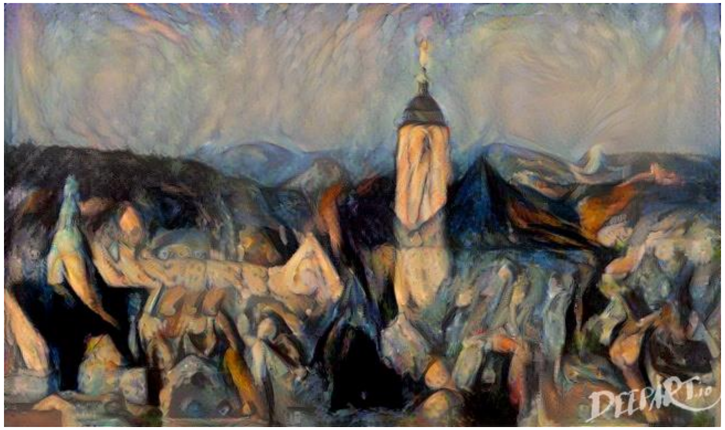
Start of the Miniproject Phase

Joint lecture by everyone! :-)



Face Recognition of Morphed Faces

- Use VGGFace or other neural network
- Explore response on morphed faces
  - We can generate Morphs easily using our 3D Morphable Model
- Does the activation of the output layer change continuously or in steps, are there intermediate activation patterns?
- What activation do we get from the average face?

- Simulate and analyze *Morphing Attack*

(Image: HU Berlin)

Project 1 [Gesture recognition from ToF data]:
Unlike conventional digital photographic cameras, which only capture the intensity of light within some spectral window (e.g., red, green, and blue windows), Time-of-Flight (ToF) cameras are able to capture the depth or distance between the camera and the different scene points. In many recognition tasks, most information relates to three-dimensional structure, which is projected onto a two-dimensional manifold in the shape of images.
In this project we propose using neural networks to perform gesture recognition from data acquired with a ToF camera.
The novelty of this project is two-fold: on the one hand the use small-size ToF data instead of RGB images as input, and on the other hand the use of "Invariant-Scattering Convolution Networks" (ISCN) as a mean for dimensionality reduction. A classical convolutional architecture is to be compared to its ISCN counterpart in terms of gesture recognition performance. Both the creation of Scattering filter banks and the corresponding feature extraction can be addressed with a single line of code in Matlab.

Project 2 [Correlated source localization from ToF data]:
In real-life scenes one can often find opportunity illuminators emitting periodically-modulated light, e.g., fluorescent tubes, discharge lamps, LEDs featuring PWM dimming, and alike. This light sources could eventually be used as opportunity illuminators in a Time-of-Flight (ToF) camera. Unfortunately, the exact 3D location of these sources is unknown a priori, what precludes their use as opportunity illuminators.
In this project we propose exploiting the fact that real scenes possess some underlying structure (e.g., planarity in walls, floors, tables, etc.) to retrieve the exact location of a single source of periodically-modulated light. A neural network is to be used to estimate the 3D coordinates of the source location from the ToF image data. The network will be trained using synthetic ToF data arising from simple piecewise planar scenes.

Autonomous Forward Parking Using CNN

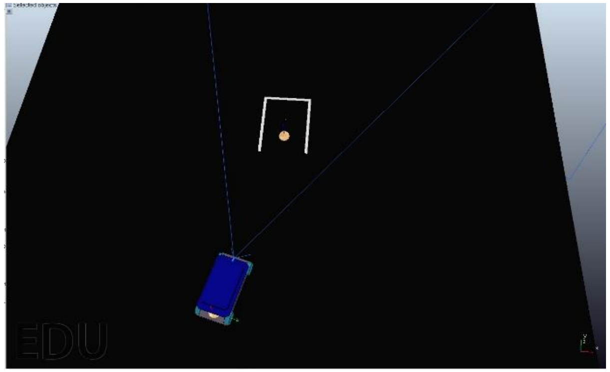
In a given simulation environment, it is required to use convolution neural network to park a 3D car model in a free parking space.

