Accurate and Robust Object Detection and Classification Based on Deep Neural Networks

Kaidong Li

Submitted to the graduate degree program in Department of Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

,

,

,

Committee members ,

,

,

,

Date defended: January 29, 2024

The Dissertation Committee for Kaidong Li certifies
that this is the approved version of the following dissertation :

Accurate and Robust Object Detection and Classification Based on Deep Neural Networks

____
,

Date approved: January 29, 2024

Abstract

Recent years have seen tremendous developments in the field of computer vision and its extensive applications. The fundamental task, image classification, benefiting from deep convolutional neural networks (CNN)'s extraordinary ability to extract deep semantic information from input data, has become the backbone for many other computer vision tasks, like object detection and segmentation. A modern detection usually has bounding-box regression and class prediction with a pre-trained classification model as the backbone. The architecture is proven to produce good results, however, improvements can be made with closer inspections. A detector takes a pre-trained CNN from the classification task and selects the final bounding boxes from multiple proposed regional candidates by a process called non-maximum suppression (NMS), which picks the best candidates by ranking their classification confidence scores. The localization evaluation is absent in the entire process. Another issue is the classification uses one-hot encoding to label the ground truth, resulting in an equal penalty for misclassifications between any two classes without considering the inherent relations between the classes. Ultimately, the realms of 2D image classification and 3D point cloud classification represent distinct avenues of research, each relying on significantly different architectures. Given the unique characteristics of these data types, it is not feasible to employ models interchangeably between them.

My research aims to address the following issues. (1) We proposed the first location-aware detection framework for single-shot detectors that can be integrated into any single-shot detectors. It boosts detection performance by calibrating the ranking process in NMS with localization scores. (2) To more effectively back-propagate gradients, we designed a super-class guided architecture that consists of a superclass branch (SCB) and a finer class branch (FCB). To further increase the effectiveness, the features from SCB with high-level information are fed to FCB to guide finer class predictions. (3) Recent works have shown that 3D point cloud models are extremely vulnerable to adversarial attacks, which pose a serious threat to many critical applications like autonomous driving and robotic controls. To gap the domain difference in 3D and 2D classification and to increase the robustness of CNN models on 3D point cloud models, we propose a family of robust structured declarative classifiers for point cloud classification. We experimented with various 3D-to-2D mapping algorithms, bridging the gap between 2D and 3D classification. Furthermore, we empirically validate that the internal constrained optimization mechanism effectively defends adversarial attacks through implicit gradients.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to my advisors, Professor Guanghui Wang and Professor Cuncong Zhong. Their devotion to academic research inspires me every day in my doctorate research journey. Professor Wang provides a rigorous and well-balanced academic plan, which paves a clear path for me. His kind personality and research insight helped me to focus on the journey. And Professor Zhong provides valuable advice both in research and life.

I also would like to thank my committee members, Professor Bo Luo, Professor Fengjun Li, Professor Haiyang Chao, and Professor Taejoon Kim for their insights, support and time. I sincerely appreciate people from ITTC support team whose response is always professional and immediate.

I have worked closely with Professor Ziming Zhang and Dr. Kuan-chuan Peng. Professor Zhang's meticulous approach to research deeply influences me. And Dr. Peng gives me inspiration on research methodology.

My sincere thanks also go to all my labmates, especially Wenchi Ma, Xiangyu Chen, Tianxiao Zhang, and Usman Sajid, for the support, discussions, and resource-sharing. And I would like to thank my friends for their accompanies.

Lastly, I would like to express my deepest gratitude to my family for their unconditional support and love. A special acknowledgment goes to my parents who deserve credit for every little achievement I have along this journey. I appreciate my mom's influence on my attitude towards life. I appreciate my dad's talks and advice. I have never told him how much I love him, and now he must have known.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent years have witnessed many breakthroughs in vision-related applications, like facial recognition, autonomous driving, etc. In this chapter, we will show the current state-of-the-art and its limitations. Then I will discuss the motivations, propose my ideas, and present some expected contributions.

1.1 Background and Related Work

In this section, I will introduce the fundamentals of deep convolutional neural networks (CNNs), vision transformers and some research directions of the computer vision field. With the development of computational power and a large number of labeled datasets [15; 16] in recent years, deep-learning-based models have been proven to be very effective in extracting intrinsic structure and high-level features. This property makes deep neural networks especially effective in areas where the data are multi-dimensional and the features are difficult to hand-engineer, including speech recognition [17; 18; 19], natural language processing [20; 21] and computer vision [2; 22; 23; 24; 25].

Computer vision has witnessed the most prominent improvement from deep CNNs [26; 27] and attention mechanism [24; 28]. Directions like image classification, object detection, segmentation, tracking, etc. have promoted rapid growth in areas that are closely related to daily life, like autonomous driving, face recognition, and countless functions on social media platforms.

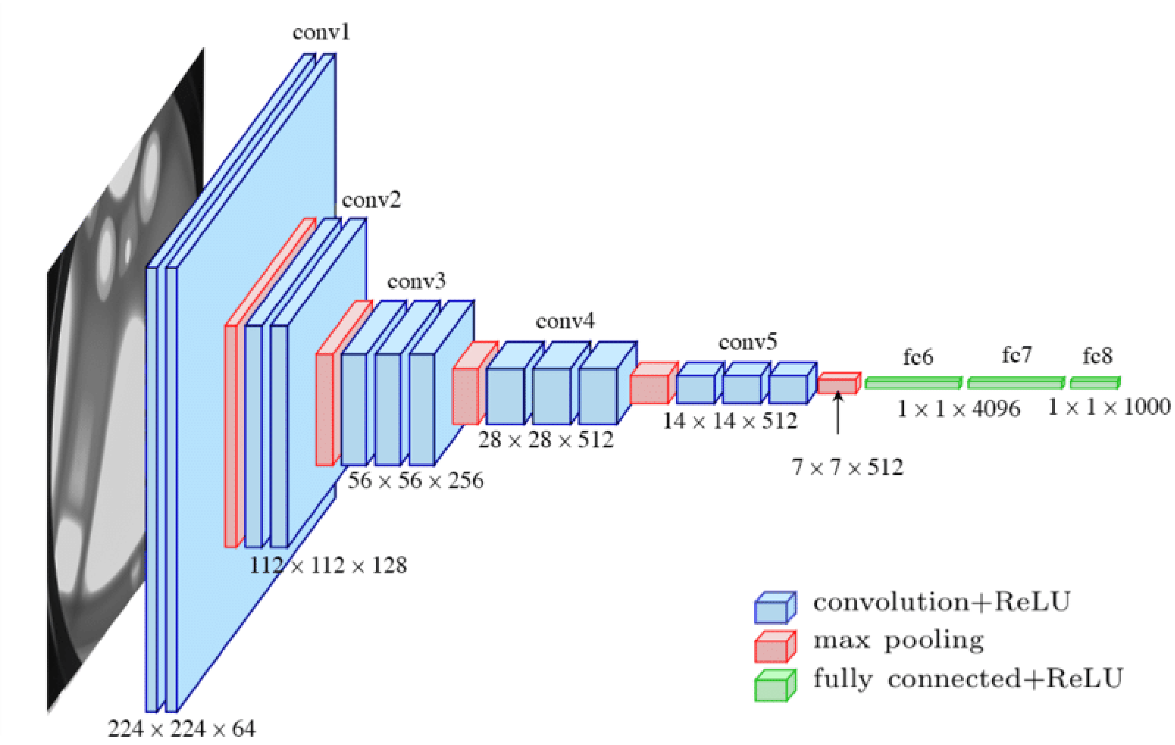


Figure 1.1: Structure of VGG-16 [1]

1.1.1 Deep CNN

Deep CNNs are the basic structure for most computer vision tasks. A deep CNN model commonly consists of a cascade of convolutional layers and fully connected layers where the convolutional layers can extract low-level information from shallow layers and high-level information from deep layers. And the fully connected layers can collect global semantic information to generate final predictions. An example structure is shown in Figure 1.1, which depicts the architecture of a VGG-16 classifier [1]. At certain levels of the convolutional layers, there are pooling layers to reduce feature dimensions, which at the same time gather information from neighboring regions. After each convolution layer, activation layers (e.g. Rectified Linear Units (ReLUs) and sigmoid) are attached to introduce non-linearity. Between convolution and ReLU layers, normalization layers are added to ensure the same scale for different data dimensions.

Convolutional Layer is the major building block for deep CNNs. A basic convolutional filter, also

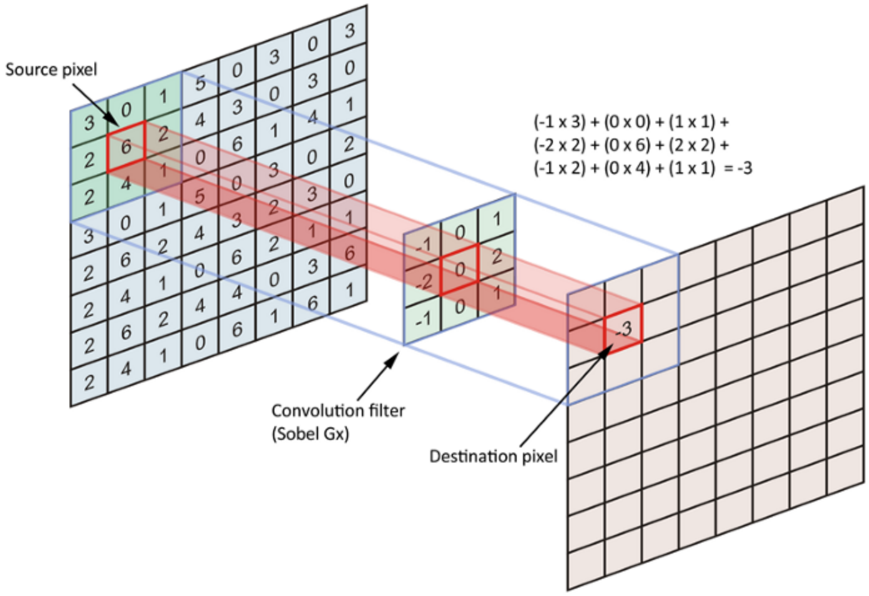

Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel
-1 0 1
-2 0 2
-1 0 1
-3

Figure 1.2: How convolutional filter works

known as the kernel, is shown in Figure 1.2. The output of a kernel contains information of the corresponding neighboring 9 pixels from input, when it is a 2D 3×3 kernel. The weights in the kernel are learned during the training process.

The common kernel size in recent years tends to be 3×3 rather than larger sizes. As pointed out by Simonyan *et al.* [1], there are two reasons behind the choice. 1) Small kernels are more efficient. A larger kernel can always be replaced by a stack of multiple 3×3 kernels. The stack of small kernels has fewer parameters compared to a single large kernel when their effective receptive fields are the same. 2) Multiple smaller kernels have more non-linear activation layers, making them more discriminative. There are other ideas like dilated convolutions [29], where downsampling is done by dilated convolution.

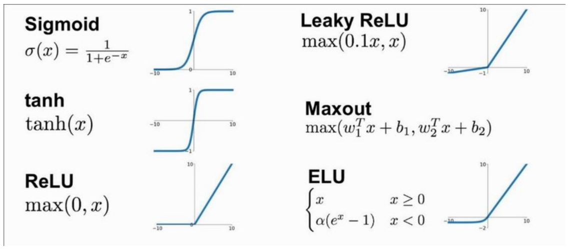


Figure 1.3: Commonly used activation functions

Activation Layer is a crucial element in CNNs, which introduces non-linearity into the CNNs. In a neuron of brain, the synapse controls its own synaptic strengths' influence on another neuron. An activation layer has a similar function. Some of the most commonly used

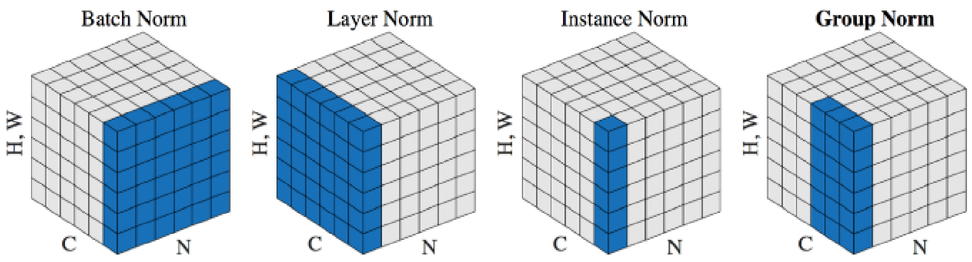


Figure 1.4: Different normalization methods. The blue part is the features to be normalized in each method. (H,W) are the height and width of a feature. C axis is the channel axis. N stands for batch size.

activation functions are shown in Figure 1.3.

Normalization is effective in stabilizing the CNN learning process by reducing internal covariate shift and normalizing the contribution of each feature. Some of the most popular normalization methods (batch norm [30], layer norm [31], instance norm [32] and group norm [33]) are shown in Figure 1.4.

Pooling Layer gathers and extracts information from neighboring regions to reduce feature dimension and allow convolutional layers to have a wider receptive field. Max pooling, average pooling and global pooling are the three most common pooling layers. The first two are easy to understand as the names suggest. The global pooling basically reduces all features in a channel to one value, making a $w \times h \times c$ feature to $1 \times 1 \times c$.

Fully Connected Layer connects all values from one layer to the next which collects all information from input. It's usually at the end of a classification task to generate the final prediction.

Overall, a deep CNN understands a visual input by extracting low-level local information from shallow layers and obtaining high-level global information in deeper layers using the building blocks introduced above. Figure 1.5 shows a visualization of first-layer kernels. Some of the kernels are effectively edge detec-

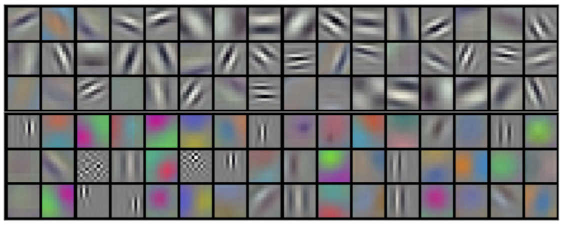


Figure 1.5: Visulization of first layer $11 \times 11 \times 3$ convolutional kernels learned in ImageNet [2] classification task

tors.

1.1.2 Vision Transformer

Another revolutionary breakthrough is the vision transformer, an idea inspired by language models. Transformers primarily use self-attention mechanism [34], which can better understand long-range dependencies and global knowledge. Intuitively it is firstly applied to natural language processing (NLP), with many successful subsequent models (BERT [35], GPT [36], etc.).

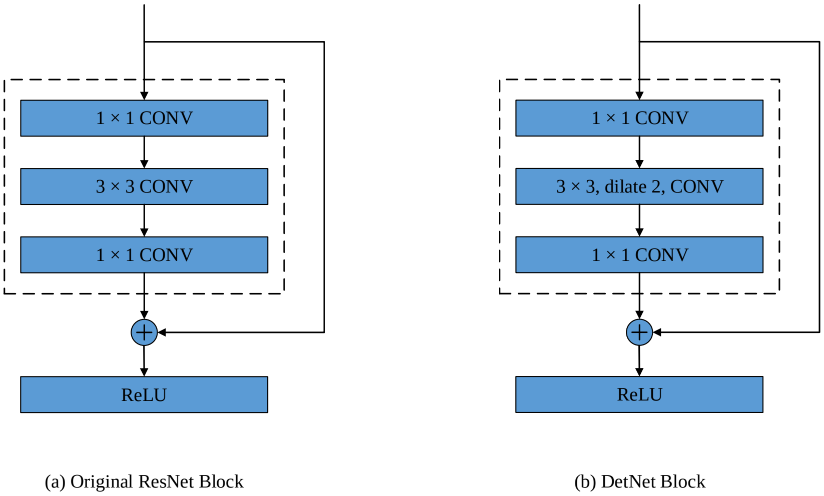
Such success naturally leads researchers to try to employ transformers in different modalities. ViT [24] is a seminal work that proposed an end-to-end transformer-based image classification model. It divides images into smaller patches and feeds the patches to the transformer. Since then, many successful applications based on transformer have been proposed, *e.g.* DETR [25] in detection, SWIN-Transformer [37] in segmentation, etc.

1.1.3 Image Classification

Image classification is fundamental for all computer vision tasks. All other vision tasks take the pretrained models from image classification as their backbone network. Thus it's crucial to improve the performance of classification models. Image classification is basically the task of taking an image as input and predicting the class of the most dominant object in the input image.

DetNet [4]: DetNet is a backbone network specifically designed to extract features, different from other detectors discussed in this section. It is designed to tackle three existing problems in previous backbone networks:

- Backbone networks have a different number of stages;
- Feature maps used to detect large objects are usually from deeper layers, which have a larger receptive field, while they are not accurate in exacting the location due to low resolution;
- Small objects are lost as the layers go deeper and resolutions become lower.



(a) Original ResNet Block

(b) DetNet Block

Figure 1.6: **DetNet structure.** The diagram shows the basic building block of ResNet [3] and DetNet [4]. (a) After each ResNet block, the resolution is reduced in half. (b) The DetNet preserves the feature map resolution and increases the receptive field by using dilated convolutions.

Li *et al.* [4] proposed DetNet-59 based on ResNet-50. It has 6 stages with the first 4 stages the same as ResNet-50. In stages 5 and 6, the spatial resolutions are fixed instead of decreasing. The fixed resolution means a convolution filter will have a smaller receptive field compared to that in lower-resolution feature maps. A dilated [29] bottleneck as shown in Figure 1.6(b) is used for compensation. In this paper, we apply the DetNet backbone to the Faster RCNN detector.

1.1.4 Object Detection

Object detection takes images as input and generates classification results of objects presented in the images and their corresponding location information. Object locations are most commonly defined by rectangular bounding boxes. The output of image segmentation contains more detailed information, such as the classification result for each pixel in the original photo, while object detection usually only produces the coordinates of four corners of each bounding box. Thus, image segmentation is usually more time-consuming. In practice, pixel-level classification is not necessary for polyp detection and classification. In this study, we focus on object detection techniques. The state-of-the-art deep learning-based object detection models can be broadly classified into two main categories: two-stage detectors and one-stage detectors.

Two-stage detector consists of a region proposal stage, followed by a classification stage. Each stage has its own dedicated deep CNN, which generally produces higher accuracy than one-stage detectors. However, this also leads to more processing time. The region proposal stage used to be the bottleneck as it is often a slow process, while the state-of-the-art two-stage detectors adopt new structures sharing part of the CNN to speed up the processing time for real-time applications [38].

One-stage detector gets rid of the region proposal stage and fuses it with the classification stage, resulting in a one-stage framework. It directly predicts bounding boxes by densely sampling the entire image in a single network pass. With simpler architecture, it often achieves real-time performance. Although earlier models had lower detection accuracy than the two-stage detectors, they are catching up and now can produce comparable results.

In this section, we will make a brief introduction of eight state-of-the-art object detection and classification models that are implemented and evaluated in this comparative study.

Faster RCNN [38]: Faster RCNN is a two-stage framework model and one of the families of RCNN networks [39; 40]. It improves the Fast RCNN network by replacing the slow selective search algorithm with a region proposal network, resulting in a faster detection rate. Furthermore, the region proposal network is trainable, which can potentially achieve better performance.

Faster RCNN is mainly composed of two modules, the region proposal network (RPN) module and the classification module, as shown in Figure 1.7. First, the backbone network (for example, ResNet 101 [3]) extracts feature maps from the input image. The features are then shared by both the RPN module and the classification module. In the RPN branch, a sliding window will be applied to regress the bounding box locations and probability scores of object and non-object. At each location, the sliding window predicts k pre-defined anchor boxes, centered at itself with different sizes and ratios to achieve multi-scale learning. With the introduction of RPN, the inference time on PASCAL VOC is reduced to $198ms$ on a K40 GPU with VGG-16 as the backbone [38]. Compared to the selective search, it is almost 10 times faster. The computational time of the proposal stage is reduced from $1,510ms$ to only $10ms$. Combined, the new faster R-CNN can achieve

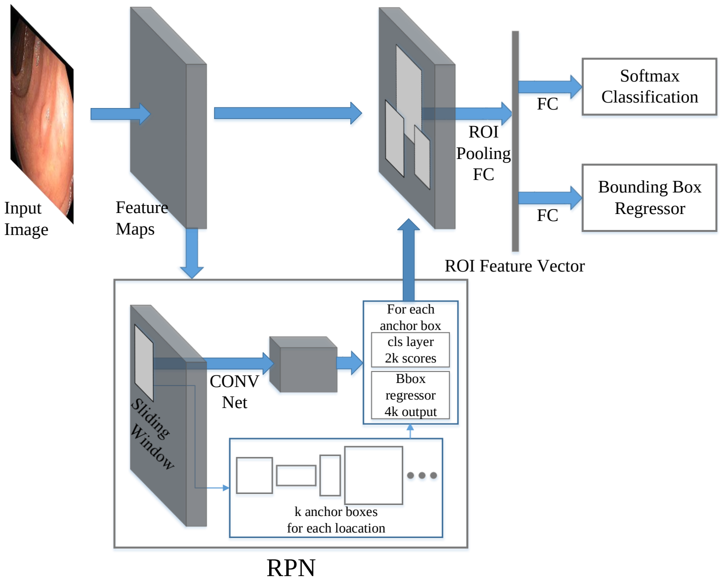


Figure 1.7: **Faster R-CNN structure.** Region proposal network (RPN) shares the same base CNN with a fast R-CNN network. The region proposal is generated by sliding a small convolutional network over the shared feature maps, and these proposals are used to produce final detection results.

5 frames per second (fps).

YOLOv3 [41]: YOLOv3 is an iterative improvement of YOLO (You Only Look Once). It improves the performance of its previous versions by introducing a new backbone network, multi-scale prediction, and a modified class prediction loss function.

YOLO is the first model of this YOLO series [42; 43; 41]. It is one of the pioneering works to get rid of the region proposal stage. The detector splits the image into $S \times S$ grids. Each cell is responsible for predicting ground truth objects with centers located inside the cell, and each cell in the grid predicts $B \times (4+1+C)$ values, where B is the number of anchor boxes in each cell, $4+1$ represents the number of bounding boxes and object confidence, and C is the total number of classes. The second version, YOLOv2, and YOLO9000, introduced several optimization tricks to improve the performance like batch normalization, high-resolution classifier, new network, multi-scale training, etc. Among the optimizations, the most effective technique is dimension priors which limit the regressed bounding boxes close to its original anchors. Without it, the regressed boxes can go anywhere in the image, resulting in unstable training [43]. YOLOv3 progressively developed a deeper CNN, DarkNet-53, from DarkNet-19 [41]. It also predicts objects from different scales. YOLOv3 achieves real-time performance. However, it often has lower detection accuracy compared to Faster RCNN.

YOLOv4 [44]: YOLOv4 is the latest improvement of YOLO. It explores the bag of freebies and bag of specials and selects some of them in the new detection model. The basic rules for a detection model are high-resolution input images for detecting relatively small objects, deeper layers for a larger receptive field, and more parameters for detecting various objects. Based on those rules, YOLOv4 selects various effective bag of freebies and bag of specials to enhance the performance of the model while maintaining high-speed inference. In addition, instead of exploiting DarkNet53 as the backbone in YOLOv3, an enhanced version of DarkNet53 (CSPDarknet53 [45]) is selected as the backbone for YOLOv4. Higher receptive field is extremely important to detectors, thus SPP [46] net is added over the backbone CSPDarknet53 [45] since this block provides larger receptive fields with almost the same inference time. YOLOv3 utilizes FPN [47] to aggregate the information from different feature levels, while YOLOv4 [44] employs PANet [48] to extract information for detector heads. Bag of freebies and bag of specials are indispensable for object detection and properly selecting and adding them to the detection models may highly boost the performance of the detectors without sacrificing too much inference cost.

SSD [49]: Single Shot Detector (SSD), as one of the most successful one-stage detectors, has become the foundation of many other studies. It takes advantage of the different sizes of feature maps and utilizes a simple architecture to generate predictions at different feature map scales. SSD can achieve a fast detection rate with competitive accuracy.

As shown in Figure 1.8. SSD combines multi-scale convolutional features to improve prediction. In CNN, feature maps progressively decrease in size from input to output. The layers closer to the input are shallow layers which have higher resolution and are better at detecting smaller objects. While the deeper layers have lower resolutions but contain more semantic information. SSD takes advantage of this natural structure of CNN and yields comparable results for objects of all sizes. SSD is an anchor-based detector. It divides the image into $m \times n$ grids similar to the YOLO

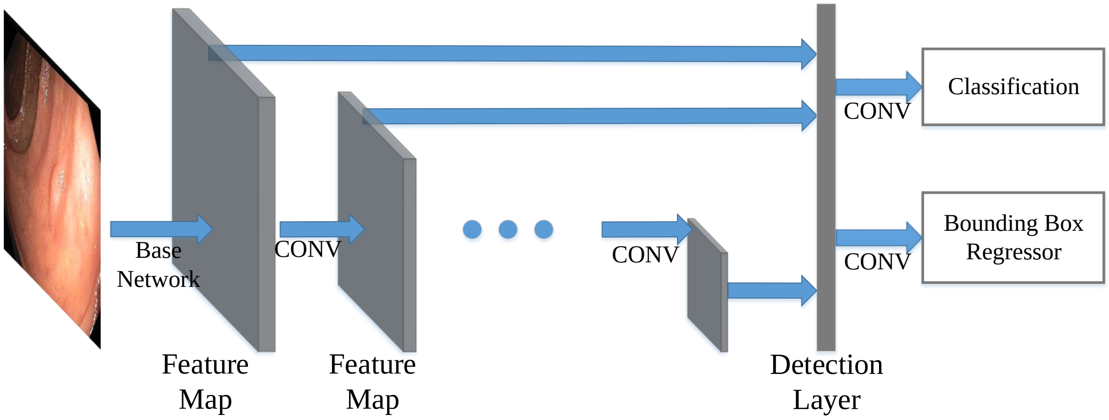


Figure 1.8: **SSD structure.** The base network is truncated from a standard network. The detection layer computes confident scores for each class and offsets to default boxes.

series. At each grid cell, the model will generate per-class scores and bounding box dimension offsets for each *k* pre-defined anchors with different ratios and scales, similar to RPN in Faster RCNN. It also introduces the use of convolutional layers for prediction which makes the detector fully convolutional, unlike YOLO [42] which uses fully connected layers for detection.

SSD makes a good trade-off between speed and accuracy. The simple one-stage framework architecture results in fast performance, achieving a real-time detection rate. Furthermore, the use of anchor boxes and multi-scale prediction enables good detection accuracy.

RetinaNet [50]: RetinaNet is a one-stage framework based on the SSD model. RetinaNet improves performance by using the Feature Pyramid Network (FPN) [47] for feature extraction and focal loss function to solve the class imbalance problem. In the SSD model, the multi-scale prediction mechanism suffers from its architectural weakness in which high-level layers do not share information with low-level layers, thus lacking high-level semantic information in detecting smaller objects. FPN concatenates feature maps from layers at different depths to improve detection at each scale. Another major contribution of this model is the use of focal loss to solve the class imbalance problem. Class imbalance refers to the imbalance between background and foreground class. It is more extreme in one-stage models as the detector scans through the entire image indiscriminately. In practice, the candidate locations can normally go up to 100*k* without the filtering of the region proposal module. Therefore, the focal loss is introduced to assign higher weights to difficult foreground objects and lower weights to easy background cases. The definition of focal loss is defined in Equation 1.1, where the balance variant, α_t, and focusing parameter, γ, are two hyper-parameters and p is the estimated probability.

$$
\begin{aligned}
FL(p_t) = &- \alpha_t(1-p_t)^{\gamma} log(p_t), \\
&\text{where } p_t = \begin{cases} p, & \text{if correct detection} \\ 1-p, & \text{otherwise} \end{cases}
\end{aligned}
\tag{1.1}
$$

In Equation (1.1), p_t is closer to 1 when the model is more correct (i.e., correct prediction with higher confidence score or wrong prediction with lower confidence score). With the original cross-entropy loss as $CE = -\alpha_t log(p_t)$, focal loss effectively gives it a factor $(1-p_t)^{\gamma}$, whose value is small when the model is correct (easy cases) and large when the model is wrong (hard cases).

RefineDet [51]: RefineDet is an SSD-based detector aiming at overcoming the following three limitations in single-stage detectors compared to the two-stage ones.

- Single-stage models lack region proposal module to eliminate the overwhelming background objects, causing inefficient learning;
- Two-stage models have both region proposal module and classification module to regress final bounding box output while one-stage models only have one stage to refine box location;
- Single-stage models generate only one set of feature maps for both tasks of localization and classification. Although recent two-stage models share the same backbone CNN, they have separate branches attached at the end of the main backbone networks for localization and classification specifically.

The architecture of RefineDet is shown in Figure 1.9. It consists of three modules: Anchor refinement module (ARM), transfer connection block (TCB), and object detection module (ODM).

Like in SSD, ARM takes feature maps from different layers. Then from each layer, it produces coarsely adjusted anchors and binary class scores (object and non-object classes). The anchors with a non-object score greater than a certain threshold θ will be filtered out, which reduces the class imbalance. Then the TCB is designed to combine features from deeper layers to the current level ARM features by element-wise addition. Deconvolution is used to facilitate the addition by increasing the resolution of deeper layer feature maps to match the shallow features. As a result, the shallow layers will have semantic information. By taking the filtered anchors from ARM and feature maps produced by TCB, ODM regresses the already refined anchors and generates multi-class scores. The results are improved because the input of ODM contains multi-level information and it refines the predicted bounding boxes in two steps.

ATSS [52]: ATSS (Adaptive Training Sample Selection) investigates the anchor-based object detectors and anchor-free object detectors and points out how to define positive samples and negative samples in the training process is the significant difference between the anchor-based models and anchor-free models. For instance, the anchor-free detector FCOS [53] first finds positive candidate samples in each feature level and then selects the final positive candidates among all features, while the anchor-based RetinaNet [50] exploits IoU (Intersection over Union) between pre-defined anchors and the ground truth bounding boxes to directly select the final positive samples among all feature levels [52]. Based on the analysis, ATSS automatically defines positive and negative candidates based on the statistical properties of the objects in the images.

For each object on the image, ATSS selects k anchor boxes based on the closest center distance between those samples and the ground truth box on each feature level. There are a total of $k \times L$ candidate positives if the number of feature pyramid levels is L. Then the IoU between these candidate samples and the ground truth is calculated and the mean m_g and standard deviation v_g are also calculated so that the IoU threshold is obtained as $t_g = m_g + v_g$. Finally, the candidates whose IoU are larger than or equal to the threshold and at the same time whose centers are inside the ground-truth box are selected as the final positive samples. ATSS introduces a mechanism that dynamically selects the positive and negative samples and bridges the gap between anchor-based

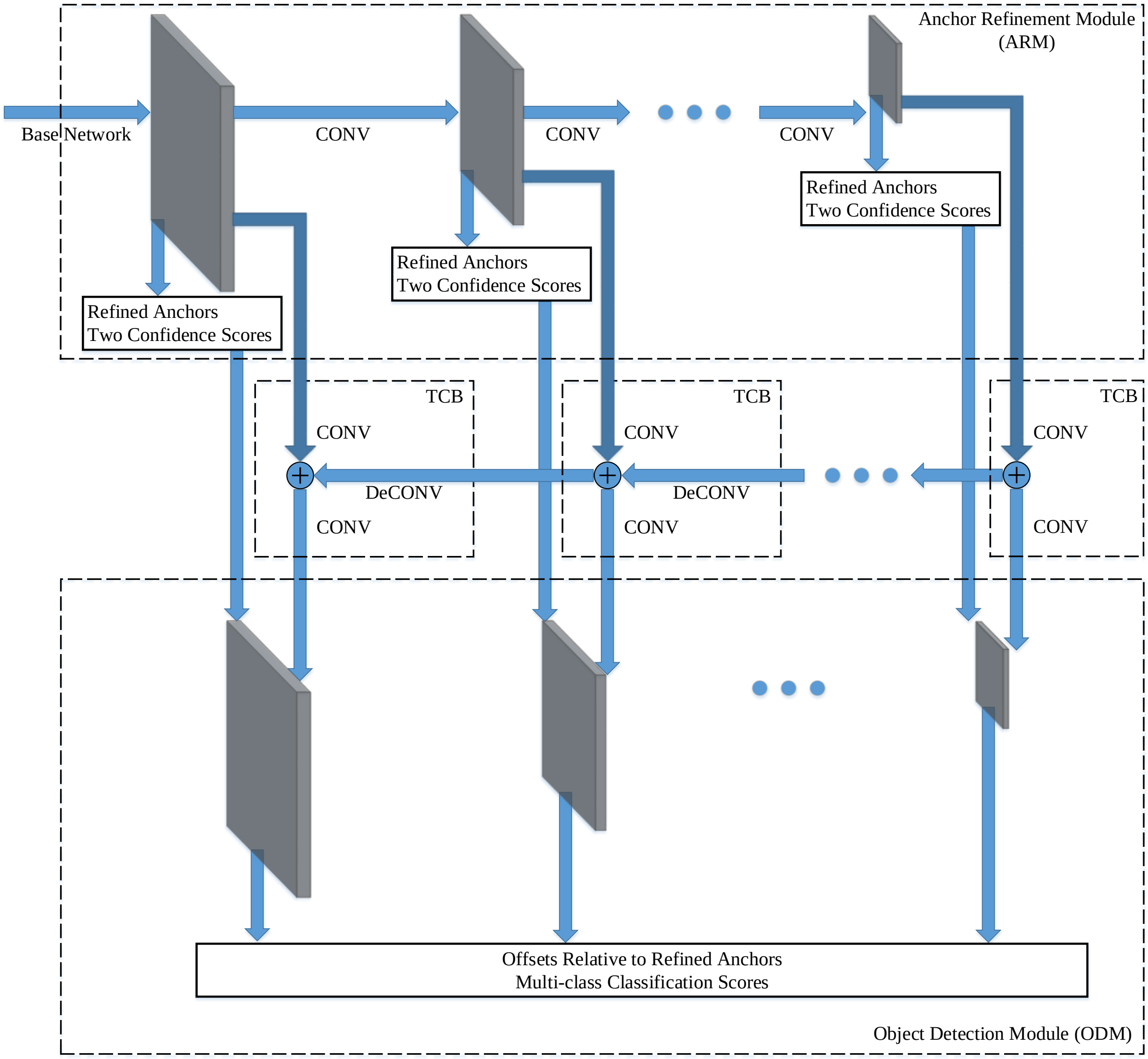


Figure 1.9: **RefineDet structure.** The architecture has three modules: Anchor refinement module (ARM), transfer connection block (TCB), and object detection module (ODM).

approaches and anchor-free approaches.

