Self-supervised learning

Alessandro Giusti

Dalle Molle Institute for Artificial Intelligence
Lugano, Switzerland

Contact: alessandrog@idsia.ch

https://idsia-robotics.github.io/

IDSIA

Two main meanings for SSL

- Systems (typically robots) that collect their own training data but then solve a standard supervised learning task
- Systems that learn to extract meaningful representations from the data itself

Two main meanings for SSL

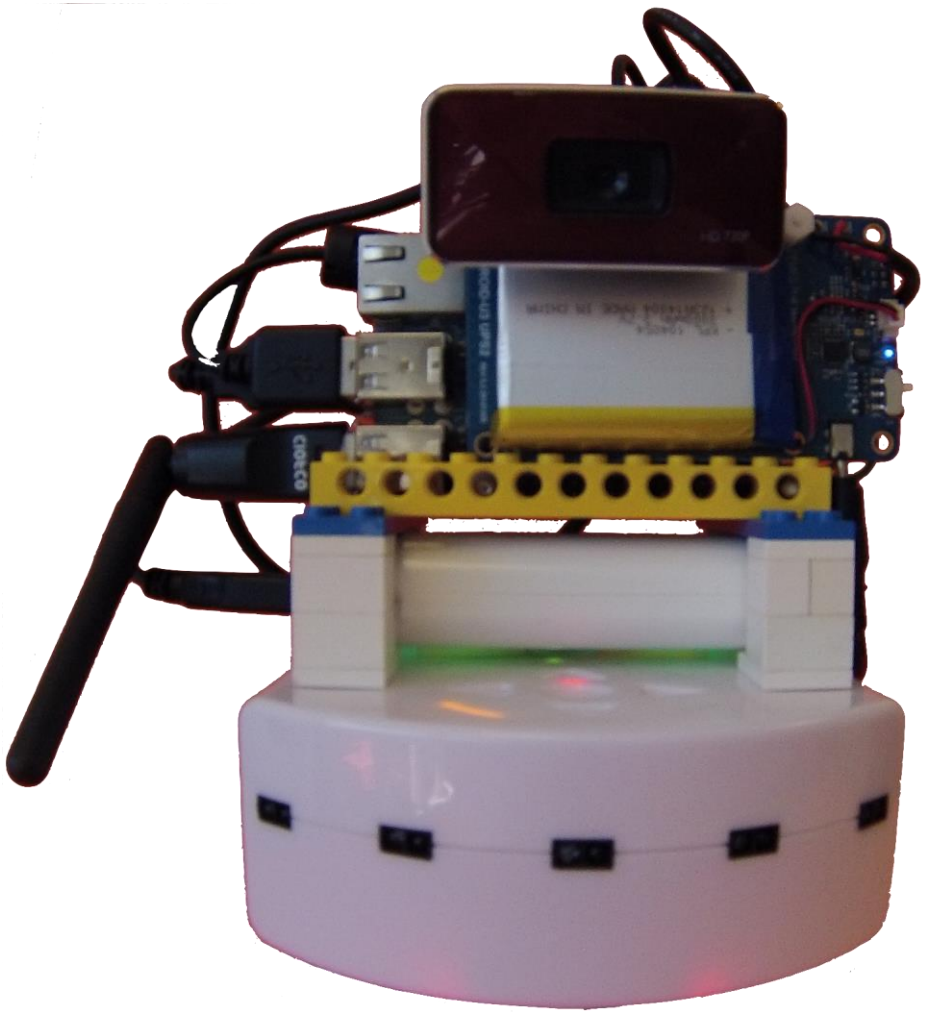
- Systems (typically robots) that collect their own training data but then solve a standard supervised learning task
- Systems that learn to extract meaningful representations from the data itself

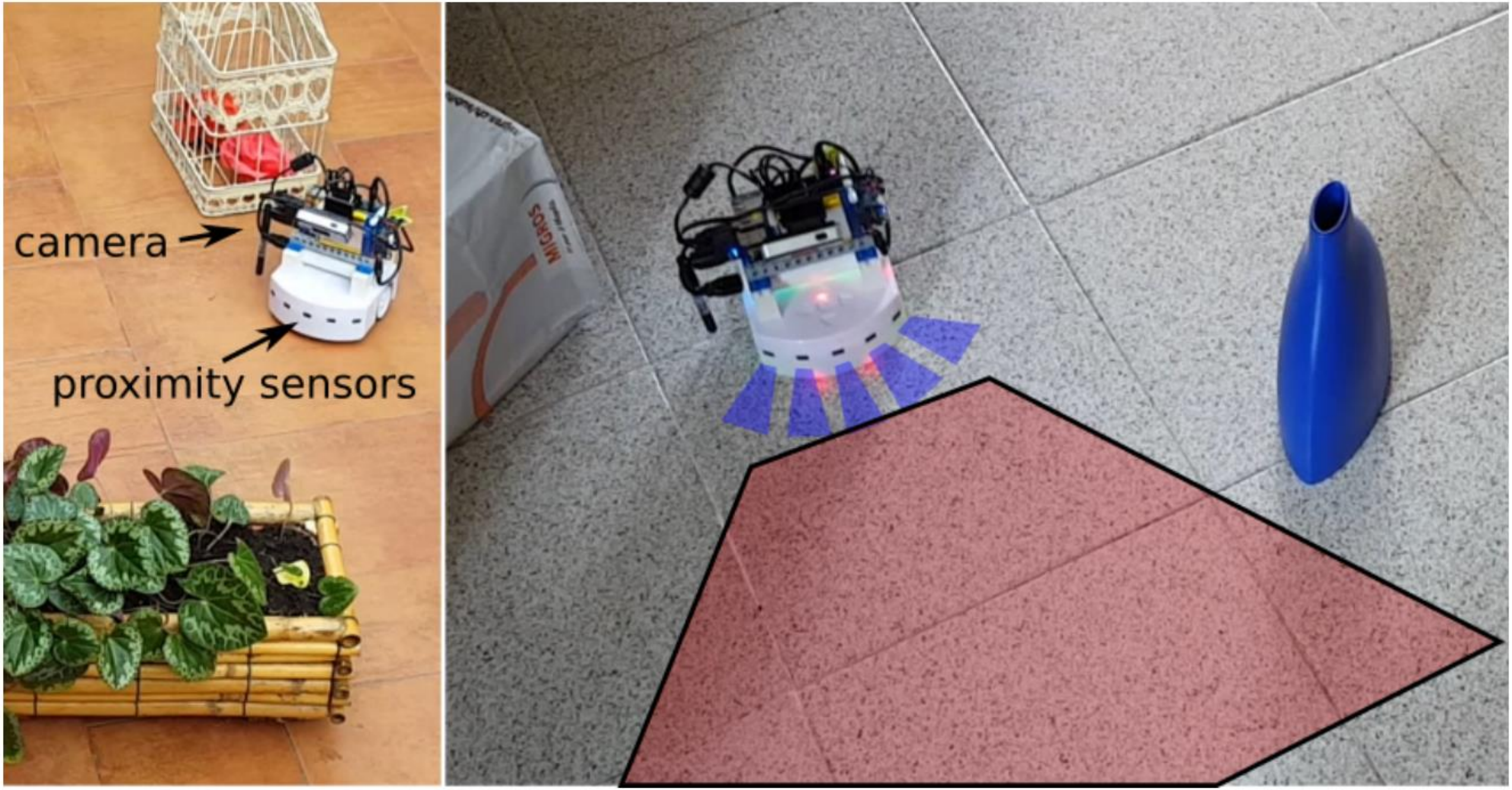
1a. An experiment in self-supervised learning for Robots

A robot that acquires its own training data...
but then solves a standard Supervised Learning problem

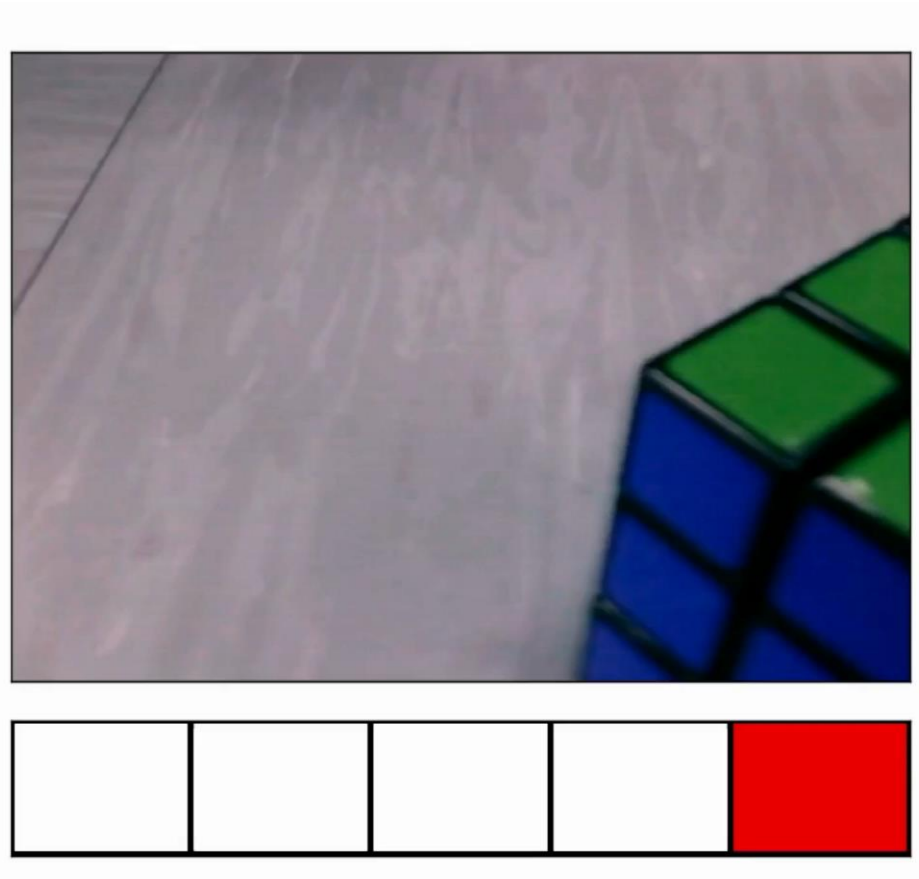
Mighty Thymio

- 5 front-facing infra-red sensors
- 720p camera
- ODRIOD C1
- Wi-Fi connectivity

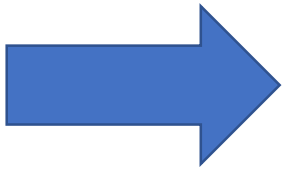


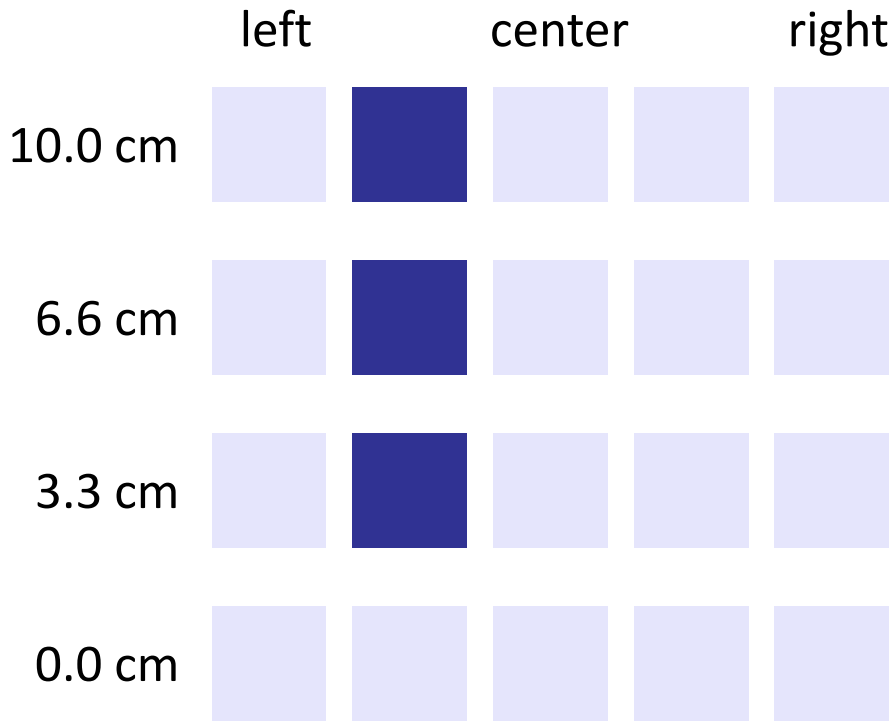

camera
proximity sensors

Cross-sensor prediction (image -> proximity)

