University of Zurich[UZH]

ROBOTICS & PERCEPTION GROUP

Vision Algorithms for Mobile Robotics

Lecture 12c
Deep Learning Tutorial

Daniel Gehrig and Elia Kaufmann

http://rpg.ifi.uzh.ch

Outline

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

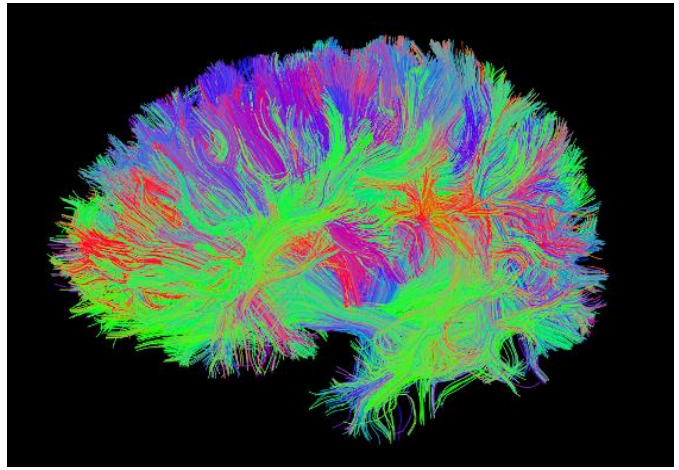
Relevant for the exam

Outline

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

The Deep Learning Revolution

Medicine

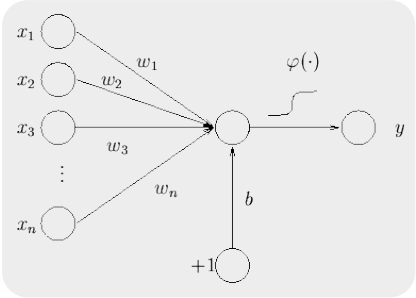


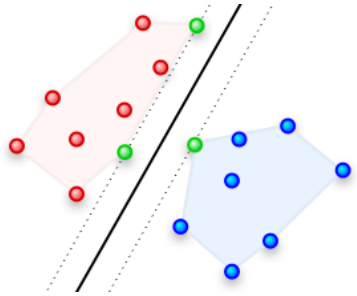
Media & Entertainment

Surveillance & Security

Autonomous Driving

Some History





Perceptron

1969

Back-Propagation

1974

AI Winter

SVM

1998

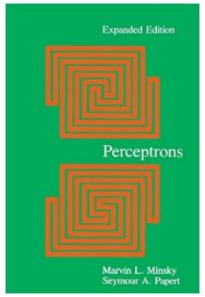
2006

AlexNet

2012

1958

Critics



1995

Convolutional Neural Networors for Handwritten Digits Recognition

Restricted Boltzman Machines

What changed?

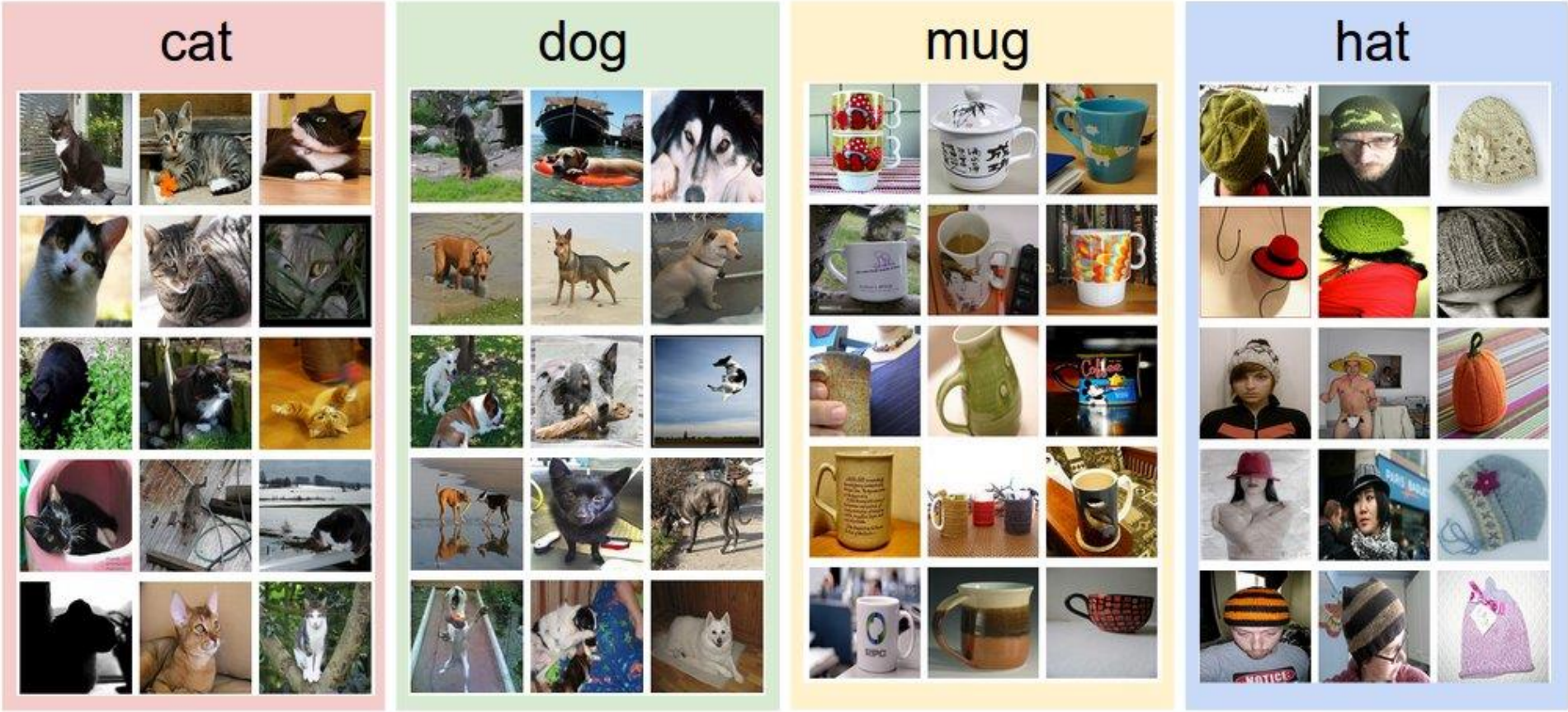
1. Hardware Improvements

2. Big Data Available

3. Algorithmic Progress

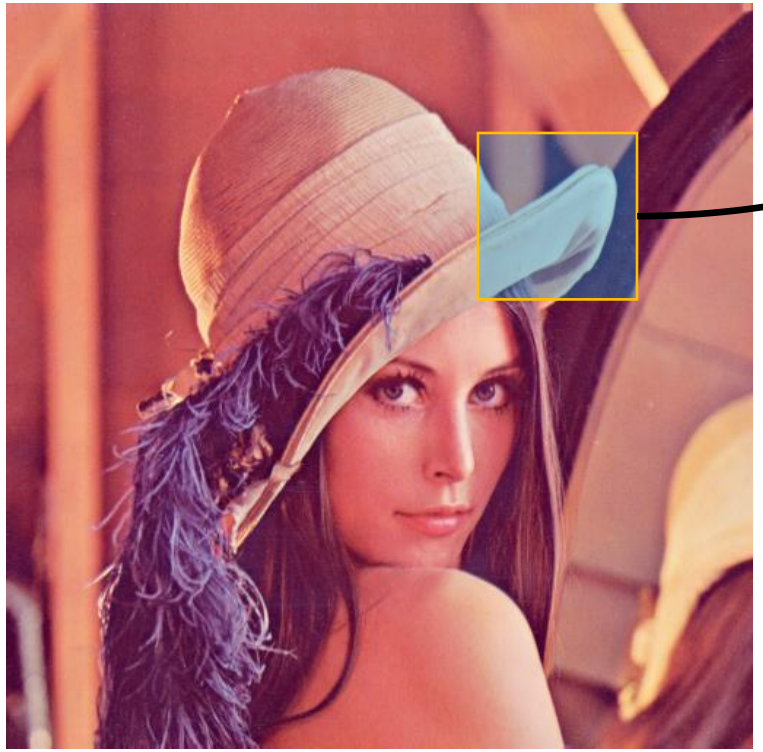
Image Classification

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



The semantic gap

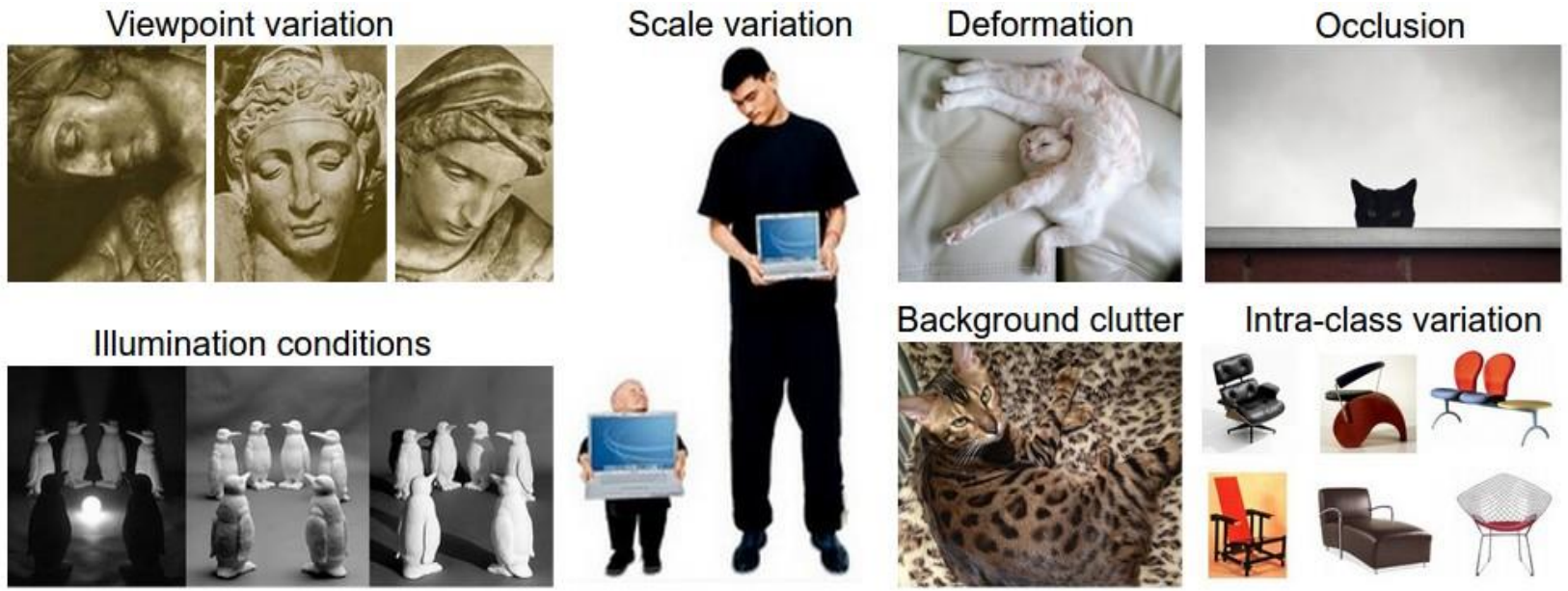
What computers see compared to what we see



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

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

Ground truth, y_i

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

Function parameters or weights

Update

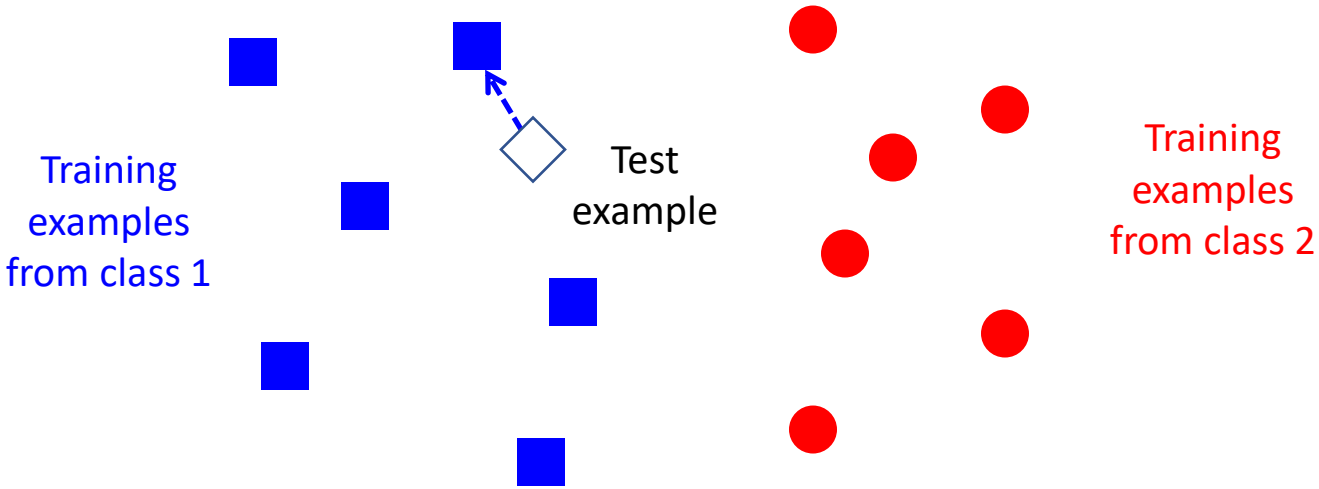
$$Loss(f(x_i, \theta), y_i)$$

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 $\boldsymbol{x}$

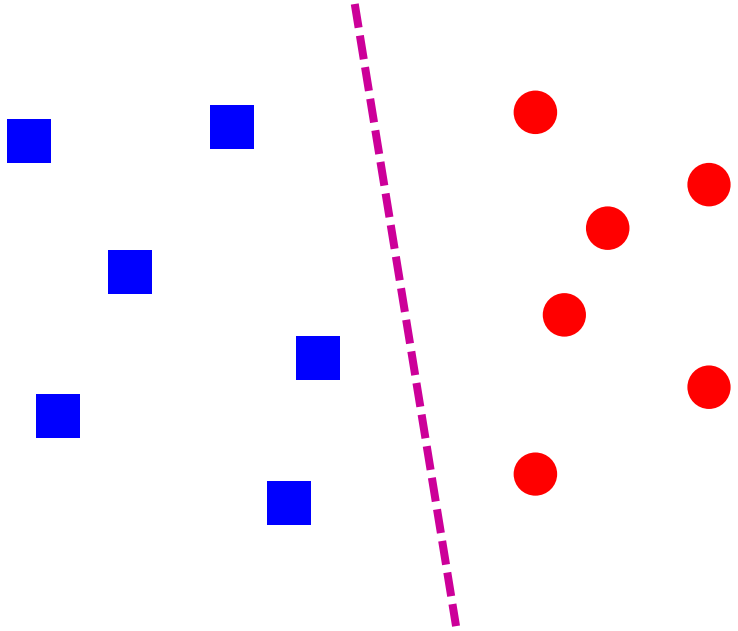
How fast is training? How fast is testing?

- 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:



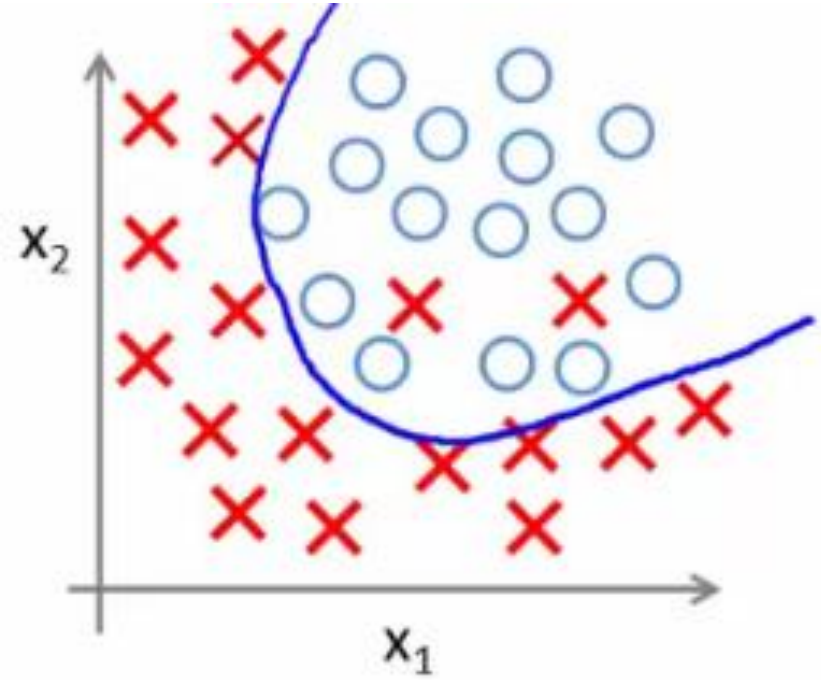
$$f(\boldsymbol{x}, \theta) = sgn(\theta \cdot \boldsymbol{x} + b)$$

What is θ? What is the dimensionality of images?

