Final Report

Shi Yuchen
School of Data Science
Fudan University
21210980116

Jin Haoliang
School of Data Science
Fudan University
21210980041

Lin Ziyue
School of Data Science
Fudan University
21110980025

Abstract

This final assignment consist of three parts. The first assignment is to implement semantic segmentation model to each frame of a driving video downloaded from the Internet and visualize it. The second assignment is to train and test the object detection models Faster R-CNN on the VOC dataset and apply transfer learning method. The third assignment is to design transformer model and compare its performance with models applied in the midterm.

1 Introduction

Deep neural networks have demonstrated potential on a variety of computer vision related fields, such as image classification, object detection and semantic segmentation. For the last three decades, image segmentation has been one of the most difficult problems in computer vision, which is different from image classification or object detection in that it is not necessary to know what the visual concepts or objects are beforehand [1]. To be specific, an object classification will only classify objects that it has specific labels for such as horse, auto, house, dog. An ideal image segmentation algorithm will also segment unknown objects, that is, objects which are new or unknown. Semantic segmentation is a deep learning algorithm that associates labels or categories with each pixel of an image [2]. It is used to identify the set of pixels that constitute a distinguishable category (Figure 1). For example, self-driving cars need to recognize vehicles, pedestrians, traffic signals, sidewalks, and other road features. Semantic segmentation can be used in a variety of applications, such as autonomous driving, medical imaging, and industrial inspection.

(a) (b)

Figure 1: (a) Motorcycle racing image. (b) Segmentation for motorcycle racing image

Traditional machine learning is characterized by training data and testing data having the same input feature space and the same data distribution. If there is a difference in distribution between the training and test data, the performance of the learning model would be unsatisfactory [3]. Therefore, it is important to create a high-performance learner for a target domain trained from a related source domain, which is the motivation for transfer learning [4].

In practice, few people train the network from scratch because the dataset is not large enough. It is common to use a pre-trained network (e.g. a network such as AlexNet [5] trained on ImageNet for classification of 1000 classes) to fine-tune or as a feature extractor. Fine-tune method (shown in

Figure 2) usually freezes some of the convolutional layers of the pre-trained model (usually most of the convolutional layers near the input) and train the remaining convolutional layers (usually some of the convolutional layers near the output) and the fully connected layers.

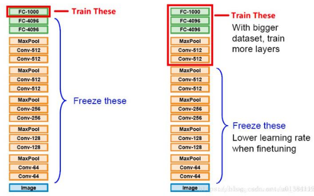


Figure 2: Some methods of transfer learning.

Transformer, first applied to the field of natural language processing, is a type of deep neural network mainly based on the self-attention mechanism, which achieved significant improvements [6]. Inspired by the major success of transformer architectures in the field of NLP, researchers have recently applied transformer to computer vision (CV) tasks [7]. Transformer has shown it is a potential alternative to CNN. Figure 3 shows the development timeline of vision transformer. ViT (Figure 6), a transformer model, applies a pure transformer directly to sequences of image patches to classify the full image [8]. In addition to image classification, transformer has been utilized to address a variety of other vision problems, including object detection [9], image processing [10], and video understanding [11]. Thanks to its exceptional performance, more and more researchers are proposing transformer-based models for improving a wide range of visual tasks.

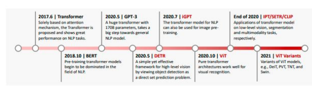


Figure 3: Key milestones in the development of transformer. The vision transformer models are marked in red.

2 Methodologies

In this section, we will discuss about the details of methodologies applied in three assignments.

2.1 Faster R-CNN Model

Faster R-CNN [12] is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [13] that uses the proposed regions. Faster R-CNN introduces a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network. RPNs are designed to efficiently predict region proposals with a wide range of scales and aspect ratios. By merging RPN and Fast R-CNN, the whole system is a single, unified network for object detection(Figure 4). The RPN component tells the unified network where to look, which works like "attention" mechanisms in neural network.

