# IL Project Proposal: Imitation Learning for Dense Traffic Driving Scenarios

UNI FREIBURG

AIS Autonomous Intelligent Systems

# The Task

- Agent learns to drive in racetrack environment via **Behavior Cloning**
  - Multiple other scripted traffic participants
  - Complex 3D environment
- Use either raw pixels or state as input for supervised learning of actions

# Project Milestones

1. Set up environment and train IL agent in no-traffic environment
2. Implement a **LSTM layer** to handle non-markovian behavior
3. Train IL agent to drive in dense traffic
4. Compare **discrete action** learning (classification) with **continuous action** learning (regression) instead of classification

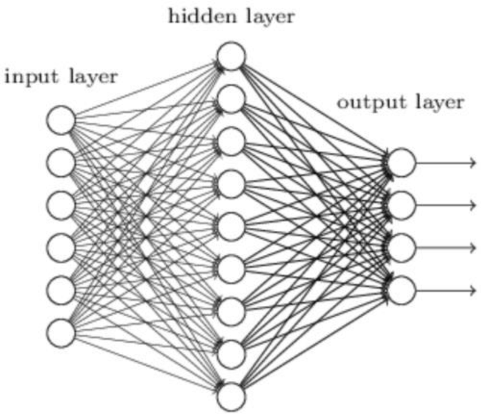
# References

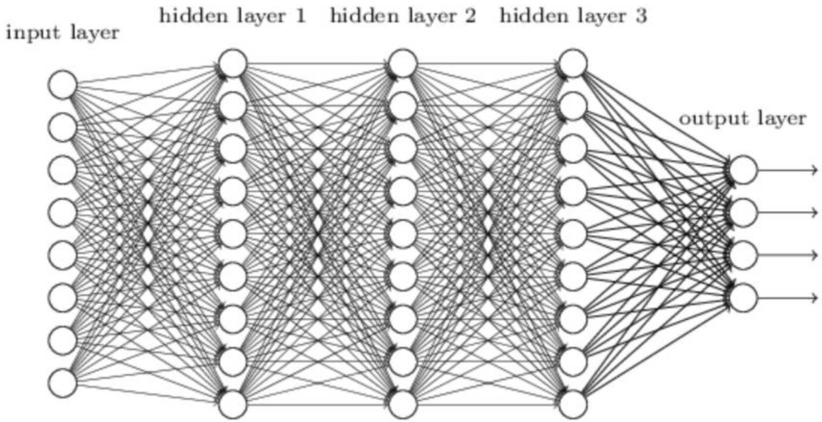

- https://github.com/madras-simulator/MADRaS
- L. Cardamone et al.: Learning Drivers for TORCS through Imitation Using Supervised Methods

# Project Proposal: Architecture Search in Reinforcement Learning

# The Task

- Investigate the influence of the network architecture on the agents performance



# Project Milestones

- Integrate the OpenAI Gym environments to your DQN implementation (RL exercise)
- Train the same agent with different networks architectures
  - Varying network depth
  - Added skip connections
  - Bonus: Use automated search
  - …
- Compare convergence speed, final performance, robustness, …

# Project Characteristics

- Open research area
  - Results are of great interest
- Interdisciplinary
  - Experience in training neural networks beneficial
  - Basics in RL and Auto-ML needed
- Compute-intensive
  - Several (10+) networks have to be trained
- Extendible to other hyperparameters
  - learning rate, …

