Lab CudaVision
Learning Vision Systems on Graphics Cards (MA-INF 4308)

CudaLab Project

03.02.2023

PROF. SVEN BEHNKE, ANGEL VILLAR-CORRALES

Contact: villar@ais.uni-bonn.de

Video Semantic Segmentation

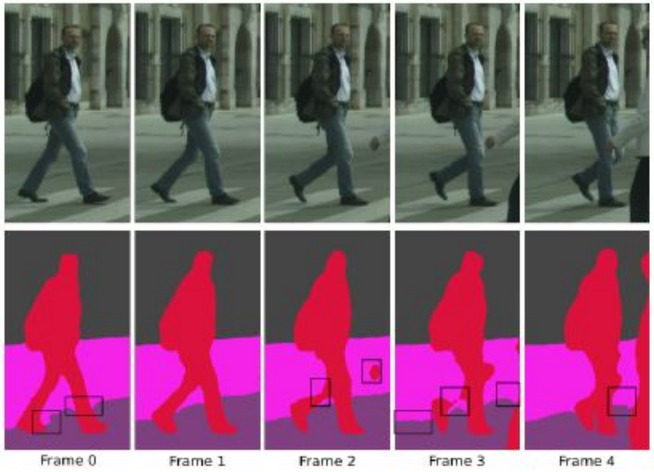
Video Semantic Segmentation

- **Semantic Segmentation:** Predicting a semantic category for every pixel in an image
- **Video Semantic Segmentation:** Predicting a semantic category for every pixel in every frame of a video sequence:
    1. Apply model frame-by-frame
    2. Exploit temporal dependencies
- **Applications:**
    - Autonomous driving
    - Robotics
    - Agriculture
    - …

Challenges

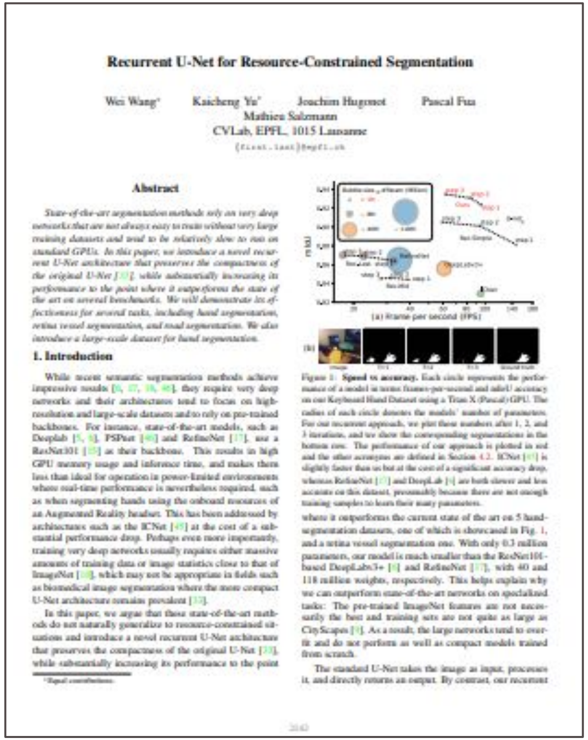
- Processing frame by frame leads to errors (flickering, ghosting, …)
- Difficult handling of occlusions

➢ Temporal consistency can correct most issues



Proposed Approach

Inspiration



Recurrent U-Net for Resource-Constrained Segmentation

Wei Wang* Kaicheng Yu* Joachim Hugonot Pascal Fua Mathieu Salzmann
CVLab, EPFL, 1015 Lausanne