Objective:

1. Dataset collection using the given simulation enviroment and programs.
2. Design a convolution neural network to park the 3D car model autonomously in the specified parking space.
3. Train and test the neural network with the help of collected dataset.
4. Deploy the trained neural network model to park the car autonomously in the V-rep simulator .

Tools:

- V-Rep robotics simulator



YOLO-v3 Object Tracking in 2D for Search & Rescue Application

Objective:

1. COCO Dataset preparation
2. Fine tuning of YOLO-v3 network from multiple classes to detecting people as a single class.
3. Test the implemented network on a live video feed; an implementation that can be applied in robot search and rescue missions.
4. Calculate the center point of the bounding boxes of the detected people.

Tools:

- COCO Dataset

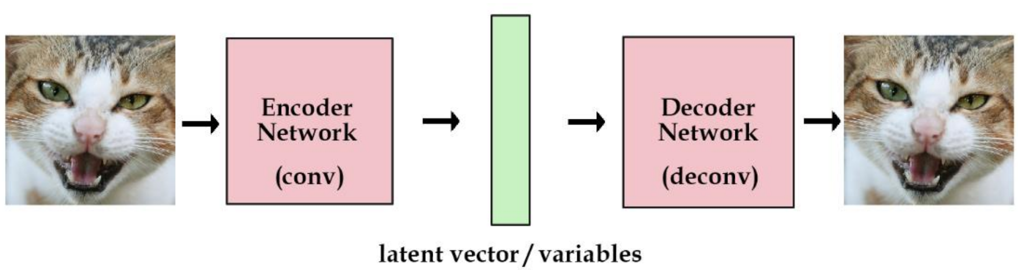
Project 1: Autoencoders with prior information

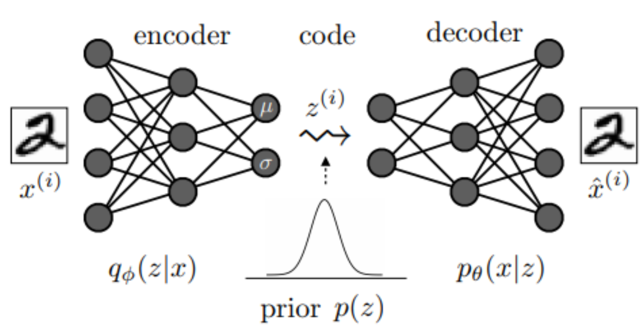
1. Read, understand and summarize some recent developments provided in
   http://bayesiandeeplearning.org/2018/papers/151.pdf

2. Implement a convolutional autoencoder on a simple image data set, e.g. MNIST

3. Penalize the entries in the semantic code to vector to have a certain meaning, e.g. a) the class, b) the centroid, c) the mean value, d) the variance, e) 4 further entries without meaning, and train the autoencoder

