Kayak and Sailboat Detection Based on the Improved YOLO with Transformer

ZIYUAN LUO, Auckland University of Technology, New Zealand
WEI QI YAN, Auckland University of Technology, New Zealand
MINH NGUYEN, Auckland University of Technology, New Zealand

In this paper, we aim at sailboat and kayak detection from digital images using deep learning. The main dataset is created by ourselves, we have collected the images at the harbour near our city, Auckland. In the sailboat and kayak detection, we search for a set of best parameters for the baseline of YOLOv5 model. In this paper, we propose a spate of backbone structures for the purpose of comparisons, we are able to find out the best structure for the kayak detection using our training dataset. Finally, we verify our proposed model and compare it with an existing well-trained model by using ensemble learning. All results are obtained based on a computer with GTX1060 6G RAM GPU.

CCS Concepts: • **Computing methodologies** → **Object detection**; Supervised learning by classification; *Object recognition*; **Neural networks**.

Additional Key Words and Phrases: Kayak Detection, Sailboat Detection, YOLO, Transformer, Dataset

ACM Reference Format:
Ziyuan Luo, Wei Qi Yan, and Minh Nguyen. 2022. Kayak and Sailboat Detection Based on the Improved YOLO with Transformer. 1, 1 (May 2022), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In computer vision, visual object detection is a classical problem. The most popular method in object detection field is YOLO network. YOLO has been improved in the last 5 years, the versions of YOLO have experienced from YOLO to YOLOv5.

Pertaining to methodology of ship detection, especially in the field of sailboat and kayak detection, there are few research methods and outcomes as well as datasets. In a country with a unique boat culture like New Zealand, there are always many different boats on the water. We conduct this research with the hypnosis that it will assist races, companies or others with boat management. The ship images include heuristic scenes of all categories. Therefore, in this project, we provide a dataset based on the images of sailboats and kayaks. In addition, we also evaluate the performance of YOLO[8][22] and Transformer[29] models, we generate excellent results for the specified tasks. In this paper, we propose our baseline model for sailboat and kayak detection, then we improve the performance of our proposed model based on deep learning.

There are a spate of sailboats moored in the harbour of our cities. In this paper, we combine deep learning and ship culture together. The main purpose of visual object detection is to identify the target object. There are two-fold methods: One is two stages methods before 2016 for locating

Authors' addresses: Ziyuan Luo, Auckland University of Technology, Auckland, New Zealand, 1010; Wei Qi Yan, Auckland University of Technology, Auckland, New Zealand, 1010; Minh Nguyen, Auckland University of Technology, Auckland, New Zealand, 1010.

XXXX-XXXX/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

visual objects and detecting the objects; other methods are based on one stage, such as YOLO (You Only Look Once) model which has been widely cited since 2016 [22].

Before CNN has been applied to visual object detection, specified features have been adopted in conventional machine learning methods[30] popularly. It makes use of a sliding window to find object after resizing and scaling the given images. R-CNN model takes use of convolution backbone for extracting feature maps, then SVM is employed for classification[8]. Fast R-CNN[7] selected neural networks as the classifiers for pattern classification. Faster R-CNN[25] came out with an end-to-end model, under the support of an RPN net layer. Mask R-CNN[11] is implemented for instance segmentation and visual object detection.

The rest of this paper is organized as follows: In Section 2, we will review our literature; In Section 3, we detail our algorithm; In Section 4, we demonstrate our results. In Section 5, we will draw our conclusion and envision our future work.

2 LITERATURE REVIEW

The main novel point and difference of YOLO model are that it takes use of inferences between visual objects and its index in one single regression. YOLO9000 [23] was proposed after YOLOv2, which detected 9,000 classes of objects and improved the performance of visual object detection. Then, YOLOv3 [24] was put forward with the backbone Darknet-53. YOLOv4[1] was proffered in 2020 to solve the training problems much efficiently, which can execute the whole model within one GPU. YOLOv4 and YOLOv5 are the state-of-the-art methods at present. YOLOv5 improved the performance by adding a few of tricks on YOLOv4. YOLO networks have been applied to real-time object detection [31][18].

