Deep Learning for Autonomous Driving

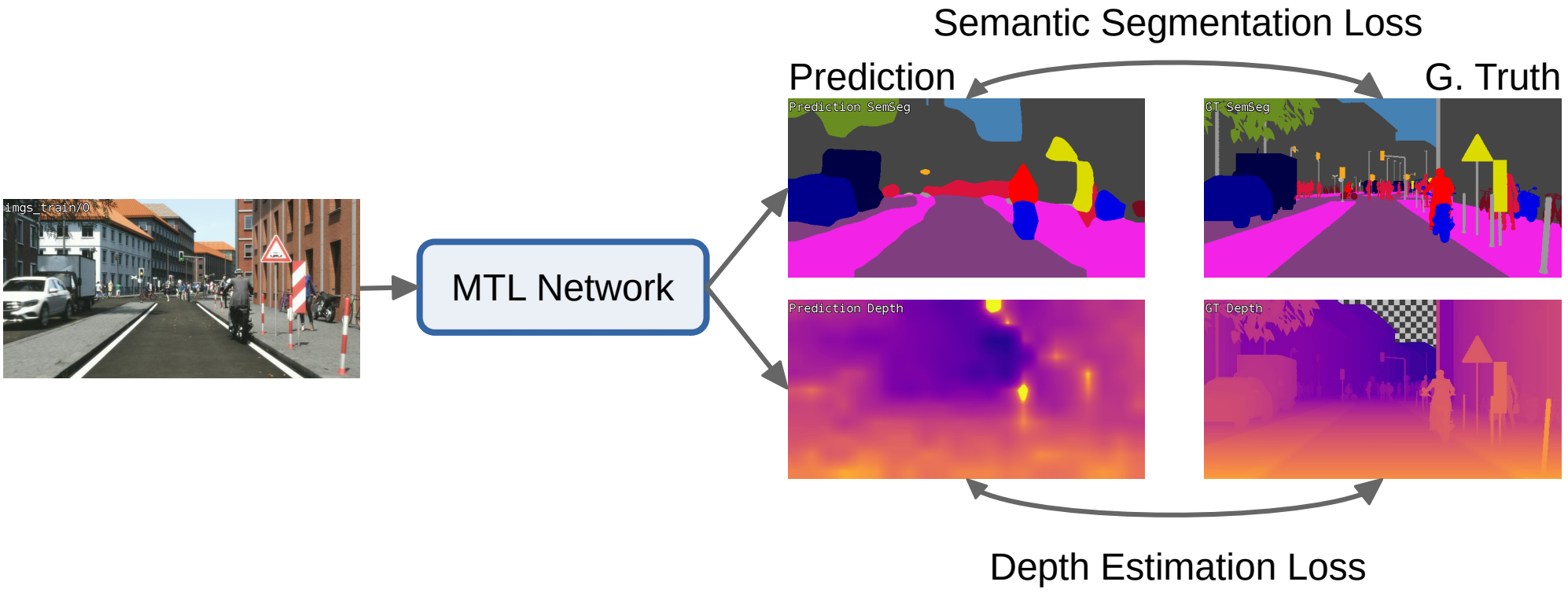
Project 2: Multi-Task Learning for Semantics and Depth

Lukas Hoyer, Qin Wang, Anton Obukhov, Dengxin Dai

ETH zürich

CVL Computer Vision Lab

Multi-Task Learning (MTL)



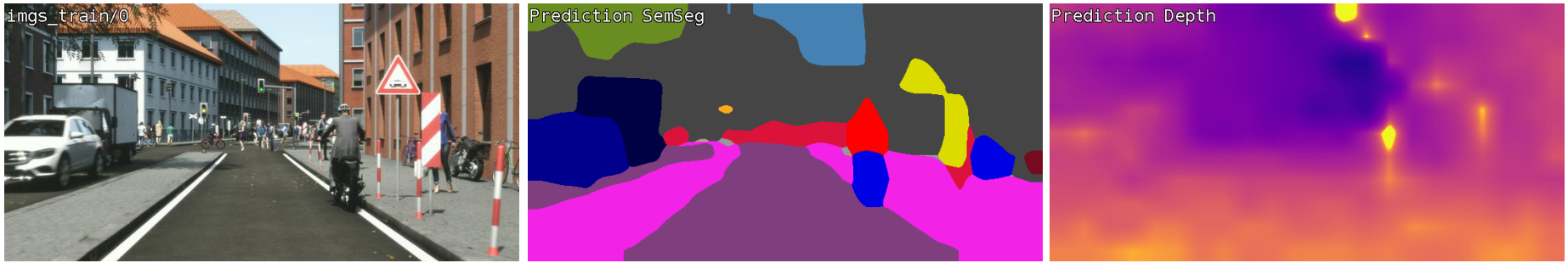
Learning Objectives

- Apply concepts of the lecture in practice
- Compose networks with multiple task branches
- Implement a selection of advanced network modules
- Train networks in the cloud with AWS
- Explore and tune hyperparameters
- Ablate and understand network components
- Present your results, observations, and discussion in a report