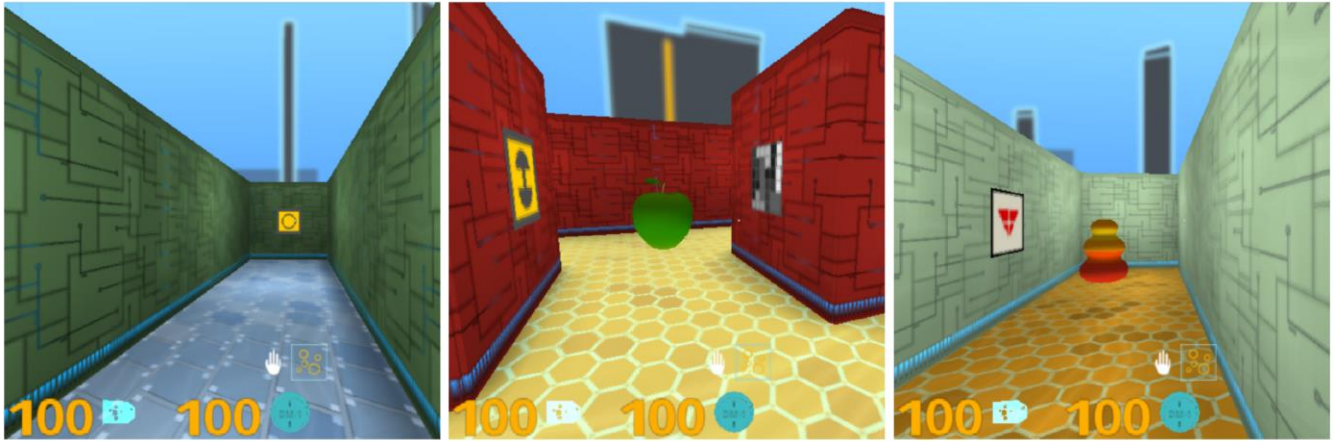
# Project Proposal:
# Exploration with Intrinsic Motivation

UNI FREIBURG

AIS Autonomous Intelligent Systems

# The Task:

- RL Agent has to navigate though a maze using first-person images only
  - Challenging exploration due to sparse rewards




[Sample images: DmLab]

# Project Milestones

Road Map:

- Integrate the DeepMind Lab environment to your DQN implementation (RL exercise)
- Reimplement an intrinsic bonus reward for efficient exploration
  - ICM [Pathak et al.]
  - RND [Burda et al.]
  - …
- Come up with your own intrinsic reward module

# Sources

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018a.

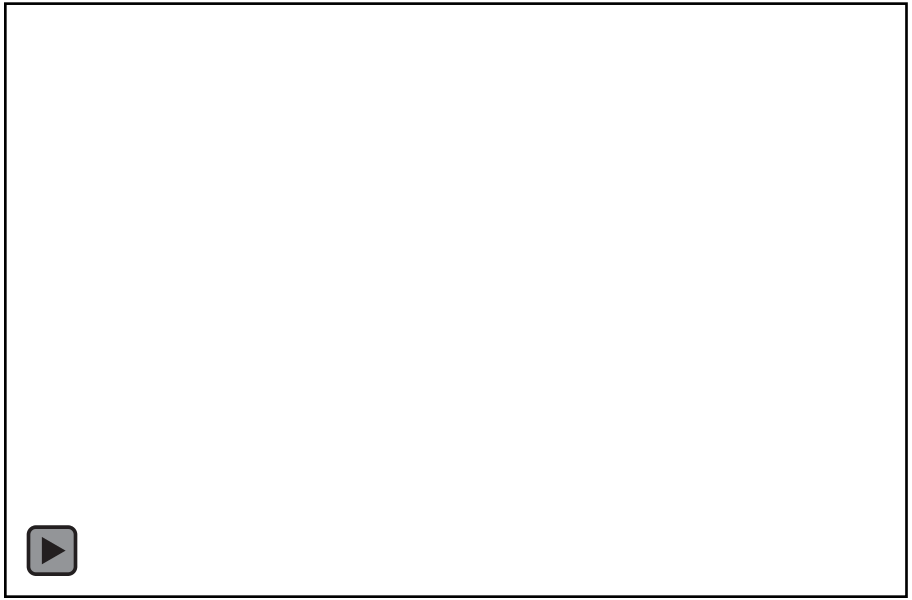
Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, 2017.

