Vision Algorithms for Mobile Robotics

Lecture 12c
Deep Learning Tutorial

Leonard Bauersfeld

https://rpg.ifi.uzh.ch

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

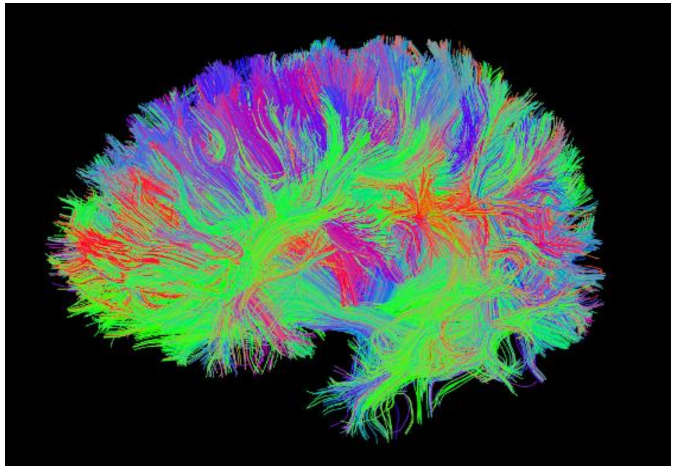
Relevant for the exam

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

The Deep Learning Revolution

Medicine

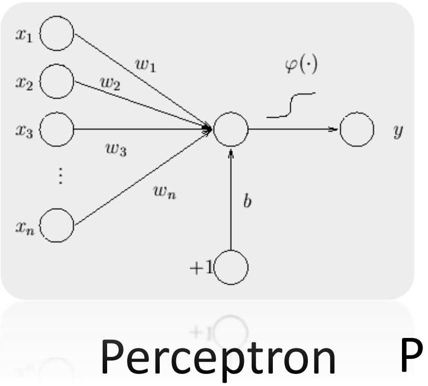


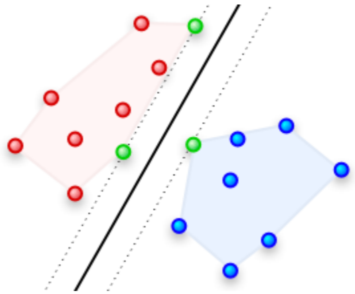
Media & Entertainment

Surveillance & Security

Autonomous Driving

Some History





Perceptron

Back-
Propagation

SVM

AlexNet

1969

AI Winter

1998

2006

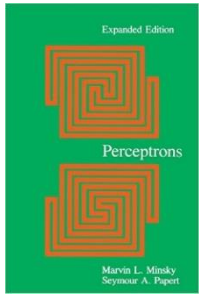
1958

1974

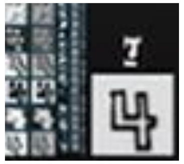
1995

2012

Critics



Convolutional Neural
Networs for Handwritten
Digits Recognition



Restricted
Boltzman
Machines

What changed?

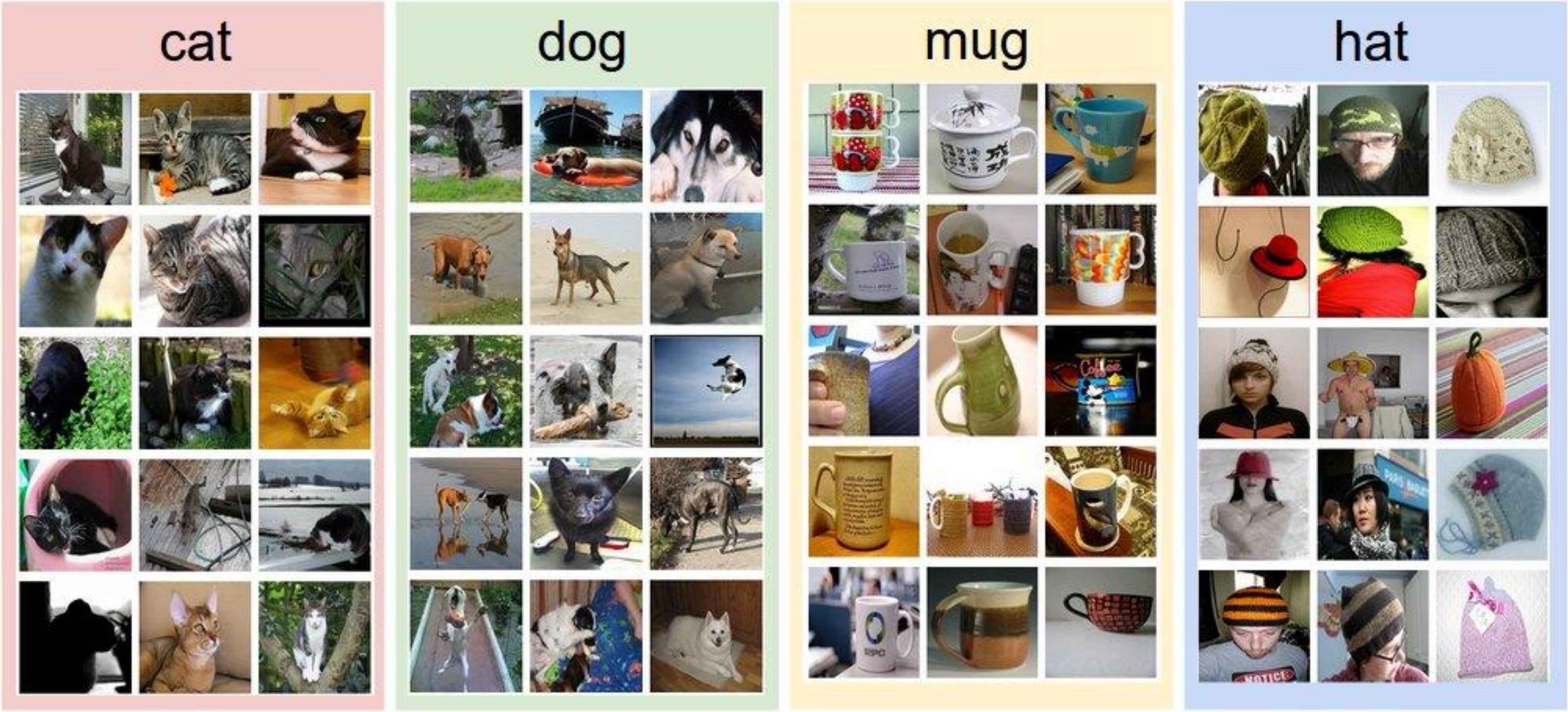
1. Hardware Improvements

2. Big Data Available

3. Algorithmic Progress

Image Classification

Task of assigning an input image a label from a fixed set of categories.



[1] Slide adapted from CNNs for Visual Recognition (Stanford) [Website]

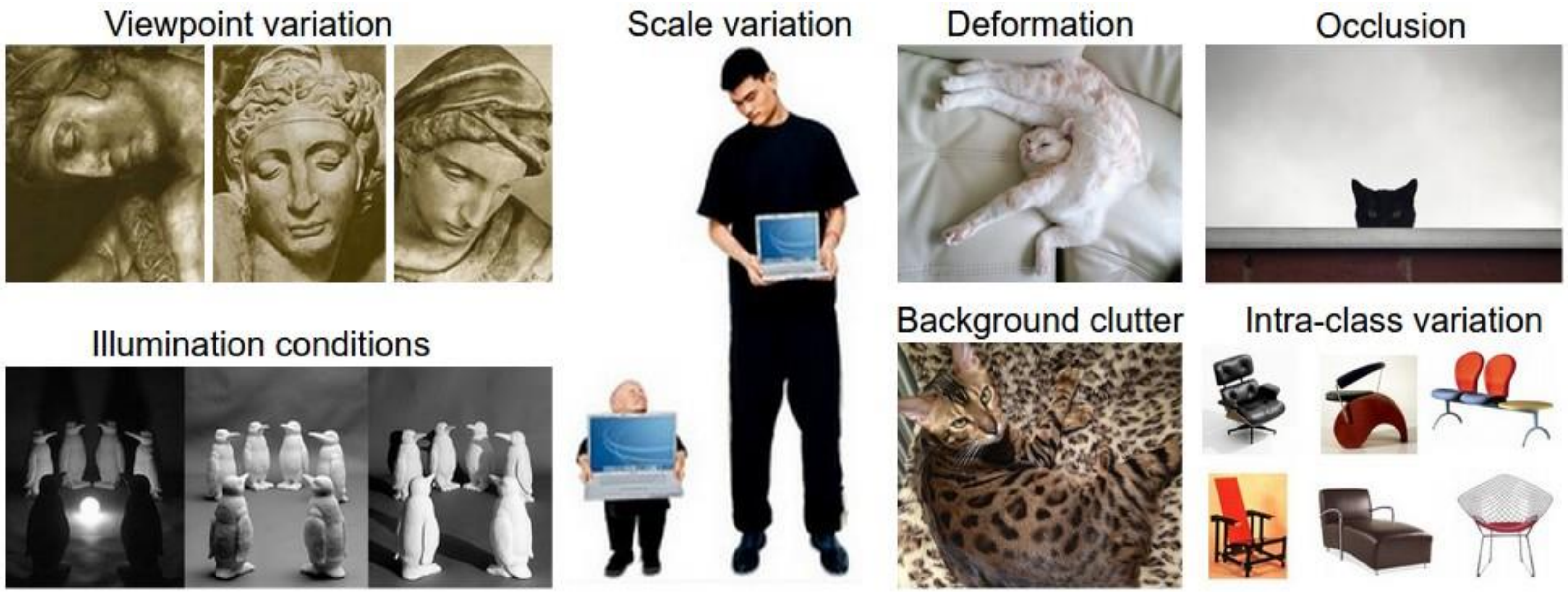
The semantic gap

What computers see compared to what we see

401	402	403	404	405	406	407	408	409	410
411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440
441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460
461	462	463	464	465	466	467	468	469	470
471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490
491	492	493	494	495	496	497	498	499	500

Classification Challenges

Directly specifying how a category looks like is impossible.



We need use a **Data Driven Approach**

Supervised Learning

Find function $f(x, \theta)$ that imitates a ground truth signal

$f(x, \theta)$

N numbers representing class scores

Predicted

$$\begin{pmatrix} 0.1 \\ 0.7 \\ \dots \\ 0.0 \end{pmatrix}$$

Ground truth, y_i

$$\begin{pmatrix} 1.0 \\ 0.0 \\ \dots \\ 0.0 \end{pmatrix}$$

Function parameters or weights

Update

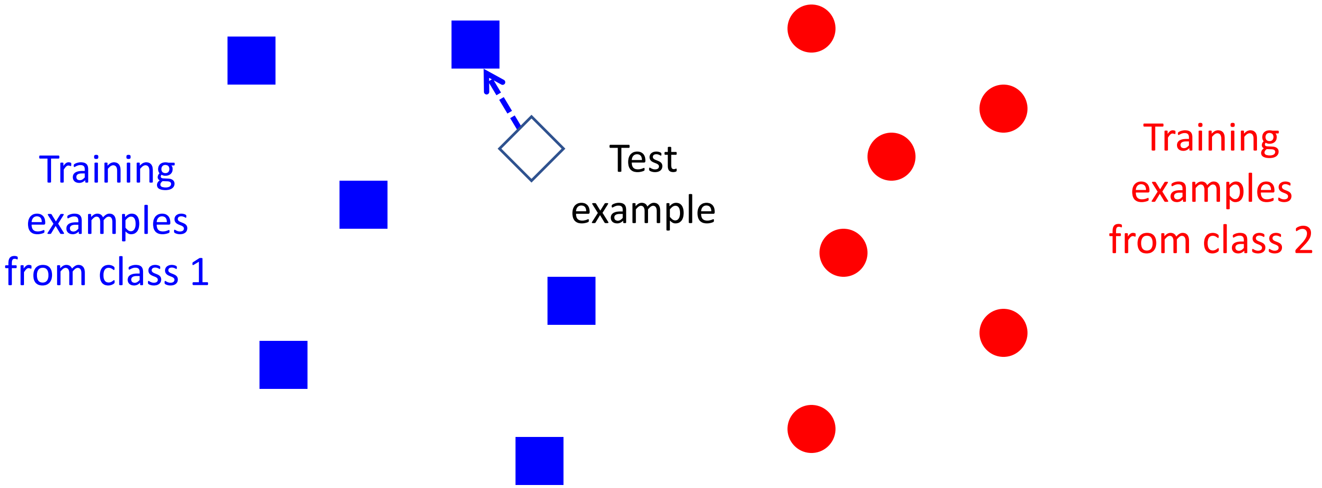
$$Loss(f(x_i, \theta), y_i)$$

Machine Learning Keywords

1. Loss: Quantify how good θ are
2. Optimization: The process of finding θ that minimize the loss
3. Function: Problem modelling → Deep networks are highly non-linear $f(x, \theta)$

Classifiers: K-Nearest neighbor

Features are represented in the descriptor space



$f(\boldsymbol{x}, \theta)$ = label of the K training examples nearest to $\boldsymbol{x}$

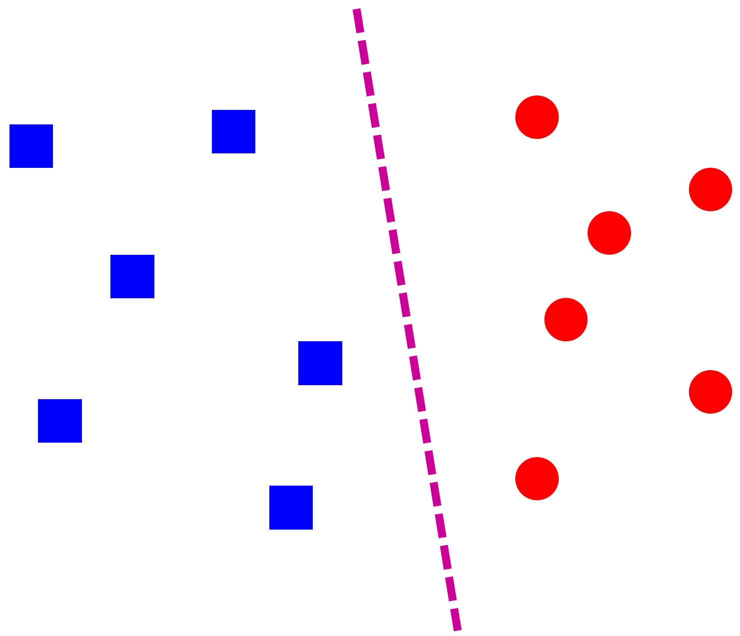
How fast is training? How fast is testing?

- O(1), O(n)

What is a good distance metric ? What K should be used? ☹

Classifiers: Linear

Find a *linear function* to separate the classes:



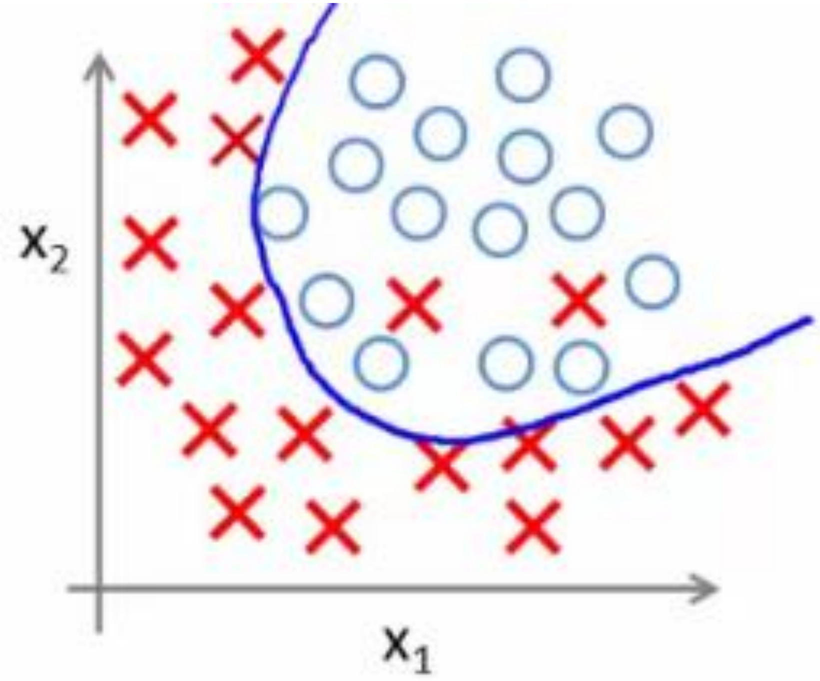
$$f(\boldsymbol{x}, \theta) = sgn(\theta \cdot \boldsymbol{x} + b)$$

What is θ? What is the dimensionality of images?