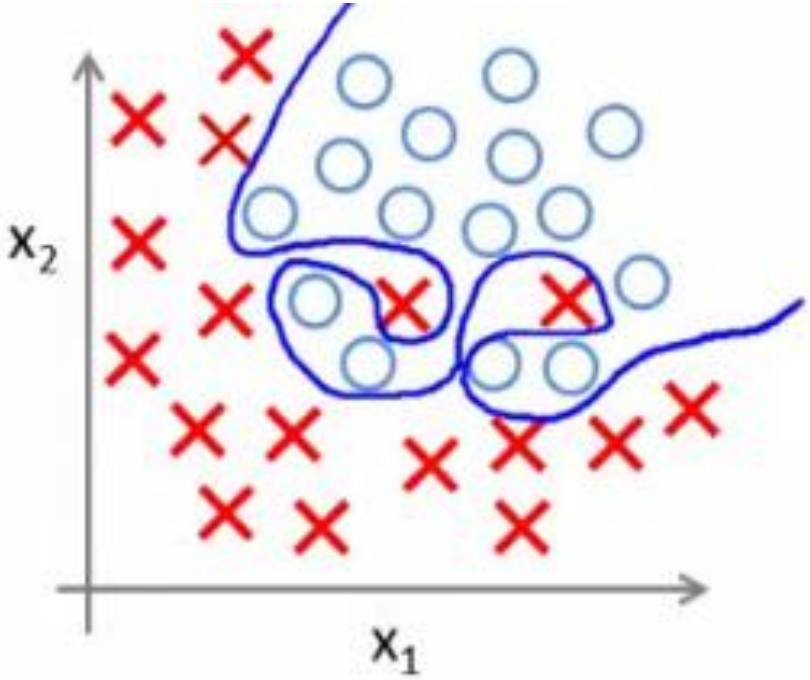
Classifiers: non-linear

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

Good classifier

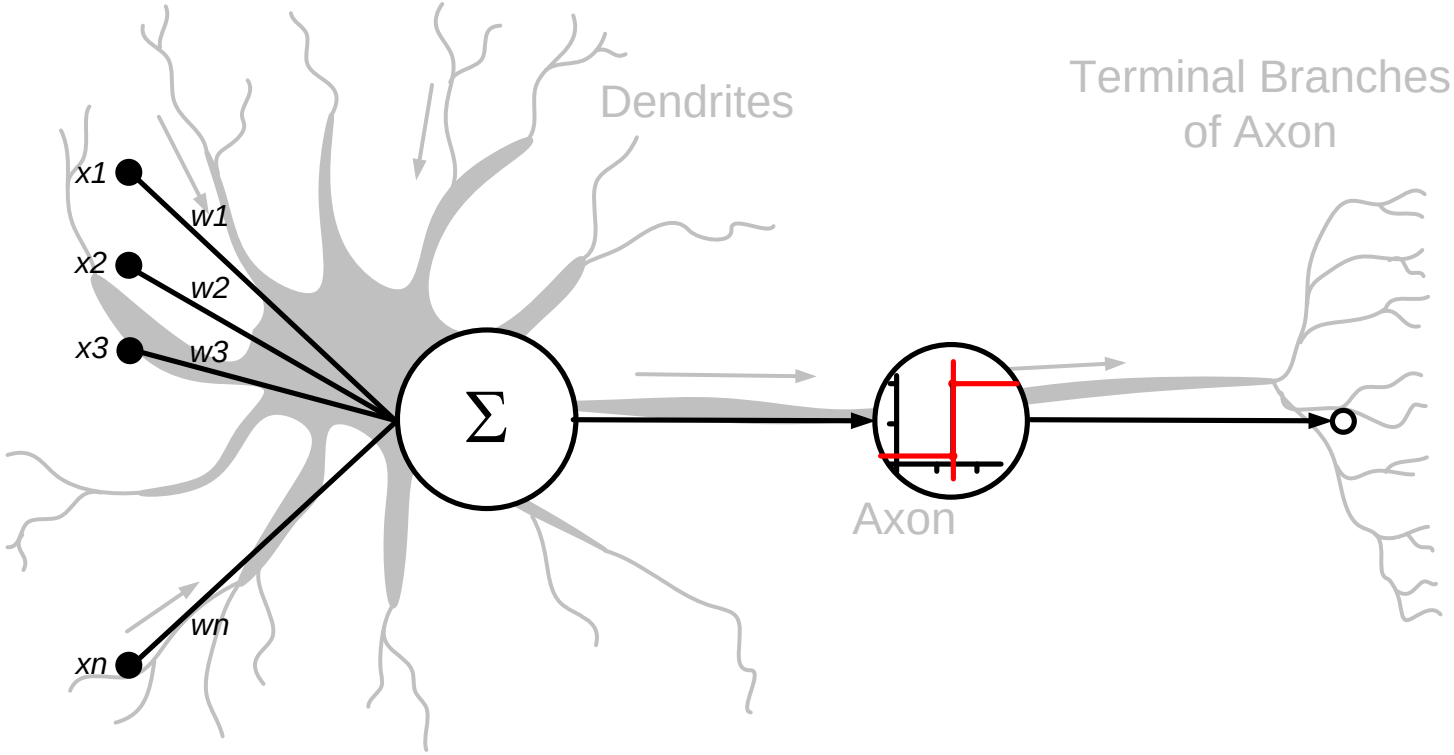


Bad classifier (over fitting)



Biological Inspiration

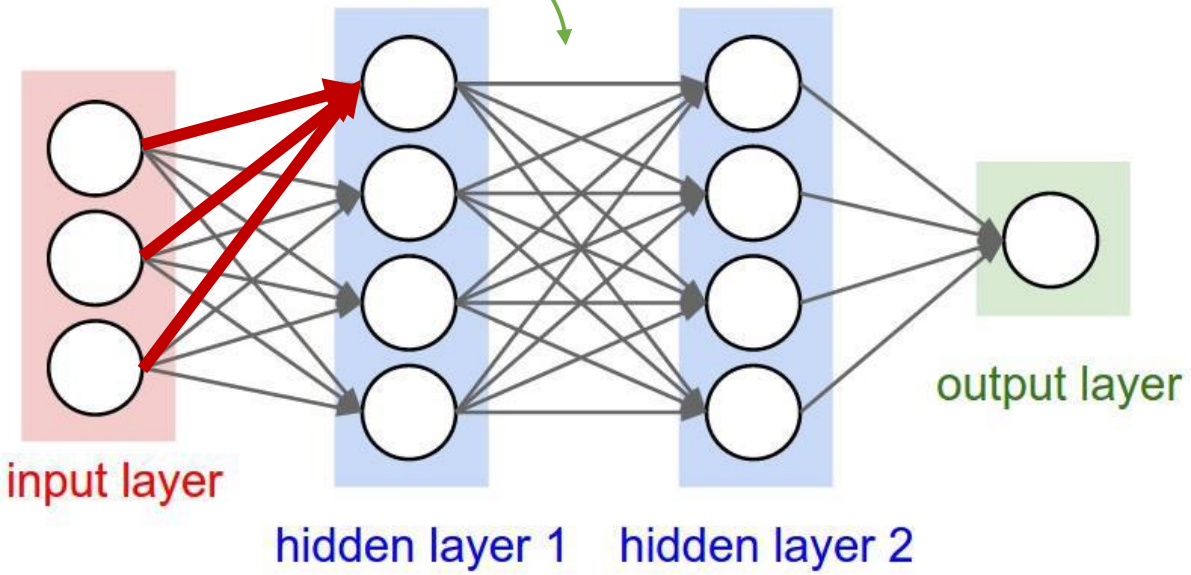
$f(x, \theta) = F(\theta x)$, F is a non-linear activation function (Step, ReLU, Sigmoid)



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

Multi Layer Perceptron

$f(x, \theta)$

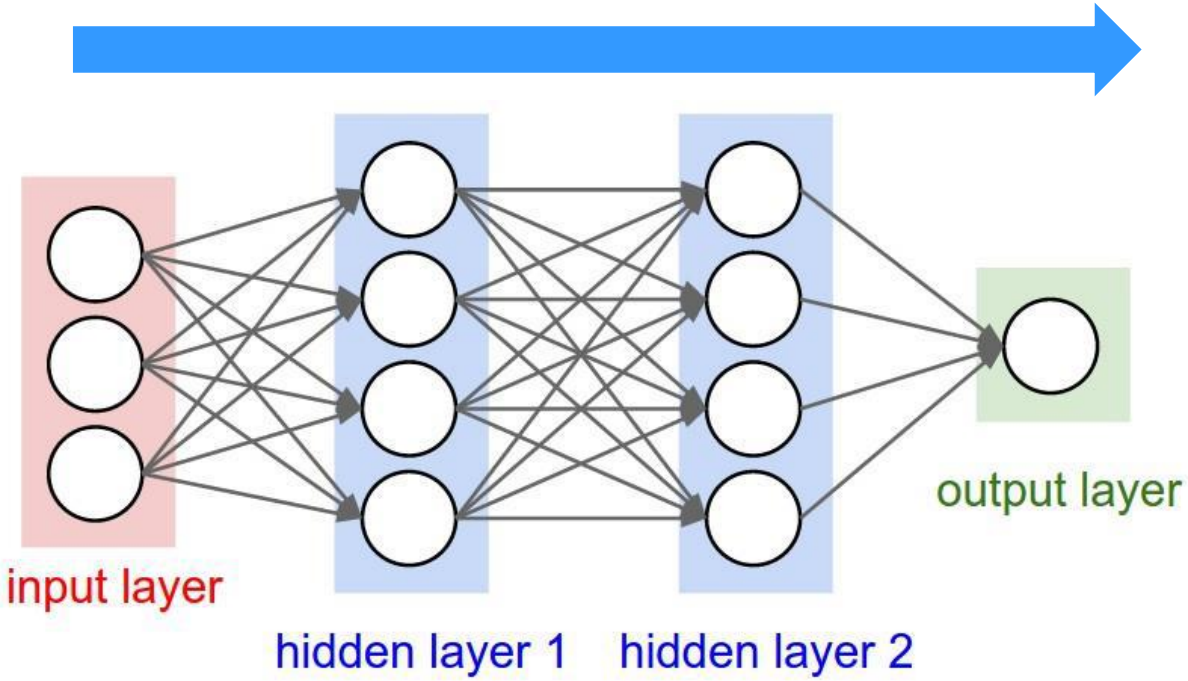


Non-linear Activation functions (ReLU, sigmoid, etc.)

