

Configuration Manual

Madhu Vanthi Sankar Ganesh

x23310928

Supervisor : John Kelly

Introduction

This document elaborates the system specification, the software and hardware used for implementation of the project. This document also lays out the steps carried out in the implementation of the research project “Enhancing Monocular Depth Estimation with Self-Supervised Learning and Transformer-Based Architectures”.

System Configuration

2.1 Software Specification:

- Windows laptop with AMD Ryzen processor where the code and training is done.
- Visual Studio Code , which is an open source application where the python codes are edited and run.
- Conda environment : Created a conda environment with necessary packages for running the training and evaluation of the project.

2.2 Hardware Specification:

- HP Victus laptop with 500GB ssd , 16 GB RAM , 5 GB GPU
- AMD processor 5000 series.

Data and Code Generation Steps:

- Download the scripts provided by Hoang Chuong Nguyen into your local machine.
- The conda commands for generating the new environment, downloading the dataset and installing necessary packages is given in the github repo.

Code Modifications:

- Created a new model folder that integrates transformers with DiffNet for training.

- Modifies the trainer folder with all the trainer files for including the `transformer_diffnet`

```
# Initialize pose network
if self.pose_net_type == 'transformer_diffnet':
    self.pose_encoder = ManyDepthResnetEncoder(18, pretrained=True, num_input_images=2, use_transformer=True)
    self.pose_decoder = ManyPoseDecoder(
        num_ch_enc=self.pose_encoder.num_ch_enc,
        num_input_features=1,
        num_frames_to_predict_for=2
    )
    self.pose_net = DiffPoseNet(self.pose_encoder, self.pose_decoder)
elif self.pose_net_type in ['manydepth', 'monodepth2', 'diffnet', 'brnet']:
    self.pose_encoder = ManyDepthResnetEncoder(num_layers=18, pretrained=True, num_input_images=2)
    self.pose_decoder = ManyPoseDecoder(
        num_ch_enc=self.pose_encoder.num_ch_enc,
        num_input_features=1,
        num_frames_to_predict_for=2
    )
    self.pose_net = ManyPoseNet(self.pose_encoder, self.pose_decoder)
elif self.pose_net_type == 'packnet':
    self.pose_net = PackNetPoseResNet(version='18pt')
else:
    raise NotImplementedError(f"Pose network type {self.pose_net_type} not implemented")

# Initialize RAFT with correct parameters
self.raft = RAFT(Namespace(small=True, dropout=True, mixed_precision=True))
```

- The yaml file used is also modified as shown below to update the file paths and the model path:

```
# ----- Shared params -----
use_pretrained_mask: True
use_mixed_precision: False # Disable to avoid compatibility issues
depth_net_type: 'transformer_diffnet' # Add missing depth_net_type
pose_net_type: 'transformer_diffnet' # Keep transformer_diffnet consistent
# Dataloading
height: 192
width: 512
img_ext: "png"
train_data_dir: "D:/mono_depth_cvpr24/data/cityscapes/cityscapes_512x1024"
train_file_path: "./splits/cityscapes_preprocessed/train_files_reduced6.txt"
val_data_dir: "C:/Users/madhu/data_mono/cityscapes_val_512x1024_1/cityscapes_val_512x1024"
val_file_path: "./splits/cityscapes_preprocessed/val_files_reduced2.txt"

val_depth_data_dir: "D:/mono_depth_cvpr24/data/cityscapes/leftImg8bit_sequence/leftImg8bit_sequence"
val_depth_file_path: "./splits/cityscapes/val_files_reduced3.txt"
val_gt_depth_path: "C:/Users/madhu/data_mono/gt_depths/gt_depths/val_gt_depths"
val_gt_mask_path: "C:/Users/madhu/data_mono/gt_dynamic_object_mask/val_mask_class"

test_depth_data_dir: "D:/mono_depth_cvpr24/data/cityscapes/leftImg8bit_sequence/leftImg8bit_sequence"
test_depth_file_path: "./splits/cityscapes/test_files_reduced1.txt"
test_gt_depth_path: "C:/Users/madhu/data_mono/gt_depths/gt_depths/test_gt_depths"
test_gt_mask_path: "C:/Users/madhu/data_mono/gt_dynamic_object_mask/test_mask_class"
# Training setting
nb_gpus: 1
num_workers: 8 # Add missing num_workers parameter
batch_size: 4 # Add missing batch_size parameter

# Logging
log_path: "D:/mono_depth_cvpr24/data/cityscapes/checkpoints/cityscapes/transformer_diffnet"
pixelwise_depthnet_load_weights_folder: null

# ----- Stage 2: Object depthnet -----
raw_mask_path: "D:/mono_depth_cvpr24/data/cityscapes/SEEM_mask/masks"
raw_road_mask_path: "D:/mono_depth_cvpr24/data/cityscapes/SEEM_mask/road_masks"
object_depthnet_load_weights_folder: null
```

- Each trainer file is also modified to include choices to train from a specific weight , start fresh or directly go to the evaluation:

```

Stage 7 checkpoints found:
1. weights (Best Model)
2. weights_11000 (Epoch 11000)
3. weights_10000 (Epoch 10000)
4. weights_9000 (Epoch 9000)
5. weights_8000 (Epoch 8000)
6. weights_7000 (Epoch 7000)
7. weights_6000 (Epoch 6000)
8. weights_5000 (Epoch 5000)
9. weights_4000 (Epoch 4000)
10. weights_3000 (Epoch 3000)
11. weights_2000 (Epoch 2000)
12. weights_1000 (Epoch 1000)
13. Start fresh (overwrite all)
14. Start from epoch 0 (keep existing)
15. Skip this stage
16. Generate outputs with best weights (skip training)