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

# Project Proposal:
# Multimodal Semantic Segmentation for Autonomous Driving

# The Task:

Design and efficient encoder-decoder architecture for semantic segmentation using RGB-D images



# Project Milestones

- Baseline: Current state-of-the-art SSMA with AdapNet++
- Input: RGB-D images of 768x384 pixels
- Roadmap:
  - Encoder: MobileNet-v2 and modified Xception
  - Mutiscale module: Dense prediction cell (DPC)
  - Decoder: Data-Dependent Decoder
  - Develop multimodal fusion module
- Benchmark: Cityscapes and KITTI

# References


- Abhinav Valada, Rohit Mohan, and Wolfram Burgard, "Self-Supervised Model Adaptation for Multimodal Semantic Segmentation“, *arXiv preprint arXiv:1808.03833*, 2018.

- Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan, “Decoders Matter for Semantic Segmentation: Data-Dependent Decoding Enables Flexible Feature Aggregation”, *arXiv preprint arXiv:1903.02120*, 2019.

- Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. "Searching for efficient multi-scale architectures for dense image prediction." In *Advances in Neural Information Processing Systems*, pp. 8699-8710. 2018.

# DL Lab CV Projects

Silvio Galesso and Christian Zimmermann

# Project 1: Multi HPE

- Task: Build a top-down Multi Human Person Estimation system and evaluate on complete Coco.
- Milestones:
  - MS1: Train and Eval Person Detection network ("Bounding Box" network)
  - MS2: Combine detection network with Single HPE network
  - MS3: Evaluate on Coco and compare to SOTA
- Extensions:
  - Retrain pose estimation on actual detections
  - Influence of Multi Task Learning (Segmentation + Keypoints)

# Project 1: Multi HPE

- Task: Build a top-down Multi Human Person Estimation system and evaluate on complete Coco.
- Material:
  - https://github.com/qfgaohao/pytorch-ssd
  - https://modelzoo.co/model/detectron-models-for-object-detection
  - http://cocodataset.org/

# Project 2: Active learning

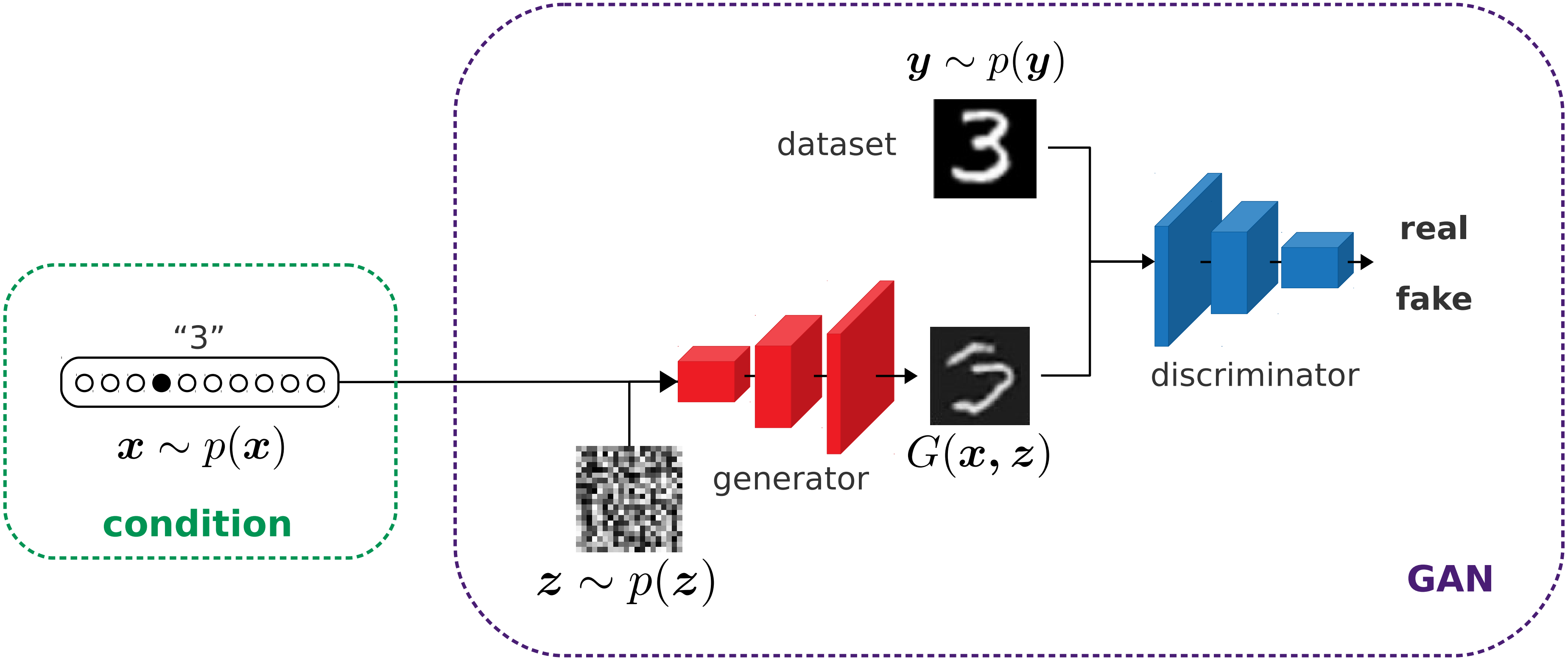
- Active learning: In order to generalize to an unlabeled set of data the learner automatically selects a subset of data a human annotator labels.
- Procedure:
  - Split training set into labeled and ‘unlabeled’.
  - Train a network on the labeled set only.
  - Use the network and a selection strategy to pick frames from the unlabeled set.
  - Add selected frames to the training set and report results on the remaining unlabeled set and the Coco evaluation set.
- Task:
  - Evaluate different strategies for active learning on the Coco subset used in the exercise.