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

Forward Propagation

Forward Pass



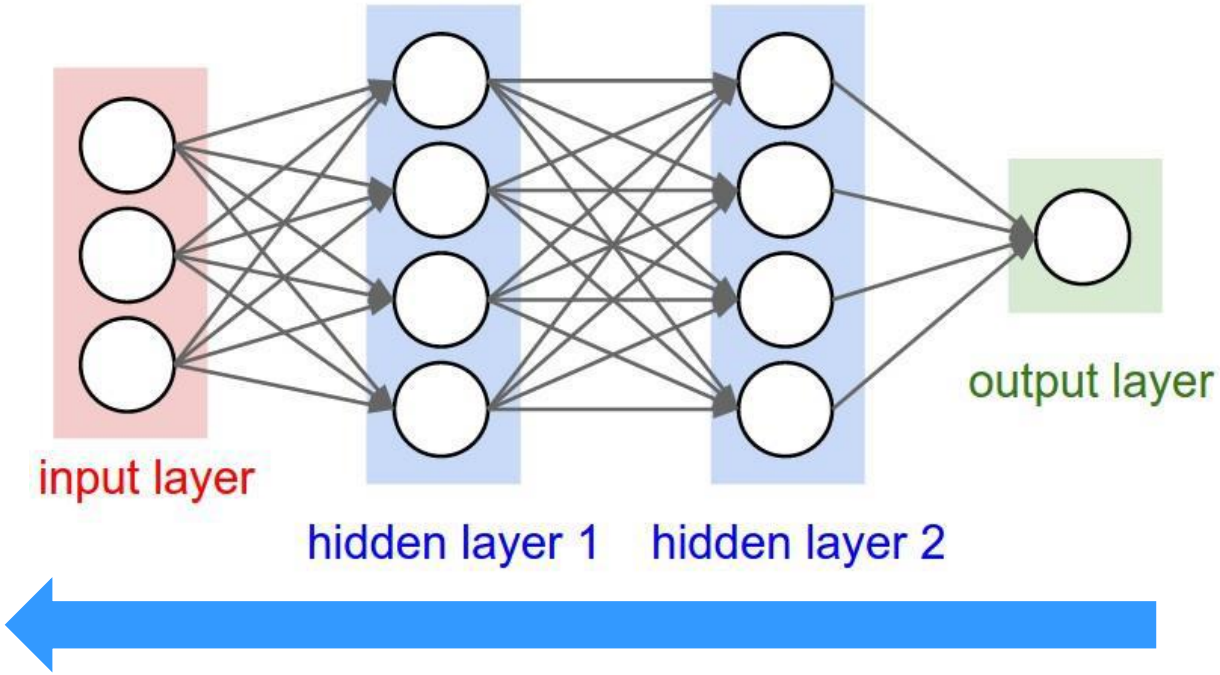
$$Loss(f(x_i, \theta), y_i)$$

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

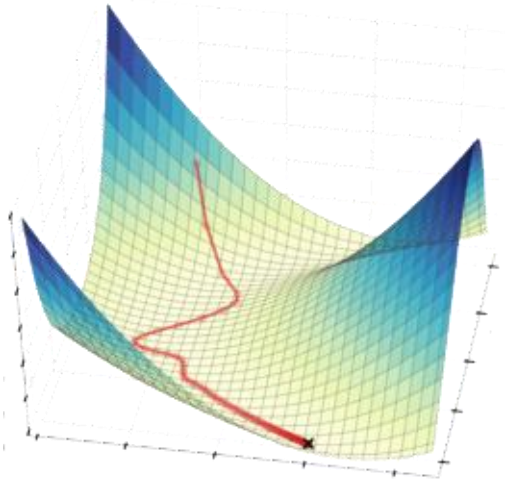
Optimization: Back-propagation

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

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



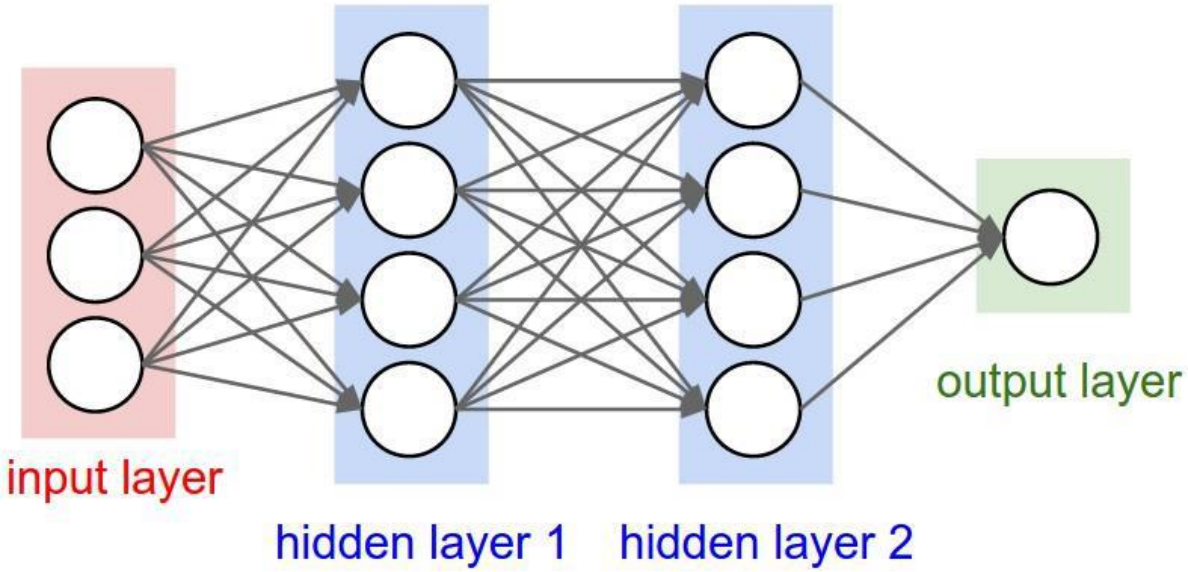
$$Loss(f(x_i, \theta), y_i)$$



[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*

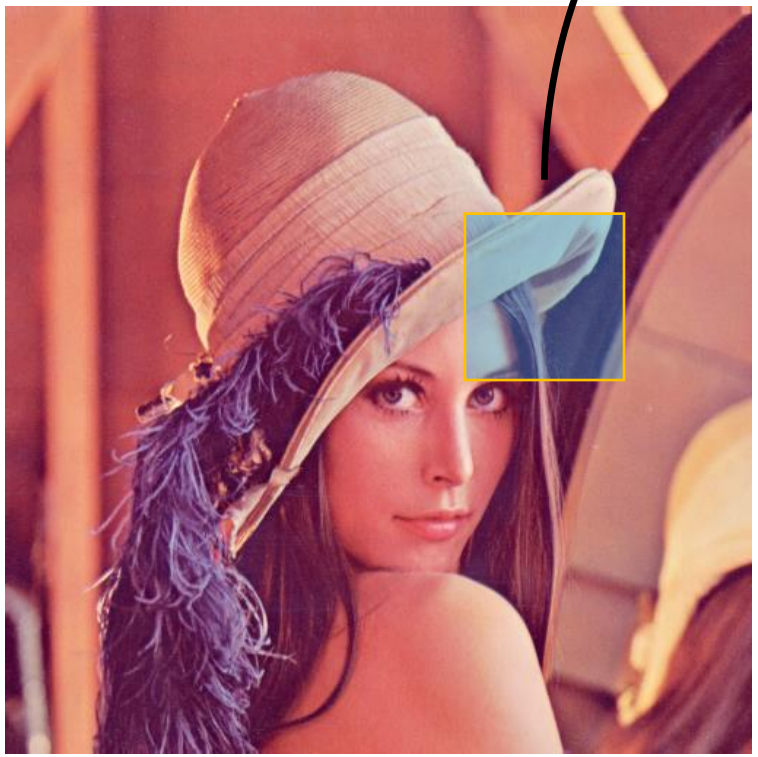
Problems of fully connected network

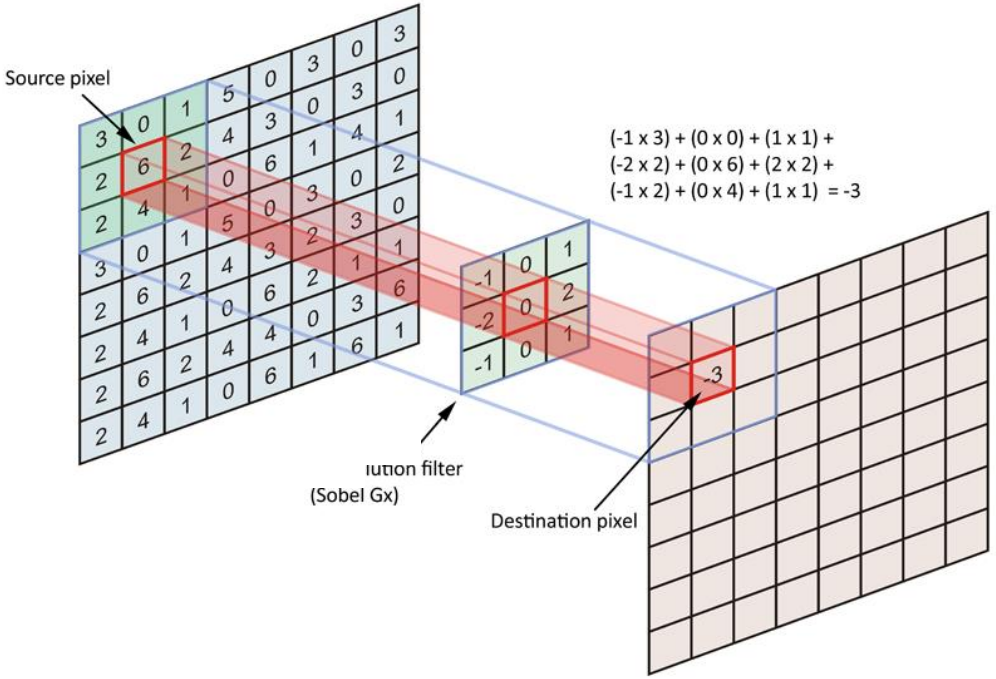
Too many parameters → **possible overfitting**.



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

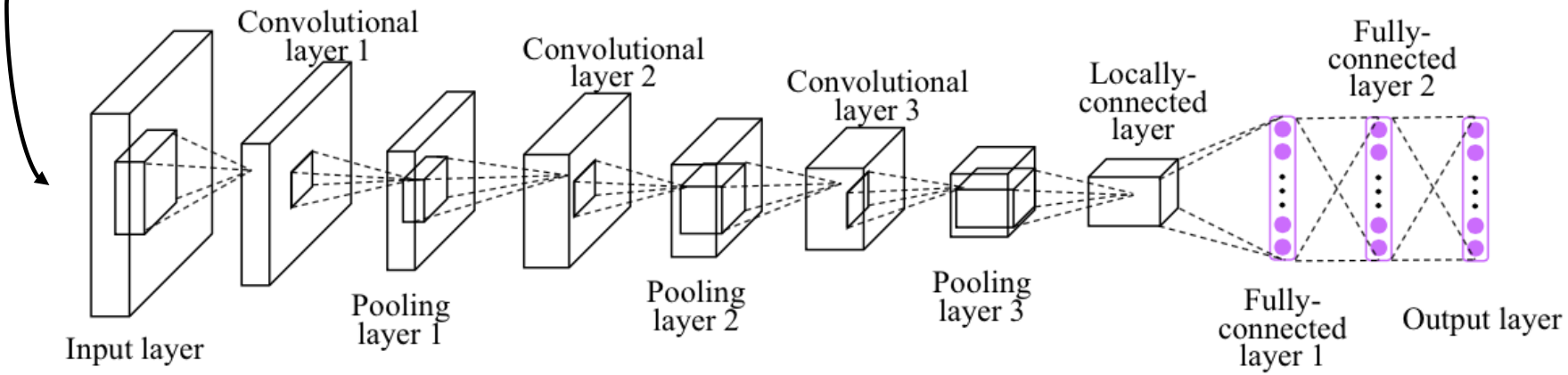
Convolutional Neural Networks

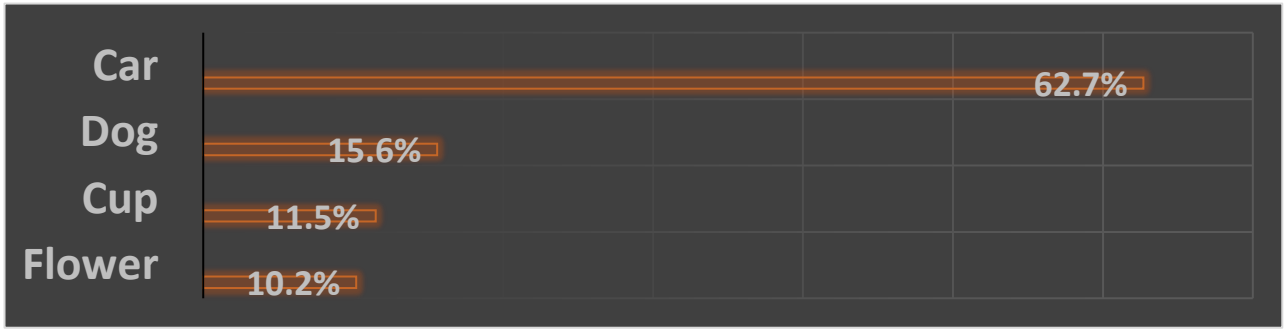




LeCun, Bottou, Bengio, Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 1998. PDF

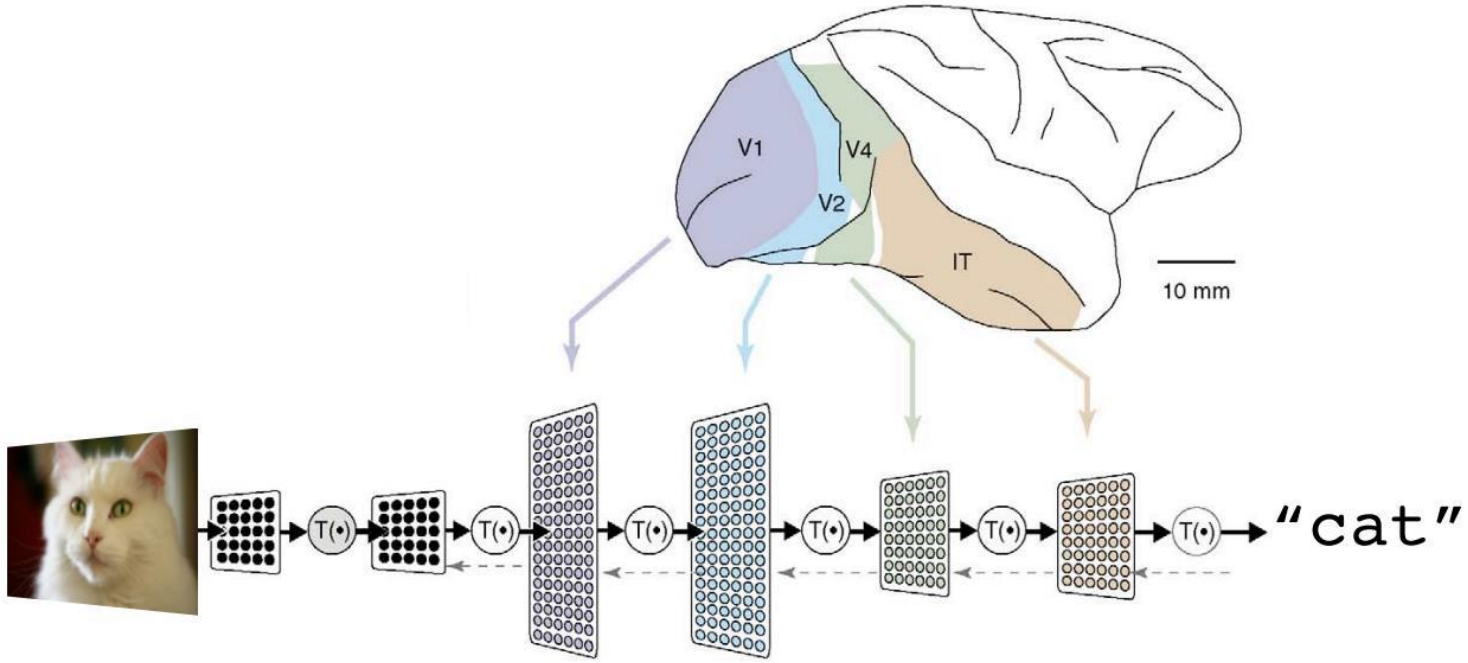
Going Deep


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
Dog
Cup
Flower
62.7%
15.6%
11.5%
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**