4. During Inference modify the semantic codes. Show the group what happens!



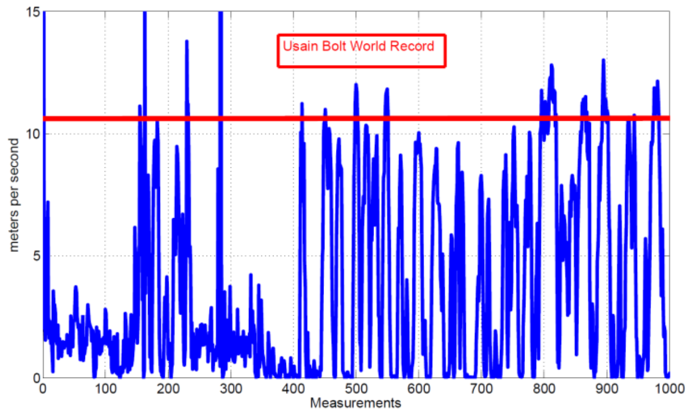


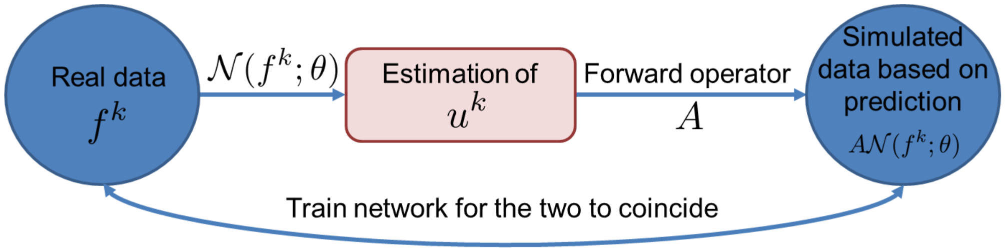
Project 2: Train a model-based autoencoder for computing derivatives

1. Read and understand a paper that I will give you

2. Implement two networks for computing the derivatives of noisy data, using
   a) A supervised training approach
   b) A model-based autoencoder

3. Let both models compete during inference. Which approach performs better? What are the details of your training procedure? What happens for different architectures?

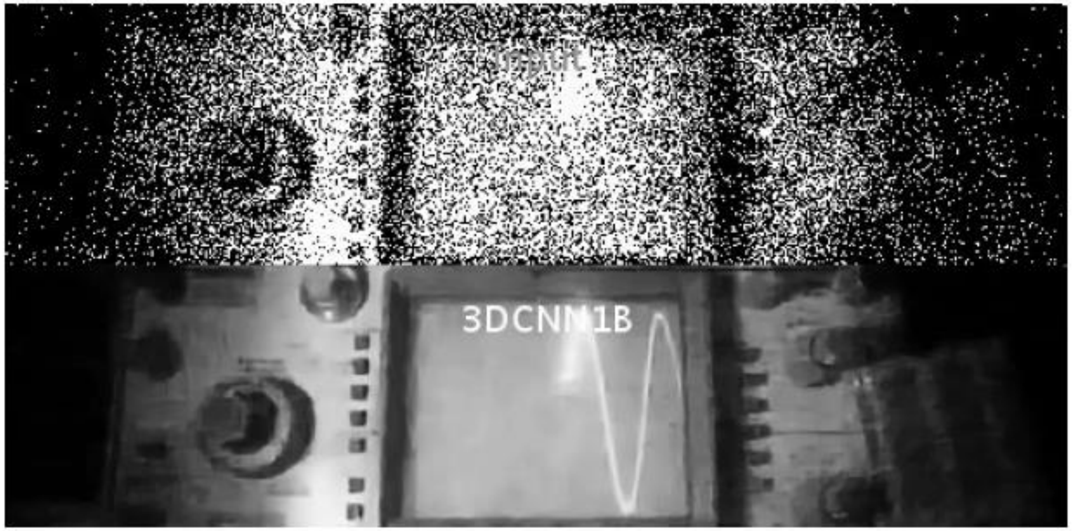




e.g. by training via $$\min_\theta \sum_{\text{data examples } k} \|A\mathcal{N}(f^k;\theta) - f^k\|^2$$

Project 1:

High Speed Imaging



Implement the video recovery algorithm described in the paper A Bit Too Much? High Speed Imaging from Sparse Photon Counts [https://arxiv.org/abs/1811.02396] in PyTorch. Experiment with Perceptual loss function

Project 2:

Image Reconstruction using a Deep Generative Prior

- Choose an existing generative model for a particular class of images (say DCGAN for CelebFace data https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html)
- Formulate an observation model for image reconstruction tasks such as super-resolution, deblurring
- Reconstruct images through the generative model following the algorith described in https://arxiv.org/abs/1703.03208
- Try more sophisticated generative models and challenging reconstruction problems
- Compare results with methods that do not consider that images are from a generative model. For the case of super-resolution, you could for instance compare with LAPSRN https://github.com/BUPTLdy/Pytorch-LapSRN

P1: Basketball Activity Recognition

Implement a Convolutional Neural Network (CNN) for the activity recognition in a basketball activity dataset.