# Project 2: Active learning

- Task: Evaluate different strategies for active learning (Random, highest probability, best vs. second best, MPE, … )
- Milestones:
  - MS 1: Train and evaluate the baseline network for different splits (use only 20%, 40%, … of the train set).
  - MS 2: Implement different selection criteria of the remaining training set.
  - MS 3: Retrain on the updated training sets and evaluate.
- Material:
  - http://calvin.inf.ed.ac.uk/wp-content/uploads/Publications/liu17iccv.pdf

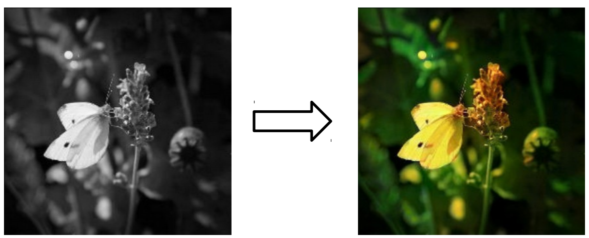
# Project 3: Image Generation with Conditional GANs

Generative Adversarial Networks can be conditioned on some feature of the training data, e.g. class labels:

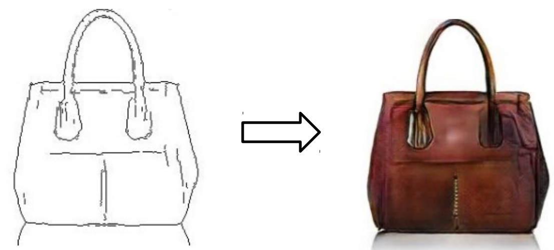


# Project 3: Image Generation with Conditional GANs

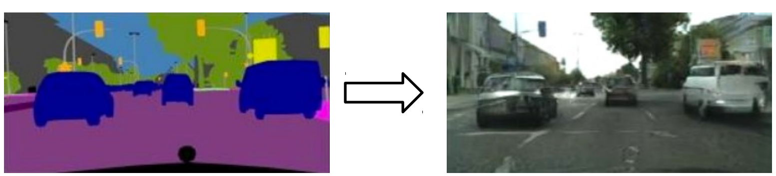
Conditioning can be done on different domains, e.g.:



Greyscale to RGB

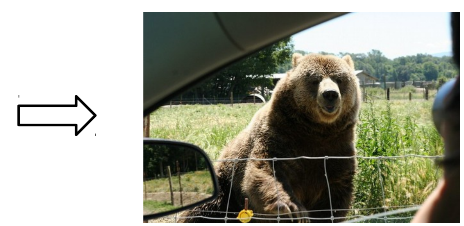


Edges to Photo



Segmentation to RGB

“a big brown bear standing inside a fenced in area”



Caption to Image

Source: Isola et al., “*Image-to-Image Translation with Conditional Adversarial Networks*”

# Project 3: Image Generation with Conditional GANs

Your task: experiment with conditional image generation on different datasets.