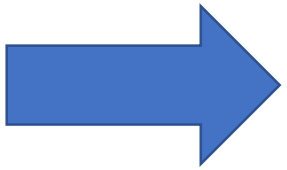


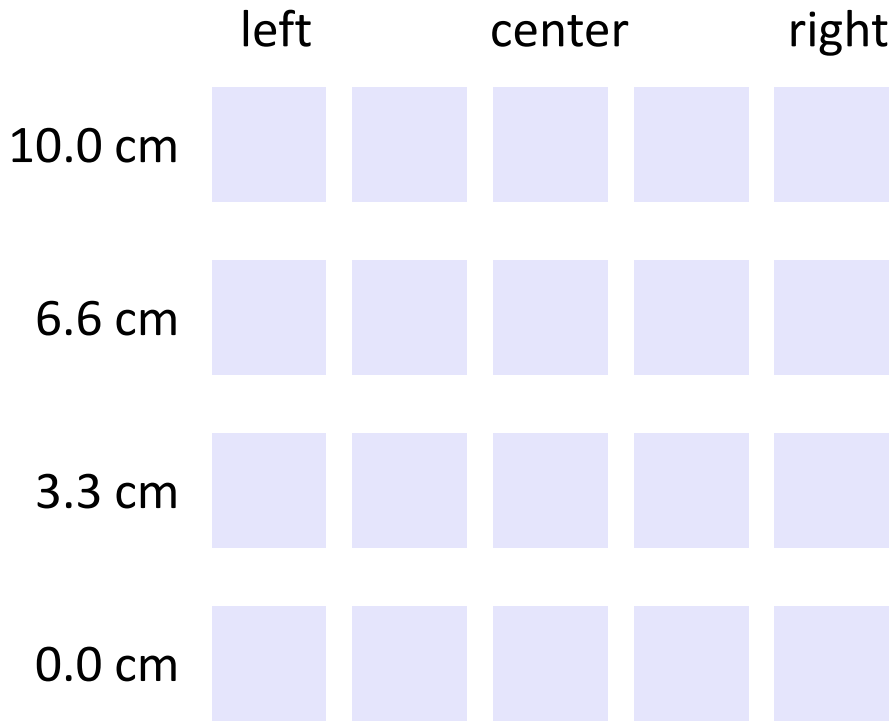
Problem definition example

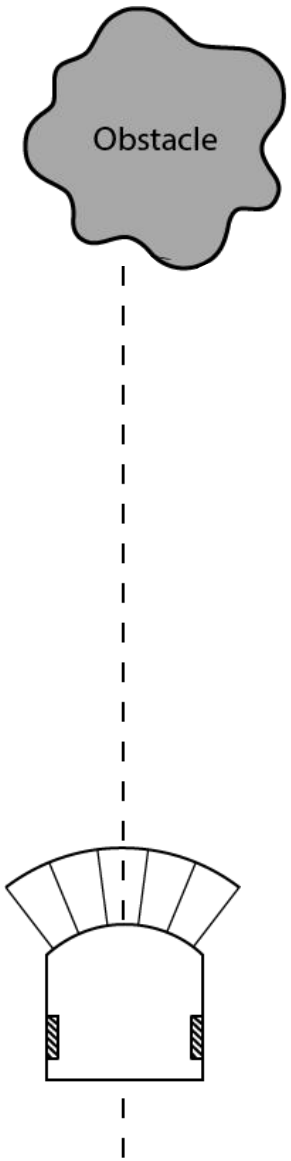


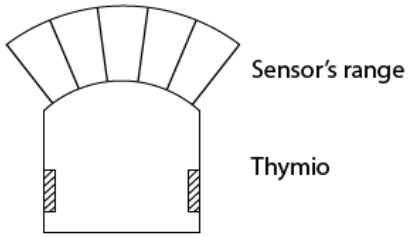


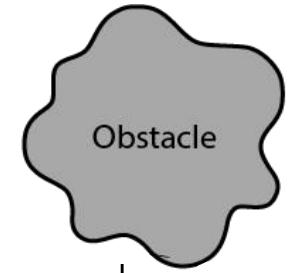
Problem definition example

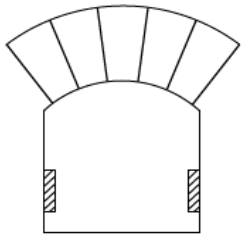


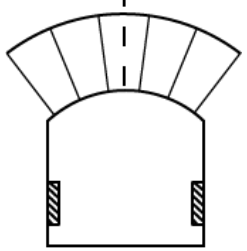


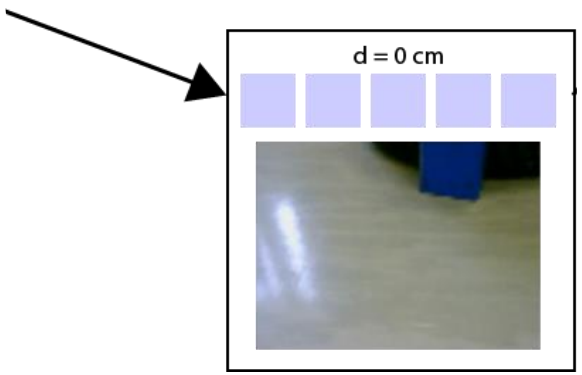

Obstacle

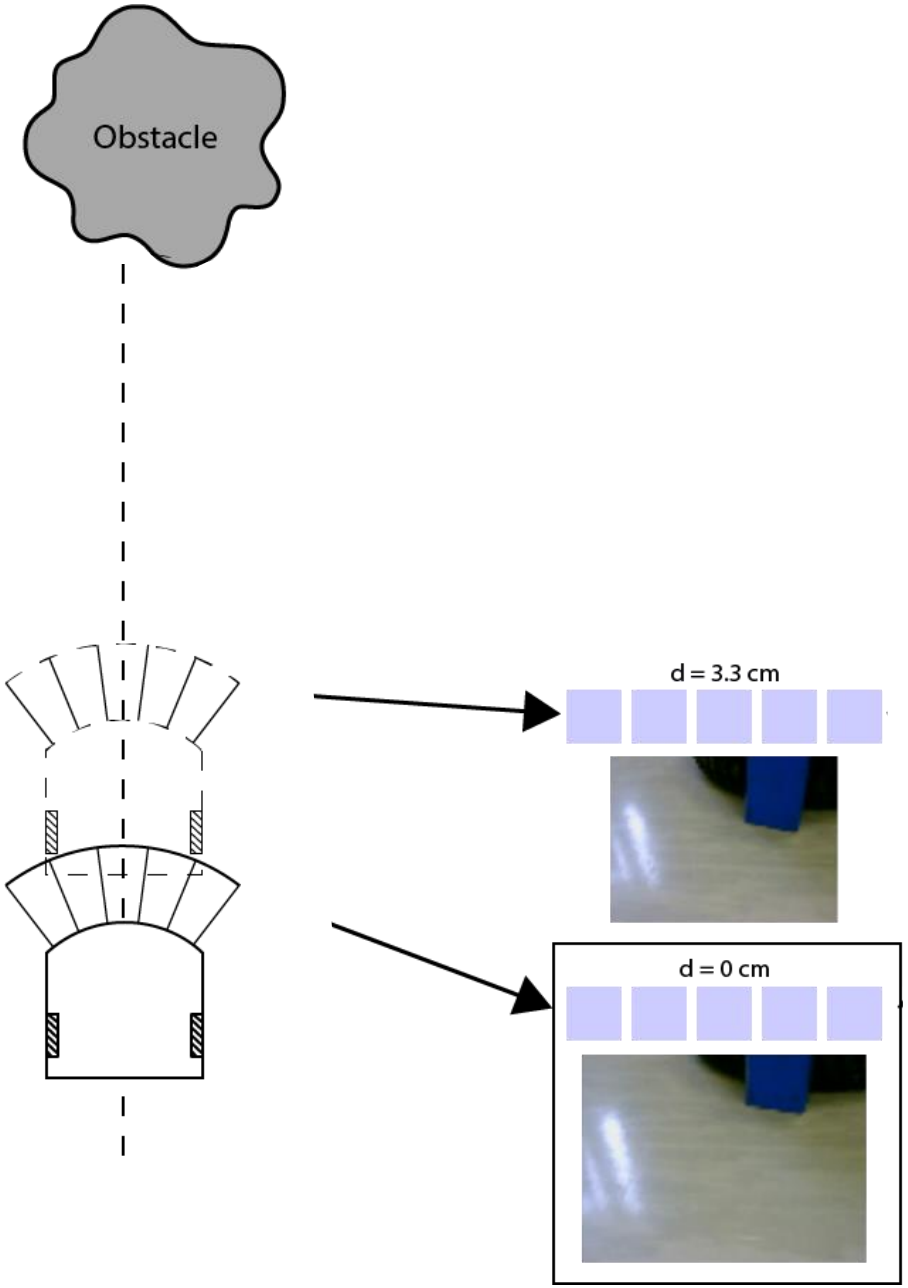

Sensor's range
Thymio

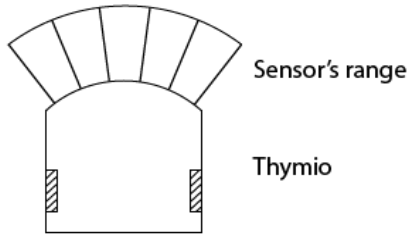

Obstacle

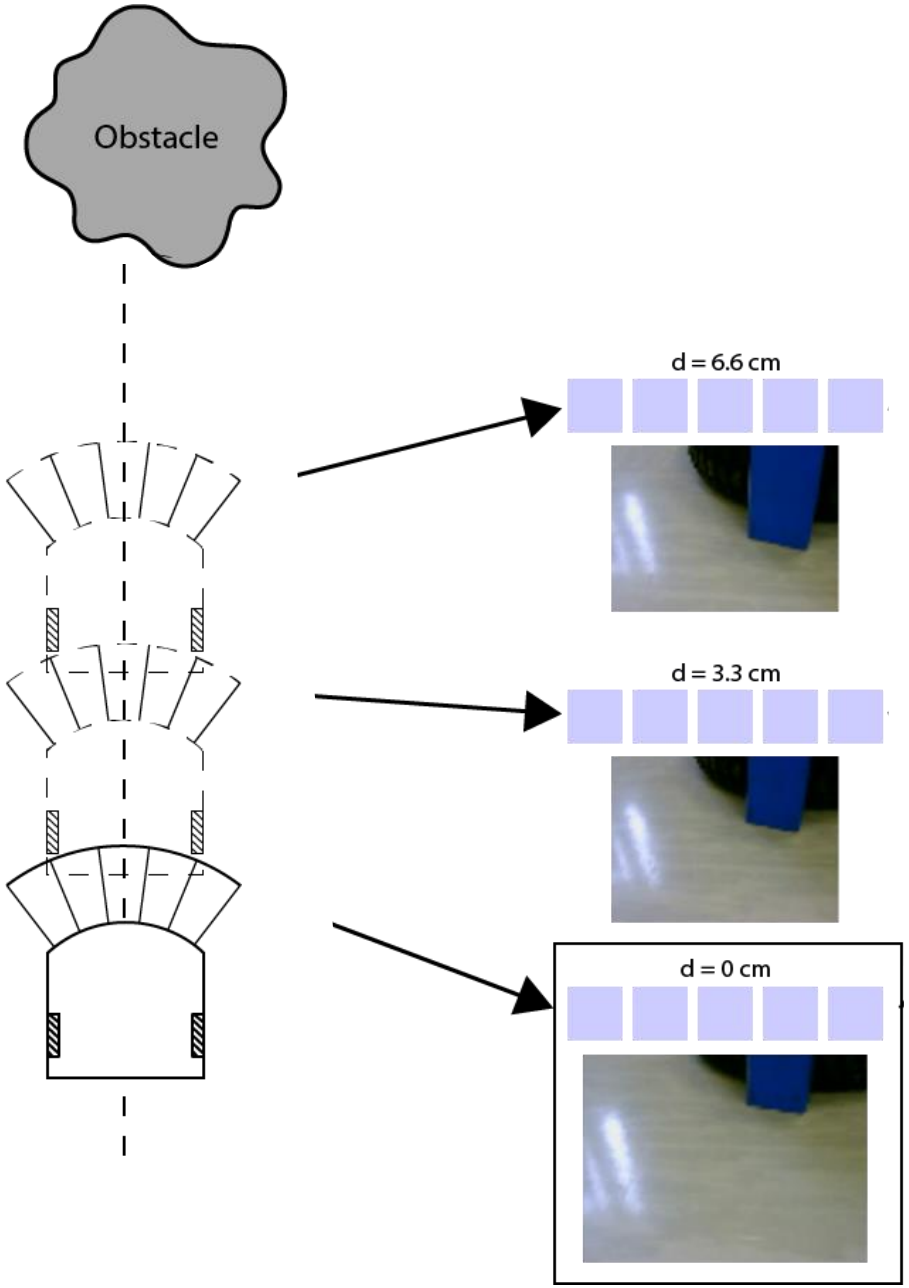

Sensor's range
Thymio

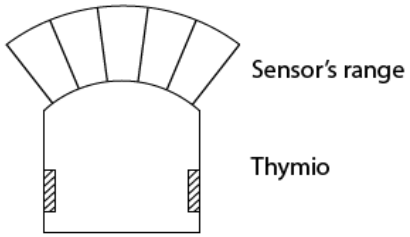


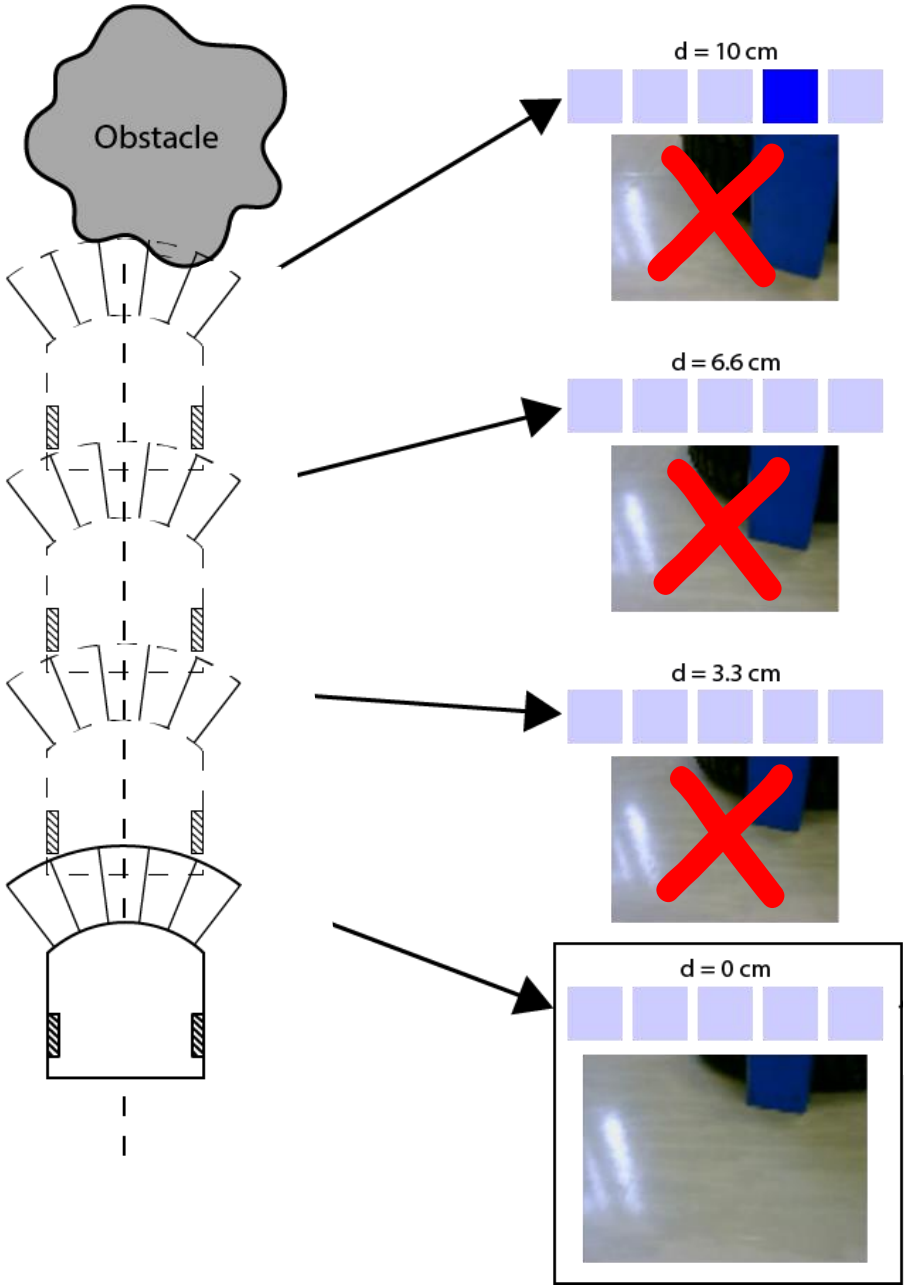

d = 0 cm

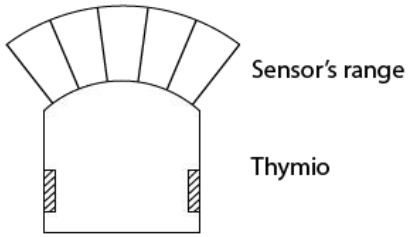

Obstacle
d = 3.3 cm
d = 0 cm


Sensor's range
Thymio


Obstacle
d = 6.6 cm
d = 3.3 cm
d = 0 cm


Sensor's range
Thymio


Obstacle
d = 10 cm
d = 6.6 cm
d = 3.3 cm
d = 0 cm


Sensor's range
Thymio

Data gathering

- Various examples:
  - Different distances and directions
  - Floors with different textures
  - Obstacles with different shapes, materials and colors
- 8 recording sessions
- 36k training examples

An (optional) controller for efficient data gathering

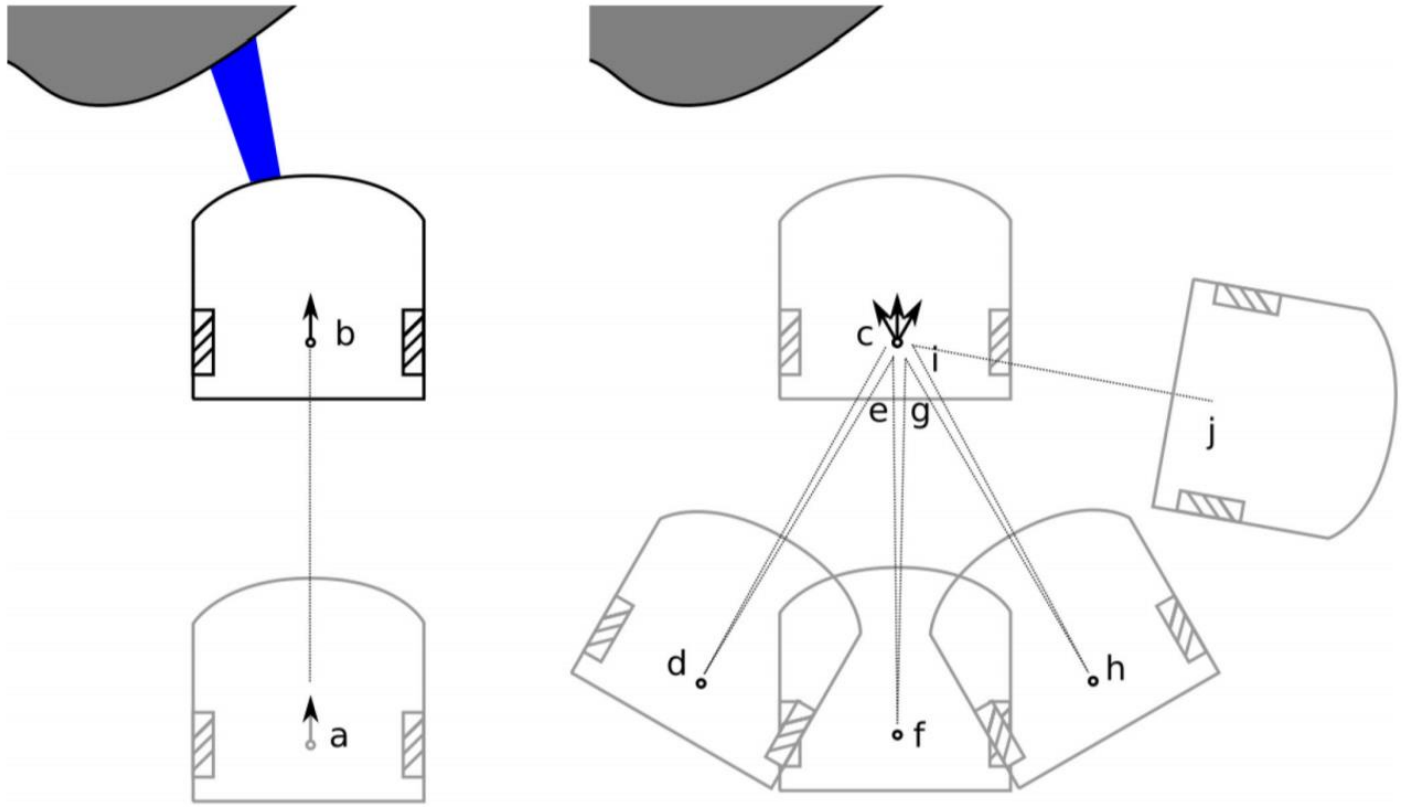
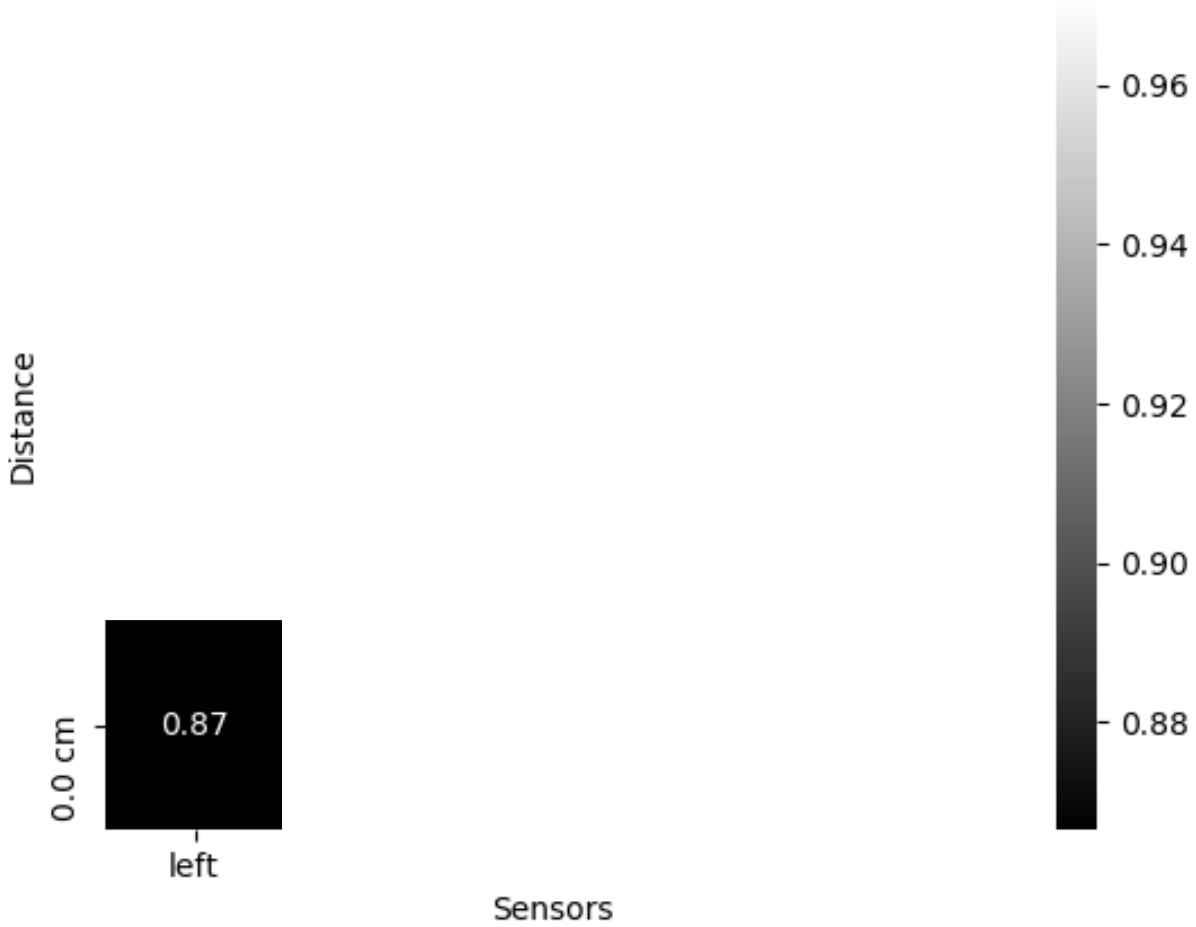


Fig. 4. Example trajectory generated by the data acquisition controller.