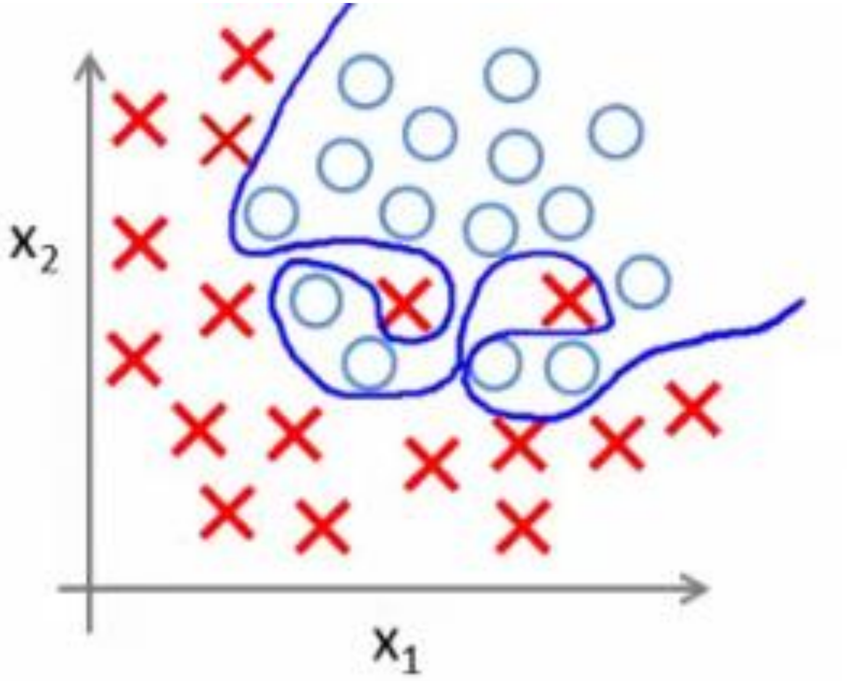
Classifiers: non-linear

What is $f(x, \theta)$?

Good classifier

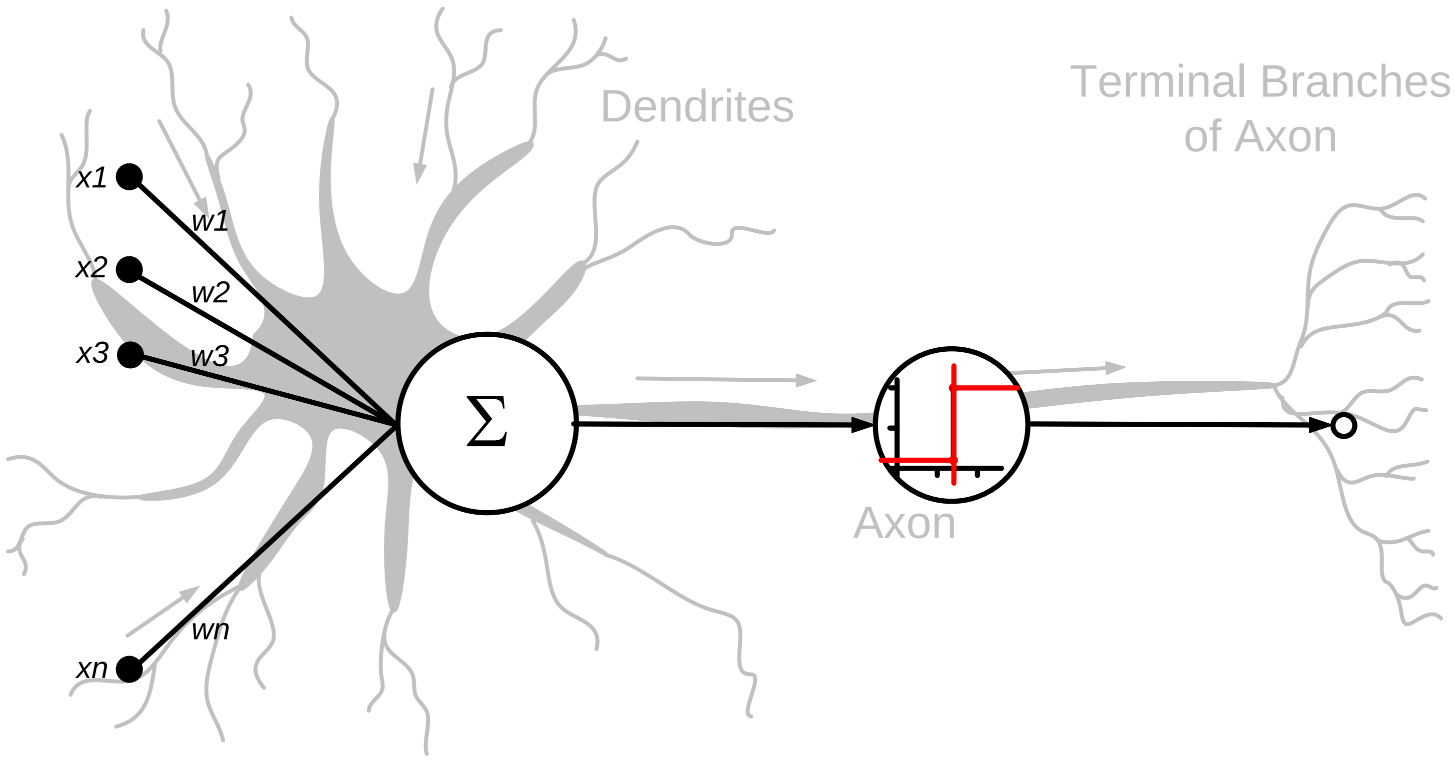


Bad classifier (over fitting)



Biological Inspiration

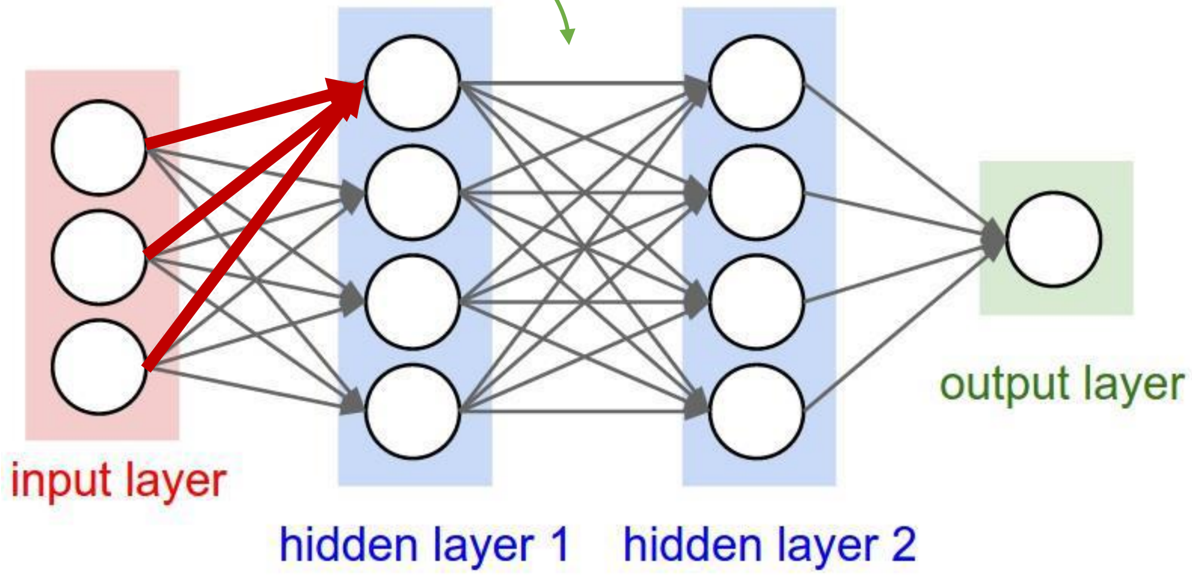
$f(x, \theta) = F(\theta x)$, F is a non-linear activation function (Step, ReLU, Sigmoid)



[1] Rosenblatt, *The perceptron: a probabilistic model for information storage and organization in the brain*, Psychological Review, 1958. PDF

Multi Layer Perceptron

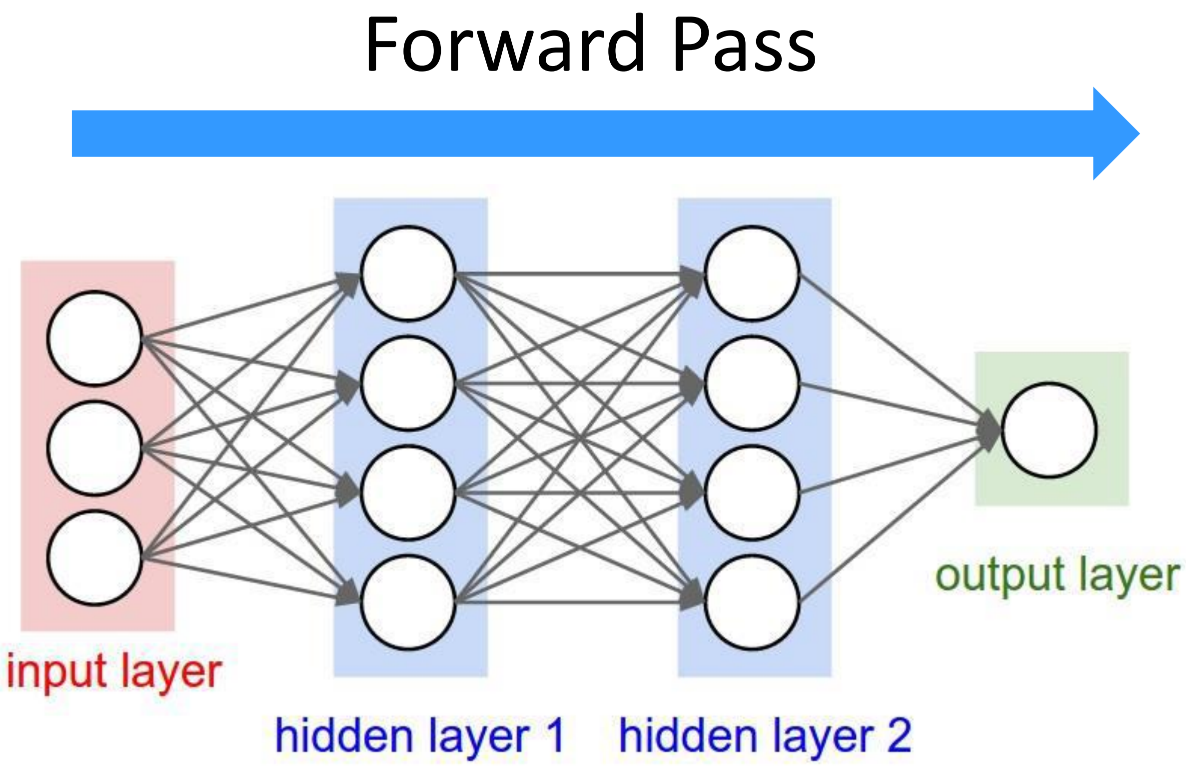
$f(x, \theta)$



Non-linear Activation functions (ReLU, sigmoid, etc.)

[1] Michael Nielsen, *Neural Networks and Deep Learning, Chapter 2* [PDF]

Forward Propagation



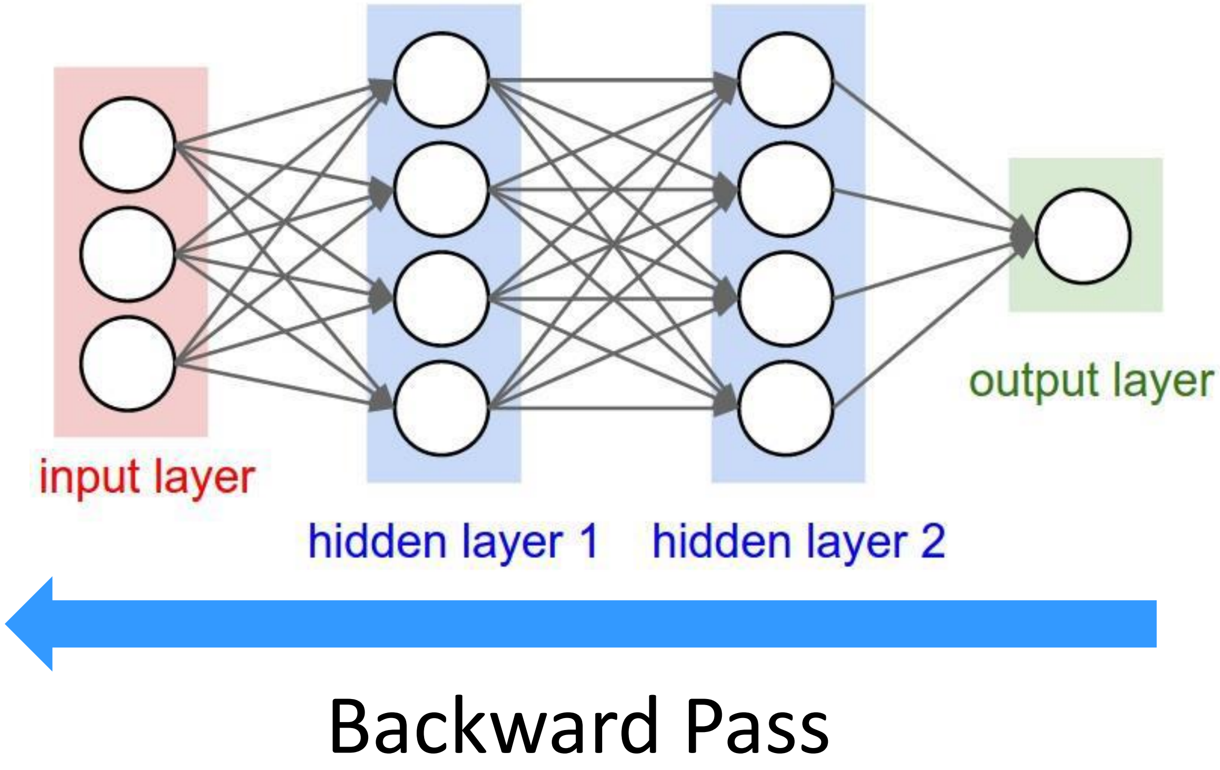
$$Loss(f(x_i, \theta), y_i)$$

[1] Michael Nielsen, *Neural Networks and Deep Learning, Chapter 2* PDF

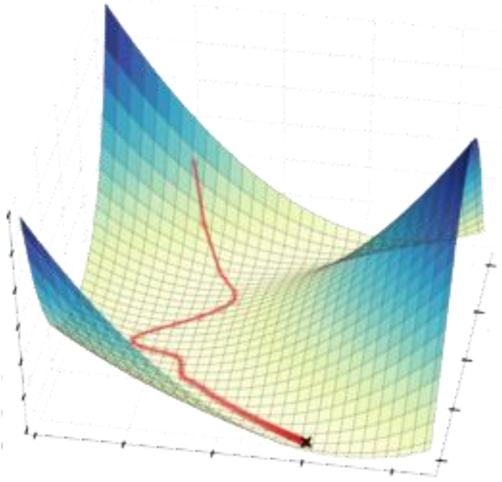
Optimization: Back-propagation

Compute gradients with respect to all parameters and perform **gradient descent**

$$\theta_{new} = \theta_{old} - \mu \nabla_\theta Loss$$



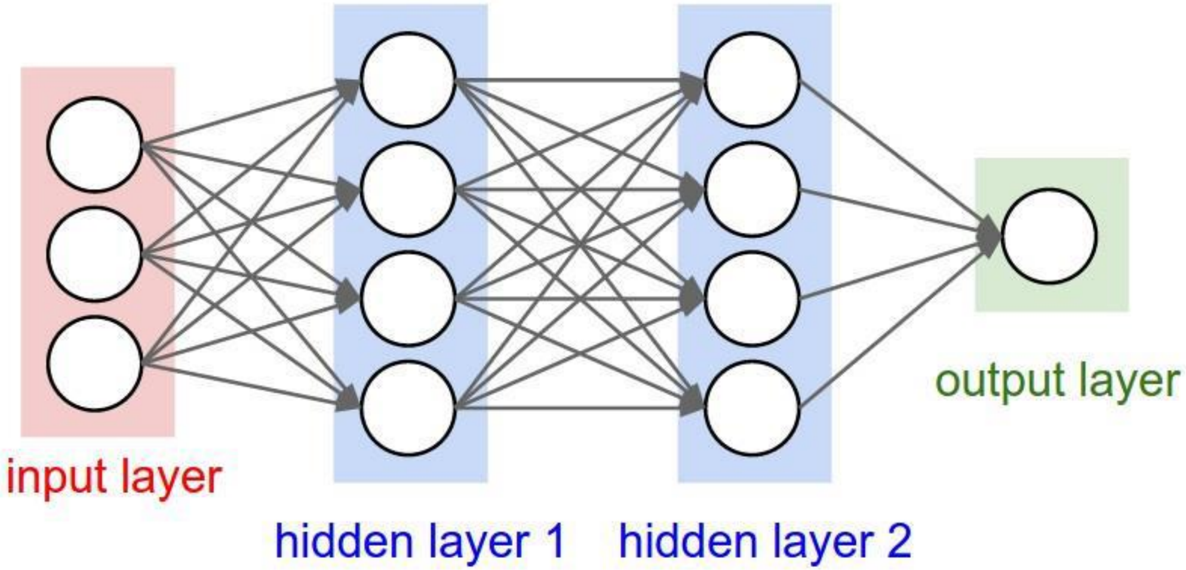
$$Loss(f(x_i, \theta), y_i)$$



[1] Michael Nielsen, *Neural Networks and Deep Learning, Chapter 2* PDF
[2] Dreyfus, *Artificial Neural Networks, Back Propagation and the Kelley-Bryson Gradient Procedure,* Journal of Guidance, 1989. *PDF*