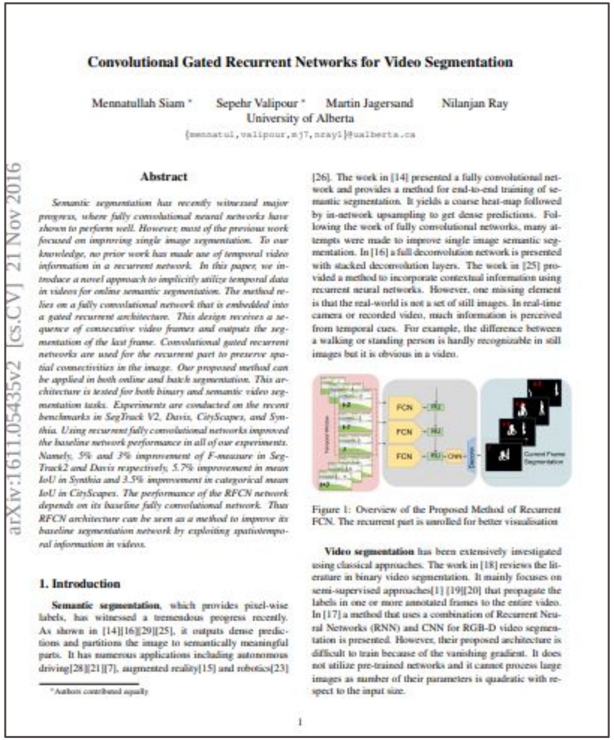
Abstract

1. Introduction

Semantic Segmentation of Video Sequences with Convolutional LSTMs

Andreas Pfeuffer[1], Karina Schulz[1], and Klaus Dietmayer[1]

I. INTRODUCTION



Convolutional Gated Recurrent Networks for Video Segmentation

Mennatullah Siam * Sepehr Valipour * Martin Jagersand Nilanjan Ray
University of Alberta