Solution Template

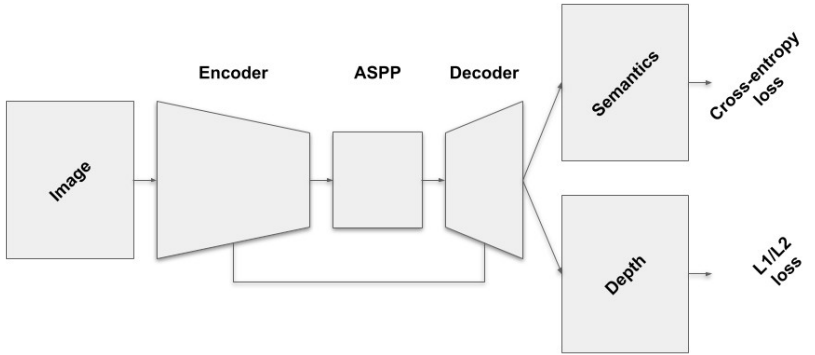
- Functional solution template: https://gitlab.ethz.ch/dlad21/exercise2
- Quite poor MTL performance
- Starting point for exercise 2

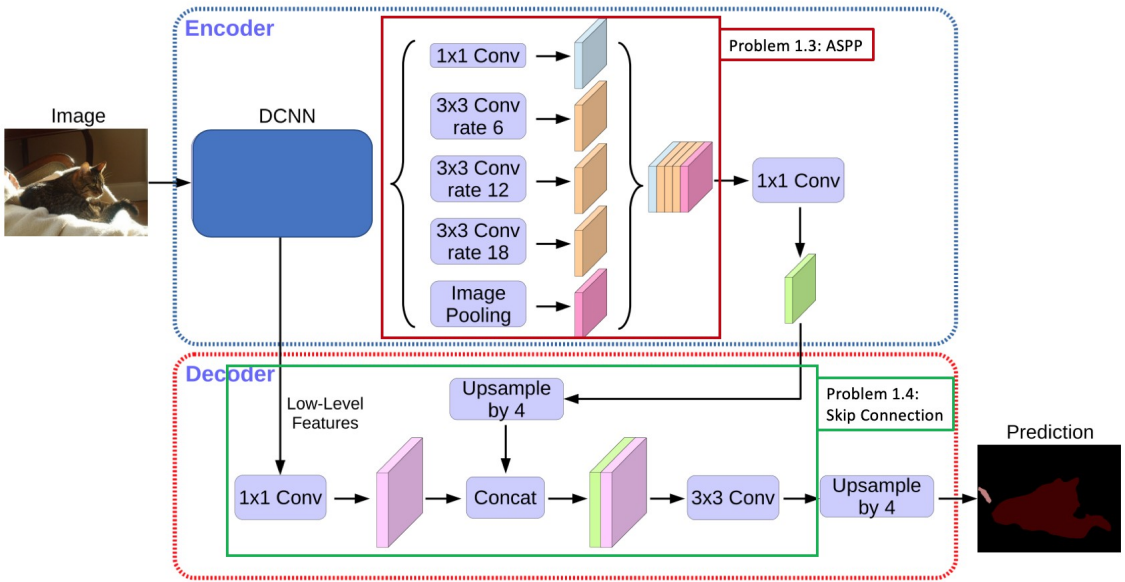
Solution template without modifications



Problem 1: Joint Architecture

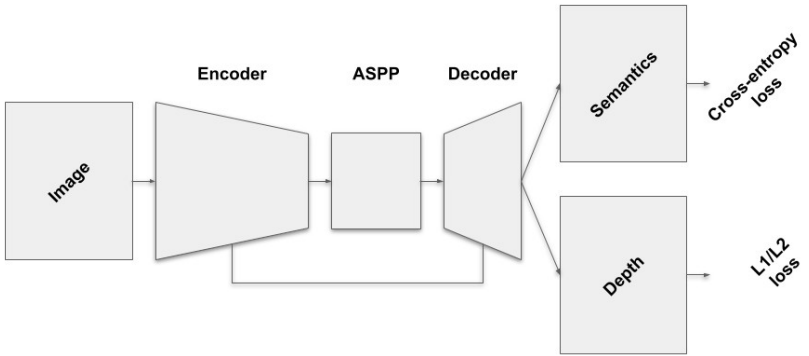
- Hyperparameters
- Encoder Initialization
- Dilated Convolutions
- ASPP
- Skip Connections