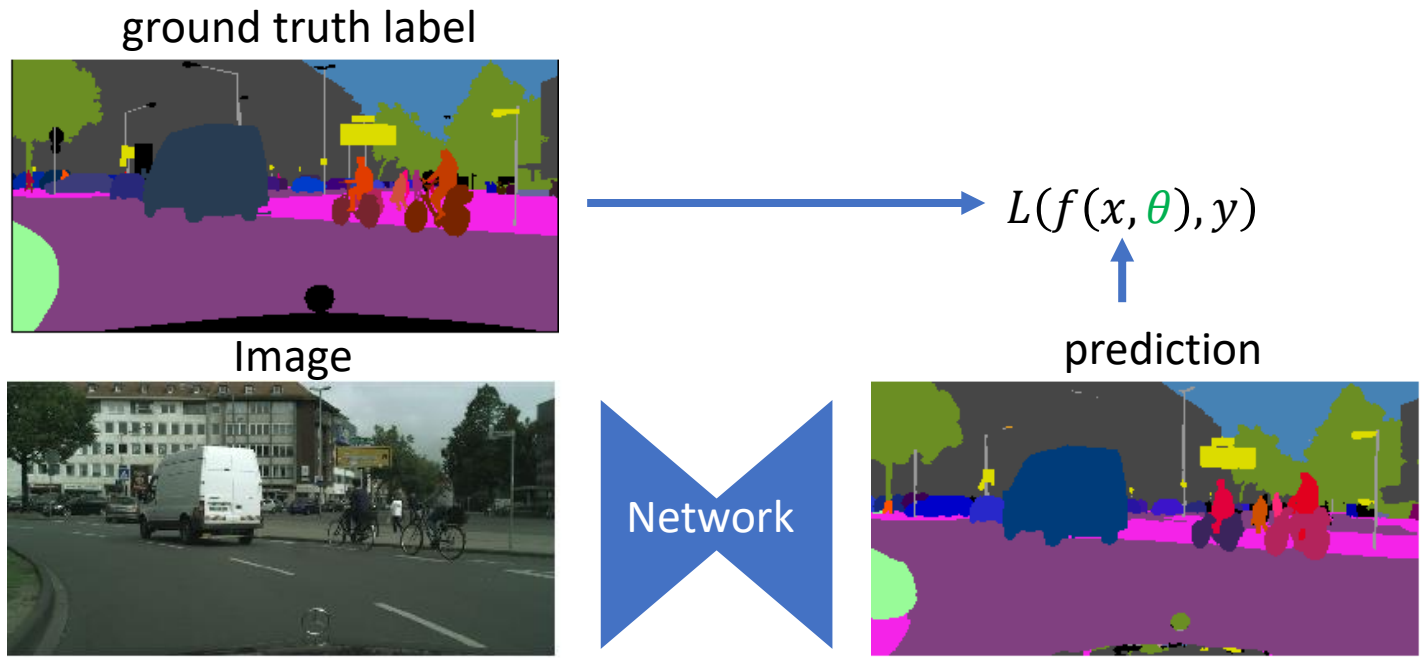


Outline

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

Supervised Learning

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



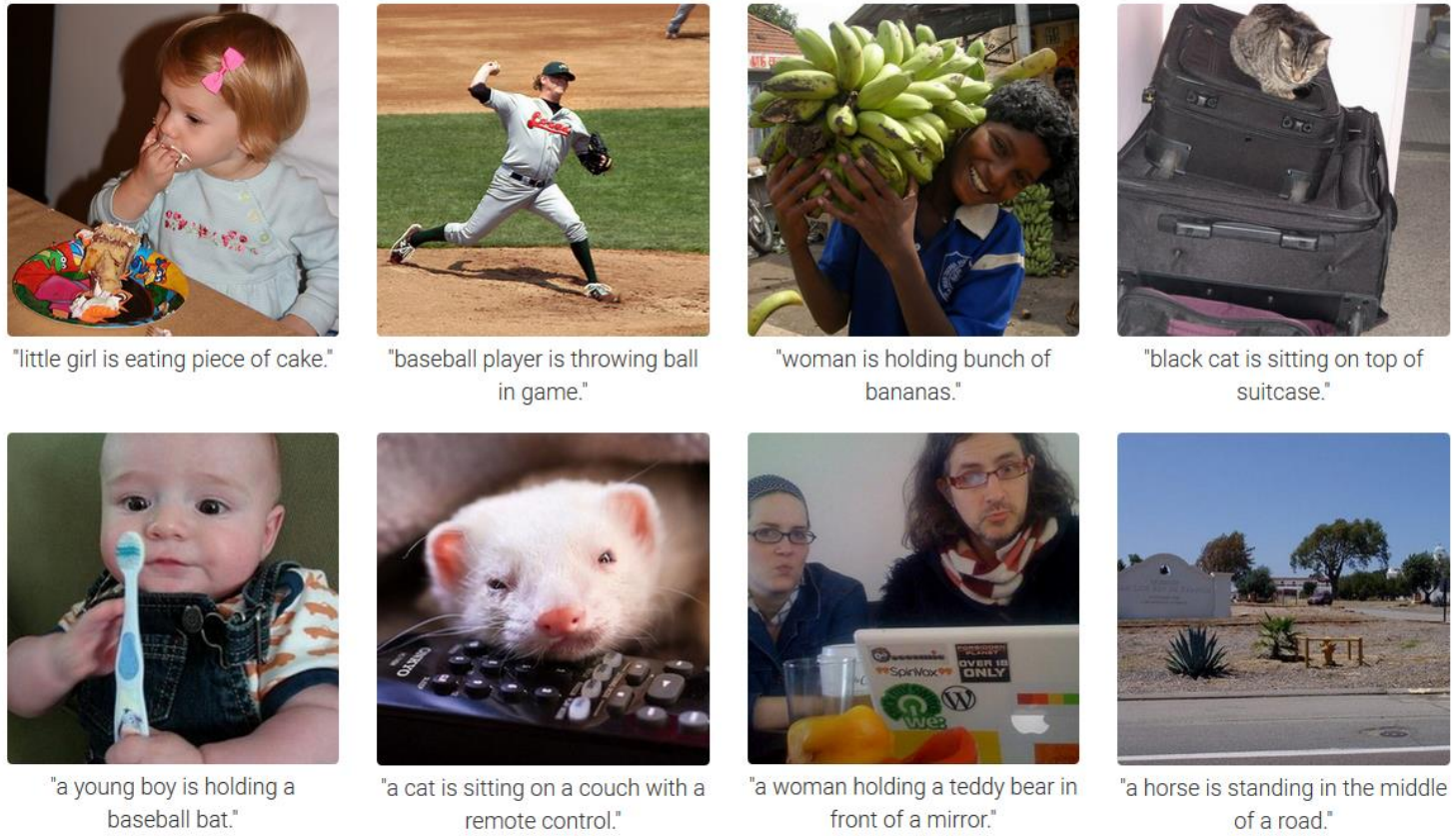
Supervised Learning

- Image Segmentation

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

Supervised Learning

- Image Captioning



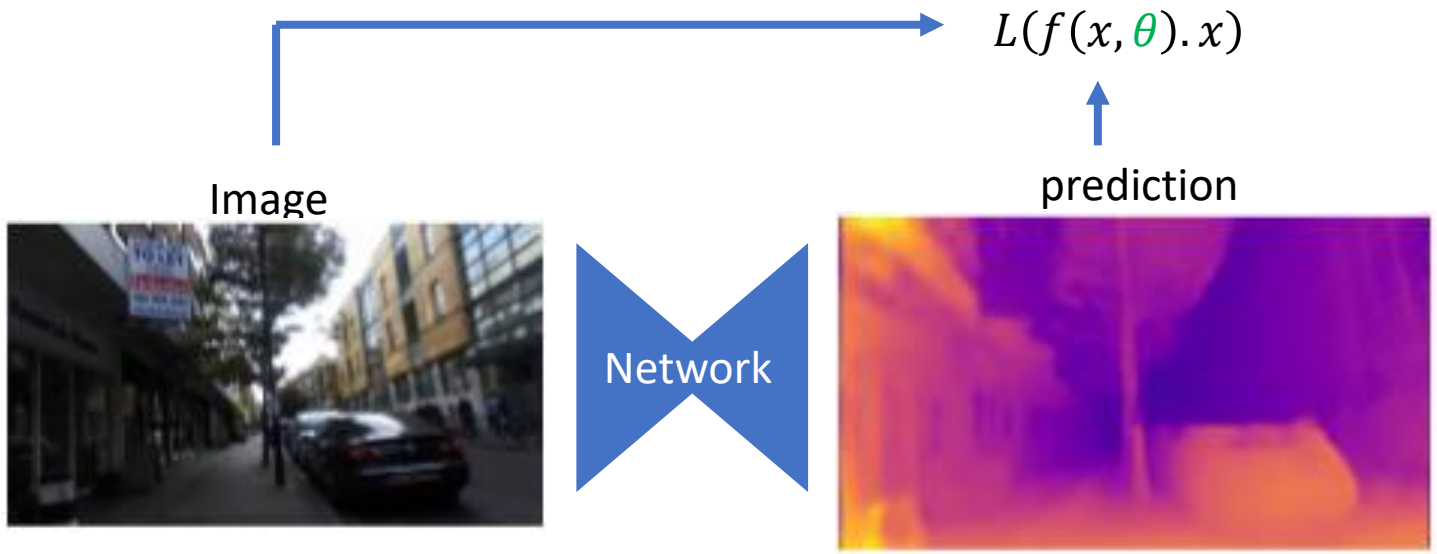
Karpathy, Fei-Fei, *Deep Visual-Semantic Alignments for Generating Image Descriptions,* Conference of Computer Vision and Pattern Recognition (CVPR), 2015. PDF

Outline

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

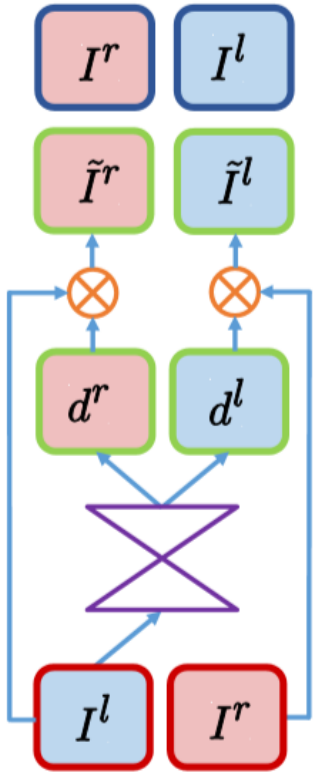
Unsupervised Learning

- 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

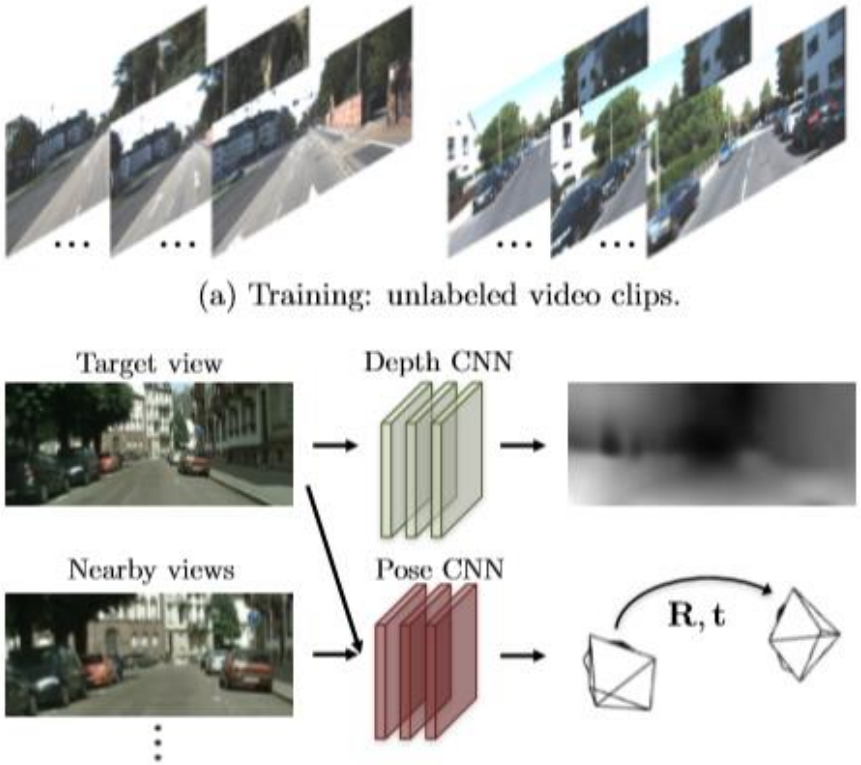
- Monocular Depth Estimation



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



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

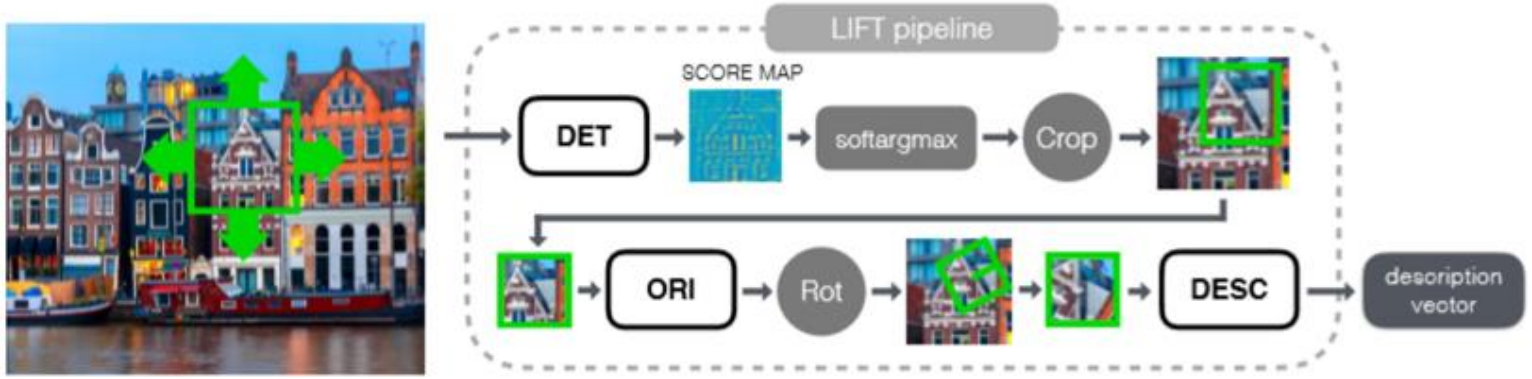
- Introduction
- Supervised Learning
- Unsupervised Learning
- Applications to Computer Vision
- Conclusions
- Machine Learning for Drones

Keypoint Detection and Description

- Deep Descriptors: Learned Invariant Feature Transform (LIFT)

LIFT consists of 3 neural networks:

- A keypoint detector
- An orientation detector
- A descriptor generator

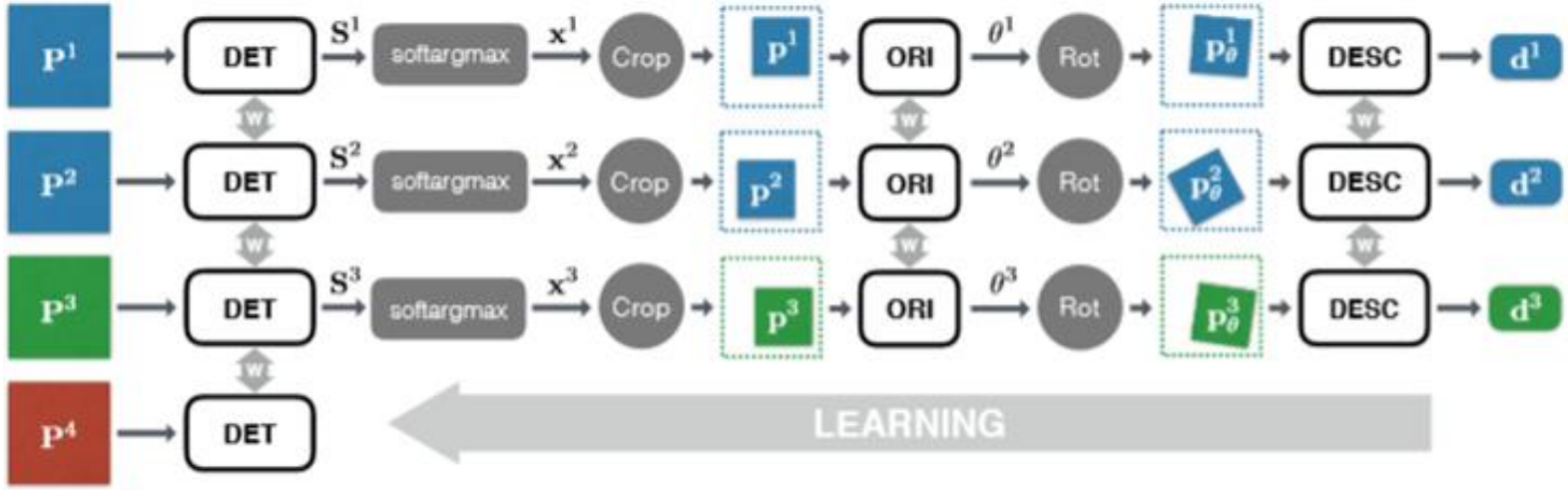


[1] Yi, Trulls, Lepetit, Fua, *LIFT: Learned Invariant Feature Transform*, European Conference of Computer Vision (ECCV), 2016. PDF

LIFT Loss

The LIFT loss has 3 components:

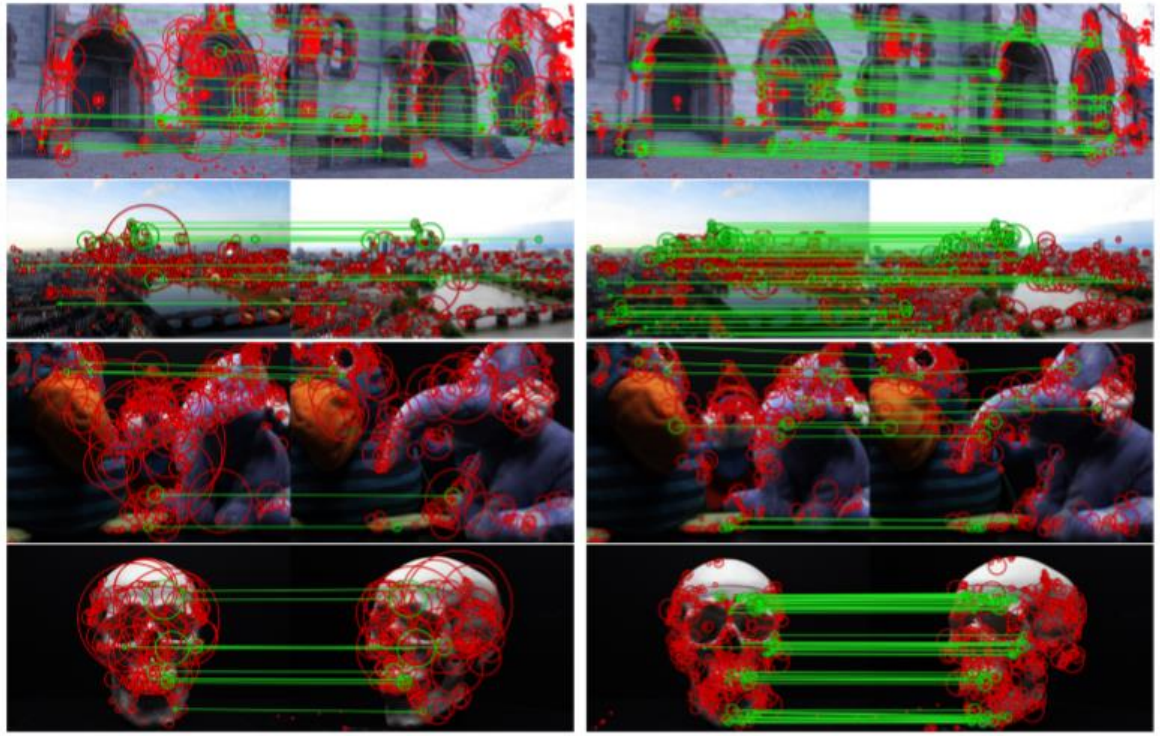
- Descriptors of **matching patches**, d^1, d^2, should be **close**
- Descriptors of **non-matching patches**, d^1, d^2, should be **far apart**
- Keypoints should not be in homogeneous regions: P4 should not be detected as a keypoint.



