Machine Vision and Applications

Li Zonghui /李宗辉[*]
Student Number/学号: 61522122
lzh22@seu.edu.cn

Abstract

Computer vision is one of the most mature fields of artificial intelligence. The following first part tests student's basic understanding of computer vision and programming ability by image classification. The second part demonstrates student's report after reading some relevant papers on CV. All the code and results in this document are available at `https://github.com/mercurystraw/Cat-dog-img-classification`.

The First Part (Assignment)

1 Task 1: Image Classification

1.1 Requirements

Design an image classifier to classify a given dataset of cats and dogs, and output the accuracy of the classifier on trainset and testset.

1.2 Methods

In this experiment, the PyTorch framework is used to implement the image classification task. And the cat and dog dataset is imported locally.

I chose Convolutional Neural Network (ResNet18) as the model and set the relevant training parameters. During training, the accuracy (train_acc) and loss (train_loss) on the trainset are calculated by comparing the original label with the predicted label. After each epoch, I evaluate the testset and update the accuracy on the testset (test_acc).

Finally, the classifier prints the accuracy of the model on trainset and testset, and the trained model is saved for later use.

[*]School of A, Southeast University, Nanjing, China.

1.3 Experimental Details

Details of the experiment are as follows:

1. The config.json file is created to configure parameters of training the model, such as DATASET, MODEL_TYPE, EPOCH, BATCH_SIZE and LR, for easier adjustment. For this task, I set MODEL_TYPE as ’ResNet18’, EPOCH as 51, BATCH_SIZE as 32, and LR as 0.001.

2. For the given dataset, create a dataset class (CustomDataset) to initialize and iterate over the images (cat and dog) in the dataset. For each image , build a tuple containing the full path to the image and an index of the category to which the image belongs (cat for 0, dog for 1).

3. The model_zoo folder was built to keep serval constructed neural network models for invocation.

4. Load data: Built a load_data function to transform the image data to the size applicable for the model input, and do random_split (training set 80% and testset 20%), return trainset and testset.

5. Model training: Define the loss function as CrossEntropyLoss, the optimizer as Adam, and set the LR scheduler (halving the learning rate every 30 epochs helps dynamically adjust the learning rate during training to improve the training effect). For the image on the trainset, compare the original label and the test label, and do the same on the testset, then print the current train_acc and test_acc after each epoch. After the training is finished, save the model and print the final train_acc and test_acc results.

1.4 Experimental Results and Analysis

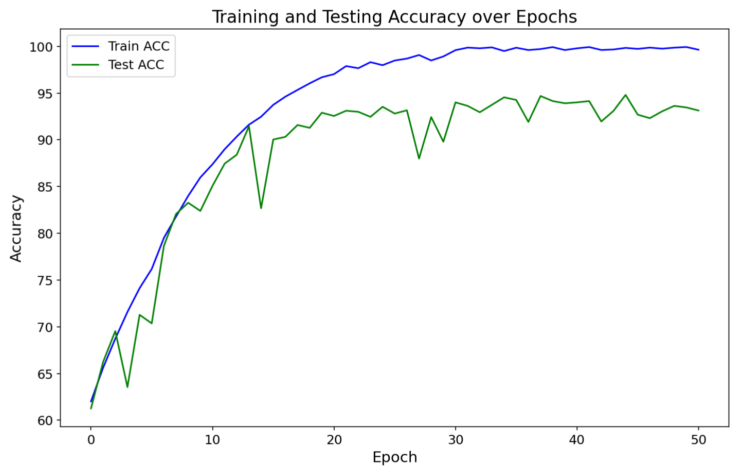

图 1: Train results

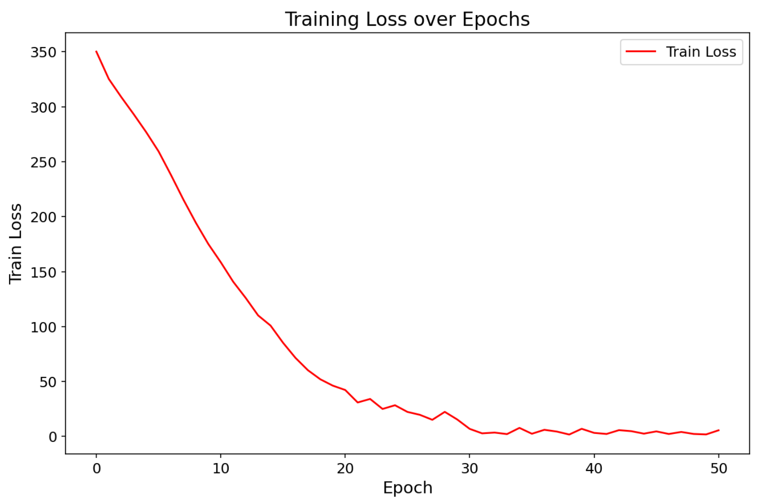

图 2: Train loss

The above two graphs demonstrate the accuracy of the classifier on trainset, testset, and the loss during the training process. With the increase of epoch, train_acc gradually increased to 99%, test_acc remained about 93%-94%, and the loss tended to 0 without too much change. I think this might be related to the random selection of training and testing sets, as this increases the uncertainty and fluctuation of the test results.