1.1.5 3D Models

Based on the point cloud representations, we simply group some typical deep networks into four categories. *Point-based networks* [54; 55; 56; 57] directly take each point cloud as input, extract point-wise features using multi-layer perceptrons (MLPs), and fuse them to generate a feature for the point cloud. *Image-based networks* [58; 59; 60; 61; 6] often project a 3D point cloud onto a (or multiple) 2D plane to generate a (or multiple) 2D image for further process. *Voxel-based networks* [62; 11; 63; 64; 65] usually voxelize each point cloud into a volumetric occupancy grid and further some classification techniques such as 3D CNNs are used for the tasks. *Graph-based networks* [66; 67; 68; 69; 70] often represent each point cloud as a graph such as KNN or adjacency graph which are fed to train graph convolutional networks (GCNs). A nice survey can be found in [71].

1.2 Limitations and Expected Contributions

In this section, I will briefly introduce the limitations of current computer vision tasks. More detailed analysis will be provided in each corresponding section.

1.2.1 One-hot Encoding in Classification

For the classification task, most models only take class names as input. All classes are treated equally without any correlation information. During training, the misclassification between "chair" and "couch" and that between "TV" and "rabbit" receive the same penalty. While in reality, "TV" and "rabbit" share almost no similarity, while "chair" and "couch" both belong to the furniture. This means the CNN models have to identify feature differences from multiple semantic hierarchies at the same time.

1.2.2 Imbalanced Training for Localization

The current detection models have an inadequate quality criterion for anchor-based bounding box candidates, which is very important for model optimization and detection evaluation. The reason lies in that the quality of bounding boxes should reflect both the spatial location accuracy and the classification probability. While as far as we know, in current deep learning-based object detection pipelines, the scores of the bounding boxes are shared with box-level classification confidence, which is predicated on the proposed features by the classifier. If a predicted object is not scored properly, it might be mistaken as a false positive or negative. It is evident that the lack of effective scoring metrics for the localization quality tends to impair the evaluation.

1.2.3 3D Attacks

Current popular PointNet[54]-based models are proven to be extremely vulnerable under adversarial attacks [72; 73; 74; 75; 76]. These attacks can successfully trick the 3D models without any noticeable changes to the original 3D point cloud as shown in Figure 1.10.

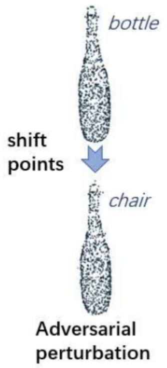


Figure 1.10: A simple sample of adversarial attack on 3D point cloud

1.2.4 Expected Contributions

We propose novel solutions to the mentioned limitations. The expected contribution is listed below.

- We generate a relatively large endoscopic dataset for polyp detection and classification. We have also evaluated and compared the performance of eight state-of-the-art deep learning-based object detectors, setting benchmark results and facilitating the computer-aided diagnosis of colorectal cancer.

- We propose a super-class guided architecture that consists of a super-class branch (SCB) and a finer class branch (FCB) for better performance in detection and classification. It can be directly applied to most image classification and object detection models.

- We propose a novel bounding box reasoning method for the single-shot networks that is aware of the spatial relationship between the box proposals and the probable target ground truth.

- We propose a family of novel robust structured declarative classifiers for 3D point clouds where the declarative nodes defend the adversarial attacks through implicit gradients. We demonstrate superior performance of our approach by comparing it with the state-of-the-art adversarial defenders under the state-of-the-art adversarial attackers.

Chapter 2

SGNet: A Super-class Guided Network for Image Classification and Object Detection

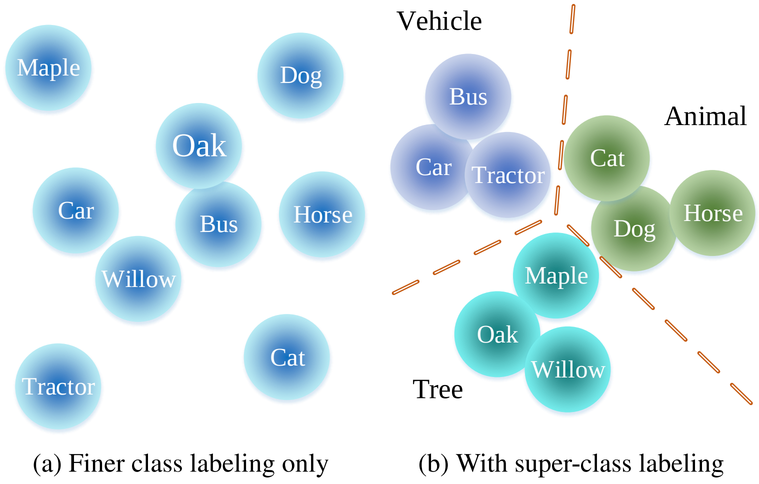
Abstract

Most classification models treat different object classes in parallel and the misclassifications between any two classes are treated equally. In contrast, human beings can exploit high-level information in making a prediction of an unknown object. Inspired by this observation, the paper proposes a super-class guided network (SGNet) to integrate high-level semantic information into the network so as to increase its performance in inference. SGNet takes two-level class annotations that contain both super-class and finer-class labels. The super-classes are higher-level semantic categories that consist of a certain amount of finer classes. A super-class branch (SCB), trained on super-class labels, is introduced to guide finer class prediction. At the inference time, we adopt two different strategies: Two-step inference (TSI) and direct inference (DI). TSI first predicts the super-class and then makes predictions of the corresponding finer class. On the other hand, DI directly generates predictions from the finer class branch (FCB). Extensive experiments have been performed on CIFAR-100 and MS COCO datasets. The experimental results validate the proposed approach and demonstrate its superior performance in image classification and object detection.

2.1 Introduction

Recent years have witnessed the fast development of convolutional neural networks (CNNs) based models in computer vision tasks. Starting from 2012 [2], CNN-based classifiers and detectors have quickly surpassed traditional models [77][78][79]. In just a few years, the classifiers have already emerged to surpass human accuracy in several benchmark datasets [80][81]. The success of CNN models is mainly owing to their ability to extract high-level semantic features from labeled data [82][83]. CNN models have also been successfully applied in many other areas, like object detection [84], depth estimation [23], crowd counting [85], and image translation [86].

What CNN-based models can achieve with simple class annotations is promising, but there are still weaknesses in this method. For the classification task, most models only take class names as input. All classes are treated equally without any correlation information. During training, the misclassification between "chair" and "couch" and that between "TV" and "rabbit" receive the same penalty. While in reality, "TV" and "rabbit" share almost no similarity, while "chair" and "couch" both belong to the furniture. This means the CNN models have to identify feature differences from multiple semantic hierarchies at the same time.



(a) Finer class labeling only (b) With super-class labeling

Figure 2.1: (a) The CNN models have to implicitly extract the semantic clustering of all classes. The distances among different clusterings depend on their semantic relations. (b) By introducing super-class labels, the model can first classify the features into high-level super-classes and then further classify them into finer classes using fine-grained details.

Our method is inspired by how human beings learn to identify an object. When creating categories, humans establish non-parallel, semantic relations between each class. Some classes, like “animals”, are high on the semantic hierarchy and consist of high-level common features. Other classes might have very detailed features but are under a high-level super-class. As human beings, we tend to assign multiple labels to different levels of the semantic hierarchy. For example, when we see steak, we instinctively classify it as “meat”, and upon closer inspection, the terms “beef” or “ribeye” might be assigned to the steak. Even if we are unsure what an object is, we are still able to classify it according to common features from a higher hierarchy. For example, even if we have not seen a dog breed before, we’re still very confident that it is a dog. This example shows how humans can generalize high-level super-class information to robustly recognize sub-class objects.

In this paper, we propose a classification method that is similar to how humans identify an object. It first creates a set of super-classes based on original classes to establish a 2-level hierarchy. The model can simultaneously predict both super-classes and finer classes from separate branches. During training, a misclassified super-class will result in a loss for both branches, thus its value will be greater than a mere finer class error. The two branches share a majority of the backbone networks but still have their own individual portions. In this architecture, the super-class portion focuses on general features shared by super-class categories, while the finer class portion can handle detailed fine-grain attributes.

In this way, we are able to utilize the knowledge base humans have obtained to help the model learn the semantic clustering of classes. The current classification training only provides the network with parallel labels without explicit semantic information. As illustrated in Figure 2.1, the difficulty of extracting the semantic clustering can be alleviated by “telling” the model the 2-level hierarchy labels. Our method is also promising to integrate human semantic knowledge into deep learning models. In this paper, we propose a super-class guided network (SGNet) and verify its effectiveness on both image classification and object detection tasks. Extensive experiments show that our proposed approach can consistently boost the performance of existing models with only a little overhead. The proposed method can be adapted to any existing image classification or object detection networks.

The main contributions of this paper include:

1. Inspired by the human cognition system, we propose a super-class guided architecture that consists of a super-class branch (SCB) and a finer-class branch (FCB). The features from SCB with high-level information are fed to FCB to guide finer class predictions.
2. By introducing high-level information and grouping existing finer classes into super-classes, current labels can be easily modified to train both SCB and FCB for better performance in detection and classification.
3. The proposed SGNet can be directly applied to most image classification and object detection models. Extensive experiments demonstrate its superior performance over existing models.

The source code of the proposed network can be accessed at `https://github.com/rucv/SGNet`.

2.2 Related Work

2.2.1 Image Captioning and Attributes

The goals of image captioning and detection by attributes can vary to a large extent. However, at certain stages, they both aim to train the models to interpret the images in a descriptive way. Image captioning models [87; 88; 89] combine computer vision and natural language processing techniques. By providing captions during the training phase, we expect the model not only to learn what the objects are, but also to understand the details of and interactions between those objects. Yao *et al.* [90] proposed an architecture to boost caption performance by incorporating attributes.

Attributes have long been used in the computer vision community. Back to around 2010, some works [91] were published to generate attribute detectors. Later on, a popular director, zero-shot learning predicts object instances whose classes might not be in the training dataset. These detectors [92; 93] often employ the attribute detector as the first stage and then predict the class categories by finding the category with the most similar attribute set.

Both of these two areas try to integrate human knowledge into deep learning models to obtain robust results. However, they have a common drawback: both methods require a specific set of annotations. For each object, they need at least several words to form the caption or attributes, which limits the amount of data accessible, unlike an object detection ground truth, which consists of only a bounding box and a class label. Moreover, captions and attributes are more subjective, which makes the annotations error-prone, which further limits the applications.

2.2.2 Hierarchical Class Labeling

Read *et al.* [94] uses conditional dependency information from classifiers' error vectors to generate optimal super-class partitioning. Then, a common multi-dimensional ensemble method is used to predict the final result. Zhou *et al.* [95] applied the super-class idea on a training dataset with unbalanced class distribution. They first partitioned the classes into super-classes to create a relatively balanced distributed dataset, which helps minority classes benefit from abundant samples under the same super-class. A weight matrix is applied to put attention on features important to a specific super-class for final prediction.

These works generate super-classes using statistic distribution from one dataset, which makes them dataset dependent. In addition, to calculate the statistic distribution, complex algorithms will be involved to complete the training process. Roy *et al.* [96] created an incremental CNN learning model with a tree structure. This model can evolve after the base model has completed training. When a new class sample is provided, the model can assign the new class to available nodes or even create new branches for the new class depending on the output. Therefore this method can add new classes without abandoning the existing classification ability.

Other methods [97; 98] directly use human knowledge to create the super-class clustering. Wang *et al.* [97] proposed a CNN architecture with separate branches for super-classes and finer classes with some shared layers. It predicts super-class results if the finer class branch is not confident and finer class results when the confidence score from the finer class is above a certain threshold. Yan *et al.* [98] also created a similar architecture. However, the model has an individual branch for each super-class and predicts super-class results first. Once the super-class result is out, the model only uses the corresponding branch to predict the finer class.

Different from our methods, these models don't take advantage of the high-level features from super-classes in finer class prediction. In our proposed approach, the high-level features are concatenated to the finer branch features. Therefore the finer class branch can focus on extracting fine-grained features. Also in [98], each super-class has its own CNN branch, which will result in a larger model size.

2.3 The Proposed Method

We propose a CNN architecture that can be applied to both classification and detection tasks. By adding a super-class branch (SCB), it is designed in a way that can be directly plugged into any existing model. The new branch is trained together with the original classification to guide the finer class training while learning the super-class information. The overall loss is calculated by summing the losses from two branches and backpropagating at the same time.

During inference, we experiment two different setups

1. **Two-step inference** (TSI) first predicts super-class, then finds the highest confidence score in the corresponding finer classes.
2. **Direct inference** (DI) directly generates classification results from the finer class branch.

TSI takes advantage of the higher accuracy in super-class prediction. Therefore, even if the final finer class prediction is not correct, it tends to avoid serious mistakes. The DI setup skips the computation in SCB which optimizes the inference time. In addition, it achieves better performance compared to the baseline, because during training, the SCB guides the model to extract the relations among different classes. The details will be discussed in Section 2.4.

The key contribution of our proposed methods is the addition of the SCB, which shares most layers with the original CNN architecture and uses the common cross entropy loss in training. The

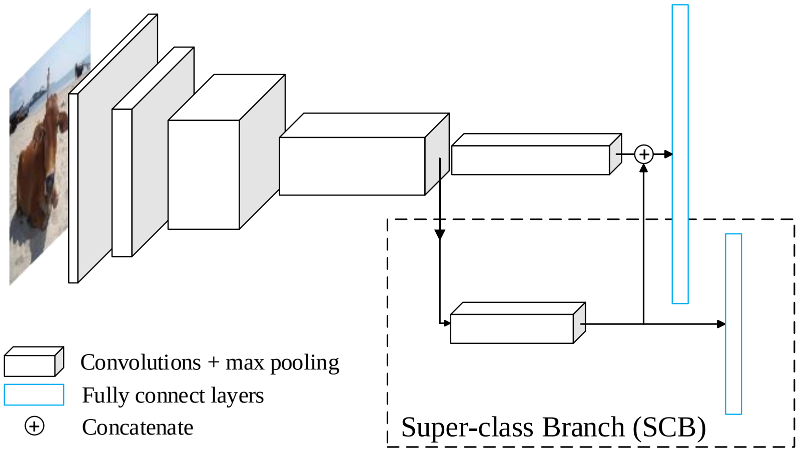


Figure 2.2: **The architecture of SGNet.** SCB is in the dashed rectangle. It takes high-level features from the original networks as input and predicts super-class results. The features from SCB are then concatenated back to the original network to guide finer class prediction. SCB usually has fewer layers than its corresponding finer class branch.

detailed architecture will be discussed next.

2.3.1 Super-class Branch for Classification

A vast majority of recent CNN-based image classification models have a common architecture as shown in Figure 2.2 without the SCB in the dashed box. They are a cascade of convolution layers to extract semantic information with downsampling layers inserted at certain levels, followed by several fully connected layers to predict image classes at the end. Most downsampling is done by max pooling, *e.g.* VGG [1] and ResNet [3]. Some recent models achieve the same effect by applying convolutional layers with stride > 1, like SqueezeNet [99] and MobileNets [100]. Our method can be adapted to both of these architectures.

Our idea can be generalized to any classification model. The basic structure is shown in Figure 2.2. The added SCB takes one hidden layer of the main network as input. After going through a separate cascade of convolutional layers, downsampling layers, and fully connected layers, it predicts super-class confidence scores. The SCB's final feature maps, which are the input of SCB's fully connected layers, are concatenated back to the final feature maps in the original branch to generate finer class scores.

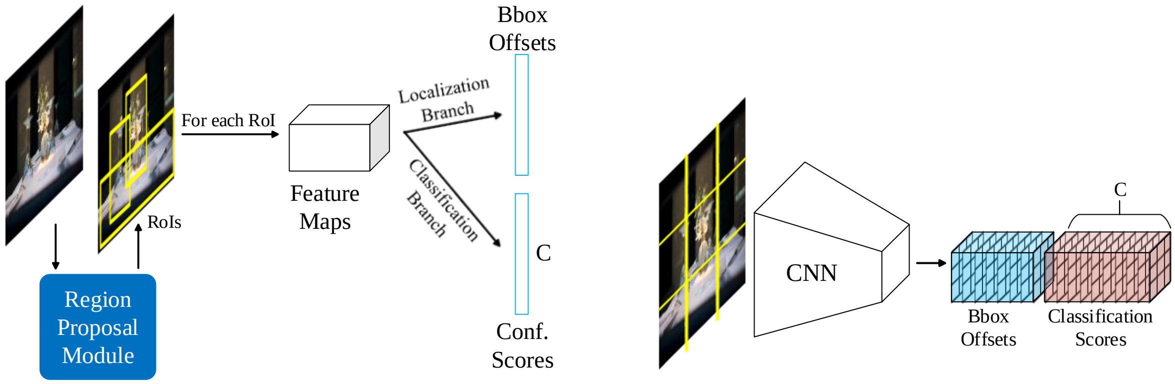
The input of SCB is taken after one of the last several downsampling layers in the main network, which contains high-level information since it is already towards the end of the architecture. The exact downsampling layer depends on the depth of the main network. For example, in VGG-16, we select the second last pooling layer to start the SCB.

The number of convolutional layers in the SCB is designed to be shallower than the rest of the original, and there should be an equal number of downsampling layers. Detecting super-class is an easier task compared to finer classes. The feature differences among super-classes are much greater than those in finer classes. A shallower network can achieve faster performance, and it will be sufficient to guide the finer classification. The downsampling layers should match since the feature maps from both branches will be concatenated. Therefore, the feature maps have to share the same dimensions.

The general architecture in the SCB is similar to a common CNN classification network. It consists of several convolutional layers followed by fully connected layers to make super-class predictions. The input of the fully connected layers, which are the final feature maps generated by SCB, will be concatenated to the feature maps of the finer class branch. The SCB feature maps contain information about common features within the same superclass. It can help the original branch only to focus on finer details to distinguish between categories within the same super-class.

2.3.2 Super-class Branch for Detection

Generally speaking, there are two types of detectors, two-stage detectors and one-stage detectors. They have different architectures. Two-stage detectors, like the RCNN family [40; 101], first propose region of interest (RoI). Then, they extract feature maps for the RoIs, from which bounding-box offsets and confident scores are calculated. Unlike two-stage detectors, one-stage detectors don't have a region proposal stage. Instead, most of them, like SSD-based detectors [49; 51], have predefined shapes (commonly referred to as anchors) evenly distributed in images using grid cells. The detectors directly generate offsets and confidence scores for the anchors. The architectures are



(a) Two-stage detectors

(b) One-stage detectors

Figure 2.3: **The architecture difference for different types of detectors** C in both (a) and (b) is the number of classes.

shown in Figure 2.3.

Although we only focus on the classification branch, there are still lots of differences among the structures. For example, Faster RCNN [38] uses fully connected layers to produce confidence scores from a single feature layer, while SSD [51] applies convolutional layers to multi-layers from the backbone network. To generalize our approach, we designed a method that will work for all existing detectors. As shown in Figure 2.3, all models output C scores for each area of interest (RoI for two-stage models and anchor for one-stage models) where C is the number of finer classes C_{FC}. For the super-class guided method, we simply make $C = C_{SC} + C_{FC}$. After the model produces the vector V of length C, it divides the vector into two parts, $V_{SC} = V[0 : C_{SC}]$ and $V_{FC} = V[C_{SC} : C_{SC} + C_{FC}]$, which are responsible for super-class and finer class predictions respectively.

2.3.3 Super-class Branch During Training and Inference

Training: During training, the setup is straightforward. The classification ground truth for super-classes is converted from their original finer class annotations. The loss for super-class prediction and finer-class prediction is calculated individually using cross-entropy loss. The new classification loss is defined by:

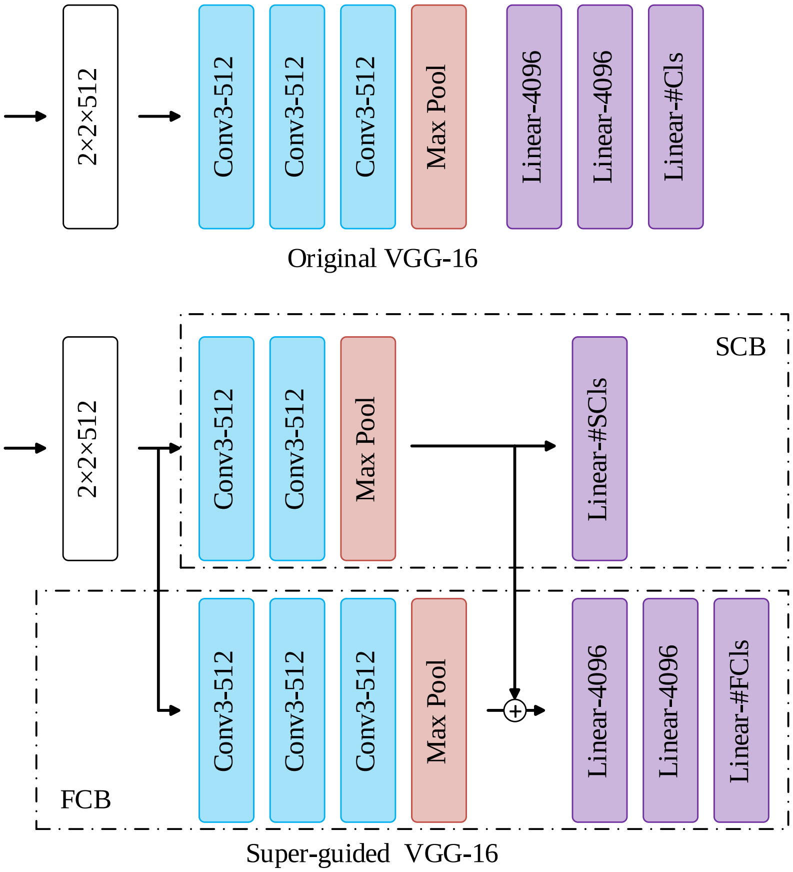


Figure 2.4: **Partial architecture of super-class guided model in VGG-16.** The modification is after the 4th max pooling layer (10th layer) of the original VGG-16 network. The feature map at this stage is a $2 \times 2 \times 32$ tensor. **Upper**: The original model has 3 convolutional layers and one max pooling layer, followed by 3 fully connected layers. **Lower**: Only 2 convolutional layers and 1 fully connected layer are used in super-class branch (SCB) for easier super-class prediction. The finer class branch (FCB) has the same depth as the original architecture. The input of FCB's fully connected layers is the concatenation of the feature maps from SCB and FCB.

Super Class	Finer Class				
aquatic mammals	beaver	dolphin	otter	seal	whale
fish	aquarium fish	flatfish	ray	shark	trout
flowers	orchids	poppies	roses	sunflowers	tulips
food containers	bottles	bowls	cans	cups	plates
fruit and vegetables	apples	mushrooms	oranges	pears	sweet peppers
household electrical devices	clock	computer keyboard	lamp	telephone	television
household furniture	bed	chair	couch	table	wardrobe
insects	bee	beetle	butterfly	caterpillar	cockroach
large carnivores	bear	leopard	lion	tiger	wolf
large man-made outdoor things	bridge	castle	house	road	skyscraper
large natural outdoor scenes	cloud	forest	mountain	plain	sea
large omnivores and herbivores	camel	cattle	chimpanzee	elephant	kangaroo
medium-sized mammals	fox	porcupine	possum	raccoon	skunk
non-insect invertebrates	crab	lobster	snail	spider	worm
people	baby	boy	girl	man	woman
reptiles	crocodile	dinosaur	lizard	snake	turtle
small mammals	hamster	mouse	rabbit	shrew	squirrel
trees	maple	oak	palm	pine	willow
vehicles 1	bicycle	bus	motorcycle	pickup truck	train
vehicles 2	lawn-mower	rocket	streetcar	tank	tractor

Table 2.1: Super-class and finer class groups for CIFAR-100

$$Loss = (1-\alpha)Loss_{FC} + \alpha Loss_{SC} \tag{2.1}$$

where α ($\in (0,1)$) determines which loss to focus on. For example, a greater α trains super-class branches faster. The features generated from backbone networks are biased more towards super-classes.

Inference: At the inference stage, the basic architecture is the same as training. The SCB first generates the super-class confidence scores. Then, the original branch produces the finer class confidence scores. To obtain the final prediction, we experimented with two setups.

1. **Two-step inference** (TSI) predicts super-class by finding the maximum confident score from SCB. Then, it takes the corresponding finer class scores into consideration. We notice there are sometimes conflicts between the super-class and finer-class predictions if we analyze the two branches individually. This setup uses the more accurate super-class results to guide

finer class tasks. Thus, the finer class branch only needs to choose between categories within one super-class. The final confidence score is calculated by applying softmax layer to only the corresponding finer class outputs from fully connected layers. Using this setup, even when the final finer class predictions are incorrect, we still likely have the correct super-class results, thus resulting in less serious consequences.

2. **Direct inference** (DI) directly generates classification results from the finer class branch without even computing the super-class prediction. The features generated by SCB are concatenated to the original features to produce the final confidence scores. Any computation starting from fully connected layers in SCB is discarded. This setup saves the time that is otherwise needed to go through the fully connected layers in SCB and the algorithm to find finer classes corresponding to the super-class prediction. During training, the finer class branch learns across all finer classes, not limited to just the classes within one super-class. Therefore, the original finer class branch has already learned how to work with SCB features to yield the correct class prediction on its own.

2.4 Experiments

We conduct our experiments for two tasks, image classification and object detection. We use VGG-16 [1] as the base network for both tasks. It is chosen because it is a simple and widely used network. For α value in Equation 2.1, we choose 0.5 so that SCB and FCB have equal contributions to the classification loss. For image classification, results are tested on CIFAR-100 [102], which consists of 50k training images. Object detection task uses MS COCO [5]. The detailed information will be presented in the corresponding subsections.

2.4.1 Classification Dataset

CIFAR-100 is a dataset with tiny images of 32×32 resolution. It has a total of 60k images, with 50k in the training set. They are evenly distributed among 100 finer classes, with 600 images each.

This dataset is chosen because it also provides its own 20 superclasses, as shown in Table 2.1. Each superclass has 5 finer classes. The evenly distributed pattern eliminates other factors and helps us to analyze our modified architecture. To evaluate the performance, we simply use the accuracy calculated from all test images.

The idea of super-class is to create a hierarchy that classes at each level should have the same semantic level. From Table 2.1, we can see the super-class categories in CIFAR-100 are not optimal. For example, *trees* and *small mammals* are obviously not at the same level. In fact, *trees* is a level higher in ImageNet [103]. Further performance improvement is expected under well-designed hierarchy. But in this paper, we just implement the basic version to evaluate the effect.

2.4.2 Classification Model Architecture

In our image classification task, we choose VGG-16 [1] as the starting point. The comparison between the original architecture and our super-class guided architecture is shown in Figure 2.4. Most of the architecture is kept the same. Our modification starts after the 4th max pooling layer. The super-class branch is shallower than the original network, while the finer-class branch (FCB) has the same depth as the original network. The difference in the FCB is the input of the fully connected layers. The concatenation results in a 1024-channel feature map, while the original network only has 512 channels. We design a 1-level fully connected layer in SCB since predicting super-class is comparatively easier. In the next subsection, we will show that this shallower network has satisfactory results.

The learning rate is set to be 0.1 initially for all training. We decay the learning rate at epochs $60, 120$, and 160 with a rate of 0.2. We also adopted a training strategy, *warmup* [3; 104]. Using a less aggressive learning rate at the start prevents the model from getting unstable at the beginning stage of training. In our experiments, we set the first epoch as the warmup stage, and used a batch size of 128.

Model	Accuracy (%)	Epoch	Inference Time	# Params
VGG-16	72.15	197	2.06ms	34.0M
SG with TSI	72.78	195	2.78ms	40.8M
SG with DI	**72.84**	180	2.42ms	40.8M

Table 2.2: Results on CIFAR-100

Mismatch	Correct SC	Correct FC	Correct Combined
537	189	132	113

Table 2.3: Error analysis on SGNet model at epoch 180

2.4.3 Classification Results

We report the results from the original VGG-16, SGNet with two-stage inference (TSI) and direct inference (DI) in Table 2.2. All the experiments are conducted on an NVIDIA TITAN Xp GPU. The performance is compared using top-1 accuracy. Other metrics are also listed to thoroughly evaluate our proposed SGNet.

From Table 2.2, we can see that SGNet with DI achieves the best performance, improving the original VGG-16 by 0.69%. SGNet with TSI also performs better than the original network, closely trailing behind DI by 0.06%. In fact, from all test results at the end of each epoch, DI consistently outperforms TSI. To analyze why DI performs better, we take the model from epoch 180. As introduced in Section 2.3, both inference strategies share the same model with the only difference being the inference algorithm.

In Table 2.3, we show the error analysis. The SCB and FCB in the model generate super-class prediction and finer-class prediction respectively. The mismatch in Table 2.3 is caused by the predicted finer class not being under the predicted super-class. In all $10,000$ testing images, 537 samples of such mismatch are found. Of the 537 mismatched predictions, FCB is proven to be correct in 132 of those images, but SCB has a better result with 189 correct predictions. In TSI, only 113 images are predicted correctly after combining both SCB and FCB results, which is less than DI. Therefore, if the finer class accuracy is the only thing that matters, then DI performs better. Note that it has more super-class prediction errors. TSI actually makes less serious mistakes in prediction. So it really comes down to the application that determines which inference is better. If the consequence of a super-class error is severe, then TSI is actually preferred.

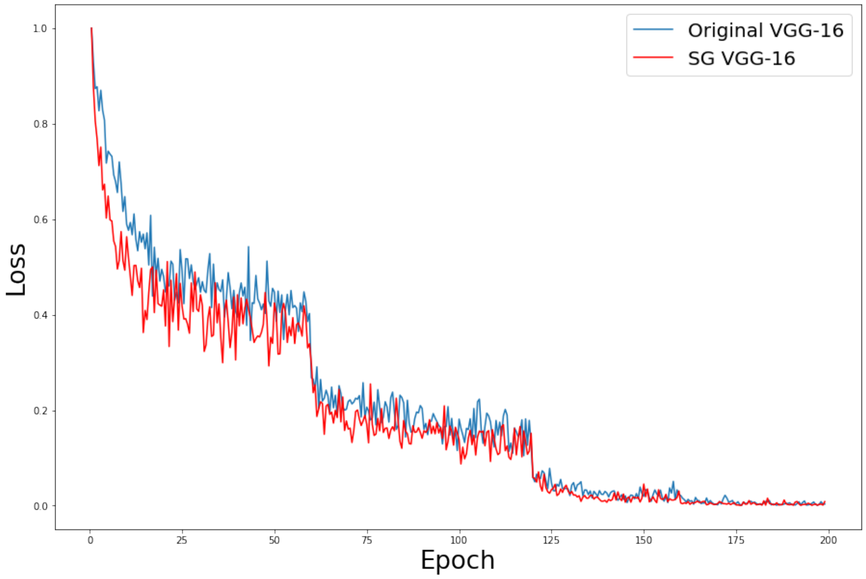


Figure 2.5: Training loss on CIFAR-100

In Table 2.2, we also show the number of parameters in the networks we experimented with. The SGNet slightly increases the model size from 34.0 million parameters to 40.8 million. Correspondingly, the inference time is also increased a little bit. From the test result, we can see that SGNet achieves better performance than the original network. The loss curve is shown in Figure 2.5. The loss values are normalized to $[0, 1]$ for the convenience of comparison. Figure 2.5 indicates that our model converges faster than the original VGG-16, especially at the early stage of training.

2.4.4 Detection Datasets

MS COCO [5] is a large-scale image dataset that can be used for object detection, object segmentation, etc. For the object detection task, it has 80 object categories. It can be divided into 12 superclasses, as shown in Table 2.4. We can see that each super-class has a different number of finer classes. We use MS COCO's `2014 train + 2014 val - 2014 minival` as the training set and `2014 minival` as the validation set. The performance is evaluated using MS COCO's metrics APs at different IoUs (from 0.50 to 0.95) and APs for different object sizes.

Super Class	Finer Class
person	person
vehicle	bicycle, car, motorcycle, airplane, bus, train, truck, boat
outdoor	traffic light, fire hydrant, stop sign, parking meter, bench
animal	bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
accessory	backpack, umbrella, handbag, tie, suitcase
sports	frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket
kitchen	bottle, wine glass, cup, fork, knife, spoon, bowl
food	banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
furniture	chair, couch, potted plant, bed, dining table, toilet
electronic	tv, laptop, mouse, remote, keyboard, cell phone
appliance	microwave, oven, toaster, sink, refrigerator
indoor	book, clock, vase, scissors, teddy bear, hair drier, toothbrush

Table 2.4: Super-class and finer class groups for MS COCO

2.4.5 Detection Model Architecture

We implement our network based on Faster RCNN [38]. For the backbone network, we chose VGG-16 [1]. We increased the output channel of the classification branch in Faster RCNN. As a two-step detector, Faster RCNN generates C scores for each RoI as shown in Figure 2.3a, where C is the number of classes. Therefore, in the original model, C is 81, background class plus 80 classes. In our method, we extend C to 94 ($= 13 + 81$), where 13 is one *background* class plus 12 super-classes. Then, we split the final 94-element array into super-class and finer-class portions.

In this detection task, we adopt a simpler learning rate strategy. It starts with 10^{-2} and decays every 5 epoches with the decay rate of 0.1. When reading the image, we use a scale of 800 and the maximum size is 1200 pixels.

2.4.6 Detection Results

The detection results on MS COCO are reported in Table 2.5. SGNet achieves better overall AP by 0.6% compared to the original VGG-16. It is evident that our model has greater improvements when the IoU threshold is smaller. At an IoU of 0.50, our model improves the AP by 1.1%. In addition, SGNet has the best effect when detecting large objects. This is caused by the components of the final loss in the training phase. In SGNet, the loss is the sum of L_{sc}, L_{fc}, and $\mathrm{L}_{\mathrm{bbox}}$. Two terms in the loss are focused on training classification tasks. Therefore, the weight on the bounding

Model	VGG-16	SGNet
Detect Time	43.0ms	43.4ms
# Params	138.06M	138.11M
AP	28.6	29.2
AP_{50}	48.6	49.7
AP_{75}	30.3	30.8
AP_S	12.8	12.5
AP_M	32.5	33.2
AP_L	40.3	41.2

Table 2.5: Results on MS COCO. VGG-16 and SGNet achieve best performance at epoch 15 and epoch 25 respectively. All experiments are carried out on one NVIDIA TITAN Xp GPU.

box training is subsequently reduced. As shown in Table 2.5, the size and the inference time of the proposed detection model are almost the same as the original VGG-16 model, while the average precision is increased by 0.6%.