Transformer is a useful method that is well known after NLP method and BERT model[4] which was proposed in 2018. The BERT model is pre-trained by a huge text dataset, BERT[40] net has different NLP tasks. On the other hand, Transformer with multi-head algorithm is employed for modeling in images[35][39] and other fields after 2018.

There are two types of Transformer backbones in computer vision. DETR [2] backbone is a representative which implements Transformer as a pipeline inside CNN in visual object detection. In the field of computer vision, convolutional neural networks have always been well explored. Especially in object detection tasks, CNN is a classic choice [10][28]. Vision Transformer[6] implements only the pipeline without CNN for backbone structure.

The base and core method of Transformer[29] is attention [13]. The Facebook team has implemented YOLO model with attention[2]. The method is based on encoder-decoder structure between CNN backbones. There are also a pretty assortment of usages of Transformer[37–39], for example, the performance of machine learning methods for visual object detection can be improved by using Transformer.

3 METHODOLOGY

In this section, we will show our discussion related to the model structure and the differences between multiple structures.

3.1 Backbone

The core difference between YOLO model and other object detection method (e.g., R-CNN [7]) is that YOLO takes use of one inference to get the probability of index. YOLO treats object classification as a regression problem, but the other method such as R-CNN deals with the problem as an object recognition problem by using pattern classification, bounding box is employed for regression. Therefore, YOLO is more suitable for object detection problem than R-CNN.

The main structure of YOLO[22] network is a group of convolution input, then a fully connected layer is followed.

On the other hand, the activate function of YOLOv5 network is use of the leaky ReLU loss function, which contains the influence of the negative value during the training process. It is much suitable for regression than ReLU. The loss function needs to consider the errors between inference and the true index. Hence, a binary loss function will be useful in YOLO network, in this case, we select BCE-loss function for YOLOv5 model.

Transformer [29] is a popular method to solve the Seq2Seq problem. The main structure of Transformer is the encoder-decoder framework, which is an end-to-end algorithm to solve the sequence to sequence problem. Seq2Seq means the input is a sequence of data and the output is also a sequential data. The input process is called encoder and the output process is called decoder, all memory will be saved as context. [16]

All the methods associated with this framework are related to encoder and decoder. The Transformer has a multi-head attention method, which simulates mechanism of human attention. Attention method usually refers to computing convex combinations of content-based vector sequences, the weight itself is a function of the input.

$$MHAtten(x, q) = \sum_{h=1}^{N_h} Atten_{W_k, w_q, w_v, w_o}(x, q) \tag{1}$$

In this case, multi-head attention method is considered as the integration of low-dimensional original attention layer. In Section 3.2, multi-head attention will always be better than single- head attention. Therefore, multi-head attention Transformer is implemented between YOLO backbone and the fully connected layer.

The main idea of a Transformer is attention, the basic framework is the encoder-decoder structure. Therefore, in order to implement Transformer+YOLO model, we need to split the original YOLO network into two parts: Backbone and the fully connected layer. As stated, the backbone of YOLO is designed for extracting visual features, the fully connected layer is implemented for output. Thus, we implement Transformer and YOLO net together, we need to connect backbone output to the encoder, link the decoder output to the fully connected layer.

YOLO model for visual object detection needs to “look at once" which inferences the index with probability directly. However, before YOLO model was proposed, the method will be considered as R-FCN which is a method to inference object first and then predict the suitable index location. It took use of a moving 2D window to search on the map to find the most suitable index coordinate. R-FCN is a typical method for object detection, ResNet-101 net is applied as its backbone structure.

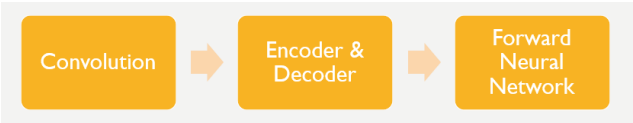

Fig. 1. The backbone of YOLO model with a Transformer.

Because deep learning methods are becoming stronger and stronger, the training dataset always needs a huge amount of data for training a deep net with the suitable weights, ImageNet[19] is a popular and useful dataset in computer vision, which contains a large number of images.

After the BERT model [4] was proffered, pre-training has been taken. The main idea of pre-training is to replace the weights of a random layer with a group of weights of the trained net. This method can be employed in the similar tasks, which may make a little bit change in feature

layer is called fine-tune or transfer learning. Based on the specific and unique characteristics of this method, we are use of the method for sailboat and kayak detection so as to save the training time.