2.2 Mask R-CNN Model

Semantic segmentation and object detection are two very classical and important applications in the field of computer vision. In the field of semantic segmentation, FCN [14] is the representative algorithm, and in the field of object detection, the representative algorithm is Faster R-CNN [12]. It is natural to think that combining FCN and Faster R-CNN can not only be a model with both object detection and semantic segmentation, but also the two functions can complement each other to improve the model accuracy, which is the motivation of the proposed Mask R-CNN.

Mask R-CNN [15], extends Faster R-CNN [12] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression (Figure 5). Since Faster R-CNN was not designed for pixel-to-pixel alignment between network inputs and outputs. To fix the misalignment, it propose a simple, quantization-free layer, called RoIAlign, that faithfully preserves exact spatial locations.

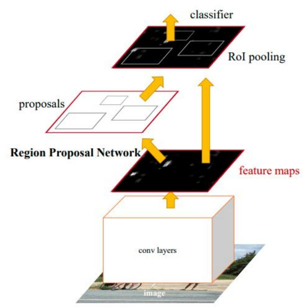


Figure 4: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

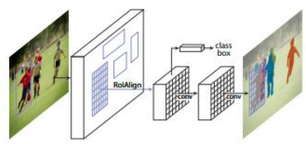


Figure 5: The Mask R-CNN framework for instance segmentation.

2.3 Vision Transformer Model

As shown in Figure 6, ViT split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, it used the standard approach of adding an extra learnable "classification token" to the sequence. When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets [16] of comparable size. However, the picture changes if the models are trained on larger datasets. ViT attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints.

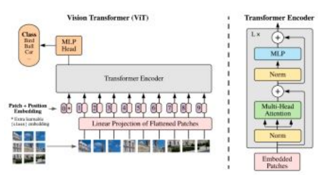


Figure 6: The framework of ViT.

3 Experiments

In this section, we report our major experimental results.

3.1 Dataset

The Pascal Visual Object Classes (VOC) challenge [17] is one of the benchmarks for supervised learning visual tasks. It provides a complete set of standardized and excellent datasets for image recognition and classification. All the objects in the VOC images are divided into 4 categories and subdivided into 20 classes, illustrated in Figure 7. **VOC2007** and **VOC2012** are two mostly used datasets. We used **VOC07+12** which trains on VOC2007 train+val dataset along with VOC2012 train+val dataset and tests on VOC2007 test dataset.

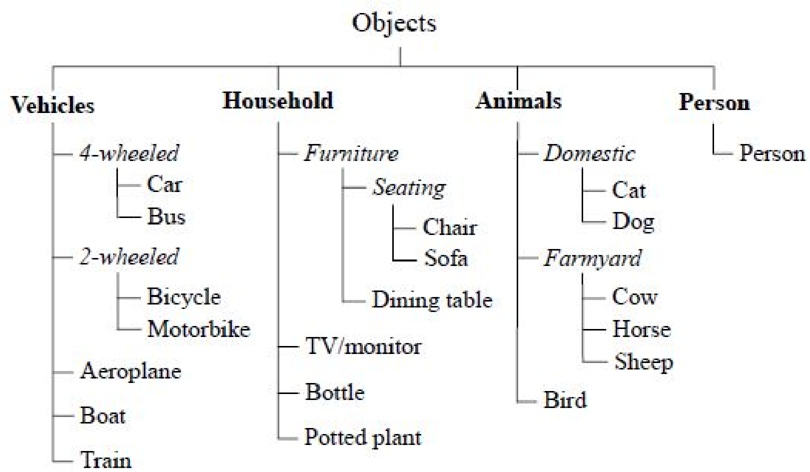


Figure 7: Object Classes in VOC Dataset (20 in total).

