi-year challenge aimed at accelerating developments in the field of computer vision. The main goal of this challenge is to recognize objects from several object classes in real scenes. This is essentially a supervised learning problem where a training set of images is labeled. It includes 20 object classes, such as: person, bird, cat, ...

Implementation Details. We first pass the input image through the ViT model trained on the ImageNet21K [26] dataset (google/vit-base-patch16-224-in21k), which is a large image dataset with millions of images and thousands of object classes. Because backbone network has been trained with the ImageNet21K dataset with an image size of 224, the input image will be adjusted to a fixed size of $3 \times 224 \times 224$. After feature extraction with the ViT model, the resulting feature map has the size [batch_size, 197, 768]. We proceed to remove the CLS Token of the model and only retain the features of the input Patches, so the resulting feature map is [batch_size, 196, 768]. However, the input of the Encoder in the DETR model receives parameters of the form [batch_size, channel_number, H, W], so the feature map is obtained from the ViT model, we perform channel transformation and reshape with the size: [batch_size, 768, 14, 14]. The feature map obtained in the above stage will be transferred to the Transformer network to conduct classification and find bounding boxes. We keep the parameters of this network as Table 1.

Table 1. The list of parameters for model training.

Parameter	Value
Encoder layers	6
Decoder layers	6
Multi-head Attention	8
Number queries	100
Optimizer	AdamW
Batch_size	10
Learning_rate	1e-4
Learning_rate_backbone	1e-5
Epochs	100
Dropout	0.1
Weight decay	1e-4
Size_image	Any
Pre-trained Backbone	ImageNet21K

The experimental results on the train dataset are displayed in Fig. 4.



Fig. 4. Progressing of training task on error rate

Results on the evaluation dataset. The results of average precision measurement based on IoU levels of bounding box areas are displayed in Table 2 where AP and AR are average precision and average recall, respectively.

Table 2. Average precision and recall measurement based on IoU levels of bounding box areas.

Metric	IoU	Area	Max Detections	Value
AP	0.50: 0.95	all	100	0.244
AP	0.50	all	100	0.444
AP	0.75	all	100	0.240
AP	0.50: 0.95	medium	100	0.026
AP	0.50: 0.95	large	100	0.351
AR	0.50: 0.95	all	1	0.282
AR	0.50: 0.95	all	10	0.345
AR	0.50: 0.95	all	100	0.345
AR	0.50: 0.95	medium	100	0.085
AR	0.50: 0.95	large	100	0.480

The compared Results with some models [27] are displayed in Table 3.

Table 3. Results compared with some models.

Model	Year	Dataset	Backbone	mAP@0.5
Faster-RCNN	2015	Pascal VOC	VGG	42,9%
RetinaNet	2027	Pascal VOC	ResNet	43,2%
Def-DETR	2022	Pascal VOC	ResNet	30,1%
V-DETR	2023	Pascal VOC	ViT	44,4%

5 Conclusion

V-DETR is an end-to-end, efficient, and fast-converging object detector. This model instead of using a convolutional network as the traditional backbone network, the model uses a completely Transformer network from start to finish. The proposed pure transformer model is utilized to extract rich features for classification and high confident bounding box processing. Due to limited computational resources, the proposed model was evaluated on a rather small dataset, so the experimental results of the trained model are not expected to compare to the current SOTA solutions. The experimental results of

the V-DETR model are slightly better than the original DETR model. However, according to this experiment, the proposed approach is a potential method to yield high results when trained on large enough datasets. We hope our work opens up new possibilities for applying the Transformer model in computer vision tasks, especially object detection tasks.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning (in English). *Nature* **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
2. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587 (2014)
3. Girshick, R.: Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448 (2015)
4. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*, vol. 28 (2015)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)
6. Nguyen, H.T., Nguyen, M.N., Phung, L.D., Pham, L.T.T.: Anomalies detection in chest x-rays images using faster R-CNN and YOLO. *Vietnam J. Comput. Sci.* **10**(04), 499–515 (2023). <https://doi.org/10.1142/S2196888823500094>
7. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1904–1916 (2015)
8. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125 (2017)
9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788 (2016)
10. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271 (2017)
11. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. *arXiv preprint arXiv:1804.02767* (2018)
12. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464–7475 (2023)
13. Pham, T.-A., Hoang, V.-D.: Chest x-ray image classification using transfer learning and hyperparameter customization for lung disease diagnosis. *J. Inf. Telecommun.* (2024). <https://doi.org/10.1080/24751839.2024.2317509>
14. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*, pp. 21–37. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
15. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988 (2017)

16. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
17. Hoang, V.-D., Vo, X.-T., Jo, K.-H.: Categorical weighting domination for imbalanced classification with skin cancer in intelligent healthcare systems. *IEEE Access* **11**, 105170–105181 (2023)
18. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pp. 213–229. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
19. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint [arXiv:2010.04159](https://arxiv.org/abs/2010.04159) (2020)
20. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *Int. J. Comput. Vision* **104**, 154–171 (2013)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
23. Vaswani, A., et al.: Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30 (2017)
24. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
25. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (2010)
26. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **115**, 211–252 (2015)
27. Maaz, M., Rasheed, H., Khan, S., Khan, F.S., Anwer, R.M., Yang, M.-H.: Class-agnostic object detection with multi-modal transformer. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pp. 512–531. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20080-9_30