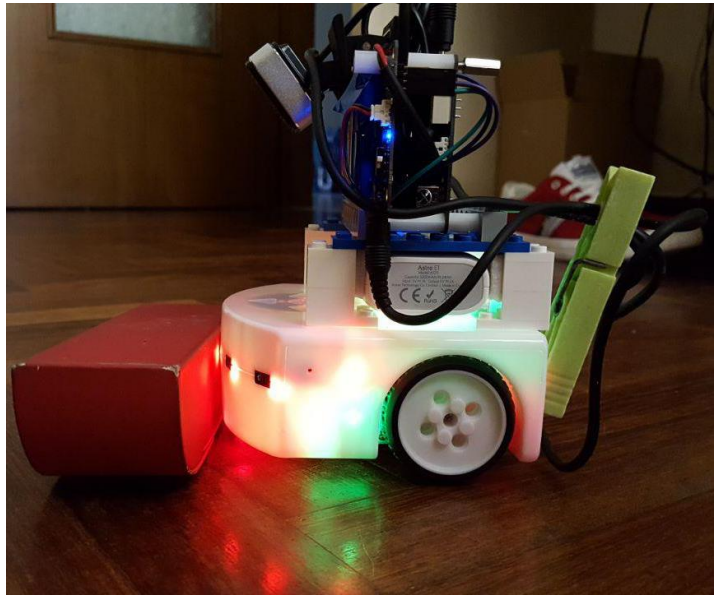
Quantitative evaluation

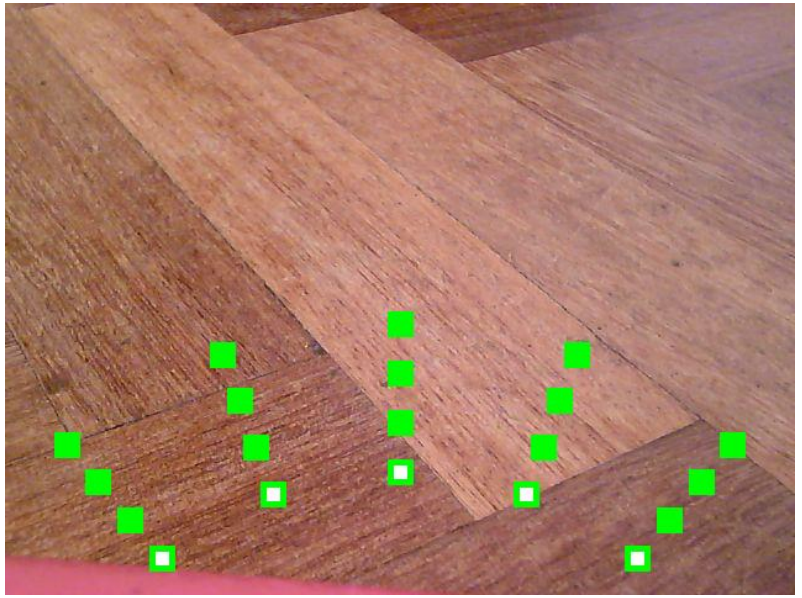
Area Under the Receiver Operating Characteristic Curve

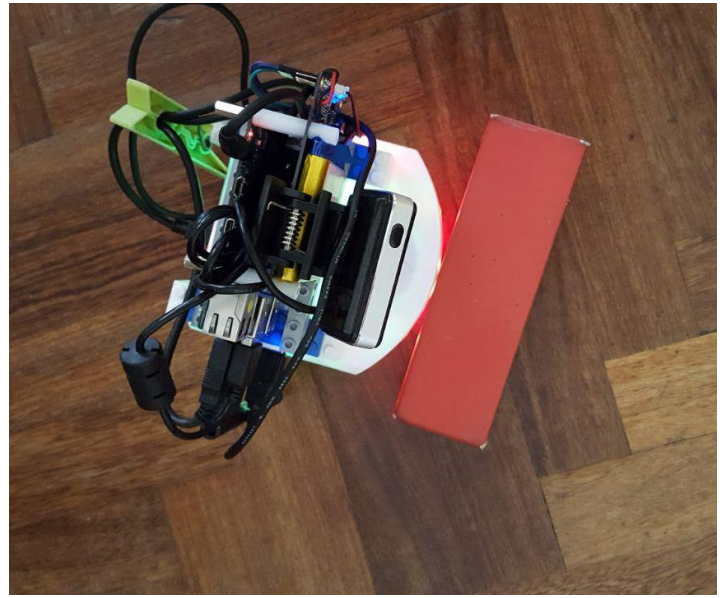


- Symmetric
- Decreases on sides
- Decreases with distance
- Distance = 0 cm is the hardest

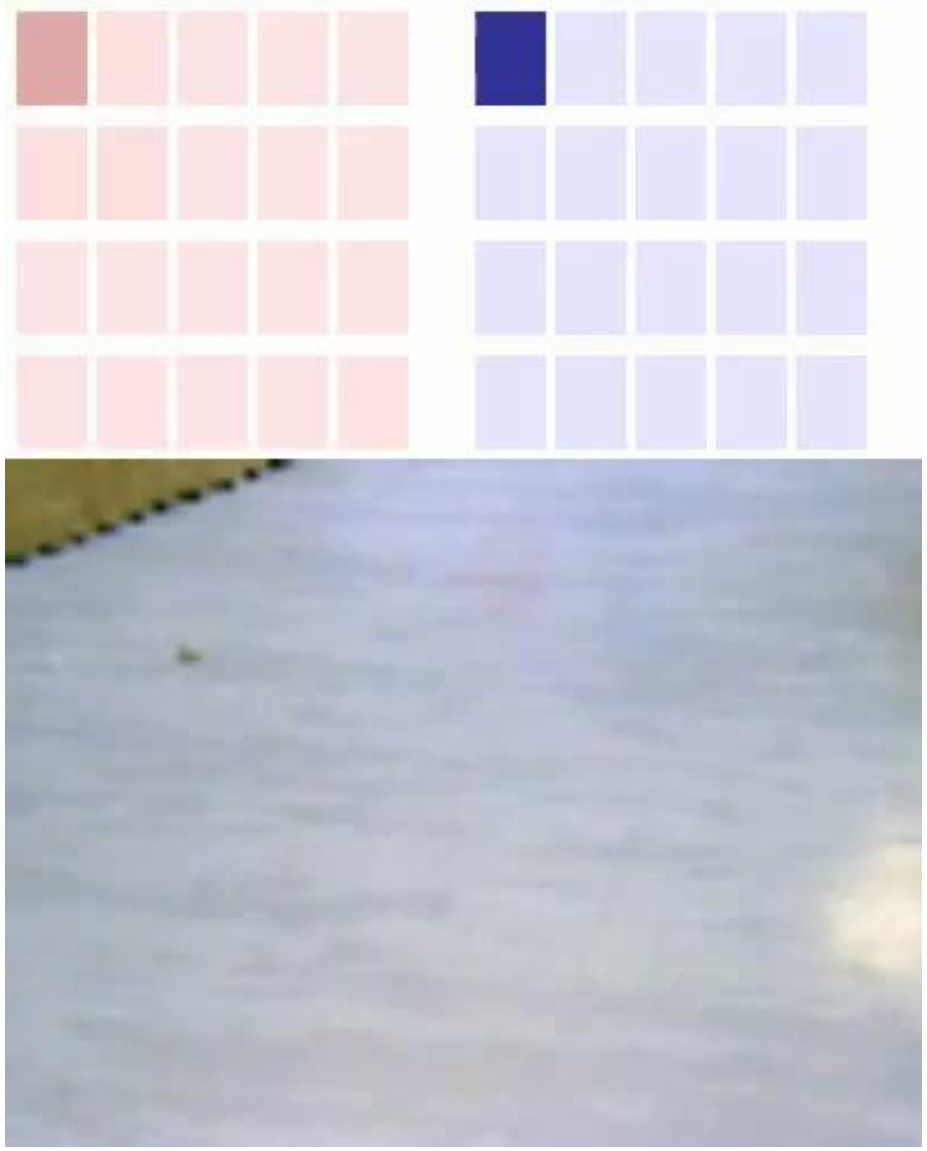
Why 0cm is so hard? the camera blind spot!