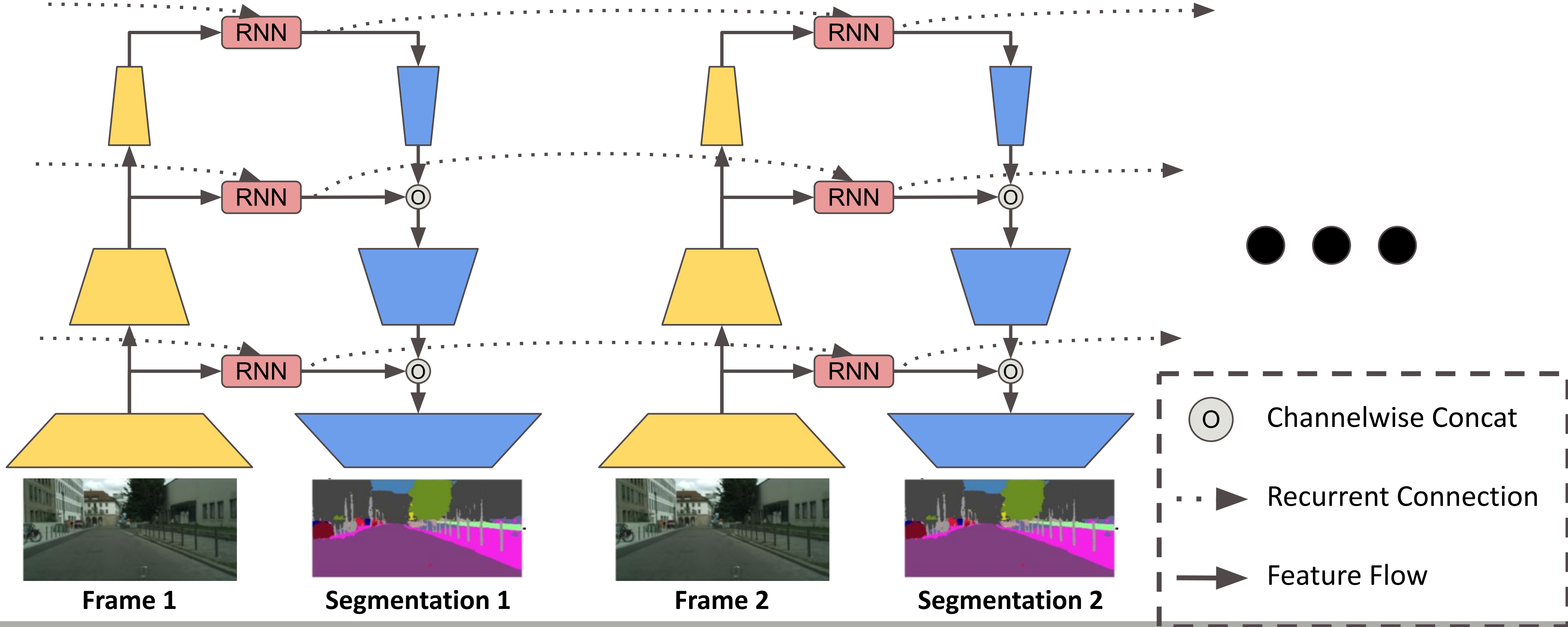
arXiv:1611.05435v2 [cs.CV] 21 Nov 2016

Abstract

1. Introduction

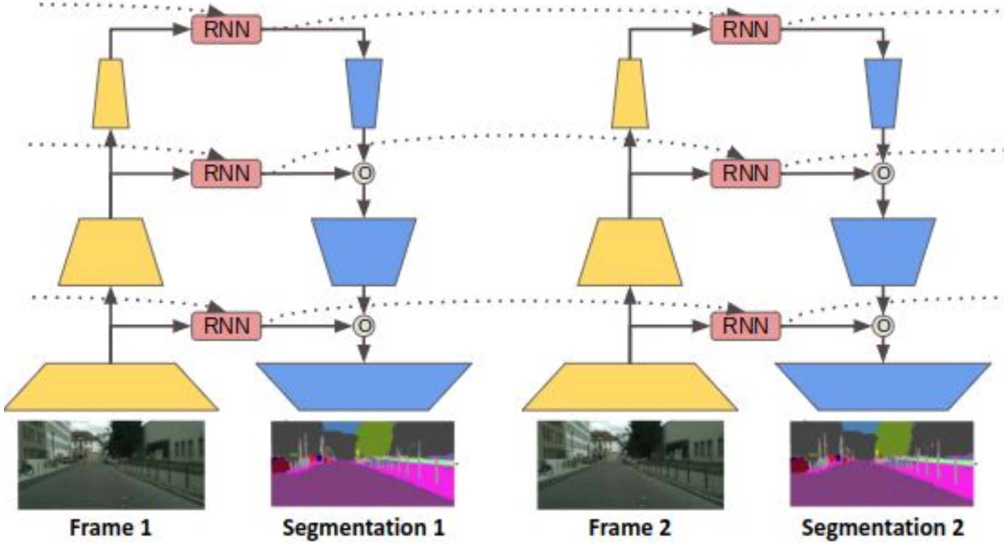
Figure 1: Overview of the Proposed Method of Recurrent FCN. The recurrent part is unrolled for better visualisation

Proposed Model



Proposed Model

- Recurrent UNet-like model
- Convolutional encoder/decoder
  - Convolutional blocks
  - Three residual paths
  - Recurrent modules



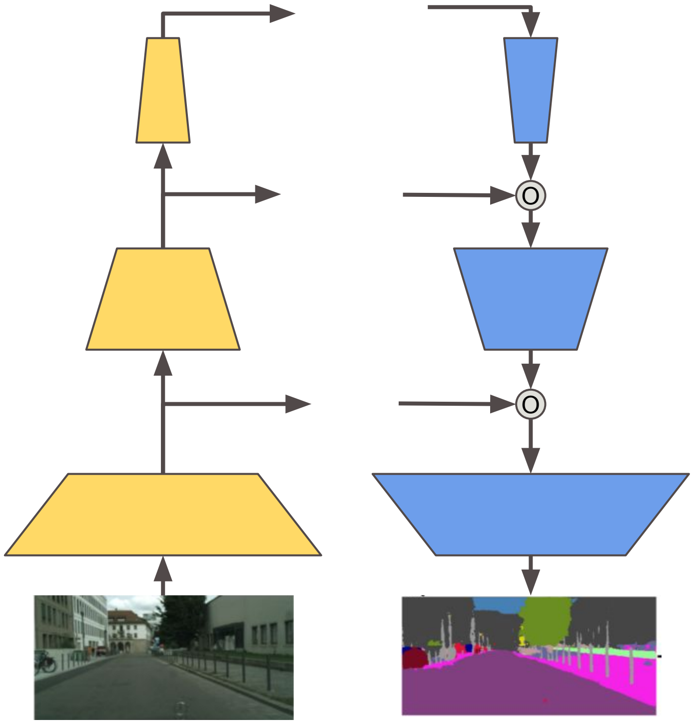
Encoder and Decoder

Encoder

- Extracts features of different level and spatial resolution from the input frames
- Convolutional module
  - ResNet-like
  - VGG-like