2.5 Conclusion

In this paper, we have proposed a super-class guided network to integrate high-level knowledge into current image classification and object detection models. The proposed model integrates feature maps from both SCB and FCB to guide the finer class prediction. By creating an SCB model, we are able to add high-level knowledge hierarchy in the form of two-level class annotations. Extensive experiments have been performed on CIFAR-100 and MS COCO datasets. The results demonstrate the proposed SGNet can improve the performance of both image classification and object detection. The proposed network can be easily plugged into most existing models.

Chapter 3

Location-Aware Box Reasoning for Anchor-Based Single-Shot Object Detection

Abstract

In the majority of object detection frameworks, the confidence of instance classification is used as the quality criterion of predicted bounding boxes, like the confidence-based ranking in non-maximum suppression (NMS). However, the quality of bounding boxes, indicating the spatial relations, is not only correlated with the classification scores. Compared with the region proposal network (RPN) based detectors, single-shot object detectors suffer the box quality as there is a lack of pre-selection of box proposals. In this paper, we aim at single-shot object detectors and propose a location-aware anchor-based reasoning (LAAR) for the bounding boxes. LAAR takes both the location and classification confidences into consideration for the quality evaluation of bounding boxes. We introduce a novel network block to learn the relative location between the anchors and the ground truths, denoted as a localization score, which acts as a location reference during the inference stage. The proposed localization score leads to an independent regression branch and calibrates the bounding box quality by scoring the predicted localization score so that the best-qualified bounding boxes can be picked up in NMS. Experiments on MS COCO and PASCAL VOC benchmarks demonstrate that the proposed location-aware framework enhances the performances of current anchor-based single-shot object detection frameworks and yields consistent and robust detection results.

3.1 Introduction

Deep networks have been dramatically driving the progress of computer vision, bringing out a series of popular models for different vision tasks [105][106], like image classification [77][107], object detection [108][109], crowd counting [85], depth estimation [23], and image translation [110]. Object detection plays an important role and serves as a prerequisite for numerous computer vision applications, such as instance segmentation, face recognition, autonomous driving, and video analysis [111; 26; 112; 113; 114]. In recent years, the performance of object detectors has been dramatically improved due to the advancement of deep network structure, well-annotated datasets, and effective optimization algorithms [50; 115].

In this paper, we aim at single-shot object detectors that yield a better trade-off between accuracy and speed, indicating a trend for future frameworks [49; 82]. We reveal the problem of an inadequate quality criterion for anchor-based bounding box candidates, which is very important for model optimization and detection evaluation. The reason lies in that the quality of bounding boxes should reflect both the spatial location accuracy and the classification probability. While as far as we know, in current deep learning-based object detection pipelines, the scores of the bounding boxes are shared with box-level classification confidence, which is predicted on the proposed features by the classifier. Most importantly, we cannot obtain the location assessment during the inference stage due to the deficiency of labels. It is insufficient to use the classification confidence to measure the bounding box quality since it only serves for distinguishing the semantic categories of proposals, while it is not aware of the assessment towards localization accuracy. The misalignment between classification confidence and bounding box quality is illustrated in Figure 3.1, from which we can see that, although the object instances obtain a high classification confidence score, the box-level localization is not unanimously accurate. If a predicted object is not scored properly, it might be mistaken as a false positive or negative, affecting the NMS process and leading to a decrease in average precision (AP). It is evident that the lack of effective scoring metrics towards the localization quality tends to impair the evaluation.

In this work, we focus on a more reasonable and effective scoring metric for anchor-based

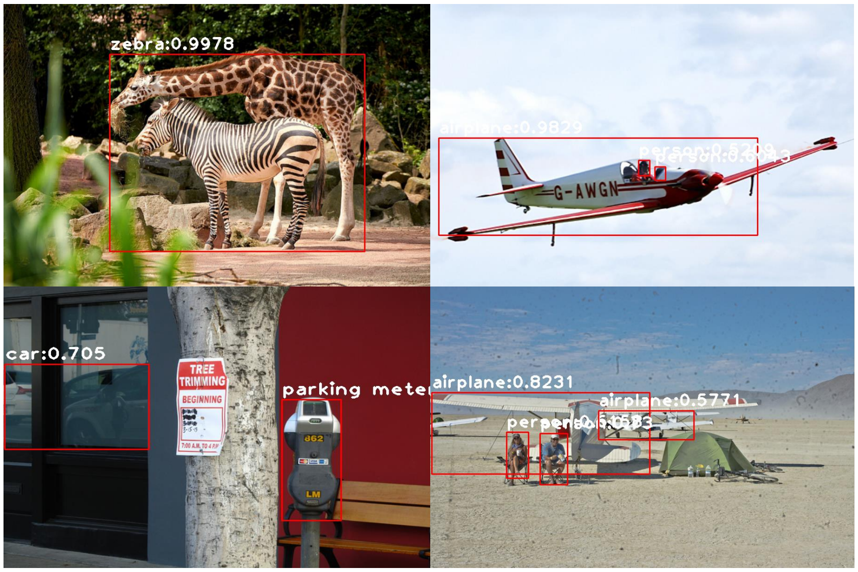


Figure 3.1: Demonstrative detection results on MS COCO [5]. The predicted bounding boxes have high classification scores while the localization is misplaced or interceptive. The left two images show misplaced cases in which the zebra is located by a much larger region, and the car is not an actual object. The airplanes in the right two images are partially located, while all of them have high confidence scores. Our method predicts the spatial relation between box proposals and their possible targets so that interception and misplacement can be minimized.

bounding box proposals. Different from most previous works that either pursue high-quality classification boxes or focus on score correction working on two-stage object detectors, we demonstrate that there is room for further improvement for popular anchor-based single-shot object detection models by introducing calibrated quality scores that take the location confidence into consideration. Compared with RPN-based frameworks, single-shot object detectors highly depend on qualified box proposals as there is no pre-sift scheme. As anchor-based methods, they are sensitive to location information, which brings challenges for box sifting. To solve this issue, we propose a calibrated quality score (**CQS**) for each box proposal to realize the location awareness. The localization score indicates the spatial relation to its most probable target ground truth and ranks the proposals based on the calibrated quality score rather than the classification confidence.

In anchor-based single-shot object detection, the bounding box proposals are regressed by the space shift relative to the anchors. Thus, the spatial relationship of an anchor and an object ground truth depicts an expectation or estimation of the location relationship between the corresponding box proposal and the target, as depicted in Figure 3.2. Inspired by the Average Precision (AP) metric of object detection using pixel-level Intersection-over-Union (IoU) between the predicted bounding box and its ground truth to describe the quality of predictions, we propose a network module to learn the IoU between the anchors and the ground truth directly. For the convenience of discussion, we call it **AIoU**. We adopt the proposed locscore to learn this AIoU during the training time, and when given the locscore in the test phase, the quality of bounding boxes is reevaluated by integrating locscore into the classification confidence so that the reasoned box proposals are aware of both the location information and the semantic categories.

Compared with localization and classification regressions that take the ground truths from the labeled dataset, the learning for AIoU only needs to calculate the IoU between the anchors and the ground truths as a target, without further labeling the dataset. Within a detection model, we implement the locscore prediction network as the locscore head, which takes the feature outputs and the calculated AIoU as inputs, and is trained with a common regression loss. We implement object detection experiments with the proposed location-aware anchor-based box reasoning module on popular single-shot object detectors. The results demonstrate that our method can promote the performance of object detection and yield consistent and robust detection results. The main contributions of our paper include:

C1. We propose a novel bounding box reasoning method that is aware of the spatial relationship between the box proposals and the probable target ground truth. It is one of the first algorithms that address the issue caused by scoring bounding box proposals only by the classification probability.

C2. This is the first location-aware detection framework designed for the single-shot networks that naturally take the pools of anchor-based box proposals as candidates, ensuring a one-shot learning fashion.

C3. The proposed plug-in locscore head can be integrated with any single-shot detection networks and regressed easily in an end-to-end fashion. By calibrating the detection quality with locscore, the bounding boxes can be penalized if it has high classification confidence while rela-

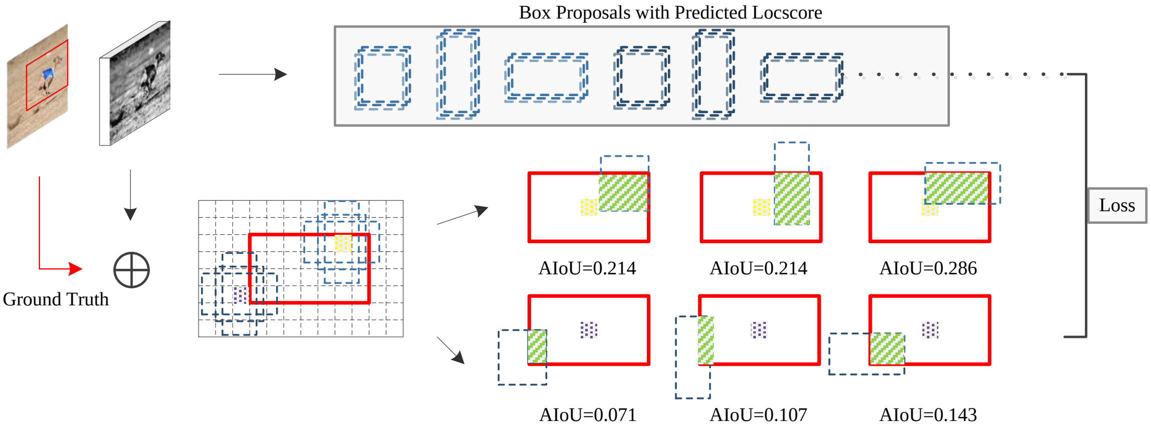


Figure 3.2: AIoU Definition and the Locscore illustration. AIoU as the target of the proposed locscore only needs the input image and its corresponding ground truth. In the fashion of convolution, we evaluate the default boxes with multiple scales at different anchor bases, as the yellow and purple color lumps shown in this figure. For each default box (represented by the blue dotted rectangular), in addition to predicting the shape offsets and the confidence scores as conventional detectors, we also predict the locscore which assesses the possibility of how close the object is to the ground truth. The locscore is learned towards the AIoU calculated by the anchor box and the ground truth, which is denoted by the red angular box in this figure. Specifically, the locscore of a box proposal is learned to match the AIoU between its corresponding anchor box and a certain ground truth box.

tively poor localization accuracy.

C4. We demonstrate the effectiveness of the location-aware anchor-based box reasoning scheme through extensive experiments. By introducing the proposed calibrated quality score into the evaluation metric of box proposals, the detection performance is further improved.

3.2 Related Work

3.2.1 Object Detection

Multiclass object detection is a core task in the context of deep learning-based computer vision projects, which is the joint work of the classification towards contents and the localization towards bounding boxes of instances. Most of these methods adopt the CNN [116] based bounding box and classification regressions, followed by a Non-Max Suppression (NMS) algorithm to sift best-qualified box proposals.

Bounding box regression was first introduced in R-NN [39]. It enables regions of interest (ROI) to estimate the updated bounding boxes with the purpose of better matching the nearest object instance. Prior works, from Fast R-CNN, Faster R-CNN [38], R-FCN [117], to YOLO [42], SSD [49], RetinaNet [50], and RefineDet [51], have demonstrated that the detection task can be improved with multiple bounding box regression stages [118], flexible anchor matching [115], the increase of the number of anchors, and the enlargement of the input image resolution, including image pyramids [47]. Among them, the most widely-used and efficient technique is the anchor-based multibox algorithm that can handle scale variation, one of the challenging problems for one-shot object detection. Anchor boxes are designed for discretizing the continuous space of all possible instance boxes into a finite number of boxes with predefined locations, scales, and aspect ratios [119]. The created instance boxes are regressed to match the ground truth bounding boxes based on the Intersection-over-Union (IoU) overlap, by location shift at a certain base anchor with the predefined locations.

However, there exist underlying limitations. On one hand, the quality of the proposed bounding boxes is only measured by the classification score, leading to the misalignment between the box score and box quality. Due to the unreliability of the box score, a proposal with higher IoU against the ground truth will be ranked with low priority if it obtains lower classification confidence. In this situation, the Average Precision (AP) can be degraded. On the other hand, compared with RPN [38], the anchor-based technique is more sensitive to box quality especially for the consistency of classification confidence and location accuracy since there is no pre-sift scheme for box proposals in the one-stage case.

3.2.2 Detection Scoring and Correction

The misalignment of the box score and actual quality has aroused much attention and several correction methods have been proposed in recent years. Tychsen-smith *et al.* [120] presented a Fitness NMS that corrects the detection score by learning the statistics of best matching detected bounding boxes with the ground truth as a corrective factor. It formulates box IoU statistics prediction as the classification task. It is specifically designed for Denet [121], which restrains its application to arbitrary object detection frameworks.

Jiang *et al.* [122] proposed a standalone IoU-Net which is based on a similar R-CNN structure with a proposal pre-sift scheme to predict IoU between the predicted boxes and the ground truths. It manually designs bounding box filtering as an addition to the data pool of box proposals. The IoU-guided NMS ranks bounding boxes by the predicted localization confidence rather than the conventional classification confidence. Cheng *et al.* adopted a separate network to correct the scores of samples by processing false-positive samples [123]. SoftMax [124] proposed to use the overlap between two boxes to correct the low score box. Neumann *et al.* [125] proposed a relaxed softmax to predict the temperature scaling factor in standard softmax for safety-critical pedestrian detection. Both of the two approaches are designed for the two-stage R-FCN based models, relying on the clean proposal data pool. Wu *et al.* proposed the IoU-aware approach scores the location results while it is merely a RetinaNet based detector [126], not an arbitrary method.

Different from the above methods, this study focuses on the essence of the evaluation towards anchor-based box reasoning in single-shot frameworks. We assign each predicted box with a location score by making aware of the spatial relation between its based anchor and the ground truth. The proposed approach takes both the classification confidence and the location accuracy into consideration to create a complete evaluation of instance box quality so as to narrow the gap between the box score and the actual quality. Furthermore, we build an independent regression branch in the single-shot object detection framework that learns the location confidence specifically and merges this information into the box quality evaluation metric of NMS so as to obtain a more reliable priority ranking.

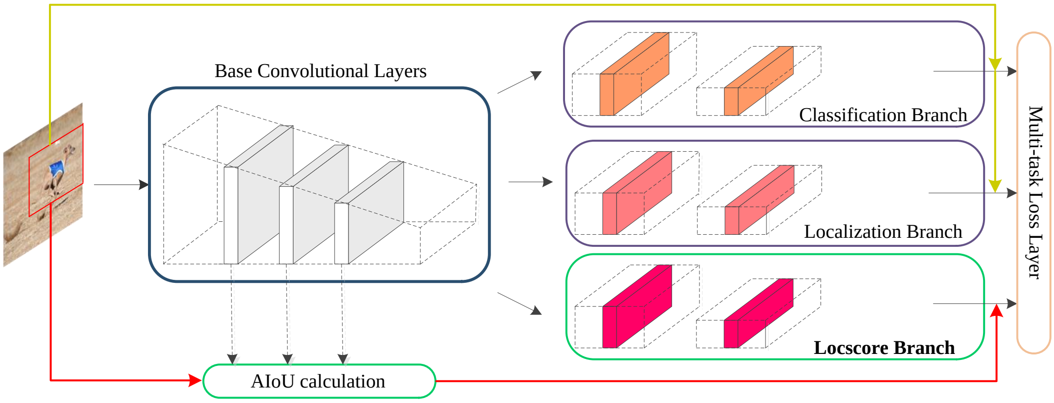


Figure 3.3: The network architecture of object detection with location-aware anchor-based reasoning. The input image is fed into the backbone network to generate feature maps with RoI information. The Locscore Branch is the standard component of the improved model. It takes the output features from the backbone network as inputs and provides a predictive locscore at the end, where its layer structure just follows the classification branch or the localization branch.

3.3 Location-Aware Box Reasoning

3.3.1 Motivation

In current object detection frameworks, the classification and localization regressions are taken as two independent processes. The evaluation towards a detection hypothesis, the detected bounding box, is determined by the highest-ranked element in the classification scores. However, there exist certain situations where the predicted box with a high classification score has low localization accuracy. This kind of hypothesis is harmful in most detection evaluation protocols, such as MS COCO. It is important that a detector can determine when the detection results are trustworthy and when they are not. This motivates us to integrate the localization score by location-aware anchor-based reasoning for every predicted bounding box based on an anchor position.

Most previous methods do not consider location confidence as one of the evaluation factors that contribute to the box quality [120][123] [124][125] and the majority of them are designed as specific detectors or merely applied for two-stage detectors with the pre-sift scheme. Although IoUNet obtains competing results with the proposed IoU-guide NMS algorithm that takes localization confidence into consideration, it ranks the boxes only by the localization confidence which highly relies on the clean data pool of proposals produced by two-stage detection models. It is hard to apply it in a single-shot fashion. We propose the location-aware anchor-based box reasoning that focuses on arbitrary single-shot detectors for a better trade-off between accuracy and efficiency.

We instantiate the location-aware reasoning module by showing how to apply it to the anchor-based single-shot detectors. Without loss of generality, we apply the proposed module to the state-of-the-art RetinaNet and SSD with an additional Locscore head that learns the IoU between the anchor and ground truth, and demonstrates our design from the following aspects: 1) how to realize location awareness for anchors; 2) how to create the branch of location score in the network; and 3) how to generate location-aware anchor-based reasoning during inference time.

3.3.2 Location Awareness

From the perspective of conception, location awareness is simple. In anchor-based detectors, the introduction of location awareness supplements the evaluation towards the quality of the bounding boxes from the perspective of location accuracy. It is realized by learning the IoU between the anchors and ground truths, producing the localization scores.

Localization Score We begin with briefly reviewing the evaluation metric towards the bounding box proposals. Following the anchor-based detection, the proposals are created based on anchors with various scales at different positions on the feature maps. The network extracts features from the backbone and performs proposal classification and bounding box regressions respectively. The former one yields confidence scores regardless of the location reference, while the latter one regresses the space migration of candidates. Although the predicted bounding boxes with classification confidence and location prediction, the quality of the box candidates can only be evaluated by the confidence score, without localization assessment. This is due to the lack of ground truth as a reference during the inference stage. Thus, there is a gap between the current metrics and the actual need for evaluating the quality of box proposals.

We define $P(lc|a_j)$ as the localization score by learning the pixel-level IoU between bounding box a_j and any object ground truth b_j. We define it as "AIoU".

$$AIoU = IoU(a_j, b_j) \tag{3.1}$$

In ideal anchor-based detectors, the object is detected by three elements: anchor, box proposal, and object ground truth. Conventionally, we build the direct correspondence between the anchors and the boxes by regressing space shift, and the relations between the boxes and the object ground truths by matching features. However, there is no direct depiction of the relationship between the anchors and object ground truths, where an underlying location link exists. The introduction of the location score complements the relational structure of the three essential elements and further calibrates the quality criterion of bounding box proposals by the definition expressed below.

$$S(a_j) = P(c|a_j) \cdot P(lc|a_j) \tag{3.2}$$

where $P(c|a_j)$ denotes the confidence score, and $S(a_j)$ is defined as the calibrated quality criterion of the bounding box proposals. Thus, $S(a_j)$ should work well on two tasks: indicating the right category that the box belongs to and regressing the IoU of the proposals and the foreground objects.

3.3.3 Localization Score Regression

Locscore Head Conventionally, classification and regression are two independent branches for all object categories. Without loss of generality, we introduce the Locscore Head as the third independent branch to predict the IoU between anchors and object ground truths. This head simply follows the layer structure of the other two existing heads so as to save the network characteristics and the advantages of the framework. Thus, it can be implemented as a plug-in module and be integrated with any arbitrary single-shot object detection model.

The Locscore Head receives the concatenation of features from the output layer of the network as its inputs. It predicts the localization score for each anchor box on the feature maps, which depicts the location relations between the box and the target ground truth. In anchor-based object detection, each predicted bounding box is created based on a certain anchor and regressed location migration. Thus, according to this correspondence among the three elements, each predicted bounding box proposal corresponds to one anchor so that the box would be assigned a localization score indicating the maximum possibility that it is related to an object from the perspective of localization.

Based on the analysis above, since the Locscore Head shares the same concatenated features with the other two Heads, classification, and box regression, the predicted three elements have an inner congruent relationship. The Locscore Head, thus, can be taken as an independent regression branch and treated as an individual learning task.

We define the Locscore Loss to regress the Localization Score. It follows the loss definition for classification regression. Then the Locscore Head is integrated into an anchor-based object detection framework, and the whole network can be trained end-to-end. Specifically, we define the classification loss and box regression loss as L_{cl} and L_{bb}, respectively. In addition, we introduce locscore loss L_{lc} as another penalty item to the cost function, as shown below,

$$L^* = \lambda_1 * L_{cl} + \lambda_2 * L_{bb} + \lambda_3 * L_{lc} \tag{3.3}$$

where L^* denotes the final loss function. In all experiments, we adopt equal weights for the three loss items in consideration for stability, so that $\lambda_1 = \lambda_2 = \lambda_3 = 1$. The locscore loss forms a lower bound in the space localization, and by training, we further pull down this lower bound.

3.3.4 Box Reasoning during Inference

Calibrated Quality Score (CQS) We define and propose a calibrated quality score by introducing localization confidence into the assessment of the predicted bounding box proposals. Thus, the quality score is disintegrated into two data spaces, where both classification confidence and location accuracy are taken into consideration, as shown in Equation 3.2. This CQS during inference time becomes a new criterion for the sifting of qualified candidates. It complements the defect of independent classification and location regressions that lead to the deficiency of localization reference during inference time.

Inference At the inference stage, the proposed candidates with coordinates, conferences, and localization scores are integrated into the non-maximum suppression (NMS) algorithm. Different from its classical counterparts, the NMS in our model does not rank the box proposals merely by classification probabilities in the first step. Instead, we use the CQS as the ranking criterion so as to push the box candidates to indicate the spatial relations with potential objects, in terms of the initial idea that the quality of bounding boxes is tightly correlated to both the spatial information and classification confidence. We assume the saved boxes after the above sifting are qualified candidates. But in order to weaken the sensitivity towards the less qualified proposals for the single-shot models, we adopt the confidence cluster [122] to further enhance the reliability of the sifted boxes by updating the confidence score S_i of box i with $S_i = max(S_i, S_j)$, where j indicates box j that is deleted by box i in NMS. Details are shown in Algorithm 1.

Specifically, suppose the network outputs N bounding boxes, the NMS firstly ranks them by the proposed CQS, then we follow the same procedure to remove the candidate boxes which overlap each other over a threshold of $\varepsilon = 0.5$. At last, the top-k scored boxes are selected and fed into the output head to generate multi-class boxes.

3.4 Experiments

We conduct experiments on the detection tasks of the MS COCO [5] and PASCAL VOC [127] datasets. MS COCO contains 80 object categories, we follow COCO 2017 settings, using the 115k images *train* split for training, 5k *validation* split for results analysis. The COCO results are reported by its evaluation metrics AP (Average Precision over IoU thresholds), including, AP@0.5 (IoU equals 0.5), AP@0.75 (IoU equals 0.75), AP (averaged on AP over IoU thresholds from 0.5-0.95 with a step size of 0.05), AP$_S$, AP$_M$, AP$_L$ (AP at different scales of objects). In the VOC experiments, we follow the same practice as in the literature, the models are trained on the union

Algorithm 1 Location-Aware Box Reasoning. Classification confidence and localization score are independently regressed during the training time but the two values are taken as combined consideration during the inference time when evaluating the anchor-based box proposals.

Input: B, P_c, P_{lc}, ε.
B: set of anchor-based bounding box proposals.
P_c: classification confidence by mapping f_c.
P_{lc}: localization score by mapping f_{lc}.
ε: IoU threshold in NMS.
Output: D, set of detected boxes with classification confidence P_c.

```
$D \leftarrow \varnothing$
while $B \neq \varnothing$ do
    $S \leftarrow P_c \times P_{lc}$
    $b_m \leftarrow \arg\max_i S(b_i)$
    $S_m \leftarrow S(b_i)$
    $P_m \leftarrow P_c(b_i)$
    $B \leftarrow B \setminus b_m$
    for $b_j \in B$ do
        if IoU($b_m$, $b_j$) > $\varepsilon$ then
            $B \leftarrow B \setminus b_j$
            if classification cluster then
                $P_m \leftarrow max(P_c(b_j), P_m)$.
    $D \leftarrow D \cup \{\langle b_m, P_m \rangle\}$
```

of PASCAL VOC2007 and 2012 trainval set (16,551 images) and tested on PASCAL VOC 2007 test set (4952 images). The overlap threshold for each one of the 20 categories in VOC is set to 0.5.

3.4.1 Implementation Details

We adopt the consistent location-aware box reasoning (LAAR) framework for all experiments. We use the ResNet-50 based FPN network and VGG-16 based SSD network for the ablation study, respectively. For ResNet-50 FPN, the input images are resized with a minimum 608px along the short axis and a maximum of 1024px along the long axis for both training and test. We train the network and choose the model at the epoch that yields the best performance. The learning rate is reduced by the Plateau strategy, the same as that of the original RetinaNet. For SSD, the input images are resized to 300×300 for both training and testing, as the common rule in literature. The rest of all configurations are identical to the realization in [49]. We train the network for 120,000 iterations and decrease the learning rate after 80,000 and 100,000 iterations. The optimizer for RetinaNet experiments is Adam with an initial learning rate of 0.00001, and for SSD experiments is SGD with a momentum of 0.9. In the test, all the results are evaluated by the NMS, where the top-100 score detection is retained for each image.

Learning Scenarios In order to identify the gains of locscore regression constraint and location-aware box reasoning respectively, we intentionally design independent learning scenarios where we do solo locscore regression constraint without quality score calibration, the complete LAAR detection framework with CQS, and the complete LAAR detection with CQS and confidence cluster. We list them as follows:

- *Independent Locscore Constraint (ILC):* We introduce the locscore regression during training time while not considering the predicted locscore to calibrate the classification score during the inference period.

- *Locscore Constraint with CQS (LC):* We conduct complete location-aware box reasoning with the given detector, which means we add locscore regression constraint during training and introduce the calibrated quality score by the predicted locscore during the test time.
- *Locscore Constraint with CQS and classification cluster ($LC+CS$):* Based on LC, we introduce the classification cluster after calibrating the quality score by the predicted locscore, as shown in Algorithm 1.

3.4.2 Ablation Studies

We evaluate the contribution of one important element to our location-aware box reasoning for object detection, the constraint brought by the Localization Score Regression.

Locscore constraint for better optimization Compared with conventional object detection frameworks, we introduce the additional constraint term in the loss function by doing the localization score regression. As far as we know, this is the first time to directly explore the relationship between the anchors and the ground truths in single-shot fashions. The proposed procedure explores the predefined prior information of anchor boxes and the ground truths from the perspective of spatial location. As is known, anchor-based fashion defines fixed anchor positions and their multiple-ratio variations on a feature map, which means for a certain image, there exist predefined location relations between the anchor boxes and the ground truths. We introduce this relation in the penalty function to help constrain the classification and localization regressions. Especially for the latter one during training, when the coordinates are learned in a deflected direction, there exists a correction by the locscore regression. This constraint results in better optimization as demonstrated in Table 3.1 and 3.2.

In Table 3.1, the ILC version leads the original RetinaNet in most cases. For $\mathrm{AP}_{0.75}$, ILC improves the detection accuracy by a rough 1% than the original ResNet. Although the ILC falls behind AP_S, it leads by large margins in both AP_M and AP_L. In Table 3.2, we obtain consistent results. In this VOC setting, we list the AP results for all the 20 categories. The ILC shows better

	R-Net	R-Net+ILC	R-Net+LC	R-Net+LC+CS
Backbone	R-50	R-50	R-50	R-50
AP	30.4 %	30.9 %	30.7 %	**30.9** %
$\mathrm{AP}_{0.5}$	47.3%	**47.7** %	47.2%	47.4 %
$\mathrm{AP}_{0.75}$	32.1 %	33.0 %	32.9%	**33.2** %
AP_S	**13.9**%	13.0 %	12.9 %	13.0 %
AP_M	33.1 %	**34.0** %	33.8%	33.9 %
AP_L	43.7%	44.1 %	44.2%	**44.3** %

Table 3.1: The mAP of RetinaNet on COCO val2017. R-50 indicates ResNet50 with FPN and R-Net refers to RetinaNet.

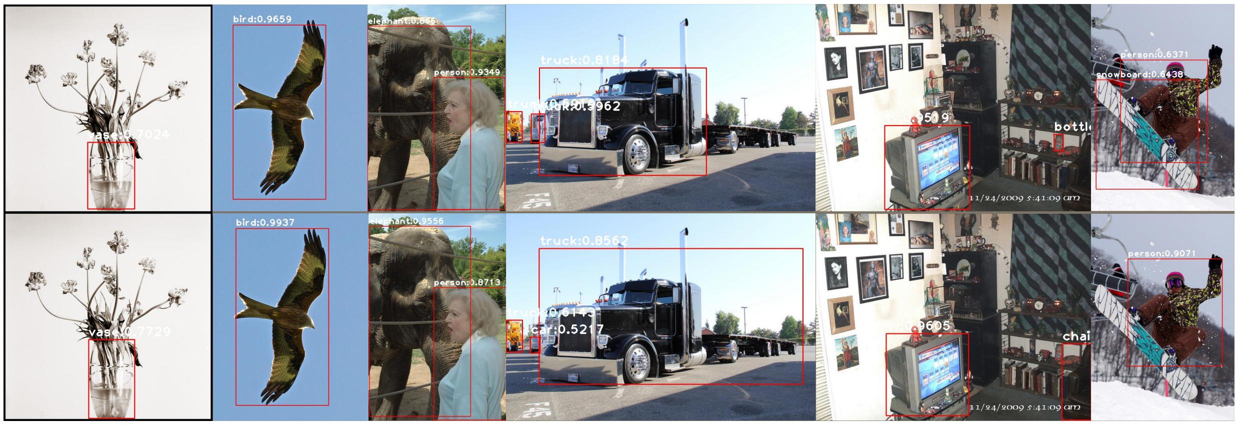


Figure 3.4: COCO cases compared between RetinaNet (1st row) and RetineNet+LC (2nd row), where LC version achieves higher localization accuracy.

mAP than the original SSD and yields clearly higher AP for most categories. These results support the idea that the introduction of locscore regression yields effectively positive constraints towards the other two regressions. Thus, the locscore branch boosts the mutual promotion of classification and localization, helping to improve the situation when the confidence score and localization accuracy are opposite, such as when the confidence score is 0.6 and localization accuracy is 0.2 compared to that when the confidence score is 0.2 and localization accuracy is 0.6, which guarantees the feasibility of the proposed approach.

	SSD	SSD+ILC	SSD+LC	SSD+LC+CS
Backbone	VGG-16	VGG-16	VGG-16	VGG-16
aeroplane	82.97 %	81.87 %	82.43 %	82.41 %
bicycle	84.18%	83.61 %	83.13%	83.18 %
bird	75.34 %	75.23 %	74.23%	75.56 %
boat	70.98%	70.35 %	70.62 %	70.71 %
bottle	50.44 %	51.97 %	51.62%	51.90 %
bus	84.29%	86.05 %	84.49%	86.03 %
car	86.32%	85.30 %	84.60%	85.53 %
cat	88.12%	88.36 %	87.45%	88.26 %
chair	61.54%	62.18 %	58.82%	62.02 %
cow	79.94%	83.03 %	82.28%	83.18 %
diningtable	77.12%	75.80 %	72.40%	76.20 %
dog	85.05%	84.31 %	82.38%	84.13 %
horse	87.60%	87.03 %	85.92%	87.68 %
motorbike	82.84%	82.95 %	81.83%	83.50 %
person	78.98%	79.09 %	77.71%	79.12 %
pottedplant	52.17%	51.94 %	50.27%	51.45 %
sheep	78.61%	77.40 %	75.40%	77.05 %
sofa	78.33%	80.29 %	77.49%	80.12%
train	87.88%	86.65 %	84.80%	87.71%
tvmonitor	76.33%	77.29 %	74.93%	77.10%
mAP	77.45%	77.53 %	76.14%	**77.64%**

Table 3.2: mAP of SSD on VOC2017. The category name indicates its corresponding AP result.

3.4.3 Quantitative Results

We extensively evaluated the proposed method with two popular detectors, SSD and RetinaNet, on Pascal VOC 2007 and COCO val2017, respectively. We also compared the proposed algorithm with state-of-the-art methods, SoftMax [120] and IoUNet [122]. Since these methods are confined to the application of R-CNN based two-stage frameworks, they are highly dependent on the pre-sift scheme by RPN, where the proposal pool could already be regarded as clean data. The proposed method aims at anchor-based single-shot detection models without the pre-sift scheme, where the anchor-based produced boxes can be regarded as rough candidates. We conduct experiments and verify that we can not directly introduce the above algorithms in the single-shot fashion as a comparison, like the one proposed by IoUNet by ranking the boxes using localization confidence. It leads to a sharp drop in mAP. It is understandable that the produced boxes by anchors are rough candidates that are unreliable for quality ranking. Therefore, these methods can not be directly applied to single-shot models.

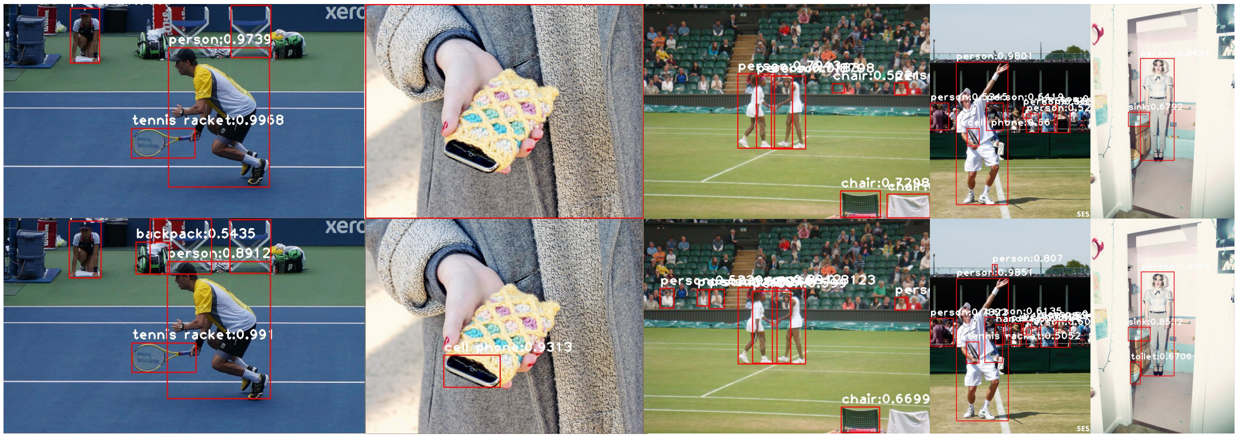


Figure 3.5: COCO cases compared between RetinaNet (1st row) and RetineNet+LC (2nd row), where LC version performs better for hard objects.