3.2 Uniform Blending

By using the theoretical analysis of uniform blending, we have

$$G(x) = \frac{1}{T}\sum_{t=1}^{T} g_t(x) \tag{2}$$

$$\begin{aligned} avg((g_t(x) - f(x))^2) &= avg(g_t^2 - 2g_t f + f^2) && (3)\\ &= avg(g_t^2) - 2Gf + f^2 && (4)\\ &= avg((g_t - G)^2) + (G - f)^2 && (5)\\ &= avg(\epsilon(g_t - G)^2) + E_{out}(G) && (6) \end{aligned}$$

When any two models are mixed, it will record generalization error as $error_1$ and $error_2$. By using an ensemble learning method based on these two models, it shows,

$$E_{new} \leq aE_1 + (1-a)E_2 \tag{7}$$

In this case, if averaging two models together, the generalization error will always equal or less than the weight sum of each single model. Therefore, the blending method will be chosen at the end of this project to improve the performance of our proposed model.

Algorithm 1 Training a model

Input: $D_{tr}^{(N)}$: Training set; N: Number of total training images; CFG: Initial parameters; F_n: Fold number; lr: Learning rate

Output: Optimal Model: M^*; Out-of-fold Prediction set: P_{oof}

initial random state and model weight W_0;
repeat
  Set random seed R;
  Divide into $D_{tr}^{(N_{trs})}$ (training set) and $D_{val}^{(N_{val})}$ (valid set), where $N_{tr} = \frac{F_n - 1}{F_n}N$ and $N_{val} = \frac{1}{F_n}N$;
  Loading the YOLO model structure by using CFG;
  Loading the pre-training weight.
  repeat
    Set Adam optimizer and a stable learning rate lr;
    Training by using $D_{tr}^{(N_{trs})}$;
    Compute the target loss $Loss_{local}$ by using BCELoss
    Update $Loss_{best}$ if $Loss_{local} < Loss_{best}$
    Update saving Model M_{fold}^* if $Loss_{local} < Loss_{best}$
    Loading valid set and compute valid loss;
    Save valid loss with best model as out-of-folder result set P_{oof}.
  until $Epoch$ times OR $Loss_{best}$ has no change for 3 epoch.
until f_n times

Algorithm 2 Transformer modeling

Input: D_{tr}: Input image;
Output: Out: Model Out;

Image resize from D_{tr} to $\hat{D_{tr}}$;
Take Batch Normalization from $\hat{D_{tr}}$ to BN_D;
Implement BN_D Linear Transform and get L_1;
Reshape L_1 to multi-head from and select 3 dimension: q,k,v;
$q \times k$ and implement transpose, gives a_1;
Take Softmax to a_1 and gives a_2;
Implement Dropout at a_2 and gives a_3;
$a_3 \times v$ and transpose back to original image shape a_4;
Taking the linear transform to a_4 and get a_5;
Dropout a_5 and gives a_6 as attention output;
Implement Drop path for a_6 and gives attention layer out: a_7;
Add original input $\hat{D_{tr}}$ and a_7 gives output: Out_1;

Take Batch Normalization to Out_1 to get m_1;
Send m_1 into full connected layer 1 and get m_2;
Using GELU as activate function to m_2 and get m_3
Dropout m_3 and gives m_4 as MLP layer 1 output;
Send m_4 into full connected layer 2 and get m_5;
Dropout m_5 and gives m_6 as MLP layer 2 output;
Implement Drop path for m_6 and gives MLP layer out: Out_2;

Combine two layer together, then gives $Out = Out_1 + Out_2$

4 EXPERIMENTS

In this section, we detail the main process, key parameters, and methods. We will also explain the whole training and testing process as well as the details of parameter searching method.

4.1 Dataset

The dataset in this project was created by ourselves and contains approximately 1, 000 images of sailboats and kayaks. This dataset includes 600 images of kayaks, and 400 images of sailboats. Most of the sailboat photos were collected from local harbors, other pictures were from the America's Cup and the Olympic canoeing matches. The labels of each picture are tagged manually, the coordinates of each sailboat and kayak location are marked as the index of visual objects. Each photo contains at least one index, there are total more than 2,700 indexed images of sailboats and kayaks. Besides, there are a few of matches video from America's Cup and Olympic games were selected for testing our proposed model.

