

Assignment 3

Robot Vision practical

Friedrich Fraundorfer (fraundorfer@icg.tugraz.at)

Deadline: June 12, 2024, 23:55h

Questions: If you have any questions about the exercises send a mail with subject prefix: “[RV KU]” to marco.pfleger@student.tugraz.at.

Submission: Submit the completed assignment (code, output data and report) using the TeachCenter submission system as a **single** ZIP File. Documents need to be submitted using the PDF format.

Compulsory submission file structure:

assign3.zip/	<i>root directory</i>
└ code/	<i>contains all code files</i>
└ assign3_task1.py	
└ assign3_task2.py	
└ assign3_task3.py	
└ assign3_task4.py	
└ report/	<i>contains a single PDF</i>
└ assign3_report.pdf	
└ fileoutput/	<i>stores all relevant output</i>

Before getting started:

Clone the repository <https://github.com/nianticlabs/monodepth2>, apply the provided .patch file (available in file data_ass3) and download the KITTI training data as described in the repository's README.

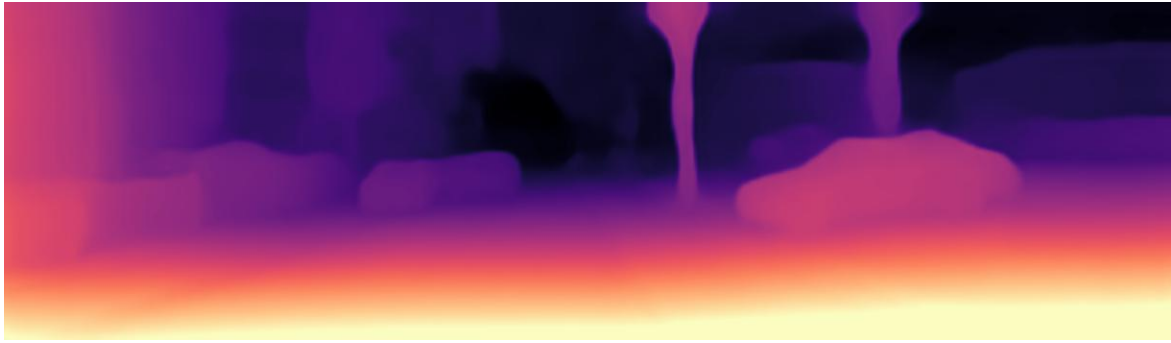
Task 1: Predict depth of given images with a Deep Network (5pts)

The goal of this task is to use a monocular depth prediction network to predict a depth image for the given set of images. We will be using the monodepth2 network, which is a state-of-the-art network for depth prediction as per the KITTI benchmark. - <https://github.com/nianticlabs/monodepth2>

KITTI Depth Vision benchmark:

http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction





You are to use the pretrained model *mono+stereo_640x192* (which has a direct scaling factor provided to convert depth to meters) to run inference on the provided directory of 10 randomly chosen KITTI images.

Deliverables:

Include your prediction as a .png file, both in the report and separately. In the report, comment on how the network performs with respect to the ground truth, i.e. if it performs badly in a certain region, etc. You should perform this evaluation for the image **2011_10_03_drive_0047_sync_image_0000000005_image_03.png**.


Task 2: Compute error between prediction and ground truth (5 pts)

Using your predictions from the previous task, please compute the error between the prediction and ground truth images using the Root Mean Squared Error (RMSE).

You will have to convert your prediction into the metric scale. This scaling factor is provided in the evaluation script (`evaluate_depth.py`) of `monodepth2` for the pretrained networks.

Deliverables:

Report the average RMSE per image between the predicted depth maps and the ground truths for each of the 10 images provided.

 **Hint:**
evaluate_depth.py also contains functions useful to evaluate RMSE and several other types of errors.

Task 3: Compare the results with a finetuned model (10 pts)

Given a pre-trained model it is possible to improve the performance initializing the training with the model weights and continuing to train.

Use the weights of *mono+stereo_640x192* to initialize the training and utilize the *SmallRAWDataset* with the *small* split as the training / validation data. Run the finetuning for 10 epochs. Once you've completed the finetuning evaluate the resulting model as you have done in Task 2.

Deliverables:

Report the average RMSE per image, between the predicted depth maps and the ground truths for

each of the 10 images provided. Compare each of the results with the non-finetuned results of *mono+stereo_640x192*, how much did the performance improve?

Additionally, report the log of your training into the report, the improvements per epoch should be visible.



Hint:

Google Colab's runtime likes to disconnect after some time, which may lead to the loss of the files. Make sure to back-up.

Task 4: Compare the quality of the monodepth method and a stereo depth method (10 pts)

For this comparison results from mono depth (use the model specified in Task 1) and the Unimatch stereo method are compared against ground truth (GT). For this, evaluations have to be performed on stereo rectified images with appropriate GT (available in file *data_ass3*). The GT is given as disparity values and valid for the left image which are from camera 2 and monodepth needs to be run on the images of camera 2 as well. Utilizing the Unimatch method from previous Assignment 1 Task 3, compute the disparity image from the provided stereo images. There is no need for additional intrinsic or extrinsic calibration files, the given images are already stereo-rectified.

To acquire depth images, you must transform the disparity values into metric depth values using the following geometric relation:

$$Z = f \frac{B}{d}$$

Compute the depth values Z for every disparity value using the equation above. The values for focal length and base line can be found in the file *cameracalibrationdata.txt*.

The evaluation should only consider valid pixels, pixels which are valid in the ground truth AND in the estimated depth maps. Certain estimated depth values could be higher than realistic depth values. Cutoff and invalidate any depth values that exceed 120 meters.

Using your stereo depth image, compute the error between the computed depth and ground truth image using the familiar Root Mean Squared Error (RMSE). By using a mask, ensure that the difference between depth and ground truth image is computed for valid values only.

Next, compute a difference image (absolute difference in depth [m] to GT) for the results from the mono depth method as well as the previously obtained stereo depth image to the provided ground truth image. The output should be a greyscale difference image. Make sure to apply the same scaling for both images. Be aware to compute the differences again for valid values only and set the excluded values to 0 in the output image.

To gain better understanding of the error distribution, a histogram plot of the errors (absolute difference in depth [m] to GT) should be created with a *resolution of 10 cm per bin*. The range of the histogram should be the *error distribution from 0 to 10 m*.

Deliverables:

Report the obtained disparity images and the RMSE values for the provided 10 images. Compare the computed RMSE against the result from the mono prediction in Task 2. Provide difference images for both the stereo depth images and the mono depth images as a .png file. Also provide your thoughts on the differences. Moreover, report the error distribution in histogram plots for the mono and stereo case. Put all the results and visualizations into the report.

Other information:

It is advised to use Google Colab to prototype your chosen networks. Colab provides a GPU enabled environment preinstalled with the latest deep learning libraries (TensorFlow, PyTorch, etc.). Colab can be used as an enhanced Jupyter Notebook with the latest ML libraries and CUDA toolkits preinstalled – indeed, Colab runs .ipynb files.

Here is a video tutorial on the basics of Colab:

<https://www.youtube.com/watch?v=vVe648dJOdl&t=625s>

The finetuning script for Task 3 requires a very specific Pytorch environment. You will be provided a Colab notebook to set the environment for this task correctly. It is only needed for Task 3.

Here is a link to a GitHub repository that points to several Colab Notebooks with ready to use implementations of many state of the art solutions to Computer Vision tasks.

<https://github.com/tugstugi/dl-colab-notebooks>

**Hint:**

Google Colab's GPU feature is disabled by default, enable it by setting the Hardware accelerator in the notebook settings.