[1] Yi, Trulls, Lepetit, Fua, *LIFT: Learned Invariant Feature Transform*, European Conference of Computer Vision (ECCV), 2016. PDF

LIFT - Results

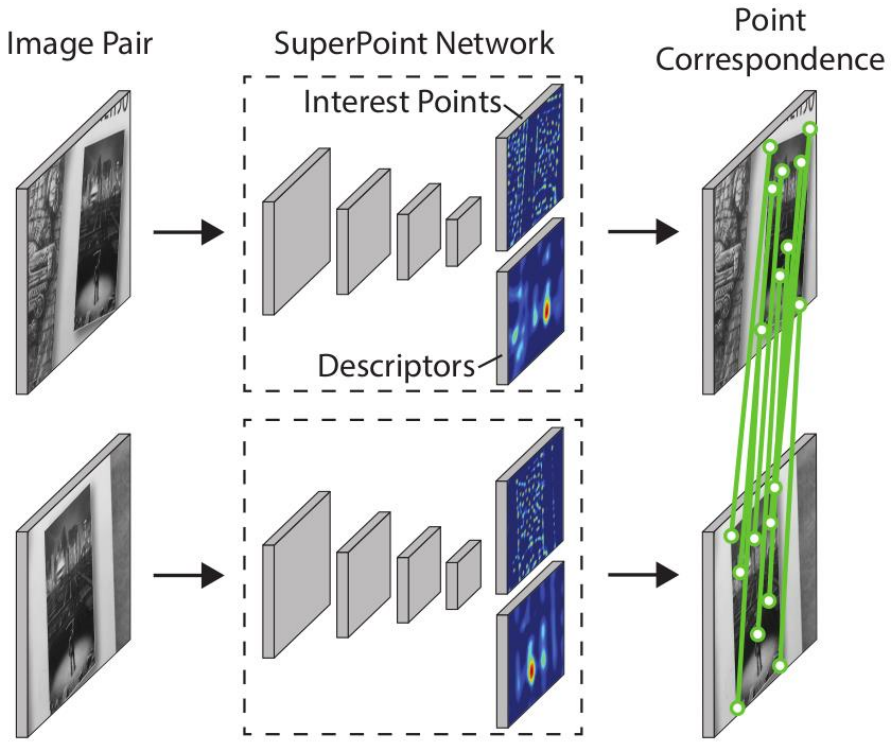
- Works better than SIFT! (well, in some datasets)



[1] Yi, Trulls, Lepetit, Fua, *LIFT: Learned Invariant Feature Transform*, European Conference of Computer Vision (ECCV), 2016. PDF

Keypoint Detection and Description

- SuperPoint: Self-Supervised Interest Point Detection and Description
  - SIFT and friends are complicated heuristic algorithms
  - Still, SIFT is a hard to beat baseline for new methods
  - Can we do better with a data driven approach?
  - SuperPoint uses a CNN to **predict keypoints and descriptors simultaneously**.
  - Detector less accurate than SIFT, but descriptor shown to outperform SIFT in some scenarios.

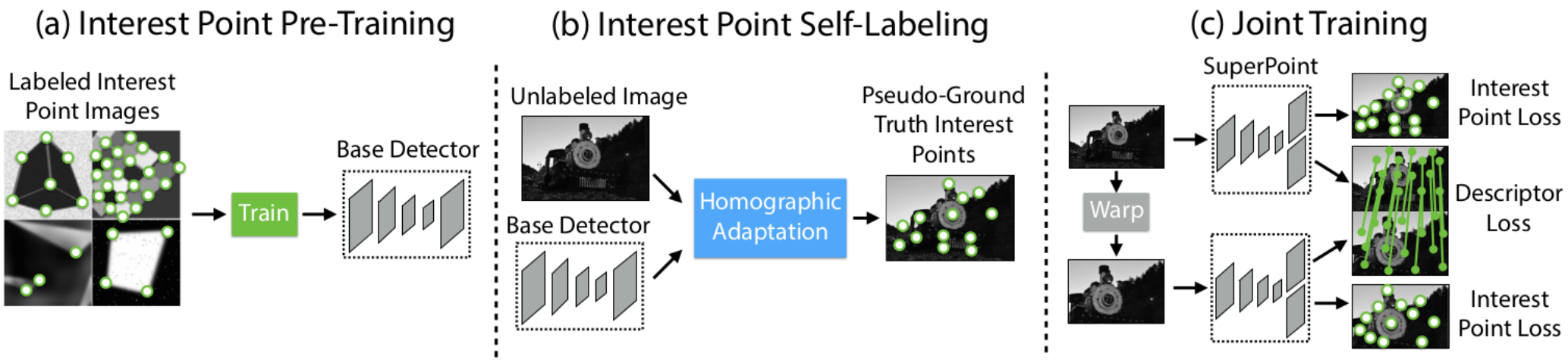


DeTone, Malisiewicz, Rabinovich, *SuperPoint: Self-Supervised Interest Point Detection and Description*, Conference of Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. PDF

SuperPoint - Training

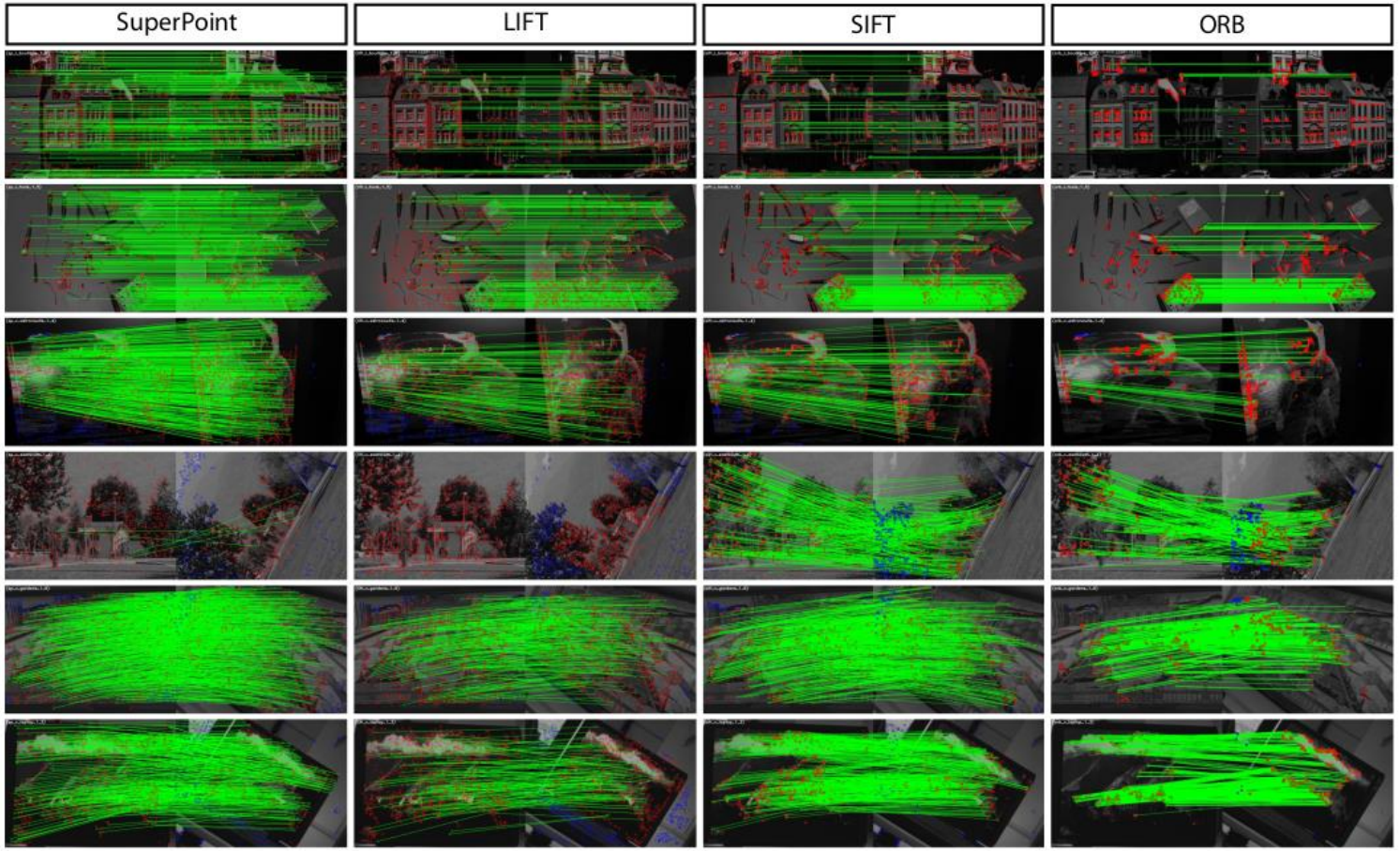
- Training Steps:

a) Training on synthetic dataset to bootstrap the detector
b) Use this detector on real images and generate groundtruth correspondences by sampling perspective transformations
c) Jointly training both detector and descriptor



DeTone, Malisiewicz, Rabinovich, *SuperPoint: Self-Supervised Interest Point Detection and Description*, Conference of Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. PDF

SuperPoint - Results



DeTone, Malisiewicz, Rabinovich, *SuperPoint: Self-Supervised Interest Point Detection and Description,* Conference of Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. PDF

SuperPoint - Results

	Homography Estimation			Detector Metrics		Descriptor Metrics	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Rep.	MLE	NN mAP	M. Score
SuperPoint	.310	**.684**	**.829**	.581	1.158	**.821**	**.470**
LIFT	.284	.598	.717	.449	1.102	.664	.315
SIFT	**.424**	.676	.759	.495	**0.833**	.694	.313
ORB	.150	.395	.538	**.641**	1.157	.735	.266

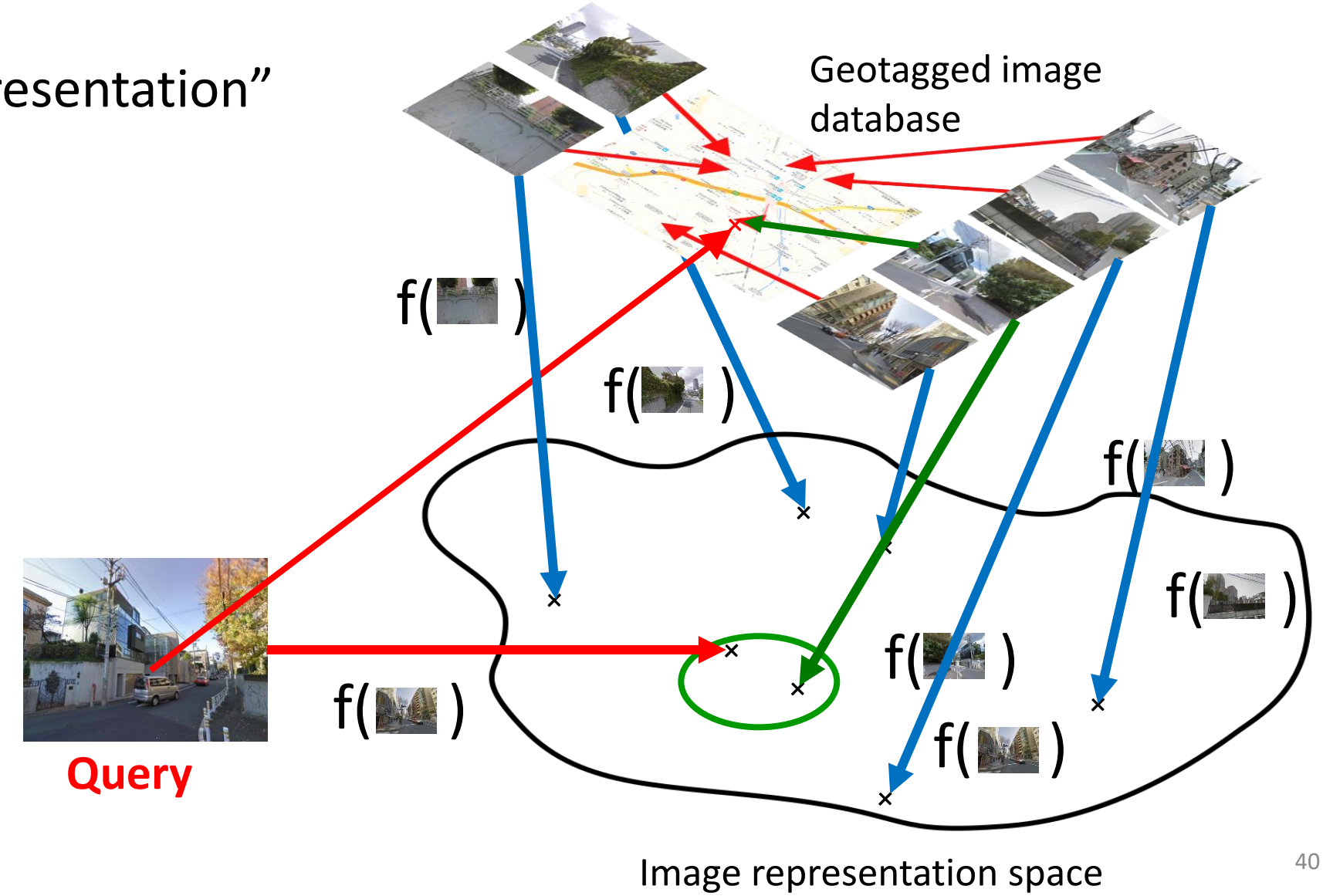
Legend

- **ε**: Pixel distance to count keypoint as correct. SIFT is very accurate due to sub-pixel refinement.
- **MLE** (Mean Localization Error): Lower is better. Metric for determining detector accuracy
- **NN mAP** (nearest-neighbor mean average precision): Measures discriminativeness of descriptors. Higher is better.
- **M. Score** (Matching Score): Evaluates detector and descriptor jointly on groundtruth correspondences.