```

Implementing the Code:

- There are 4 stages of training this architecture for depth estimation. The first stage is pixel-wise depth estimation , to run it paste the following command after the prerequisite commands below:

```

(base) C:\Users\madhu>conda activate mono_consistent_depth
(mono_consistent_depth) C:\Users\madhu>cd repo
(mono_consistent_depth) C:\Users\madhu\repo>cd mono-consistent-depth
(mono_consistent_depth) C:\Users\madhu\repo\mono-consistent-depth>set PYTORCH_CUDA_ALLOC_CONF=max_split_size_mb:128
(mono_consistent_depth) C:\Users\madhu\repo\mono-consistent-depth>set PYTORCH_NO_CUDA_MEMORY_CACHING=1
(mono_consistent_depth) C:\Users\madhu\repo\mono-consistent-depth>python runner.py ./configs/cityscapes/diffnet_pretrain
ed_mask.yaml --stage 1
Starting Multi-Stage Depth Estimation Pipeline
Stages to run: [1]
Project path: D:/mono_depth_cvpr24/data/cityscapes/checkpoints/cityscapes/transformer_diffnet
The system will automatically detect existing checkpoints and offer resumption options

Preparing to run Stage 1...
Found 8 checkpoints for Stage 1:
weights_0 (Epoch 0)
weights (Best Model)
weights_1 (Epoch 1)
weights_2 (Epoch 2)
weights_3 (Epoch 3)
weights_4 (Epoch 4)
weights_6 (Epoch 6)
weights_8 (Epoch 8)

```

Anaconda Prompt Commands:

```

conda activate mono_consistent_depth
cd repo
cd mono-consistent-depth
set PYTORCH_CUDA_ALLOC_CONF=max_split_size_mb:128
set PYTORCH_NO_CUDA_MEMORY_CACHING=1
set PYTORCH_NO_CUDA_MEMORY_CACHING=1
python runner.py ./configs/cityscapes/diffnet_pretrained_mask.yaml --stage 1

```

- Do the same , just change stage 1 to stage 5 , then 6 and finally 7.
 - The [runner.py](#) file is what combines all the stages in the trainer folder and runs it in a sequence. It is also possible to run the command as below :
- ```
python runner.py ./configs/cityscapes/diffnet_pretrained_mask.yaml
```

- This command will automatically run from stage 1 to 5 ,6 and 7 on its own but the code was modified to include stage wise training.
- To run transformer\_diffnet in the yaml file do the below:

```
----- Shared params -----
use_pretrained_mask: True
use_mixed_precision: False # Disable to avoid compatibility issues
depth_net_type: 'transformer_diffnet' # Add missing depth_net_type
pose_net_type: 'transformer_diffnet' # Keep transformer_diffnet consistent
```

- To change to diffnet which was introduced in the base paper where the base code was retrieved, just change the above depth\_net type and pose\_net type to “diffnet”.

## Evaluation:

The final evaluation is done after training the stage 7 and the results for training for 10 epochs transformer\_diffnet is below :

```
Evaluating depth: 100% | 40/40 [00:21<00:00, 1.87it/s]
Evaluation completed: 40/40 batches processed successfully
100% | 157/157 [00:21<00:00, 7.42it/s]
Final Depth Evaluation (All Region): [0.26909762 2.81681554 10.34886944 0.34123774 0.5490215 0.81549225
0.94070489]
Final Depth Evaluation (Static): [0.26692051 2.83373229 10.50183656 0.34240108 0.54837782 0.81448846
0.94040035]
Final Depth Evaluation (Dynamic): [0.28466628 2.90621618 7.21683676 0.29396813 0.5761824 0.84553119
0.93735464]

=====
🔥 REGENERATED FINAL EVALUATION RESULTS
=====
***** All region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.269 & 2.817 & 10.349 & 0.341 & 0.549 & 0.815 & 0.941 \\
***** Static region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.267 & 2.834 & 10.502 & 0.342 & 0.548 & 0.814 & 0.940 \\
***** Dynamic region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.285 & 2.906 & 7.217 & 0.294 & 0.576 & 0.846 & 0.937 \\

=====
📄 Regenerated results saved to: D:/mono_depth_cvpr24/data/cityscapes/checkpoints/cityscapes/transformer_diffnet/final_depthnet/final_evaluation_results.txt
✅ Stage 7 outputs regenerated successfully!
✅ All requested stages completed successfully!
```

For DiffNet :

```
Evaluating depth: 100% | 40/40 [00:27<00:00, 1.47it/s]
Evaluation completed: 40/40 batches processed successfully
100% | 157/157 [00:22<00:00, 6.98it/s]
Final Depth Evaluation (All Region): [0.28178386 2.53807083 8.93980058 0.32539875 0.53868286 0.81985655
0.95396593]
Final Depth Evaluation (Static): [0.27643266 2.47316057 9.00473391 0.32297848 0.54052586 0.82155127
0.95640227]
Final Depth Evaluation (Dynamic): [0.33873427 3.65085884 7.74130904 0.32326487 0.53390088 0.82025988
0.91805826]

=====
🔥 REGENERATED FINAL EVALUATION RESULTS WITH DIFFNET
=====
***** All region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.282 & 2.538 & 8.940 & 0.325 & 0.539 & 0.820 & 0.954 \\
***** Static region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.276 & 2.473 & 9.005 & 0.323 & 0.541 & 0.822 & 0.956 \\
***** Dynamic region *****
abs_rel | sq_rel | rmse | rmse_log | a1 | a2 | a3 |
& 0.339 & 3.651 & 7.741 & 0.323 & 0.534 & 0.820 & 0.918 \\

=====
```

## References:

<https://github.com/HoangChuongNguyen/mono-consistent-depth>