When Active Learning and Data Augmentation meet at Object Detection

Yicheng Xiao
12210414@mail.sustech.edu.cn

Shengli Zhou
12212232@mail.sustech.edu.cn

Abstract

Object detection and tracking is currently crucial to the development of robotics, autonomous system and vision-related tasks. Different models are proposed to recognize the latent patterns in real world data and a vast variety of methods are proposed to improve the performance in terms of accuracy and latency. We will be using the current state-of-the-art model RT-DETR, which stands for Real-Time Detection Transformer. Also, we will be exploring relevant techniques to enhance model's performance. Active learning is a method to incrementally select data that the model interests from the dataset along the training iterations to reducing the usage of data, while data augmentation is a method to enlarge the dataset and shift the distribution in order to align the trained distribution with the true distribution. This project aims to study the effect of the combination of these two methods and study how model learn the true distribution through these methods.

1. Background and Significance

The field of object detection has emerged as a cornerstone of modern technological innovation, driving significant advancements in automation, security, and efficiency across various industries. Object detection involves the identification and classification of objects within digital images or videos, enabling machines to interpret and interact with their environment in a manner similar to human perception. This capability is pivotal for the development of autonomous systems, such as self-driving cars, drones, and robots, which rely on accurate and reliable object detection to navigate and perform tasks safely and efficiently [2].

The importance of object detection extends beyond these applications to broader societal benefits. By automating routine tasks, it frees up human resources for more complex and creative endeavors, thereby boosting productivity and innovation.

The field of object detection is not only technologically fascinating but also economically and socially significant. Its potential to revolutionize multiple sectors makes it a highly valuable area of study and research, attracting substantial investment and attention from both academic institutions and industry leaders.

Despite the current detection models having achieved promising results, research [11] indicates that the capabilities of these models have not been fully utilized. If given more training data, the models could achieve even better outcomes. However, at the same time, there are challenges in expanding the training dataset due to the high costs of visual data collection and annotation. In this project, we will address this issue from two perspectives: active learning and data augmentation.

2. Analysis of Current Research Status

Object detection is a fundamental task in computer vision with widespread applications in autonomous driving, surveillance, robotics, and healthcare. The goal of object detection is to localize and classify objects in an image or a video stream, typically represented as bounding boxes around objects with corresponding class labels. Recent advancements in deep learning, particularly convolutional neural networks (CNNs) and transformers, have revolutionized the field.

2.1. Recent Progress in Object Detection

Recent research in object detection can be classified into several key areas:

2.1.1. Two-Stage Object Detection (Region-Based CNNs)

Two-stage methods first generate region proposals and then classify and refine those proposals. This paradigm has been dominant for several years, with frameworks such as R-CNN [5] and its variants such as Fast R-CNN [4], Faster R-CNN [14], and Mask R-CNN [6] continuing to perform well. These methods excel in accuracy, but the region proposal stage can be computationally expensive.

- Faster R-CNN improved the region proposal network (RPN) to generate region proposals directly

Field	Advantages	Disadvantages	Research Points
Two-Stage Object Detection	High accuracy, robust for overlapping or small objects, well-established frameworks like R-CNN variants	Slower inference due to region proposal step, higher computational cost, requires more training time	Enhancing efficiency and inference speed, integrating with transformer models, reducing region proposal network overhead
Single-Stage Object Detection	Faster inference, suitable for real-time applications, simpler design and implementation, models like YOLO and SSD are widely used	Lower accuracy compared to two-stage methods, struggles with small or overlapping objects, susceptible to imprecise boundary localization	Bridging the accuracy gap with two-stage methods, improving detection of small objects, enhancing generalization in complex scenes
Transformer-Based Detection	Captures long-range dependencies, strong generalization across diverse tasks, flexible feature extraction	High computational cost and latency, requires large datasets for training, complex to implement	Scaling transformers for detection tasks, reducing model size and computational requirements, incorporating multimodal input (e.g., text + image)
3D Object Detection	Captures spatial and depth information, crucial for autonomous systems (e.g., driving), processes multimodal data (LiDAR + RGB)	Computationally intensive, requires high-quality data for robustness, limited real-time capabilities	Improving LiDAR-based methods, enhancing multimodal data fusion, increasing real-time performance

Table 1. Comparison of Object Detection Methods

within the network, significantly improving speed while maintaining high accuracy.
- Mask R-CNN extended Faster R-CNN by adding a branch for object segmentation, enabling pixel-level object detection.