The training set takes 80% of the data from the dataset for training and 20% of the data for testing. It has enough visual objects of sailboat and kayak in the collected images, there are more than 500 indexes in testing set. In order to calculate the true generalization errors, all the testing set will not be used for training model. Validation set takes 20% of the training set, which was employed to implement a 5-fold of cross-validation[20] to evaluate the performance of each model.

To solve a object detection problem, it not only needs to recognize the target object but also demands to find the index of it. In this case, the model will give a series of object location and its

probability. Therefore, the evaluation method of the model will be based on the accuracy and the true value (with 1). The true value of true negative will be marked as '0', the false positive will be marked as '1'.

4.2 Implementations

In this section, we will probe how to get the best parameters of each models after training. As described, a pre-training method will be employed for this task. The ImageNet-1k [19] pre-training weight is taken as the initial value of YOLO network, which will include the weights of all layers. We start the first run with epoch 10, learning rate as 1.0×10^{-4}, and the default binary cross-entropy loss.

The quality of dataset will influence the performance of the proposed model. The image size of input photo will impact the efficiency of training process. The dataset is collected by using mobile cameras, original image resolution is around 4000×4000, it will waste a lot of computing resouces, YOLOv5 net will set the input image size as 640×640 [21]. Under this condition, the input images will also be resized as 512×512 and 256×256.

Table 1. Training progress of multiple input sizes of images

Data-set	*Quantity*	*Image size*	*Epoch*	*Time*
S&K-1000-Original	2787	640×640	3	5.70 (h)
S&K-1000-Cleaned	2749	512×512	3	4.50 (h)
S&K-1000-Cleaned	2749	256×256	3	3.60 (h)

The modeling environment is a 6G RAM GPU. From the testing result, we decide the input size by considering the hardware, it shows 512×512 will generate better training parameters and an excellent performance.

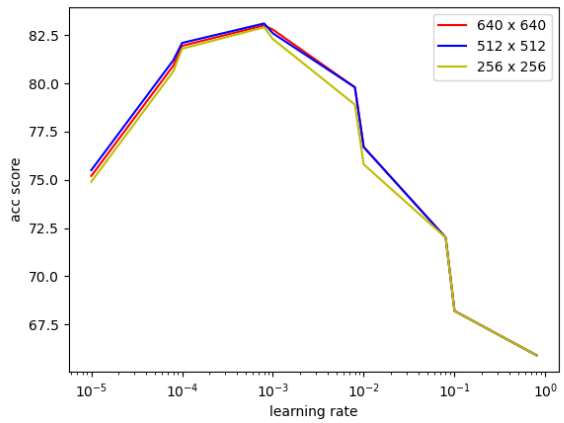

Fig. 2. We compared the results of learning rate searched by using previous input size 640×640, 512×512, 256×256. Finally, it shows 512×512 to have the best accuracy with 83.10%. To test the best learning rate, the epoch will be set as 50.

We freeze the parameters before searching for the best learning rate. Fig. 2 shows 8.00×10^{-4} is a good learning rate for YOLOv5 model. Other training method will start by using this group parameters.

In the next, we will test the loss function to replace cross-entropy function. The log loss, exponential loss, hinge loss and categorical cross-entropy functions are employed to get a better accuracy, because this problem is much like a binary classification. According to the training outcomes, the categorical cross-entropy function is the best one with YOLOv5 model.

Table 2. Cross-validation scores with multiple loss functions

Loss function	***Epoch***	***Minimal Loss***	***CV Score***
Cross-entropy	5	0.0018	0.2314
Log loss	5	0.0033	0.3124
Exponential loss	5	0.0037	0.2928
Hinge loss	5	0.0089	0.2882
Categorical cross-entropy	5	0.0012	0.2135

[a] All methods are working at the epoch 5

From Table. 2, we see the training progress converged if epoch equals to 5, so we set all methods with epoch as 5, this will save our computing time and get the best outcomes. Thus, we take use of the same parameters in the Transformer model.