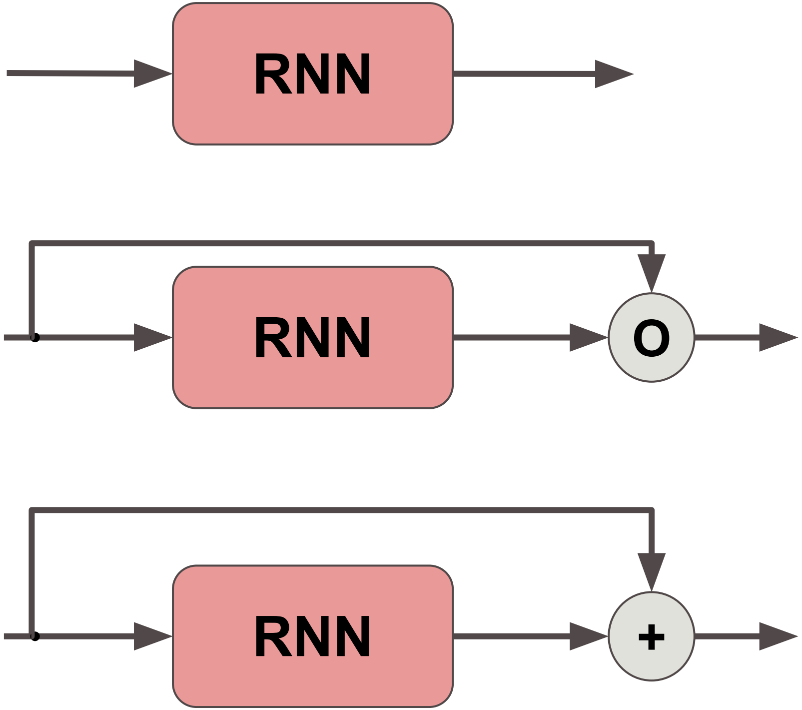
Decoder

- Decodes feature maps into segmentation masks
- Combines features of different level via concatenation in a U-Net manner
- Usually the mirrored version of encoder



Recurrent Lateral Connections

- Provide features of different level and spatial resolution to the decoder
- Model temporal dependencies via recurrent cells
- Different possible cells:
  - ConvLSTM
  - ConvGRU
- Design choices:
  - Standard (top)
  - Dense-connection (middle)
  - Residual-connection (bottom)

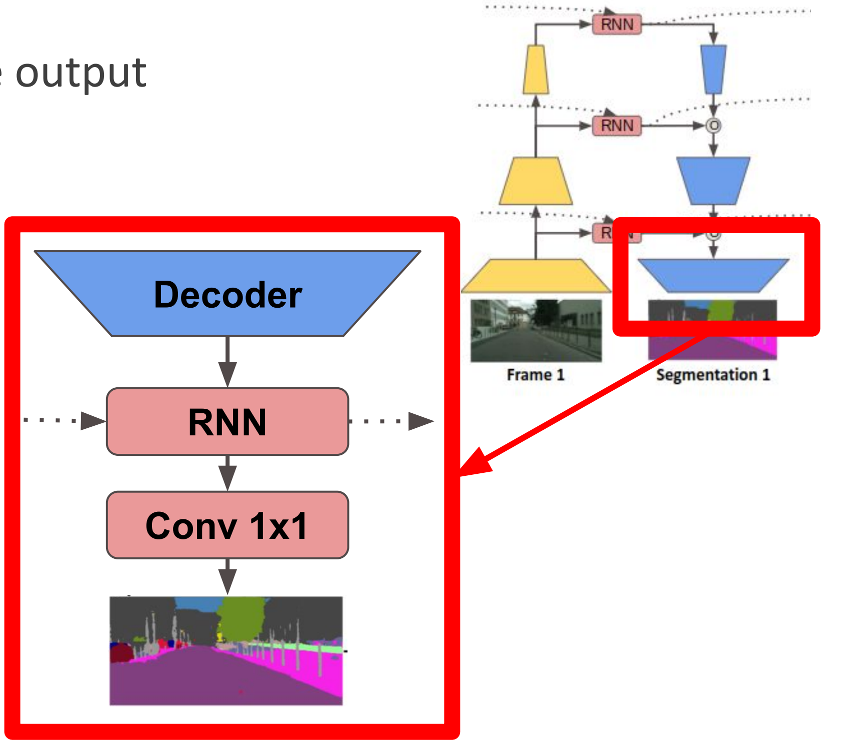


Final RNN?