2.1.2. Single-Stage Object Detection

Single-stage object detection methods eliminate the need for region proposal networks, offering a more efficient approach. Notable single-stage detectors include YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector).

- YOLO [12] pioneered real-time object detection by framing it as a regression problem that simultaneously predicts class labels and bounding box coordinates.
- SSD [9] combined multi-scale feature maps to detect objects at various scales, improving accuracy and real-time performance.

Recent work on single-stage detectors has focused on improving accuracy, handling small objects, and optimizing the trade-off between speed and accuracy.

2.1.3. Transformer-based Object Detection

Transformers, initially developed for natural language processing, have been adapted to object detection with architectures like DETR [1] and Swin Transformer [10].

- DETR used the transformer architecture for end-to-end object detection without the need for anchor boxes or non-maximum suppression. While DETR has improved the quality of object localization, it faces challenges in handling small objects and long training times.
- Swin Transformer introduced a hierarchical transformer structure that performs well across a wide range of tasks, including object detection, by using shifted windows to model local and global information.

2.1.4. 3D Object Detection

While 2D object detection has been extensively studied, the detection of 3D objects from point clouds and images is gaining traction, particularly for autonomous driving. Recent models such as PointPillars [8] and VoxelNet [15] have shown significant progress by directly processing LiDAR data or combining it with RGB images to detect objects in 3D.

2.2. Recent Progress in Data Augmentation

Current Data Augmentation methods can be categorized into four subsets, Geometric Transformations Photometric Transformations, Random Occlusion and Deep Learning based Approaches. [7]

Geometric transformations modify the pixel positions in an image within the training dataset. These transformations include rotation, scaling, flipping, cropping, padding, translation, and affine transformations, among others.

Photometric augmentations modify the pixel values or intensities, primarily by altering the image's color components. These include adjustments to brightness, contrast, hue, saturation, and adding noise. Color is a key aspect of an image, often helping to recover the scene's physical properties.

Random occlusion augmentations simulate occluded images during training, helping the model learn better by providing more varied experiences. Common methods include random erase, cutout, hide and seek, grid mask, cutmix, and mosaic augmentation.

The above augmentations do not always recognize all the disparities in the environments. These techniques can sometimes lead to the loss of information or features from the original dataset, as they alter the geometry or lighting conditions of the images. However, deep learning-based data augmentation methods offer more effective transformations. Notable techniques include Neural Style Transfer (NST), adversarial training, and Generative Adversarial Networks (GANs).

2.3. Recent Progress in Active Learning

Active learning (AL) is a machine learning approach that aims to maximize a model's performance gains while minimizing the number of labeled samples annotated. The fundamental premise of AL is to select the most informative samples from an unlabeled dataset for labeling by an oracle (e.g., a human annotator). By doing so, AL reduces the overall labeling cost while striving to maintain high model performance. AL can be categorized based on different application scenarios into three approaches: membership query synthesis, stream-based selective sampling, and pool-based sampling.

Membership query synthesis allows the learner to request labels for any unlabeled sample in the input space, including samples generated by the learner itself. Stream-based selective sampling evaluates each sample individually in a data stream to decide whether to query its label, while pool-based sampling selects the best query sample from the entire unlabeled dataset based on evaluation and ranking.

Traditionally, AL has shown significant potential in enhancing labeling efficiency, achieving considerable performance improvements with fewer labeled samples. However, early methods in AL often struggled with high-dimensional data, primarily due to their inability to effectively leverage the complex relationships present in such datasets. Consequently, the integration of active learning with deep learning (DL), termed Deep Active Learning (DeepAL), emerged as a promising solution to bridge this gap. By combining AL's sample selection strategies with DL's feature extraction capabilities, DeepAL aims to address the high annotation costs involved in generating labeled datasets, especially in expert-intensive fields like medical imaging and speech recognition. Overall, AL continues to be a vital research area, providing mechanisms that can greatly enhance the efficiency and effectiveness of machine learning processes across various domains.[13]

3. Method

3.1. Problem Statement

3.1.1. Preliminaries

In this project, we are exploring the SOTA model RT-DETRv2 performance using Active Learning and Advanced Data Augmentation methods.