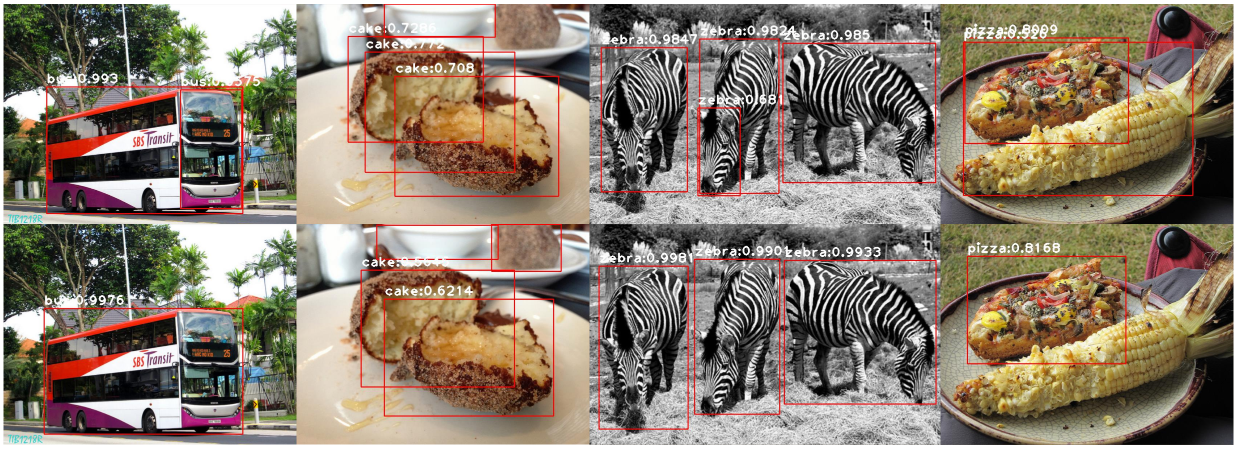


Figure 3.6: COCO cases compared between RetinaNet (1st row) and RetinaNet+LC (2nd row), where LC version has better filtering for low-quality boxes.

To make a fair comparison, we integrate the core idea of IoU-guided NMS produced by IoUNet to merge the classification cluster into our algorithm and form the 'LC+CS', described in lines 11 and 12 of Algorithm 1. We then conduct a comparison and show quantitative results in Table 3.1 and 3.2. From Table 3.1 we can see that, compared with RetinaNet, LC model achieves stable improvement in most cases. Specifically, LC obtains better AP, $AP_{0.75}$, AP_M, and AP_L. For $AP_{0.75}$, LC achieves an enhancement of 0.8%, and for AP_M, LC improves by a margin of 0.7%. We can conclude that, for the detection accuracy with high requests and objects with the most common sizes, our method exhibits clear advantages. In Table 3.2, the LC version does not lead the ranking although its corresponding ILC version performs better than the original SSD model. This reflects the fact that the anchor-based single-shot models are sensitive to location accuracy and the produced rough candidates have less reliable location outputs than the expectation. From Tables 3.1 and 3.2 we can see that ‘LC+CS’ achieves the best results in both experiments, with over 1% enhancement in some cases, such as in RetinaNet at $\mathrm{AP}_{0.75}$. The results demonstrate that the introduction of classification clustering could compensate for the uncertainty caused by the location outputs.

3.4.4 Comparison and Discussion

In this section, we first discuss the quality of the predicted bounding boxes and investigate the upper bound of the performance of the LAAR model, and analyze the benefits of locscore learning. Here, all the results are evaluated on COCO2017 validation set using RetinaNet and ReinaNet with the LC models.

3.4.4.1 Fitter and tighter bounding boxes

In Figure 3.4, It is evident that the LC model predicts higher-qualified boxes than RetinaNet, and most of the boxes are tighter. Specifically, the boxes for the vase, the eagle, and the truck, are more accurate, and the box for the chair shows better performance in occluded cases. This demonstrates that the introduction of locscore learning can improve the accuracy of bounding boxes, and help select the best proposals that have the maximum alignment between the quality score and box quality. Tighter boxes in practice can help clear up some current dilemmas in the industry.

3.4.4.2 Better for hard objects

In Figure 3.5, it can be seen that the LC model is able to detect harder objects, like the small backpacks, the occluded cellphone, and the persons in the audience. These small or occluded objects are easy to be overlooked during the suppression process, as their classification scores can be small due to the deficiency of effective features. While with the locscore, the detection score can be calibrated as these hard objects could probably have high localization scores if they are labeled.

Thus, the actual detection accuracy can be raised by calibrating the box quality score especially when the original classification confidence is low.

3.4.4.3 Waiving low-quality boxes

In addition, from Figure 3.6, we can see the LC model is better at sifting out some low-qualified boxes, like the smaller bus box, the redundant cake, and zebra box. Similarly, this can be explained by the calibrated quality score. Although some boxes' classification scores are high, if their location scores are low, these boxes can still be regarded as low-quality objects, which could be disregarded in NMS.

3.5 Conclusion

This paper reveals the problem of object detection score as one of the primary limitations of current anchor-based single-shot object detectors. To address this issue, we have proposed the localization score (locscore) regression and location-aware box reasoning, where the classification score is aligned with the predicted locsocre so that the localization accuracy is taken into the assessment of the quality of the bounding box proposals, which has been overlooked in most popular object detection frameworks. Extensive experimental results show that the proposed approach can consistently improve the detector's performance to yield reliable bounding boxes. The proposed module can be directly applied to any single-shot object detection models to improve their performance in both classification and localization.

Chapter 4

Robust Structured Declarative Classifiers for 3D Point Clouds: Defending Adversarial Attacks with Implicit Gradients

Abstract

Deep neural networks for 3D point cloud classification, such as PointNet, have been demonstrated to be vulnerable to adversarial attacks. Current adversarial defenders often learn to denoise the (attacked) point clouds by reconstruction, and then feed them to the classifiers as input. In contrast to the literature, we propose a family of robust structured declarative classifiers for point cloud classification, where the internally constrained optimization mechanism can effectively defend against adversarial attacks through implicit gradients. Such classifiers can be formulated using a bilevel optimization framework. We further propose an effective and efficient instantiation of our approach, namely, *Lattice Point Classifier (LPC)*, based on structured sparse coding in the permutohedral lattice and 2D convolutional neural networks (CNNs) that is end-to-end trainable. We demonstrate state-of-the-art robust point cloud classification performance on ModelNet40 and ScanNet under seven different attackers. For instance, we achieve **89.51%** and **83.16%** test accuracy on each dataset under the recent JGBA attacker that outperforms DUP-Net and IF-Defense with PointNet by ∼70%.

4.1 Introduction

Point clouds are unstructured data which is widely used in real-world applications such as autonomous driving. To recognize them using deep neural networks, point clouds can be represented

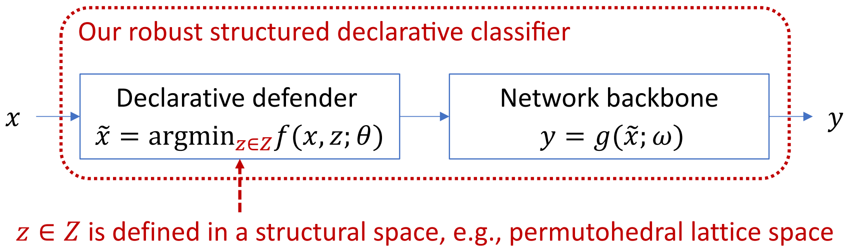

Our robust structured declarative classifier
x
Declarative defender
$\tilde{x} = \mathrm{argmin}_{z \in Z} f(x, z; \theta)$
Network backbone
$y = g(\tilde{x}; \omega)$
y
$z \in Z$ is defined in a structural space, e.g., permutohedral lattice space

Figure 4.1: Illustration of robust structured declarative classifiers, where our defender is optimized in a structural space.

as points [54], images [6], voxels [11], or graphs [67]. Recent works [128; 129; 76; 130; 74; 131; 132] have demonstrated that such deep networks are vulnerable to (gradient-based) adversarial attacks. Accordingly, several adversarial defenders [133; 134; 135] have been proposed for robust point cloud classification. The basic idea is often to denoise the (attacked) point clouds before feeding them into the classifiers as input to preserve their prediction accuracy.

Obfuscated gradients. In white-box adversarial attacks, the attackers are assumed to have full access to both classifiers and defenders. To defend against such attacks, one common way is to *break* the gradient over the input data in the backpropagation (either inadvertently or intentionally, *e.g.*, a defender is non-differentiable or prevents gradient signal from flowing through the network) so that the attackers fail to be optimized. Such scenarios are called obfuscated gradients. In [136] Athalye *et al.* have discussed the false sense of security in such defenders and proposed new methods, such as Backward Pass Differentiable Approximation (BPDA), to attack them successfully. Take DUP-Net [133] for example, where a non-differentiable Statistical Outlier Removal (SOR) defense strategy was proposed. In [76] Ma *et al.* proposed Joint Gradient Based Attack (JGBA) that can compute the gradient with a linear approximation (an instantiation of BPDA) of the SOR defense to attack DUP-Net successfully.

Implicit gradients. Now let us consider the scenarios where both defenders and classifiers are differentiable. Then in order to defend the adversarial attacks, one way is to make the calculation of the gradient challenging. To this end, implicit gradients [137] may be more suitable for designing the defenders. *An implicit gradient, $\frac{\partial y}{\partial x}$, is defined by a differentiable function h that takes x, y as its input,* i.e., $\frac{\partial y}{\partial x} = h(x, y)$. Such an equation can be also considered as a first-order ordinary differential equation (ODE), which is solvable (approximately) using Euler's Method [138]. Here we assume that the gradients through the classifiers can be easily computed, which often holds empirically. To our best knowledge, so far there is no work on designing adversarial defenders for 3D point clouds based on implicit gradients.

Declarative networks. Implicit gradients require equations that contain both the input and output of a defender. One potential solution for this is to introduce optimization problems as the defenders, where the first-order optimality conditions provide such equations. In the literature, there have been some works [139; 140; 141; 137] that proposed optimization as network layers in deep neural networks. Recently in [142] Gould *et al.* generalized these ideas and proposed deep declarative networks. A *declarative* network node is introduced where the exact implementation of the forward processing function is not defined; rather the input-output relationship $(x \mapsto \tilde{x})$ is defined in terms of behavior specified as the solution to an optimization problem $\tilde{x} \in \arg\min_{z \in \mathscr{Z}} f(x, z; \theta)$. Here f is an objective function, θ denotes the node parameters, and $\mathscr{Z}$ is the feasible solution space. In [142] a robust pooling layer was proposed as a declarative node using unconstrained minimization with various penalty functions such as Huber or Welsch, which can be efficiently solved using Newton's method or gradient descent. The effectiveness of such pooling layers was demonstrated for point cloud classification.

Our approach. Motivated by the methods above, in this paper, we propose a novel robust structured declarative classifier for 3D point clouds by embedding a declarative node into the networks, as illustrated by Fig. 4.1. Different from robust pooling layers in [142], our declarative defender is designed to reconstruct each point cloud in a (learnable) structural space as a means of denoising. To this end, we borrow the idea from structured sparse coding [143; 144; 145] by representing each point as a linear combination of atoms in a dictionary. Together with the backbone networks, the training of our robust classifiers can lead to a bilevel optimization problem. Considering the inference efficiency, one plausible instantiation of our classifiers, as illustrated in Fig. 4.2, is to define the structural space using the permutohedral lattice [146; 147; 148], project each point cloud onto the lattice, generate a 2D image based on the barycentric weights, and feed the image to a

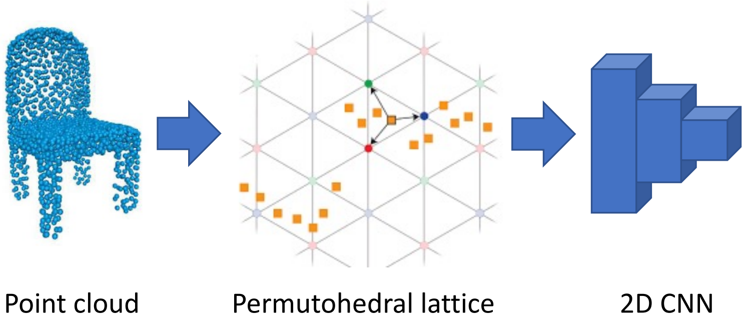


Figure 4.2: Illustration of our Lattice Point Classifier (LPC).

2D convolutional neural network (CNN) for classification. We call this instantiation *Lattice Point Classifier (LPC)*.

Our contributions. We summarize our contributions below:

- We propose a family of novel robust structured declarative classifiers for 3D point clouds where the declarative nodes defend the adversarial attacks through implicit gradients. To the best of our knowledge, we are the *first* to explore implicit gradients in robust point cloud classification.
- We propose a bilevel optimization framework to learn the network parameters in an end-to-end fashion.
- We propose an effective and efficient instantiation of our robust classifiers based on the structured sparse coding in the permutohedral lattice and 2D CNNs.
- We demonstrate superior performance of our approach by comparing it with the state-of-the-art adversarial defenders under the state-of-the-art adversarial attackers.

4.2 Related Works

Deep learning for 3D point clouds. Based on the point cloud representations, we simply group some typical deep networks into four categories. *Point-based networks* [54; 55; 56; 57] directly take each point cloud as input, extract point-wise features using multi-layer perceptrons (MLPs), and fuse them to generate a feature for the point cloud. *Image-based networks* [58; 59; 60; 61; 6] often project a 3D point cloud onto a (or multiple) 2D plane to generate a (or multiple) 2D image for further process. *Voxel-based networks* [62; 11; 63; 64; 65] usually voxelize each point cloud into a volumetric occupancy grid and further some classification techniques such as 3D CNNs are used for the tasks. *Graph-based networks* [66; 67; 68; 69; 70] often represent each point cloud as a graph such as KNN or adjacency graph which are fed to train graph convolutional networks (GCNs). A nice survey can be found in [71].

Adversarial attacks on point clouds. Such attacks aim to modify the input point clouds in a way that is not noticeable but can fool a classifier. Attackers can either have a target or not. Targeted attackers try to fool the classifier to predict a specified wrong class, while untargeted ones do not care about the predicted class as long as it is wrong. Nice surveys on adversarial attacks can be found in [149; 150; 151]. Below we summarize some typical attackers for point clouds:

- *Point perturbation [72; 73; 74; 75; 76].* Inspired by the Fast Gradient Sign Method (FGSM) [152], point perturbation-based adversarial attackers, such as recent JGBA [76], have been proposed by adding a small perturbation on each point. Such perturbations are measured by a perturbation distance used in learning the adversarial examples. For instance, in [72] the perturbation is constrained onto the surface of an ε-ball.
- *Point addition [74; 131; 73].* Independent point attackers initially pick some points from target classes, add small perturbations to them, and finally append these points to the victim point clouds. Cluster attackers similarly pick the most critical points and then find a specified number of small point clusters to append to the victim point clouds. Object attackers attach foreign object point clouds to the original point clouds by scaling and moving the foreign objects to the center of the clusters obtained in the same way as cluster attackers do.
- *Point dropping.* As an important strategy, adversarial attackers can pick some critical points from each input point cloud, and drop them to fool the classifier. However, dropping points is a non-differentiable operation. To address this issue in learning, Zheng *et al.* [153] proposed creating saliency maps by viewing dropping a point as moving this point to the cloud center. Wicker *et al.* [131] designed an algorithm to iteratively find the points to drop by minimizing a predefined objective function, similarly in [73].

- *Others.* Hamdi *et al.* [154] proposed a transferable adversarial perturbation attacker based on an adversarial loss that can learn the data distribution. Zhao *et al.* [13] proposed a black-box (*i.e.,* no access to the models) attacker with zero loss in isometry, as well as a white-box (*i.e.,* full access to the models) attacker based on the spectral norm. LG-GAN [128] is a generative adversarial network (GAN) based attacker that learns and incorporates target features into victim point clouds. Backdoor attackers were proposed in [129; 155; 156] to trick the 3D models by inserting adversarial point patterns into the training set so that the victim models learn to recognize the adversarial patterns during inference.

Adversarial defense on point clouds. Adversarial defenses aim to denoise the input point clouds to recover the ground-truth labels from the classifiers. Nice surveys on adversarial defenses can be found in [149; 150; 151]. Below we summarize some typical defenders for point clouds:

- *Statistical outlier removal (SOR) [157].* SOR can be used to remove local roughness on the (smooth) surface as a means of defense. SOR is not differentiable, producing obfuscated gradients for defenders. DUP-Net [133] uses SOR and an upsampling network to reconstruct higher-resolution point clouds. Similarly, IF-Defense [134] utilizes SOR, followed by a geometry-aware model to recover the surface by encouraging evenly distributed points. However, such defenders have been demonstrated to be attackable successfully in [158; 76]. Dong *et al.* [159] proposed replacing SOR with an attention mechanism.
- *Random sampling.* Yang *et al.* [73] suggested that 3D models with random sampling are robust to adversarial attacks. PointGuard [135] proposed majority voting for point cloud classification by predicting multiple randomly subsampled point clouds.
- *Data augmentation.* Tramer *et al.* [160] demonstrated that data augmentation can effectively account for adversarial attacks. Tu *et al.* [161] proposed generating physically realizable adversarial examples to train robust Lidar object detectors. Zhang *et al.* [162] proposed randomly permuting training data as a simple data augmentation strategy. PointCutMix [163] pairs two training clouds and swaps some points between the pair to generate new training data.

Permutohedral lattice. Permutohedral lattice is a powerful operation to project the coordinates from a high dimensional space onto a hyperplane that defines the lattice. It has been widely used in high dimensional filtering [146; 164] that consists of three components, *i.e.,* splat, blur and slice. In particular, we illustrate the splat in Fig. 4.2, where each square represents a projection (*i.e.,* projected point) from a 3D coordinate. The splat first locates the enclosing lattice simplex for the 3D point and calculates the vertex coordinates of the simplex. Then each projection distributes its value to the vertices using barycentric interpolation with barycentric weights that are calculated as the normalized triangular areas between the projection and any pair of its corresponding lattice vertices. We refer the readers to [146] for more details. Recently permutohedral lattice has been successfully explored in point cloud segmentation [147; 148; 165] with remarkable performance.

4.3 Robust Structured Declarative Classifiers

4.3.1 Structured Declarative Defender

Recall that adversarial defenders often aim to denoise the input point clouds by reconstructing them in certain ways, and sparse coding [166] is one of the classic approaches for finding a sparse representation of the input data in the form of a linear combination of basic elements as well as those basic elements themselves. Due to its simplicity, we consider using sparse coding as a means to construct declarative nodes.

Specifically, in the 3D space given a (learnable) dictionary $\mathbf{B} \in \mathbb{R}^{3\times N}$ with $N \gg 3$ atoms, (structured) sparse coding aims to solve the following optimization problem for each point $\mathbf{x}_i \in \mathbb{R}^3$ in a point cloud $\mathbf{x} = \{\mathbf{x}_i\} \subseteq \mathbb{R}^3$:

$$\tilde{\mathbf{x}}_i \in \arg\min_{\mathbf{z}\in\mathscr{Z}} f(\mathbf{x}_i, \mathbf{z}) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{B}\mathbf{z}\|^2 + \phi(\mathbf{z}), \tag{4.1}$$

where ϕ denotes a regularization term, and $\mathscr{Z} \subseteq \mathbb{R}^N$ denotes a structural feasible solution space.

Obfuscated & implicit gradients in $\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}_i}\Big|_{\mathbf{z}=\tilde{\mathbf{x}}_i}$. To see this, we can rewrite Eq. (4.1) as $\tilde{\mathbf{x}}_i \in \arg\min_{\mathbf{z}} F(\mathbf{x}_i, \mathbf{z}) = f(\mathbf{x}_i, \mathbf{z}) + \phi(\mathbf{z}) + \delta(\mathbf{z})$, where $\delta(\mathbf{z})$ denotes the Dirac delta function returning 0 if $\mathbf{z} \in \mathscr{Z}$ holds, otherwise, $+\infty$. Therefore, F will become non-differentiable when $\mathbf{z} \notin \mathscr{Z}$ or ϕ is non-differentiable over $\mathbf{z} \in \mathscr{Z}$, leading to obfuscated gradients. Otherwise, based on the first-order optimality condition, we have $\mathbf{B}^T\mathbf{B}\tilde{\mathbf{x}}_i - \mathbf{B}^T\mathbf{x}_i + \phi'(\tilde{\mathbf{x}}_i) = \mathbf{0}$, where $(\cdot)^T$ denotes the matrix transpose operator and ϕ' denotes the first-order derivative of ϕ. By taking another derivative on both sides, we then have a linear system $\left(\mathbf{B}^T\mathbf{B} + \phi''(\tilde{\mathbf{x}}_i)\right) \cdot \frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i} = \mathbf{B}^T$, if the second-order derivative, ϕ'', exists at $\mathbf{z} \in \mathscr{Z}$. Clearly, solving this linear system may be challenging, because $\phi''(\tilde{\mathbf{x}}_i)$ may not be computable and the matrix $\mathbf{B}^T\mathbf{B} + \phi''(\tilde{\mathbf{x}}_i)$ may be rank-deficient. Such phenomena lead to implicit gradients.

4.3.2 Parameter Learning via Bilevel Optimization

Fig. 4.1 illustrates our approach with two components, *i.e.,* a declarative defender f and a network backbone g that takes the outputs of the defender as input and then makes predictions. Equivalently we can formulate the training of such networks as a bilevel optimization problem as follows:

$$\begin{aligned} &\min_{\mathbf{B},\omega} \sum_{(\mathbf{x},y)\in\mathscr{X}\times\mathscr{Y}} \ell\Big(g(\tilde{\mathbf{x}};\omega), y\Big), \\ &\text{s.t.}\, \tilde{\mathbf{x}}_i \in \arg\min_{\mathbf{z}\in\mathscr{Z}} \frac{1}{2}\|\mathbf{x}_i - \mathbf{B}\mathbf{z}\|^2 + \phi(\mathbf{z}), \forall \mathbf{x}_i \in \mathbf{x}, \end{aligned} \tag{4.2}$$

where $(\mathbf{x}, y) \in \mathscr{X} \times \mathscr{Y}$ denotes a training sample with data $\mathbf{x}$ and label y, ℓ denotes a loss function such as cross-entropy, and $\mathbf{B}, \omega$ denote the defender and network parameters, respectively. Similar to training deep networks, we can solve this optimization problem using (stochastic) gradient descent.

Validity of $\frac{\partial g}{\partial \tilde{\mathbf{x}}_i} = \frac{\partial g}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\tilde{\mathbf{x}}_i}$ in the structural space. In a gradient-based adversarial attack, the gradient for modifying an input point $\mathbf{x}_i$ through backpropagation can be written as $\frac{\partial g}{\partial \tilde{\mathbf{x}}_i}\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i}$. Assuming that $\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i}$ can be computed exactly, then the gradient in the attack would hold in general only if $\mathbf{z}$ was unconstrained so that $\frac{\partial g}{\partial \tilde{\mathbf{x}}_i}$ is valid. Unfortunately in our case, this is not true as we constrain $\mathbf{z} \in \mathscr{Z}$. Therefore, the attack in the declarative node will produce inaccurate gradients, and such errors will be propagated to the adversarial examples, leading to failure cases together with $\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_i}$.

4.3.3 Instantiation: Lattice Point Classifier

So far we have explained the learning principles and defense philosophy in our approach. The key challenge now is how to design the structural space $\mathscr{X}$ and the regularizer ϕ to achieve robust point cloud classifiers that can be trained and inferred effectively and efficiently. To address this issue, we borrow the idea from permutohedral lattice, and propose an instantiation, namely, Lattice Point Classifier (LPC).

Geometric view on barycentric coordinates and their weights. Barycentric coordinates ($\overrightarrow{A}$, $\overrightarrow{B}$, $\overrightarrow{C}$ in Fig. 4.3) can be used to express the position of any point ($\overrightarrow{P}$) located on the entire triangle with three scalar weights (α, β, γ). To compute $\overrightarrow{P}$ using barycentric coordinates we can always use the following equation:

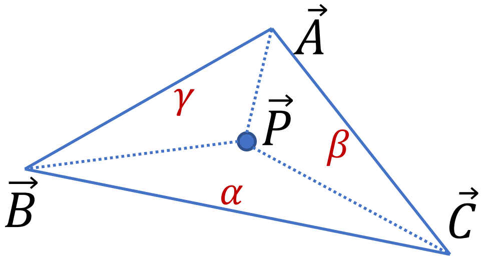


Figure 4.3: Illustration of barycentric coordinates and weights.

$$\overrightarrow{P} = \alpha\overrightarrow{A} + \beta\overrightarrow{B} + \gamma\overrightarrow{C}, \exists \alpha, \beta, \gamma \geq 0, \alpha + \beta + \gamma = 1. \tag{4.3}$$

Note that this equation holds for an arbitrary dimensional space, including the 3D space. Now given $\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{C}, \overrightarrow{P}$, we can compute α, β, α using the normalized areas. Taking γ for example, we can compute it as follows:

$$\gamma = \frac{\|\overrightarrow{AB} \times \overrightarrow{AP}\|}{\|\overrightarrow{AB} \times \overrightarrow{AC}\|} \propto \|\overrightarrow{AB} \times \overrightarrow{AP}\|, \tag{4.4}$$

where $\overrightarrow{AB} = \overrightarrow{B} - \overrightarrow{A}, \overrightarrow{AP} = \overrightarrow{P} - \overrightarrow{A}, \overrightarrow{AC} = \overrightarrow{C} - \overrightarrow{A}$, $\times$ denotes the cross product operator, and $\|\cdot\|$ denotes the ℓ_2 norm of a vector measuring its length. Clearly, the barycentric weights define a nonlinear mapping that can be computed efficiently using the splat operation for the permutohedral lattice.

Constructing $\mathscr{X}$ and ϕ using barycentric weights. By substituting Eq. (4.3) into Eq. (4.1), we manage to define a structured sparse coding problem, where the structural space $\mathscr{X}$ and the regularizer ϕ can be constructed as follows:

$$\mathscr{Z} \overset{def}{=} \left\{ \mathbf{z} \mid \mathbf{z}^T \mathbf{e} = 1, \mathbf{z} \succeq \mathbf{0} \right\}, \tag{4.5}$$

$$\phi(\mathbf{z}) \overset{def}{=} \lambda \sum_{n=1}^{N} \|\mathbf{B}\mathbf{z} - \mathbf{B}_n\| \cdot 1_{\{\mathbf{z}_n > 0\}}, \tag{4.6}$$

where $\mathbf{e}$ denotes a vector of 1's, $\succeq$ denotes the entry-wise operator of $\geq$, $\mathbf{B}_n \in \mathbb{R}^3$ denotes the n-th column in $\mathbf{B}$, $1_{\{\cdot\}}$ denotes a binary indicator returning 1 if the condition holds, otherwise 0 (*i.e.,* the *binarization* of barycentric weights), and $\lambda \geq 0$ is a small constant controlling the contribution of ϕ to the objective so that it will not be dominated by ϕ.

Proposition 1 *Supposing that* $\mathbf{B}$ *in Eq.* (4.1) *represents the vertices in a permutohedral lattice that is large enough to cover all possible projections from points among the data, then there exists a solution to minimize the reconstruction loss using three vertices, at most, and the minimum loss is equal to the projection loss onto the lattice.*

This is because Eq. (4.3) defines a lossless representation using a linear combination of three vertices. The only loss occurs when projecting a point to the permutohedral lattice.

Defense mechanism in LPC. Fig. 4.4 illustrates how our declarative defender works with the permutohedral lattice, where the triangle, circles, and squares represent a lattice cell, 3D points and their projections on the cell, respectively. In the adversarial point cloud, the attacker modifies a point from $\mathbf{x}_i$ to $\mathbf{x}_i'$. Then during the inference, our defender projects $\mathbf{x}_i'$ to $\tilde{\mathbf{x}}_i'$ on the lattice.

Proposition 2 *Supposing* $\tilde{\mathbf{x}}_i = \alpha\overrightarrow{A} + \beta\overrightarrow{B} + \gamma\overrightarrow{C}$ *where* α, β, γ *are the barycentric weights, then in order to guarantee that* $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_i'$ *lie in different cells, the distance between* $\mathbf{x}_i$ *and* $\mathbf{x}_i'$ *should be bigger than the shortest distance be-*

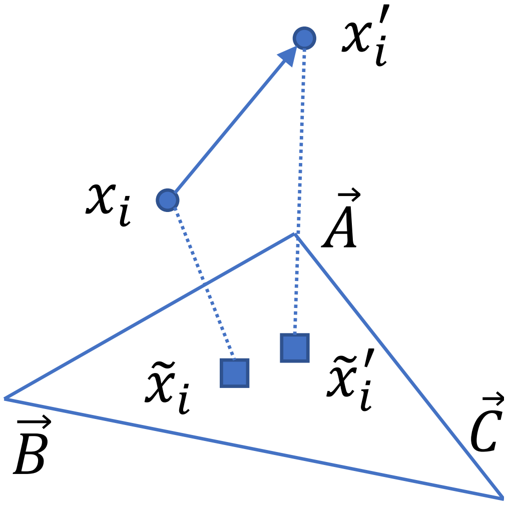


Figure 4.4: Illustration of our defense mechanism.

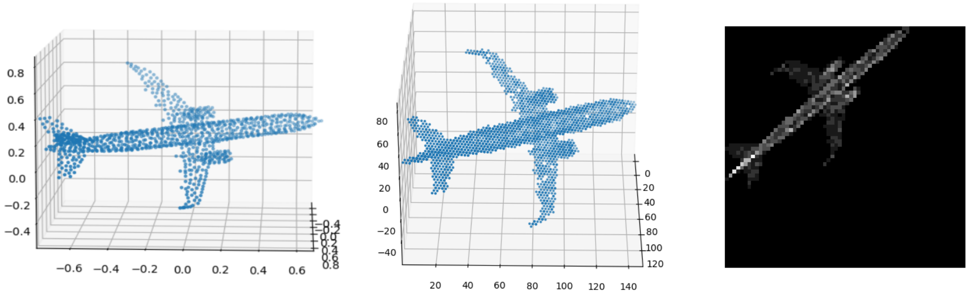


Figure 4.5: Illustration of **(left)** a point cloud, **(middle)** its lattice representation, **(right)** its image for classification.

tween $\tilde{\mathbf{x}}'_i$ and the boundary of the triangle, that is:

$$\|\mathbf{x}_i - \mathbf{x}'_i\| > \min\left\{\frac{\alpha}{\|\overrightarrow{BC}\|}, \frac{\beta}{\|\overrightarrow{AC}\|}, \frac{\gamma}{\|\overrightarrow{AB}\|}\right\} \cdot s, \tag{4.7}$$

where s denotes the area of the triangle. In particular, if the triangle is equilateral, then $\|\mathbf{x}_i - \mathbf{x}'_i\| > \frac{\sqrt{3}}{2} l \cdot \min\{\alpha, \beta, \gamma\}$ where l denotes the side length.

It will be more intuitive to understand this result if we binarize the barycentric weights before feeding them into the backbone network for classification because different projections lying in the same lattice cell will lead to the same representation. This could be an effective way to remove the adversarial noise in the data. Also, this result indicates that (1) the points whose projections are closer to the boundary are easier to change their sparse representations, (2) the movements that make such changes are proportional to the scale of the lattice cell. In general, larger cells will be more tolerant to the adversarial noise, but they may sacrifice the generalization of the classifiers.

Workflow of LPC. To summarize, it is operated as follows:

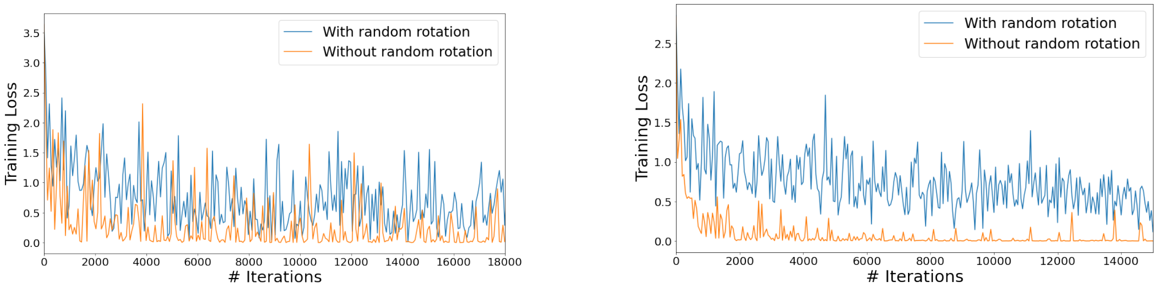
1. Given a point cloud, the barycentric weights of each point are computed using the splat for permutohedral lattice;
2. Generate an image for the point cloud by averaging the barycentric weights over all the points (after binarization if applied) and aligning the lattice with the image representation (see Fig. 4.5 for illustration);
3. Apply a 2D CNN as the backbone network to classify the image produced by the point cloud.

Implementation. The key challenge in our implementation of LPC is how to determine the projection matrix for the hyperplane and the scale of each permutohedral lattice cell.

- *Projection matrix:* By referring to [146], we initialize the projection matrix as $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$ and train the network in an end-to-end fashion. However, we observe that a big update of this matrix will make the training crash, and to avoid this issue, the update per iteration has to be tiny, leading to almost unnoticeable change eventually. The reason for this phenomenon is that this matrix has to satisfy certain requirements (see [146]), and thus the unconstrained update in backpropagation cannot work here. Therefore, in our experiments, we initialize and fix the projection matrix. We utilize the code[1] in [148] for the lattice transformation.
- *Scale of lattice cell:* The parameter is simply not differentiable in backpropagation, and thus we tune it as a predefined hyper-parameter using cross-validation with grid search, same for the other hyper-parameters such as learning rate. Such scales have a significant impact on the image resolution used in 2D CNNs for classification.