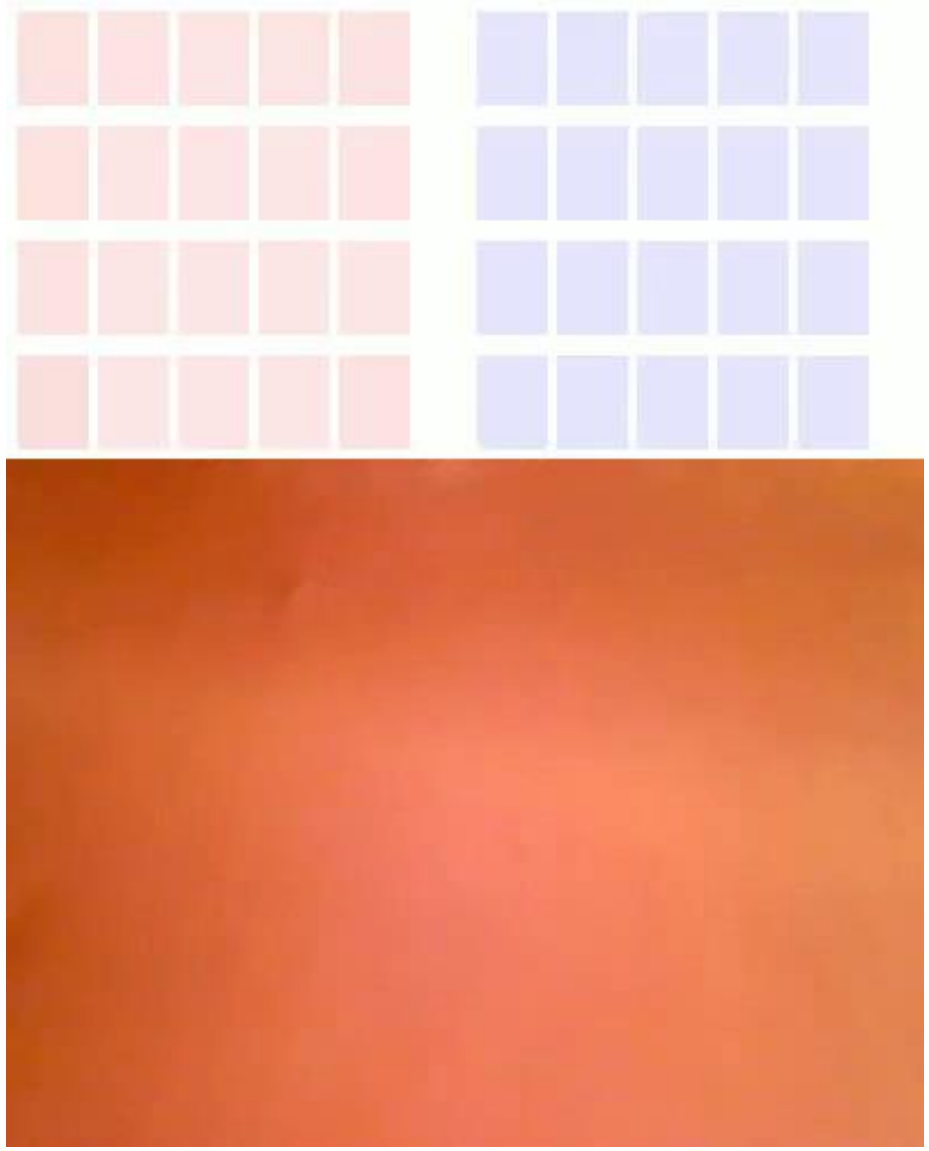




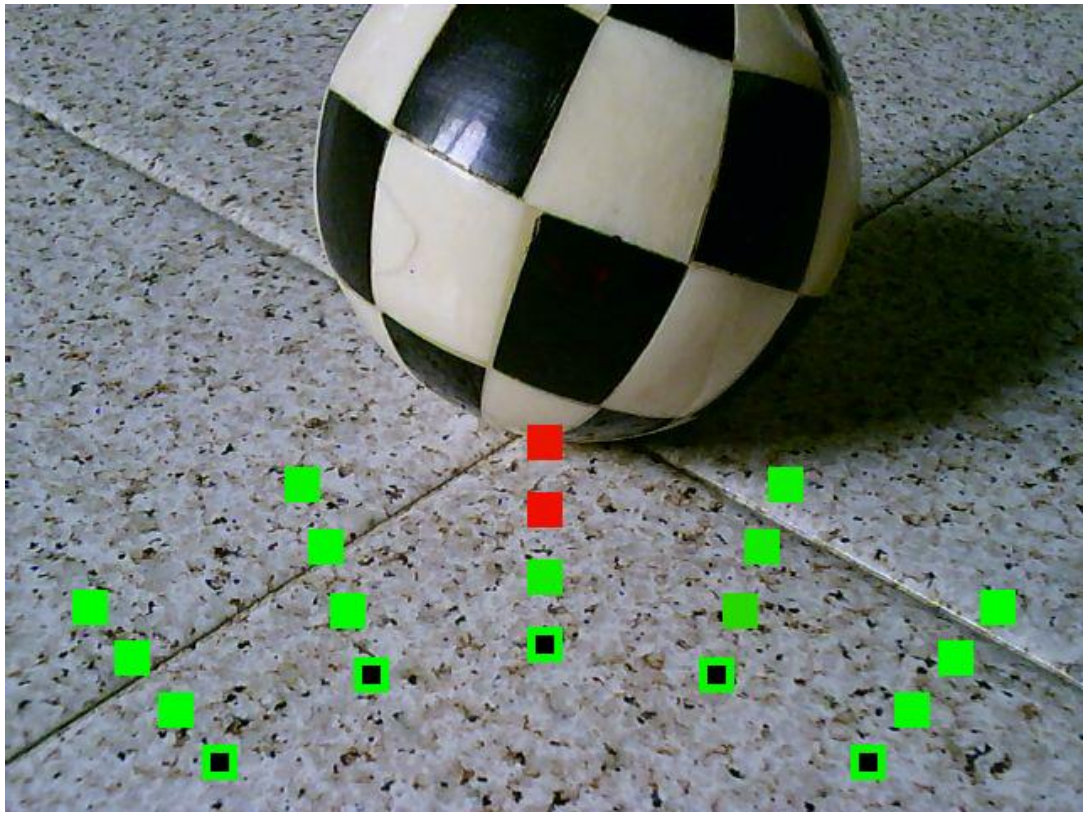
Prediction Target

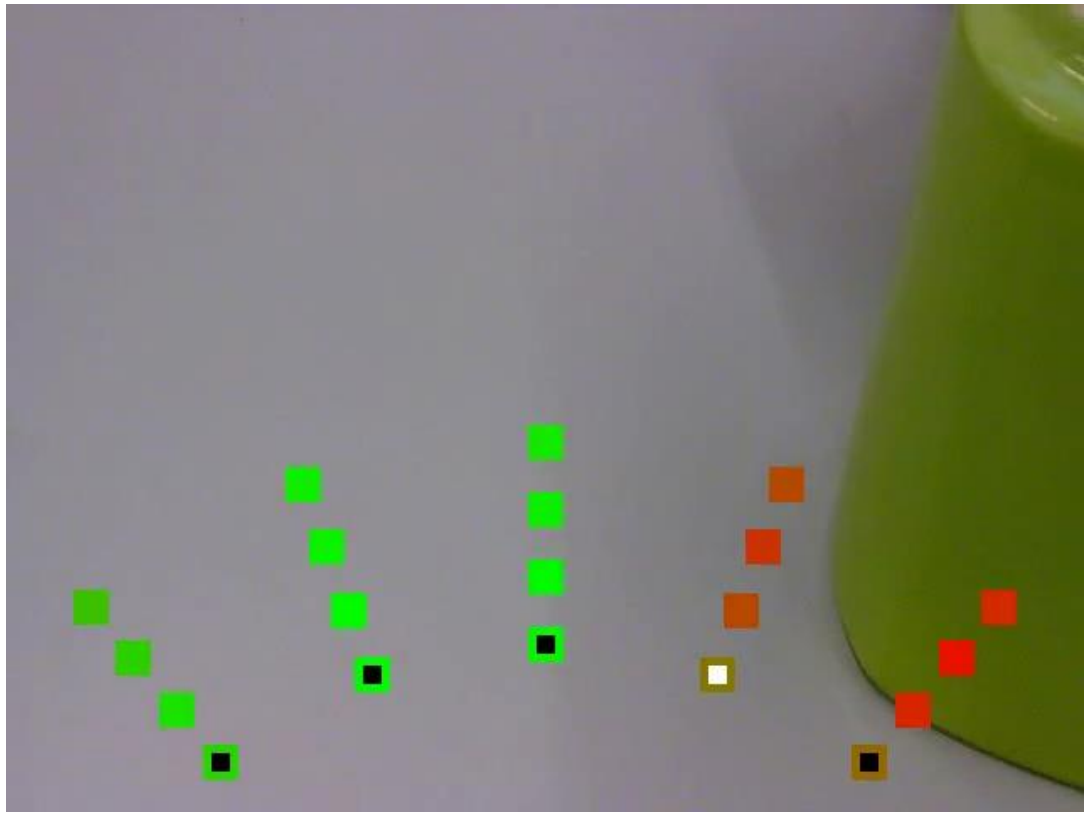


Prediction Target

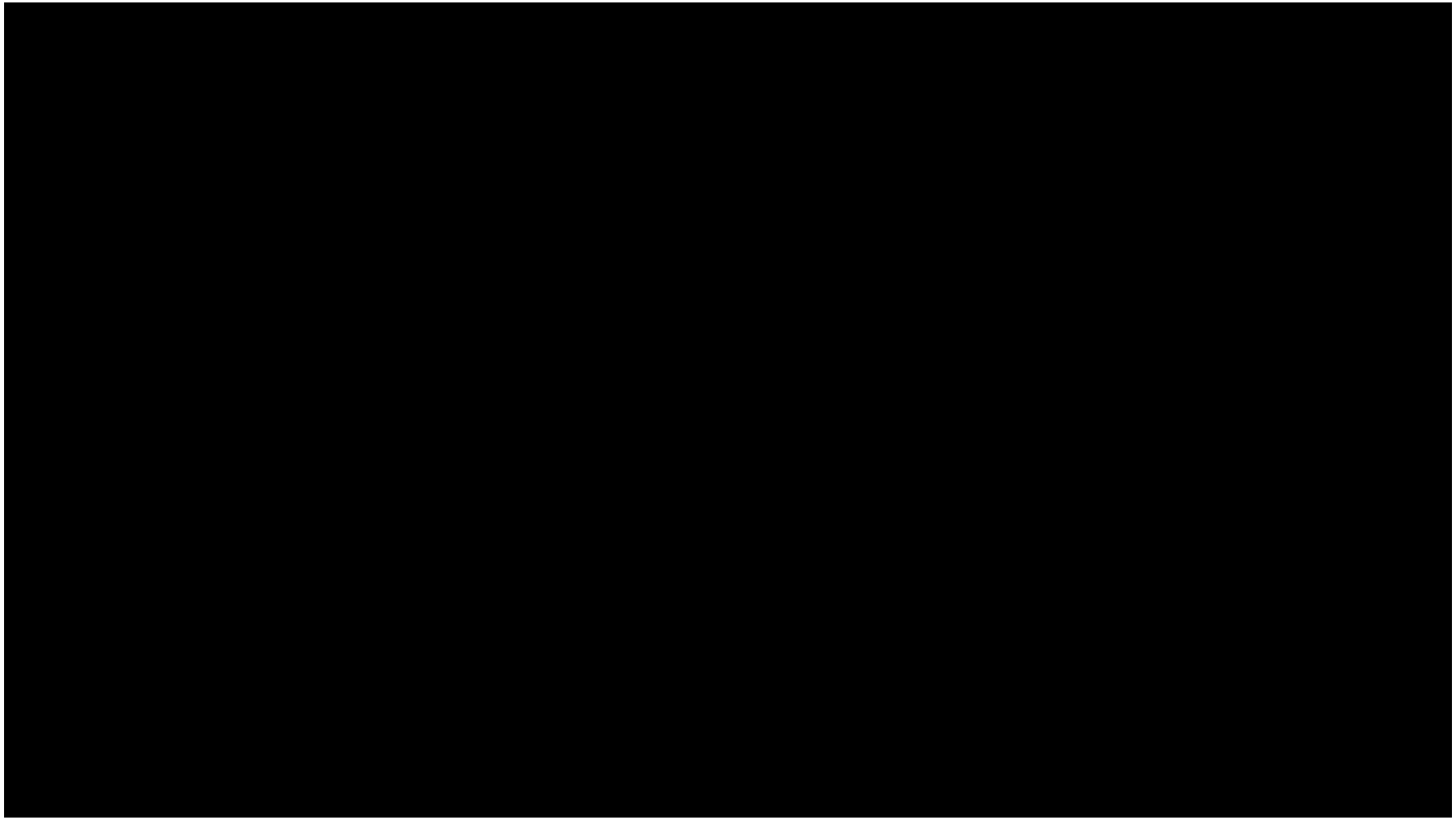


It works!





Video



Video

Generalizing…

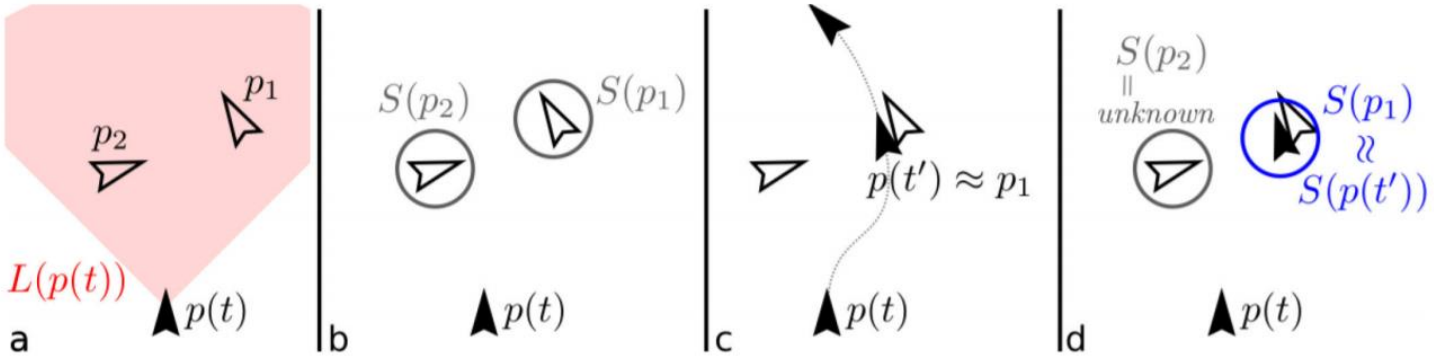


Fig. 2. (a) A mobile robot at pose $p(t)$ has a long-range sensor L (red) and (b) a short-range sensor S. Our objective is to predict the value of S at n target poses $p_1, p_2, \ldots p_n$ from the value of $L(p(t))$. (c, d) For a given instance, we generate ground truth for a subset of labels by searching the robot's future trajectory for poses close to the target poses.

Learning Long-Range Perception Using Self-Supervision From Short-Range Sensors and Odometry

Mirko Nava, Jérôme Guzzi, R. Omar Chavez-Garcia, Luca M. Gambardella, and Alessandro Giusti

***Abstract*—We introduce a general self-supervised approach to predict the future outputs of a short-range sensor (such as a proximity sensor) given the current outputs of a long-range sensor (such as a camera). We assume that the former is directly related to some piece of information to be perceived (such as the presence of an obstacle in a given position), whereas the latter is information rich but hard to interpret directly. We instantiate and implement the approach on a small mobile robot to detect obstacles at various distances using the video stream of the robot's forward-pointing camera, by training a convolutional neural network on automatically-acquired datasets. We quantitatively evaluate the quality of the predictions on unseen scenarios, qualitatively evaluate robustness to different operating conditions, and demonstrate usage as the sole input of an obstacle-avoidance controller. We additionally instantiate the approach on a different simulated scenario with complementary characteristics, to exemplify the generality of our contribution.**

***Index Terms*—Range sensing, computer vision for other robotic applications, deep learning in robotics and automation.**

Fig. 1. The Mighty Thymio robot in two environments; five proximity sensors can easily detect obstacles at very close range (blue areas), whereas the camera has a much longer range (red area) but its outputs are hard to interpret.

VIDEOS, DATASETS, AND CODE

Videos, datasets, and code to reproduce our results are available at: https://github.com/idsia-robotics/learning-long-range-perception/

I. INTRODUCTION

WE CONSIDER a mobile robot capable of odometry and equipped with at least two sensors: a long-range one, such as a camera or laser scanner; and a short-range sensor such as a proximity sensor or a contact sensor (bumper). We then consider a specific perception task, such as detecting obstacles while roaming the environment. Regardless on the specific choice of the task and sensors, it is often the case that the long-range sensors produce a large amount of data, whose interpretation for the task at hand is complex; conversely, the short-range sensor readings directly solve the task, but with limited range. For example, detecting obstacles in the video stream of a forward-pointing camera is difficult but potentially allows us to detect them while they are still far; solving the same task with a proximity sensor or bumper is straightforward as the sensor directly reports the presence of an obstacle, but only works at very close range.

In this letter we propose a novel technique for solving a perception task by learning to interpret the long-range sensor data; in particular, we adopt a self-supervised learning approach in which future outputs from the short-range sensor are used as a supervisory signal. We develop the complete pipeline for an obstacle-detection task using camera frames as the long-range sensor and proximity sensor readings as the short-range sensor (see Figure 1). In this context, the camera frame acquired at time t (input) is associated to proximity sensor readings obtained at a different time $t' \neq t$ (labels); for example, if the robot's odometry detects it has advanced straight for 10 cm between t and t', the proximity sensor outputs at t' correspond to the presence of obstacles 10 cm in front of the pose of the robot at t. These outputs at time t' can be associated to the camera frame acquired at time t as a label expressing the presence of an obstacle 10cm ahead. The same reasoning can be applied to other distances, so that we define a multi-label classification problem with a single camera frame as input, and multiple binary labels expressing the presence of obstacles at different distances.

The approach is *self-supervised* because it does not require any explicit effort for dataset acquisition or labeling: the robot acquires labeled datasets unattended and can gather additional

Manuscript received September 10, 2018; accepted January 15, 2019. Date of publication January 23, 2019; date of current version February 15, 2019. This letter was recommended for publication by Associate Editor L. Paull and Editor C. Stachniss upon evaluation of the reviewers' comments. This work was supported by the Swiss National Science Foundation through the NCCR Robotics. *(Corresponding author: Mirko Nava.)*

The authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano 6928, Switzerland (e-mail: mirko@idsia.ch; jerome@idsia.ch; omar@idsia.ch; luca@idsia.ch; alessandrog@idsia.ch).

Digital Object Identifier 10.1109/LRA.2019.2894849

https://github.com/idsia-robotics/learning-long-range-perception

1b. A seminal paper from 2006

A robot that acquires its own training data...
but then solves a standard Supervised Learning problem

Self-supervised online learning for big-ass Robots

Improving Robot Navigation through Self-Supervised Online Learning

Boris Sofman, Ellie Lin, J. Andrew Bagnell, John Cole, Nicolas Vandapel, and Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
e-mail: bsofman@ri.cmu.edu, elliel@ri.cmu.edu, dbagnell@ri.cmu.edu, jgcole@ri.cmu.edu, vandapel@ri.cmu.edu, axs@ri.cmu.edu

Received 8 April 2006; accepted 1 November 2006

In mobile robotics, there are often features that, while potentially powerful for improving navigation, prove difficult to profit from as they generalize poorly to novel situations. Overhead imagery data, for instance, have the potential to greatly enhance autonomous robot navigation in complex outdoor environments. In practice, reliable and effective automated interpretation of imagery from diverse terrain, environmental conditions, and sensor varieties proves challenging. Similarly, fixed techniques that successfully interpret on-board sensor data across many environments begin to fail past short ranges as the density and accuracy necessary for such computation quickly degrade and features that are able to be computed from distant data are very domain specific. We introduce an online, probabilistic model to effectively learn to use these scope-limited features by leveraging other features that, while perhaps otherwise more limited, generalize reliably. We apply our approach to provide an efficient, self-supervised learning method that accurately predicts traversal costs over large areas from overhead data. We present results from field testing on-board a robot operating over large distances in various off-road environments. Additionally, we show how our algorithm can be used offline with overhead data to produce a priori traversal cost maps and detect misalignments between overhead data and estimated vehicle positions. This approach can significantly improve the versatility of many unmanned ground vehicles by allowing them to traverse highly varied terrains with increased performance.

1. INTRODUCTION

Autonomous robot navigation in unstructured natural environments has been demonstrated extensively in a large variety of terrain, sensor payload, and mission scenarios [see for example Kelly et al. (2006), Bodta and Camden (2004), and Goldberg, Maimone & Matthies (2002)]. Even though powerful at sensing, modeling, and interpreting the environment, these systems required significant tuning of parameters, either by hand or supervised training, to best adjust their algorithms to the local environment where the tests are conducted.

Contract grant sponsor: Defense Advanced Research Projects Agency (DARPA).
Contract grant number: MDA972-01-9-0005.

Journal of Field Robotics 23(11/12), 1059–1075 (2006) © 2007 Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20169

The task

Predict the traversal cost of terrain given overhead data

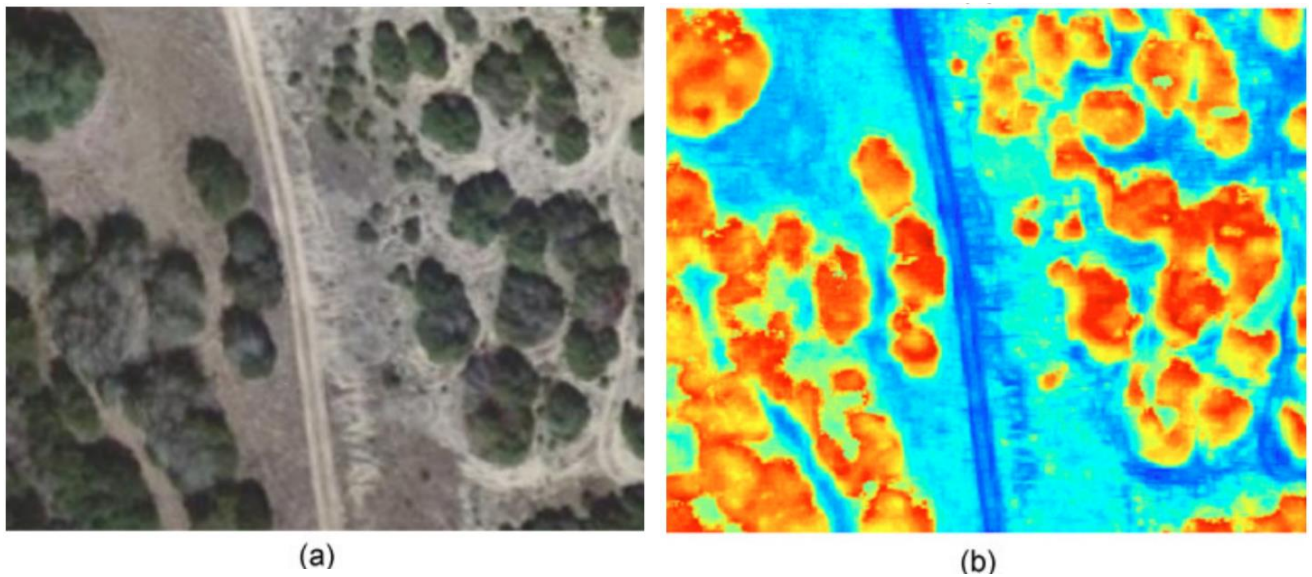


Figure 2. Sample results of terrain traversal cost predictions. (a) 0.35 m resolution color overhead imagery used by our online learning algorithm and (b) corresponding predictions of terrain traversal costs. Traversal costs are color-scaled for improved visibility. Blue and red correspond to lowest and highest traversal cost estimates, respectively.