The CIFAR100 dataset has 100 classes. Each class has 600 color images of size 32 × 32, of which 500 are used as the training set and 100 as the test set. For each image, it has two labels, "fine" labels and "coarse" labels, which correspond to classes and superclasses in the Figure 8.

Superclass	**Classes**
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

Figure 8: The list of classes in the CIFAR-100.

3.2 Experimental Settings

For Assignment 1, we downloaded a trained Mask R-CNN model from Detectron2, tested it on every frame of a driving video downloaded from https://www.youtube.com/watch?v=3-DwOlaekow and visualized it.

For Assignment 2, we set Faster R-CNN with 16 batch size and 0.02 learning rate. The optimizer is SGD with momentum. The loss function is consisted of 4 parts, classification loss, bounding box regression loss, RPN classification loss and RPN regression loss. The iterations is about 80K and the metric is mAP.

For Assignment 3, we implemented vision transformer network. We set batch size as 128, initial learning rate as $1e-3$. Learning rate decay policy was CosineAnnealingLR with $eta\_min = 1e-5$

and warmup is from 0 to $1e-3$ in first epoch. The optimizer is Adam with $betas = (0.9, 0.999)$ and weight decay is $5e-5$. The epoch is 200. The loss function is CrossEntropyLoss. We used common metrics in classification tasks, including top1 error and top5 error to evaluate the results. The parameter of network is shown in Table 1

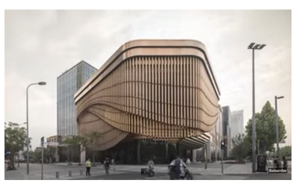
Table 1: Network parameter of Assignment 3

Parameter	Value
patch size	8
dim	512
depth	6
heads	6
mlp_dim	3072
dropout	0.1
emb_dropout	0.1

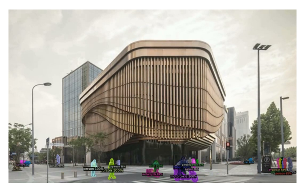
3.3 Result

3.3.1 Assignment 1

In this assignment, we completed semantic segmentation task. The input video downloaded from Youtube included passers-by, motorcycles, cars and so on. The output video can be found in https://github.com/Lightblues/NN-pj/blob/main/final/mask-RCNN/video-output.mkv. We show a frame of the video in Figure 9. It indicates that persons and motorcycles in the image are all recognized.



(a) Origin image

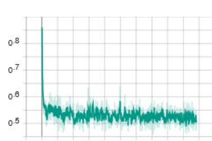


(b) Output image

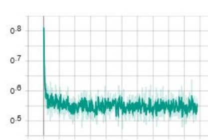
Figure 9: An example of semantic segmentation model result.

3.3.2 Assignment 2

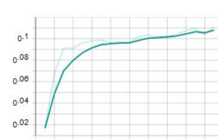
This assignment consists of three parts. The first part is random initialization training Faster R-CNN on VOC dataset. The second part is to pre-train backbone network on ImageNet dataset, then fine tune on VOC dataset. The third part is to initialize the backbone network of Faster R-CNN using backbone network parameters of Mask R-CNN trained by coco, and then fine tune using on VOC dataset. The curve of training loss, validation loss and mPA_0.5 of three parts are shown in Figure 10 11 12. They can easily indicate that the first model performed worst, which also demonstrates the effectiveness of transfer learning method. As for mPA_0.5, the third model performed best.