Specifically, we evaluate our LPC comprehensively based on three different CNNs, *i.e.,* VGG16 [1], ResNet50 [3], and EfficientNet-B5 [167] as the backbone network with random initialization. By default, the image resolution for each backbone network is 512×512, 128×128 and 456×456, respectively. On ModelNet40 [11] we train the three models using a learning rate of 10^{-4}, but on ScanNet [168] we only train our best model, *i.e.,* EfficientNet-B5, using a learning rate of 5×10^{-5}. We use Adam [169] as our optimizer in all of our experiments with weight decay of 10^{-4} and learning rate decay of 0.7 for every 20 epochs. Dropout [170] and data augmentation are applied as well when needed.

[1] https://github.com/laoreja/HPLFlowNet



(a) ModelNet40

(b) ScanNet

Figure 4.6: Training loss comparison on both datasets using EfficientNet-B5 as the backbone.

4.4 Experiments

Datasets. We conduct our experiments on ModelNet40 [11] and ScanNet [168]. ModelNet40 has a collection of 12,311 3D CAD objects from 40 common categories. It is split into 9,843 training and 2,468 test samples. Following [54; 74], we uniformly sample 1,024 points from the surface of the original point cloud per object and scale them into a unit ball. ScanNet contains 1,513 RGB-D scans from over 707 real indoor scenes with 2.5 million views. Following [56], we generate 12,445 training and 3,528 test point clouds from 17 categories, with 1,024 points for each point cloud as well.

Baselines. We compare our Lattice Point Classifier (LPC) with different adversarial defenders for point clouds that work with PointNet [54], including DUP-Net [133] (with SOR and the upsampling network), IF-Defense [134] (with ConvONet [171]), and robust pooling layer (RPL) [142]. We utilize public code to train PointNet[2] using the default setting and evaluate DUP-Net and IF-Defense based on the same code[3]. Then we report the best performance for each defender after fine-tuning. Specifically, we set $k = 2$ (the number of neighbor points) in KNN and $\alpha = 1.1$ (the percentage of outliers) for SOR, use 2 for the upsampling rate in DUP-Net, and choose the Welsch penalty function [172] for RPL[4] to replace the max pooling layer in PointNet. The model with RPL is trained with adversarial point clouds where 10% of input points are replaced by random

[2] https://github.com/yanx27/Pointnet_Pointnet2_pytorch
[3] https://github.com/Wuziyi616/IF-Defense
[4] https://github.com/anucvml/ddn

outliers.

Adversarial attackers. Seven attackers are utilized to evaluate the robustness of different point cloud classifiers, including two untargeted attackers (FGSM[5] [152] and JGBA[6] [76]), four targeted attackers (perturbation, add, cluster and object attackers[7] from [74]), and an isometry transformation based attacker[8] [13] consisting of the untargeted black-box TSI attack and the targeted white-box CTRI attack.

We apply both FGSM and JGBA to the full test set of both datasets. We slightly modify FGSM for attacking DUP-Net and IF-Defense under the white-box setting as both defenders are non-differentiable. To do so, during an attack we simulate the SOR process and obtain the indices of the remaining points based on which the gradient is passed to the remaining points. By default, the parameter ε in FGSM is set to 0.1, and the perturbation norm constraint ε, number of iterations n and step size α in JGBA are set to 0.1, 40, 0.01, respectively. Such parameters work well in practice.

For the targeted attackers, by following [74] we pick 10 large classes from ModelNet40, where a batch of 6 point clouds per class is randomly selected and attacked using the other 9 classes as the targets, leading to $10 \times 9 \times 6 = 540$ victim-target pairs. Similarly, from ScanNet we randomly select a batch of 6 point clouds as well from the 7 classes that contain more than 100 point clouds in test data. The learning rate of all the targeted attackers is set to 0.01. For DUP-Net and IF-Defense, we first attack clean PointNet to obtain adversarial point clouds, and then feed them into DUP-Net and IF-Defense for prediction. The perturbation attack first introduces small random perturbations on all original points, and uses L_2 norm to constrain the adversarial shifts. Using the add attacker, we add 60 points to each cloud and use Chamfer distance as the metric. Cluster and object attackers generate the initial clusters with parameter $\varepsilon = 0.11$ in DBSCAN [173]. Using the cluster attacker, we add 3 clusters of 32 points to the original clouds. Using the object attacker, three 64-point adversarial objects are attached to each original point cloud.

[5] https://pytorch.org/tutorials/beginner/fgsm_tutorial.html

[6] https://github.com/machengcheng2016/JGBA-pointcloud-attack

[7] https://github.com/xiangchong1/3d-adv-pc

[8] https://github.com/skywalker6174/3d-isometry-robust

Table 4.1: Our learning choice comparison in terms of test accuracy (%) with learning rate 10^{-4}.

T-Net [54]		✓			✓	
Binarized weights			✓		✓	✓
Random rotation				✓		✓
ResNet-50 on ModelNet40	88.2	87.3	**88.9**	60.0	88.2	75.2
EfficientNet-B5 on ModelNet40	-	-	**89.5**	83.3	-	84.8
EfficientNet-B5 on ScanNet	80.5	-	80.0	**82.6**	-	-

Table 4.2: Running time (ms) analysis on an NVIDIA V100S GPU with batch size 1 under perturbation attacks.

	Declarative node	EfficientNet-B5	Total
Inference	17.0	41.7	58.7
Attack	15.5	165.3	180.8

For the TSI/CTRI attacker in [13], we evaluate the performance of LPC on ModelNet40 using EfficientNet-B5. By following [13] we use this attacker with the default settings to attack 2,000 randomly selected point clouds. The attacker applies the black-box TSI attacker first to trick the models, and then the white-box CTRI attacker if TSI fails.

4.4.1 Ablation Study

Learning choices. In Table 4.1 we list three learning choices that we would like to evaluate for improving the performance of vanilla LPC, *i.e.,* T-Net used in PointNet, binarized barycentric weights, and random rotation in data augmentation (together with point cloud random shifting and dropping [54]). We can see that: (1) T-Net seems to deteriorate the performance always. (2) Binarized weights work better on ModelNet40 than ScanNet, but compared with using barycentric weights the difference is <1%. (3) Random rotation improves the performance on ScanNet, but worsens it on ModelNet40. This phenomenon can be partially explained by the training loss curves as shown in Fig. 4.6, where on ScanNet the overfitting occurs clearly without random rotation while on ModelNet40 random rotation makes the convergence slower and unstable.

Running time. Table 4.2 lists our running time for each key component as well as the whole net-

Table 4.3: Test accuracy (%) comparison (higher is better) on the full test datasets, where "-" indicates no result.

		PointNet	PointNet w/ DUP-Net [133]	PointNet w/ IF-Defense [134]	PointNet w/ RPL [142]	LPC w/ VGG16	LPC w/ ResNet50	LPC w/ EfficientNet
ModelNet40	No attack	**90.15**	89.30	87.60	84.76	88.65	88.90	89.51
	FGSM [152]	45.99	61.63	38.75	0.04	88.65	88.90	**89.51**
	JGBA [76]	0.00	1.14	5.37	0.00	88.65	88.90	**89.51**
ScanNet	No attack	**84.61**	83.62	80.19	76.02	-	-	83.16
	FGSM [152]	45.66	73.67	71.14	1.70	-	-	**83.16**
	JGBA [76]	0.00	7.77	13.45	0.00	-	-	**83.16**

Table 4.4: Attack success rate (%) comparison (lower is better), where "-" indicates no result. The standard deviations of our LPC range from 0% to 0.28% in success rate on ModelNet40.

		PointNet	PointNet w/ DUP-Net [133]	PointNet w/ IF-Defense [134]	PointNet w/ RPL [142]	LPC w/ VGG16	LPC w/ ResNet50	LPC w/ EfficientNet
ModelNet40	FGSM [152]	48.99	30.77	55.78	99.95	**0.00**	**0.00**	**0.00**
	JGBA [76]	100.00	98.73	93.85	100.00	**0.00**	**0.00**	**0.00**
	Perturbation [74]	100.00	**0.095**	**0.095**	3.95	0.56	0.37	0.38
	Add [74]	99.72	**0.095**	**0.095**	3.33	0.56	0.19	0.19
	Cluster [74]	98.34	6.76	6.30	17.04	1.11	**0.93**	1.21
	Object [74]	98.43	1.02	1.11	74.07	0.93	1.11	**0.75**
ScanNet	FGSM [152]	46.03	10.29	11.28	97.76	-	-	**0.00**
	JGBA [76]	100.00	90.55	83.21	100.00	-	-	**0.00**
	Perturbation [74]	100.00	3.17	**2.38**	3.03	-	-	12.70
	Add [74]	100.00	**1.98**	2.38	18.65	-	-	2.78
	Cluster [74]	100.00	30.16	23.81	40.87	-	-	**9.92**
	Object [74]	100.00	7.14	7.54	85.71	-	-	**5.95**

work. Clearly, the gradient passing through the declarative node takes relatively constant time in both feedforward and backpropagation. Considering the depth of EfficientNet-B5, the time spent on the declarative node is actually pretty long (recall that there is no learnable parameter inside), especially during inference. This partially validates our intuition of defending the adversarial attacks using implicit gradients. We can also improve the running time using shallower networks such as VGG16 (from 17fps to 45fps) by slightly sacrificing the performance.

4.4.2 State-of-the-art Performance Comparison

We measure the robustness of classifiers using two metrics, *i.e.,* classification accuracy, and attack success rate. Classification accuracy under attacks is measured by feeding the entire adversarial attack test set to the victim models. It is only calculated for untargeted attackers. The attack

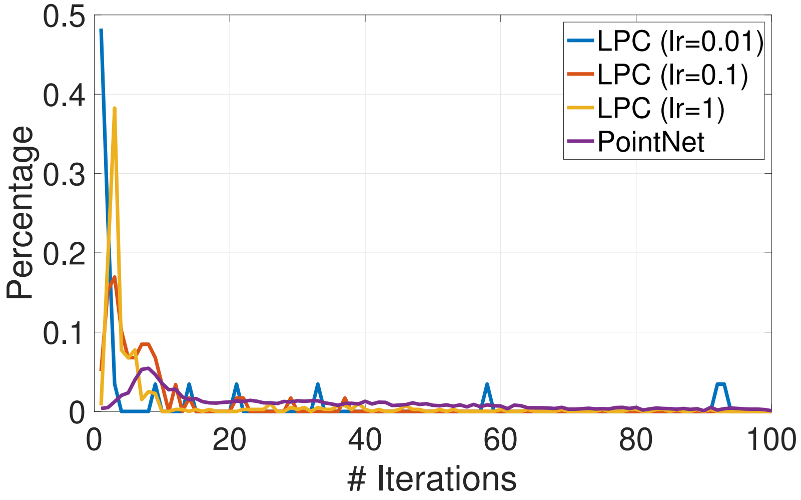


Figure 4.7: Distribution comparison of successful perturbation attacks on ModelNet40. To avoid the sparsity, LPC statistics are collected based on the cases from all three models.

success rate is the ratio between the number of successful attacks to the number of all attempted attacks. For untargeted attacks, tricking the model to predict a wrong class is considered a success, while for targeted attacks it will have to trick the victim model to a specific class to be considered as successful. Untargeted attackers will only attack the point clouds that are correctly classified by the victim models, and targeted attackers will attack victim-target pairs.

4.4.2.1 Classification Accuracy

We summarize our comparison in Table 4.3. On ScanNet, we only show the performance of LPC with EfficientNet-B5, because it achieves the best accuracy on ModelNet40. We can see that: (1) On the clean test data with no attack, PointNet outperforms all the robust classifiers by small margins. (2) Using both attackers, our LPC variants work consistently and significantly better than the other defender-based robust classifiers as well as vanilla PointNet by large margins. For instance, compared with PointNet with IF-Defense under the JGBA attacks, the performance gaps are 84.14% and 69.71% on ModelNet40 and ScanNet, respectively. (3) For the backbone networks in LPC, it seems that the difference in performance is small among different CNNs. Though more evaluations are needed to confirm this, it also demonstrates the robustness of our approach.

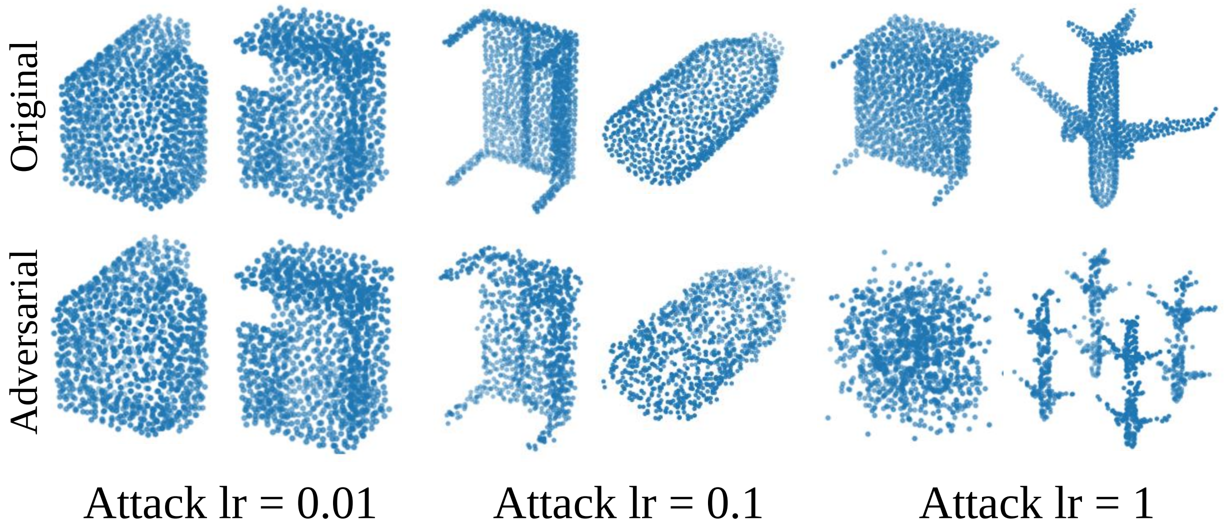


Figure 4.8: Successful adversarial perturbation examples on ModelNet40 for LPC with EfficientNet-B5.

4.4.2.2 Attack Success Rate

Performance summary. We list our comparison in Table 4.4 using six different attackers. We can see that: (1) Our LPC variants achieve perfect results under both FGSM and JGBA attacks. Notice that compared with the other attackers, these two attackers aim to find adversarial examples fully based on gradients with no sampling. Their failure again strongly demonstrates the power of implicit gradients in defending gradient-based adversarial attacks. (2) The SOR defense mechanism seems to work better for the perturbation and add attackers, but worse for the cluster and object attackers, compared with our LPC. However, overall our LPC still achieves the best performance.

Using the TSI/CTRI attacker [13], our LPC with EfficientNet-B5 achieves the same success rate for both TSI and CTRI attackers on ModelNet40. Without random rotation, the result is 99.54%, but with random rotation, the performance decreases to 67.33% which is significantly better than PointNet (99.50% and 99.55% for TSI and CTRI, respectively). Such results also demonstrate that random rotation to point clouds as a means of data augmentation improves not only accuracy but also model robustness.

Importance of gradients in finding successful adversarial examples for LPC. To attack each point cloud, the four targeted attackers in [74] conduct 10 random searches where 500 iterations are done to find the optimal adversarial samples. We notice that for PointNet, the best attacks could occur at any time within these 500 iterations, but for our LPC most optimal attacks happen at the first few iterations, as shown in Fig. 4.7 with different learning rates for the perturbation attacker. This behavior indicates that, in order to attack LPC, random search (a sampling-based method) has contributed more to most of the successful attacks, rather than gradient-based iterations. With the increase of the learning rate, the gradients start to find more optimal adversarial samples. However, as we illustrate in Fig. 4.8, with larger learning rates, the adversarial samples will not look similar to the original point clouds. For instance, with a learning rate of 1, the point cloud of airplane has been totally changed to a mixture of smaller airplane point clouds, which is not an adversarial attack anymore. To sum up, such analysis again demonstrates the great potential of implicit gradients in defending adversarial attacks.

4.4.3 Results under CTRI Attack

This section provides additional quantitative results on CTRI attack [13] and more qualitative adversarial samples to the main paper. Combined targeted restricted isometry (CTRI) attack first deploys a black-box attack, namely Thompson Sampling Isometry (TSI) attack, on the victim models, and if TSI fails CTRI then uses a gradient-based attacker [13]. The results of CTRI attack

Table 4.5: Attack success rate (%) comparison (lower is better) of CTRI attack [13], where "-" indicates no result. And the model with (*) means it is trained with random rotation as data augmentation.

	ModelNet40		ScanNet	
	TSI [13]	CTRI [13]	TSI [13]	CTRI [13]
PointNet [54]	99.50	99.55	99.94	100.00
PointNet w/ DUP-Net [133]	99.16	99.78	99.64	100.00
PointNet w/ RPL [142]	99.71	99.77	99.93	100.00
LPC w/ VGG16	99.43	99.43	–	–
LPC w/ ResNet50	98.98	98.98	–	–
LPC w/ EfficientNet	99.66	99.66	99.81	99.81
LPC* w/ EfficientNet	47.49	47.49	65.20	65.20

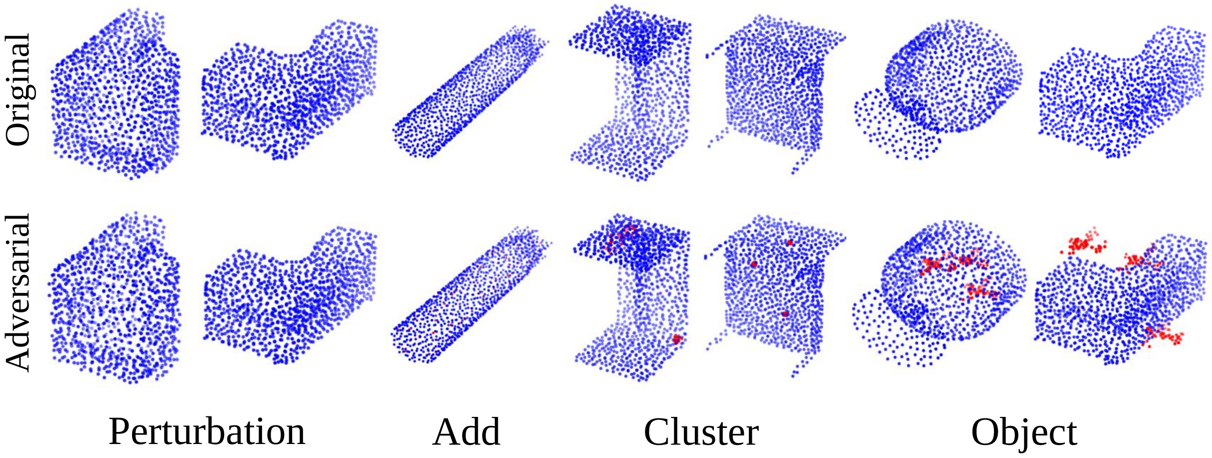


Figure 4.9: Successful adversarial examples on ModelNet40. For add, cluster and object attacks, the added points are in red.

in Table 4.5 is obtained by randomly attacking 2,000 point clouds. As we can see, CTRI on our LPC models doesn't increase the attack success rate compared to TSI-only attack, indicating the gradient-based attacker in CTRI fails to trick LPC. It again proves the effectiveness of implicit gradient as a defense mechanism against gradient-based attackers. The results also show data augmentation such as random rotation significantly improves model robustness. It reduces the black-box attack success rate from $> 99\%$ to 47.49% and 65.20% on ModelNet40 and ScanNet respectively.

4.4.4 Visualization

We select some additional successful adversarial samples from different attackers on ModelNet40. It is shown in Figure 4.9. Cluster and object attacks alter the surface more significantly than perturbation and add attacks, which explains why all defenses have lower performances against cluster and object attacks.

4.5 Conclusion

In this paper, we aim to address the problem of robust 3D point cloud classification by proposing a family of novel robust structured declarative classifiers, where a declarative node is defined by a constrained optimization problem such as the reconstruction of point clouds. The key insight in our approach is that the implicit gradients through the declarative node can help defend the adversarial attacks by leading them to wrong updating directions for inputs. We formulate the learning of our classifiers based on bilevel optimization, and further propose an effective and efficient instantiation, namely, Lattice Point Classifier (LPC). The declarative node in LPC is defined as structured sparse coding in the permutohedral lattice, whose outputs, *i.e.,* barycentric weights, are further transformed into images for classification using 2D CNNs. LPC is end-to-end trainable, and achieves state-of-the-art performance on robust classification on ModelNet40 and ScanNet using seven different adversarial attackers.

Limitations. Currently, the projection in the permutohedral lattice transformation in LPC is not learned but simply fixed as initialization. Also, more evaluations for demonstrating the robustness of our approach across different backbone networks and datasets are desirable. Therefore, in our future work, we will investigate more on how to properly learn the projection with its physical conditions and conduct more experiments to further demonstrate our robustness.

Chapter 5

Robust 3D Point Clouds Classification based on Declarative Defenders

Abstract

3D point cloud classification requires distinct models from 2D image classification due to the divergent characteristics of the respective input data. While 3D point clouds are unstructured and sparse, 2D images are structured and dense. Bridging the domain gap between these two data types is a non-trivial challenge to enable model interchangeability. Recent research Lattice Point Classifier (LPC) highlights the feasibility of cross-domain applicability. However, the lattice projection operation in LPC generates 2D images with disconnected projected pixels. In this paper, we explore three distinct algorithms for mapping 3D point clouds into 2D images. Through extensive experiments, we thoroughly examine and analyze their performance and defense mechanisms. Leveraging current large foundation models, we scrutinize the feature disparities between regular 2D images and projected 2D images. The proposed approaches demonstrate superior accuracy and robustness against adversarial attacks. The generative model-based mapping algorithms yield regular 2D images, further minimizing the domain gap from regular 2D classification tasks.

5.1 Introduction

Point clouds consist of data points sampled through light detection and ranging (LiDAR) sensors, playing a crucial role in various 3D vision tasks. LiDAR is widely employed in applications like autonomous driving because it provides accurate 3D measurements. With the advancements in 2D image tasks [3; 34; 79; 174], 3D point clouds also witnessed rapid development, primarily driven by the integration of deep neural networks (DNNs). However, significant work is required to design modules to adapt 3D data for DNNs.

In recent years, several approaches have been proposed for 3D classification. *Multi-view-based methods* [58; 175] projects 3D objects into multiple 2D data, which can either be 2D images or 2D features from different views. Although this approach can directly apply 2D models, it inevitably loses depth information. Moreover, the generated 2D images are usually scattered pixels, deviating from normal 2D images. *Graph-based methods* [67; 176] employs graph representation to preserve spatial information like depth in 3D point cloud. Then graph convolution networks (GCNs) are utilized for end-to-end training. *Voxel-based methods* [11; 177] creates 3D voxels by dividing the space into occupancy grids. The features extracted in individual voxels are then gathered using 3D sparse convolutions. However, due to its computation complexity, its resolution is limited [178]. *Point-based methods* [54; 55; 56] extract point-wise features and aggregate them using customized modules. To effectively merge the traditionally separate branches of DNNs, it is essential to explore mapping algorithms that enable a seamless integration of 2D models.

Recent work proposed by Li *et al.* [175] showed that existing DNNs are vulnerable to adversarial attacks and proposed a new approach to 3D point cloud classification. Exploring correct attack algorithms [74; 76], perfect attack results can be achieved on almost all of the state-of-the-art classifiers. Since most 3D classifier applications are safety-critical, like autonomous driving, it is imperative to adopt a robust classifier under adversarial attacks. In this paper, we extend the lattice point classifier (LPC) in [175] and propose three additional mapping algorithms as shown in Figure 5.1. We design a suite of experiments to evaluate and analyze their performance and defense abilities.

Direct projection. Inspired by LPC [175] which projects the 3D point cloud onto a lattice hyperplane, we introduce a simpler version to analyze the performance. Like LPC, both algorithms project 3D points onto a hyperplane and generate sparse pixels on the 2D plane. However,

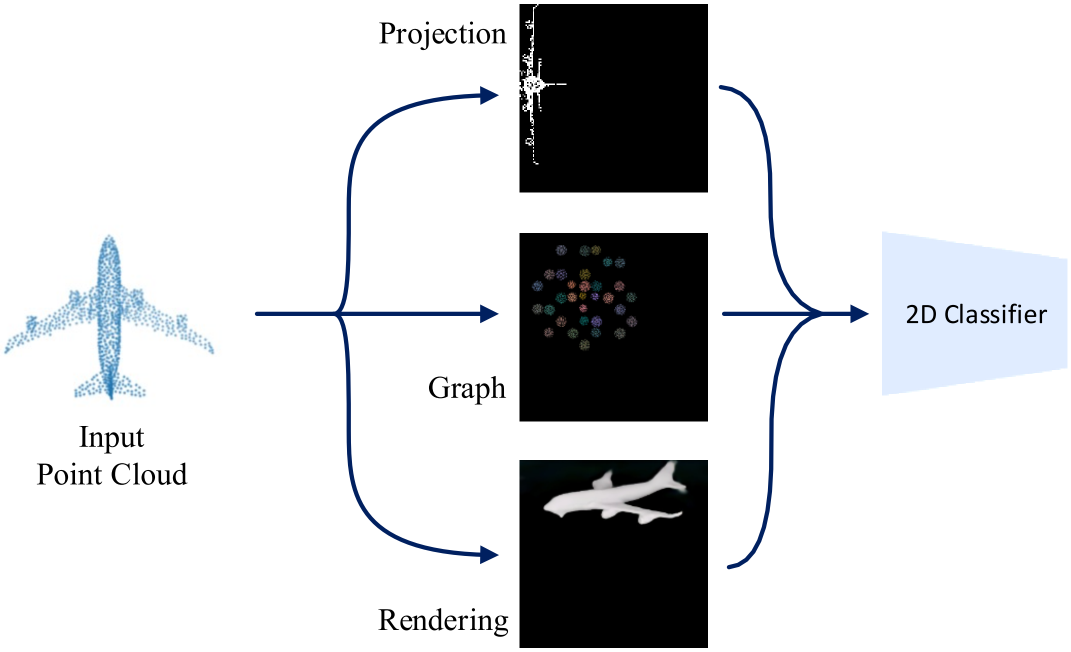


Figure 5.1: Illustration of mapping algorithms to transform 3D point clouds to 2D images.

these geometric projection methods will lose depth information. *Graph drawing.* Lyu *et al.* [6] proposed a point cloud part segmentation model based on an efficient hierarchical graph-drawing algorithm to represent 3D point clouds using 2D grids. The spatial relationship can be preserved by graphs. We extend the algorithm to further improve its classifier performance and analyze its defense mechanisms.

These mapping algorithms are inspired by multi-view and graph methods. The generated 2D images contain sparse projected pixels with the majority of the images filled with dark backgrounds, as shown in Fig. fig:main. To further align the generated 2D images with regular 2D images, we also introduce a generative model-based mapping algorithm to reduce the domain gap.

The main contributions are summarized below.

- We propose a family of structured declarative classifiers employing three distinct mapping algorithms. Through extensive experiments, we thoroughly analyze and enhance the performance of these classifiers. Our proposed classifier emerges as the top performer within the family of structured declarative classifiers, showcasing superior classification results.

- The proposed classifiers serve to narrow the gap between generated 2D images and real 2D images. We conduct experiments to showcase that our GAN-based classifier exhibits a minimal domain gap when compared to real 2D datasets.
- We analyze gradient propagation in declarative classifiers and investigate the defense mechanisms employed in the models. Basic projection and rendering classifiers exhibit strong resilience to attacks.

5.2 Related Work

A brief summary of recent works on 3D point cloud classification, 3D adversarial attacks and defenses can be found in Chapter 4.2. In the area of 3D defense, some newer approaches employ the diffusion model to perform point purification. Ada3Diff [179] estimates distortion based on points distance to their best-fitting plane. Li *et al.* [180] proposed a method using a graph network together with a data augmentation method to ensure robustness.

5.2.1 Graph Drawing

The goal of graph drawing is to represent original 3D point clouds using graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ and draw the graph on a low-dimensional space [181; 182; 6]. By different graph properties (e.g., spatial relationships among vertices and edges), graph representations can be divided into many families. For example, *k*-planar drawing [183] limits the number of edge crossings, and RAC drawings [184] will only have perpendicular crossings on their edges. Lyu *et al.* [6] proposed a Delaunary triangulation-based [7] graph drawing method for 3D segmentation. To directly apply normal 2D networks, a novel hierarchical approximation algorithm is designed to generate normal images from graph representation while preserving local information in point clouds.

5.2.2 Reconstruction and Rendering

Reconstruction surface from point clouds is an important and long-standing research direction in computer graphics, bridging the gap between points and images. A common approach is to form triangulations to compute a volume tetrahedralization [185; 186]. But these methods tend to create undesirable holes especially when points get noisy [8]. Implicit surface reconstruction [187; 188] can handle noise better at the cost of requiring more computational resources. Katz *et al.* [189] proposed a method to reconstruct the surface only from one single view using hidden point removal. The rapid development of DNNs shines light on the potential of bypassing manually designed priors. Hanocka *et al.* [190] proposed a DNN-based method to deform an initial mesh to shrink-wrap around an input point cloud. A differentiable surface splatting algorithm is proposed by Yifan *et al.* [191] to update point locations and normals. Z2P [8] proposed a method to view the rendering as a point-depth-map-to-image problem, which employs a modified U-Net [9] to render 2D images, achieving robust performance under noises and non-uniformly sampling.

5.2.3 Multi-modal Large Language Models

Multi-modal Large Language Models (MLLMs) are designed to achieve a broad understanding across various modalities, encompassing audio [192], image [14], point cloud [193], and more. The underlying idea is that incorporating language modality can enhance a model's ability to comprehend high-level interactions within diverse input data. Within this domain, one category of MLLMs employs language as a means to interact with other modalities [194; 195]. These models showcase versatile capabilities, particularly when provided with text inputs as prompts. Another category of MLLMs adopts individual encoders for each modality, aligning features from diverse modalities [14; 193]. These models demonstrate notable zero-shot generalization capabilities. Noteworthy examples include CLIP [14], which has inspired a myriad of novel applications. For instance, PointCLIP [196] leverages CLIP for zero-shot 3D multi-view classification, while Xue *et al.* [193] utilize CLIP's multi-modal features to supervise a 3D point cloud encoder, aligning three modalities. Given CLIP's proven robustness across multiple tasks, we leverage CLIP to

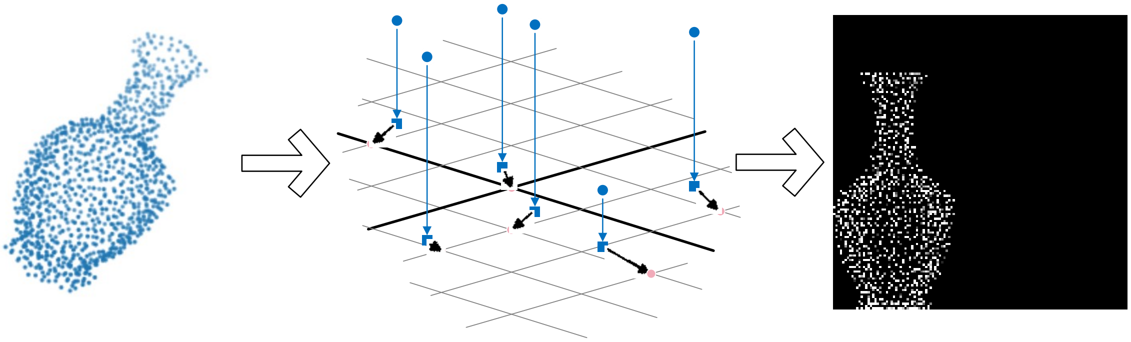


Figure 5.2: Projection Classifier. Input point clouds are projected onto the $x-y$ axis plane by removing z-axis information. The coordinates of projected points are rounded using the floor function.

assess the domain gap of generated 2D images when compared to typical 2D images.

5.3 Mapping Algorithms

The section describes the three mapping algorithms that can be used to map 3D point clouds into 2D space.

5.3.1 Basic Projection Classifier

The Lattice Point Classifier (LPC) was introduced as an implementation of a robust declarative classifier (RDC) with robust defense capabilities against gradient-based attacks [175]. However, LPC involves a sophisticated mapping algorithm. It first projects 3D point clouds onto a lattice hyperplane [146], which is populated with a triangular lattice. Subsequently, each projected point is splatted onto the three enclosing triangular lattice vertices. In this context, we delve into a more straightforward mapping algorithm for 3D-to-2D projection by simply dropping one dimension of the data as shown in Figure 5.2. This basic projection serves as a baseline for performance comparison, and the defensive behavior of RDC is scrutinized in conjunction with this basic projection.

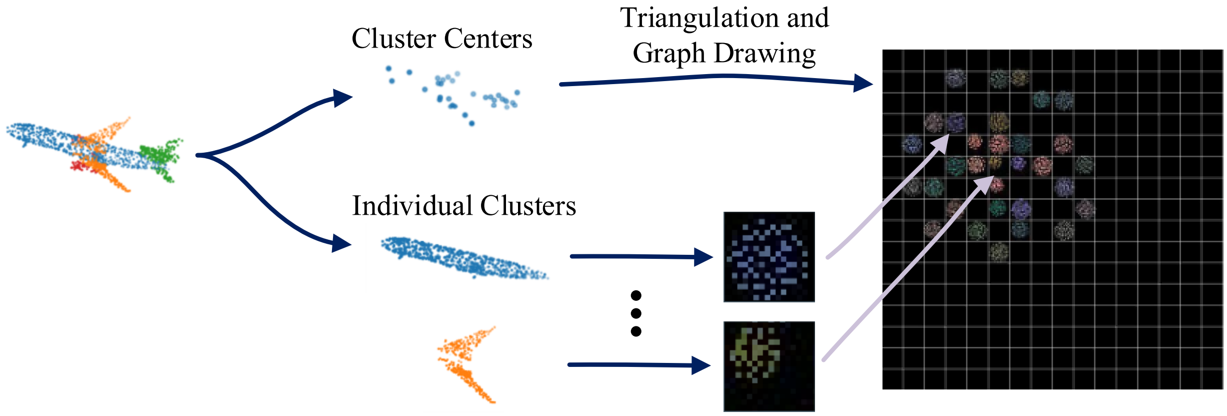


Figure 5.3: Graph Drawing Classifier. Input point clouds are divided into 32 clusters using balanced KMeans clustering [6]. Delaunay triangulation [7] is applied on the 32 cluster centers and within each individual cluster, generating 2-level graphs. Then the top-level graph is mapped to a 16×16 grid, where each cluster center occupies a grid cell. Then each grid cell is filled with a lower-level 16×16 grid, obtained from within-cluster graphs, yielding the final 256×256 image.