Supervision

- Short range ladar
- Robot assigns traversal costs to areas in front of itself from features computed by interpreting the position, density, and point cloud distributions of sensed obstacles

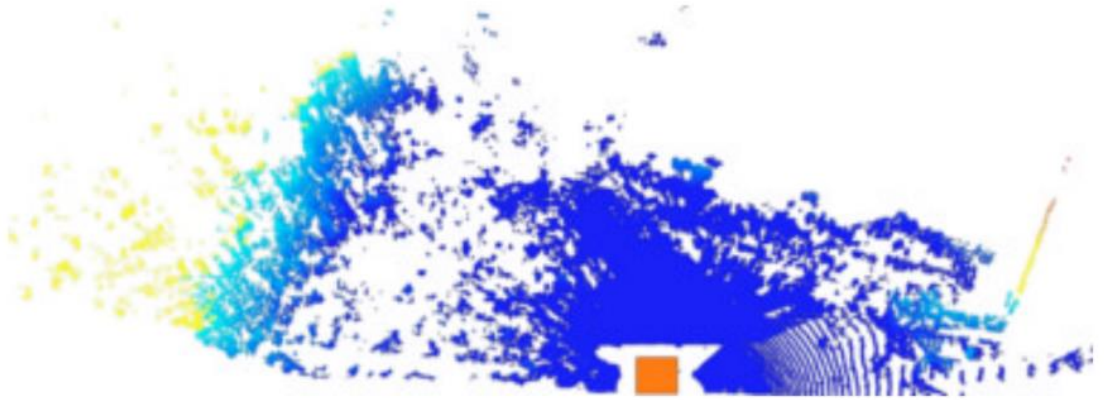
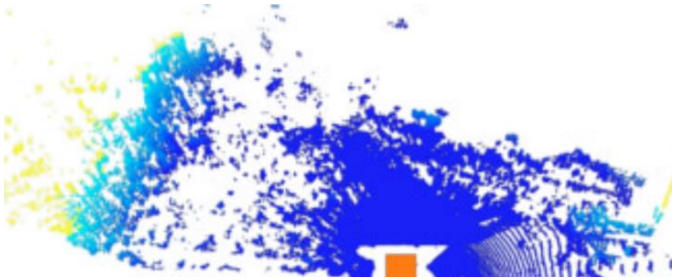
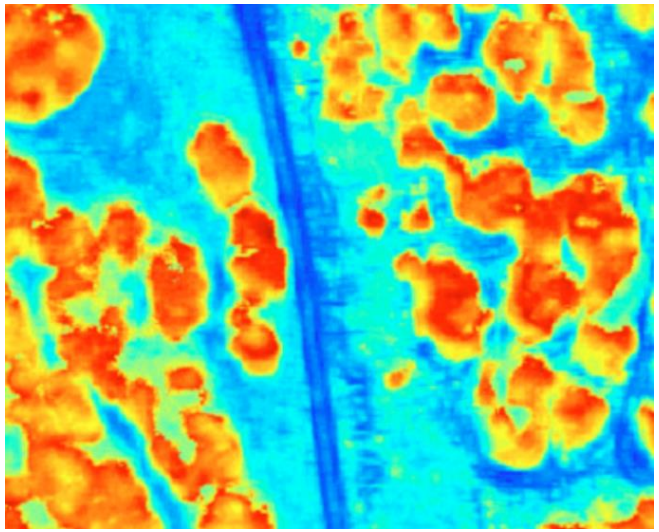
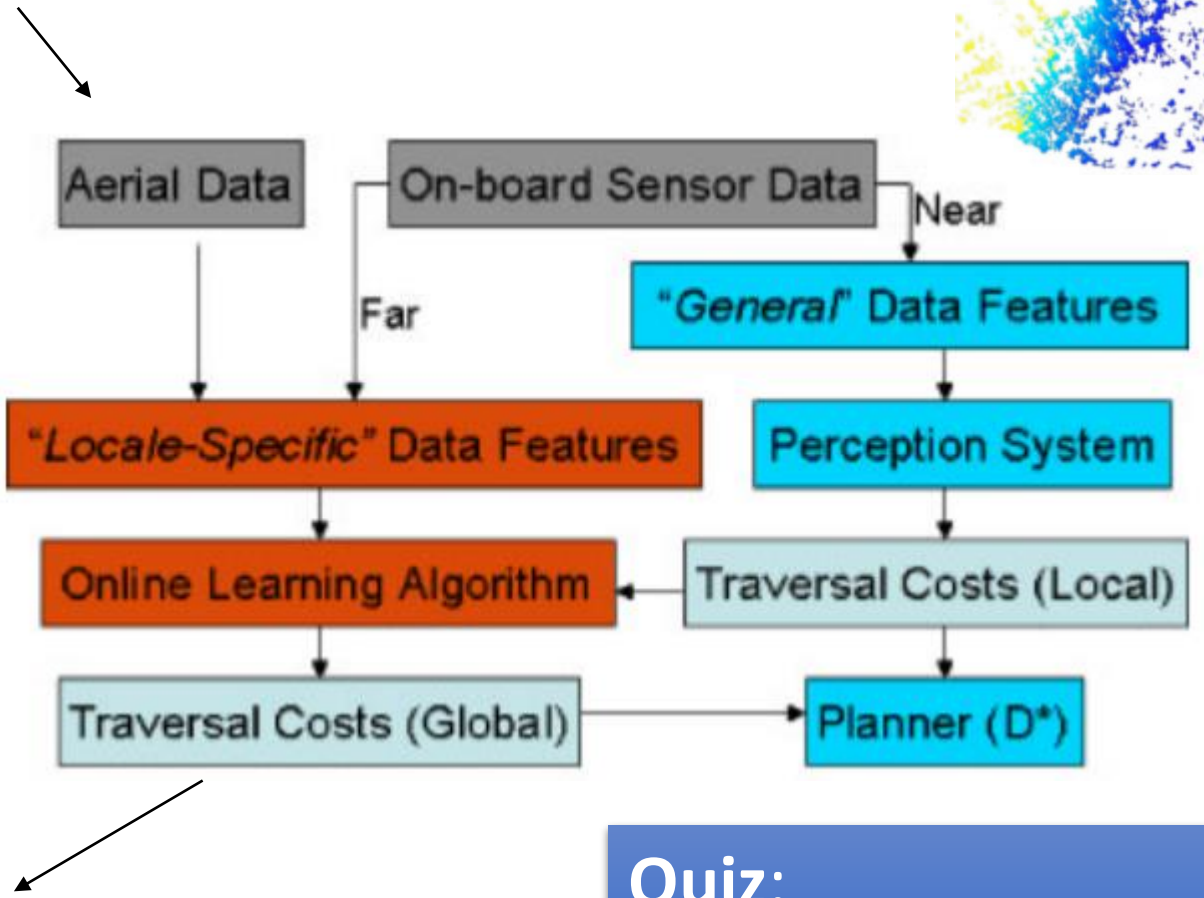


Figure 1. Typical ladar response from vehicle's perception system. Ladar points are color coded by elevation with lowest points appearing in blue and highest points appearing in yellow. Vehicle position is shown by the orange square. Notice the large drop in ladar response density (especially on the ground) as distance from the vehicle increases. Large objects such as the trees on the left generate ladar responses even at far ranges but are difficult to interpret through fixed techniques across different environments.



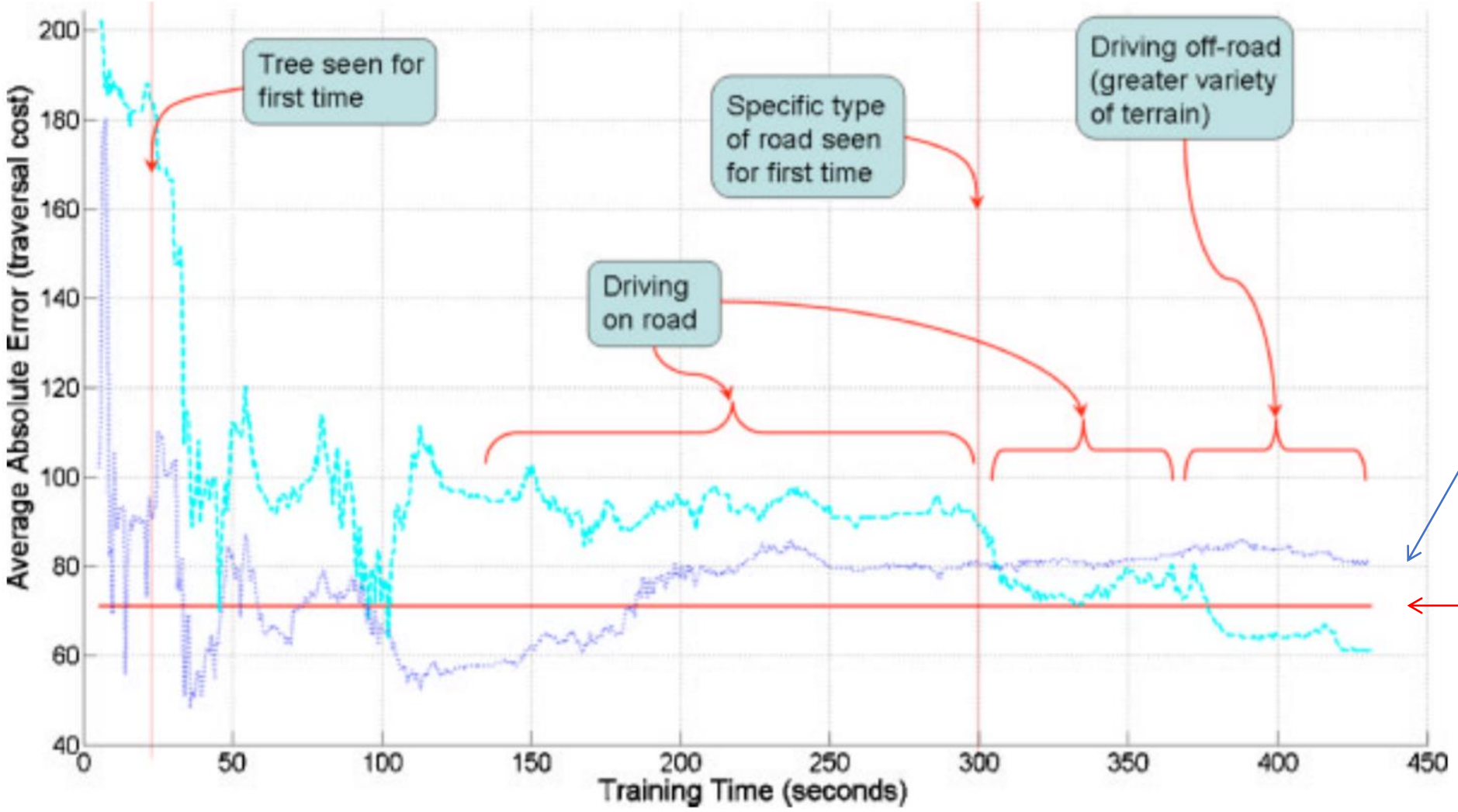




This!

Quiz:
What would you call "supervisory signal" here?

A big advantage: online learning



Performance of self-supervised models

Performance of a supervised system trained from human collected data

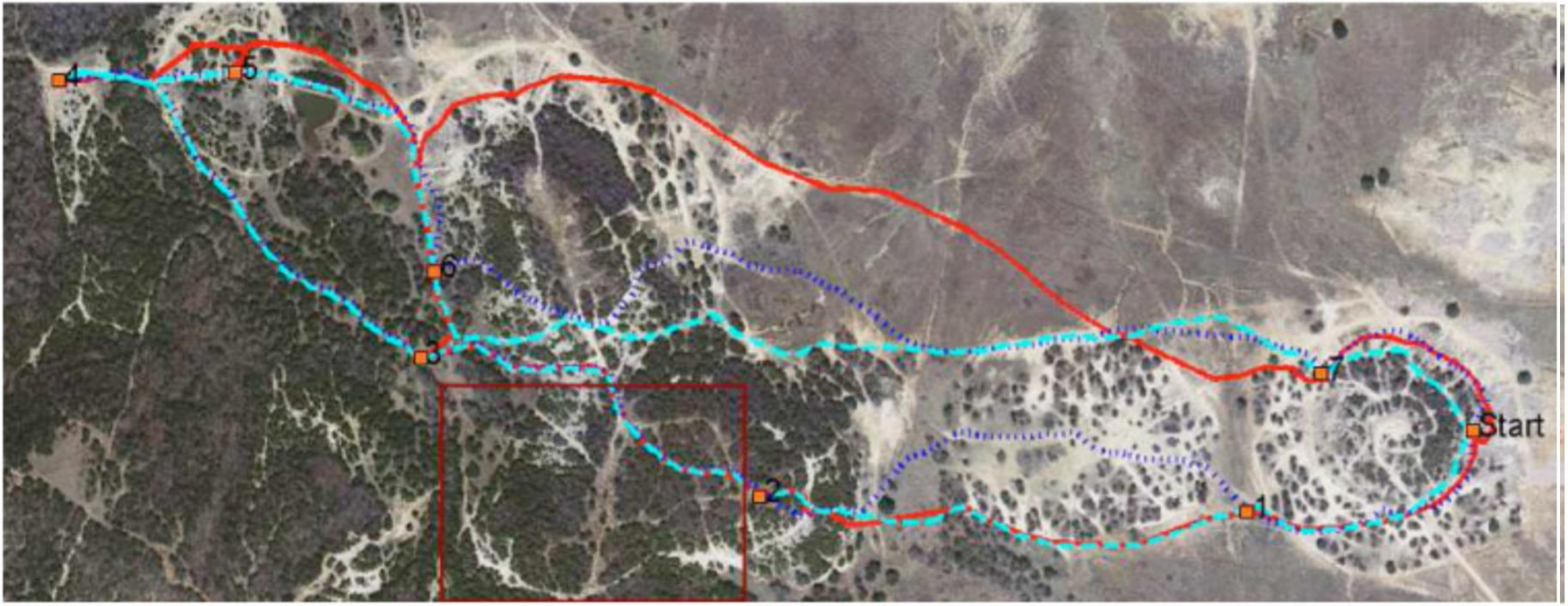
How do you evaluate something like this?

Figure 7. Comparison of paths executed by our robot for shown course when using only on-board perception (in solid red) and with OOLL (in dashed blue) and FROLL (in dotted cyan) used in real-time on-board the robot. Course started at the top right and ended at the bottom left.

Table I. Statistics for course traversals with and without online learning algorithm

	Without algorithm	With OOLL
Total Traversal time (s)	1369.86	1000.82
Total distance traveled (m)	1815.71	1681.73
Average speed (m/s)	1.33	1.68
No. of interventions	1	0

How do you evaluate something like this?



Two main meanings for SSL

- Systems (typically robots) that collect their own training data but then solve a standard supervised learning task
- Systems that learn to extract meaningful representations from the data itself

Two main meanings for SSL

- Systems (typically robots) that collect their own training data but then solve a standard supervised learning task
- Systems that learn to extract meaningful representations from the data itself

2. Self-supervised (aka self-taught) deep learning

The data itself is a source of supervision

Comprehensive source: Slides from Andrew Zisserman, 2018
https://project.inria.fr/paiss/files/2018/07/zisserman-self-supervised.pdf

Shades of supervision: full supervision

To some extent, any visual task can be solved now by:

1. Construct a large-scale dataset labelled for that task
2. Specify a training loss and neural network architecture
3. Train the network and deploy

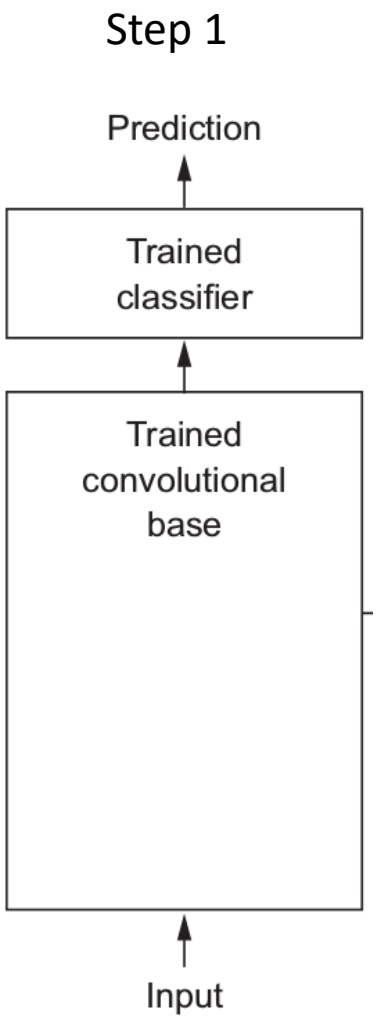
Classification error on imagenet

But...

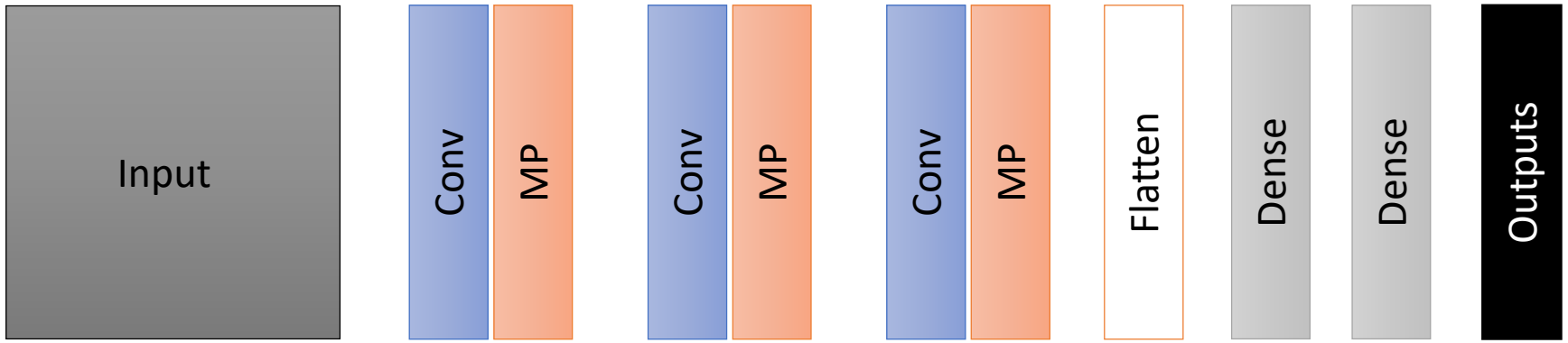
- Labeled data is expensive (eg medical, or whatever problem they are paying you to solve)
- **Huge** amounts of unlabeled data
  - Facebook: one billion images uploaded per day
  - 300 hours of video are uploaded to YouTube every minute