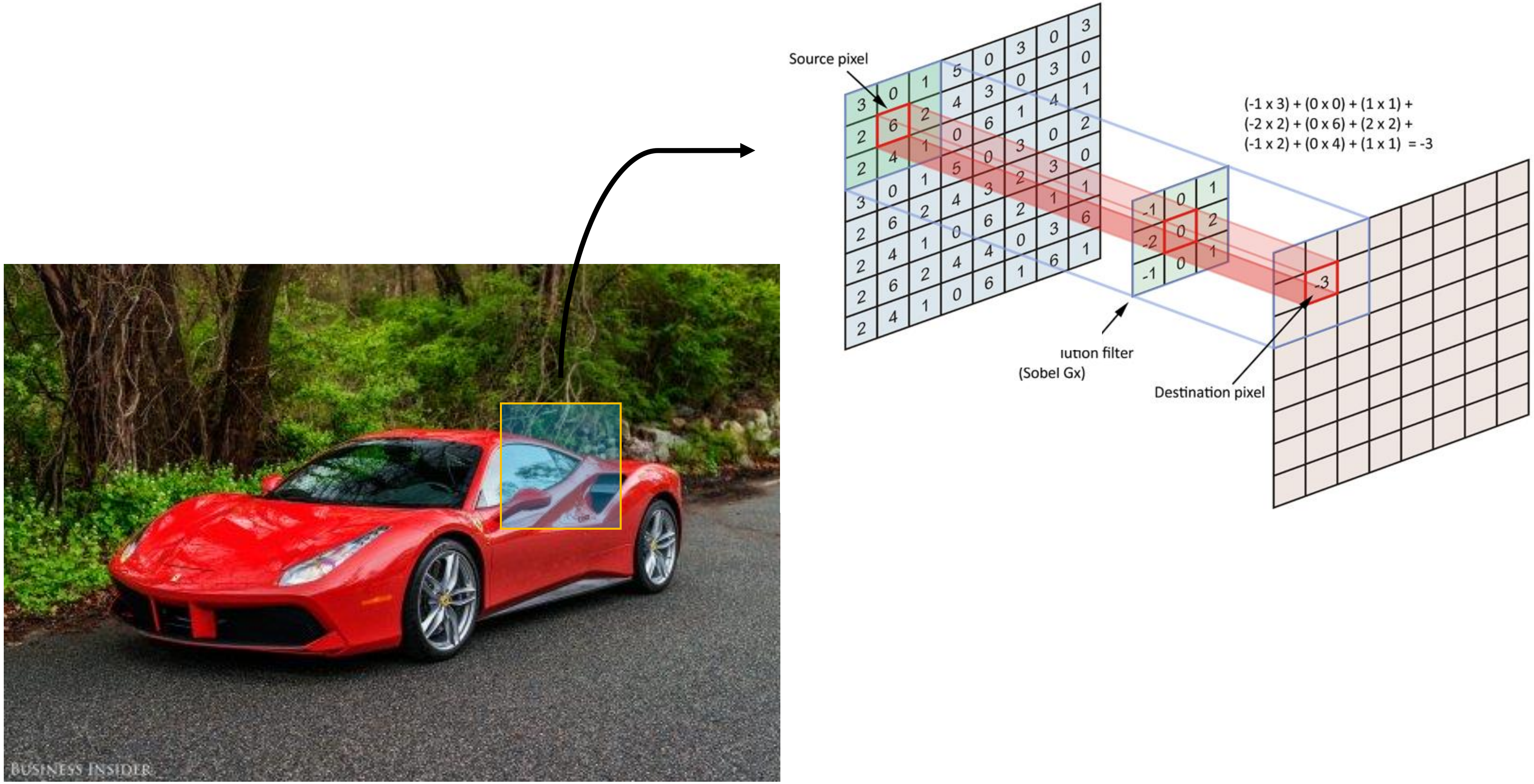
Problems of fully connected network

Too many parameters → **possible overfitting**.



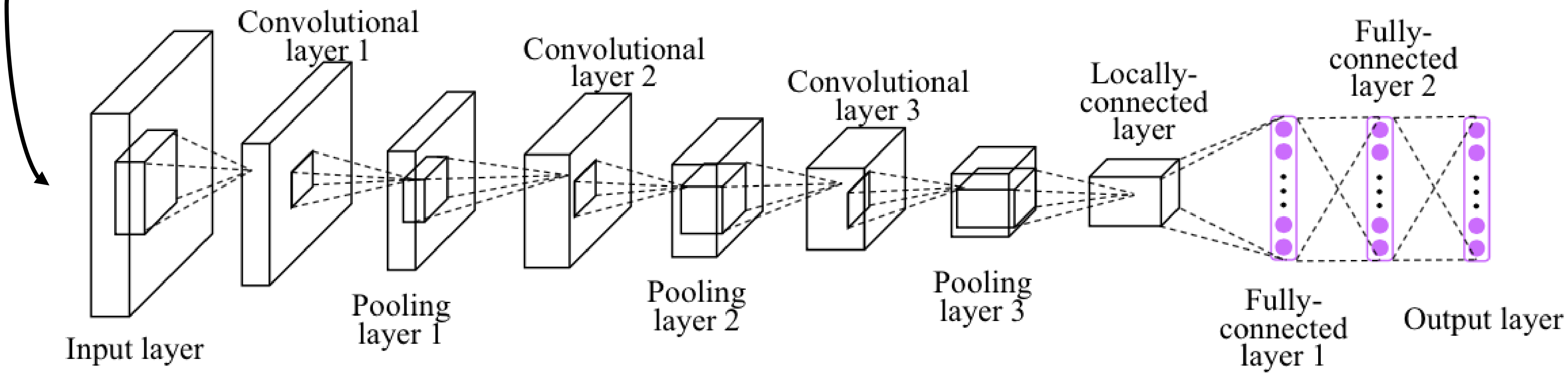
However, we are not using the fact that inputs are images!

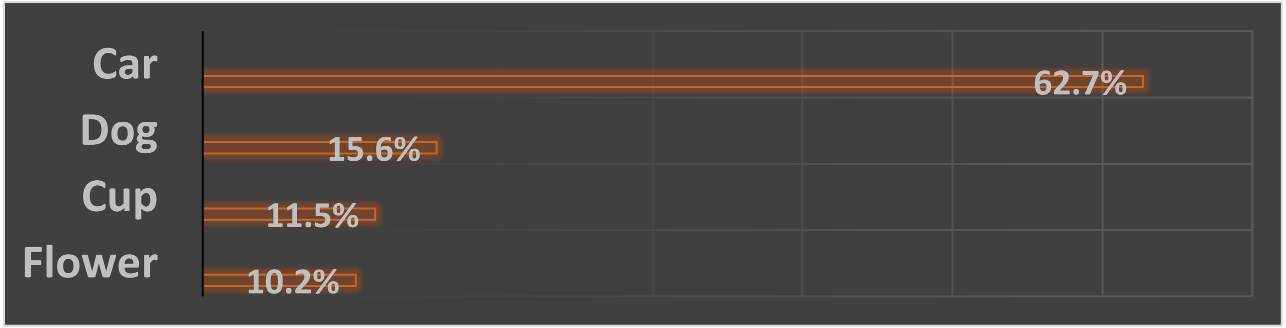
Convolutional Neural Networks



[1] LeCun, Bottou, Bengio, Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 1998. PDF

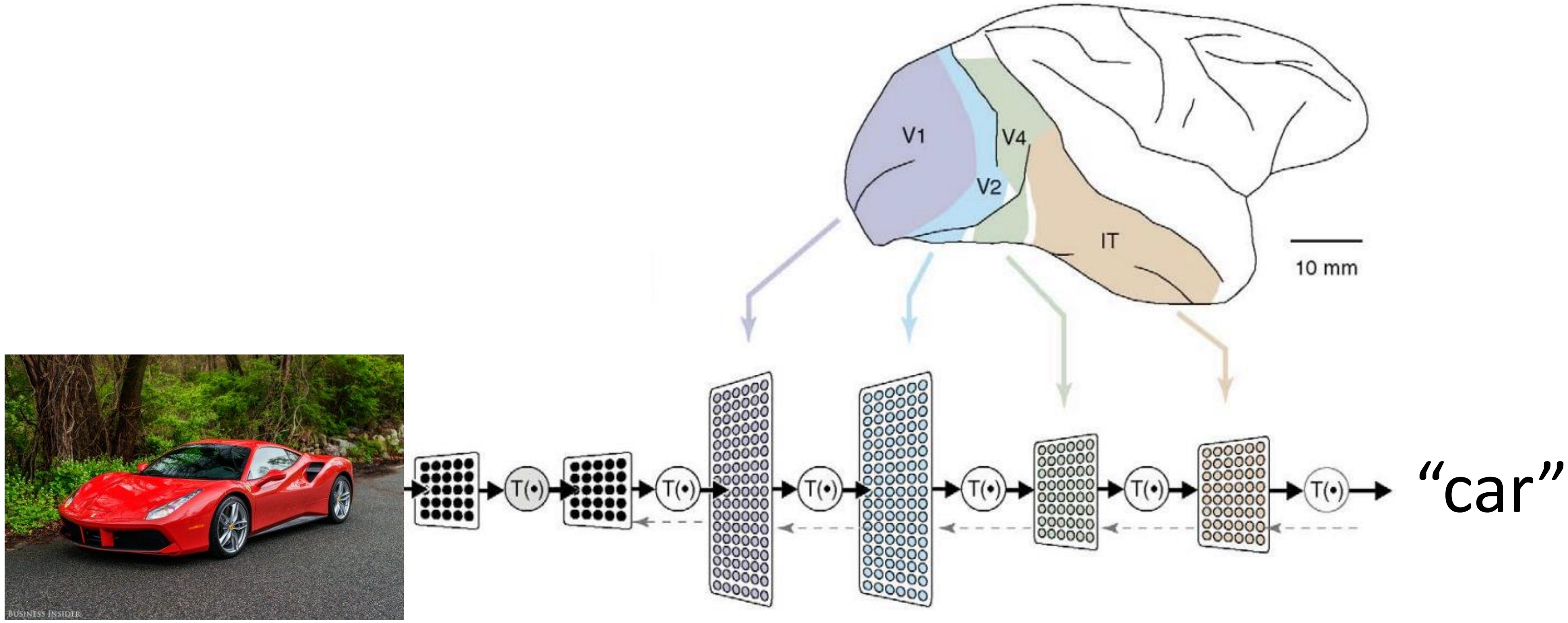
Going Deep


Input layer
Convolutional layer 1
Pooling layer 1
Convolutional layer 2
Pooling layer 2
Convolutional layer 3
Pooling layer 3
Locally-connected layer
Fully-connected layer 1
Fully-connected layer 2
Output layer


Car
62.7%
Dog
15.6%
Cup
11.5%
Flower
10.2%

Why Deep?

1. Inspired by the **human visual system**
2. Learn **multiple layers** of transformations of input
3. Extract progressively more **sophisticated representations**

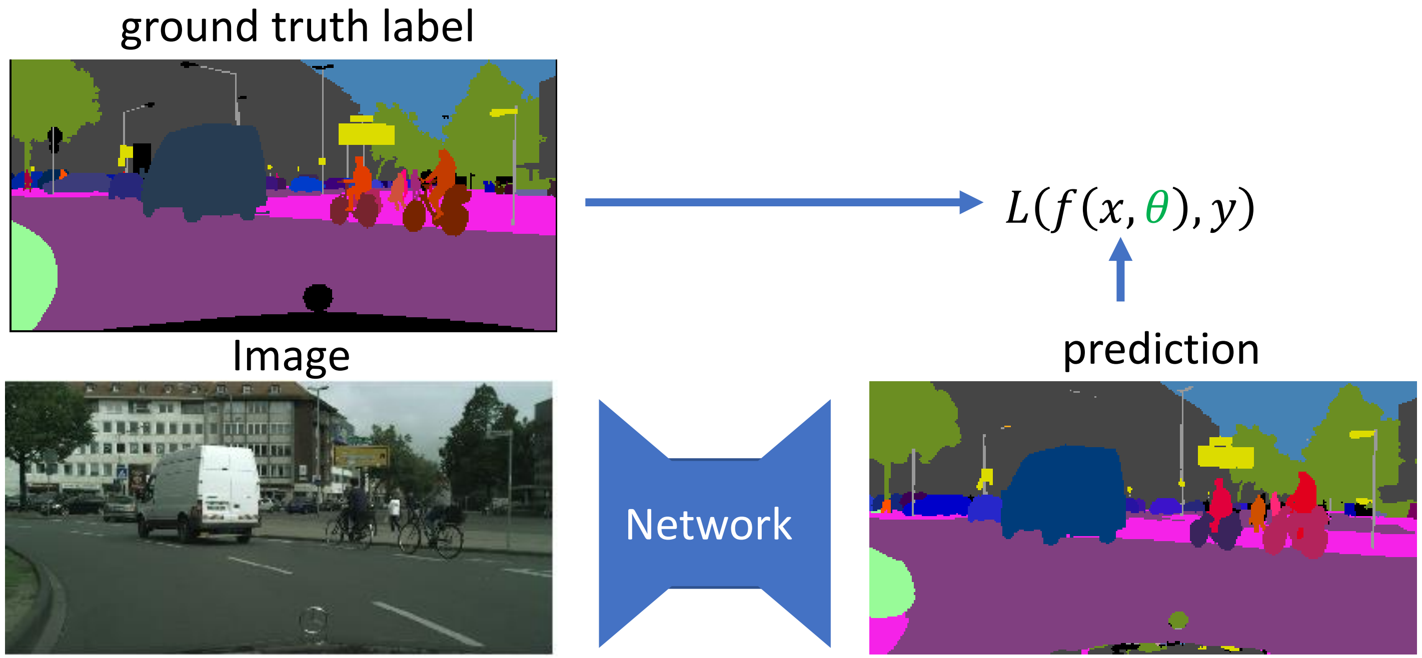


Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

Supervised Learning

- In supervised learning we assume have access to both input data or **images** and **ground truth labels**.
- Networks trained with supervision usually perform best
- However, getting ground truth is hard, since it often must be **hand-labelled**



Supervised Learning

- Image Segmentation

[1] Long, Shelhamer, *Fully Convolutional Networks for Semantic Segmentation,* Conference of Computer Vision and Pattern Recognition *(CVPR)*, 2015. [PDF](PDF)

Supervised Learning

- Image Captioning

"little girl is eating piece of cake."

"baseball player is throwing ball in game."

"woman is holding bunch of bananas."

"black cat is sitting on top of suitcase."

"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

"a horse is standing in the middle of a road."