The finetune dataset we use is Kitti 2017, which is a dataset targeted for autonomous driving system with categories Cars, Pedestrians and Cyclers. After transforming the Kitti Dataset into Coco Format, the input for the model now containes Type, 2D BBox and Occluded information.

The metrics used are as follows:

- Mean Average Precision (mAP): Measures the accuracy of the detections.
- Intersection over Union (IoU): Evaluates overlap between predicted and ground-truth bounding boxes.

Then, we will be using video collected in the campus to test the real-time object detection performance of our finetuned model.

Conventionally, Active Learning is a way for us to evaluate which data is more valuable for training. The earlier chosen data will have more impact over the whole training process. By using Data Augmentation, we enlarge our dataset, thus providing more samples with larger variety for learner to choose from. Studying the selection strategy can help understand the possible bias lied inside the training process.

3.1.2. Active Learning

In Active Learning, the core idea of our strategy is to formulate the uncertainty of the model w.r.t. a predicted bounding box in the image. The more uncertain the model is about the predicted outcome, the more

random the predicted values become, and the closer the predicted distribution is to a uniform distribution. Therefore, we can formally define the model's uncertainty as:

$$U = \mathcal{D}_{\mathrm{KL}}(\mathcal{P}||\mathcal{U}) \tag{1}$$

where $\mathcal{P}$ is the predicted distribution and $\mathcal{U}$ is uniform distribution. By the definition of KL divergence, we have:

$$U = \int_x \mathcal{P}(x) \log\left(\frac{\mathcal{P}(x)}{k}\right) \mathrm{d}x \tag{2}$$

where k is a constant. Thus,

$$\begin{aligned} U &= \int_x \mathcal{P}(x) \left[\log \mathcal{P}(x) - \log k\right] \mathrm{d}x && (3)\\ &= \int_x \mathcal{P}(x) \log \mathcal{P}(x) \mathrm{d}x - \int_x \mathcal{P}(x) \log k \ \mathrm{d}x && (4)\\ &= \int_x \mathcal{P}(x) \log \mathcal{P}(x) \mathrm{d}x - \log k \int_x \mathcal{P}(x) \mathrm{d}x && (5)\\ &= \int_x \mathcal{P}(x) \log \mathcal{P}(x) \mathrm{d}x - \log k && (6)\\ &= -\mathbb{H}(w) - \log k && (7) \end{aligned}$$

Hence, as the model needs to find instances with the largest uncertainty for the oracle to label, it is equivalent to choose the predictions with the largest entropy.

Similarly, according to BALD [3], we can also measure the information gain of the model according to the expected entropy in the model space.

3.1.3. Data Augmentation

Given a labeled dataset, we can use $\mathcal{D}_L = \{X, y\}$ to represent it, where X stands for the data, and y stands for the labels. Data augmentation can be represented by a certain function f_θ, such that the augmented dataset $\hat{\mathcal{D}_L} = \{\hat{X}, \hat{y}\}$ can be derived by a certain function f_θ.

$$f_\theta : \mathcal{D}_L = \{X, y\} \longrightarrow \hat{\mathcal{D}_L} = \{\hat{X}, \hat{y}\}$$

In RT-DETR-v2, basic transformations including zooming, cropping, resizing are applied. However, stronger augmentation methods are not implemented in the RT-DETR-v2 framework. We therefore conducted several individual augmentation methods on the dataset. The modification of individual augmentation methods can be regarded as a shift from its original distribution to an another distribution by adding a perturbation $\epsilon(x_i)$. The goal of data augmentation is to approximate the true data distribution $P_{\mathrm{true}}(x)$ by expanding the observed data distribution $P_{\mathrm{data}}(x)$. To evaluate the similarity between augmented and original data distributions, we can use KL Divergence

$$\mathcal{D}_{\mathrm{KL}}(P_{\mathrm{aug}}||P_{\mathrm{true}}) = \sum_x P_{\mathrm{aug}}(x) \log \frac{P_{\mathrm{aug}}(x)}{P_{\mathrm{true}}(x)}$$

Therefore, by evaluating the effect of the Data Augmentation methods using Active Learning, we are hoping to find what data augmentation methods can help us minimize the KL Divergence.