→ we want to exploit unlabeled data, at least in part

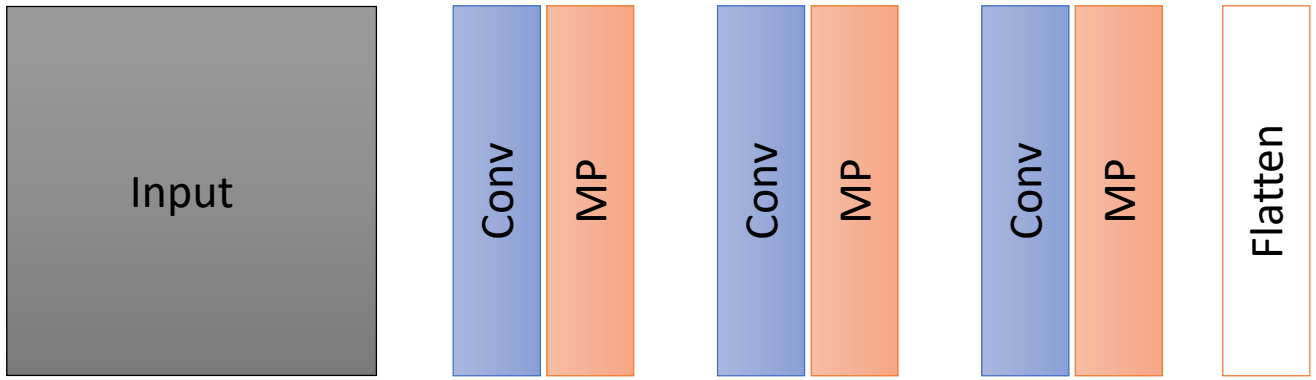
Using pretrained weights



Option 1



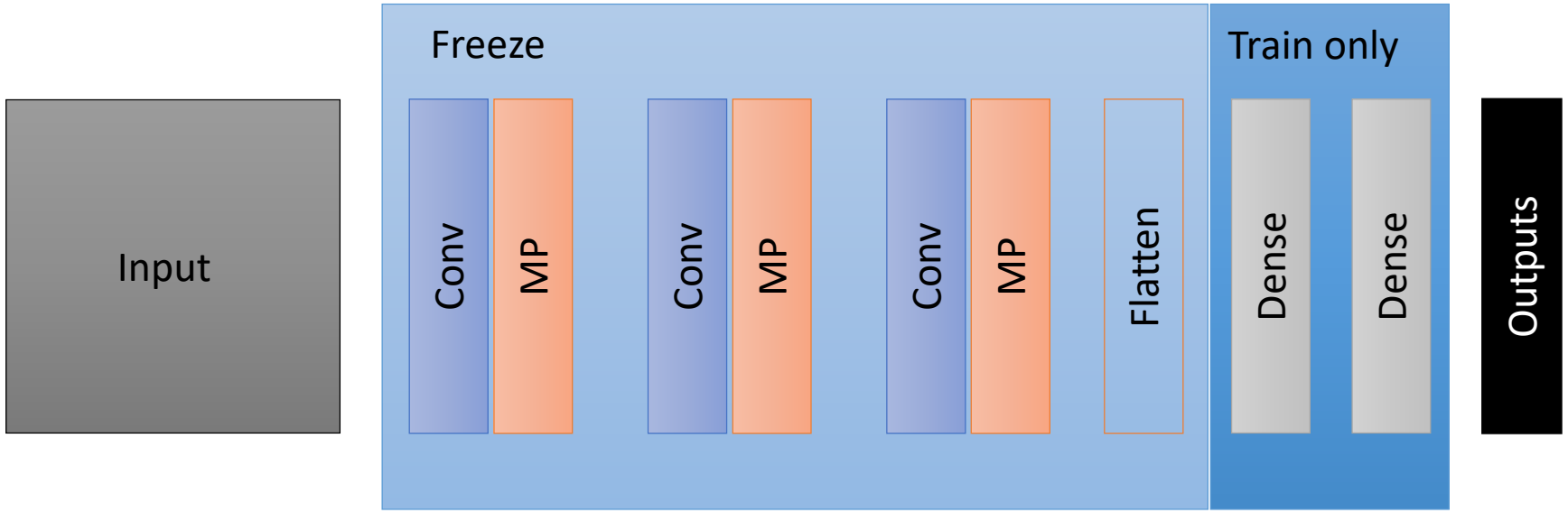
Option 1



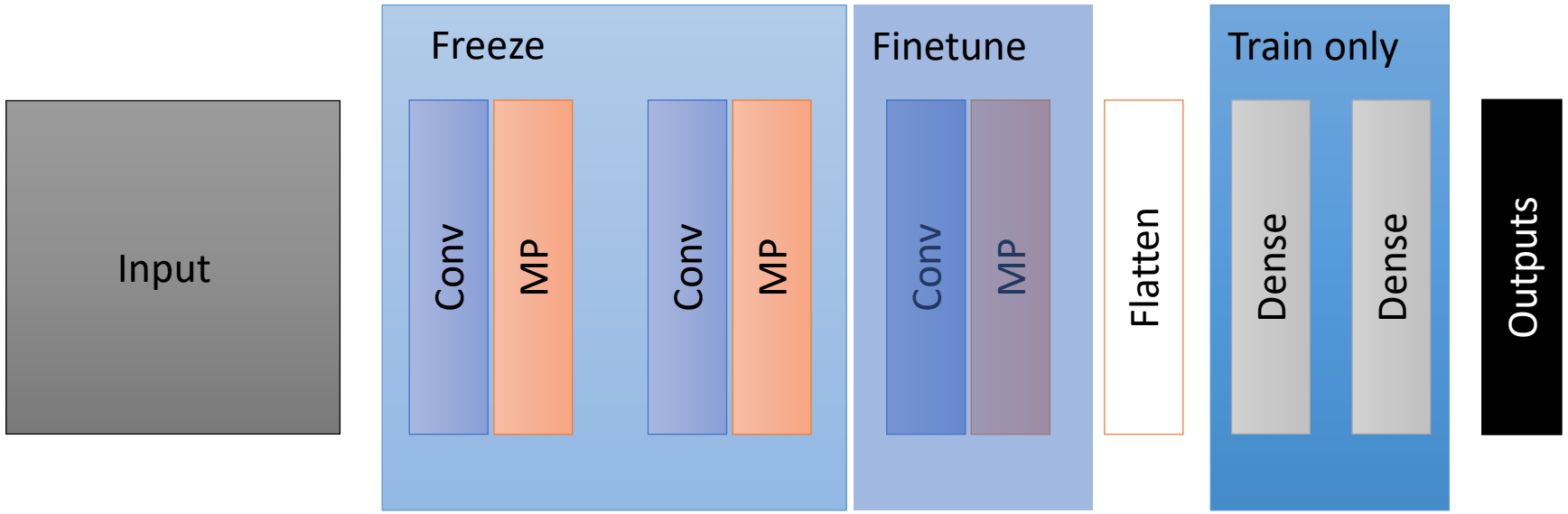
Save these features for the whole training and testing datasets.

Then, train a new classifier that uses these features as input

Option 2


Input
Freeze
Conv
MP
Conv
MP
Conv
MP
Flatten
Train only
Dense
Dense
Outputs

Option 3


Input
Freeze
Conv
MP
Conv
MP
Finetune
Conv
MP
Flatten
Train only
Dense
Dense
Outputs

Shades of supervision: self-supervised learning

Can we learn something WITHOUT labels?

How do we (humans) learn?!?

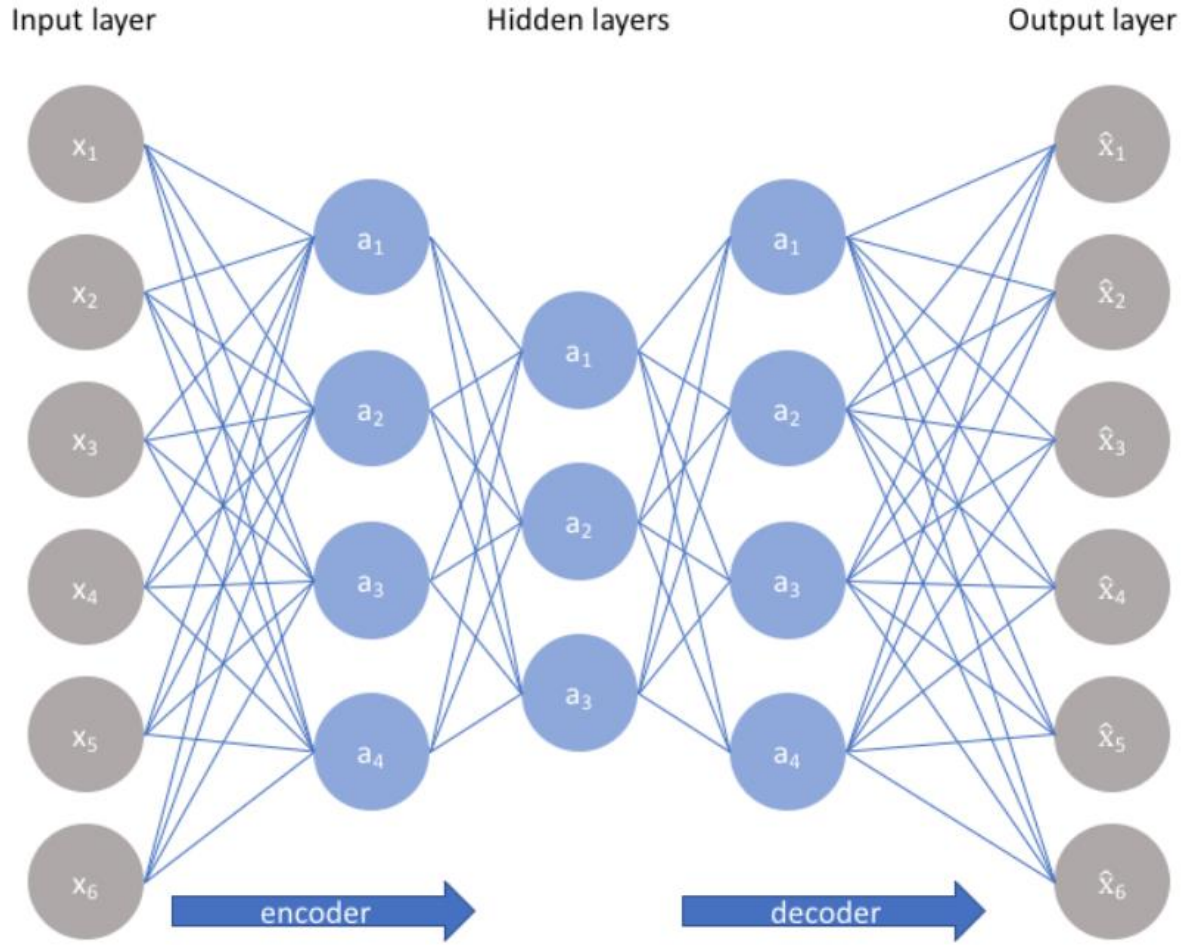
The Scientist in the Crib: What Early Learning Tells Us About the Mind
by Alison Gopnik, Andrew N. Meltzoff and Patricia K. Kuhl
The Development of Embodied Cognition: Six Lessons from Babies
by Linda Smith and Michael Gasser

Definition (attempt to)

- You are interested in solving problem A
- Take a lot of data similar to the one you'll use, without labels
  (of course: you are lazy)
- Invent a problem B (*pretext task*) on the data for which
  - you can get a ground truth for free from the data itself
  - you need to "**understand**" the data in order to solve it
- Train a network for B

→ The network has learned something valuable for A, i.e. to understand the data

You already know at least one method to achieve this: **autoencoders**



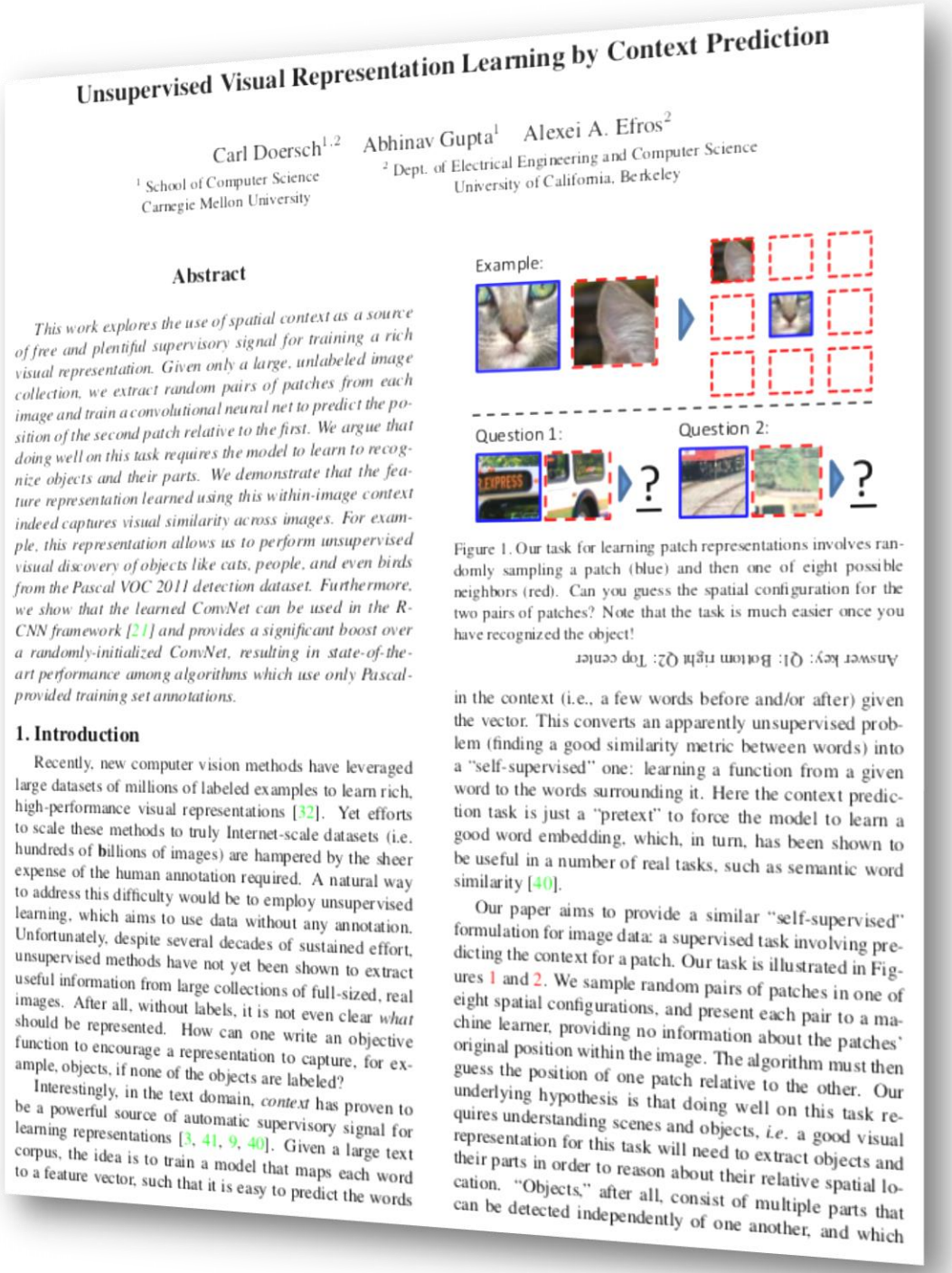
Pretext task desiderata:

- you can get a ground truth for free from the data itself
- you need to "**understand**" the data in order to solve it

https://www.jeremyjordan.me/autoencoders/

Unsupervised Visual Representation Learning by Context Prediction

https://arxiv.org/abs/1505.05192, 2015



Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch[1,2] Abhinav Gupta[1] Alexei A. Efros[2]

[1] School of Computer Science
Carnegie Mellon University

[2] Dept. of Electrical Engineering and Computer Science
University of California, Berkeley

Abstract

This work explores the use of spatial context as a source of free and plentiful supervisory signal for training a rich visual representation. Given only a large, unlabeled image collection, we extract random pairs of patches from each image and train a convolutional neural net to predict the position of the second patch relative to the first. We argue that doing well on this task requires the model to learn to recognize objects and their parts. We demonstrate that the feature representation learned using this within-image context indeed captures visual similarity across images. For example, this representation allows us to perform unsupervised visual discovery of objects like cats, people, and even birds from the Pascal VOC 2011 detection dataset. Furthermore, we show that the learned ConvNet can be used in the R-CNN framework [21] and provides a significant boost over a randomly-initialized ConvNet, resulting in state-of-the-art performance among algorithms which use only Pascal-provided training set annotations.

1. Introduction

Recently, new computer vision methods have leveraged large datasets of millions of labeled examples to learn rich, high-performance visual representations [32]. Yet efforts to scale these methods to truly Internet-scale datasets (i.e. hundreds of billions of images) are hampered by the sheer expense of the human annotation required. A natural way to address this difficulty would be to employ unsupervised learning, which aims to use data without any annotation. Unfortunately, despite several decades of sustained effort, unsupervised methods have not yet been shown to extract useful information from large collections of full-sized, real images. After all, without labels, it is not even clear *what* should be represented. How can one write an objective function to encourage a representation to capture, for example, objects, if none of the objects are labeled?