Optional: Implement Transfer Learning

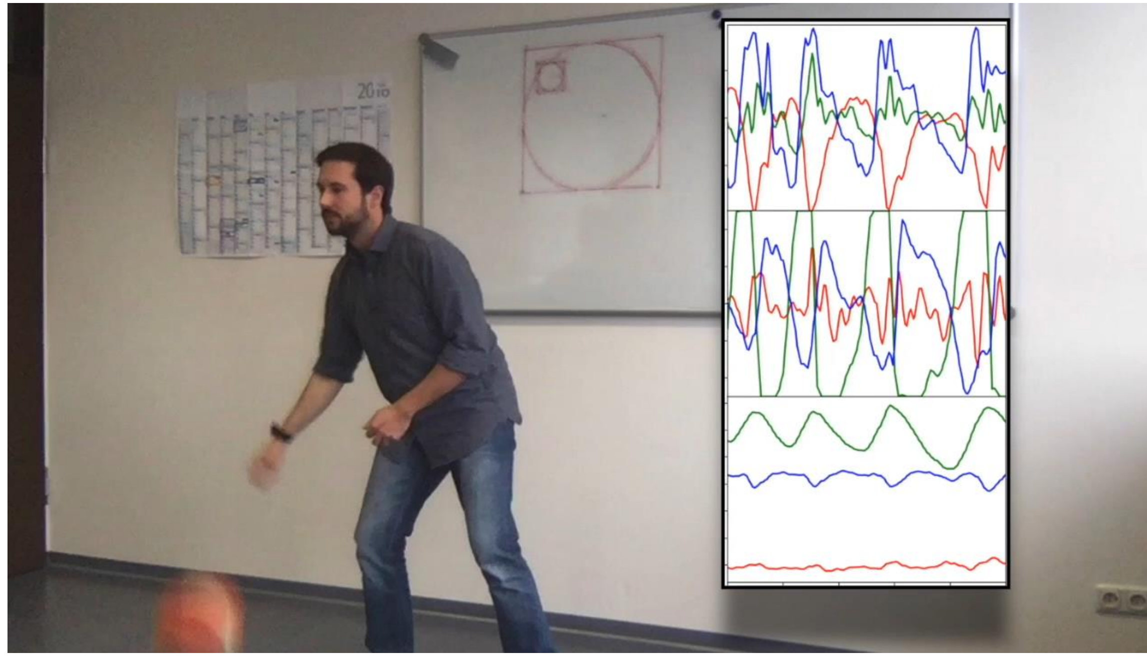
- For example transferring a trained model of known activities to recognize a yet unknown activity.
- Can we use the knowledge of how fast dribbling looks like during charge to recognize a slower dribbling frequency?

Description:

In this project you should try out activity recognition with CNNs based on our basketball activity dataset.

Optionally, Transfer Learning should be applied by recognizing a previously unknown activity, which is very similar to an already known.

P1: Basketball Activity Recognition



- 4 classes:
  - low dribbling
  - high dribbling
  - crossover
  - void-class
- 5 participants
- Labeled data
- 300 Hz sampling rate
- IMU with 3 3D-sensors
- 9 input channels
- 30 minutes of dribbling activity recorded in the lab

P2: Heartbeat Detection

Implement a Convolutional Neural Network (CNN) for the heartbeat detection in optical blood flow PPG measurements.