- Model temporal dependencies directly at the output
  - Can be seen as filtering

- Two needed components:
- **Recurrent cell**: temporal modelling
- **Conv. Layer**: aligns the number of channels and produces desired output

- This recurrent module is sometimes used, however it is not mandatory

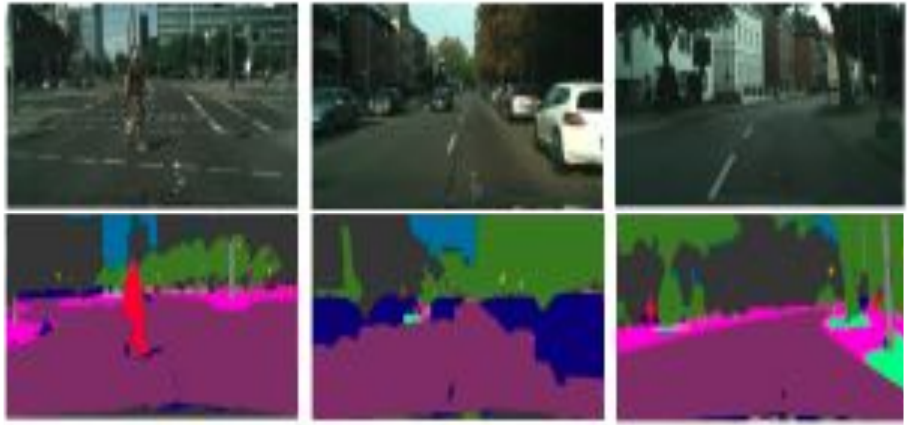


Datasets

Cityscapes Dataset

- Dataset for autonomous driving related tasks
  - Object detection
  - Semantic segmentation
  - Depth estimation

- 5000 images of size (1024x2048)
  - 2975 training imgs
  - 500 validation imgs
  - 1525 test imgs (no ground truth)
- 30 classes, but we will only use 19
- Sequences of 30 frames, but only 20th is labelled



Available in `home/nfs/inf6/data/datasets/cityscapes/leftImg8bit_sequence/`

Dataset is Small

Data Augmentation

- Standard to augment Cityscapes
  - Taking image crops
  - Resizing
  - Mirroring
  - …
- Temporal augmentations
  - Skip frames
  - Interpolation
  - …
- Try to get the most out of the few images that you have

Pretraining on MS-COCO

- Pretrain the segmentation-only model on a larger dataset, e.g., MS-COCO
- MS-COCO
  - Over 45K annotated images
  - 91 semantic categories
- This is a large dataset, and training here will take a long time (perhaps days)