Interestingly, in the text domain, *context* has proven to be a powerful source of automatic supervisory signal for learning representations [3, 41, 9, 40]. Given a large text corpus, the idea is to train a model that maps each word to a feature vector, such that it is easy to predict the words

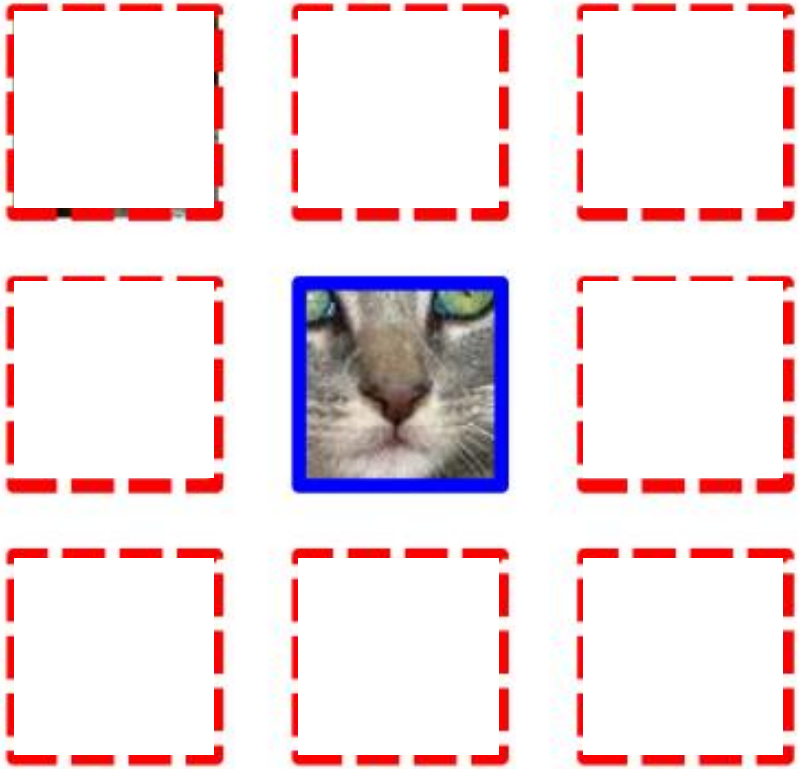
Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center

in the context (i.e., a few words before and/or after) given the vector. This converts an apparently unsupervised problem (finding a good similarity metric between words) into a "self-supervised" one: learning a function from a given word to the words surrounding it. Here the context prediction task is just a "pretext" to force the model to learn a good word embedding, which, in turn, has been shown to be useful in a number of real tasks, such as semantic word similarity [40].

Our paper aims to provide a similar "self-supervised" formulation for image data: a supervised task involving predicting the context for a patch. Our task is illustrated in Figures 1 and 2. We sample random pairs of patches in one of eight spatial configurations, and present each pair to a machine learner, providing no information about the patches' original position within the image. The algorithm must then guess the position of one patch relative to the other. Our underlying hypothesis is that doing well on this task requires understanding scenes and objects, *i.e.* a good visual representation for this task will need to extract objects and their parts in order to reason about their relative spatial location. "Objects," after all, consist of multiple parts that can be detected independently of one another, and which

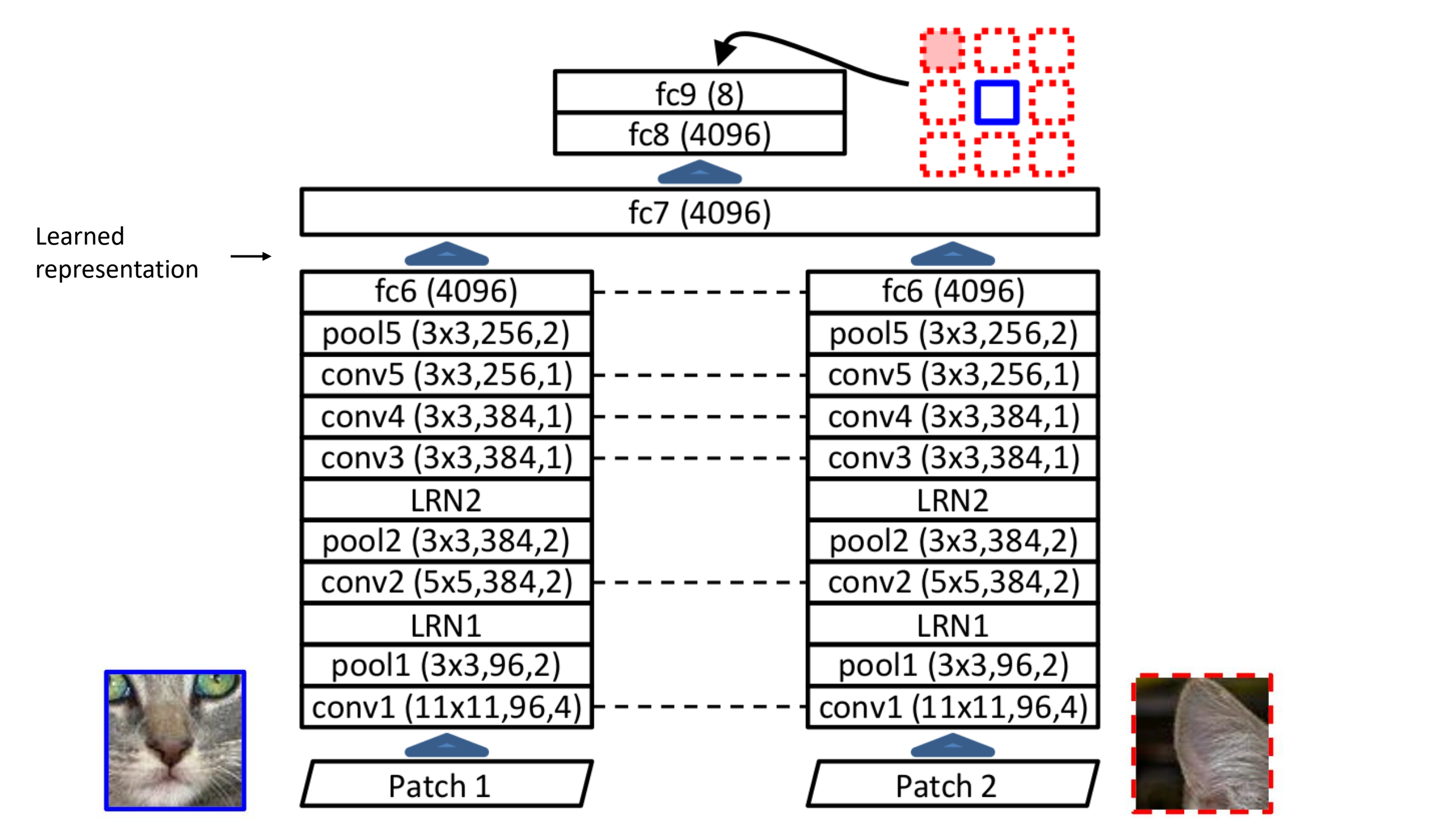
Think!



Some more…

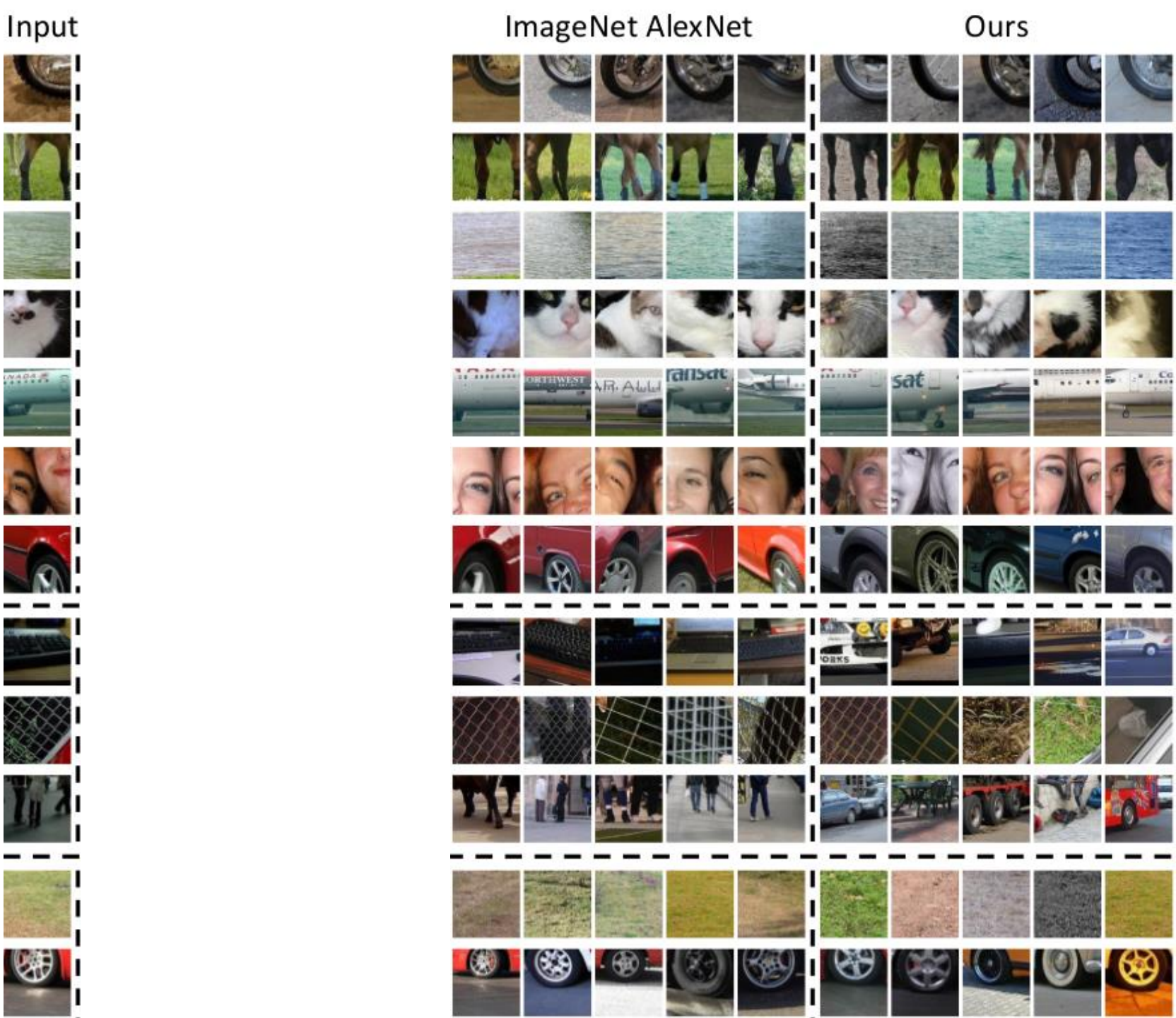
Pretext task desiderata:

- you can get a ground truth for free from the data itself
- you need to "**understand**" the data in order to solve it


fc9 (8)
fc8 (4096)
fc7 (4096)
Learned representation
fc6 (4096)
pool5 (3x3,256,2)
conv5 (3x3,256,1)
conv4 (3x3,384,1)
conv3 (3x3,384,1)
LRN2
pool2 (3x3,384,2)
conv2 (5x5,384,2)
LRN1
pool1 (3x3,96,2)
conv1 (11x11,96,4)
Patch 1
fc6 (4096)
pool5 (3x3,256,2)
conv5 (3x3,256,1)
conv4 (3x3,384,1)
conv3 (3x3,384,1)
LRN2
pool2 (3x3,384,2)
conv2 (5x5,384,2)
LRN1
pool1 (3x3,96,2)
conv1 (11x11,96,4)
Patch 2

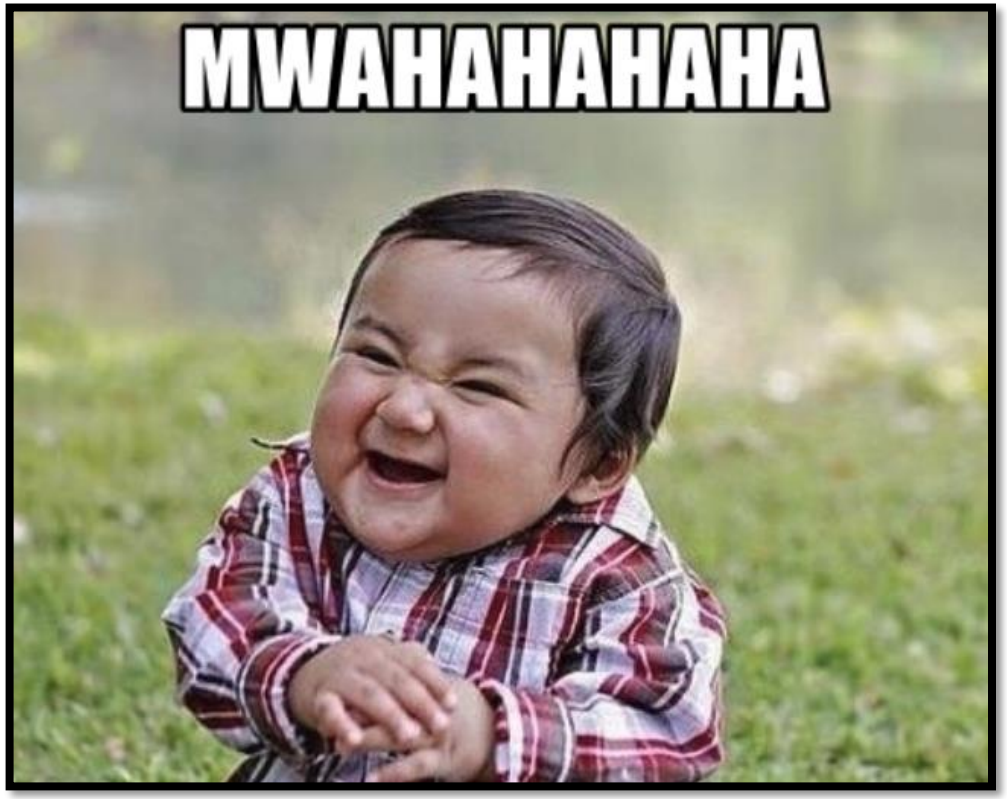
How can we evaluate whether the representation makes sense?

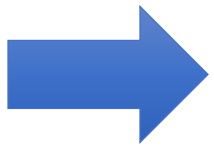
- Given a query patch, we can look for nearest neighbors in the dataset
  - Are these semantically similar?
- It turns out that... Yes, they are
- Surprisingly they also are somewhat similar if the network is randomly initialized (!)



Find the bug

- The network will CHEAT if it can
- When designing a pretext task, care must be taken to ensure that the task forces the network to extract the desired information (high-level semantics, in our case), without taking “trivial” shortcuts.

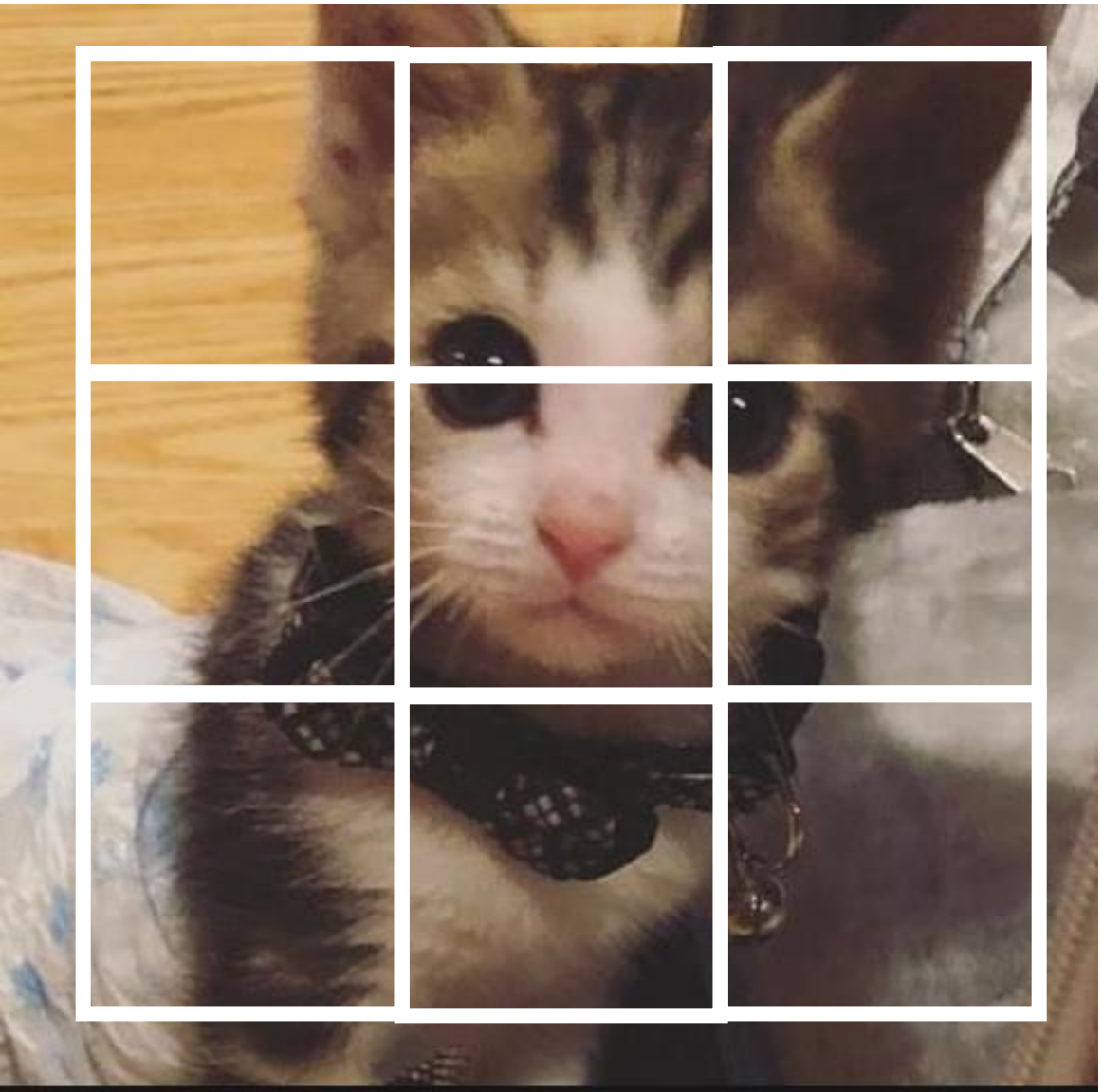




Pretext task desiderata:

- you can get a ground truth for free from the data itself
- you need to “**understand**” the data in order to solve it

Brainstorm: you are a lazy neural network

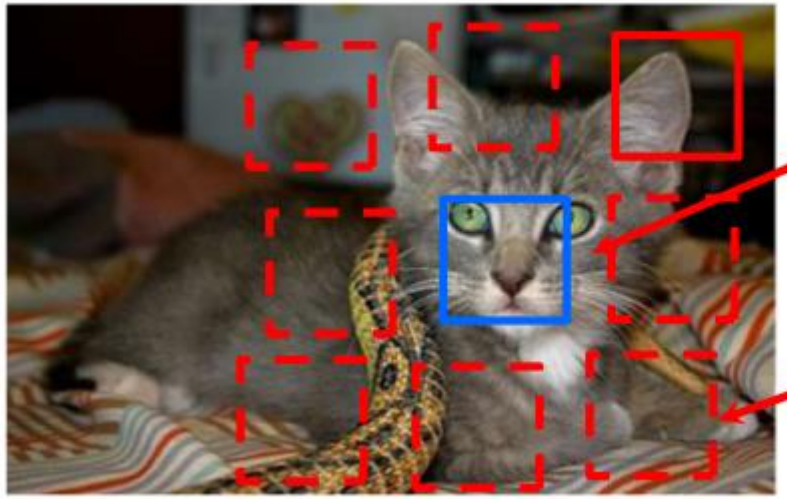


You are a network that, given the center patch and one of the others, has to predict the relative position of the second wrt the first (8 possible classes).