(a) Training loss

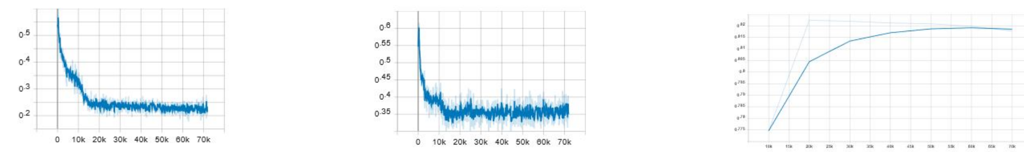


(b) Validation loss



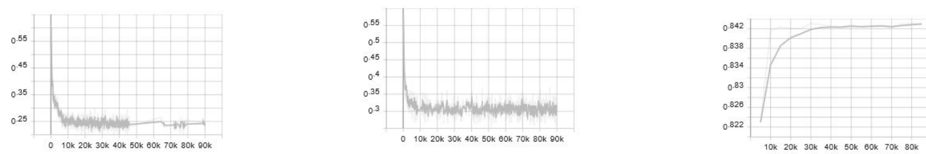
(c) The mAP_0.5

Figure 10: Results of part a.



(a) Training loss (b) Validation loss (c) The mAP_0.5

Figure 11: Results of part b.



(a) Training loss (b) Validation loss (c) The mAP_0.5

Figure 12: Results of part c.

What's more, we picked three images out of the VOC dataset to test the performance of three models. The output is shown in Figure 13. It also indicate the better performance of transfer learning method. As for the furniture and person images, the first model even don't detect any objects. Therefore, it would be of great help to stand on the shoulders of giants when training the model.

Figure 13: Bounding box of three parts.

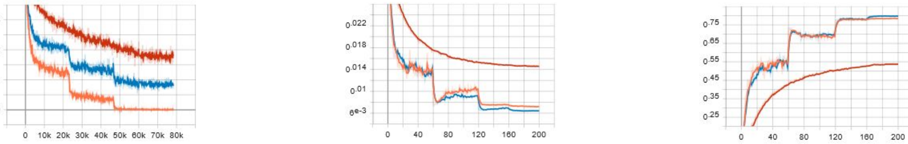
3.3.3 Assignment 3

We implemented vision transformer model on CIFAR-100 dataset with data augmentation method, Mixup. We compare the performances of this model with those trained in midterm project. We display curves of Resnet-50 without data augmentation, Resnet-50 with Mixup and ViT with Mixup in Figure 14. It can indicate that Resnet-50 outperforms ViT.

Furthermore, table 2 shows the top1 error and top5 error of ViT and 4 models in the midterm project. It also demonstrates that the performance of ViT is much worse than Resnet.

4 Conclusions

In this report, we complete three assignments. The first assignment is to implement semantic segmentation model to each frame of a driving video downloaded from the Internet and visualize it. The second assignment is to train and test the object detection models Faster R-CNN on the VOC dataset and apply transfer learning method. The third assignment is to design transformer model and compare its performance with models applied in the midterm.



(a) Training loss (b) Testing loss (c) Testing accuracy

Figure 14: The performances of three models. The orange line represents Resnet-50 without data augmentation, the blue line represents Resnet-50 with Mixup and the red line represents ViT with Mixup.

Table 2: Result of Assignment 3

Model	Parameters number	Data augmentation	Top1 error	Top5 error
ResNet-50	23,705,252	None	0.2147	0.0553
ResNet-50	-	Mixup	0.2039	0.0566
ResNet-50	-	Cutout	0.2196	0.0589
ResNet-50	-	CutMix	0.2131	0.0557
ViT	23,790,180	Mixup	0.4608	0.1994

Code and final model are uploaded in github and the links are listed below:

Code: https://github.com/Lightblues/NN-pj/tree/main/final

Model: https://github.com/Lightblues/NN-pj/releases

References

[1] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," vol. 7, no. 2, pp. 87–93.

[2] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1451–1460, 2018.

[3] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[4] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," vol. 3, no. 1, p. 9.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[7] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[9] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *CoRR*, vol. abs/2012.15840, 2020.

[10] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," *CoRR*, vol. abs/2012.00364, 2020.

[11] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," *CoRR*, vol. abs/1804.00819, 2018.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[13] R. Girshick, "Fast r-cnn," 2015.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[15] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," vol. 88, no. 2, pp. 303–338.