Description

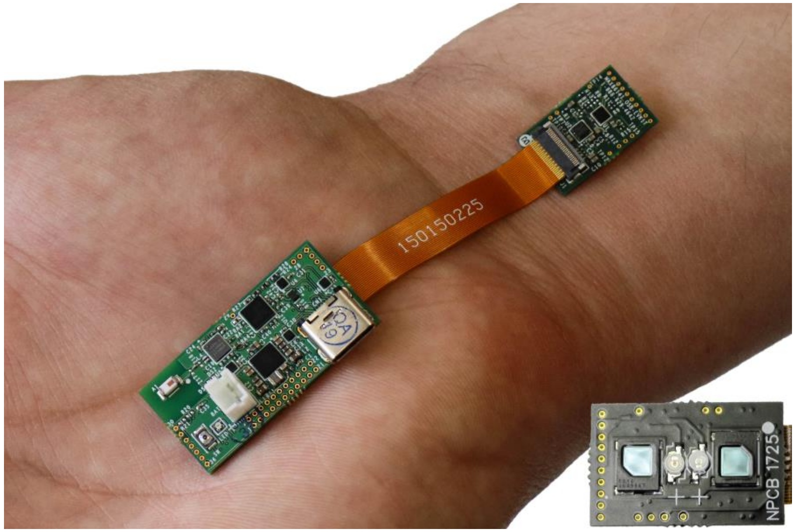
Photoplethysmography (PPG) is an optical measurement principle that is applied in modern wearable devices like smartwatches to obtain the vital signs, such as heart rate.

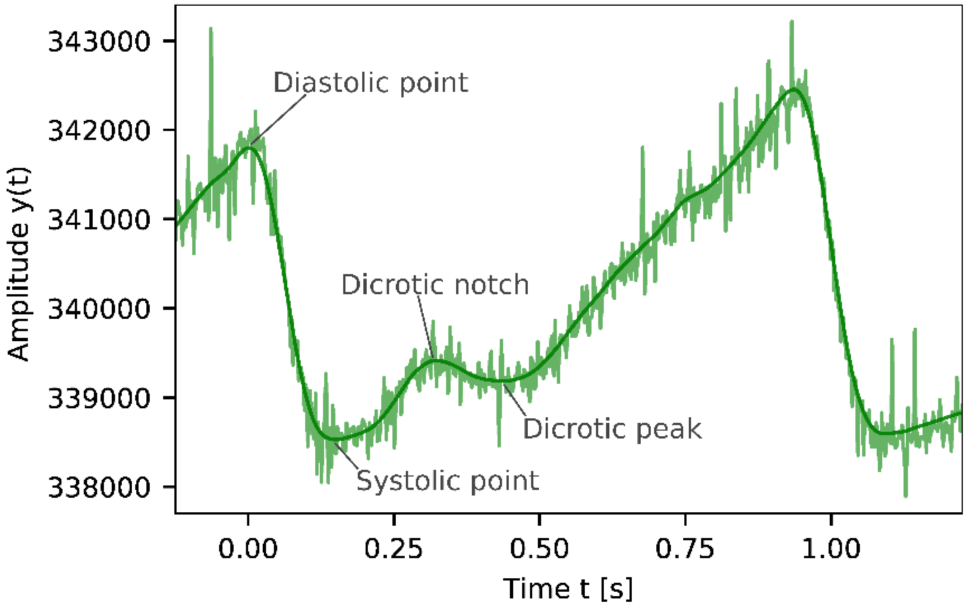
In this project, you are going to implement the detection of the particular pulses within the time series of data from an optical sensor by training a CNN.