3.2. Proposed Methodology

3.2.1. Active Learning

To evaluate the impact of Active Learning on enhancing Detection models, we apply Active Learning strategies to the real-time detection model RT-DETR. Starting from epoch 0, the model will perform inference on all unlabeled data after every Δ_E epochs' training and select the top Δ_K images that the model is most interested in. The model keeps selecting images until all images are selected.

In terms of data selection strategy, we employed three methods: random, entropy, and information gain. By comparing the improvements of the latter two strategies over the random strategy in terms of performance increase and reduction in training steps, we can determine if the model has achieved better results through Active Learning. This method also helps us select a more effective Active Learning strategy.

3.2.2. Data Augmentation

We use python package albumentations to help us with the Data Augmentation process. By using the package, we can easily control the data augmentation methods by using several parameters.

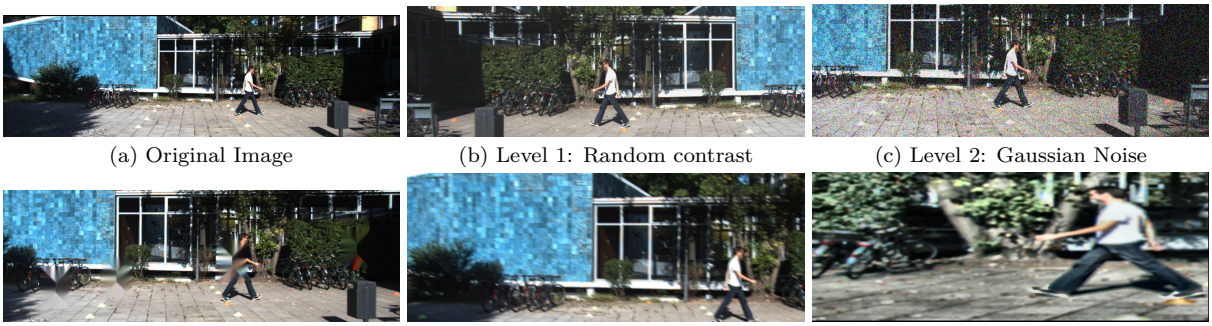
We separate our augmentation methods into different intensity levels, namely light, medium and strong. Then we test the combining effect and separate effect of different augmentation methods to analyze which data augmentation methods will help us approximate the reality distribution.

4. Experiments

4.1. Experiment Setup

The experiments are conducted under the following environment:

- CPU: 16 Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz 256GB Memory with no swap area.
- GPU: 8 NVIDIA GeForce RTX 2080 Ti with 12GB Memory.



(a) Original Image (b) Level 1: Random contrast (c) Level 2: Gaussian Noise

(d) Level 3: Grid Dropout (Inpainting) (e) Level 4: Motion Blur (f) Level 5: Strong Augmentation

Figure 1. Augmentation Methods demonstrated.

- Dataset: Kitti 2017.
- Base model: RT-DETR-v2.

In the comparative experiment for evaluating the effect of active learning, we choose $\Delta_E = 5$ and $\Delta_k = 1000$, i.e., the model choose 1000 images after each 5 epochs.

4.1.1. Data Augmentation Operations

We have chosen five set of operations that are not contained in the default data augmentation methods provided by RT-DETRv2 itself.

Table 2. Summary of Data Augmentation Methods (All contains possible horizontal flip and random crop)

Level	Transformations Applied
1	Random brightness and contrast.
2	Gaussian noise.
3	Grid dropout (in-painting style).
4	Motion blur.
5	RGB shift, Blur, random brightness and contrast, CLAHE (for contrastness enhancement).

Referenced data augmentation examples are shown in Grid 1.