The best parameter of YOLOv5 baseline is the input size 512×512, learning rate with 0.0008, and the categorical cross-entropy function is set as the loss function. In order to continue improving the performance of the baseline model, we ensemble the models to expect a better CV score.[12] The core ensemble learning method will include the voting and blending, where the blending method is applied to calculate the probability(confidence) of each class of visual objects, the voting is to decide which object it is at last.

Each model will generate a group of prediction results with probabilities. In order to blend object detection results, it needs to count how much index in totally, and assign each index with other objects. It needs to make sure that all indexes have a series of prediction with label “Kayak”, “Sailboat”, and “Other Boat”. Then, we will calculate the average of each model and assign every model a weight. The next step is to take the weighted mean value as the final probability. In order to find the best weight of each model, we search for the best value by using the cross-validation prediction result.

After got the probability of each model, the next step is to vote with these models. Each model has the vote value as same as the probability, it will determine which class received the most votes, the final class will be marked as it. On the other hand, if the probability is too low, it will also be regarded as a wrong prediction, and this index will be removed. Table 3 shows the ensemble learning method reduces the cross-validation error and gives a better result.

Table 3. Ensemble learning results

Model Structure	***Number***	***CV Score***	***Test set Score***
YOLOv5	1	0.2298	0.3014
Transformer	2	0.2043	0.2833
1+2 Ensemble	3	0.1989	0.2798

4.3 Results

Fig. 4 shows that the index is correct but it gives a probability around 80%. In this case, the final accuracy will be much better, but the cross-validation error and generalization error will be influenced by this “un-confidence" probability.

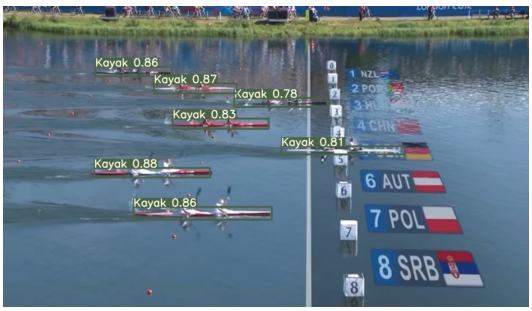

Fig. 3. The result with a high confidence and the output with a high probability

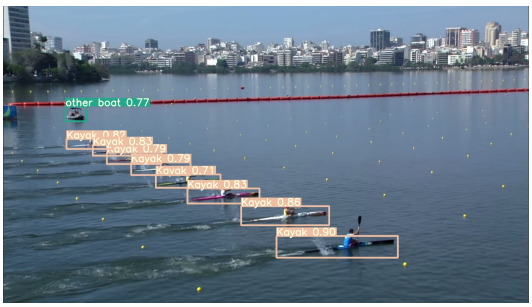

Fig. 4. The test image from Olympic games, which contains Kayaks and other boats. It shows that the output of the target model will contain a group of indexes and probabilities.

5 CONCLUSION AND FUTURE WORK

In this project, we implement a method for detecting kayaks, sailboats and other types of boats, we successfully blend YOLOv5 and Transformer together for visual object detection. For each model, our experiments have been conducted to find the best variables and parameters such as input size, learning rate, and the best loss function. Finally, we ensemble these models and get a model with a better cross-validation error and generalization error. In next stage, we will look for other suitable structures and expect to get better results [17, 33, 34].

REFERENCES

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with Transformers. In *European Conference on Computer Vision*. Springer, 213–229.

[3] LDAA George Dahl, Jack Stokes, and Li Deng. 2012. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. 20 (2012), 30–42. Issue 1.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[7] Ross Girshick. 2015. Fast R-CNN. *arXiv e-prints* (2015).

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.

[9] Qishen Ha, Bo Liu, and Hongwei Zhang. 2021. Google Landmark Retrieval 2021 Competition Third Place Solution. *arXiv preprint arXiv:2110.04619* (2021).

[10] Ryo Hasegawa, Yutaro Iwamoto, and Yen-Wei Chen. 2020. Robust Japanese road sign detection and recognition in complex scenes using convolutional neural networks. *Journal of Image and Graphics* 8 (2020), 59–66.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *IEEE Conference on Computer Vision*. 2961–2969.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[13] Huanhuan Ji, Zhenbing Liu, Wei Qi Yan, and Reinhard Klette. 2019. Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. In *Asian Conference on Pattern Recognition*. Springer.