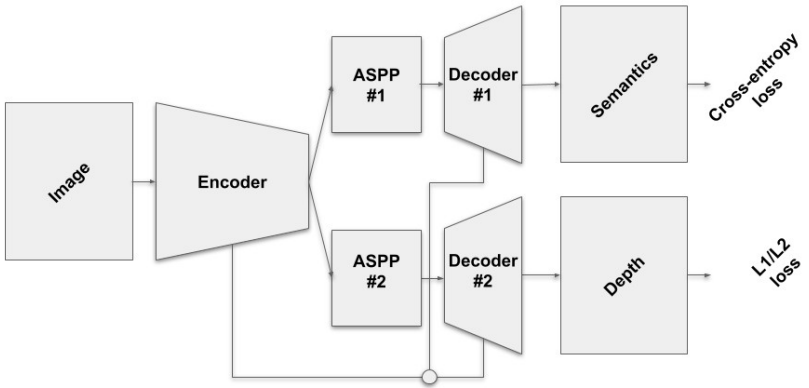
Problem 2: Branched Architecture

Joint Architecture



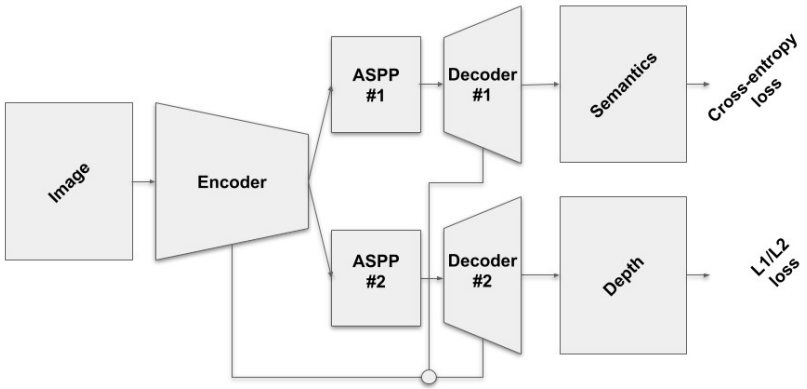
Introduce Task-Specific Decoder

Branched Architecture



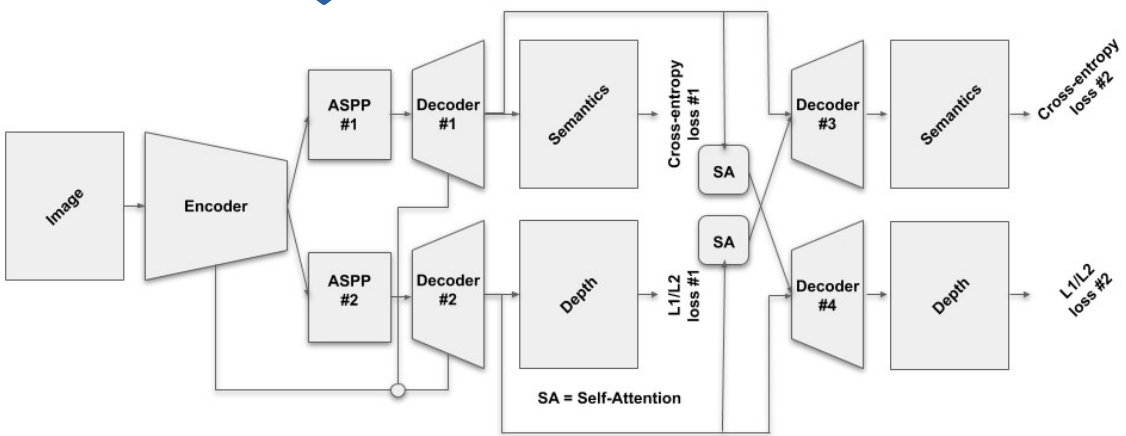
Problem 3: Task Distillation

Branched Architecture



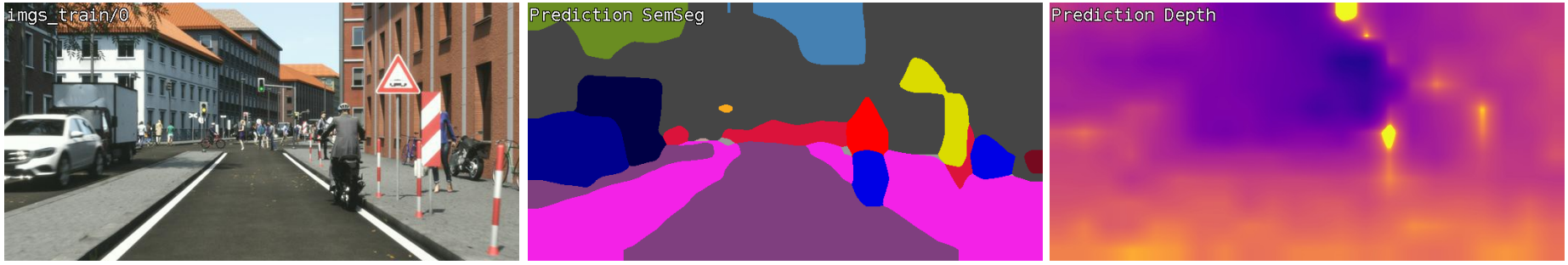
Introduce Cross-Task Distillation

Task Distillation



What you will achieve

Solution template without modifications

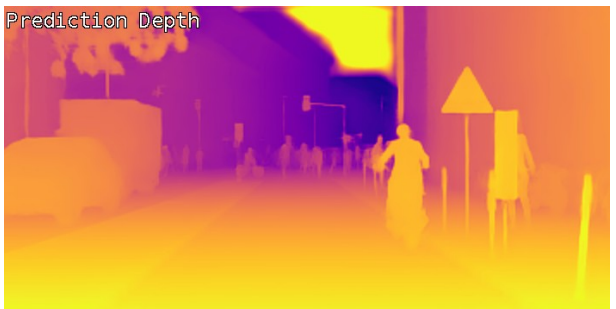


After completing problem 3 (results can vary)

Problem 4: Open Challenge

- Explore your own ideas to improve MTL!
- Inspirations are provided in the handout
- Grading
  - MTL performance
  - Implementation effort
  - In-depth analysis (ablations, parameter study, discussion, ...)
  - Discussion of related works

My personal best results. Can you beat it?



Training on AWS

- Setup AWS: https://gitlab.ethz.ch/dlad21/aws-tools
- Launch training from exercise 2 folder:
  `python aws_start_instance.py`
  - Prompts team information
  - Creates AWS bucket
  - Start AWS instance
  - Syncs local source code to instance
  - Starts training (with Weights & Biases monitoring)
  - Provides ssh access information
- Monitor AWS instances:
  https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Instances
- Further details: https://gitlab.ethz.ch/dlad21/exercise2/

Weights & Biases (wandb)

Example predictions

Performance Curves

Weights & Biases (wandb)