4.2. Experiments and Results

4.2.1. Active Learning Strategy Selection

In this experiment, we train three RT-DETR-v2 models using Active Learning based on Random, Entropy and Information Gain respectively. To evaluate model's performance, we record the best cumulative score in Figure 2. The line graph illustrates that by applying either Entropy or Information Gain as Active Learning strategy, the performance under same training cost and the required training steps for a certain score outperforms Random Selection strategy.

Under non-randomized Active Learning strategies, the model first learns more common cases with higher learning rate and forms a basis for its knowledge. Then, by utilizing such prior knowledge, the model can perform selection more informatively and select samples that are more uncertain to the model. Which both addresses the problem of unbalance-sampled data and provides the model with training data of "special cases". Finally, the model fine-tunes its knowledge on these special cases. As the fine-tuning process is regularized by the prior knowledge, the model can limit the impact of special cases. Thus improving model's performance.

4.2.2. Data Augmentation Operation Selection

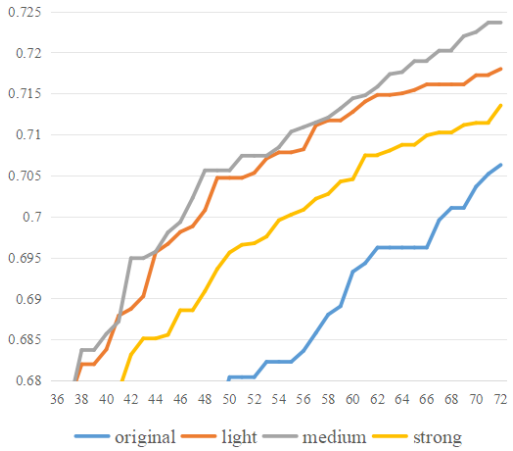


Figure 3. The cumulative best mAP score - training epoch line graph. Light uses Level 1 data augmentation. Medium uses Level 3 data augmentation. Strong uses Level 4 data augmentation. All experiments are done under Active Learning with Information Gain Selection strtegy with $\Delta_K = 2000$.

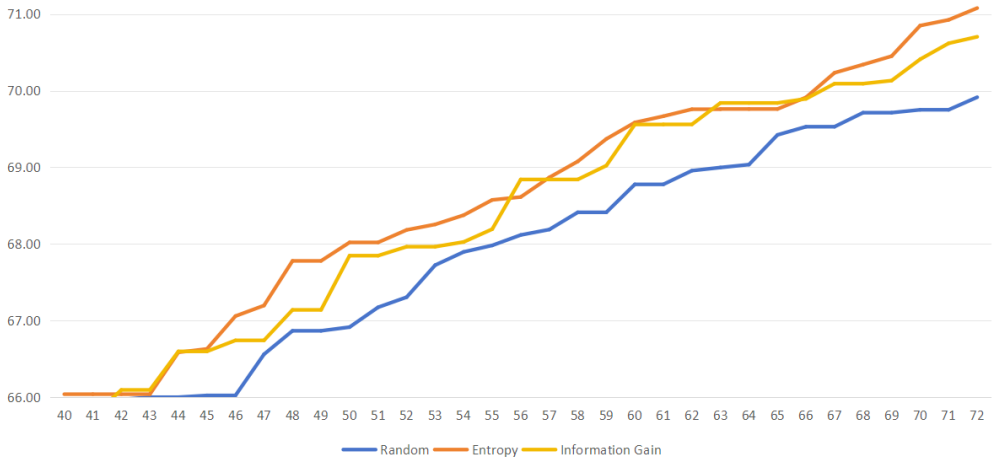


Figure 2. The cumulative best mAP score of the models trained under different Active Learning strategies. When the model utilizes either Entropy or Information Gain as Active Learning strategy, the performance under same training cost and the required training steps for a certain score outperforms Random Selection strategy.

By analyzing the Light, Medium and Strong dataset separately, we can see that the Medium dataset can lead to the highest accuracy, which indicates that the Level 4 data augmentation methods can better approach the true distribution of the test dataset.