5.3.2 Graph Drawing Classifier

Both basic projection and permutohedral lattice projection unavoidably lead to the loss of depth information perpendicular to the 2D hyperplanes. In contrast, graph representations showcase the capability to preserve 3D spatial information, rendering them an optimal choice for the mapping algorithm.

Given the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}) = h(\mathbf{X})$, where $\mathbf{X} = \{x_i \in \mathbb{R}^3\}$ denotes input point cloud, the function $h : \mathbf{X} \to \mathscr{G}$ transform input point clouds to graph representations. Graph drawing is originally employed to transform a graph $\mathscr{G}$ into continuous drawings. To facilitate the utilization of standard 2D neural networks, we want to find a function $f : \mathscr{G} \to \mathbb{Z}^2$ that converts graphs to 2D integer coordinates. Inspired by the graph drawing segmentation model [6], we adopt an efficient graph drawing classifier. This method reduces computational costs by creating two hierarchical levels of graphs and incorporating balanced KMeans clustering. The outlined approach is illustrated in Fig. 5.3.

Two-level graph drawing. The computational complexity of Delaunay triangulation is $\mathscr{O}(n^{\lceil d/2 \rceil})$ [197], where n is the number of data points and d is the dimension of each data point. For 3D point

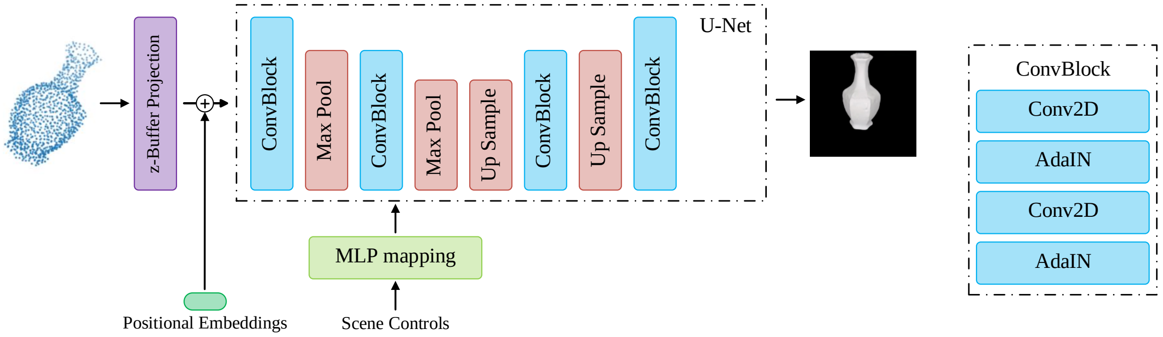


Figure 5.4: Rendering Classifier. Input point clouds are converted to a 2D depth map using z-buffer projection. Positional embeddings are appended to the depth map to boost global context [8]. Then the generated features are fed to a modified U-Net [9] to produce final rendered images. In the modified U-Net, the normalization layer is replaced with a style-based adaptive instance normalization (AdaIN) [10], which also takes scene control features to influence output image style.

cloud, the complexity becomes $\mathcal{O}(n^2)$. Thus, instead of directly applying graph drawing on all the 3D points $\mathbf{X}$, we employ a two-level method that divides $\mathbf{X}$ into 32 clusters and creates a two-level hierarchical data, $\mathbf{C} = \{\mu_1, \mu_1, \cdots, \mu_{32}\}$ and $\mathbf{S} = \{S_i\}$, where $S_i = \{x_j \in \mathbb{R}^3\}$ and μ_i is the center of cluster S_i. The Delaunay triangulation [7] is then applied to the point set $\mathbf{C}$ and S_i. This approach will reduce computational complexity.

5.3.3 Rendering Classifier

As shown in Fig. 5.1, the images generated from the above mapping algorithm still have distinctive visual artifacts. The rendering-based method can generate realistic 2D images by reconstructing object surfaces, which can further reduce the domain gap from regular 2D tasks. Z2P [8] is a lightweight DNN-based rendering algorithm. The pipeline of the mapping is shown in Fig. 5.4. It first generates a z-buffer projection from the original point cloud, and then renders the 2D image through a modified U-Net.

z-buffer Projection. Like the basic projection method, it projects points onto a 2D hyperplane, obtaining a pixel-point pair $(x_k, I(i, j))$, where x_k is a 3D point and $I(i, j)$ is the corresponding pixel value at image coordinate (i, j). The pixel value is determined by the distance of the 3D point from the image plane d. The pixel value can be calculated by $I(i, j) = e^{-(d-\alpha)/\beta}$, meaning the farther the points are, the darker they appear on the depth image. The pixel value is shared within a 3×3 pixel window centered at $I(i, j)$. The pixels without a corresponding point are given a value 0.

Adaptive Instance Normalization (AdaIN). To enhance control over the colors and shadows of the rendered images, we replace the original normalization layers with AdaIN [10]. In addition to the normal DNN features $\mathbf{f}$ of size $H \times W \times C$, AdaIN takes a scene-control vector w as input. Each AdaIN layer contains two affine transformations, denoted as A_s and A_b, which are utilized to compute styles $y_s = A_s \cdot w$ and $y_b = A_b \cdot w$. The output of AdaIN is determined by

$$\text{AdaIN}(f_i, w) = y_{s,i} \frac{f_i - \mu(f_i)}{\sigma(f_i)} + y_{b,i} \tag{5.1}$$

where $\mu(f_i)$ and $\sigma(f_i)$ are the mean and standard deviation (std) at each feature channel f_i.

5.4 Experiments

5.4.1 Dataset

ModelNet40 [11]: The dataset comprises clean synthetic 3D objects generated from CAD models, encompassing 40 distinct categories, with a total of 12,311 objects. These objects are split into 9,843 training samples and 2,468 testing samples. In our experimental setup, 3D points are uniformly sampled from the mesh surface, adhering to a specified methodology in [54; 74].

5.4.2 Training Settings

We perform classification training on the ModelNet40 dataset. For all training tasks, we run 200 training epochs using Adam optimizer [169] with 0.0001 decay rate and $(0.9, 0.999)$ beta values. A simple step learning rate scheduler is used with a step size of 20 and a starting learning rate of 0.001 unless otherwise specified. The point clouds are evenly sampled from the object surface following the practice in [54]. During training, the points are augmented by random point dropout, scale, and shift. The max dropout rate is 0.875. The random scale rate is set from 0.8 to 1.0. The max random shift distance is 0.1. When random rotation is enabled, the max random rotation angle is π. No augmentation techniques are applied during the inference.

Basic Projection Classifier. We scale the point clouds to generate 456×456 images. During data augmentation, points projected outside the image will be set to pixel $I(0,0)$.

Graph Drawing Classifier. Using the balanced KMeans [6], the original input point clouds are divided into $K = 32$ clusters. The balanced KMeans clustering algorithm initially generates unbalanced clusters h, and then starts to reduce the number of points in oversize clusters. A cluster is considered oversized when the number of points $|h| > \alpha \cdot |\mathbf{X}|/K$ where $|\mathbf{X}|$ is the total number of points in input point clouds, and α is set to 1.2. For the graph drawing, both the lower-level and top-level grid sizes are 16×16, making the final graph drawing output to be 256×256.

Rendering Classifier. The classifier undergoes a two-step training process. In the first step, the rendering model is trained for 10 epochs with a learning rate of 3×10^{-4}. Throughout the z-buffer projection, the splat window size is set to 3. During inference for scene control, the generated object color RGB values are fixed at $(255, 255, 255)$, and lighting is positioned at the origin point for simplicity. The resolution of the generated rendered 2D image is set to 313×313. The rendering training data is generated following [8] using [198].

5.4.3 Domain Gap

A crucial factor for the application of regular 2D classifiers is minimizing domain gaps between generated and regular images. To quantify this domain gap, we leverage well-established foundation models [14; 199] that have been trained on extensive general domain data with over 15 million images. These models have demonstrated notable zero-shot classification accuracy across diverse datasets. For instance, CLIP [14] achieves a 76.2% zero-shot classification accuracy on the extensive ImageNet dataset [103]. The inherent alignment of text-image features in these pre-trained

Table 5.1: Zero-shot classification accuracy (%) using CLIP [14]. Text inputs are a list of the ModelNet40 [11] class names. Image inputs are 2D images generated by three tested projection algorithms.

	Instance Accuracy	Class Accuracy
LPC [175]	2.64	3.75
Basic Projection	5.17	5.43
Graph Drawing	2.32	1.95
Rendering	**27.08**	**23.78**

models simplifies their adaptation to diverse classification tasks across datasets with varying categories.

We use CLIP [14] to run zero-shot classification on the generated 2D image from LPC [175] and our three declarative classifiers. The image backbone is ViT-B/32 [24] which takes images of resolution 224×224. We directly utilize CLIP's image normalization mean and std, $(0.4814, 0.4578, 0.4082)$ and $(0.2686, 0.2613, 0.2758)$, for fair comparison. The text inputs are the category names of each ModelNet40 class, ["'airplane", "bathtub", "bed", ..., "wardrobe", "xbox"].

5.4.3.1 CLIP Zero-shot Accuracy

The zero-shot classification results are shown in Table 5.1. We have three observations from the results. (i) Graph drawing 2D images have the lowest accuracy, hovering around the same level of total random prediction accuracy $\frac{1}{NUM\_CLASS} = 1/40 = 2.5\%$. It is not surprising when we take a look at the visualization of graph drawing images depicted in Fig. 5.6. However, considering that the primary goal of graph representation is to encode point clouds into latent features rather than human-understandable images, this approach remains promising, as evidenced by the 85.58% classification accuracy. (ii) The two projection-based methods (LPC and basic projection classifier) still demonstrate a substantial domain gap. The sparse enabled pixels in Fig. 5.6 are different from regular dense images. However, the accuracy difference between LPC (2.64%) and basic projection (5.17%) indicates that point of view can significantly affect classifier performance. This

Table 5.2: Classification accuracy (%) using test projection algorithms.

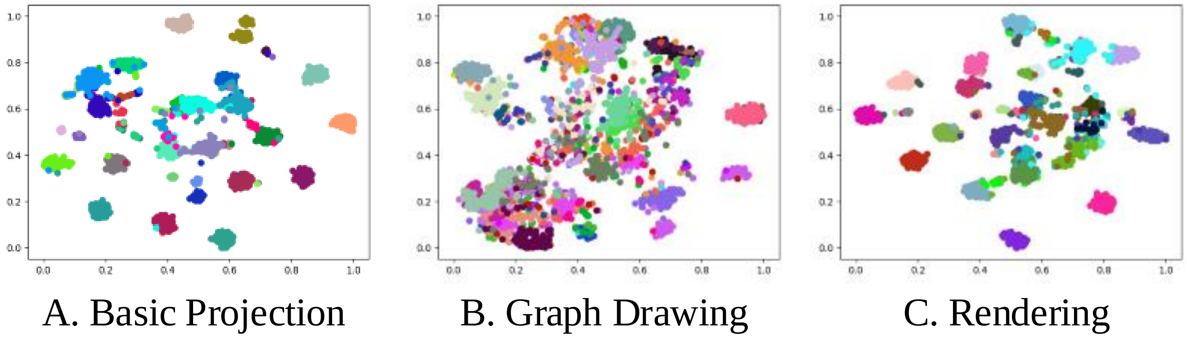
	Instance Accuracy	Class Accuracy
LPC [175]	89.51	86.30
Basic Projection	**91.02**	**88.42**
Graph Drawing	84.97	81.21
Rendering	88.30	85.78

point is further validated in subsequent experiments on rendering classification in the ablation study. (iii) Despite rendering classifier 2D images achieving a noteworthy reduction in domain gap compared to other mapping algorithms, other classifiers do not achieve satisfactory CLIP zero-shot performance as shown in Table 5.1.

5.4.4 Classification Accuracy

For all four classification models, we employ EfficientNet-B5 [167] as the 2D backbone networks, and the results are summarized in Table 5.2. The basic projection model achieves the highest test accuracy at 91.02%, outperforming LPC [175]. As visualized in Fig. 5.6, LPC and basic projection essentially represent high-to-low-dimensional projections from different viewpoints, which significantly influences the classification performance. This observation is further validated in subsequent ablation studies. The performance of graph drawing demonstrates that DNNs can be trained to learn distinct data representations without any modifications. Despite generating more realistic 2D images, the surprising outcome is that rendering falls behind the two projection methods. We conduct a more in-depth analysis of the model's performance through t-SNE visualization [12] in subsequent sections.

t-SNE [12] is a technique to visualize high-dimensional data to two or three-dimensional data, offering insights into data separation. In our analysis, we input feature vectors from each mapping algorithm, obtained before linear layers, into the t-SNE algorithm. The resulting visualization is depicted in Fig. 5.5. It is evident that graph drawing's clusters are more ambiguous, with less sepa-



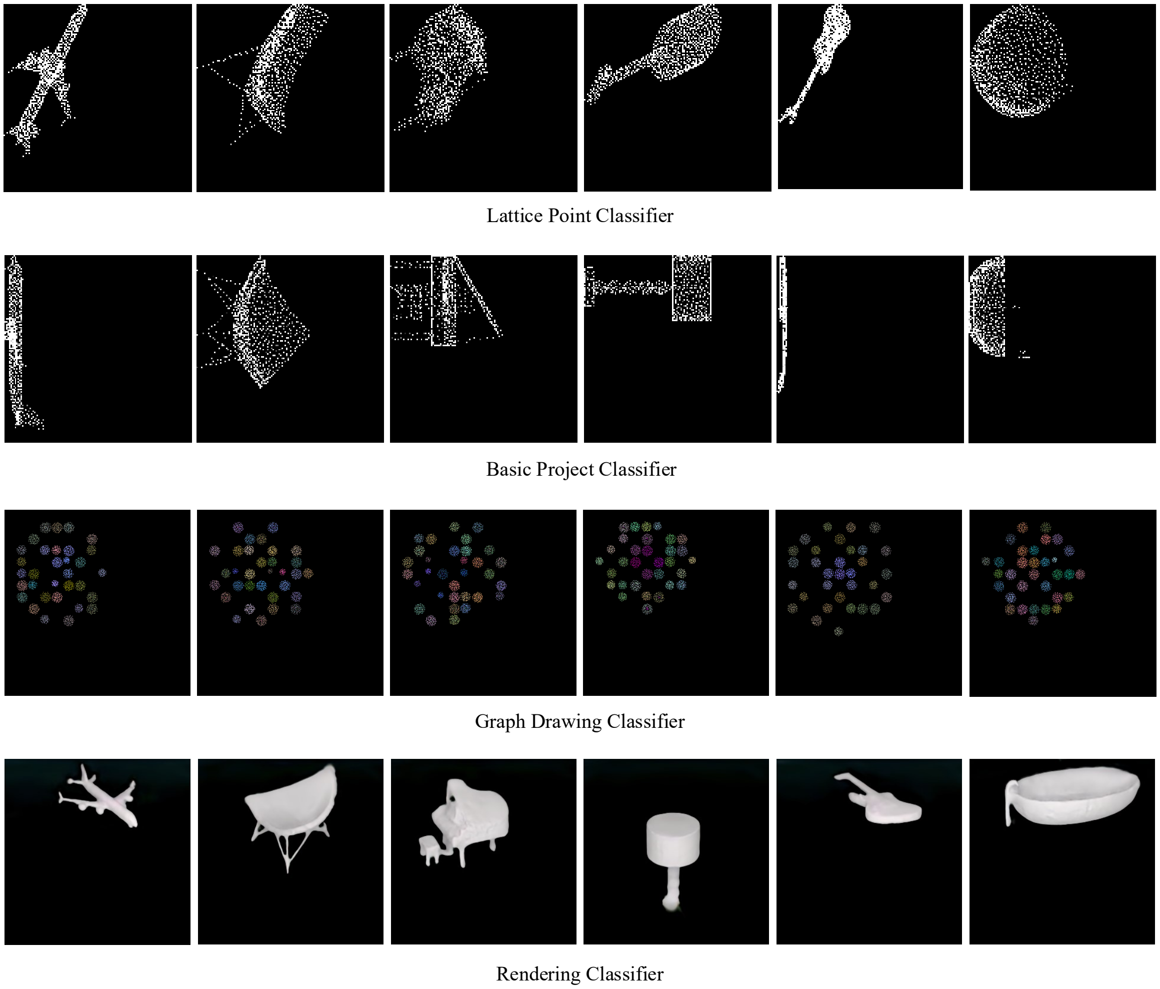
A. Basic Projection B. Graph Drawing C. Rendering

Figure 5.5: Visualization of 2,468 ModelNet40 [11] test point clouds using t-SNE [12]. Each plot is based on the DNN features generated by corresponding mapping algorithms before linear layers.

ration and blurred boundaries. Comparatively, upon closer inspection, the t-SNE representation for basic projection reveals more defined clusters. Each class forms a smaller, more separated cluster compared to the graph generated by rendering. Moreover, in the central region of the t-SNE visualizations, rendering exhibits more ambiguous data points, providing an explanation for its lower accuracy.

5.4.5 Ablation Study

Rendering Classifier. We evaluate the contribution of different setups on our rendering classifier. Many applications have proven the power of foundation models trained on large amounts of data [14; 200; 201; 202]. We employ two different CLIP [14] image encoders (ViT [24] and ResNet50 [3]), which produce 768 channel and 1024 channel image features. We append 3 fully connected layers after the foundation image encoders, with 4096 as the fully connected layers' hidden channel dimension. The ablation results can be found in Table 5.3.



Lattice Point Classifier

Basic Project Classifier

Graph Drawing Classifier

Rendering Classifier

Figure 5.6: Visualization of 2D images generated by LPC, basic projection, graph drawing, and rendering classifiers.

5.4.5.1 Learnable Backbone Weights

For the foundation model encoders, we run training with the encoder weights frozen or trainable to study the effects. From the foundation ViT experiments (1 and 2) and ResNet50 experiments (2 and 3), we can see making the foundation encoder learnable will reduce the accuracy. Specifically, the accuracy decreases by 2% to 76.53% for ViT and 10% to 56.81% for ResNet50. Examining the training data accuracy, we observe that the frozen and learnable backbones using ResNet50 have 58.93% and 56.38% accuracy respectively. The small training accuracy gap indicates learnable foundation encoder tends to overfit the data.

Table 5.3: Ablation study for rendering classification. The table shows the inference accuracy (%) with different setups. The mean and std of the rendered normalization are calculated from the training set of rendered 2D images. When upright is not checked, the model is trained with default point cloud orientation.

Backbone	Learnable Backbone	Rendered Normalization	Upright	Random Rotation	Instance Accuracy	Class Accuracy
ViT-1					78.53	74.64
ViT-2	✓				76.53	71.53
ViT-3		✓			77.46	73.53
ResNet50-1					67.50	60.15
ResNet50-2		✓			67.06	59.59
ResNet50-3	✓	✓			56.81	49.86
EfficientNet-B5	✓				83.65	79.31
EfficientNet-B5	✓	✓			84.29	80.54
EfficientNet-B5	✓		✓		86.61	82.86
EfficientNet-B5	✓	✓	✓		86.90	83.70
EfficientNet-B5	✓	✓	✓	✓	**88.30**	**85.78**

5.4.5.2 Normalization

We conducted a comparison between different normalizations: CIFAR-100 normalization and rendered image normalization with mean values of (0.1136, 0.1135, 0.1128) and standard deviations of (0.2729, 0.2719, 0.2761) during training. Utilizing EfficientNet-B5 [167], we observed an improvement in accuracy by 0.64% and 0.29%. However, this conclusion does not extend to foundation encoders. Specifically, with the foundation ViT backbone, the calculated normalization results in a decrease in accuracy by 1.07%. We suggest that foundation models might have a sufficient volume of data, allowing them to adapt to various data distributions.

5.4.5.3 Observation View Point

As discussed in Section 5.4.3.1, the viewpoint might have a significant impact on classification accuracy. As shown in Table 5.3, rotating object upright significantly improves classification accuracy. When CIFAR-100 normalization is used, the inference accuracy is boosted by 2.96%, and with accurate normalization, the improvement is 2.61%.

Table 5.4: Defense Performance (%).

	Accuracy		Attack Success Rate
	No Attack	FGSM [152]	
PointNet [54]	90.15	45.99	48.99
LPC [175]	89.51	89.51	**0.00**
Basic Projection	**91.02**	**91.02**	**0.00**
Graph Drawing	84.97	24.76	70.86
Rendering	88.30	88.30	**0.00**

5.4.5.4 Random Rotation

The rendering classifier training is carried out in three steps: initially to train the rendering model, then to generate rendered images, and finally training on 2D images to save time. Consequently, during the training of the 2D image classifier, it only has exposure to the 3D object from a single viewpoint. To enhance the dataset, four additional sets of rendered images are created through the rotation of the original input point clouds. Specifically, we rotate $\pm\frac{\pi}{9}$ along x-axis and $\pm\frac{\pi}{9}$ along z-axis separately to generate the four sets of rendered images. This data augmentation strategy contributes to an increase in inference accuracy by 1.4%.

5.4.6 Defense Experiment Settings

To test the defense performance of each mapping algorithm, we run FGSM [152] similar as LPC [175] under various gradient-based attacks. The FGSM attack is applied on the entire test set from ModelNet40 [11]. The attack learning rate of FGSM is set to 0.1.

5.4.7 Defense Performance

The attack results are listed in Table 5.4. From the results, we can observe: (i) Basic projection and rendering classifier exhibit similar defense behavior as LPC [175]. They are all robust under gradient-based attackers. (ii) Not all robust declarative classifiers are guaranteed to be robust under

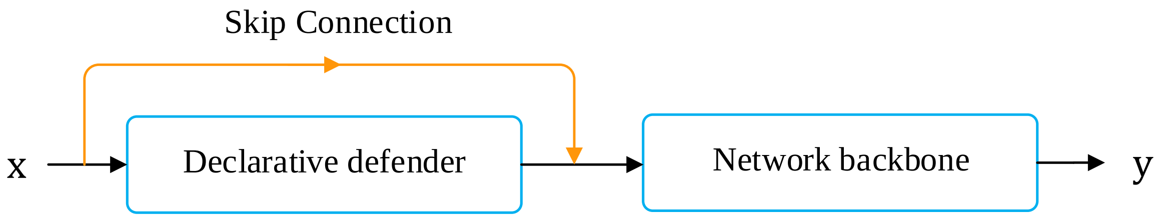


Figure 5.7: Gradient skip connection. The normal declarative classifier is the module without the orange skip connection. Declarative defenders will block back-propagation using implicit gradients. In graph drawing, 3D point coordinates are directly assigned as image intensities, creating a skip connection and bypassing declarative defender.

attackers. Graph drawing classifier is still vulnerable to adversarial attacks.

We designed more experiments to further investigate the factors that influence defense performance. An illustration of the original declarative classifier can be found in Fig. 5.7 without the orange connection. The classifier can be formulated as

$$y = g(h(\mathbf{x}); \omega) \tag{5.2}$$

where $h(\mathbf{x}) = \arg\min_{z \in \mathscr{Z}} f(x, z; \theta)$, $g(\tilde{\mathbf{x}}; \omega)$ is the network backbone, which takes the generated images $\tilde{\mathbf{x}} = h(\mathbf{x})$ as input, and $h(\mathbf{x})$ is the declarative defender. In a gradient-based attack, the attack gradient can be easily back-propagated through $g(\tilde{\mathbf{x}}; \omega)$. The declarative defender $h(\mathbf{x})$ will block gradient propagation using implicit gradients.

In the context of graph drawing, a particular operation functions as a skip connection, compromising the defense capability. Each point in 3D space is mapped to a corresponding pixel in the resulting 2D image, and the coordinates of each point are then assigned as pixel intensities. This establishment of a skip connection circumvents the declarative defense mechanism, as illustrated in Fig. 5.7. To validate this hypothesis, we conducted experiments where we leaked 3D coordinates as pixel intensity in the generated 2D images for the basic projection classifier. The attack success rate increased to 35.7%, effectively defeating the defense capability.

5.5 Conclusion

In this paper, we have investigated three distinct mapping algorithms as declarative defenders for 3D point cloud classification, demonstrating the robust performance of declarative nodes through diverse 3D-to-2D mapping techniques. The majority of the proposed declarative defenders exhibit resilience against adversarial attacks. Through the use of a rendering classifier, we discover that the observation viewpoint and random rotation can substantially impact classification accuracy. Notably, upon observing the diminishing defense capability in the graph drawing classifier, we conclude that the presence of a skip connection circumvents implicit gradients, thereby defeating the defense capability.

There is room for improvement in the rendering classifier. Currently, the scene control parameters are manually set, mainly due to the scarcity of training data with actual object textures. A potential avenue for future work could involve designing a module to learn texture priors from the point cloud. This approach aims to generate more realistic 2D images by incorporating learned texture information.

Chapter 6

Polyps Dataset Creation and Evaluation

Abstract

Colorectal cancer (CRC) is one of the most common types of cancer with a high mortality rate. Colonoscopy is the preferred procedure for CRC screening and has proven to be effective in reducing CRC mortality. Thus, a reliable computer-aided polyp detection and classification system can significantly increase the effectiveness of colonoscopy. In this paper, we create an endoscopic dataset collected from various sources and annotate the ground truth of polyp location and classification results with the help of experienced gastroenterologists. The dataset can serve as a benchmark platform to train and evaluate the machine learning models for polyp classification. We have also compared the performance of eight state-of-the-art deep learning-based object detection models. The results demonstrate that deep CNN models are promising in CRC screening. This work can serve as a baseline for future research in polyp detection and classification.

6.1 Motivation

Colorectal cancer (CRC) is one of the most common cancers diagnosed throughout the world [203; 204]. From the data of both sexes combined, CRC contributes to 10.2% of all cancer cases in 2018 as the third most common cancer, following lung cancer (11.6%) and breast cancer (11.6%) [205]. It is the second deadliest cancer in terms of mortality causing 9.2% of the total cancer deaths [205]. According to the statistics [204], both male and female are almost affected equally. Nevertheless, despite the high incidence and mortality rates, the deaths caused by CRC have been decreasing with an accelerating decline rate since 1980 for both men and women [206]. This trend mainly reflects the progress achieved in early detection and treatment.

Early detection plays a significant role in fighting CRC. It not only brings down the mortality but also prevents excessive treatment costs by diagnosing before CRC spreads to distant organs [207]. According to [204], the stages at which the disease is diagnosed highly correlate to survival, with a 90% 5-year survival rate for the localized stage, 70% for the regional stage, and 10% for distant metastatic cancer. Another reason we should rely on early detection is due to the nature of the symptoms and development of CRC. Although no symptoms can be easily observed before the tumor reaches a certain size (typically several centimeters) [207], it would typically take several years to as long as a decade for CRC to develop [208], starting from precancerous polyps. Both facts add up to show the significance and potential of diagnosing CRC by regular screening at an early stage, even before polyps become cancerous.

6.1.1 CRC screening options

There are several common CRC screening options, which can be roughly divided into two categories: visual examinations and stool-based tests. Each method has its advantages and limitations. The evaluation needs to take into account a broad range of factors including statistical data and psychological effects. The most important metric, like many other screening tests, is ‘sensitivity’ [207], which is also called ‘recall’ in some other fields, determined as the percentage of patients with the disease that is actually detected. From sensitivity, we know the possibility of a patient walking out of the clinic with lesion undetected, the consequences of which are severe. Therefore, in many instances, it is the single most important metric to optimize.

Another statistical measure that often comes along with sensitivity is ‘specificity’, which is measured as the fraction of healthy people that are correctly identified. It indicates the potential of a test to falsely detect lesions in healthy clients. This will cause mental stress on the clients, and the following treatment might result in unnecessary physical harm and financial burdens. Thus, a high-specificity test is also preferred. For a screening method in real clinical settings, there is generally a trade-off between the sensitivity and the specificity. With the consequences of missing a lesion much more grave than a false diagnosis, sensitivity is usually preferred over specificity. A higher specificity screening can always follow a high sensitivity test to filter out the falsely diagnosed cases [207]. Other factors include how easy the preparation is, how accessible the facility is, how much the test costs, etc. Since individuals who need to be screened are oftentimes asymptomatic, the experience will affect their compliance, which is an important part of an effective screening program [207]. In the following session, some common CRC screen methods and their properties are discussed.

Colonoscopy is the recommended CRC visual examination screening method. The advantages of colonoscopy include high sensitivity, the ability to remove lesions at detection and full access to proximal and distal portions of the colon [207]. The colonoscopy can reach a sensitivity of 95% in detecting CRC according to Rex *et al.* [209]. The disadvantages are mostly related to the way colonoscopy is conducted [210; 207]. At least one day before the test, it requires a complicated bowel preparation, which requires the participant to change diet and take medicine to cause diarrhea. During the test, sedation or anesthesia might be performed, and there is a risk of post colonoscopy bleeding. Thus, the suggested 10-year screening interval has a low compliance rate [207]. Narrow-Band Imaging (NBI) is a newly developed technique by modifying light source using optical filters in an endoscope system [211]. Compared to normal colonoscopy, intensified lights of a certain wavelength can better present the mucosal morphology and vascular pattern [212]. Studies show that NBI performs better in CRC detection than conventional colonoscopy [211; 212].

Computed Tomography (CT) Colonoscopy is a structural radiologic examination that employs software to reconstruct 3D views of the entire colon to detect lesions. Although it has a slightly less sensitivity of $> 90\%$, the less-invasive nature of CT colonoscopy results in a higher participation rate [213; 214]. The limitations include unpleasant bowel preparation before the test, uncomfortable inflation of the colon with air during the test, and safety concerns over the use of radiation. Compared to colonoscopy, CT colonoscopy is not studied thoroughly, *e.g.*, uncertain screening interval [207]. With the fact that CT colonoscopy requires follow-up colonoscopy with lesion detected and its sensitivity highly dependant on radiologists' expertise, it is only recommended to individuals whose physical conditions are not fit for the invasive examination of the colon [207]. A similar screening method, double-contrast Barium Enema, is also not recommended due to similar limitations and even more complicated procedures [215].

Sigmoidoscopy is similar to colonoscopy, but it can only access the distal part of the colon. It shares the same high sensitivity as colonoscopy and can remove lesions at the detection. In addition, it requires less complicated bowel preparation and usually does not need sedation [215]. However, sigmoidoscopy has limited accessibility to only the distal colon rather than the proximal part, making it less effective due to the higher risk of proximal CRC among elder individuals and women [215]. Therefore, it is recommended to pair sigmoidoscopy with other screen methods [207].

Wireless Capsule Endoscopy uses a miniaturized camera in a swallowable capsule to transmit gastrointestinal images to portable receiver units that can be easily worn [216]. Although a typical examination takes about 7 hours [217], the process does not impact patients' life quality compared to other methods. This wireless capsule can also examine the entire small bowel that is not accessible to other endoscopy practices [217]. Nevertheless, wireless capsule endoscopy has some drawbacks as well. For example, it has no therapeutic capability [217]. Also, it does not take images in distended bowel as other methods [217], the practitioners need training to interpret the images.

Fecal Occult Blood Test (FOBT) and Fecal Immunochemical Test (FIT) both detect hemoglobin in the stool to indicate if a lesion exists. Both tests are non-invasive and easy to carry out even at home, but their sensitivities suffer for earlier stages of lesions due to less frequent bleeding [207]. In addition, some dietary intakes can alter the test results, reducing the performance of FOBT and FIT.

There are other screening tests like the DNA test, wireless capsule endoscopy, etc. However, due to low sensitivity and lack of sufficient supportive studies, they normally need a subsequent colonoscopy when the result is positive.

6.1.2 Goals

As the reference CRC screening test, colonoscopy has obvious advantages over its alternatives. However, its performance depends on several variables, like the bowel preparation, the number of polyps, and the part of the colon where the polyps are located [218; 219; 220]. Furthermore, human factors can influence screening sensitivity and specificity. Inexperienced gastroenterologists have higher miss-detection rates compared to those who are well-trained. According to Leufkens *et al.* [219], participants before training showed significantly lower performance than post-training results. Colonoscopy is also subjected to the physical and mental fatigue of the gastroenterologists. The screening process requires prolonged concentration and is usually repeated throughout the day. A study by Chan *et al.* [218] showed that 20% more polyps are detected from early morning screenings.

It is obvious that a fine-grained deep learning framework to automatically detect polyps is needed to help physicians locate and classify the lesions. This deep learning framework can assist physicians during screening in real-time and prompt the detected region and polyp category. Thus, such a computer-aided system can help eliminate the miss rate due to physical and mental fatigue and allow the gastroenterologists to focus on regions where lesions actually exist. This automated system also ensures high performance in clinics where access to experienced gastroenterologists is difficult. An accurate detection system can also improve the detection rate of smaller pre-cancerous polyps using the Convolutional Neural Network (CNN) models. The sensitivity of current colonoscopy suffers as the size of the colon becomes smaller [215; 208; 207]. This can be improved because the state-of-the-art CNN models can extract features from objects at different scales.

Deep learning models require larger datasets to exploit their full potential. Recent benchmark datasets for general computer vision tasks all have more than 10*k* images [221]. We want to build a polyp classification dataset based on the videos from the colonoscopy procedure with a reasonable number of samples to train deep neural network models. The images in the dataset contain polyps from different stages and are representative of different types of polyps. We will label each frame with accurate polyp locations and categories. Although constructing such a dataset is time-consuming and labor-intensive, it will benefit the research community to develop more accurate and robust deep learning models to achieve a higher detection rate and to reduce CRC mortality rate. The dataset could also standardize and facilitate the training of medical professionals in endoscopy.

Using the developed dataset, we have evaluated and compared the performance of the state-of-the-art deep learning models for polyp detection and classification. The dataset and the corresponding annotations can be downloaded via *https://doi.org/10.7910/DVN/FCBUOR*.

6.2 Related Work

Deep learning has achieved more and more attention in recent years with wide applications across a variety of areas. It boosts the performance by a significant margin in tasks like computer vision, speech recognition, natural language processing, data analysis, etc. [79; 2; 18; 27; 222; 174]. The success is largely owing to the development of deep Convolutional Neural Networks (CNN) which have been proven to be especially effective in extracting high-level features. Among all these areas, deep learning has achieved huge success in computer vision applications, with early CNN models almost halving the error rate in the ImageNet classification challenge compared to classic models [2]. In recent years, CNN-based models have demonstrated their outstanding capabilities in many complicated vision tasks, like object detection, image segmentation, object tracking, etc. [82; 114; 223; 224; 110].