DeTone, Malisiewicz, Rabinovich, *SuperPoint: Self-Supervised Interest Point Detection and Description,* Conference of Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. PDF

Place Recognition

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

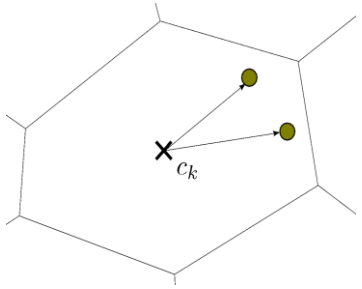


NetVLAD

- Mimic the classical pipeline with deep learning

Image I

Extract local features (SIFT)

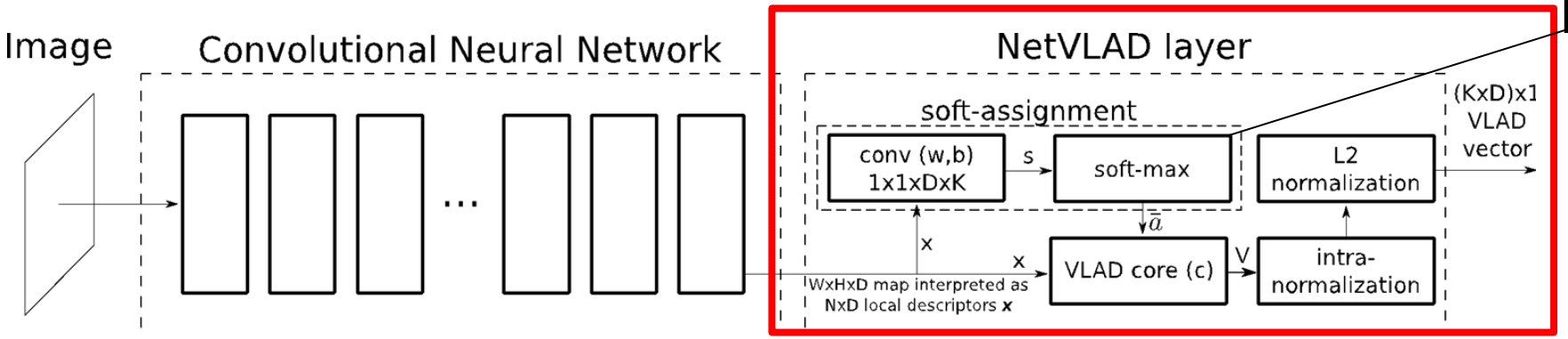


Aggregate (BoW, **VLAD**, FV)

$$\begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ \dots \end{pmatrix}$$

F(I)

$$\boxed{\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}}$$



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

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

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

margin

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

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

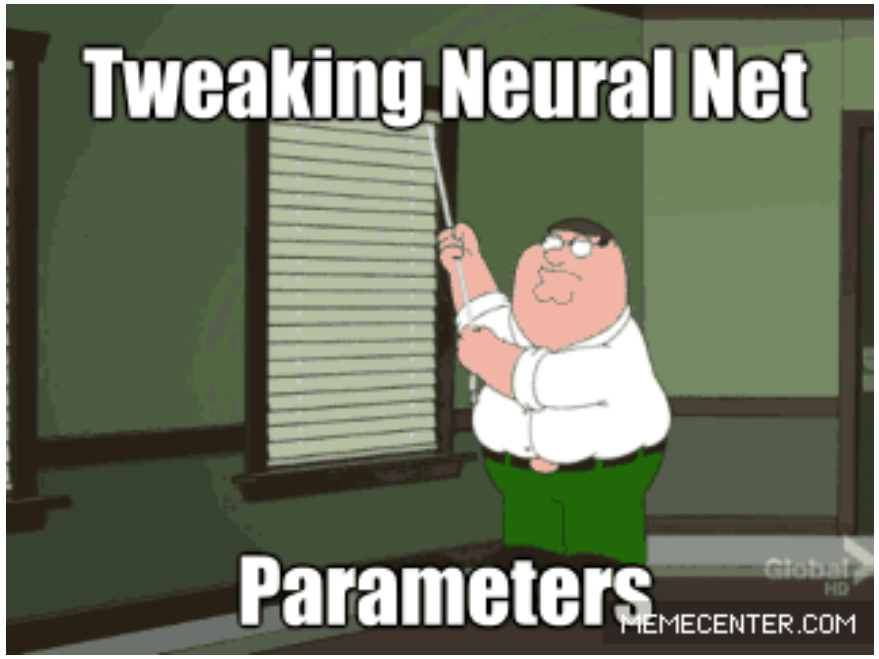
Green: Correct Red: Incorrect

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

Deep Learning Limitations

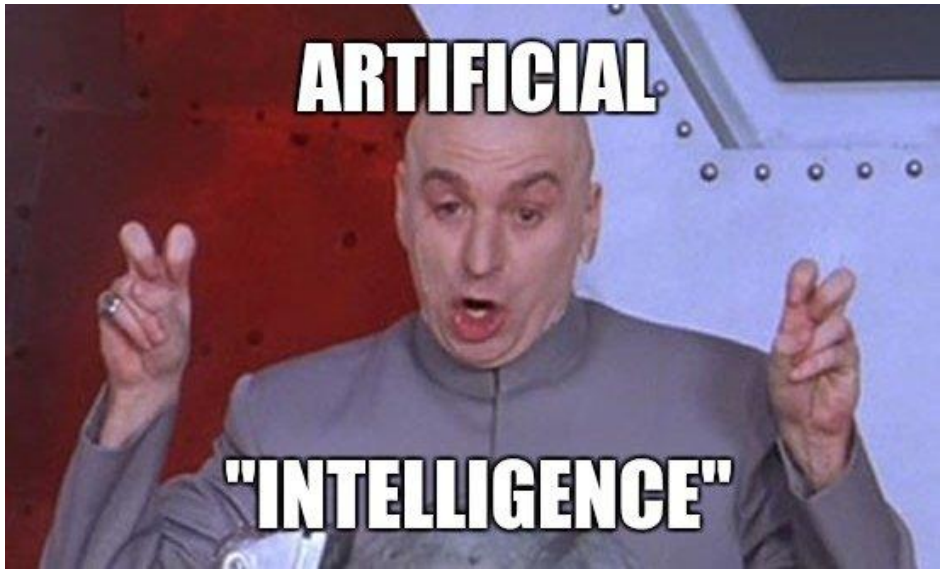
- Requires a **lot of data** to learn
- **Difficult debugging** and finetuning
- **Poor generalization** across similar tasks

Neural Networks Practitioners



Things to Remember

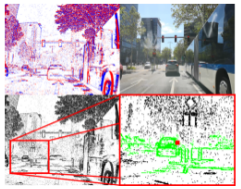
- Deep Learning is able to **extract meaningful patterns** from data.
- It can be applied to **a wide range of tasks**.
- **Artificial Intelligence ≠ Deep Learning**



Come over for projects in DL!

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

Asynchronous Processing for Event-based Deep Learning - Available



Description: Event cameras such as the Dynamic Vision Sensor (DVS) are recent sensors with large potential for high-speed and high dynamic range robotic applications. Since their output is sparse traditional algorithms, which are designed for dense inputs such as frames, are not well suited. The goal of this project is explore ways to adapt existing deep learning algorithms to handle sparse asynchronous data from events. Applicants should have experience in C++ and python deep learning frameworks (tensorflow or pytorch), and have a strong background in computer vision.

Goal: The goal of this project is explore ways to adapt existing deep learning algorithms to handle sparse asynchronous data from events.

Contact Details: Daniel Gehrig (dgehrig at ifi.uzh.ch)
Thesis Type: Semester Project / Master Thesis
See project on SiROP

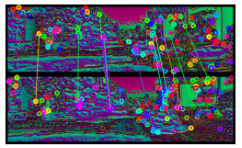
Designing a New Event Camera with Events and Images - Available

Description: Event cameras such as the Dynamic Vision Sensor (DVS) are recent sensors with a lot of potential for high-speed and high dynamic range robotic applications. They have been successfully applied in many applications, such as high speed video and high speed visual odometry. Due to their high speed and

Goal: The goal of this project is to design a new event camera that combines events and standard images.

Contact Details: Daniel Gehrig (dgehrig (at) ifi (dot) uzh (dot) ch), Mathias Gehrig (mgehrig (at) ifi (dot) uzh (dot) ch)
Thesis Type: Semester Project / Master Thesis
See project on SiROP

Data-driven Keypoint Extractor for Event Data - Available

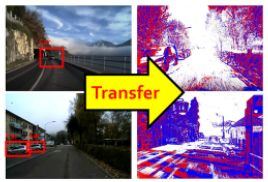


Description: Neuromorphic cameras exhibit several amazing properties such as robustness to HDR scenes, high-temporal resolution, and low power consumption. Thanks to these characteristics, event cameras are applied for camera pose estimation for fast motions in challenging scenes. A common technique 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.

Goal: The project aims to develop a data-driven keypoint extractor, which computes interest points in event data. Based on the current advances of learned keypoint extractors for traditional frames, the approach will leverage neural network architectures to extract and describe keypoints in an event stream. The student should have prior programming experience in a deep learning framework and completed at least one course in computer vision.

Contact Details: Contact Details: Nico Messikommer [nmessi (at) ifi (dot) uzh (dot) ch], Mathias Gehrig [mgehrig (at) ifi (dot) uzh (dot) ch]
Thesis Type: Semester Project / Master Thesis
See project on SiROP

Domain Transfer between Events and Frames - Available



Description: During the last years, a vast collection of frame-based datasets was collected for countless tasks. In comparison, event-based datasets represent only a tiny fraction of the available datasets. Thus, it is highly promising to use labelled frame datasets to train event-based networks as current data-driven approaches heavily rely on labelled data.

Goal: In this project, the student extends current advances from the UDA literature for traditional frames to event data in order to transfer multiple tasks from frames to events. The approach should be validated on several tasks (segmentation, object detection, etc.) in challenging environments (night, high-dynamic scenes) to highlight the benefits of event cameras. As several deep learning methods are used as tools for the task transfer, a strong background in deep learning is required. If you are interested, we are happy to provide more details.

Contact Details: Nico Messikommer [nmessi (at) ifi (dot) uzh (dot) ch], Daniel Gehrig [dgehrig (at) ifi (dot) uzh (dot) ch]
Thesis Type: Semester Project / Master Thesis
See project on SiROP

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/1IjesolDoUu1-IVX14nqJRCT-wpEQB22_/view

How are current commercial drones controlled?

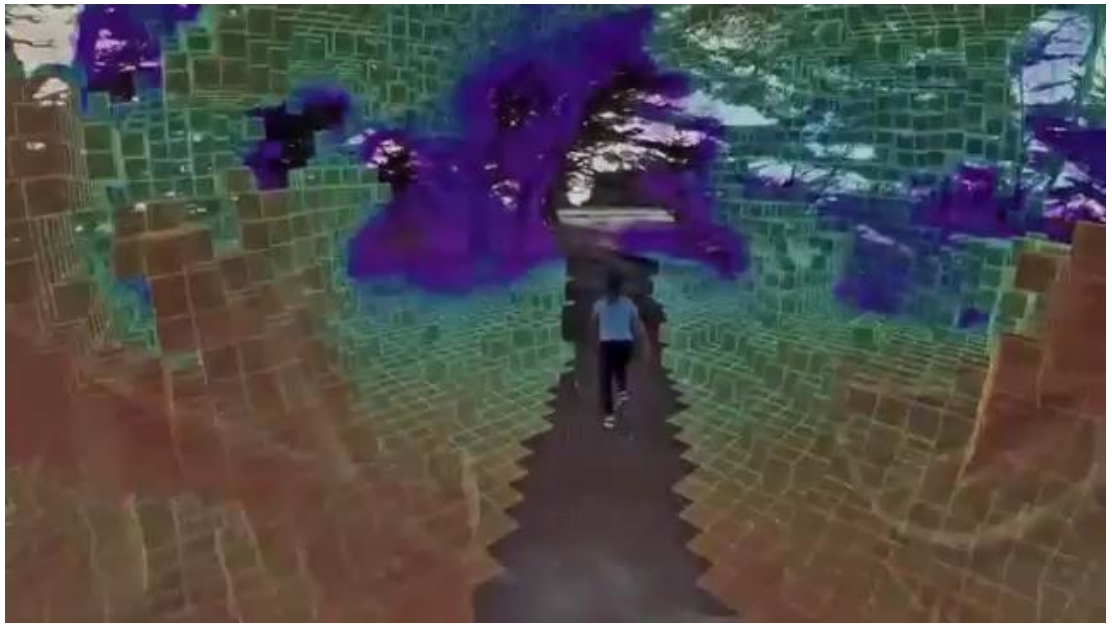
- **By a human pilot**
  - requires **line of sight** or **video link**
  - requires a **lot of training**

- **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)

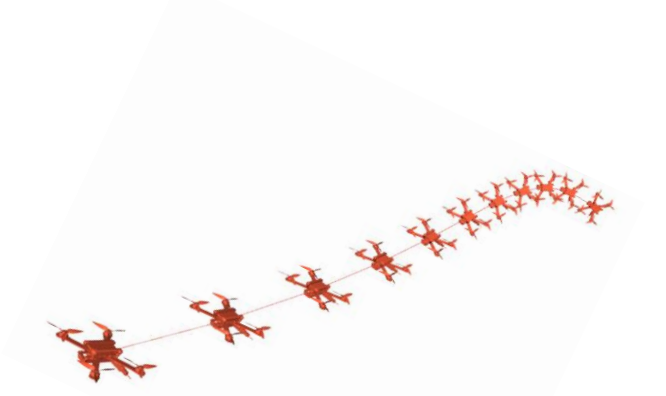


2020

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

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

Related Work: The Traditional Approach



Images
IMU

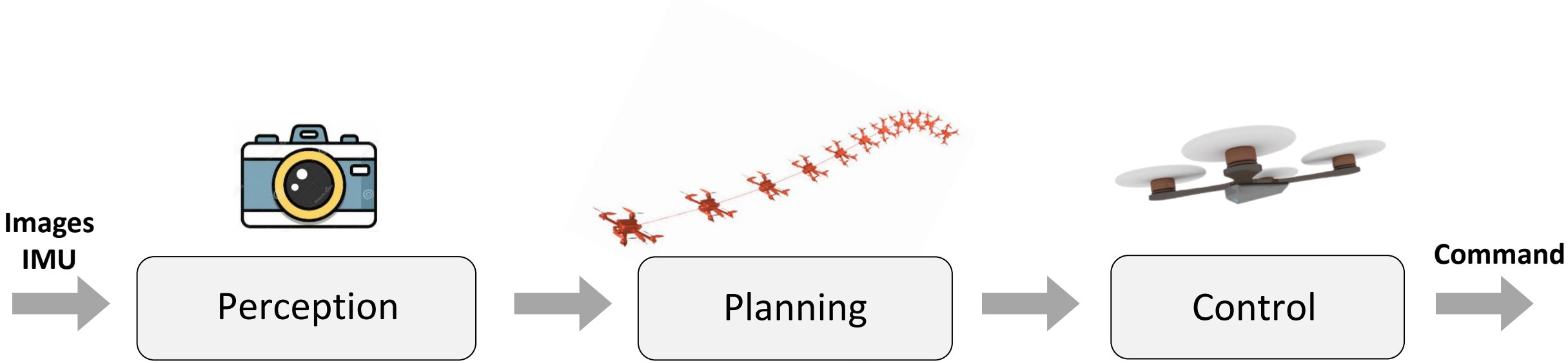
→ Perception → Planning → Control → **Command**

State Estimation
Mapping

Polynomial Trajectories
Trajectory Optimization

PID Control
Model Predictive Control
Reinforcement Learning
Iterative Learning Control

Related Work: The Problem



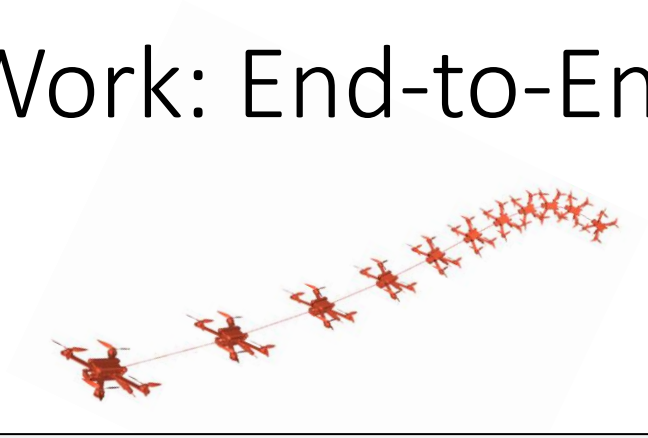
Mature Algorithms, but brittle during high speed due to motion blur.

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

Needs significant tuning, especially at high speed.

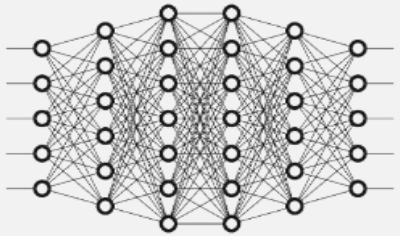
! This fine-grained modularity makes the robotic system fragile:
The modules do not interact with each other.

Related Work: End-to-End Learning



Images
IMU

Neural Network



Command

High-Level commands (forward, left, right)

! Don't exploit the agile dynamics of the drone

Low-Level commands (collective thrust, bodyrates)

! 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

- 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**?