[1] Karpathy, Fei-Fei, *Deep Visual-Semantic Alignments for Generating Image Descriptions,* Conference of Computer Vision and Pattern Recognition (CVPR), 2015. PDF

Supervised Learning

- Image Localization

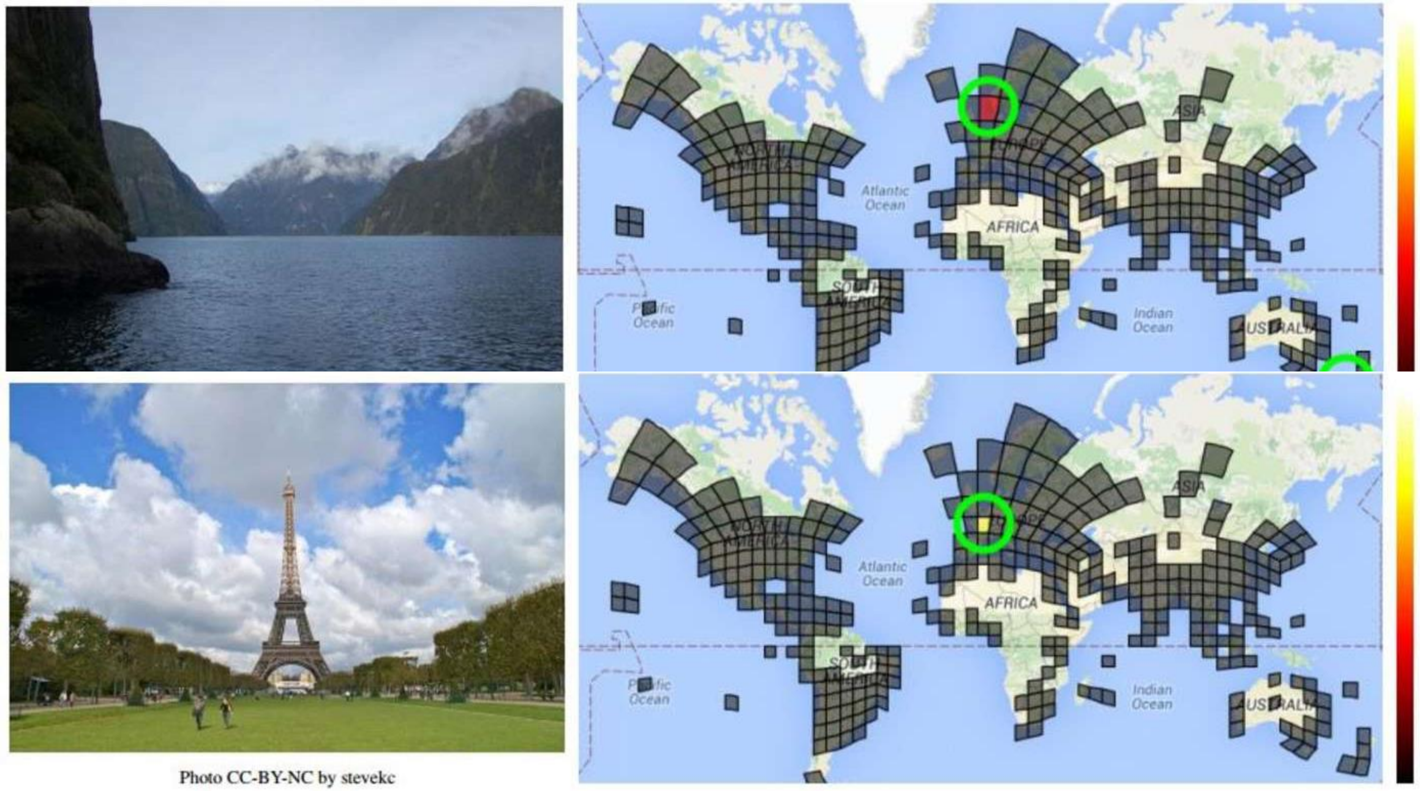


Photo CC-BY-NC by stevekc

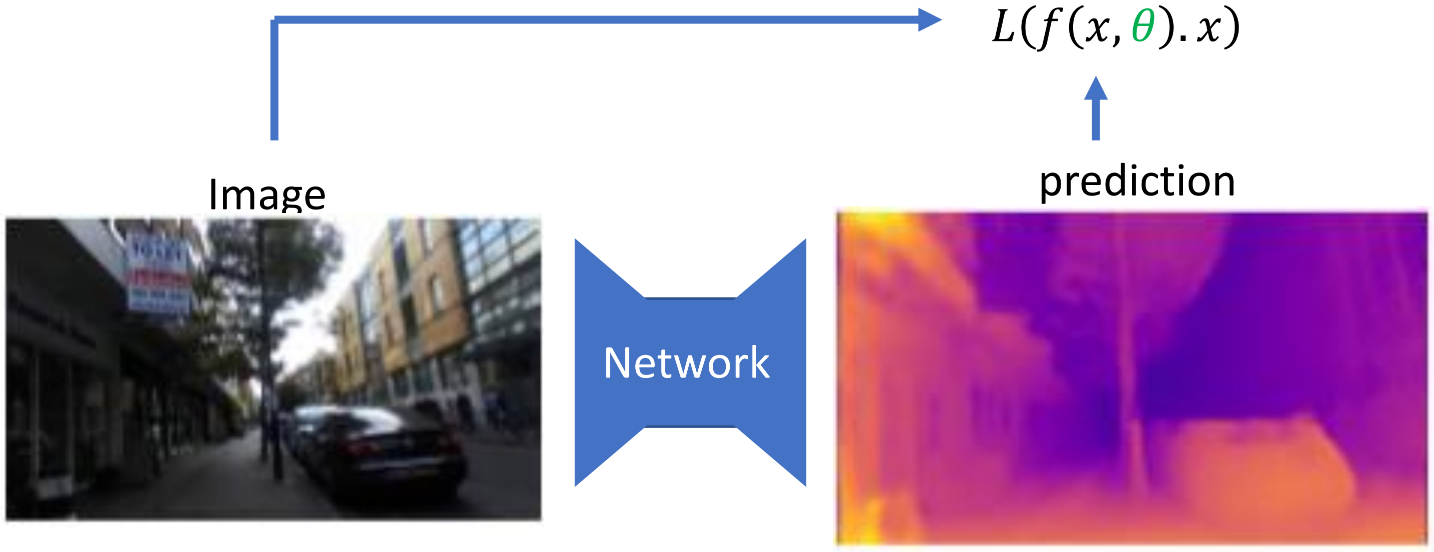
[1] Weyland, Kostrikov, Philbin, *PlaNet - Photo Geolocation with Convolutional Neural Networks,* European Conference on Computer Vision (ECCV), 2016. PDF

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

Unsupervised Learning

- In unsupervised learning we only have access to input data or **images**.
- Usually, these methods are more popular because they can use much larger datasets that do not need to be manually labelled.



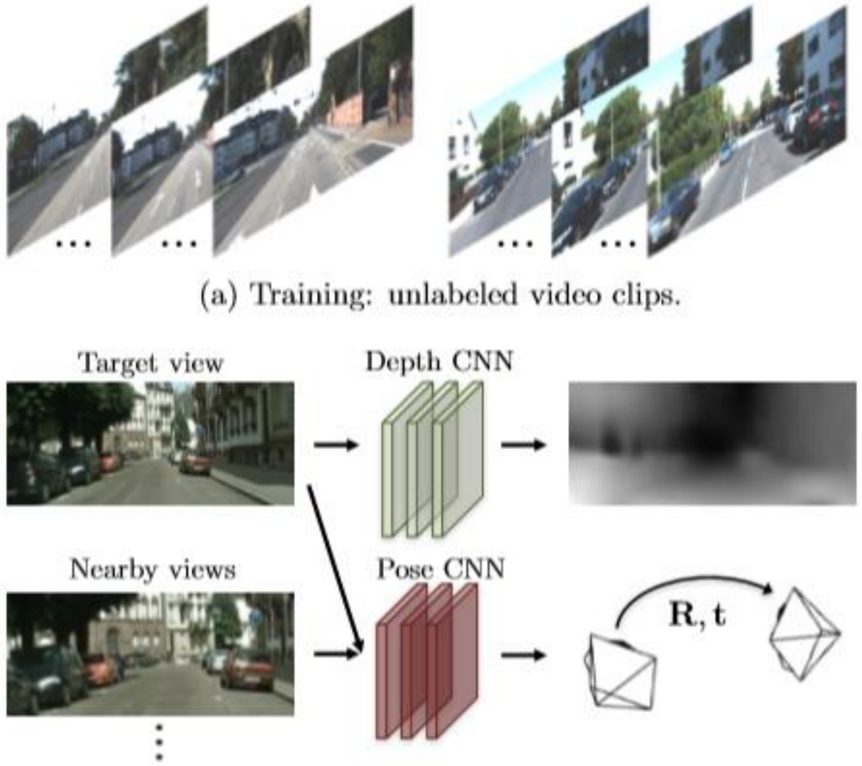
Unsupervised Learning

- Monocular Depth Estimation

[1] Godard, Mac Aodha, Brostow, *Unsupervised Monocular Depth Estimation with Left-Right Consistency* , Conference of Computer Vision and Pattern Recognition (CVPR), 2017. PDF

Unsupervised Learning

- Structure from Motion



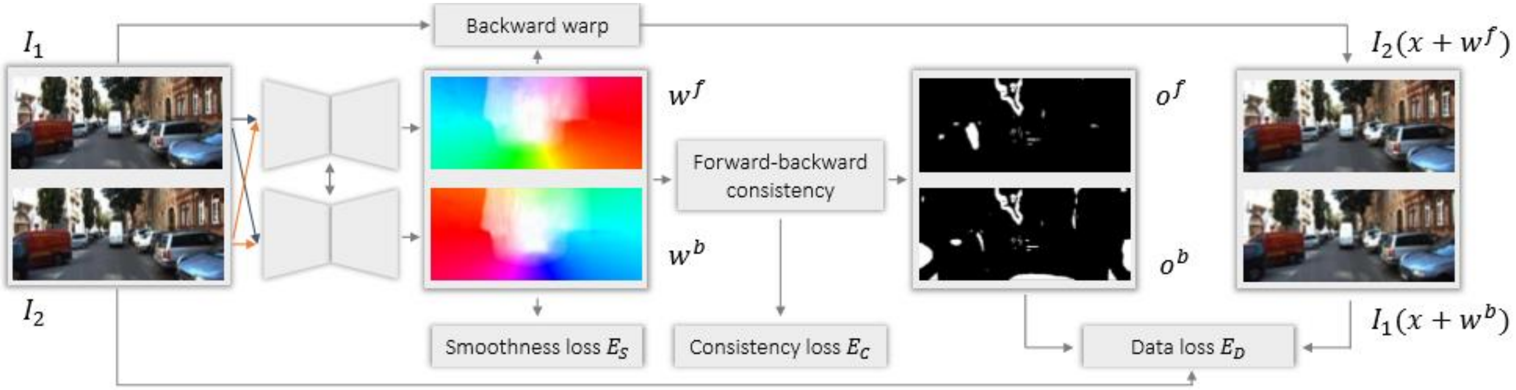
[1] Zhou, Brown, Snavely, Lowe, *Unsupervised Learning of Depth and Ego-Motion from Video,* Conference of Computer Vision and Pattern Recognition (CVPR), 2017. PDF

Unsupervised Learning

- Dense Optical Flow

Characteristic of the learned flow:

- Robustness against light changes (Census Transform)
- Occlusion handling (Bi-directional Flow)
- Smooth flow



[1] Meister, Hur, Roth, *Unsupervised Learning of Optical Flow with a Bidirectional Census Loss*, Association for the Advancement of Artificial Intelligence (AAAI), 2018. PDF

Unsupervised vs. Supervised learning

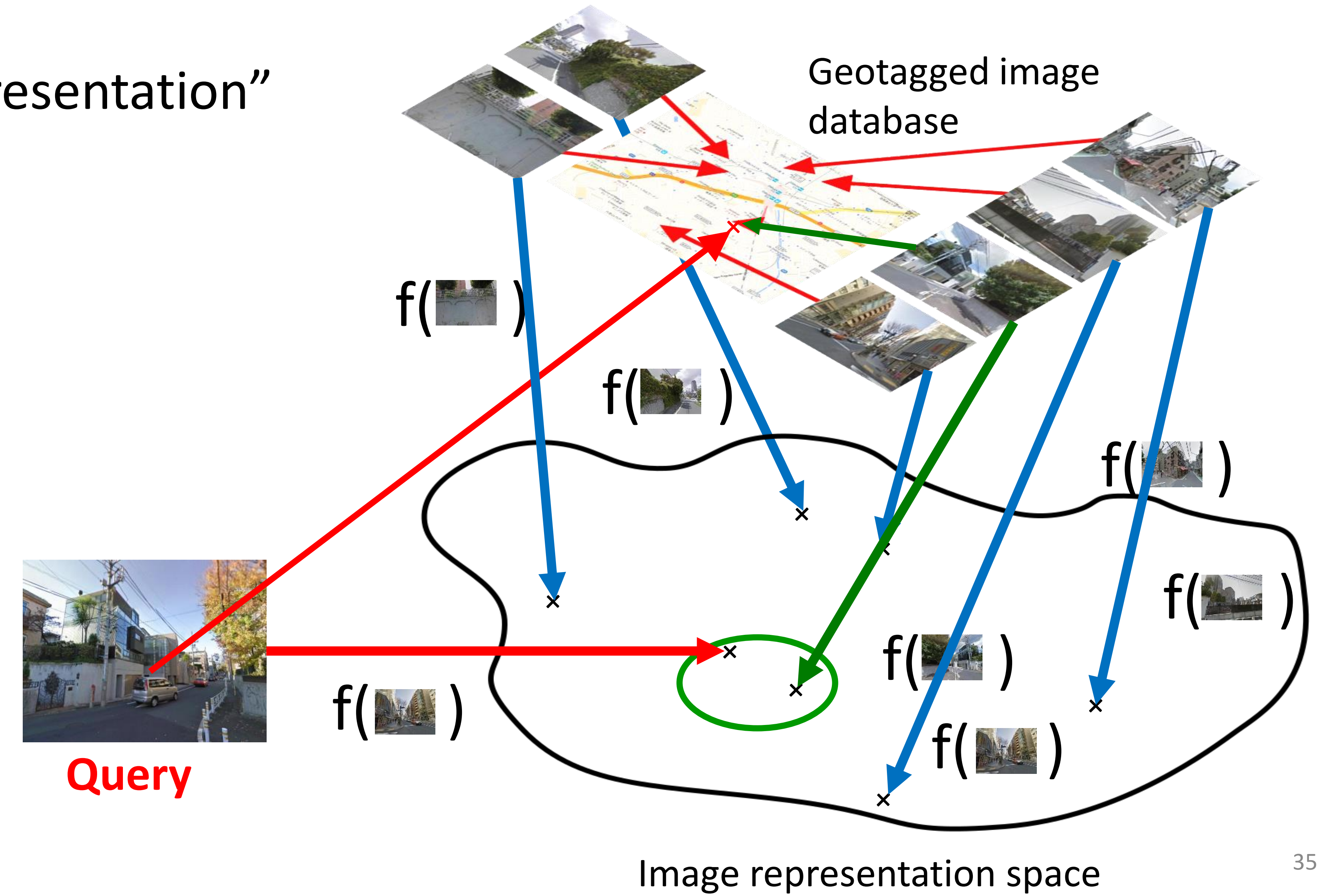
	Supervised	**Unsupervised**
Performance	Usually better for the same dataset size.	Usually worse, but can outperform supervised methods due to larger data availability.
Data availability	Low, due to manual labelling.	High, no labelling required.
Training	Simple, ground truth gives a strong supervision signal.	Sometimes difficult, loss functions have to be engineered to get good results.
Generalizability	Good, although sometimes the network learns to blindly copy the labels provided, leading to poor generalizability.	Better, since unsupervised losses often encode the task in a more fundamental way.

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

Place Recognition – NetVLAD

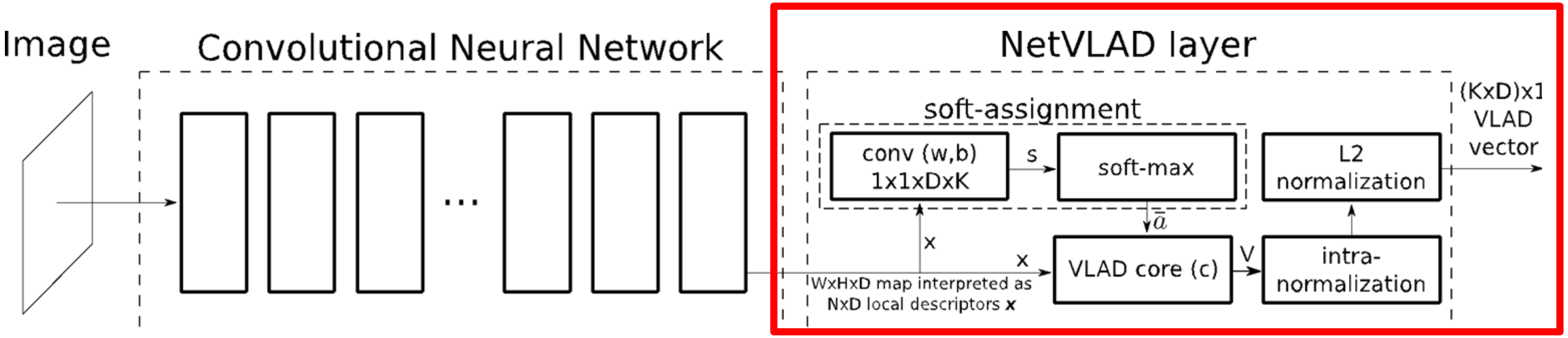
- Design an “image representation” extractor $f(I, \theta)$



NetVLAD – Method

- Mimic the classical pipeline with deep learning

Image I | Extract local features (SIFT) | Aggregate (BoW, **VLAD**, FV) | F(I)



Trainable pooling layer

Arandjelović, Gronat, Torii, Pajdla, Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, Conference of Computer Vision and Pattern Recognition (CVPR), 2016. PDF

NetVLAD – Loss

- Triplet loss formulation

$D_p = ||F_\theta($  $) - F_\theta($  $)||^2$ ⟶ Matching samples

$D_n = ||F_\theta($  $) - F_\theta($  $)||^2$ ⟶ Non matching samples

$$L_\theta = \sum_{samples} \max(D_{p(\theta)} + \underset{\text{margin}}{m} - D_{n(\theta)}, 0)$$

Disclaimer: The actual NetVlad loss is a slightly more complicated version of the one above

Arandjelović, Gronat, Torii, Pajdla, Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, Conference of Computer Vision and Pattern Recognition (CVPR), 2016. PDF

NetVLAD – Results

- Code, dataset and trained network online: give it a try here!