6.2.1 Computer Vision in Medical Applications

Researchers have been trying to use computer vision techniques in medical applications as early as 1970 [225]. At that time, image processing was only a low-level task like edge finding and basic shape fitting. As the handcrafted models became more sophisticated, some studies showed success in areas like salient object detection and segmentation [226; 227]. The ability of these models to analyze the surface pattern and appearance prompts their application in a wide range of medical fields, such as neuro, retinal, digital pathology, cardiac, and abdominal [225]. Bernal *et al.* [228] proposed a model that considers polyps as protruding surfaces and utilizes valley information along with completeness, robustness against spurious responses, continuity, and concavity boundary constraints to generate energy map related to the likelihood of polyp presence. In the study [229], the model exploits the color feature extraction scheme based on wavelet decomposition and then uses linear discriminant analysis to classify the region of interest. Other handcrafted feature approaches can be found in [230].

The limiting factor of hand-engineering models is the need for researchers to understand and design filters. They tend to perform better for low-level features. Deep learning models can automatically generate parameters with deeper layers and extract high-level semantic features. Especially in recent years, many new models [231; 232] and techniques [233; 107; 105; 108] have been published to set new records in various computer vision tasks. [234] employs multi-scale architecture with 3 layers of CNN and 3 layers of max-pooling followed by fully connected layers. Another model uses a slightly different approach using 3 different extracted features, color and texture clues, temporal features, and shape to feed an ensemble of 3 CNN models [235]. Deep learning models have been widely applied to medical problems like anatomical classification, lesion detection, and polyp detection and classification in colonoscopy [236; 237; 238; 239; 240; 241]. In [236], Six classical image classification models have been compared to determine the categories of detected polyps. It assumes all polyps have been detected and cropped out from the original sequences. An enhanced U-Net structure has been proposed in [242] for polyp segmentation. In this paper, we focus on polyp detection from the endoscopic sequences to assist gastroenterologists in both polyp detection and classification. We evaluate and benchmark the state-of-the-art detection models for colonoscopy images.

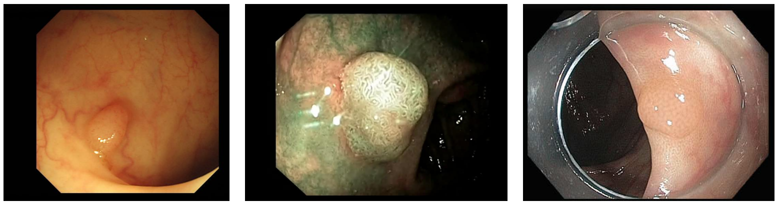
6.2.2 Object Detection

Different computer vision techniques can be adapted to perform polyp detection, such as object detection, segmentation, and tracking. We will use the eight state-of-the-art object detection models (Faster-RCNN [38], Faster-RCNN [38] with DetNet [4] as backbone, YOLOv3 [41], YOLOv4 [44], SSD [49], RetinaNet [50], RefineDet [51] and ATSS [52]) introduced in Section 1.1 to evaluate their performance on our dataset in this comparative study.

6.3 Dataset Build

The performance of a CNN model is highly dependent on the dataset. During training, a CNN model learns from a large number of examples how to extract semantic features, on which localization and classification are based. Therefore, CNN detectors perform better when the dataset consists of representative examples of all categories. For example, images that are taken from different viewpoints, various illumination conditions, multiple sizes, etc. The more representative the dataset is, the more likely the CNN models can learn meaningful features for detection and classification. Then, at the inference time, the trained CNN models will have a higher ability to generalize the feature extraction on new input images.

In the research community, there are several small collections of endoscopic video datasets for different research purposes, such as MICCAI 2017, Gastrointestinal Lesions in Regular Colonoscopy Data Set (GLRC) [243] and CVC colon DB [228] dataset. However, after careful observation and analysis, we found that these datasets differ greatly from each other in terms of resolution and color temperature, as shown in Figure 6.1. This is largely due to the setups and characteristics of different imaging equipment used for data collection. As pointed out in [244], two of the main reasons why current CNN models perform worse in the real world compared to benchmark test sets are the variance in image backgrounds and image quality. As shown in Figure 6.1, the images in different datasets vary greatly. If we train the models using only one of these datasets, the models may have poor generalization ability, and their performance will suffer when being applied to



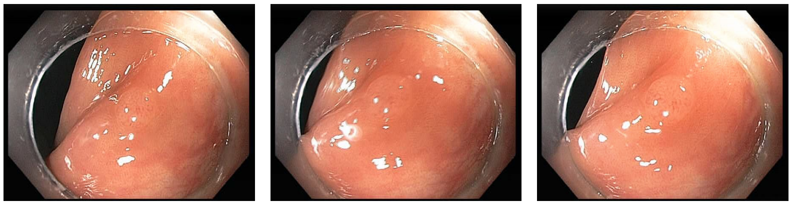
(a) MICCAI (b) GLRC (c) KUMC

Figure 6.1: Sample frames from different colonoscopy. (a) has a higher resolution and a warm color temperature; (b) has lower resolution and a green tone; (c) is more natural in color tone but has a transparent cover around the frame edges.

colonoscopy images from different devices in another medical facility, as demonstrated in Section Experiments and Section Results and Analysis. More recently, there are several large datasets published on colonoscopy [245; 246], like Hyper-Kvasir [247] and Kvasir-SEG [248]. Hyper-Kvasir is a general-purpose dataset for gastrointestinal endoscopy. It detects 23 different classes of findings in the images and videos, including polyp, Angiectasia, Barretts, etc. [247] However, it does not provide the hyperplastic and adenomatous classification. Similarly, Kvasir-SEG provides labels in segmentation format. Thus, they could not be used to train detection models to predict polyp categories.

Another big limiting factor is the lack of distinct training examples. Although the available dataset seems to have many images, these images are actually extracted from a small number of video sequences. Each endoscopic video sequence only contains a single polyp viewed from different viewpoints. If we inspect the polyp frame by frame, we can see that most of the frames are taken from almost identical viewpoints and distances as shown in Figure 6.2. Some video sequences do not have noticeable movement across 1000 frames. Thus, there are significant redundancies in these datasets, especially for polyp classification, which requires a large collection of distinct videos (polyps) to train the classifier. Considering recent benchmark datasets like MS COCO [5] with over 300k distinct images, more colonoscopy data are needed to achieve reasonable performance.

In order to make the best use of the recent development of deep learning technologies for object detection. We collected and created an endoscopic dataset and compared the performance of the



(a) Frame 1 (b) Frame 70 (c) Frame 146

Figure 6.2: A colonoscopy sequence. From frame 1 to frame 146, the camera shows unnoticeable movement.

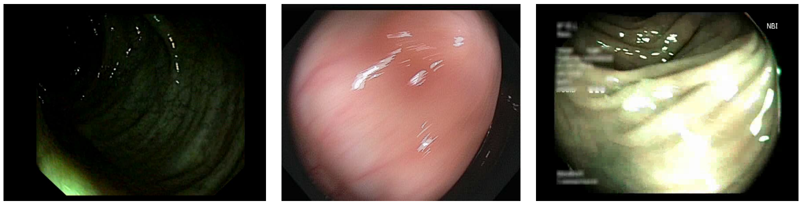
state-of-the-art detectors for polyp detection and classification. These datasets come from various sources and serve different purposes as will be discussed in the following subsection. To integrate them together, we refer to PASCAL VOC [221] object detection task to standardize the annotation. The dataset only contains two categories of polyps: **hyperplastic** and **adenomatous polyps**. It is important to train a model that could reliably differentiate them since adenomatous polyps are commonly considered as precancerous lesions that require resection while hyperplastic polyps are not [243].

6.3.1 Datasets Selection and Annotation

In this study, we have collected all publicly available endoscopic datasets in the research community, as well as collected a new dataset from the University of Kansas Medical Center. All datasets are deidentified without revealing the patient information. With the help of three endoscopists, we annotated the polyp classes of all collected video sequences and the bounding boxes of the polyp in every frame. Below is an introduction to each dataset.

MICCAI 2017: This dataset is designed for Gastrointestinal Image ANAlysis (GIANA), a sub-challenge of the Endoscopic Vision Challenge [249]. It contains 18 videos for training and 20 videos for testing. The dataset is only labeled with polyp masks to test the ability to identify and localize polyps within images. There are no classification labels in this dataset. We converted the polyp masks into bounding boxes for each frame and annotated the polyp class.

CVC colon DB: The dataset has 15 short colonoscopy videos with a total of 300 frames [228].



(a) Underexposed (b) Blurry (c) Out of Focus

Figure 6.3: Some bad examples of colonoscopy frames

The labels are in the form of segmentation masks, and there are no classification labels. We extracted the bounding boxes and labeled the polyp class.

GLRC Dataset: The Gastrointestinal Lesions in Regular Colonoscopy Dataset (GLRC) contains 76 short video sequences with class labels [243]. There is no label for polyp location. We manually annotated the bounding box of each polyp frame by frame.

KUMC Dataset: The dataset was collected from the University of Kansas Medical Center. It contains 80 colonoscopy video sequences. We manually labeled the bounding boxes as well as the polyp classes for the entire dataset.

6.3.2 Frame Selection

The video sequences from these datasets consist of different numbers of frames. For example, CVC colon DB only has 300 frames in total, averaging 20 frames per video sequence, while the number of frames in MICCAI 2017 varies from 400 to more than 1000 with a median value of around 300 in each sequence. The extreme imbalance among different lesions will reduce the representativeness of the dataset. In addition, many frames in a long sequence are redundant since they are taken with very small camera movement. To avoid some long videos overwhelming others, we adopt an adaptive sampling rate to extract the frames from each video sequence based on the camera movement and video lengths to reduce the redundancy and homogenize the representativeness of each polyp. After sampling, we extracted around 300 to 500 frames for long sequences to maintain a balance among different sequences, while for small sequences like CVC colon DB, we simply keep all image frames in the sequence.

After extracting all frames, we carefully checked the generated dataset and manually removed some frames that contain misleading or unuseful information. For example, when there is a sharp movement of the camera, the captured images may be severely blurred, out of focus, or subject to significant illumination change, as shown in Figure 6.3. These images cannot be accurately labeled, so they are removed. While some less flawed frames are kept to improve the model's robustness under imperfect and noisy conditions.

Polyp classification only by visual examination is a big challenge, as reported in [243], the accuracy is normally below 70% even for experienced endoscopists. In clinical practice, the results have to be confirmed by further biopsy tests. However, since we only have video sequences, when the endoscopist could not reach an agreement on the classification results, we simply remove those sequences from the dataset, otherwise, the models may not learn the correct information for classification. Eventually, the dataset contains 155 video sequences (37,899 image frames) with the labeled ground truth of the polyp classes and bounding boxes.

6.3.3 Dataset Split

In order to train and evaluate the performance of different learning models, we need to divide the combined dataset into training, validation, and test sets. For most benchmark datasets for generic object detection, the split is normally based on images. However, this does not apply to the endoscopic dataset. Because all frames in one video sequence correspond to the same polyp, if we split the dataset at the image level, then the same polyp will simultaneously appear in the training, validation, and test sets. This will falsely increase the classification performance since the models have already seen the polyps to be tested during the training stage. Therefore, we split the dataset at the video level.

Since the final dataset is combined from four different datasets captured by different equipment with different data distributions. To increase the representativeness of the dataset, as well as the balance of the two classes of polyps, we make the division for each dataset and polyp class independently. For each class in one dataset, we randomly select 75%, 10%, and 15% sequences

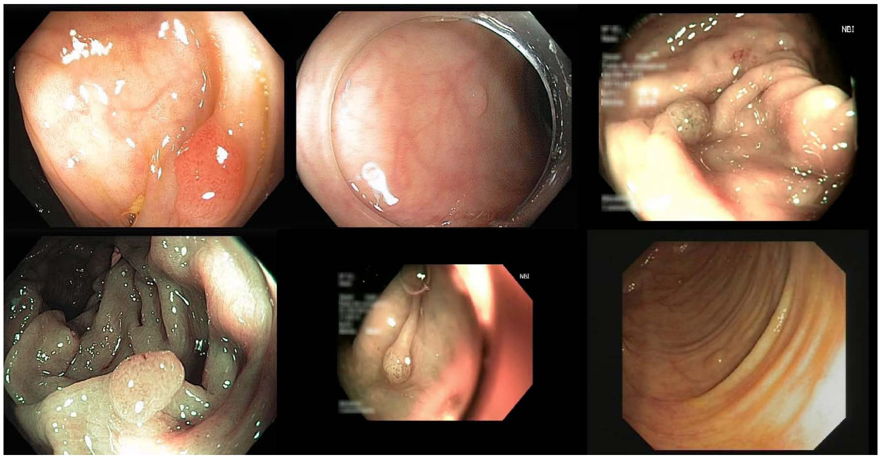


Figure 6.4: **Six sample frames from the generated dataset.**

Table 6.1: **Dataset Organization**

Dataset	MICCAI2017						CVC colon DB						GLRC						KUMC 80					
before selection	38						15						76						80					
after selection	23						15						41						76					
# of Seqs	ad			hp			ad			hp			ad			hp			ad			hp		
	13			10			10			5			26			15			38			38		
	train		val		test		train		val		test		train		val		test		train		val		test	
	20		1		2		11		1		3		29		5		7		56		10		10	
	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp	ad	hp
	12	8	0	1	1	1	7	4	1	0	2	1	19	10	3	2	4	3	27	29	5	5	6	4

to form the training, validation, and test sets, respectively. For example, the GLRC [243] has 41 videos, with 26 adenomatous and 15 hyperplastic sequences. We split the 26 adenomatous sequences and the 15 hyperplastic sequences independently according to the same ratio to guarantee the class balance in the final dataset.

In summary, we have generated 116 training, 17 validation, and 22 test sequences, with 28773, 4254, and 4872 frames, respectively, for each set. Some sample frames from the dataset are shown in Figure 6.4. For the training set, we combine all frames from the 116 sequences into one folder and shuffle them. While for the validation and test sets, we keep the sequence split in order to evaluate the model performance based on polyps (i.e., sequences). The details of the dataset organization are shown in Table 6.1. The dataset can be accessed from this *https://doi.org/10.7910/DVN/FCBUOR*.

Table 6.2: **Experiment Setup**

	Batch Size	Image Size	Learning Rate	Weight Decay	NMS Threshold	Confidence Threshold	Epoch/Iter (2-class det)	Epoch/Iter (1-class det)
Faster RCNN	8	600	10^{-3}	10^{-1}	0.45	0.50	7	3
SSD	8	300	4×10^{-4}	10^{-4}	0.45	0.50	8	$35k$ iter
YOLOv3	32	416	10^{-3}	5×10^{-4}	0.45	0.50	$24k$ iter	
RetinaNet	1	608	10^{-5}		0.50	0.50	9	1
DetNet	8	600	10^{-3}	10^{-4}	0.45	0.50	2	5
RefineDet	8	512	10^{-4}	5×10^{-4}	0.45	0.50	$35k$ iter	$130k$ iter
YOLOv4	4	416	10^{-4}	5×10^{-4}	0.45	0.50	40	26
ATSS	16	600	5×10^{-3}	10^{-4}	0.45	0.50	$15k$ iter	$10k$ iter

6.4 Experiments

Using the generated dataset, we evaluated eight state-of-the-art object detection models, including Faster RCNN [38], YOLOv3 [41], SSD [49], RetinaNet [50], DetNet [4], RefineDet [51], YOLOv4 [44] and ATSS [52]. To set the benchmark performance, three different experiment setups are tested: **frame-based two-class polyps detection**, **frame-based one-class polyps detection**, and **sequence-based two-class polyps classification**. The performance of the two frame-based detections is measured using regular object detection metrics. For the sequence-based classification, regular detection models will be applied to each frame. Then a voting process picks the mostly predicted polyp category as the final classification result. More specific details will be presented below.

The eight detection models are mostly proposed with good performance on generic object detection tasks. These models are adopted from the originally published setups, with slightly modified hyperparameters to optimize their performance on the polyp dataset. The hyperparameter setups are listed in Table 6.2. We employ the following three metrics as shown in Equation 6.1 to evaluate the performance of each model: precision, recall, and F-score.

$$\begin{aligned} Precision &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \\ Recall &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \\ F_1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{6.1}$$

- **Precision** measures the percentage of correct predictions. Polyp detection indicates confidence in the prediction when a positive detection occurs. Higher precision can reduce the chances of a false alarm, which will cause the financial and mental stress of a client.

- **Recall** is the fraction of the objects that are detected. It is very important in polyps detection since a higher recall ensures more patients receive a further check and appropriate treatment in time. It can also reduce mortality and prevent excessive costs to patients.

- **F-score** takes both precision and recall into consideration. It measures a balanced performance of a model between false positive and false negative.

6.4.1 Frame-based Two-class Polyp Detection

This experiment predicts polyps for individual frames. It is a test of a model's localization and classification ability. The CNN models are trained using our training set which consists of a mix of frames from different video sequences. During the validation and test phase, we treat each frame individually and evaluate the performance.

Since the state-of-the-art CNN detectors have fast detection speed and can be implemented in real-time. This allows the endoscopists to find the lesions and provide category suggestions during colonoscopy. As human operators may suffer from fatigue and focus loss after long hours of work, this automated process could alert and assist the endoscopists to focus on suspected lesions and avoid miss detection.

To test the effectiveness of the proposed dataset with respect to a single dataset mentioned above, we also perform the frame-based two-class detection using a single dataset. In this controlled experiment, we train all the models trained using the KUMC dataset. Since this dataset contains a variety of more sequences and video frames than other datasets, it guarantees the convergence of all involved models. After training, we test the models on the same combined test set as in other experiments. As shown in the results, the performance of all models will drop significantly when trained using only a single dataset. This experiment verifies the effectiveness of the combined dataset.

6.4.2 Frame-based One-class Polyp Detection

This experiment has almost the same setups as the frame-based two-class polyp detection except for the class number. Hyperplastic and adenomatous polyps are treated as a single class *polyps*. For annotation files, instead of providing a separate set of annotation files, we read the same ground truth as the previous experiment, and discard the information about polyp categories during training and inference time.

In colorectal cancer screening, it is more important to accurately detect whether polyps are developed than classifying polyp categories, because further screening and diagnosis are always followed after colonoscopy finds suspected lesions. This experiment aims to test whether a higher performance could be achieved by only localizing polyps in general. Without the more challenging task of classifying polyp categories, CNN models could be trained to extract more generalized features to distinguish polyps. Screening methods with higher precision like biopsy or polypectomy then could be followed to determine the categories of lesions.

6.4.3 Sequence-based Two-class Polyp Classification

This experiment adopts the same setup as Frame-based Two-class tests, however, we only make one prediction for each video sequence since it only contains the same polyp in the sequence. During the test period, it will generate the prediction based on individual frames at first, then we collect all results from every frame of a video sequence and classify the video based on the mostly predicted polyp category. Although there may be better ways to classify a video sequence such as based on the confidence score of the prediction for each frame, we only adopt the basic approach as a benchmark to see how much improvement we can achieve for sequence-based prediction.

Sequence-based classification is the practice of clinical application since all frames in the sequence are observing the same polyp from different viewpoints. It also has the potential to achieve better performance. To classify the polyp only from a frame is difficult, for example, the polyp

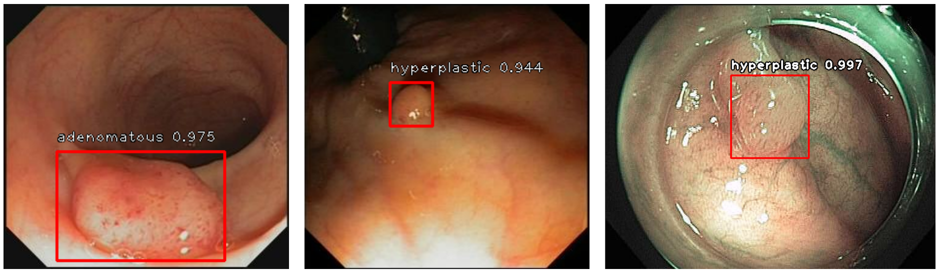


Figure 6.5: Three examples of the detection results with the predicted classes and confidence scores

may be partly occluded in some frames or appear small when viewed from a far distance. All these scenarios will make it hard to be accurately classified. However, in video-based classification, we are combining information from different viewpoints which can reduce the influence caused by those hard frames. Thus, at the clinic, the endoscopist usually takes the colonoscopic video from various viewpoints to ensure a reliable classification of the polyps.

6.5 Results and Analysis

In the experiments, the frame-based and sequence-based two-class detection and classification share the same CNN model. All hyperparameters for the compared models are summarized in Table 6.2. The final models chosen for the test are based on the validation performance. Precision, recall, and F1 scores are all calculated at the confidence threshold of 0.5 to ensure a fair comparison. The best-performance CNN models are mostly produced before epoch 10. An exception is RefineDet one-class detection, with 130*k iter* equaling around 45 *epochs*. However, it has achieved similar validation performance, 88.05% *mAP*, as early as 30*k iter* compared to 88.12% at 130*k iter*. We suggest the best CNN model for polyps detection is usually generated at the earlier training stage.

6.5.1 Frame-based Two-class Polyp Detection

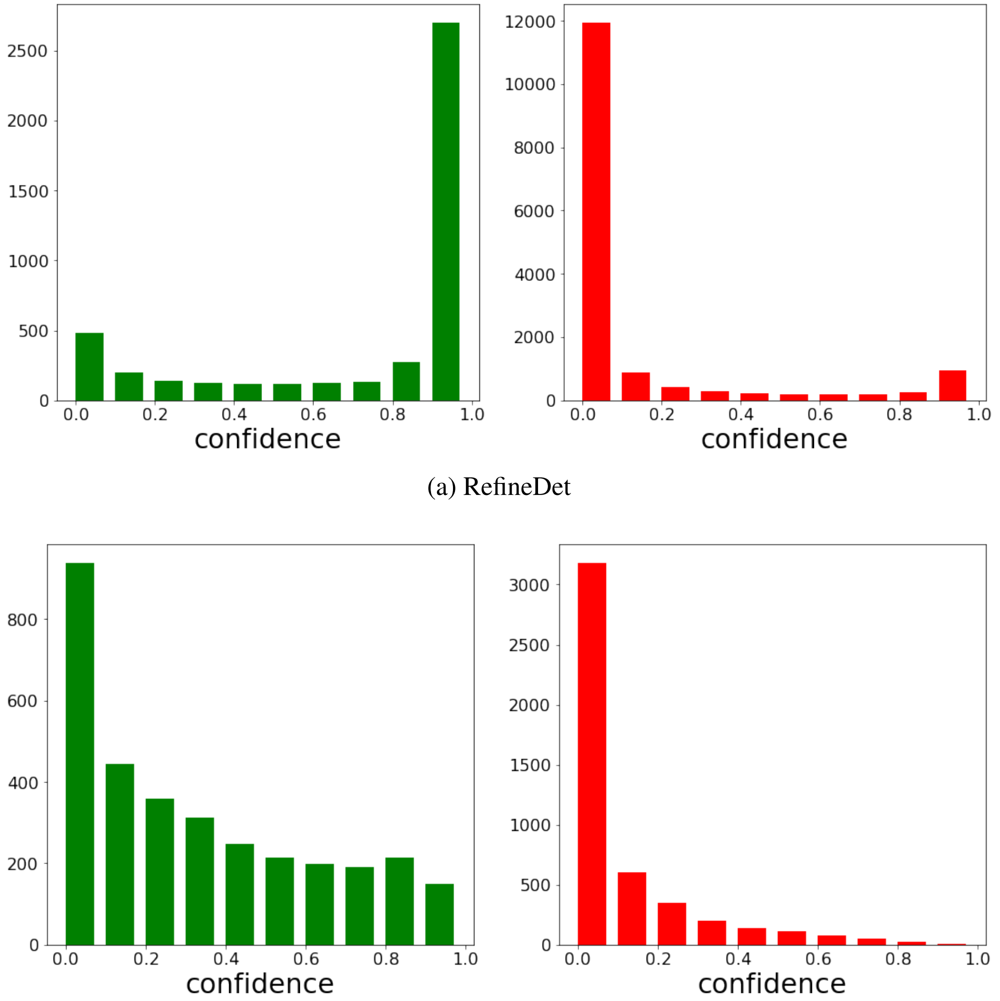
The results are shown in Table 6.3. Overall, all detectors have achieved better performance for adenomatous polyps since they are larger in size and their shape and texture are easier to distinguish

Table 6.3: **Results for Frame-based Two-class Polyp Detection**

	Category	Precision	Recall	F1-score	AP
Faster RCNN	ad	72.8	73.0	72.9	72.9
	hp	42.2	63.1	50.6	42.5
	Mean	57.5	68.1	62.3	57.7
SSD	ad	82.7	77.4	80.0	82.7
	hp	54.6	51.8	53.1	52.5
	Mean	68.6	64.6	66.5	67.6
YOLOv3	ad	89.7	23.2	36.9	61.1
	hp	60.0	16.2	25.5	35.0
	Mean	**74.9**	19.7	31.2	48.0
RetinaNet	ad	85.4	59.1	69.8	57.9
	hp	52.9	43.7	47.9	40.5
	Mean	69.2	51.4	59.0	49.2
DetNet	ad	73.0	67.5	70.2	60.4
	hp	46.0	65.0	53.8	42.2
	Mean	59.5	66.2	62.7	51.3
RefineDet	ad	92.2	61.3	73.6	81.1
	hp	49.1	86.3	62.6	65.9
	Mean	70.7	**73.8**	**72.2**	**73.5**
YOLOv4	ad	90.5	54.0	67.6	70.4
	hp	54.0	40.6	46.3	42.7
	Mean	72.3	47.3	57.2	56.6
ATSS	ad	79.5	76.3	77.9	80.7
	hp	57.2	68.0	62.2	58.4
	Mean	68.4	72.2	70.2	69.5

from the colonic wall. RefineDet has achieved the best-combined performance. It achieves the highest mean F1-score, mAP, and mean recall than all other models. YOLOv3 yields the best precision by sacrificing its recall, which is abnormally lower than other detectors. Figure 6.5 shows some examples of the detection results. We pick a confidence threshold of 0.5. As shown in the examples, the models are very confident about their predictions. They mostly have only one prediction with a confidence score over 0.5 on each frame. The predicted bounding boxes are very tight and precise on the lesions, which shows great potential in assisting colonoscopy practice.

To analyze the difference between recalls from YOLOv3 and other detectors at the confidence threshold of 0.5, we have plotted the count of true positives (TP) and false positives (FP) over



(a) RefineDet

(b) YOLOv3

Figure 6.6: True positive (green plot) and false positive (red plot) count w.r.t. confidence. We discard any predictions with a confidence score below 0.01 since they tend to be random predictions.

different confidence scores. In Figure 6.6, we only show the charts from RefineDet and YOLOv3 since RefineDet has similar patterns as the other four detectors. RefineDet and other detectors show a clear maximum peak for TP count at confidence > 0.9 and another weaker peak for confidence < 0.1. While YOLOv3 has fewer predictions with high confidence. Therefore, although YOLOv3 is a conservative prediction that has high accuracy, it misses a large proportion of lesions and results in low recall.

SSD yields the best adenomatous polyps detection recall, F1 score, and AP value. Overall, its mAP (67.6%) ranks third, closely matching the most recent detector ATSS and leading the following detector, Faster RCNN with mAP of 57.7%, by a considerable margin. For the harder task of hyperplastic polyp detection, RefineDet yields the highest scores for recall, F1, and AP. These results show that SSD-based detectors, SSD, RetinaNet, and RefineDet, are generally doing well in detecting polyps. RefineDet, by roughly adjusting anchors first, obtains better localization knowledge before generating final predictions. Faster RCNN has a similar two-step architecture. Therefore, it also has decent performance. This indicates the possibility of improving polyp detection performance by adding more refined location information before making final predictions. YOLOv4 outperforms YOLOv3 in almost every aspect, indicating that the tricks on general-purpose detectors are also effective in polyps detection. ATSS ranks second at hyperplastic polyp detection precision after YOLOv3. It also consistently performs on par with RefineDet, especially in hyperplastic detection.

6.5.2 Generalizability and Comparison with Previous Dataset

The generalization ability refers to the adaptivity of the trained models to new, previously unseen data. This is very crucial in practical applications since the test images may have different distributions from the ones used to create the model. In order to test if the newly generated dataset can increase the generalizability of the trained models, we compare our results with the models only trained on a single dataset.

We conduct the frame-based two-class polyp detection only on a single dataset, the KUMC dataset. The models are trained using the images from KUMC and tested on the full combined test set as in other experiments that consist of frames from different datasets. The results of different models are shown in Table 6.4. We can see that, on average, the performance is dropped by 8%, when we compare the results in Table 6.3 where all models are trained using the proposed dataset. The performance drop is mainly caused by the representativeness and the number of training samples. Although KUMC contains more variable sequences and frames than the other datasets combined, the color and illumination of different datasets may differ greatly, as shown in Figure 6.1. Therefore, the models trained on a single dataset may suffer poor generalization.

Table 6.4: **Result for training on KUMC and testing on the full combined test set**

	Faster RCNN	SSD	YOLO v3	Retina Net	DetNet	RefineDet	YOLOv4	ATSS
mAP	52.7	56.1	42.6	36.9	51.5	60.8	51.4	60.8

Table 6.5: **Result for Frame-based One-class Polyp Detection**

	Precision	Recall	F1 score	AP	Inference Time	FPS
Faster RCNN	63.9	**89.8**	74.7	85.6	52ms	19
SSD	91.3	82.0	86.4	86.3	**17ms**	**59**
YOLOv3	**95.9**	78.0	86.0	81.0	**17ms**	**59**
RetinaNet	86.1	86.6	86.3	87.9	61ms	16
DetNet	85.8	81.8	83.7	80.5	64ms	16
RefineDet	91.2	86.2	**88.6**	**88.5**	31ms	32
YOLOv4	89.8	74.4	81.3	83.9	30ms	33
ATSS	92.1	84.7	88.3	88.1	53ms	19

6.5.3 Frame-based One-class Polyp Detection

The results for detection only without classification are shown in Table 6.5. We can see YOLOv3 achieves the highest precision among all detectors, which is consistent with the two-class results. With a reasonable recall, it also yields a high F1 score. Compared to its two-class detection performance, there is evidence that YOLOv3 is better at detecting than classifying polyps. YOLOv3 generates classification scores and bounding box adjustments at the same time. Since classification performance is based on the anchor information, YOLOv3's original anchors might not contain sufficient portions of a polyp due to its small size. We suggest that refined location information is more important for distinguishing polyp categories than for locating them.

Table 6.6 shows the detailed localization results for adenomatous and hyperplastic polyps. We can see that Faster RCNN achieves the best recall, which is the most important metric in clinical settings. For adenomatous polyps, Faster RCNN achieves 93.3% recall, on par with recent clinical screening results. It is one of the only three detectors (with RefineDet and ATSS) that achieve over 80% recall for the hyperplastic polyps. Recall that in the above two-class detection, Faster RCNN also achieves the top three recall scores. Thanks to the region proposals, two-stage

Table 6.6: **Frame-based One-class Polyp Detection Results for each Class**

	Category	Precision	Recall	F1-score	AP
Faster RCNN	ad	75.9	**93.3**	83.7	90.2
	hp	50.0	**84.3**	62.8	74.3
SSD	ad	96.1	87.6	91.7	89.8
	hp	83.5	73.1	77.9	79.9
YOLOv3	ad	**98.6**	86.4	92.1	**90.5**
	hp	**90.5**	64.6	75.4	77.6
RetinaNet	ad	93.4	91.7	92.6	90.3
	hp	75.2	78.5	76.8	81.4
DetNet	ad	93.5	86.6	89.9	81.7
	hp	74.4	74.1	74.2	75.1
RefineDet	ad	96.1	89.6	**92.7**	90.3
	hp	83.7	80.8	82.2	**85.2**
YOLOv4	ad	92.9	79.7	85.8	86.0
	hp	84.4	65.9	74.0	79.3
ATSS	ad	96.1	87.3	91.5	89.6
	hp	86.0	80.6	**83.2**	84.2

detectors usually have more chances to detect the polyps. While YOLOv3 also achieves competing performance in one-class detection. It yields the highest precision with a reasonable recall score.

RefineDet still yields the best overall performance with the highest F1 score and AP. All SSD-based detectors perform almost equally well. The focal loss of RetinaNet does not show significant improvement on the original SSD model. DetNet does not show improvement over Faster RCNN, however, it makes the detector more balanced by increasing the precision by 20%+, resulting in a better F1 score.

We also evaluated the inference time of different models in frame-based one-class detection. All models are evaluated on an NVIDIA TESLA P100 GPU. As shown in Table 6.5, The single-stage detectors (SSD, YOLOv3, RetinaNet, and RefineDet) are faster than the two-stage detectors (Faster RCNN and DetNet). SSD and YOLOv3 achieve the fastest inference time at 17ms, which is over 60 frames per second (fps). However, even for the slowest model DetNet, it still achieves 64ms, which is above 15 fps. Please note that a deeper backbone network will require more inference time than a shallower backbone network. For example, RetinaNet with ResNet-50 increases

Table 6.7: **Result for Sequence-based Two-class Polyp Classification**

	Precision	Recall	F1-score
Faster RCNN	81.2	81.2	81.2
SSD	86.6	85.0	85.8
YOLOv3	72.2	60.3	65.7
RetinaNet	81.8	82.9	82.4
DetNet	85.8	**86.8**	**86.3**
RefineDet	78.1	73.1	75.5
YOLOv4	**87.5**	80.8	84.0
ATSS	81.2	81.2	81.2

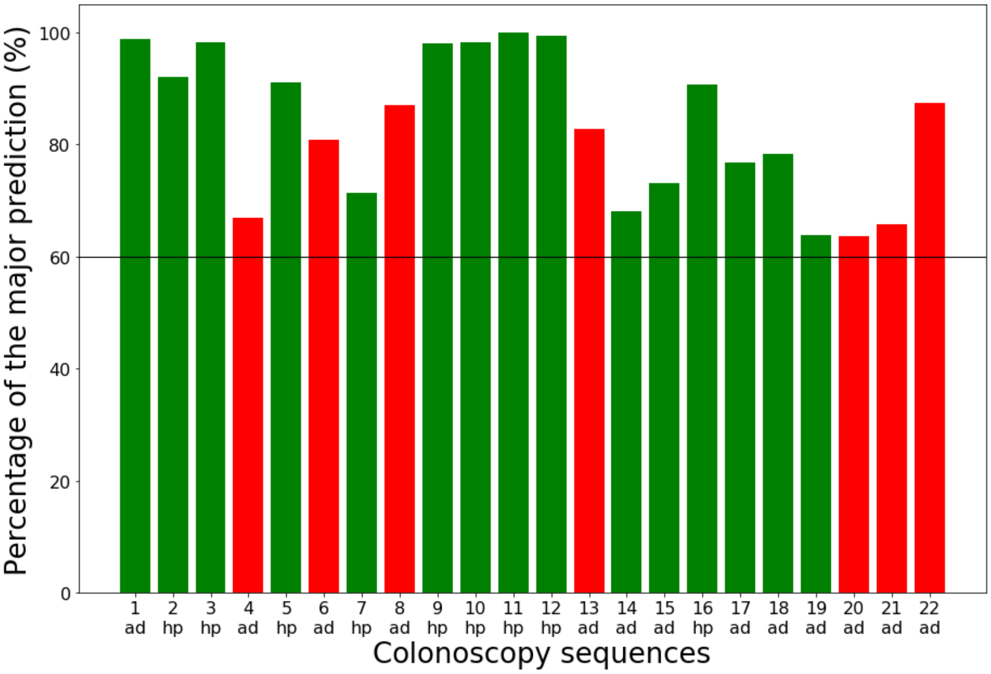
the inference time to 61ms from 17ms for SSD with VGG-16.