- **Milestone 1**: read the paper (arxiv.org/abs/1611.07004) and train the model (https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix) on one of the paper's simple settings.
- **Milestone 2**: train your own implementation on a simple different task, e.g.
  - Flowers (attributes, segmentation)
    http://www.robots.ox.ac.uk/~vgg/data/flowers/
  - Birds (attributes, part locations, bounding boxes)
    http://www.vision.caltech.edu/visipedia/CUB-200-2011.html
  - Faces (attributes, keypoints/landmarks)
    http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
    http://www.ifp.illinois.edu/~vuongle2/helen/
    https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/aflw/
- **Milestone 3**: train your own implementation on a hard task, e.g. COCO (human pose, segmentation, caption, class...)

# Bayesian Neural Networks

Deep Learning Lab Course

Univeristy of Freiburg

May 29, 2019

# Problem Statement

- Point-Estimate Neural Networks
  1. fit well the training data
  2. poor performance on unseen data
  3. incapable of assessing uncertainty in the training data
  4. overconfident decisions

- Bayesian Neural Network
  1. measure of uncertainty in the predictions

# Problem Statement

Given training inputs $X = \{x_1, \ldots, x_N\}$ and outputs $Y = \{y_1, \ldots, y_N\}$, we want to infer the parameters $\omega$ of $y = f^{\omega}(x)$ that are **likely to have generated** our outputs with a Bayesian approach.

We would put a **prior** distribution $p(\omega)$ and define a **likelihood** $p(y|x, \omega)$ (softmax likelihood for classification and Gaussian likelihood for regression).

Given prior and likelihood, we look for the **posterior distribution** $p(\omega|X, Y)$.

The posterior is used to predict for a new input $x^*$

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega)p(\omega|X, Y)d\omega$$

# Problem Statement

The **posterior** is **not tractable** in general.

There are different approximation techniques:

- classical methods based on Markov chain Monte Carlo sampling:
  it allows to draw samples from a distribution even if we do not compute it.
- variational inference

$$\min KL\left(q(\omega)||p(\omega|X, Y)\right)$$

# Project Goals

- study the literature and becoming familiar with state-of-the-art Bayesian Neural Networks
- implement at least 3 different Bayesian Neural Networks techniques
- test their predictive performances on different regression and classification tasks.

# Lab Projects

**Topic 1: Neural Processes as a alternative for Gaussian Processes?**

Neural Processes (NP)[1] are introduced in 2018 as a computationally efficient alternative to Gaussian Processes (GP). The task is to implement both NP and GP and compare in Bayesian optimization setting.

**Topic 2: Transformer as an alternative for LSTMs?**

Transformer models[2] are introduced to eschewing recurrent neural networks in machine translation. But are they also an alternative for RNNs on other tasks? Implement LSTMs and Transformer and compare them in terms of performance and runtime on a time series task like phoneme recognition.


[1] Garnelo, Marta, et al. "Neural processes." arXiv preprint arXiv:1807.01622 (2018)
[2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

# Lab Projects

**Topic 3: Neural Feature Search**

Implement a automated neural feature search for tabular data inspired by recent advances from neural architecture search, such as DARTS [3]. Main focus is on an efficient and numeric stable implementation and optimization.

**Topic 4: Runtime Dropout as a alternative to Parameter Noise**

To increase exploration in RL you could use Parameter Noise [4]. But another kind of noise in the parameter space could be created by dropout during inference. Implement a RL algorithm with parameter noise and dropout and compare both in terms of exploration and overall performance on a toy task.

[3] Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055.
[4] Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., ... & Andrychowicz, M. (2017). Parameter space noise for exploration. arXiv preprint arXiv:1706.01905.