4.2.3. AL Selection Strategy on Augmented Data

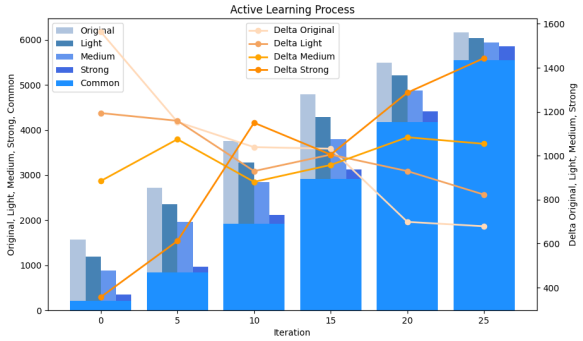


Figure 4. Active Learning selection distribution on augmented data. We profile the selection distribution over each selection epoch Δ_E. Also, the *delta* fields document the increment/decrement in each selection epoch compared with the last epoch.

From this profiling results, we can see that at each selection operation, the Active Learning strategy tends to select more original data than Light dataset than Medium dataset than Strong dataset. Therefore, there must be some underlying latent distribution for each augmentation strategy to affect the selection of the active learning strategy.

From individual analysis, we can see that the model tends to select Original data than augmented data. However, as the augmented level grows lighter, the selection becomes more even. Also, the increment for selecting the augmented data grows as the selection epoch increases. We can see that usually at Epoch 10, the model tends to select more augmented data than the original one. The common pictures are calculated based on the original image id. If the AL selects two pictures with the same base picture, we increment the common picture size by 1. As we can see from the results, the number of common pictures decreases as we use stronger augmentation strategies.

5. Discussion

According to Figure 4, the model shifts the selection from light-augmented data to heavily-augmented data. This is because the latter's latent variables tend to distribute more sparsely. Moreover, as the distribution of latent variables of images in Kitti dataset differs from that in Coco dataset, there are more light-augmented data having variables that distributes far away from Coco's distribution at the beginning and those images are selected by the model. During training, the model generally shifts its distribution from Coco's distribution to Kitti's distribution. Thus, more heavily-augmented data are selected by the model. Hence, the shift in the type of data selected by the model verifies the correctness of our Active Learning strategy.

Furthermore, as shown in Figure 3 and Figure 4, Active Learning strategies can be used to evaluate the quality of data augmentation under model's perspective (i.e., its uncertainty) and goal's perspective (i.e.,

(a) Light dataset selection profile. (b) Medium dataset selection profile. (c) Strong dataset selection profile.

Figure 5. Invidual profiling results with different strategy.

the model's performance). Hence providing a guidance for applying more useful augmented data for object detection.

6. Conclusion

In this project, we explore two common strategies to improve the performance of Object Detection model RT-DETRv2. Then, through conducting combining experiments, we discover a way to combine these strategies together to help us understand the principle of Active Learning and Data Augmentation. This may help create auto selection methods for Data Augmentation strategies and Active Learning pair calculation algorithm. Our final results can exhibit a 75.4% accuracy over the test dataset, which is a large-margin 6% increase from our initial baseline. Finally, by analyzing the results, we prove the positive effect of applying Active Learning strategies and provide a guidance for data augmentation in object detection task.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. arXiv preprint arXiv:2005.12872, 2020. 2

[2] Bogusław Cyganek. Object Detection and Recognition in Digital Images: Theory and Practice. Wiley, 2013. 1

[3] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, page 1183–1192. JMLR.org, 2017. 4

[4] Ross Girshick. Fast r-cnn. arXiv preprint arXiv:1504.08083, 2015. 1

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014. 1

[6] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017. 1

[7] Parvinder Kaur, Baljit Singh Khehra, and Er. Bhupinder Singh Mavi. Data augmentation for object detection: A review. In 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pages 537–543, 2021. 3

[8] Alexander H Lang, Saurabh Vora, Holger Caesar, Bin Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12697–12705, 2019. 2

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, David Cournapeau, Alexander C Berg, and Milan Sonka. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 2

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030, 2021. 2

[11] Yadan Luo, Zhuoxiao Chen, Zijian Wang, Xin Yu, Zi Huang, and Mahsa Baktash. Exploring active 3d object detection from a generalization perspective. ArXiv, abs/2301.09249, 2023. 1

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016. 2

[13] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. ACM Comput. Surv., 54(9), 2021. 3

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 1

[15] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4490–4499, 2017. 2