6.5.4 Sequence-based Two-class Polyp Classification

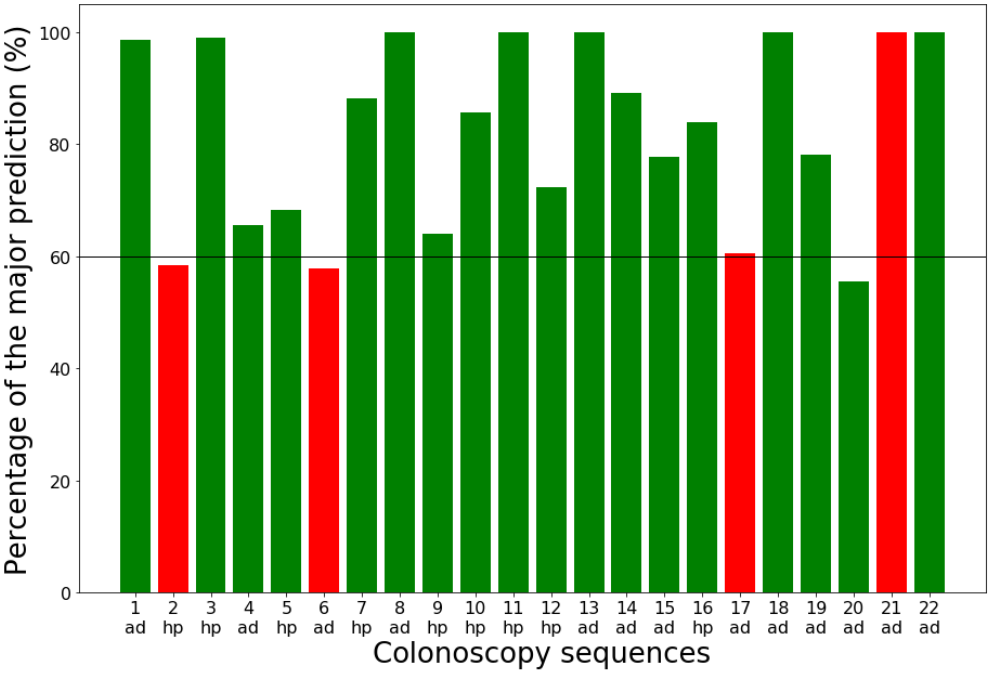
From Table 6.7, we can see that both SSD, DetNet and YOLOv4 stand out in terms of precision, recall, and F1 score. This means that they are better at predicting correct polyp categories. Another interesting observation is that, although some detectors produce more consistent results for different frames in the same sequence, they do not yield higher precision. It becomes obvious when we plot the percentage of the dominant predicted category in each video sequence in Figure 6.7. We only show the plots for RetinaNet and RefineDet as examples. DetNet, FasterRCNN, and RetinaNet are not very consistent in predicting the polyp class for some of the video sequences, with close to 50% dominant class. This means the predictions are not robust with only a few frames to swing the result. RefineDet, SSD, YOLOv4, and ATSS, on the other hand, are relatively more robust in predicting the polyp class with most sequences above 70%.

6.6 Conclusion

In this paper, we have developed a relatively large endoscopic dataset for polyp detection and classification. We have also evaluated and compared the performance of eight state-of-the-art deep learning-based object detectors. Our results show that deep CNN models are promising in assisting CRC screening. Without too much modification, general object detectors have already



(a) RefineDet



(b) RetinaNet

Figure 6.7: Percentage of the dominant class. Detectors predict the polyp category in each individual frame. The category with more than 50% of all frames is the dominant category for that video sequence. The charts show the percentage of frames classified as the dominant class in each test sequence. (ad) and (hp) on the bottom means ground truth class adenomatous and hyperplastic respectively. Correct predictions are in green and misclassifications are in red.

achieved adenomatous polyps detection sensitivity of 91% in the one-class detector and around 70% precision in the classification task. Among all the detectors we have tested, YOLOv4, ATSS, and RefineDet perform relatively well in all tests with balanced precision and recall scores and consistent results for the same lesions. Our experiments also show the refinement of location information before classification will effectively boost the performance.

This study can serve as a baseline for future research in polyp detection and classification. The developed dataset can serve as a standardized platform and help researchers design more specialized CNN models for polyp classification. Looking back at the fast development in the computer vision field in recent years, the availability of the benchmark dataset plays a significant role. We hope our dataset will greatly facilitate the computer-aided diagnosis of colorectal cancer.

Chapter 7

Conclusion and Future Work

We have identified problems in existing classification and detection models. To solve them, we proposed specific modules to improve semantic understanding, boost detection location capability, increase training efficiency, and align research directions.

In the first two parts of our work, we propose novel modules to increase DNN models' capability to understand semantic information and improve models' location capability. During this period, researchers have identified weaknesses of DNN-based models and tried to tackle them by introducing novel architectures and modifying weights to improve efficiency. In the first part of our work, we introduce a super-class guided network designed to incorporate high-level knowledge into existing models for image classification and object detection. The second part proposes the localization score (locscore) branch and location-aware box reasoning, which is often overlooked in most object detection frameworks.

The third part of our work aims to address the problem of 3D point cloud classification by proposing a family of novel robust structured declarative classifiers that at the same time provide defense mechanism against adversarial attacks. In this work, we further propose an efficient instantiation, Lattice Point Classifier (LPC).

The fourth chapter explores the generalization of the previously proposed declarative defenders. We implemented three distinct mapping algorithms as declarative defenders for 3D point cloud classification. During the experiments, we observe the diminishing defense capability in the graph drawing classifier. Further investigation concludes that the presence of a skip connection circumvents implicit gradients, thereby defeating the defense capability.

In the fifth work, we have developed a relatively large endoscopic dataset for polyp detection and classification. We have also evaluated and compared the performance of eight state-of-the-art deep learning-based object detectors. In addition to the dataset mentioned in this article, we also organize a dataset for aphid detection in agriculture applications. These studies can serve as a baseline for future research in medical and agriculture applications. We hope our dataset will greatly facilitate computer-aided applications.

In the future, we hope to extend our study on 3D classification tasks to more complicated 3D detection and segmentation tasks, which serve as the fundamental part of autonomous driving and many other real-life applications. With recent rapid development in multi-modal foundation models, many emerging research and application directions become possible. These foundation models are trained with billions of data points and incorporate high-level language information, which can be utilized to perform sophisticated tasks.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Detnet: A backbone network for object detection," *arXiv preprint arXiv:1804.06215*, 2018.

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[6] Y. Lyu, X. Huang, and Z. Zhang, "Learning to segment 3d point clouds in 2d image space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12255–12264, 2020.

[7] B. Delaunay *et al.*, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, no. 793-800, pp. 1–2, 1934.

[8] G. Metzer, R. Hanocka, R. Giryes, N. J. Mitra, and D. Cohen-Or, "Z2p: Instant visualization of point clouds," in *Computer Graphics Forum*, vol. 41, pp. 461–471, Wiley Online Library, 2022.

[9] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[10] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[11] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

[12] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[13] Y. Zhao, Y. Wu, C. Chen, and A. Lim, “On isometry robustness of deep 3d point cloud models under adversarial attacks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1201–1210, 2020.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[15] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[16] I. Arel, D. C. Rose, and T. P. Karnowski, “Deep machine learning-a new frontier in artificial intelligence research [research frontier],” *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[17] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.

[18] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[19] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for lvcsr," in *2013 IEEE workshop on automatic speech recognition and understanding*, pp. 315–320, IEEE, 2013.

[20] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[21] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, 2011.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[23] L. He, G. Wang, and Z. Hu, "Learning depth from single images with deep neural network embedding focal length," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4676–4689, 2018.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[26] W. Ma, Y. Wu, Z. Wang, and G. Wang, "Mdcn: Multi-scale, deep inception convolutional neural networks for efficient object detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2510–2515, IEEE, 2018.

[27] Z. Zhang, Y. Wu, and G. Wang, "Bpgrad: Towards global optimality in deep learning via branch and pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3301–3309, 2018.

[28] X. Chen, Q. Hu, K. Li, C. Zhong, and G. Wang, "Accumulated trivial attention matters in vision transformers on small datasets," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3984–3992, 2023.

[29] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, 2017.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[32] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[33] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "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.

[40] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[41] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

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

[43] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[44] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[45] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[48] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.

[49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[51] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for

object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, 2018.

[52] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9759–9768, 2020.

[53] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9627–9636, 2019.

[54] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[55] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[56] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, pp. 820–830, 2018.

[57] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5589–5598, 2020.

[58] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

[59] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5648–5656, 2016.

[60] T. Yu, J. Meng, and J. Yuan, “Multi-view harmonized bilinear network for 3d object recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 186–194, 2018.

[61] Z. Yang and L. Wang, “Learning relationships for multi-view 3d object recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7505–7514, 2019.

[62] D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, IEEE, 2015.

[63] G. Riegler, A. Osman Ulusoy, and A. Geiger, “Octnet: Learning deep 3d representations at high resolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3577–3586, 2017.

[64] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, “O-cnn: Octree-based convolutional neural networks for 3d shape analysis,” *ACM Transactions On Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[65] T. Le and Y. Duan, “Pointgrid: A deep network for 3d shape understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9204–9214, 2018.

[66] L. Landrieu and M. Boussaha, “Point cloud oversegmentation with graph-structured deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7440–7449, 2019.

[67] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[68] W. Shi and R. Rajkumar, “Point-gnn: Graph neural network for 3d object detection in a point cloud,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1711–1719, 2020.

[69] K. Fu, S. Liu, X. Luo, and M. Wang, “Robust point cloud registration framework based on deep graph matching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8893–8902, 2021.

[70] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, “Pu-gcn: Point cloud upsampling using graph convolutional networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11683–11692, 2021.

[71] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, “Deep learning for 3d point clouds: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[72] D. Liu, R. Yu, and H. Su, “Extending adversarial attacks and defenses to deep 3d point cloud classifiers,” in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2279–2283, IEEE, 2019.

[73] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian, “Adversarial attack and defense on point sets,” *arXiv preprint arXiv:1902.10899*, 2019.

[74] C. Xiang, C. R. Qi, and B. Li, “Generating 3d adversarial point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, 2019.

[75] J. Kim, B.-S. Hua, T. Nguyen, and S.-K. Yeung, “Minimal adversarial examples for deep learning on 3d point clouds,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7797–7806, 2021.

[76] C. Ma, W. Meng, B. Wu, S. Xu, and X. Zhang, “Efficient joint gradient based attack against

sor defense for 3d point cloud classification," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1819–1827, 2020.

[77] F. Cen and G. Wang, "Boosting occluded image classification via subspace decomposition-based estimation of deep features," *IEEE transactions on cybernetics*, 2019.

[78] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[79] K. Li, W. Ma, U. Sajid, Y. Wu, and G. Wang, "Object detection with convolutional neural networks," *Deep Learning in Computer Vision: Principles and Applications*, vol. 30, no. 31, p. 41, 2020.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[81] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye, "Learning to match anchors for visual object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[82] W. Ma, Y. Wu, F. Cen, and G. Wang, "Mdfn: Multi-scale deep feature learning network for object detection," *Pattern Recognition*, vol. 100, p. 107149, 2020.

[83] F. Cen, X. Zhao, W. Li, and G. Wang, "Deep feature augmentation for occluded image classification," *Pattern Recognition*, vol. 111, p. 107737, 2020.

[84] W. Ma, K. Li, and G. Wang, "Location-aware box reasoning for anchor-based single-shot object detection," *IEEE Access*, vol. 8, pp. 129300–129309, 2020.

[85] U. Sajid, H. Sajid, H. Wang, and G. Wang, "Zoomcount: A zooming mechanism for crowd counting in static images," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[86] W. Xu, K. Shawn, and G. Wang, "Toward learning a unified many-to-many mapping for diverse image translation," *Pattern Recognition*, vol. 93, pp. 570–580, 2019.

[87] H. Liu, S. Zhang, K. Lin, J. Wen, J. Li, and X. Hu, "Vocabulary-wide credit assignment for training image captioning models," *IEEE Transactions on Image Processing*, vol. 30, pp. 2450–2460, 2021.

[88] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4651–4659, 2016.

[89] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

[90] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4894–4902, 2017.

[91] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, IEEE, 2009.

[92] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen, "Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5975–5984, 2016.

[93] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[94] J. Read, C. Bielza, and P. Larrañaga, "Multi-dimensional classification with super-classes," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 7, pp. 1720–1733, 2013.

[95] Y. Zhou, Q. Hu, and Y. Wang, "Deep super-class learning for long-tail distributed image classification," *Pattern Recognition*, vol. 80, pp. 118–128, 2018.

[96] D. Roy, P. Panda, and K. Roy, "Tree-cnn: a hierarchical deep convolutional neural network for incremental learning," *Neural Networks*, vol. 121, pp. 148–160, 2020.

[97] L. Wang and A. Sohmshetty, "Learning image representations to understand and predict semantic hierarchies,"

[98] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition," in *Proceedings of the IEEE international conference on computer vision*, pp. 2740–2748, 2015.

[99] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[100] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[101] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated rcnn and multi-class deep sort for multi-object detection and tracking," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[102] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[103] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale

hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[104] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[105] Z. Zhang, W. Ma, Y. Wu, and G. Wang, "Self-orthogonality module: A network architecture plug-in for learning orthogonal filters," *arXiv preprint arXiv:2001.01275*, 2020.

[106] W. Xu, G. Wang, A. Sullivan, and Z. Zhang, "Towards learning affine-invariant representations via data-efficient cnns," in *The IEEE Winter Conference on Applications of Computer Vision*, pp. 904–913, 2020.

[107] Y. Wu, Z. Zhang, and G. Wang, "Unsupervised deep feature transfer for low resolution image classification," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[108] W. Xu, Y. Wu, W. Ma, and G. Wang, "Adaptively denoising proposal collection for weakly supervised object localization," *Neural Processing Letters*, vol. 51, no. 1, pp. 993–1006, 2020.

[109] K. Li, W. Ma, U. Sajid, Y. Wu, and G. Wang, "Object detection with convolutional neural networks," *arXiv preprint arXiv:1912.01844*, 2019.

[110] W. Xu, S. Keshmiri, and G. Wang, "Adversarially approximated autoencoder for image generation and manipulation," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2387–2396, 2019.

[111] Y. He, X. Ma, X. Luo, J. Li, M. Zhao, B. An, and X. Guan, "Vehicle traffic driven camera placement for better metropolis security surveillance," *arXiv preprint arXiv:1705.08508*, 2017.

[112] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, “Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3980–3989, 2017.

[113] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[114] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, “Mask scoring r-cnn,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6409–6418, 2019.

[115] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, “Freeanchor: Learning to match anchors for visual object detection,” in *Advances in Neural Information Processing Systems*, pp. 147–155, 2019.

[116] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[117] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in *Advances in neural information processing systems*, pp. 379–387, 2016.

[118] S. Gidaris and N. Komodakis, “Object detection via a multi-region and semantic segmentation-aware cnn model,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1134–1142, 2015.

[119] C. Zhu, Y. He, and M. Savvides, “Feature selective anchor-free module for single-shot object detection,” *arXiv preprint arXiv:1903.00621*, 2019.

[120] L. Tychsen-Smith and L. Petersson, “Improving object localization with fitness nms and bounded iou loss,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6877–6885, 2018.

[121] L. Tychsen-Smith and L. Petersson, "Denet: Scalable real-time object detection with directed sparse sampling," in *Proceedings of the IEEE international conference on computer vision*, pp. 428–436, 2017.

[122] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 784–799, 2018.

[123] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting rcnn: On awakening the classification power of faster rcnn," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 453–468, 2018.

[124] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, pp. 5561–5569, 2017.

[125] L. Neumann, A. Zisserman, and A. Vedaldi, "Relaxed softmax: Efficient confidence autocalibration for safe pedestrian detection," 2018.

[126] S. Wu, X. Li, and X. Wang, "Iou-aware single-stage object detector for accurate localization," *Image and Vision Computing*, p. 103911, 2020.

[127] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[128] H. Zhou, D. Chen, J. Liao, K. Chen, X. Dong, K. Liu, W. Zhang, G. Hua, and N. Yu, "Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10356–10365, 2020.

[129] Z. Xiang, D. J. Miller, S. Chen, X. Li, and G. Kesidis, “A backdoor attack against 3d point cloud classifiers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7597–7607, October 2021.

[130] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, “Simple black-box adversarial attacks,” in *International Conference on Machine Learning*, pp. 2484–2493, PMLR, 2019.

[131] M. Wicker and M. Kwiatkowska, “Robustness of 3d deep learning in an adversarial setting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11767–11775, 2019.

[132] D. Liu, R. Yu, and H. Su, “Adversarial shape perturbations on 3d point clouds,” in *European Conference on Computer Vision*, pp. 88–104, Springer, 2020.

[133] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, “Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1961–1970, 2019.

[134] Z. Wu, Y. Duan, H. Wang, Q. Fan, and L. J. Guibas, “If-defense: 3d adversarial point cloud defense via implicit function based restoration,” *arXiv preprint arXiv:2010.05272*, 2020.

[135] H. Liu, J. Jia, and N. Z. Gong, “Pointguard: Provably robust 3d point cloud classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6186–6195, 2021.

[136] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in *International conference on machine learning*, pp. 274–283, PMLR, 2018.

[137] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine, “Meta-learning with implicit gradients,” in *Advances in neural information processing systems*, 2019.

[138] A. Kag, Z. Zhang, and V. Saligrama, "Rnns incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients?," in *International Conference on Learning Representations*, 2019.

[139] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*, pp. 136–145, PMLR, 2017.

[140] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in Neural Information Processing Systems*, vol. 32, pp. 9562–9574, 2019.

[141] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

[142] S. Gould, R. Hartley, and D. J. Campbell, "Deep declarative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[143] A. Szlam, K. Gregor, and Y. LeCun, "Fast approximations to structured sparse coding and applications to object classification," in *European Conference on Computer Vision*, pp. 200–213, Springer, 2012.

[144] S. Karygianni and P. Frossard, "Structured sparse coding for image denoising or pattern detection," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3533–3537, IEEE, 2014.

[145] W. Wei, L. Zhang, C. Tian, A. Plaza, and Y. Zhang, "Structured sparse coding-based hyperspectral imagery denoising with intracluster filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 6860–6876, 2017.

[146] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permuto-

hedral lattice," in *Computer graphics forum*, vol. 29, pp. 753–762, Wiley Online Library, 2010.

[147] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2530–2539, 2018.

[148] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang, "Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3254–3263, 2019.

[149] M. Ozdag, "Adversarial attacks and defenses against deep neural networks: a survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018.

[150] C. Wang, J. Wang, and Q. Lin, "Adversarial attacks and defenses in deep learning: A survey," in *International Conference on Intelligent Computing*, pp. 450–461, Springer, 2021.

[151] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

[152] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[153] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1598–1606, 2019.

[154] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," in *European Conference on Computer Vision*, pp. 241–257, Springer, 2020.

[155] C. Bian, W. Jiang, J. Zhan, Z. Song, X. Wen, and H. Lei, "A physically realizable backdoor attack on 3d point cloud deep learning: work-in-progress," in *Proceedings of the 2021 In-*

ternational Conference on Hardware/Software Codesign and System Synthesis, pp. 27–28, 2021.

[156] Y. Zhang, Y. Zhu, Z. Liu, C. Miao, F. Hajiaghajani, L. Su, and C. Qiao, "Towards backdoor attacks against lidar object detection in autonomous driving," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pp. 533–547, 2022.

[157] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[158] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 954–962, 2020.

[159] X. Dong, D. Chen, H. Zhou, G. Hua, W. Zhang, and N. Yu, "Self-robust 3d point recognition via gather-vector guidance," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11513–11521, IEEE, 2020.

[160] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[161] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun, "Physically realizable adversarial examples for lidar object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13716–13725, 2020.

[162] J. Zhang, B. Liu, L. Chen, B. Ouyang, J. Zhu, M. Kuang, H. Wang, and Y. Meng, "The art of defense: letting networks fool the attacker," *arXiv preprint arXiv:2104.02963*, 2021.

[163] J. Zhang, L. Chen, B. Ouyang, B. Liu, J. Zhu, Y. Chen, Y. Meng, and D. Wu, "Pointcutmix:

Regularization strategy for point cloud classification," *arXiv preprint arXiv:2101.01461*, 2021.

[164] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4452–4461, 2016.

[165] R. A. Rosu, P. Schütt, J. Quenzel, and S. Behnke, "Latticenet: fast spatio-temporal point cloud segmentation using permutohedral lattices," *Autonomous Robots*, pp. 1–16, 2021.

[166] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE access*, vol. 3, pp. 490–530, 2015.

[167] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

[168] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[169] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[170] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[171] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 523–540, Springer, 2020.

[172] J. E. Dennis Jr and R. E. Welsch, “Techniques for nonlinear least squares and robust regression,” *Communications in Statistics-simulation and Computation*, vol. 7, no. 4, pp. 345–359, 1978.

[173] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *kdd*, vol. 96, pp. 226–231, 1996.

[174] K. Li, N. Y. Wang, Y. Yang, and G. Wang, “Sgnet: A super-class guided network for image classification and object detection,” *arXiv preprint arXiv:2104.12898*, 2021.

[175] K. Li, Z. Zhang, C. Zhong, and G. Wang, “Robust structured declarative classifiers for 3d point clouds: Defending adversarial attacks with implicit gradients,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15294–15304, 2022.

[176] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.

[177] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[178] H. Lu and H. Shi, “Deep learning for 3d point cloud understanding: A survey,” *arXiv preprint arXiv:2009.08920*, 2020.

[179] K. Zhang, H. Zhou, J. Zhang, Q. Huang, W. Zhang, and N. Yu, “Ada3diff: Defending against 3d adversarial point clouds via adaptive diffusion,” in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8849–8859, 2023.

[180] G. Li, G. Xu, H. Qiu, R. He, J. Li, and T. Zhang, “Improving adversarial robustness of 3d point cloud classification models,” in *European Conference on Computer Vision*, pp. 672–689, Springer, 2022.

[181] W. Schnyder, "Embedding planar graphs on the grid," in *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pp. 138–148, 1990.

[182] W. Didimo, G. Liotta, and F. Montecchiani, "A survey on graph drawing beyond planarity," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–37, 2019.

[183] J. Pach and G. Tóth, "Graphs drawn with few crossings per edge," *Combinatorica*, vol. 17, no. 3, pp. 427–439, 1997.

[184] W. Didimo, P. Eades, and G. Liotta, "Drawing graphs with right angle crossings," *Theoretical Computer Science*, vol. 412, no. 39, pp. 5156–5166, 2011.

[185] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[186] F. Cazals and J. Giesen, *Delaunay triangulation based surface reconstruction: ideas and algorithms*. PhD thesis, INRIA, 2004.

[187] Y. Nagai, Y. Ohtake, and H. Suzuki, "Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction," in *Computer Graphics Forum*, vol. 28, pp. 1339–1348, Wiley Online Library, 2009.

[188] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[189] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," in *ACM SIGGRAPH 2007 papers*, pp. 24–es, 2007.

[190] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, "Point2mesh: A self-prior for deformable meshes," *arXiv preprint arXiv:2005.11084*, 2020.

[191] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–14, 2019.

[192] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu, *et al.*, "Audiogpt: Understanding and generating speech, music, sound, and talking head," *arXiv preprint arXiv:2304.12995*, 2023.

[193] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1179–1189, 2023.

[194] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[195] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin, "Pointllm: Empowering large language models to understand point clouds," *arXiv preprint arXiv:2308.16911*, 2023.

[196] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552–8562, 2022.

[197] N. Amenta, D. Attali, and O. Devillers, "Complexity of delaunay triangulation for points on lower-dimensional˜ polyhedra," in *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, pp. 1106–1113, 2007.

[198] B. O. COMMUNITY, "Blender - a 3d modelling and rendering package."

[199] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*, pp. 4904–4916, PMLR, 2021.

[200] Z. Wang, W. Liu, Q. He, X. Wu, and Z. Yi, “Clip-gen: Language-free training of a text-to-image generator with clip,” *arXiv preprint arXiv:2203.00386*, 2022.

[201] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *International Conference on Machine Learning*, pp. 8821–8831, PMLR, 2021.

[202] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 36479–36494, 2022.

[203] K. Thanikachalam and G. Khan, “Colorectal cancer and nutrition,” *Nutrients*, vol. 11, no. 1, p. 164, 2019.

[204] F. A. Haggar and R. P. Boushey, “Colorectal cancer epidemiology: incidence, mortality, survival, and risk factors,” *Clinics in colon and rectal surgery*, vol. 22, no. 04, pp. 191–197, 2009.

[205] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: a cancer journal for clinicians*, vol. 68, no. 6, pp. 394–424, 2018.

[206] American Cancer Society, “Cancer facts & figures 2015,” *Atlanta: American Cancer Society*, 2015.

[207] K. Simon, “Colorectal cancer development and advances in screening,” *Clinical interventions in aging*, vol. 11, p. 967, 2016.

[208] F. Stracci, M. Zorzi, and G. Grazzini, “Colorectal cancer screening: tests, strategies, and perspectives,” *Frontiers in public health*, vol. 2, p. 210, 2014.

[209] D. K. Rex, E. Y. Rahmani, J. H. Haseman, G. T. Lemmel, S. Kaster, and J. S. Buckley, "Relative sensitivity of colonoscopy and barium enema for detection of colorectal cancer in clinical practice," *Gastroenterology*, vol. 112, no. 1, pp. 17–23, 1997.

[210] J. Regula, M. Rupinski, E. Kraszewska, M. Polkowski, J. Pachlewski, J. Orlowska, M. P. Nowacki, and E. Butruk, "Colonoscopy in colorectal-cancer screening for detection of advanced neoplasia," *New England Journal of Medicine*, vol. 355, no. 18, pp. 1863–1872, 2006.

[211] H. Machida, Y. Sano, Y. Hamamoto, M. Muto, T. Kozu, H. Tajiri, and S. Yoshida, "Narrow-band imaging in the diagnosis of colorectal mucosal lesions: a pilot study," *Endoscopy*, vol. 36, no. 12, pp. 1094–1098, 2004.

[212] C.-C. Chang, C.-R. Hsieh, H.-Y. Lou, C.-L. Fang, C. Tiong, J.-J. Wang, I.-V. Wei, S.-C. Wu, J.-N. Chen, and Y.-H. Wang, "Comparative study of conventional colonoscopy, magnifying chromoendoscopy, and magnifying narrow-band imaging systems in the differential diagnosis of small colonic polyps between trainee and experienced endoscopist," *International journal of colorectal disease*, vol. 24, no. 12, p. 1413, 2009.

[213] C. D. Johnson, M.-H. Chen, A. Y. Toledano, J. P. Heiken, A. Dachman, M. D. Kuo, C. O. Menias, B. Siewert, J. I. Cheema, R. G. Obregon, *et al.*, "Accuracy of ct colonography for detection of large adenomas and cancers," *New England Journal of Medicine*, vol. 359, no. 12, pp. 1207–1217, 2008.

[214] E. M. Stoop, M. C. de Haan, T. R. de Wijkerslooth, P. M. Bossuyt, M. van Ballegooijen, C. Y. Nio, M. J. van de Vijver, K. Biermann, M. Thomeer, M. E. van Leerdam, *et al.*, "Participation and yield of colonoscopy versus non-cathartic ct colonography in population-based screening for colorectal cancer: a randomised controlled trial," *The lancet oncology*, vol. 13, no. 1, pp. 55–64, 2012.

[215] D. A. Lieberman, “Screening for colorectal cancer,” *New England Journal of Medicine*, vol. 361, no. 12, pp. 1179–1187, 2009.

[216] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain, “Wireless capsule endoscopy,” *Nature*, vol. 405, no. 6785, pp. 417–417, 2000.

[217] D. G. Adler and C. J. Gostout, “Wireless capsule endoscopy,” *Hospital Physician*, vol. 39, no. 5, pp. 14–22, 2003.

[218] M. Y. Chan, H. Cohen, and B. M. Spiegel, “Fewer polyps detected by colonoscopy as the day progresses at a veteran’s administration teaching hospital,” *Clinical Gastroenterology and Hepatology*, vol. 7, no. 11, pp. 1217–1223, 2009.

[219] A. Leufkens, M. Van Oijen, F. Vleggaar, and P. Siersema, “Factors influencing the miss rate of polyps in a back-to-back colonoscopy study,” *Endoscopy*, vol. 44, no. 05, pp. 470–475, 2012.

[220] S. G. Patel, P. Schoenfeld, H. M. Kim, E. K. Ward, A. Bansal, Y. Kim, L. Hosford, A. Myers, S. Foster, J. Craft, *et al.*, “Real-time characterization of diminutive colorectal polyp histology using narrow-band imaging: implications for the resect and discard strategy,” *Gastroenterology*, vol. 150, no. 2, pp. 406–418, 2016.

[221] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[222] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[223] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional

siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, 2016.

[224] U. Sajid and G. Wang, "Plug-and-play rescaling based crowd counting in static images," *arXiv preprint arXiv:2001.01786*, 2020.

[225] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[226] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *Proceedings of the IEEE international conference on computer vision*, pp. 1404–1412, 2015.

[227] J. Huo, J. Wu, J. Cao, and G. Wang, "Supervoxel based method for multi-atlas segmentation of brain mr images," *NeuroImage*, vol. 175, pp. 201–214, 2018.

[228] J. Bernal, J. Sánchez, and F. Vilarino, "Towards automatic polyp detection with a polyp appearance model," *Pattern Recognition*, vol. 45, no. 9, pp. 3166–3182, 2012.

[229] S. A. Karkanis, D. K. Iakovidis, D. E. Maroulis, D. A. Karras, and M. Tzivras, "Computer-aided tumor detection in endoscopic video using color wavelet features," *IEEE transactions on information technology in biomedicine*, vol. 7, no. 3, pp. 141–152, 2003.

[230] B. Taha, N. Werghi, and J. Dias, "Automatic polyp detection in endoscopy videos: A survey," in *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, pp. 233–240, IEEE, 2017.

[231] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

[232] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9259–9266, 2019.

[233] F. Cen, X. Zhao, W. Li, and G. Wang, "Deep feature augmentation for occluded image classification," *Pattern Recognition*, vol. 111, p. 107737, 2021.

[234] S. Park, M. Lee, and N. Kwak, "Polyp detection in colonoscopy videos using deeply-learned hierarchical features," *Seoul National University*, 2015.

[235] N. Tajbakhsh, S. R. Gurudu, and J. Liang, "Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pp. 79–83, IEEE, 2015.

[236] K. Patel, K. Li, K. Tao, Q. Wang, A. Bansal, A. Rastogi, and G. Wang, "A comparative study on polyp classification using convolutional neural networks," *PloS one*, vol. 15, no. 7, p. e0236452, 2020.

[237] H. R. Roth, C. T. Lee, H.-C. Shin, A. Seff, L. Kim, J. Yao, L. Lu, and R. M. Summers, "Anatomy-specific classification of medical images using deep convolutional nets," *arXiv preprint arXiv:1504.04003*, 2015.

[238] H. R. Roth, J. Yao, L. Lu, J. Stieger, J. E. Burns, and R. M. Summers, "Detection of sclerotic spine metastases via random aggregation of deep convolutional neural network classifications," in *Recent advances in computational methods and clinical applications for spine imaging*, pp. 3–12, Springer, 2015.

[239] G. Pappalardo, D. Allegra, F. Stanco, and G. M. Farinella, "On the exploitation of temporal redundancy to improve polyp detection in colonoscopy," in *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 58–63, IEEE, 2020.

[240] S. Mathew, S. Nadeem, S. Kumari, and A. Kaufman, "Augmenting colonoscopy using extended and directional cyclegan for lossy image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4696–4705, 2020.

[241] X. Mo, K. Tao, Q. Wang, and G. Wang, "An efficient approach for polyps detection in endoscopic videos based on faster r-cnn," in *2018 24th international conference on pattern recognition (ICPR)*, pp. 3929–3934, IEEE, 2018.

[242] K. Patel, A. M. Bur, and G. Wang, "Enhanced u-net: A feature enhancement network for polyp segmentation," *arXiv preprint arXiv:2105.00999*, 2021.

[243] P. Mesejo, D. Pizarro, A. Abergel, O. Rouquette, S. Beorchia, L. Poincloux, and A. Bartoli, "Computer-aided classification of gastrointestinal lesions in regular colonoscopy," *IEEE transactions on medical imaging*, vol. 35, no. 9, pp. 2051–2063, 2016.

[244] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339–3348, 2018.

[245] D. Jha, S. Ali, N. K. Tomar, H. D. Johansen, D. Johansen, J. Rittscher, M. A. Riegler, and P. Halvorsen, "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning," *Ieee Access*, vol. 9, pp. 40496–40510, 2021.

[246] P. H. Smedsrud, V. Thambawita, S. A. Hicks, H. Gjestang, O. O. Nedrejord, E. Næss, H. Borgli, D. Jha, T. J. D. Berstad, S. L. Eskeland, *et al.*, "Kvasir-capsule, a video capsule endoscopy dataset," *Scientific Data*, vol. 8, no. 1, pp. 1–10, 2021.

[247] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen, *et al.*, "Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, pp. 1–14, 2020.

[248] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. de Lange, D. Johansen, and H. D. Johansen, “Kvasir-seg: A segmented polyp dataset,” in *International Conference on Multimedia Modeling*, pp. 451–462, Springer, 2020.

[249] J. Bernal, N. Tajkbaksh, F. J. Sánchez, B. J. Matuszewski, H. Chen, L. Yu, Q. Angermann, O. Romain, B. Rustad, I. Balasingham, *et al.*, “Comparative validation of polyp detection methods in video colonoscopy: results from the miccai 2015 endoscopic vision challenge,” *IEEE transactions on medical imaging*, vol. 36, no. 6, pp. 1231–1249, 2017.