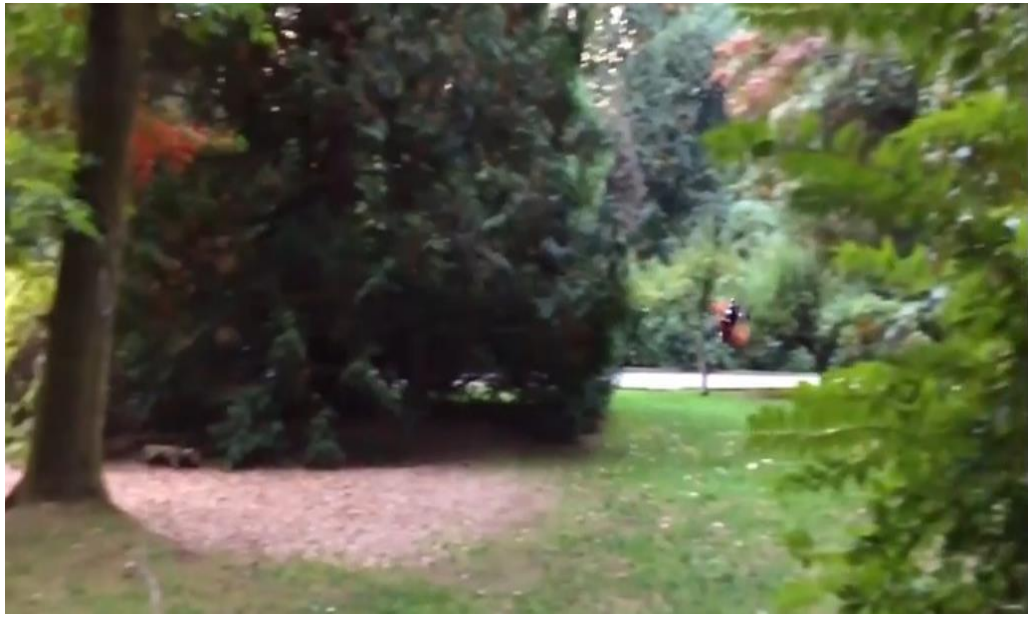
Human pilots fly under similar assumptions!

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).

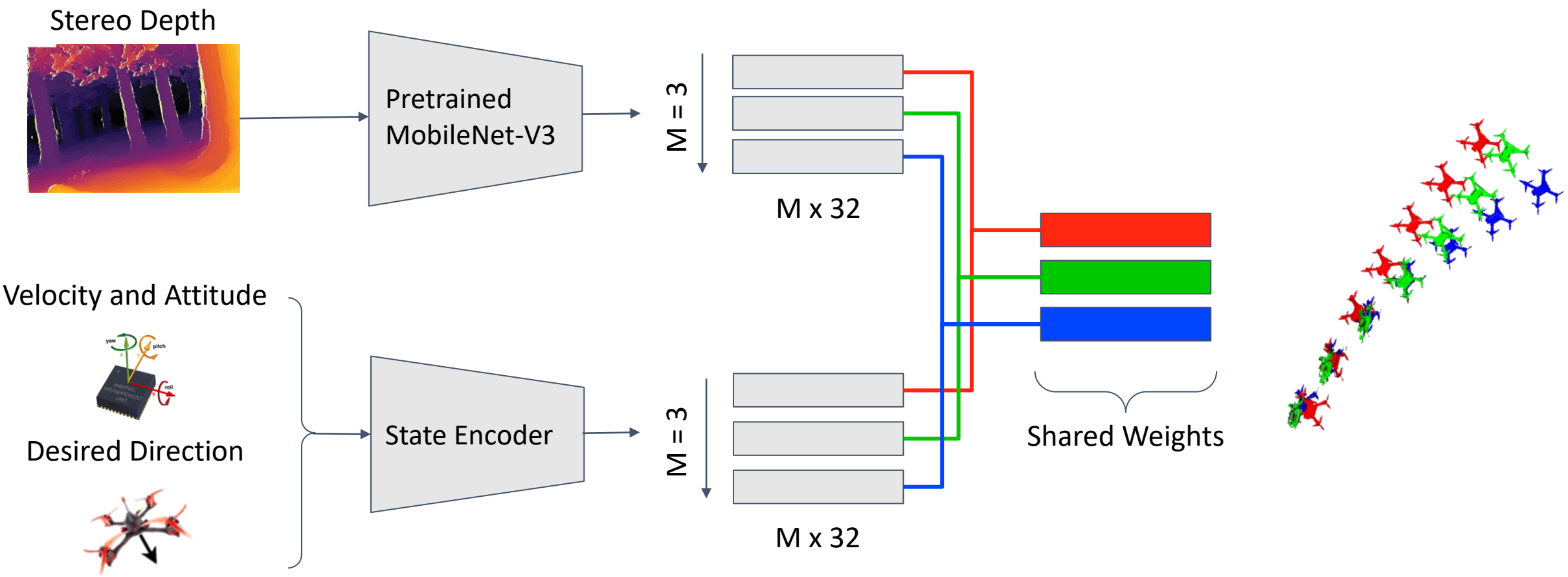


Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. PDF, Video, Code

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 <u>expert planner</u> 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!

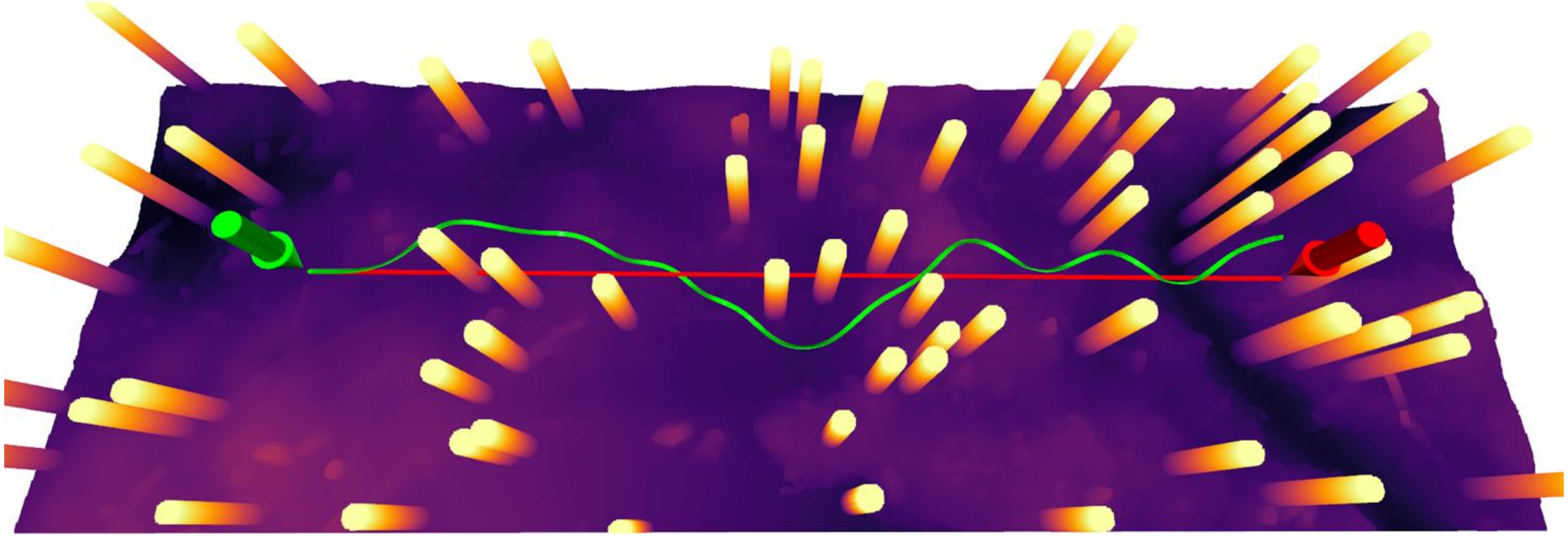
* Impossible to collect a dataset of real-world demonstrations since it is not possible (or very expensive) to have a perfect map of the environment.

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

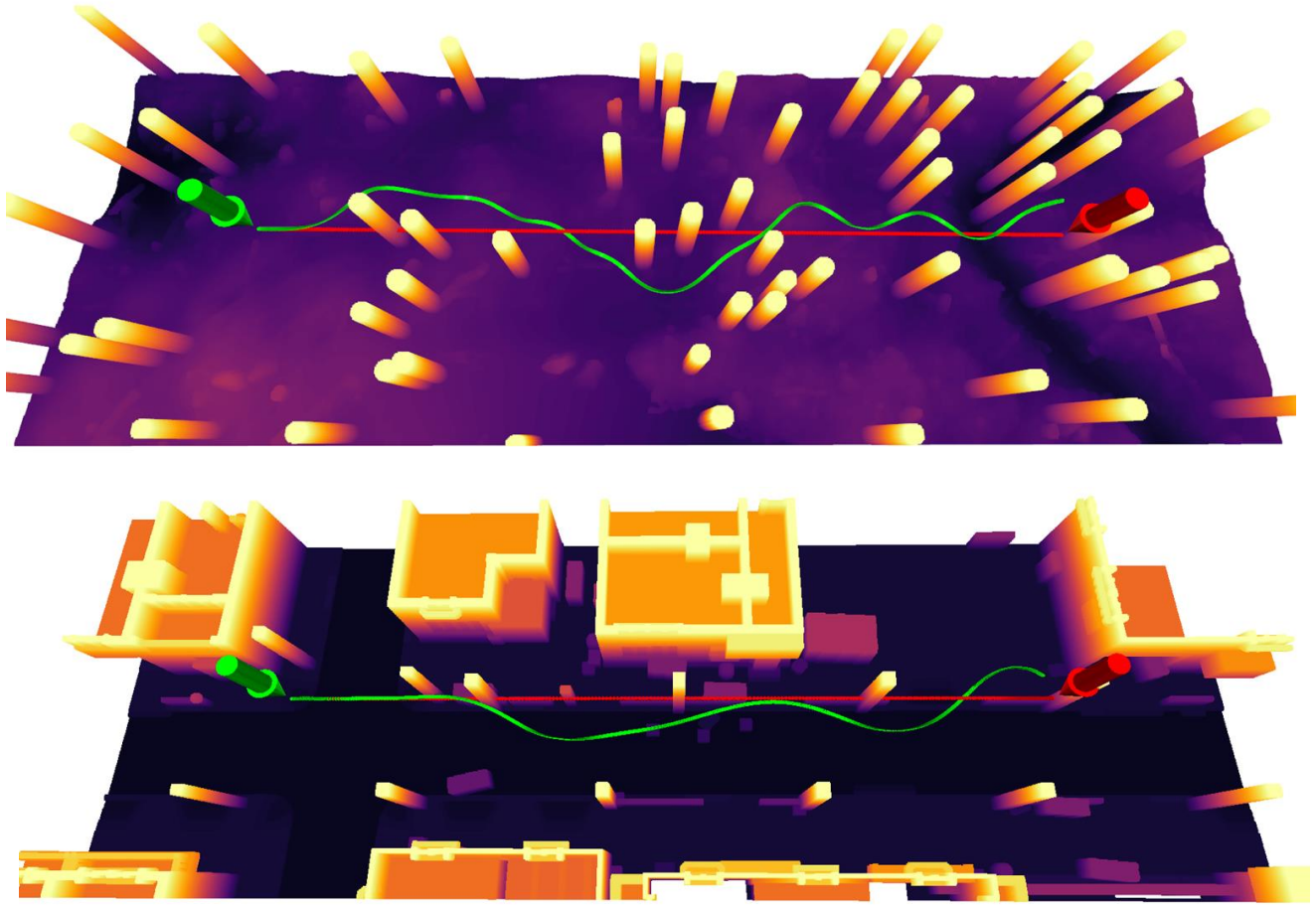
Evaluate on the task of reaching a goal with no prior knowledge about the scene.

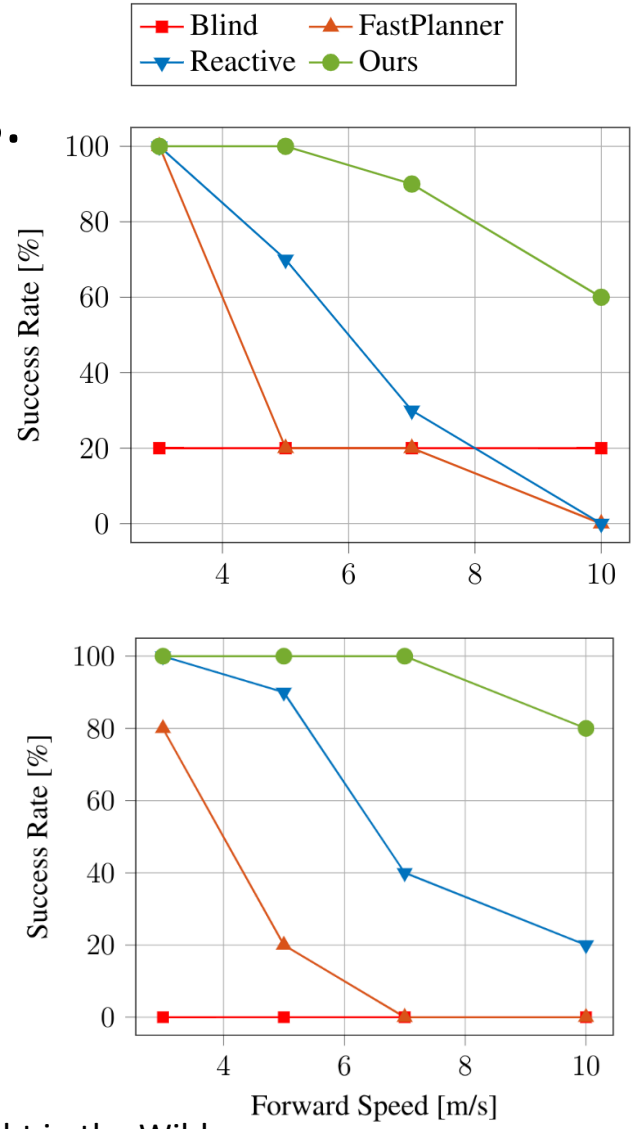


Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. PDF, Video, Code

Controlled Experiments

Comparison with state-of-the-art navigation methods.



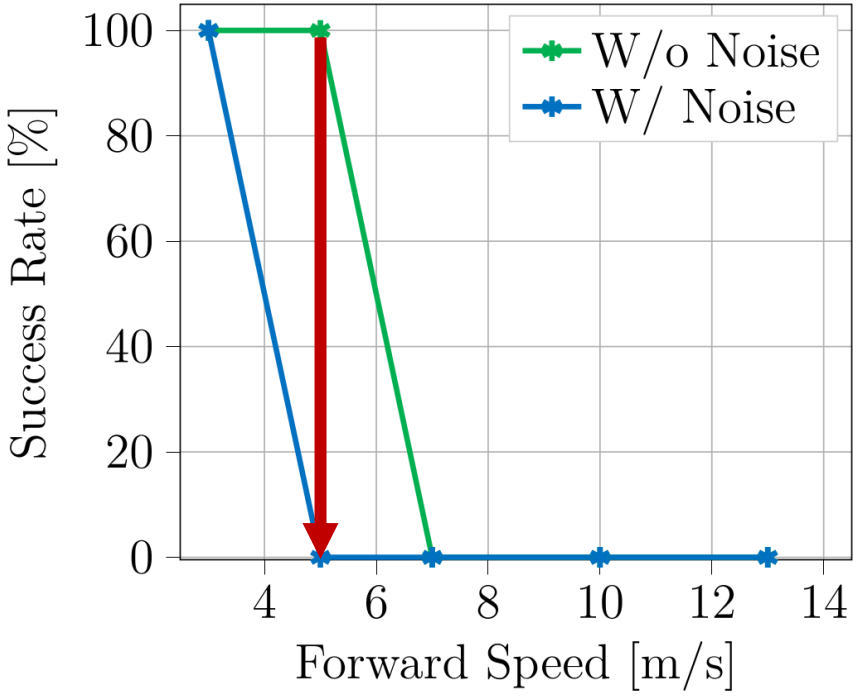


Loquercio, Kaufmann, Ranftl, Mueller, Koltun, Scaramuzza: Learning, High-Speed Flight in the Wild, Science Robotics, 2021. PDF, Video, Code