The final output after 51 epochs is:

Epoch: 50, Train ACC: 99.641, Test ACC: 93.136, Train Loss: 5.60930876

2 Task 2: Transfer Learning

2.1 Requirements

Build an image classifier. But this time, we need to use the transfer learning method, utilizing a pre-trained neural network, setting the parameters that the pre-trained network needs to update and adding new network layers. You can choose your own pre-training network (VGG, ResNet, etc.), Finally, you need to output the accuracy of model on trainset and testset.

2.2 Methods

The solution to Task2 is similar to Task1's, except that transfer learning is used to build the model and requires the use of a pre-trained neural network. The pre-trained ResNet50 is selected here, and the full-connection layer at the end of model is modified for mapping the input size 2048 to the output 2, which is suitable for binary classification tasks.

2.3 Experimental Details

Same as Task1, except that the Epoch was set to 35, and the pre-trained ResNet50 is used as the model.

2.4 Experimental Results and Analysis

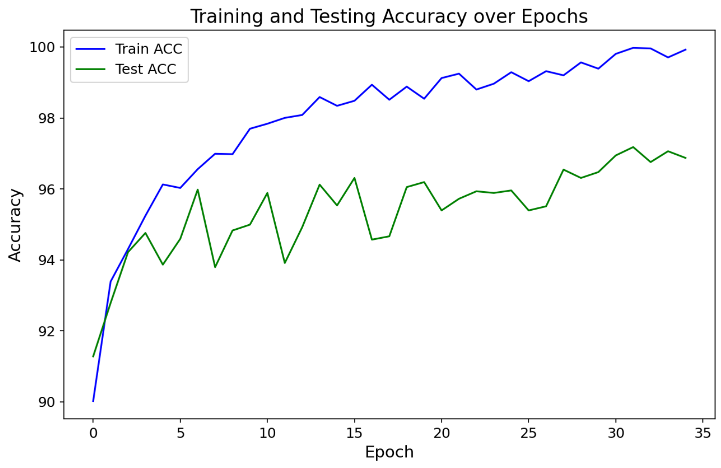

图 3: Train results

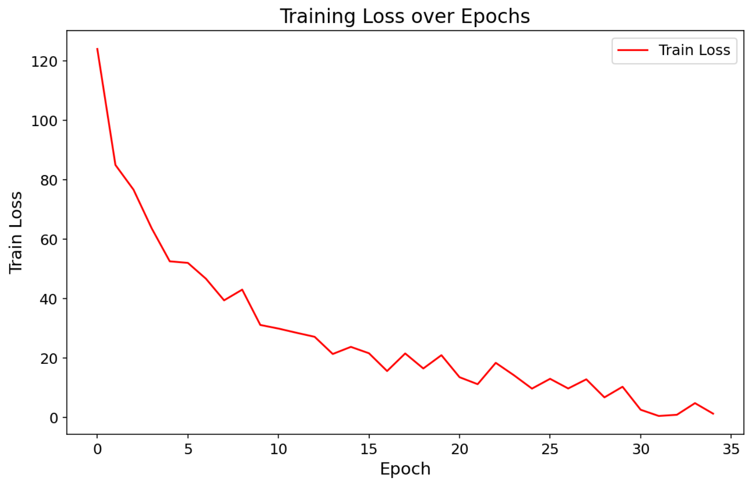

图 4: Train loss

From the above two images, it can be observed that using a pre-trained model yields an accuracy (acc) of 90% from the first epoch, and as the number of epoch increases, the train_acc gradually rises to 99%, while the test_acc gradually rises to 96%-97%. Additionally, the train_loss also gradually converges to 0. I think, by extensively training on large-scale datasets, pre-trained models have learned rich feature representations that are generic across various tasks. Consequently, when fine-tuned for specific tasks, these models can converge faster to optimal solutions, resulting in higher accuracy, lower error rates, and improved generalization capabilities on test sets. The final output after 35 epochs is:

```
Epoch: 34, Train ACC: 99.924, Test ACC: 96.874, Train Loss: 1.33042112
```

The Second Part (Report)

1 The first report

Our group chose one paper named 《NiteDR: Nighttime Image De-Raining with Cross-View Sensor Cooperative Learning for Dynamic Driving Scenes》 Cidan Shi, Lihuang Fang, Han Wu, Xiaoyu Xian, Yukai Shi, Liang Lin, Fellow, IEEE, 15 April 2024. to share. And my job is to explain Methodology part of the paper.