Query

Top result

Green: Correct   Red: Incorrect

Arandjelović, Gronat, Torii, Pajdla, Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, Conference of Computer Vision and Pattern Recognition (CVPR), 2016. PDF

Novel View Synthesis – Neural Radiance Fields (NeRF)

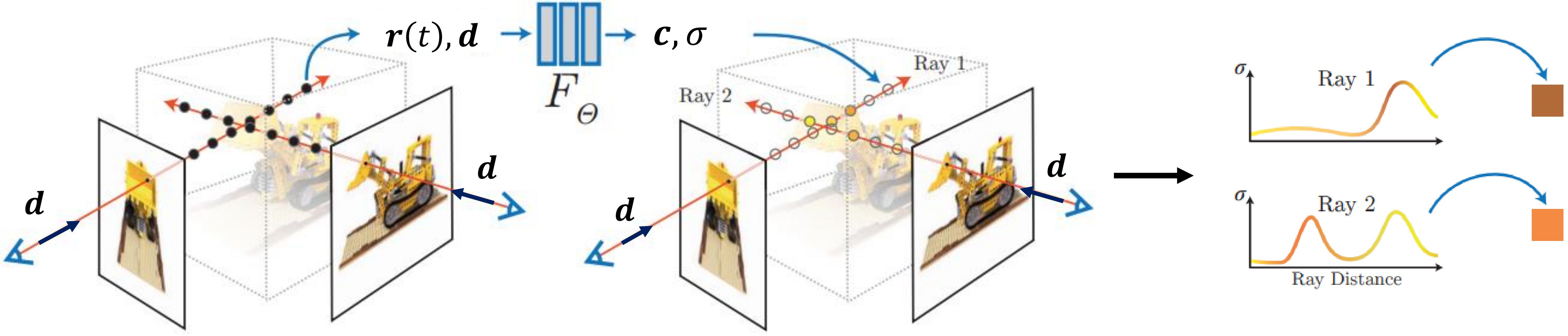
Render new views from a set of images with corresponding poses.

images with poses | reconstructed neural radiance field | synthesized novel viewpoints

[1] Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng, *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, European Conference of Computer Vision (ECCV), 2020. PDF
[2] An overview and a reference for many follow-up works can be found here

Neural Radiance Fields (NeRF): Method

Images are rendered by integrating *transmittance and color* along a ray both of which are modelled with a *multilayer perceptron*



Position along ray: $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$

Transmittance: $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))ds\right)$

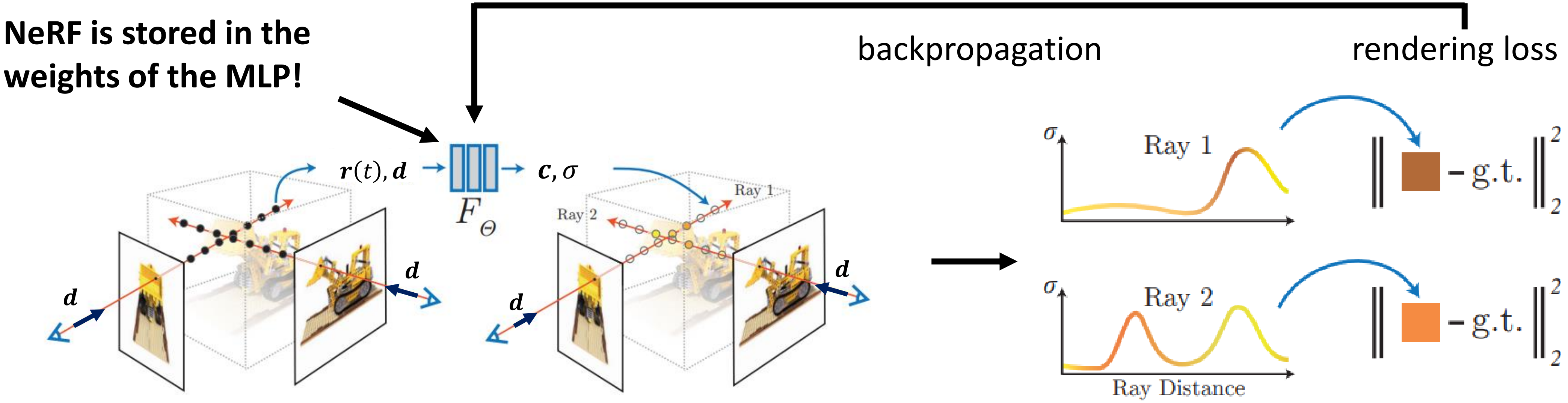
Ray direction: $\boldsymbol{d}$

Color: $C(r) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})dt$

[1] Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng, *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, European Conference of Computer Vision (ECCV), 2020. PDF
[2] An overview and a reference for many follow-up works can be found here

Neural Radiance Fields (NeRF): Training

We train this multilayer perceptron by minimizing the *rendering loss* on the input images, thereby effectively *overfitting.*



[1] Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng, *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, European Conference of Computer Vision (ECCV), 2020. PDF

[2] An overview and a reference for many follow-up works can be found here

Neural Radiance Fields (NeRF): Results

Compared to previous approaches, NeRF generates highly photorealistic, and consistent novel views.

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

Conclusions

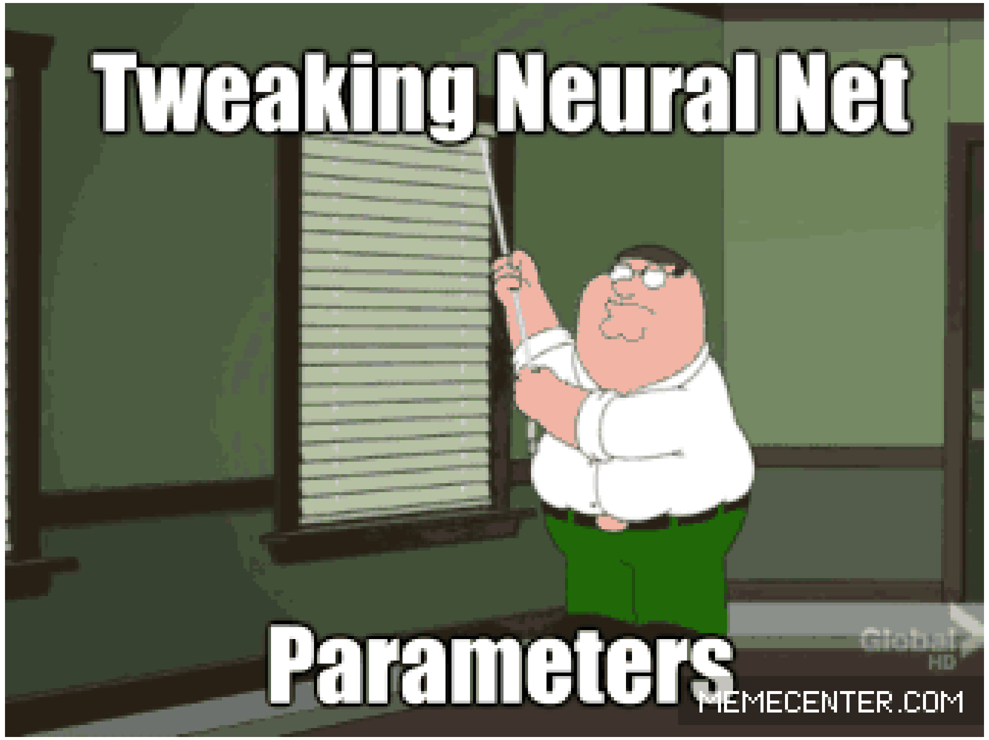
Deep learning, when applied in the correct circumstances, can achieve remarkable performance on a variety of tasks by *learning patterns from data*

It works especially well when

- Sufficient data is available
- All operations are differentiable

Make sure to avoid the following pitfalls:

- Make sure to optimize the correct metric
- Test your model to an inch of its life
- Always monitor generalization



Additional Readings

- Nielsen, *Neural Networks and Deep Learning*, 2018. PDF
- Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*, 2012. PDF
- Goodfellow, Bengio, Courville, *Deep Learning*, 2016 Website

Outline

- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

The drone market is valued $24 billions today

Inspection

Agriculture

Transport

Search and Rescue

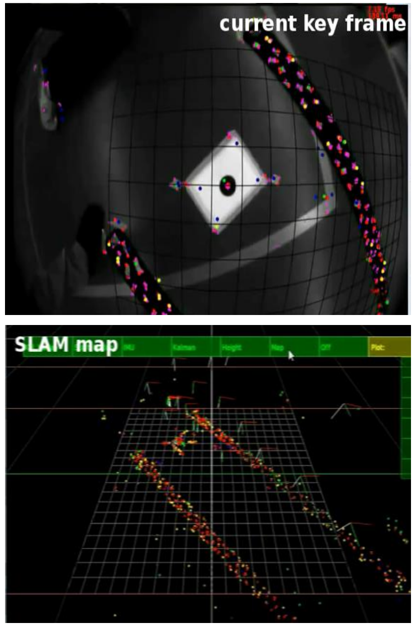
Source: Swiss Drone Industry Report 2021, p. 22:
https://drive.google.com/file/d/1ljesolDoUu1-IVX14nqJRCT-wpEQB22_/view

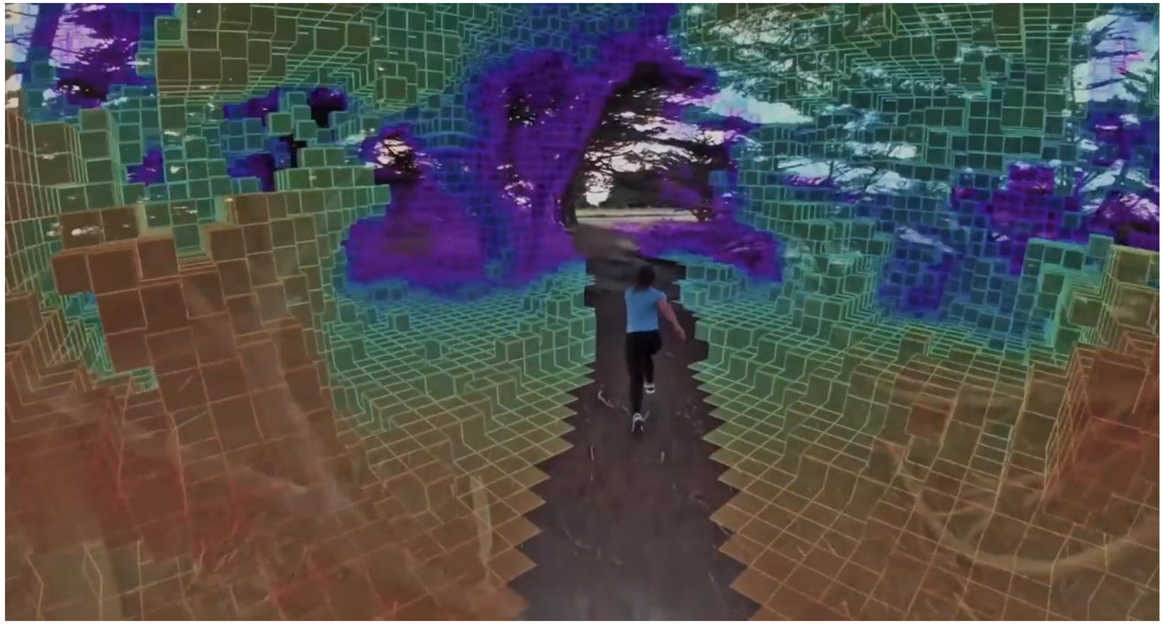
How are current commercial drones controlled?

- **By a human pilot**
  - requires **line of sight** or **video link**
  - requires a **lot of training**

- **By an autopilot:** autonomous navigation
  - **GPS**: doesn’t work in GPS denied or degraded environments
  - **Lidar** (e.g., Exyn): expensive, heavy, power hungry
  - **Cameras** (e.g., Parrot, DJI, Skydio): cheap, lightweight, passive (i.e., low power)

Last 10-years Progress on Autonomous Vision-based Flight





2010
EU SFLY Project (2009-2012)

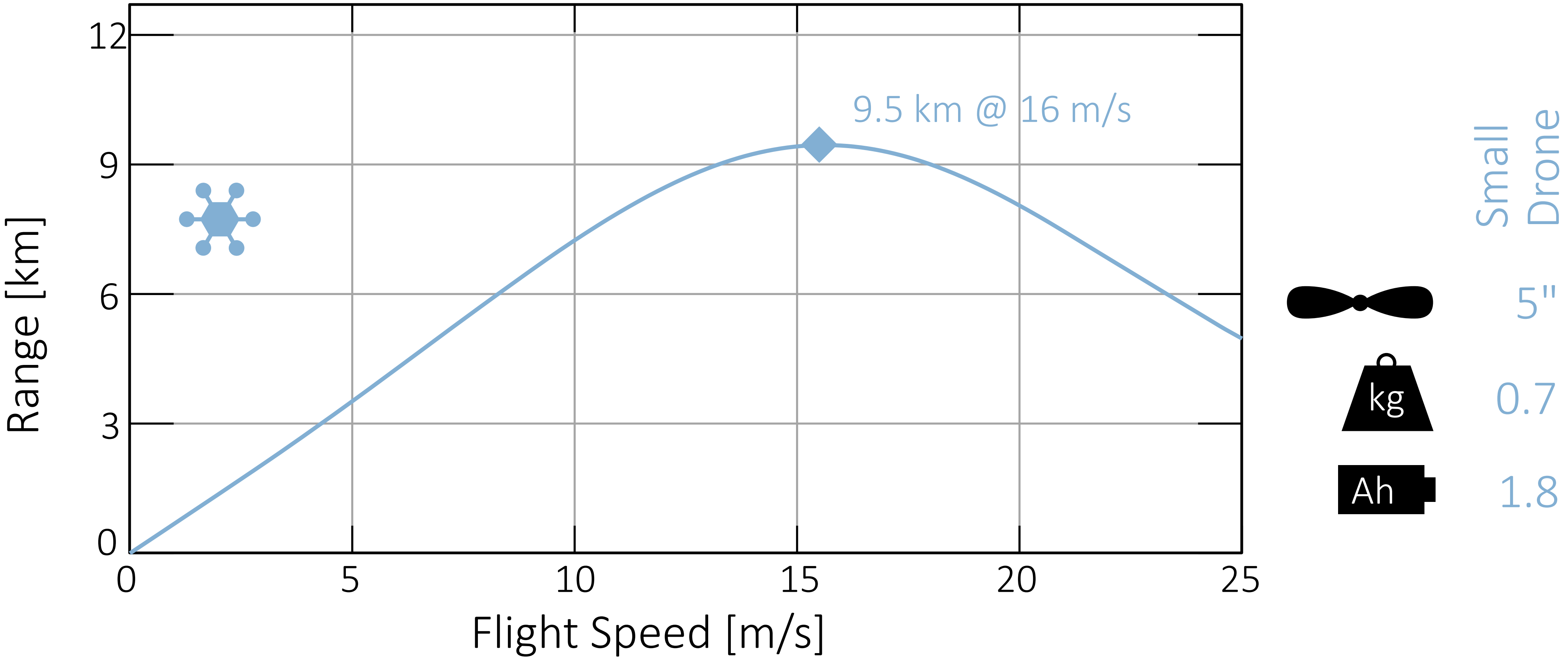
[Bloesch, ICRA 2010]
1st onboard goal-oriented vision-based flight
(previous research focused on reactive navigation)

2020

- **Skydio** (2018-2020),
- **DJI** (2018-2020),
- **NASA Mars Helicopter** (2020)

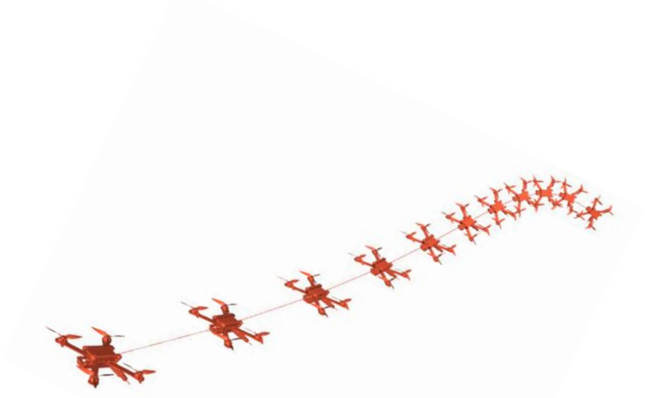
1st products in the market or sent to another planet ☺

Flying Fast to Fly Far



Bauersfeld, Scaramuzza, *Range, Endurance, and Optimal Speed Estimates for Multicopters*, IEEE RAL, 2022. PDF

Related Work: The Traditional Approach


Images
IMU

Perception → Planning → Control → Command

State Estimation
Mapping

Polynomial Trajectories
Trajectory Optimization

PID Control
Model Predictive Control
Reinforcement Learning
Iterative Learning Control

Related Work: The Problem

Images
IMU

Perception → Planning → Control

Command

Mature Algorithms, but brittle during high speed due to motion blur.