P2: Heartbeat Detection

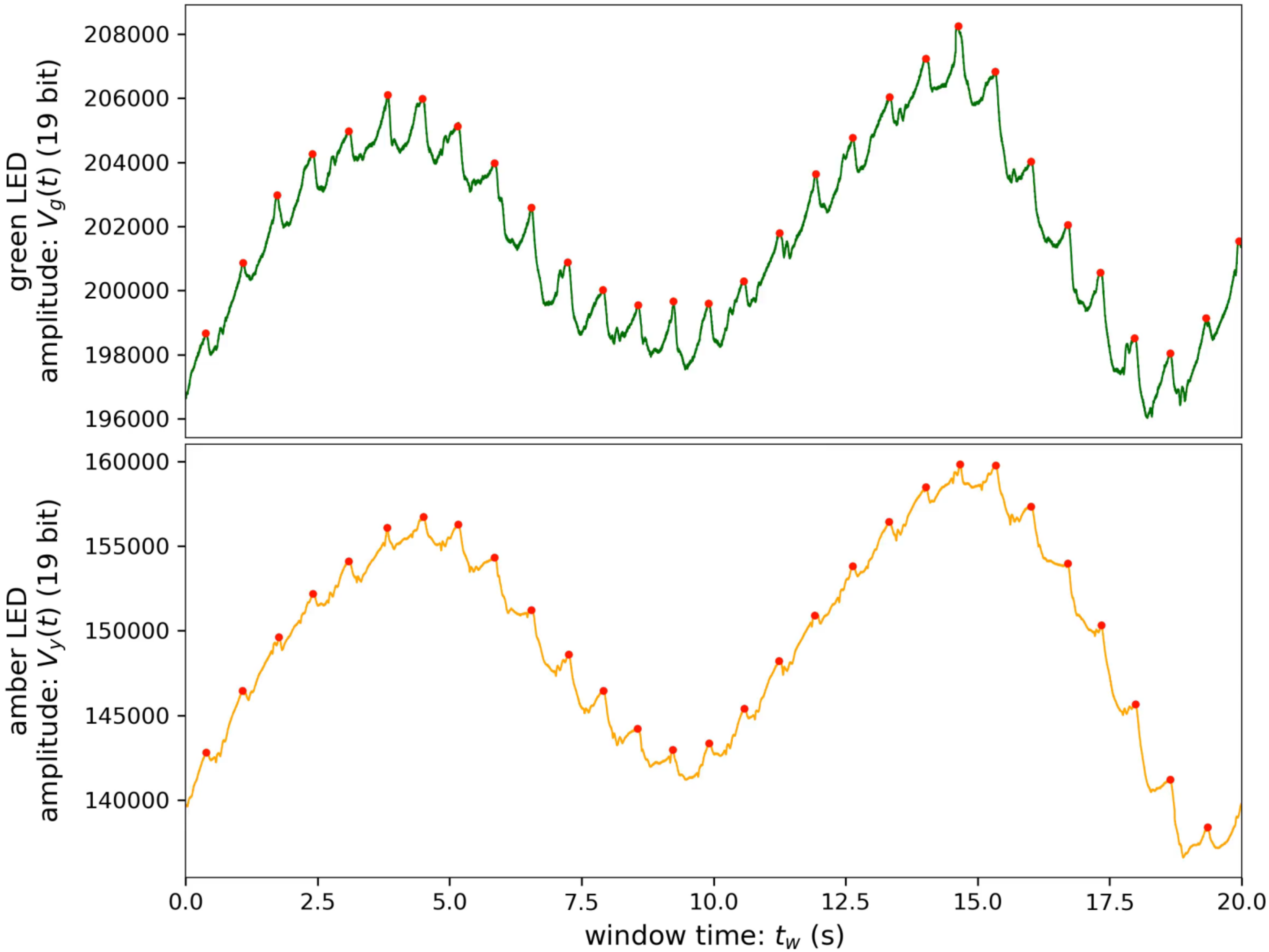
Photoplethysmography Dataset

- 5 subjects
- 6 experiments (3 min. each)
- 3 measurement locations
  - Traditional wrist
  - Forehead
  - Earlobe
- 2 channels (green, yellow light)
- 512 S/s (Hz)
- 240 min. of labeled data





P2: Heartbeat Detection



Project Assignment

https://simpleassign.com/poll/-LfKiGsY70UXYSeQSaWG

Currently, 11 students are signed up for the course!

The assignments will be send out via unisono – students who are not registered will not get a project!!