[14] Kyungmin Kim, Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Zhicheng Yan, Peter Vajda, and Seon Joo Kim. 2021. Rethinking the self-attention in vision Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3071–3075.

[15] Woosuk Kim, Hyunwoong Cho, Jongseok Kim, Byungkwan Kim, and Seongwook Lee. 2020. Target classification using combined YOLO-SVM in high-resolution automotive FMCW radar. In *IEEE Radar Conference (RadarConf20)*. IEEE, 1–5.

[16] Ying Liu, Luyao Geng, Weidong Zhang, Yanchao Gong, and Zhijie Xu. 2021. Survey of video based small target detection. *Journal of Image and Graphics* 9 (2021), 122–134.

[17] Ziyuan Luo, Minh Nguyen, and Wei Qi Yan. 2021. Sailboat detection based on automated search attention mechanism and deep learning models. In *International Conference on Image and Vision Computing New Zealand*. IEEE.

[18] Mansi Mahendru and Sanjay Kumar Dubey. 2021. Real time object detection with audio feedback using YOLO vs. YOLO_v3. In *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 734–740.

[19] Anamika Maurya and Satish Chand. 2021. Exploiting Pre-trained encoder with receptive fields and squeeze-excitation module for road segmentation. In *International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 139–143.

[20] Krystian Mikolajczyk and Cordelia Schmid. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 10 (2005), 1615–1630.

[21] Jong-Chan Park, Hye-Youn Lim, and Dae-Seong Kang. 2021. Predicting Rebar endpoints using sin exponential regression model. *arXiv preprint arXiv:2110.08955* (2021).

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.

[23] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7263–7271.

[24] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems* 28 (2015).

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[27] Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2016. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2016), 640–651.

[28] Florian Spiess, Lucas Reinhart, Norbert Strobel, Dennis Kaiser, Samuel Kounev, and Tobias Kaupp. 2021. People detection with depth silhouettes and convolutional neural networks on a mobile robot. *Journal of Image and Graphics* 9 (2021), 135–139.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[30] Paul Viola and Michael Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1. IEEE, I–I.

[31] Tian-Hao Wu, Tong-Wen Wang, and Ya-Qi Liu. 2021. Real-time vehicle and distance detection based on improved YOLOv5 network. In *World Symposium on Artificial Intelligence (WSAI)*. IEEE, 24–28.

[32] Cheng Xu, Weimin Wang, Shuai Liu, Yong Wang, Yuxiang Tang, Tianling Bian, Yanyu Yan, Qi She, and Cheng Yang. 2021. 3rd place solution to Google landmark recognition competition 2021. *arXiv preprint arXiv:2110.02794* (2021).

[33] Wei Qi Yan. 2019. *Introduction to Intelligent Surveillance - Surveillance Data Capture, Transmission, and Analytics, Third Edition.* Springer.

[34] Wei Qi Yan. 2021. *Computational Methods for Deep Learning: Theoretic, Practice and Applications.* Springer Nature.

[35] Jing Zhang, Jun Wu, Hui Wang, Yuchen Wang, and Yunsong Li. 2021. Cloud detection method using CNN based on cascaded feature attention and channel attention. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–17.

[36] Li Zhang and Yuxuan Hu. 2021. A fine-tuning approach research of pre-trained model with two stage. In *IEEE International Conference on Power Electronics, Computer Applications (ICPECA).* IEEE, 905–908.

[37] Liming Zhang and Weisi Lin. 2013. Background of Visual Attention-Theory and Experiments. (2013).

[38] Liming Zhang and Weisi Lin. 2013. Validation and Evaluation for Visual Attention Models. (2013).

[39] Jingyuan Zhou, Chak Tou Leong, and Congduan Li. 2021. Multi-scale and attention residual network for single image dehazing. In *International Conference on Intelligent Computing and Signal Processing (ICSP).* IEEE, 483–487.

[40] Ya Zhou and Can Tao. 2020. Multi-task BERT for problem difficulty prediction. In *International Conference on Communications, Information System and Computer Engineering (CISCE).* IEEE, 213–216.