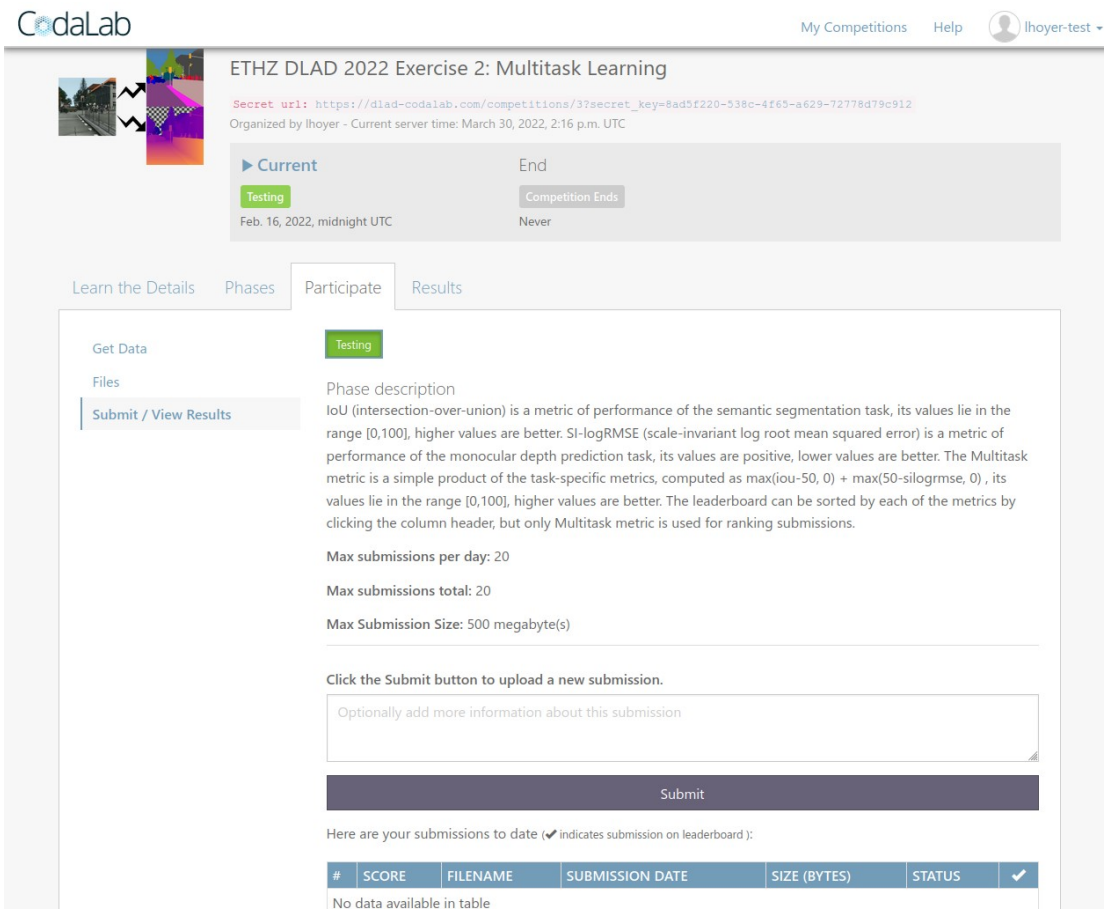
Export for W&B sheet
(part of project submission)

Select visible columns

Overview of Training Runs

- Commands & Links for AWS
- Hyperparameters
- Metrics

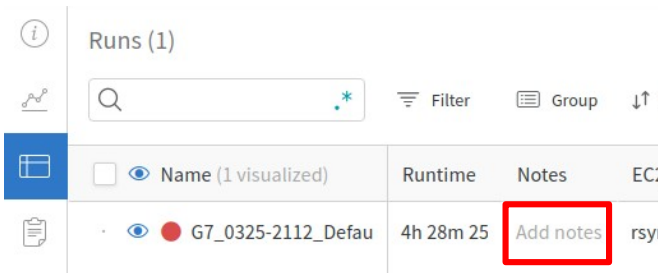
Codalab Grader



- Evaluate your model on the (private) test set
- Scores for project submission
- Public leaderboard

Final Submission

- Deadline: 13-05-2022
- 25% of course grade
- Project Report
  - Describe solution and provide source code snippets
  - Answer questions based on experiments
  - Present experimental results clearly and concisely (tables, example predictions, diagrams)
  - Describe observations and discuss possible reasons
- W&B Score Sheet
  - Annotate and export data (show all columns)
  - Do not delete AWS S3 data of runs used in report
- Grader Score Sheet (exercise2/doc/dlad22_teamXX_ex2_scores.csv)
  - Enter CodaLab grader scores of best run for each task
- Detailed instructions & grading criteria in handout



Hints

- **Start training early!**
  - A training run can take up to 24 hours (or even more for open challenge)
  - Your training might be interrupted when AWS is crowded (e.g. before a conference deadline)
  - You might discover a bug that requires retraining previous experiments
- You are allowed to use two p2.xlarge spot instances
- Read the handout carefully, consider the evaluation criteria, and answer all questions

Resources

- Exercise Handout (with detailed instructions and grading information): https://gitlab.ethz.ch/dlad21/exercise2/-/blob/master/doc/handout.pdf
- AWS Tutorial: https://gitlab.ethz.ch/dlad21/aws-tools
- Solution Template (with technical instructions in README.md): https://gitlab.ethz.ch/dlad21/exercise2
- CodaLab Grader: http://dlad-codalab.com/competitions/3?secret_key=8ad5f220-538c-4f65-a629-72778d79c912