1.1 Background

In real-world environments, outdoor imaging systems are often affected by disturbances such as rain degradation.Especially, in nighttime driving scenes, insufficient and uneven-lighting shrouds the scenes in darkness, resulting degradation of both the image quality and visibility. To address these challenges, authors developed an image deraining framework tailored for rainy nighttime driving scenes. It aims to remove rain artifacts, enrich scene representation, and restore useful information.

1.2 Methodology

Images captured from different sensors, such as visible and infrared (IR) images, have complementary information and hold the potential for significant improvements in nighttime deraining tasks. Thus, authors aim to address the challenge of nighttime image de-raining by employingcross-view sensor information fusion.

They design a two-stage framework for information cleaning and fusion. The first-stage network performs information cleaning on the rainy-visible images and reconstructs clean outputs. These clean-visible images are subsequently fused with their corresponding infrared images across sensors in the second stage.

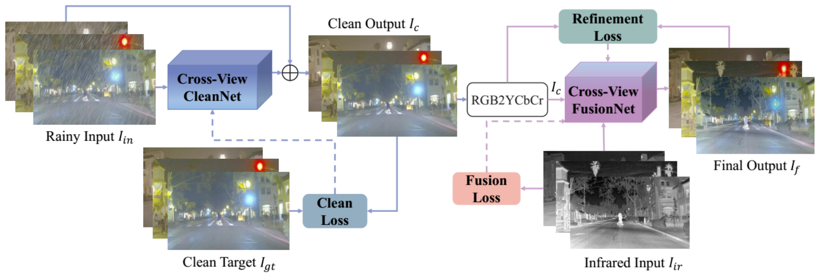

图 5: Overall framework

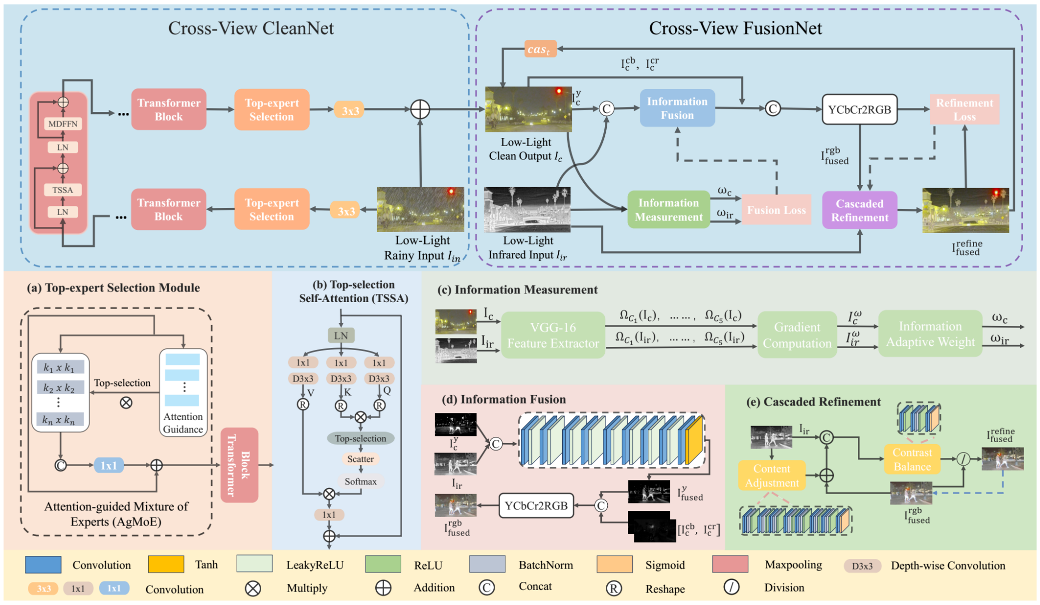

图 6: Overall architecture

For CleanNet, I mainly introduced three modules:

1. Top-selection self attention (TSSA): This module uses top-selection algorithm to construct a novel TSSA to capture long-range dependencies within global features more effectively.

2. Attention-guided mixture of experts (AgMoE): This module uses combination of multi-expert system and self-attention mechanism to better extract the local features of spatially varying rainwater distribution.

3. Mutual deformable feed-forward network (MDFFN): This module achieves multi-scale feature representations by cross-scale feature fusion via channel-wise concatenation, enhancing the extraction of rainfall degradation features and promoting rainwater removal.

For FusionNet, I mainly introduced two modules:

1. Information Measurement: This module is to preserve the critical information of input image. It performs on input clean image and infrared image, and keeps the two corresponding output values utilized in the fusion loss function to control the retention of different source information.