Require strong assumptions about the environment (e.g, CAD of scene).

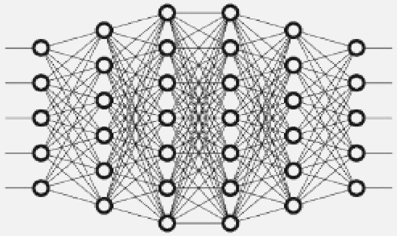
Needs significant tuning, especially at high speed.

! This fine-grained modularity makes the robotic system fragile:
The modules do not interact with each other.

Related Work: End-to-End Learning

Images
IMU

Neural Network



Command

High-Level commands (forward, left, right)

Low-Level commands (collective thrust, bodyrates)

! Don’t exploit the agile dynamics of the drone

! Too sample inefficient to be used on a physical drone. Only shown in Sim.

Our Research

Augment the traditional robotic cycle with learning-based methods.

Hypothesis:
Neural Networks can distill the knowledge of mature robotics algorithms into computationally efficient and robust sensorimotor policies.

Projects

- Learning High-Speed Flight in the Wild

- NeuroBEM: Hybrid Aerodynamic Quadrotor Model

- Autonomous Drone Racing

Learning High-Speed Flight in the Wild

What does it take to achieve similar **spatial awareness** to a human **with comparable sensing (and computing)** in the context of **high-speed flight**?

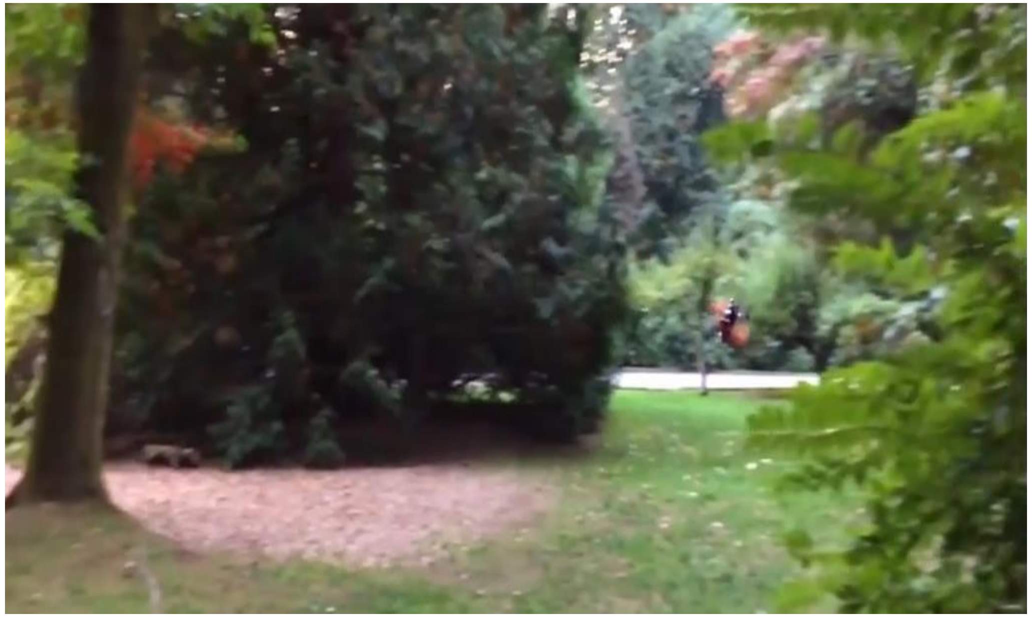
Assumptions:

- No external sensing or computing.
- Test environment not seen in advance.
- Possibly dynamic environment.

Available Information:

- Visual Feedback (multiple cameras).
- Inertial Feedback.
- An intention (e.g. fly straight).

Human pilots fly under similar assumptions!

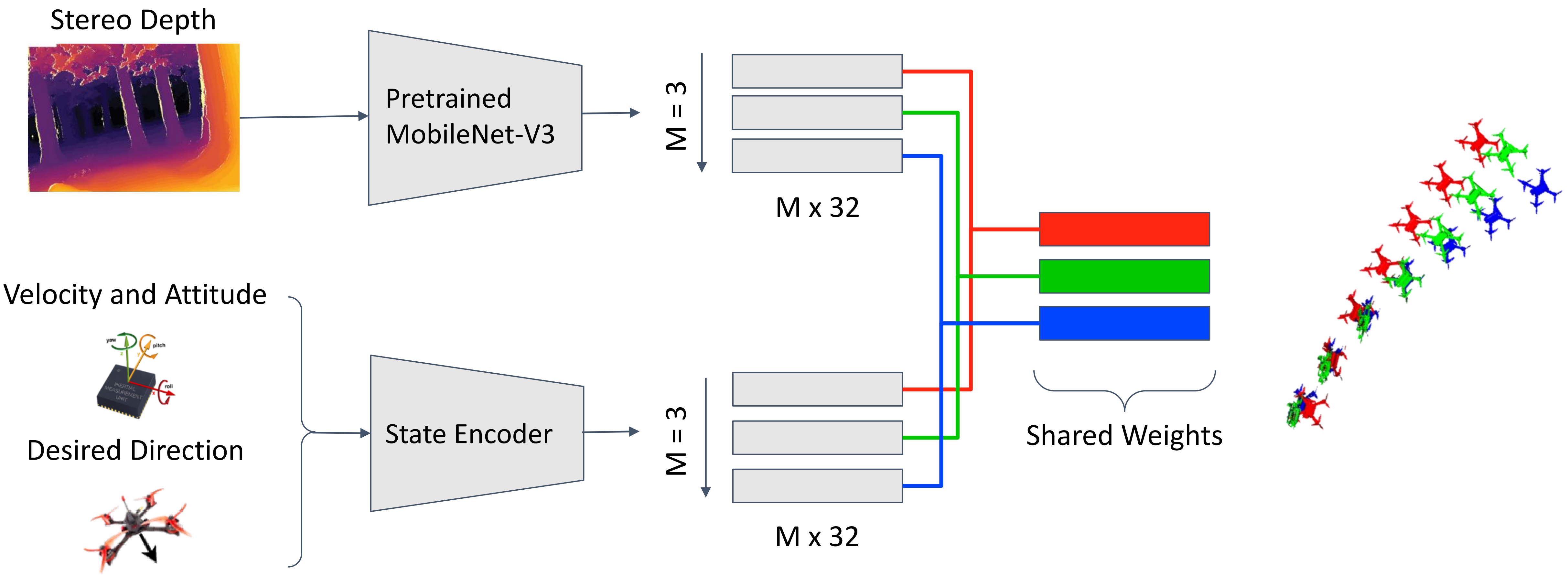


Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. [PDF](), [Video](), [Code]()

Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild,
Science Robotics, 2021. [PDF](), [Video](), [Code]()

Multiple-Hypothesis Action Prediction

We predict collision-free **receding-horizon trajectories** using a **neural network** with access to visual and inertial observations, as well as a reference velocity.



Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. [PDF](), [Video](), [Code]()

Training Procedure

We follow the **privileged learning paradigm** to train the network **purely in simulation***.

1. **Design an expert planner with access to full knowledge of the environment.**
   This expert uses a fine-grained point-cloud of the scene to find collision-free trajectories with sampling.

2. **Distill the knowledge of the expert into a deep neural network**.
   Basically do imitation learning from a set of expert demonstrations.

This simple idea hides quite some challenges!

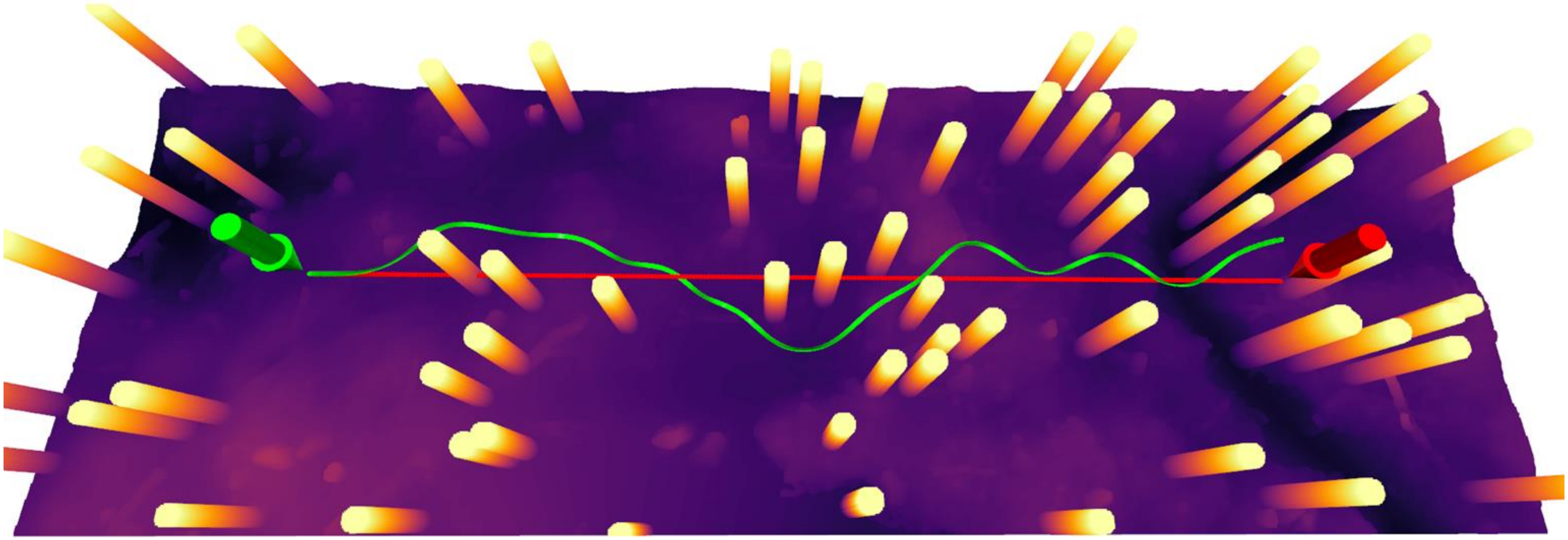
***** Impossible to collect a dataset of real-world demonstrations since it is not possible (or very expensive) to have a perfect map of the environment.

Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. PDF, Video, Code

Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild,
Science Robotics, 2021. PDF, Video, Code

Controlled Experiments

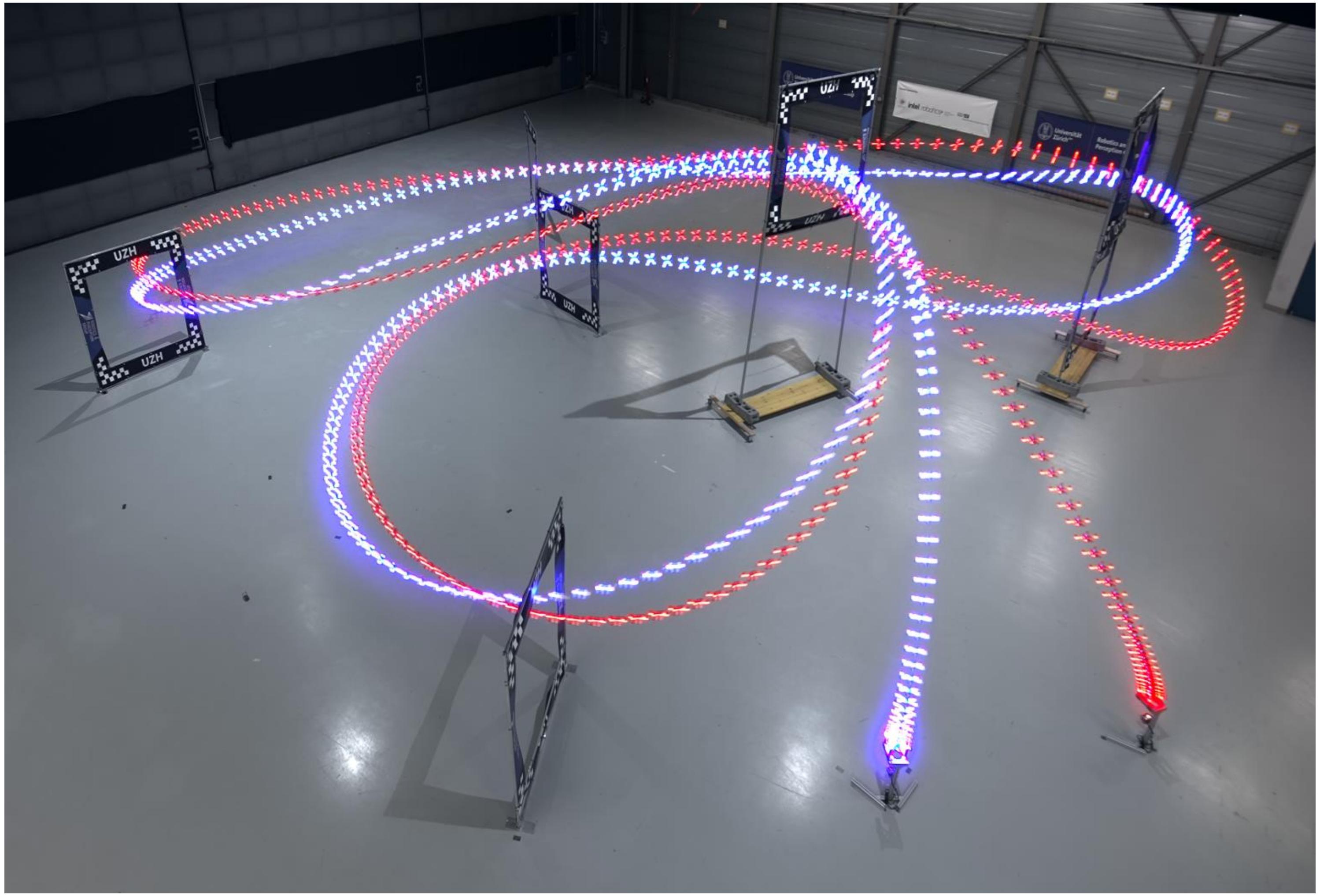
Evaluate on the task of reaching a goal with no prior knowledge about the scene.



Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. [PDF](), [Video](), [Code]()

Drone Racing – A Proxy Task

- pass a sequence of gates in the correct order
- fly a given number of laps in minimum time
- be quicker than the opponent



Our "Swift" Drone

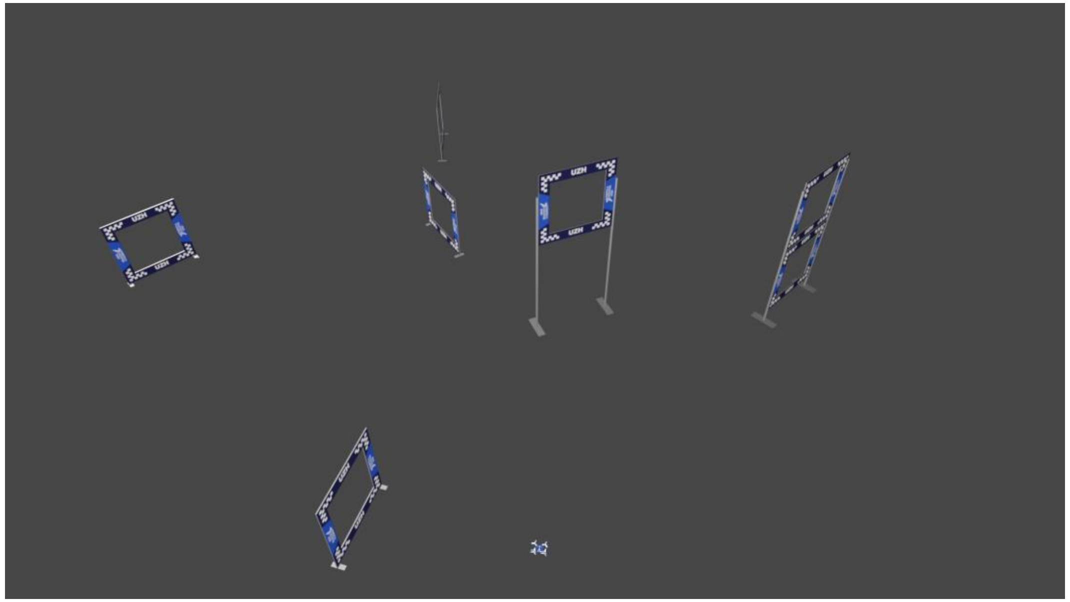
- Jetson TX2
- Realsense T265
  - Images
  - IMU/VIO

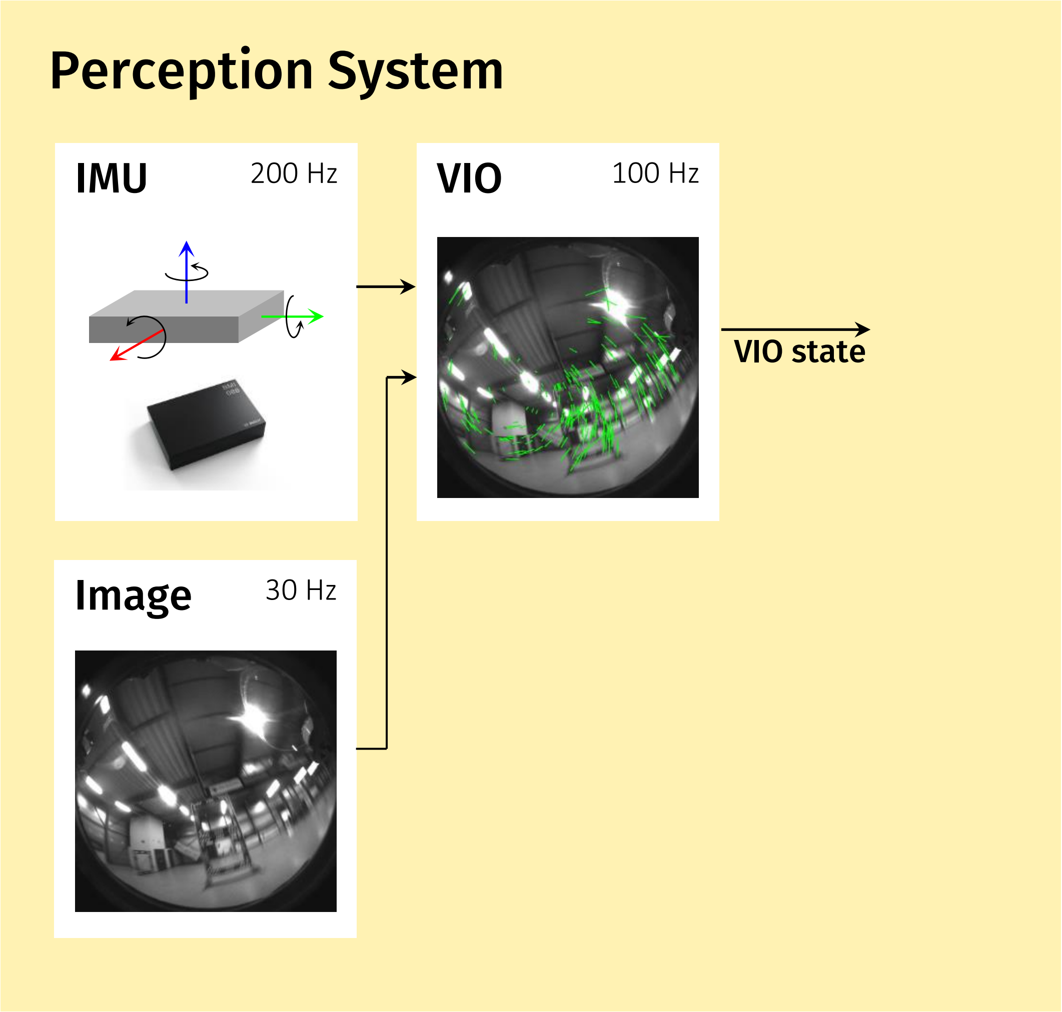
- Weight: 870g
- Thrust: 39N
- TWR: 4.5

Localization

VIO Performance

- VIO drift accumulates over time
- no robust feature tracking
- IMU forward integration

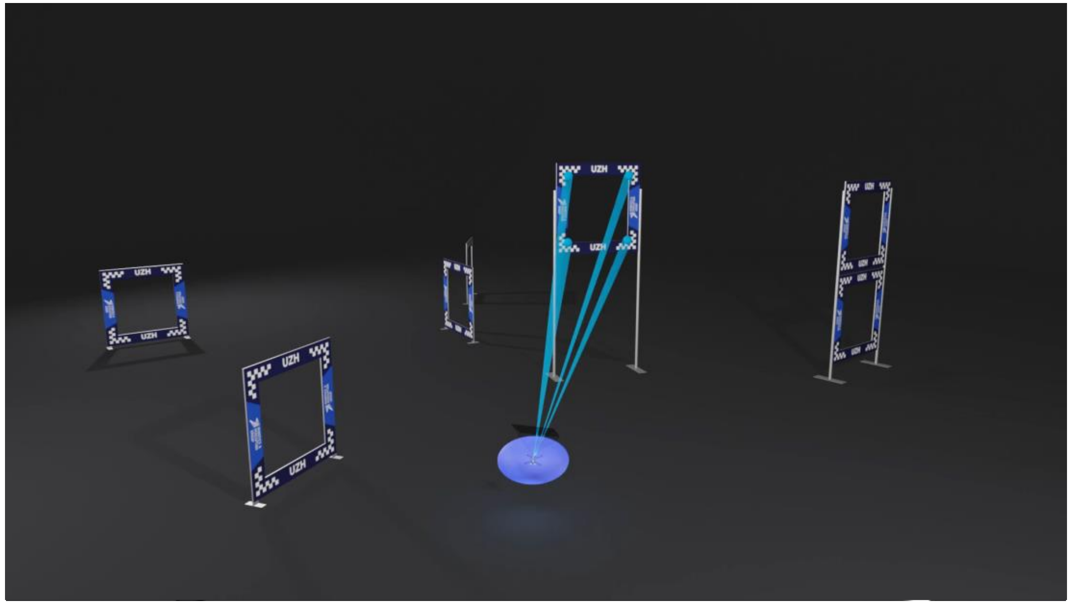


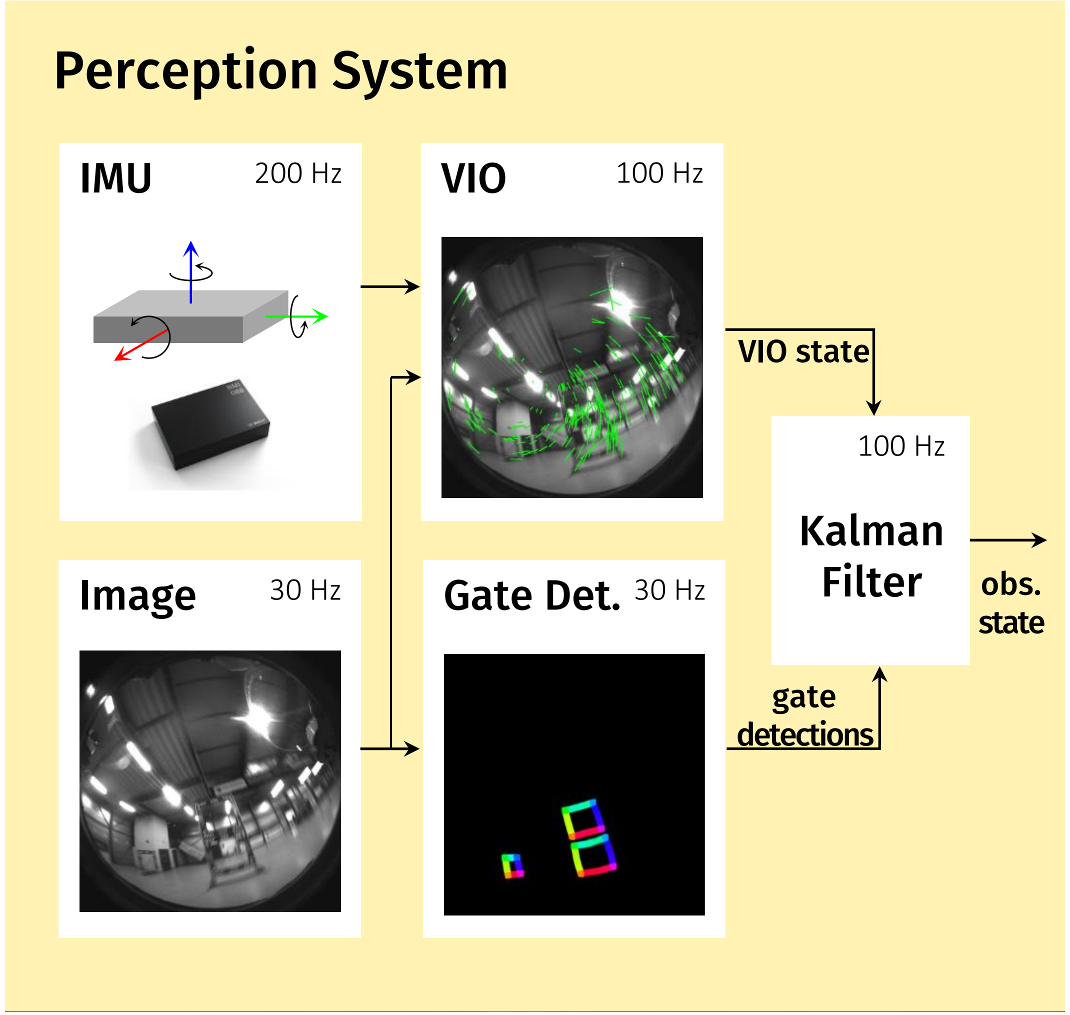


Localization

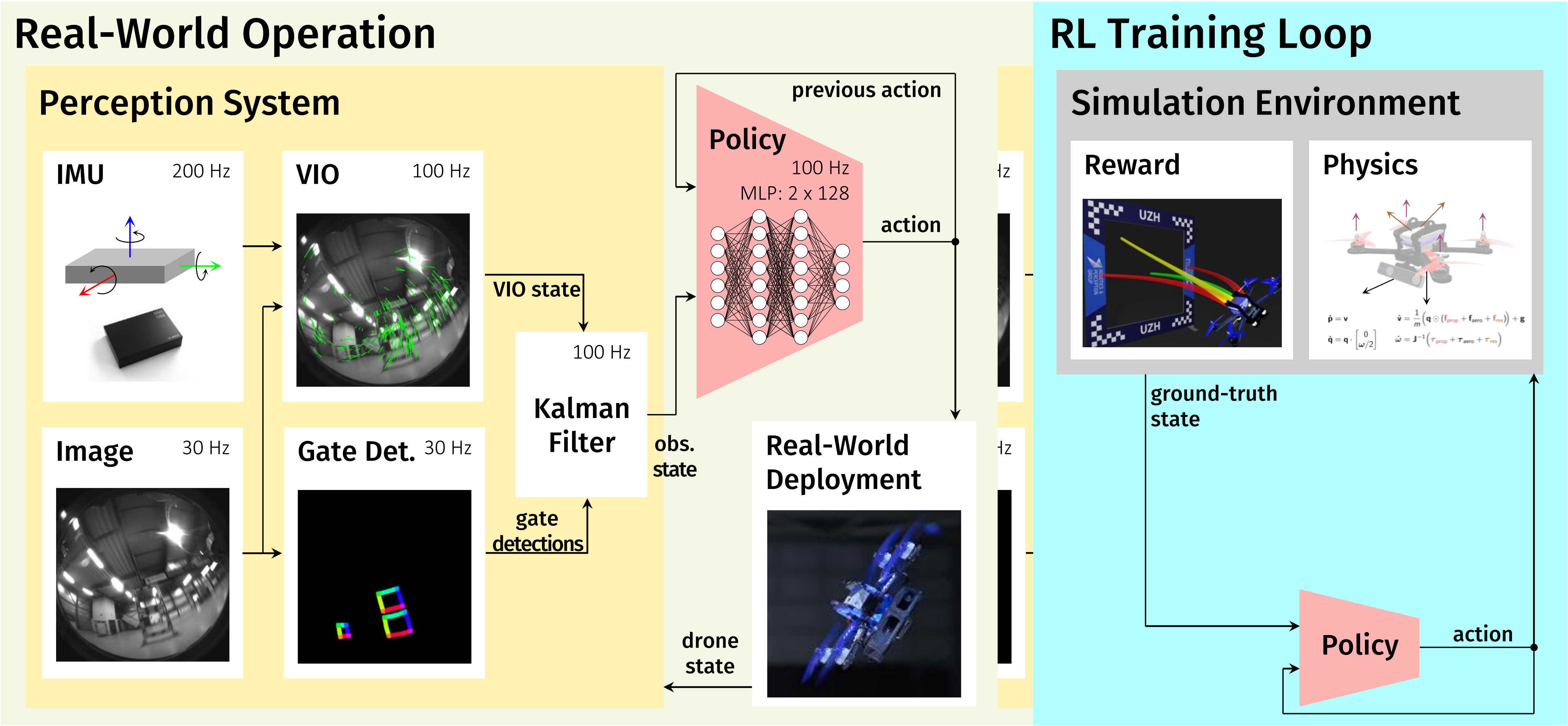
Gate Detections

- CNN-Unet detecting gates
- PnP for localization
- Kalman filter to fuse VIO+Gates





RL Policy Training



RL Policy Training

Reward

- progress reward

$$r_t^{prog} = \lambda_1 \left(d_{t-1}^{Gate} - d_t^{Gate}\right)$$

- perception reward

$$r_t^{perc} = \lambda_2 \exp(\lambda_3\, \delta_{cam}^4)$$

- command reward

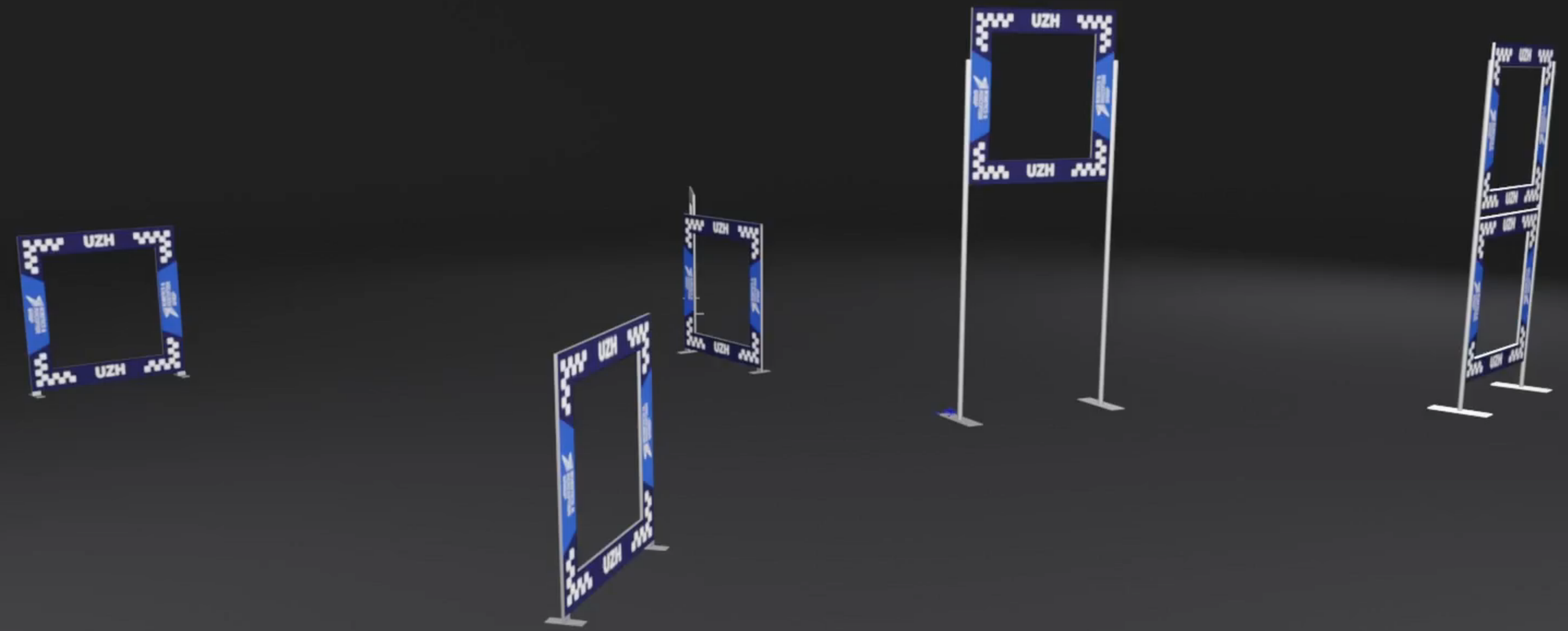
$$r_t^{cmd} = \lambda_4 |a_t^{\omega}| + \lambda_5 |a_t - a_{t-1}|^2$$