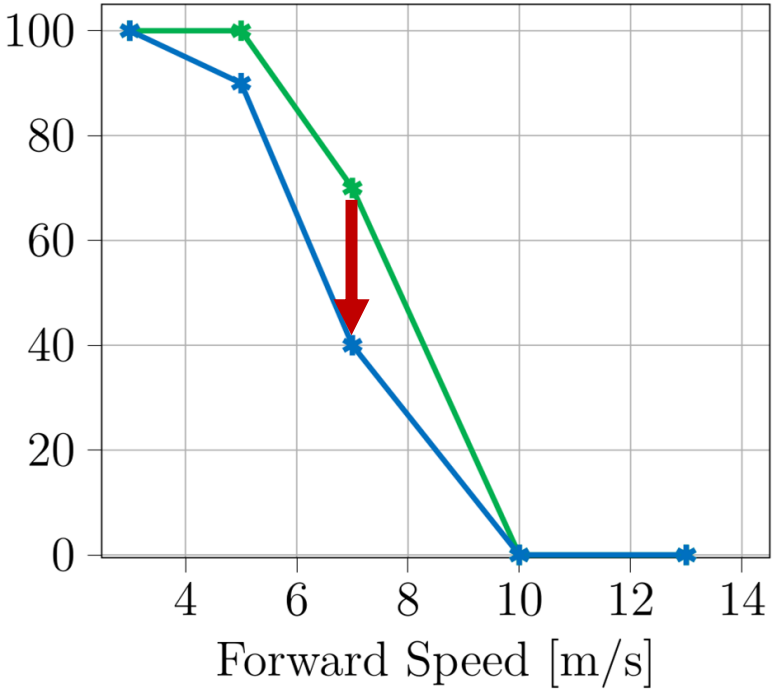
Controlled Experiments

Robustness to sensor noise:

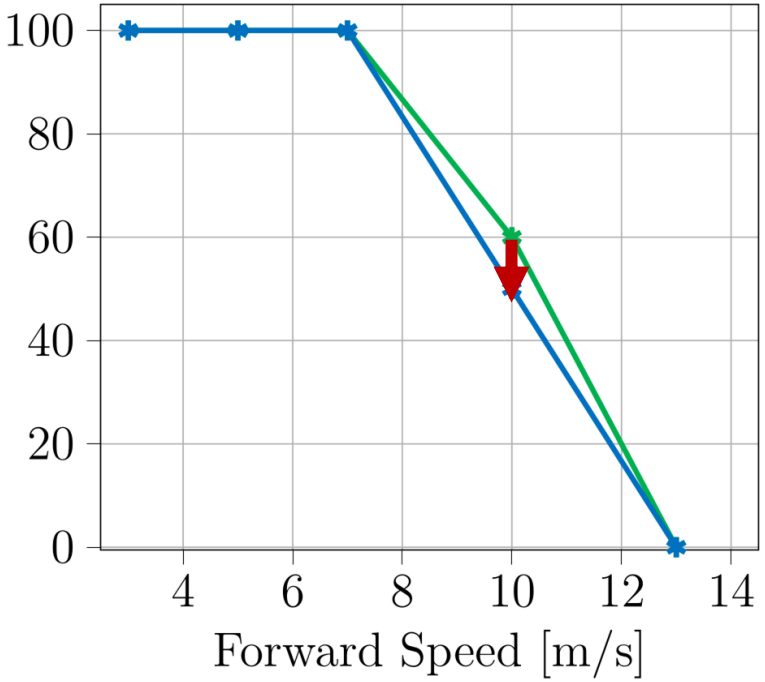
Zhou et al., *TRO-2020*



Florence et al., *AFR-2020*



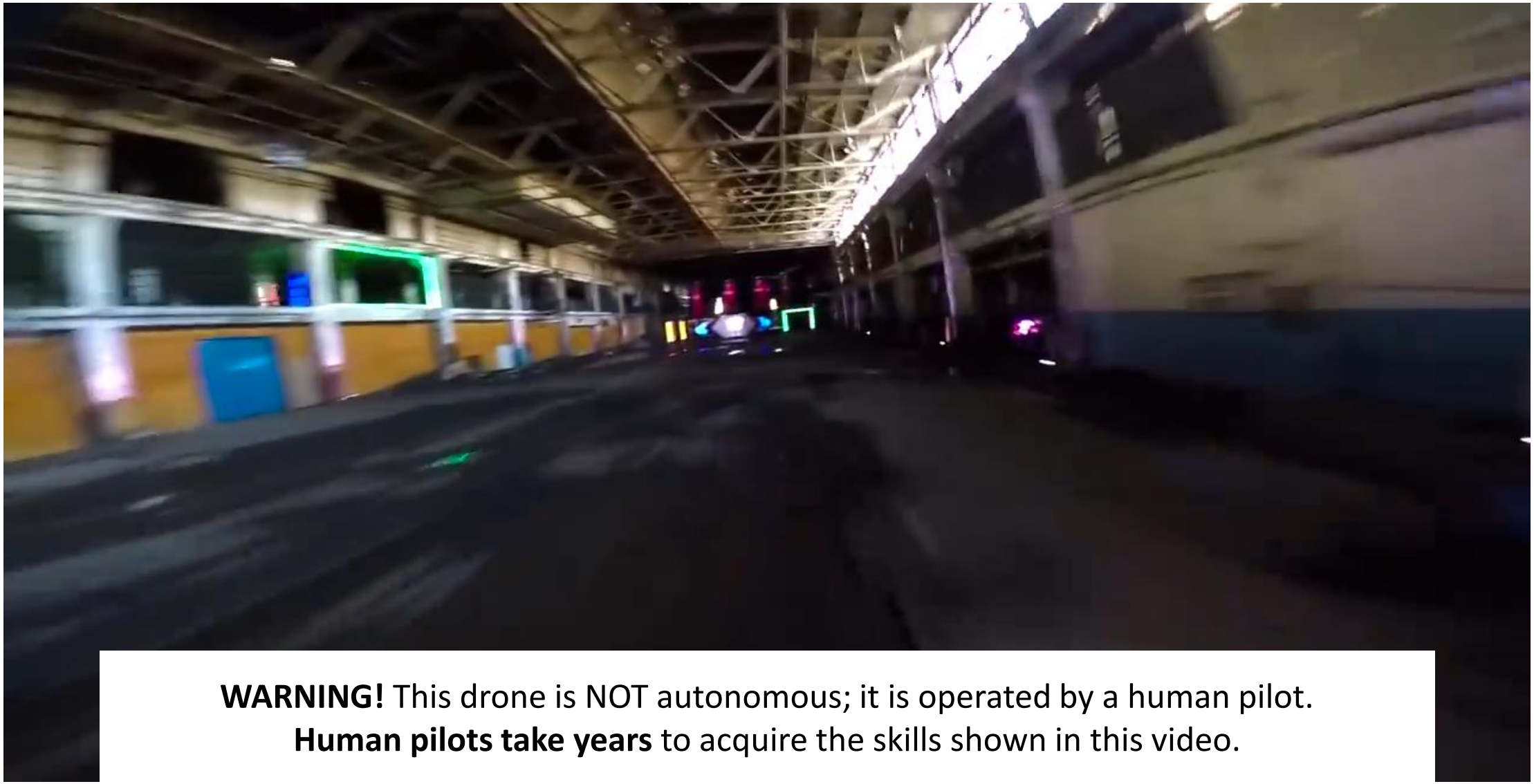
Kaufmann et al. (Ours)



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

Weight:	900g
Maximum Thrust:	45N
Thrust-to-Weight:	ca. 5

Autonomous Drone Racing



WARNING! This drone is NOT autonomous; it is operated by a human pilot.
Human pilots take years to acquire the skills shown in this video.

Challenges of Autonomous Drone Racing

- **Real-time coupling** of **perception** and **action**
- Coping with **inaccurate models** of **sensors**, **actuators**, **environment**
- Coping with **dynamically changing** environments
- Coping with **unreliable perception** and **state estimation**:
  - low texture
  - HDR scenes
  - motion blur

Autonomous Racing in Motion Capture

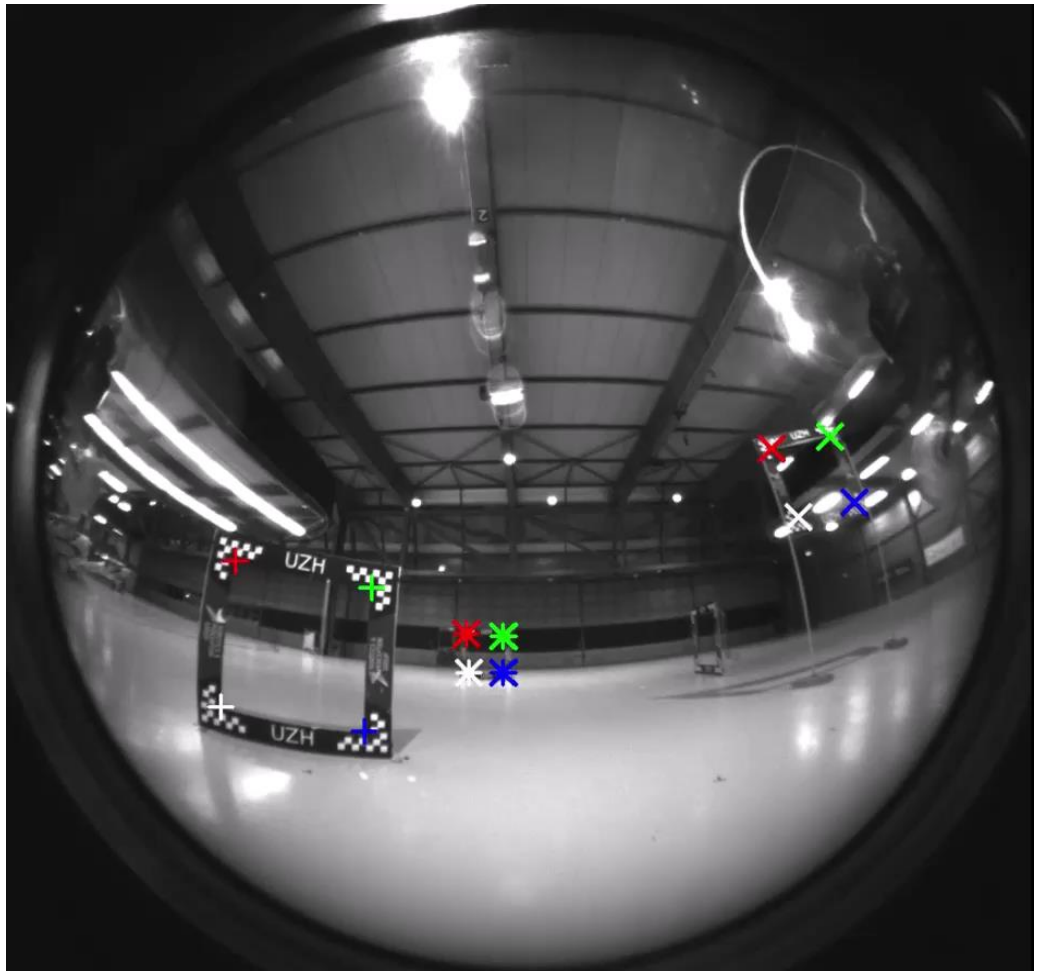
[1] Foehn et al., *Time-Optimal Planning for Quadrotor Waypoint Flight*, **Science Robotics**, 2021. PDF. Video. Code. **Featured on Forbes magazine**.
[2] Foehn et al., *AlphaPilot: Autonomous Drone Racing*, **RSS** 2020, **Best Systems Paper Award**. PDF Video. **2nd place at AlphaPilot Challenge**
[3] Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video **1st place at IROS'18 Drone race**. Video.
[4] Loquercio, et al., *Deep Drone Racing,* **IEEE Transactions on Robotics** 2020. **Best Paper Award finalist.** PDF. Video
[5] Song et al, *Autonomous Drone Racing with Deep Reinforcement Learning*, IROS'21. PDF. Video
[6] **Simulator used**: Song et al., *Flightmare: A Flexible Quadrotor Simulator*, CORL'20, PDF Video Website

Autonomous Racing with Onboard Sensing

[1] Foehn et al., *Time-Optimal Planning for Quadrotor Waypoint Flight*, **Science Robotics**, 2021. PDF. Video. Code. **Featured on Forbes magazine**.
[2] Foehn et al., *AlphaPilot: Autonomous Drone Racing*, **RSS** 2020, **Best Systems Paper Award**. PDF Video. **2nd place at AlphaPilot Challenge**
[3] Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video **1st place at IROS'18 Drone race**. Video.
[4] Loquercio, et al., *Deep Drone Racing,* **IEEE Transactions on Robotics** 2020. **Best Paper Award finalist.** PDF. Video
[5] Song et al, *Autonomous Drone Racing with Deep Reinforcement Learning*, IROS'21. PDF. Video
[6] **Simulator used**: Song et al., *Flightmare: A Flexible Quadrotor Simulator*, CORL'20, PDF Video Website

Autonomous Racing with Onboard Sensing

- VIO estimates are combined with learning-based gate detections to compensate for drift.
- VIO, neural network inference are running onboard



[1] Foehn et al., *Time-Optimal Planning for Quadrotor Waypoint Flight*, **Science Robotics**, 2021. PDF. Video. Code. **Featured on Forbes magazine**.
[2] Foehn et al., *AlphaPilot: Autonomous Drone Racing*, **RSS** 2020, **Best Systems Paper Award**. PDF Video. **2nd place at AlphaPilot Challenge**
[3] Kaufmann et al., *Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing*, ICRA'19. PDF. Video **1st place at IROS'18 Drone race**. Video.
[4] Loquercio, et al., *Deep Drone Racing,* **IEEE Transactions on Robotics** 2020. **Best Paper Award finalist.** PDF. Video
[5] Song et al, *Autonomous Drone Racing with Deep Reinforcement Learning*, IROS'21. PDF. Video
[6] **Simulator used**: Song et al., *Flightmare: A Flexible Quadrotor Simulator*, CORL'20, PDF Video Website

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**

Check out our student projects!

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

Reinforcement Learning for Offboard Control of a Racing Drone - Available

Description: Autonomous drone racing using offboard control is very challenging because a limited amount of sensor data are available to the offboard computer. Offboard control however, allows using extremely lightweight drones that can reach much higher performances than heavier drones using onboard control. This project has two goals: First, implement and test a communication interface between a ROS-based flight stack and a C/Python-based codebase for low-latency high-bandwidth communication between the drone and an offboard computer. Second, use reinforcement learning to develop a policy that can successfully fly an offboard-controlled racing drone in the real world. Requirements: Strong background in robotics and machine learning is required. ROS, C++, and Python skills. Experience with quadrotor hardware and drone flight is a plus.

Benchmarking Algorithms for Autonomous Drone Racing - Available

Description: Drone racing is an increasingly popular sport and many world-class pilots use simulators for practicing their skills. This project aims to deploy control algorithms for autonomous drone racing in the Liftoff Drone Racing simulator (https://www.liftoff-game.com/), collect data from the autonomous and human-piloted flight, and perform a benchmark comparison between both. The student will implement a control interface between the Liftoff simulator and an existing ROS-based flight stack, collect and analyze flight performance data. Requirements: Strong programming experience in ROS, C++, and Python. Experience in Unity3D is a plus.

Perception Aware Minimum-time Planning in Cluttered Environments - Available

Description: Autonomous drone racing requires planning trajectories that pass through a sequence of gates as fast as possible to beat other competitors. However, planning high-speed trajectories in obstacle-cluttered environments for dynamic systems like drones is a challenging problem. Moreover, the planning algorithms should support autonomous vision-based flight by creating trajectories with respect to the onboard camera used for both state estimation and gate detection. The goal of this project is to develop planning algorithms for drones that consider known cluttered environments and plan perception aware minimum-time trajectories. Applications should have strong experience in C++ and experience with ROS.

Deep Learning for Model Predictive Contouring Control - Available

Description: Model Predictive Contouring Control (MPCC) has shown to achieve very good results in the task of time-optimal multi-waypoint flight. MPCC methods have the freedom to select the optimal states of the system at runtime, dropping the need for a computationally expensive reference trajectory. Our recent work shows MPCC can achieve better lap times than state-of-the-art planning+tracking approaches, and that the method can be run in real-time.

Goal: One of the extra benefits of the MPCC approach is that there are only two relevant parameters to be tuned in the cost function: contour weight and progress weight. In this project, we aim to exploit the low dimensionality of this tuning parameter space and apply learning techniques to find a mapping from a high-level task (track waypoints in a certain order in minimum time, for example) to MPCC tuning parameters.