2. Information Fusion: This module is to fuse the two input features from clean image and infrared image, and generate the final output.

Besides, authors also proposed a cascaded refinement module for self-supervised optimization. It includes content adjustment and contrast balance structures. The fused image iteratively interacts with the infrared content to obtain contrast information, refining the image for contrast balance and enhancement. This iterative process is conducted in multiple cascaded stages. Each level of refined-fused image serves as the visible input for the next level of fusion, thus achieving a cascaded refinement.

At last, I introduced the loss funtion of the whole framework. Details are as follows in the table.

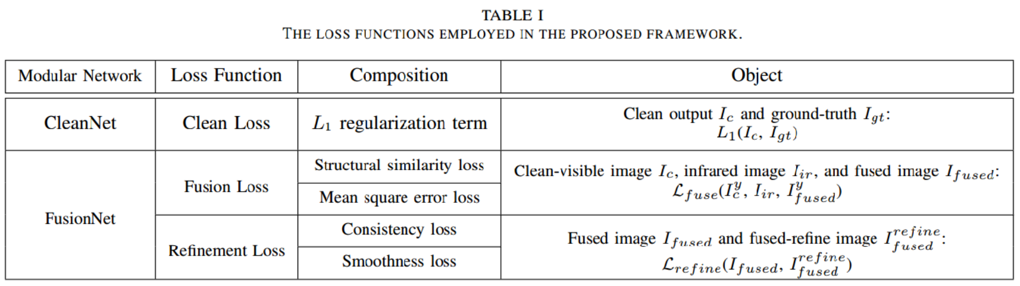

TABLE I
THE LOSS FUNCTIONS EMPLOYED IN THE PROPOSED FRAMEWORK.

Modular Network	Loss Function	Composition	Object
CleanNet	Clean Loss	L_1 regularization term	Clean output I_c and ground-truth I_{gt}: $L_1(I_c, I_{gt})$
FusionNet	Fusion Loss	Structural similarity loss	Clean-visible image I_c, infrared image I_{ir}, and fused image I_{fused}: $\mathcal{L}_{fuse}(I_c^y, I_{ir}, I_{fused}^y)$
		Mean square error loss	
	Refinement Loss	Consistency loss	Fused image I_{fused} and fused-refine image I_{fused}^{refine}: $\mathcal{L}_{refine}(I_{fused}, I_{fused}^{refine})$
		Smoothness loss	

图 7: Loss function

1.3 Summary

At first, we chose this paper because we were interested in the work of image deraining. After reading the paper, I think it inspires me a lot, introducing me to a whole new field and some innovative methods. I appreciated this deepened my understanding of CV.

2 The second report

Our group chose a survey about data augmention, introduced its background, methods, experimental effectiveness. I mainly introduced modern approaches used in data augmentation.

2.1 Background

Data augmentation is a technique that enables the creation of new training data by applying various transformations to the existing data. It is widely used in computer vision tasks to improve the generalization performance of models.

2.2 Modern Approaches

Modern approaches used in data augmentation include:

1. Input sqpce data augmentation: a class of approaches that directly modify the input image (or parts of it) for the purpose of creating variability to improve generalization.

It mainly includes: geometric transformations, photometric transformations, deeply learned transformation operations for sample-space data augmentation, region-level data augmentation.

2. Feature space data augmentation approaches: transform data in the feature space in ways that encourage the resulting deep learning model to learn representations that are invariant to image transformations in the input space.

It mainly includes: deep feature transformation, feature addition and mixing, feature elimination.

3. Image synthesis approaches: use artificially generated data for training machine learning models.

It mainly includes: computer graphics modeling, augmentation with synthetic 2D images and static 3D scenes, data augmentation based on dynamic 3D scene modeling, neural rendering, generative adversarial network (GAN).

4. Meta-learning approaches:

It mainly includes: Meta-learning approaches and Optimization-based methods.

2.3 Summary

After reading the survey on data augmentation, I gained a deeper understanding of the importance of data in deep learning systems. The quality of data directly affects the performance and reliability of models, making data augmentation particularly crucial when training data is limited or of poor quality. The article provides a comprehensive overview of data augmentation methods in the field of computer vision, covering both traditional approaches and modern techniques, including region-level transformations, feature space augmentation, and data synthesis methods. Notably, it discusses cutting-edge techniques such as CAD modeling, neural style transfer, generative modeling, and neural rendering. These methods offer researchers and developers diverse options to address the needs of various tasks. Overall, this survey provided me with a comprehensive understanding of the complexity and diversity of data augmentation, while also inspiring me to think about how to effectively leverage these techniques in practical applications.

End of Document