- crash penalty

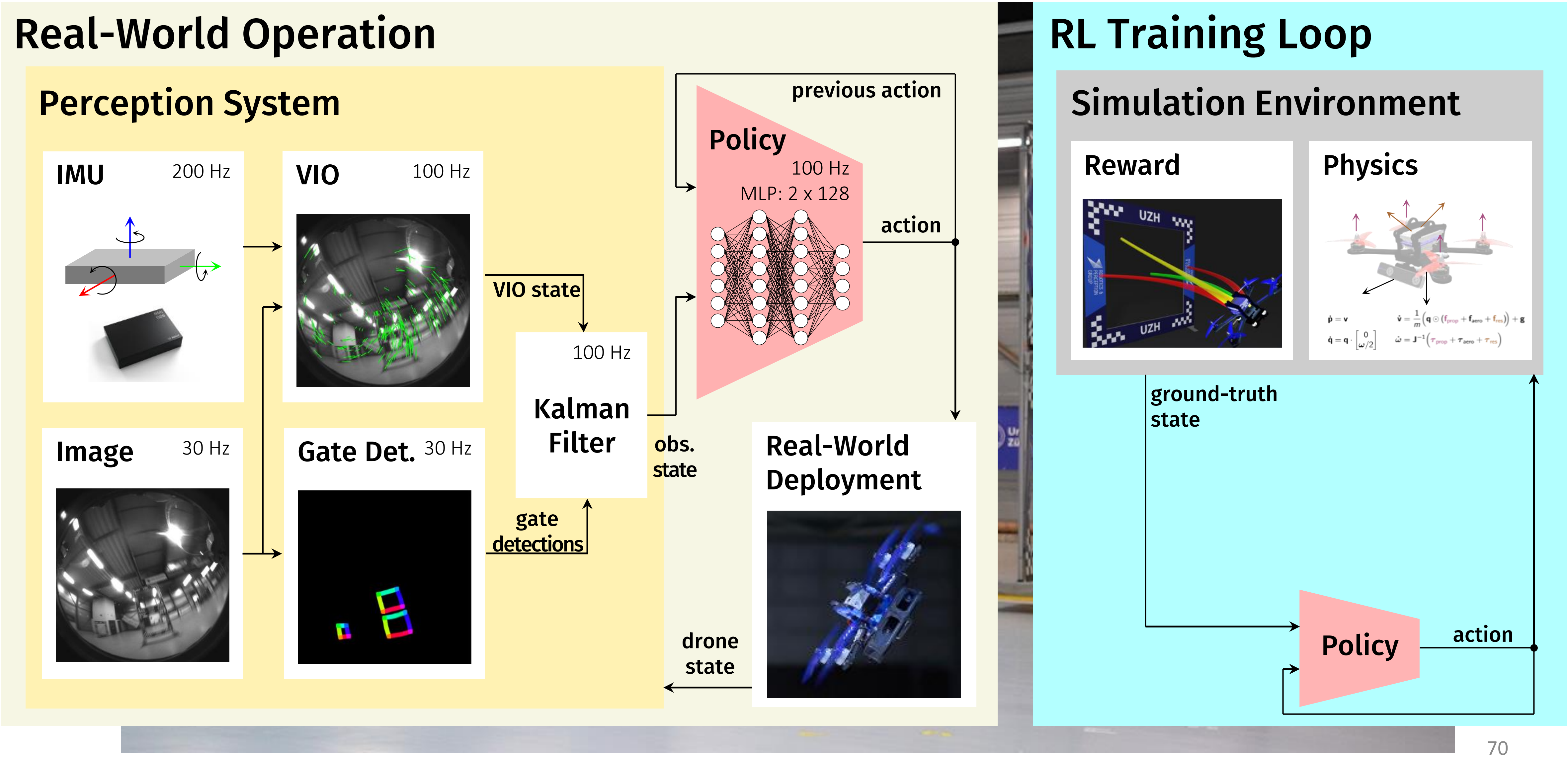
$$r_t^{crash} = \begin{cases} -5.0, & if\ crash \\ 0, & otherwise \end{cases}$$

Training Details

- training with PPO
  - 100M environment interactions
  - 50 min wall-time
  - 23 days sim-time
- value and policy network share architecture
- network:
  - 2 layer MLP
  - 128 nodes per layer
  - activation: LeakyReLU
  - optimizer: Adam


UZH

Is it good enough?

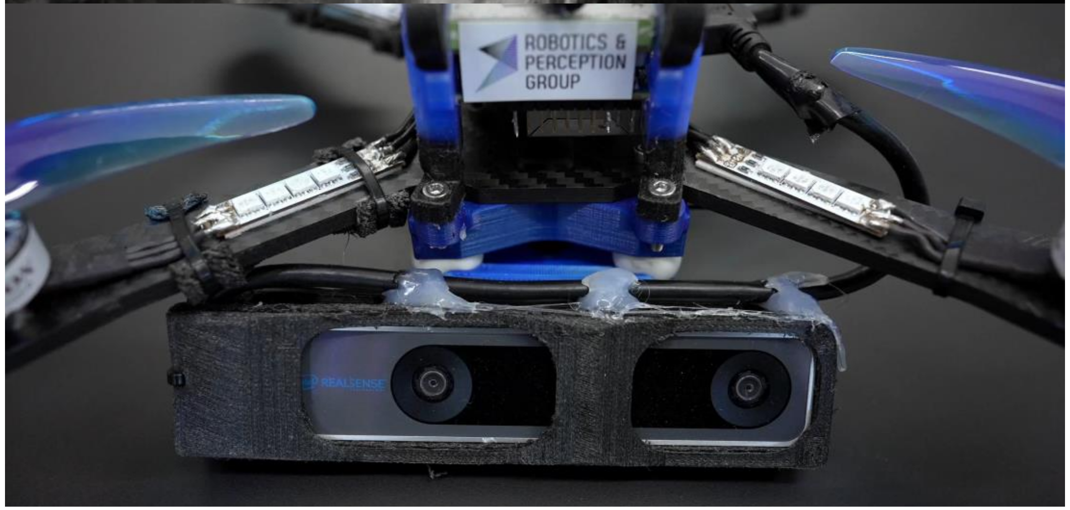


Residual Models

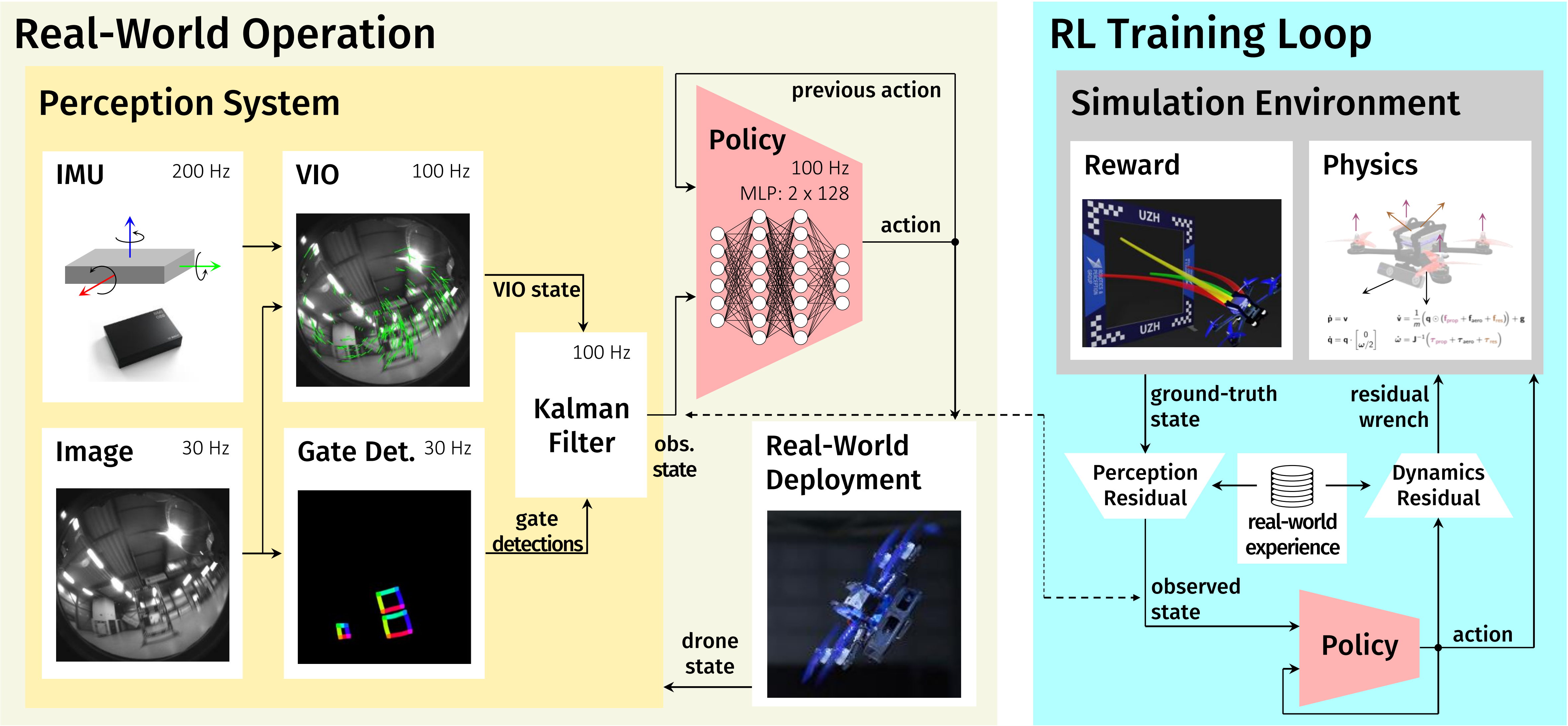
Unmodelled Effects

- Aerodynamic effects
  - turbulence & downwash
  - ground effect
- Mechanical effects
  - soft dampers to shield IMU from motor vibrations
  - camera moves w.r.t. drone body
- Perception effects
  - illumination & background changes

need policy-specific models



The Swift System



Head-to-Head Race Results

	Number of Races	Best time-to-finish	Wins	Losses	Win ratio
A. Vanover vs. Swift	9	17.956	4	5	0.44
T. Bitmatta vs. Swift	7	18.746	3	4	0.43
M. Schäpper vs. Swift	9	21.160	3	6	0.33
Swift vs. human pilots	25	17.465	15	10	0.60

A. Vanover

T. Bitmatta

M. Schäpper

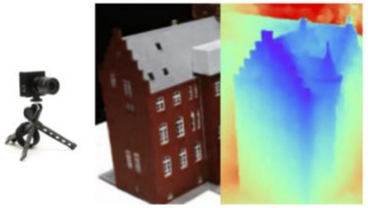
Conclusions and Takeaways

- Autonomous **vision-based agile flight** as a new research topic (at least 10 years to solve it)
  - **Pushes the limit of existing algorithms** in extreme situations
  - Raises **fundamental problems** for robotics **research**

Come over for projects in DL!

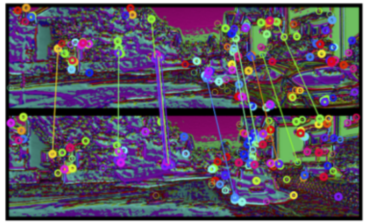
- Visit our webpage for projects! http://rpg.ifi.uzh.ch/student_projects.php

Neural-based scene reconstruction and synthesis using event cameras - Available



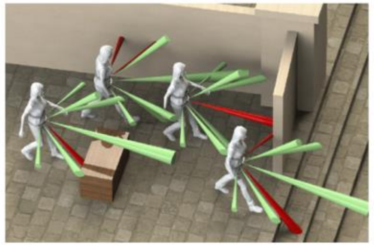
Description: Purely learning-based methods leveraging implicit scene representations have shown impressive results in the reconstruction and synthesis of complex scenes from just a few images, largely surpassing those of traditional methods such as Structure-from-motion, photogrammetry, and image-based rendering. Due to their recent introduction, their advantages over traditional methods are still being explored in the field of computer vision. In particular, their use in conjunction with event-based cameras, bio-inspired sensors with improved latency, temporal resolution, and dynamic range, is still under-explored.

Data-driven Keypoint Extractor for Event Data - Available



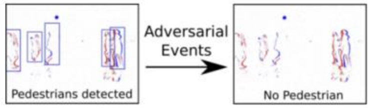
Description: Neuromorphic cameras exhibit several amazing properties such as robustness to HDR scenes, high-temporal resolution, and low power consumption. Thanks to these characteristics, event cameras are applied for camera pose estimation for fast motions in challenging scenes. A common technique for camera pose estimation is the extraction and tracking of keypoints on the camera plane. In the case of event cameras, most existing keypoint extraction methods are handcrafted manually. As a new promising direction, this project tackles the keypoint extraction in a data-driven fashion based on recent advances in frame-based keypoint extractors.

Developing Smart Vision Assistive Technology - Available



Description: More than 200 million people are estimated to have moderate or severe vision impairment in 2020. Their lack of autonomy limits the completion of many daily living activities. In this project, we will focus on applying robotics techniques, such as state estimation and path planning, to help visually impaired people to navigate unknown and unstructured environments. Image credits: Katzschmann et al. 2018.

Adversarial Robustness in Event-Based Neural Networks - Available



Description: The robustness and reliability of neural networks are of utmost importance in several computer vision applications, especially in automotive applications where real-time predictions are crucial for safe and efficient operation. In this context, event-based cameras, due to their unique property of capturing changes in the scene, have shown impressive performance in low-latency prediction tasks such as object detection, tracking, and optical flow prediction. However, in order to be widely adopted in the real world, the robustness and reliability of such event-based networks have to be properly studied and verified. Until now, however, these aspects have been overlooked in the event-based literature. We look for students with strong programming (Pyhton/Matlab) and computer vision background. Additionally, knowledge in machine learning frameworks (pytorch, tensorflow) is required.

Check out our student projects!

- Visit our webpage: https://rpg.ifi.uzh.ch/student_projects.php

Learning Rapid UAV Exploration with Foundation Models - Available

Description: In this project, our objective is to efficiently explore unknown indoor environments using UAVs. Recent research has demonstrated significant success in integrating foundational models with robotic systems. Leveraging these foundational models, the drone will employ learned semantic relationships from large-world-scale data to actively explore and navigate through unknown environments. While most prior research has focused on ground-based robots, this project aims to investigate the potential of integrating foundational models with aerial robots to introduce more agility and flexibility. Applicants should have a solid understanding of mobile robot navigation, machine learning experience (PyTorch), and programming experience in C++ and Python.

Bayesian Optimization for Racing Aerial Vehicle MPC Tuning - Available

Description: In recent years, model predictive control, one of the most popular methods for controlling constrained systems, has benefitted from the advancements of learning methods. Many applications showed the potential of the cross fertilization between the two fields, i.e., autonomous drone racing, autonomous car racing, etc. Most of the research efforts have been dedicated to learn and improve the model dynamics, however, the controller tuning, which has a crucial importance, have not been studied much.

Bayesian Optimization for Racing Aerial Vehicle MPC Tuning - Available

Description: In recent years, model predictive control, one of the most popular methods for controlling constrained systems, has benefitted from the advancements of learning methods. Many applications showed the potential of the cross fertilization between the two fields, i.e., autonomous drone racing, autonomous car racing, etc. Most of the research efforts have been dedicated to learn and improve the model dynamics, however, the controller tuning, which has a crucial importance, have not been studied much.

Efficient Learning-aided Visual Inertial Odometry - Available

Description: Recent works have shown that deep learning (DL) techniques are beneficial for visual inertial odometry (VIO). Different ways to include DL in VIO have been proposed: end-to-end learning from images to poses, replacing one/more block/-s of a standard VIO pipeline with learning-based solutions, and include learning in a model-based VIO block. The project will start with a study of the current literature on learning-based VIO/SLAM algorithms and an evaluation of how/where/when DL is beneficial for VIO/SLAM. We will use the results of this evaluation to enhance a current state-of-the-art VIO pipeline with DL, focusing our attention on algorithm efficiency at inference time. The developed learning-aided VIO pipeline will be compared to existing state-of-the-art model-based algorithms, with focus on robustness, and deployed on embedded platforms (Nvidia Jetson TX2 or Xavier).