Available in `/home/nfs/inf6/data/datasets/coco`

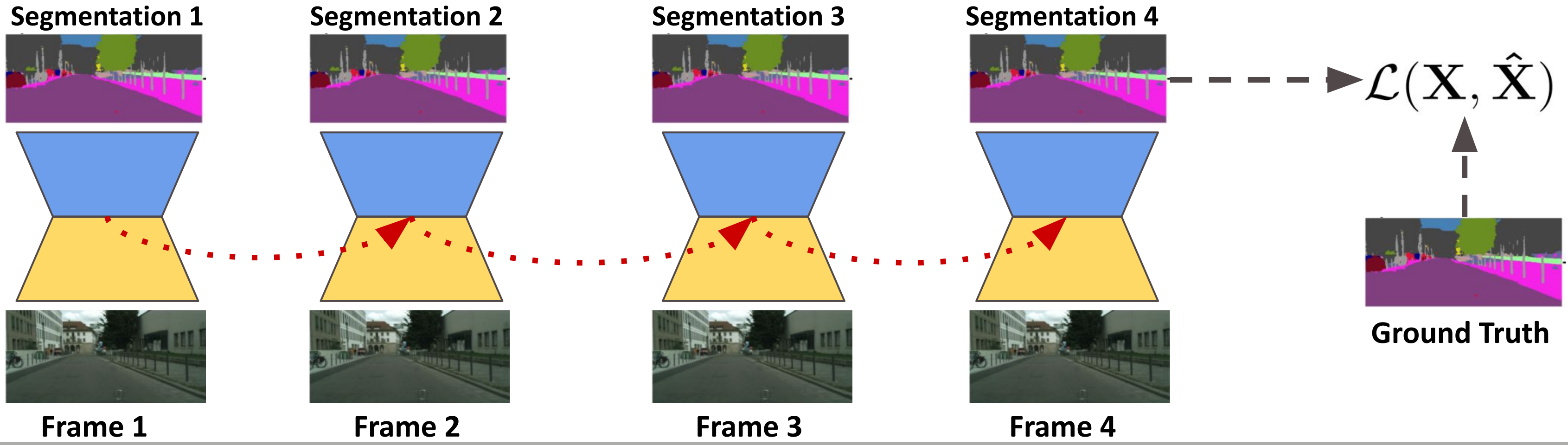
Training & Evaluation

Train/Eval on Cityscapes

- Training:
  - Image crops of size: (3, 512, 1024)
  - Sequences of 5 frames
- Evaluation:
  - Original image size: ≅(3, 1024, 2048)
  - Sequences of 5 frames and 12 frames
- *CrossEntropy* Loss function for training
- mAcc and mIoU quantitative evaluation metrics
- Make GIFs for qualitative evaluation

Training Protocol

- Only the 20th frame in every sequence of 30 is labelled
- We cannot enforce full supervision
- ➢ Compute the loss only on the annotated frame
- ➢ Temporal Regularization?



Project Goals and Deliverables

Passing Requirements

1. Implement the required model, pipelines and utils

2. Train your models to achieve best possible results on Cityscapes
   - You must implement and train the described model
   - ➢ Make changes and train further model variants to achieve better results

3. Beat the naive framewise baseline
   - **Baseline:** applying the image segmentation model frame-by-frame

4. Create overview notebook

5. Write project report

Deliverables

- Complete codebase
  - Clean and structured
  - Not just a notebook!
- Trained model checkpoint and (tensorboard, WandB, …) logs
- Overview notebook (.ipynb & .html) showing main functionalities:
  - Load data and display some samples
  - Load pretrained model and display the structure or some stats
  - Display some qualitative results (e.g. results on 5 sequences)
  - Show the quantitative evaluation
- Project report

Grading

- Results and Experiments **55%-60%:**
  - Performing several experiments and obtaining good results
  - **Additional experiments**: ablation study, changes in the model, …
  - This grade partly depends on how your results compare to the class
- Codebase & Overview Notebook **20-25%:**
  - Implement all functionalities
  - Modularity and structure
- Report **20%-25%**

Project Report

- Document your work in the project report
- Try to be brief, but readable and informative
- Include figures and tables
- Use *BibTex* for the references
- I expect 8-12 pages, but highly depends on number and size of imgs/tables
- Use the following template
  - https://www.overleaf.com/read/tmnvhrsdmjrp

Additional Experiment Ideas

- Try your own ideas!

- Training and pretraining:
  - Supervised or self-supervised pretraining
  - Use a encoder pretrained on ImageNet and perform transfer learning

- Tweak the model
  - Use a nice backbone (e.g. ResNet or ConvNext)
  - Adapt a different architecture (e.g. DeepLab v3+)
  - Investigate the type and positioning of the recurrent modules

- Investigate different training strategies:
  - Use different loss functions
  - Regularization to enforce temporal consistency
  - Use adversarial supervision
  - Advanced data augmentation (e.g. mix-up) and regularization (e.g. label smoothing)

Important Dates

- **03.02**: Starting date
- **10.03-21.03**: Revision session (flexible dates)
- **20.03**: Draft submission due
- **31.03**: Final submission:

Questions?

References

1. Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2016.
2. Ronneberger, Olaf, et al. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention (MICCAI). 2015.
3. Siam, Mennatullah, et al. "Convolutional gated recurrent networks for video segmentation." IEEE International Conference on Image Processing (ICIP). 2017.
4. Wang, Wei, et al. "Recurrent U-Net for resource-constrained segmentation." IEEE/CVF International Conference on Computer Vision (ICCV). 2019.
5. Pfeuffer, Andreas, Karina Schulz, and Klaus Dietmayer. "Semantic segmentation of video sequences with convolutional lstms." IEEE Intelligent Vehicles Symposium. 2019.
6. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European Conference on Computer Vision. (ECCV), 2014.