Pitch lazy ways to solve the problem without actually understanding the image!

https://answergarden.ch/1278976

Anti-cheat 1 and 2!



Include a gap

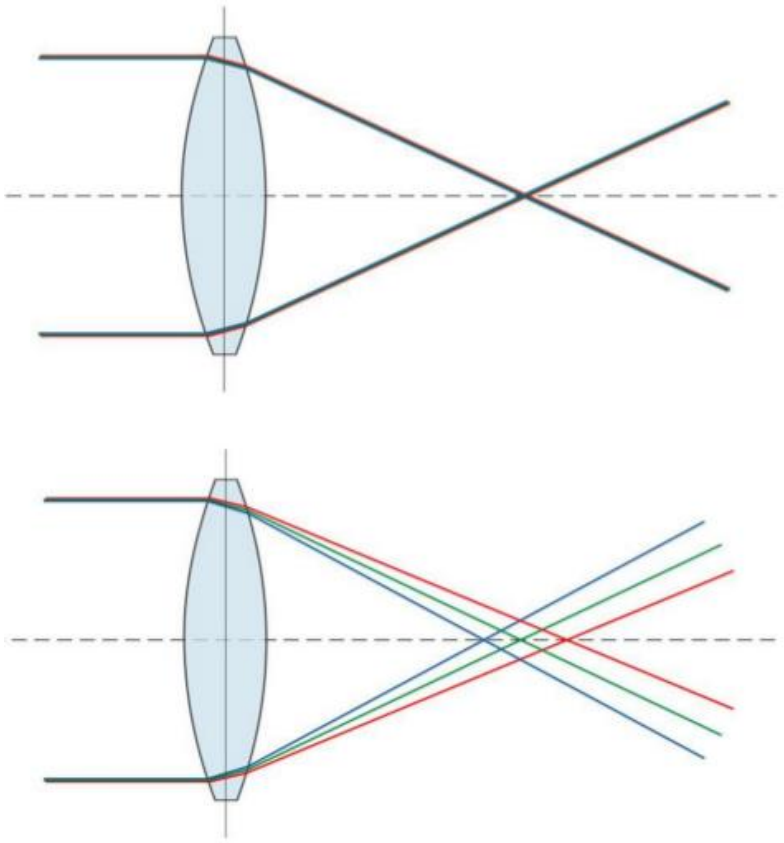
Jitter the patch locations

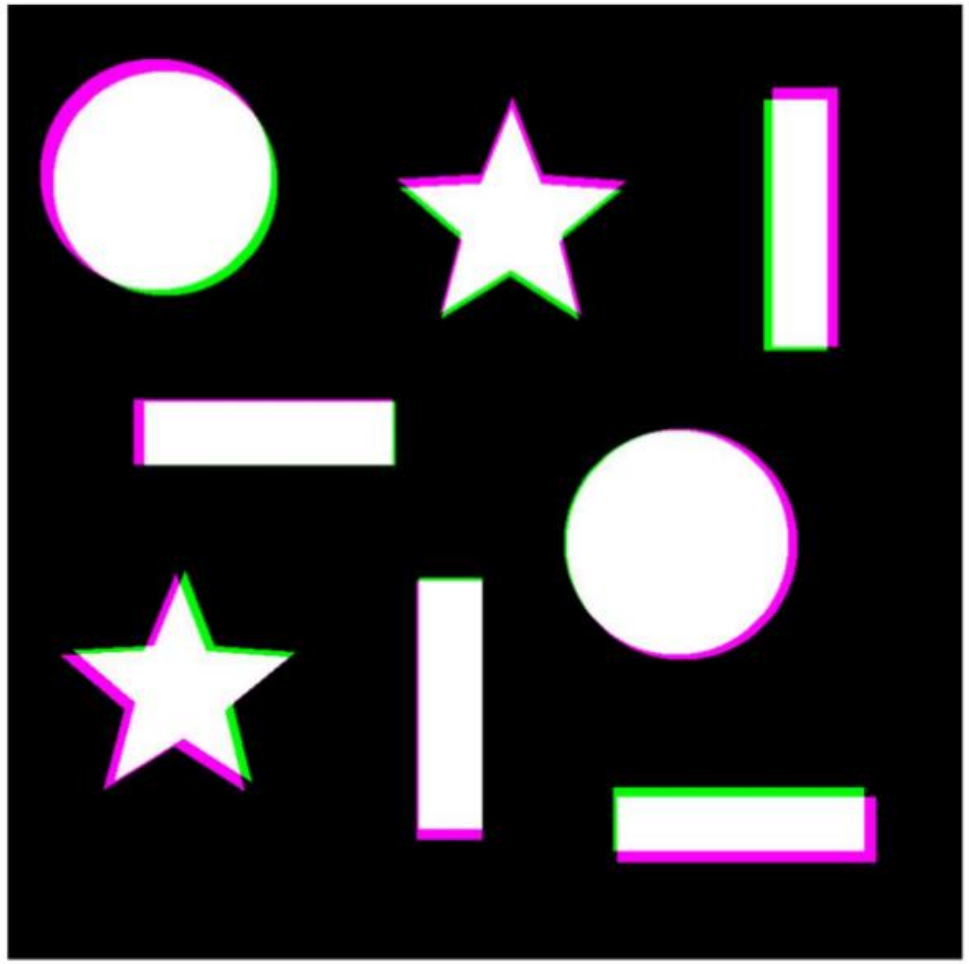
low-level cues like boundary patterns or textures continuing between patches could potentially serve as a lazy shortcut

it is possible that long lines spanning neighboring patches could could give away the correct answer

What is cheat 3? Hint…

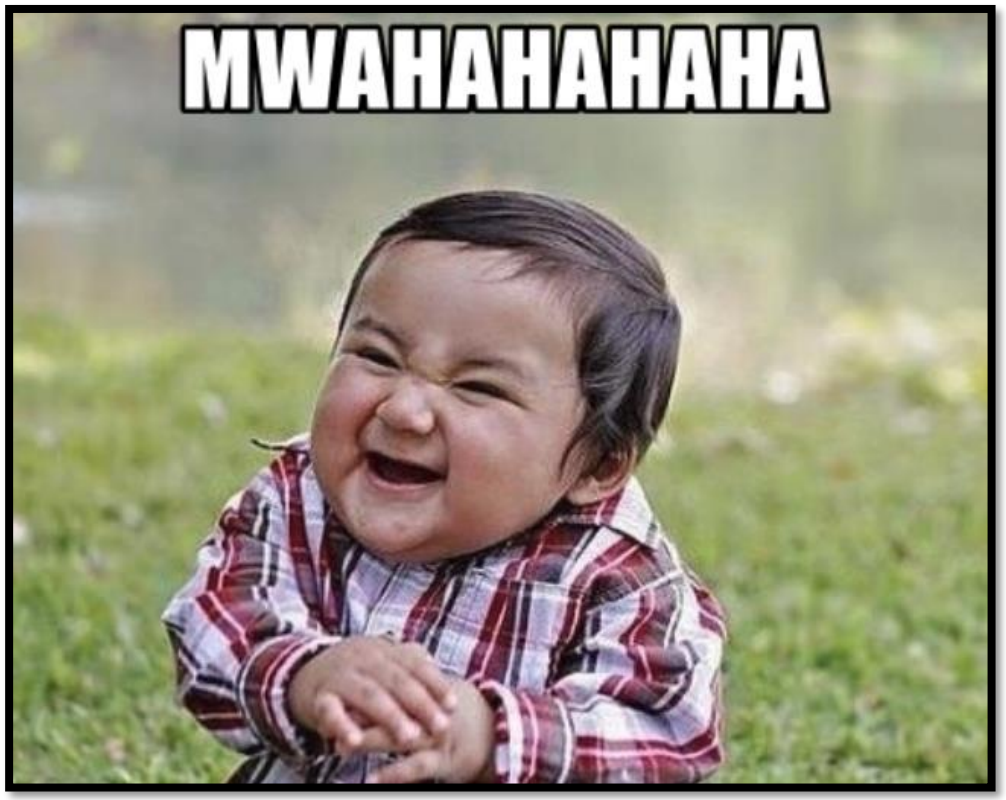
Chromatic aberration



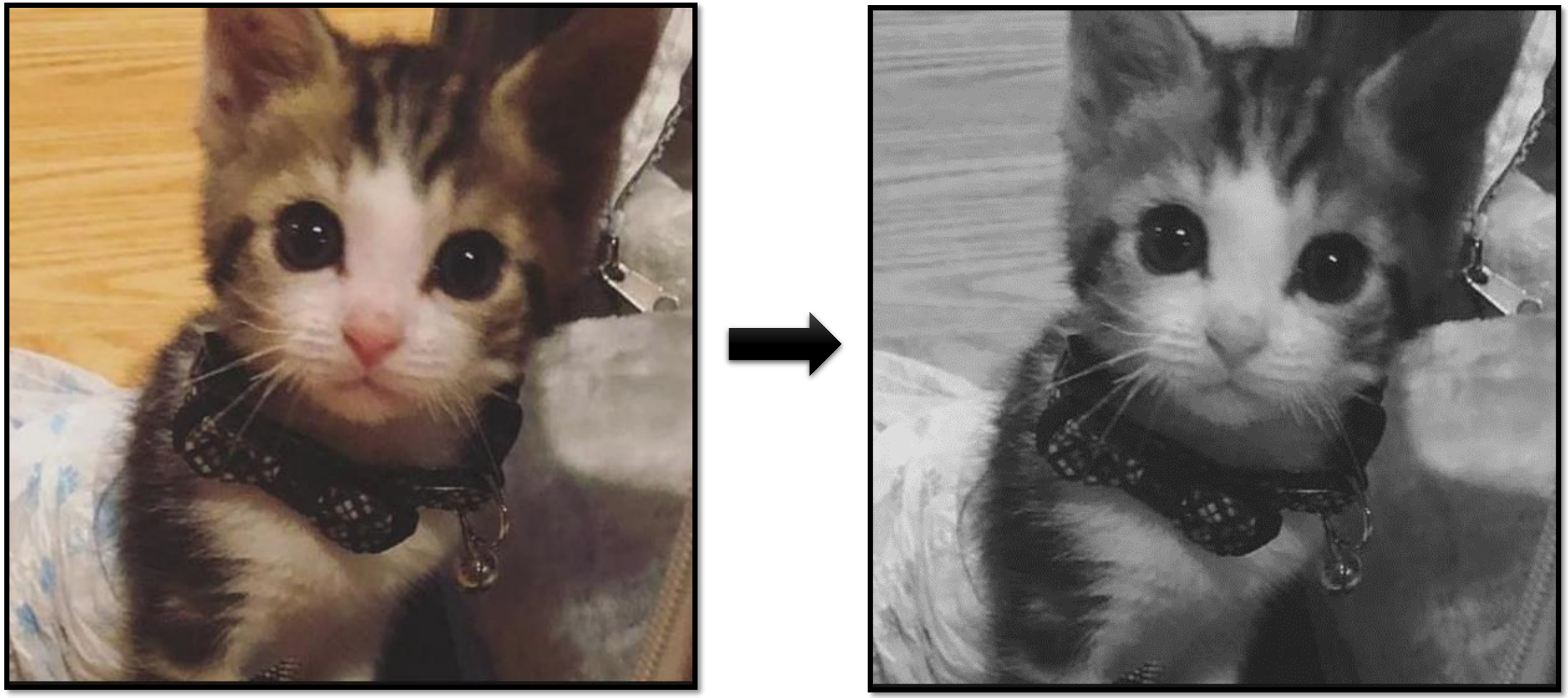


Cheat 3 (genius!)

- Chromatic aberration arises from differences in the way the lens focuses light at different wavelengths. In some cameras, one color channel (commonly green) is shrunk toward the image center relative to the others.
- A ConvNet, it turns out, can learn to localize a patch relative to the lens itself simply by detecting the separation between green and magenta (red + blue).
- Once the network learns the absolute location on the lens, solving the relative location task becomes trivial.

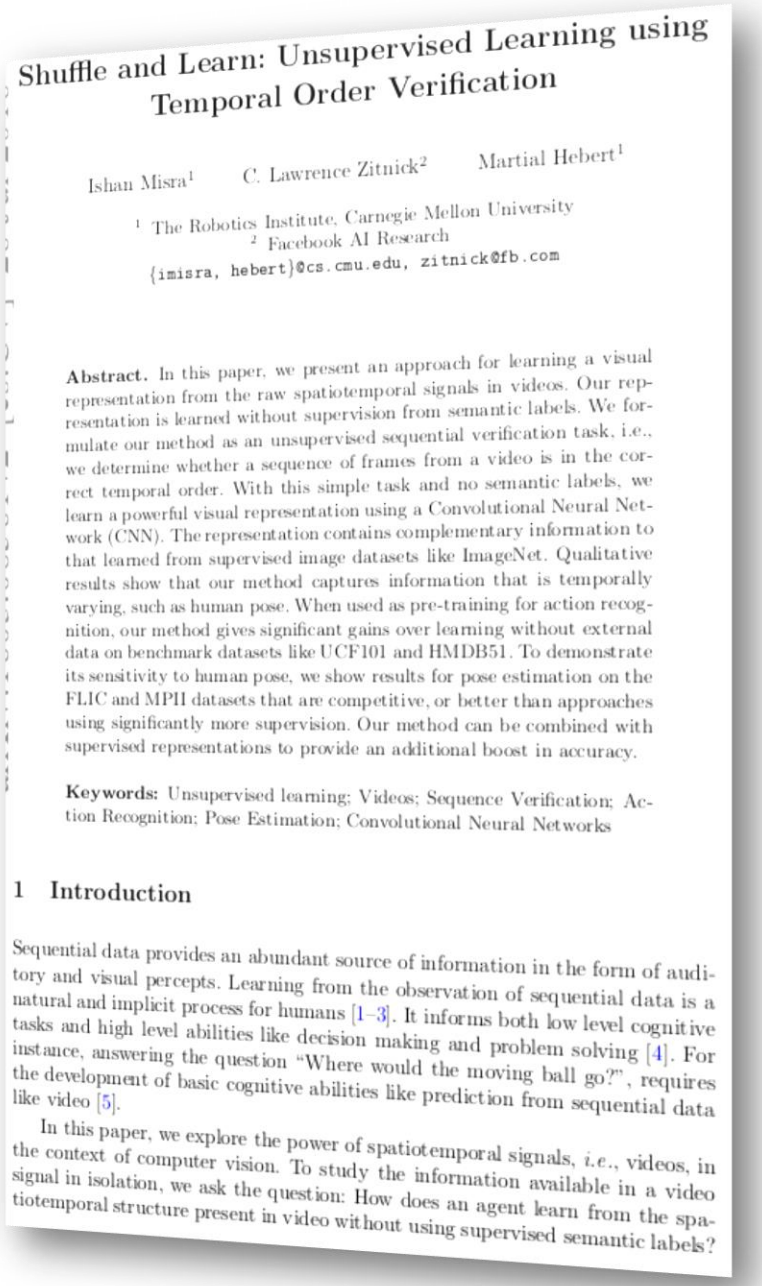


Anti-cheat 3



Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

https://arxiv.org/abs/1603.08561, 2016



Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

Ishan Misra[1] C. Lawrence Zitnick[2] Martial Hebert[1]

[1] The Robotics Institute, Carnegie Mellon University
[2] Facebook AI Research
{imisra, hebert}@cs.cmu.edu, zitnick@fb.com

Abstract. In this paper, we present an approach for learning a visual representation from the raw spatiotemporal signals in videos. Our representation is learned without supervision from semantic labels. We formulate our method as an unsupervised sequential verification task, i.e., we determine whether a sequence of frames from a video is in the correct temporal order. With this simple task and no semantic labels, we learn a powerful visual representation using a Convolutional Neural Network (CNN). The representation contains complementary information to that learned from supervised image datasets like ImageNet. Qualitative results show that our method captures information that is temporally varying, such as human pose. When used as pre-training for action recognition, our method gives significant gains over learning without external data on benchmark datasets like UCF101 and HMDB51. To demonstrate its sensitivity to human pose, we show results for pose estimation on the FLIC and MPII datasets that are competitive, or better than approaches using significantly more supervision. Our method can be combined with supervised representations to provide an additional boost in accuracy.

Keywords: Unsupervised learning; Videos; Sequence Verification; Action Recognition; Pose Estimation; Convolutional Neural Networks

1 Introduction

Sequential data provides an abundant source of information in the form of auditory and visual percepts. Learning from the observation of sequential data is a natural and implicit process for humans [1–3]. It informs both low level cognitive tasks and high level abilities like decision making and problem solving [4]. For instance, answering the question "Where would the moving ball go?", requires the development of basic cognitive abilities like prediction from sequential data like video [5].

In this paper, we explore the power of spatiotemporal signals, *i.e.*, videos, in the context of computer vision. To study the information available in a video signal in isolation, we ask the question: How does an agent learn from the spatiotemporal structure present in video without using supervised semantic labels?

Are these frames in the correct order or not?



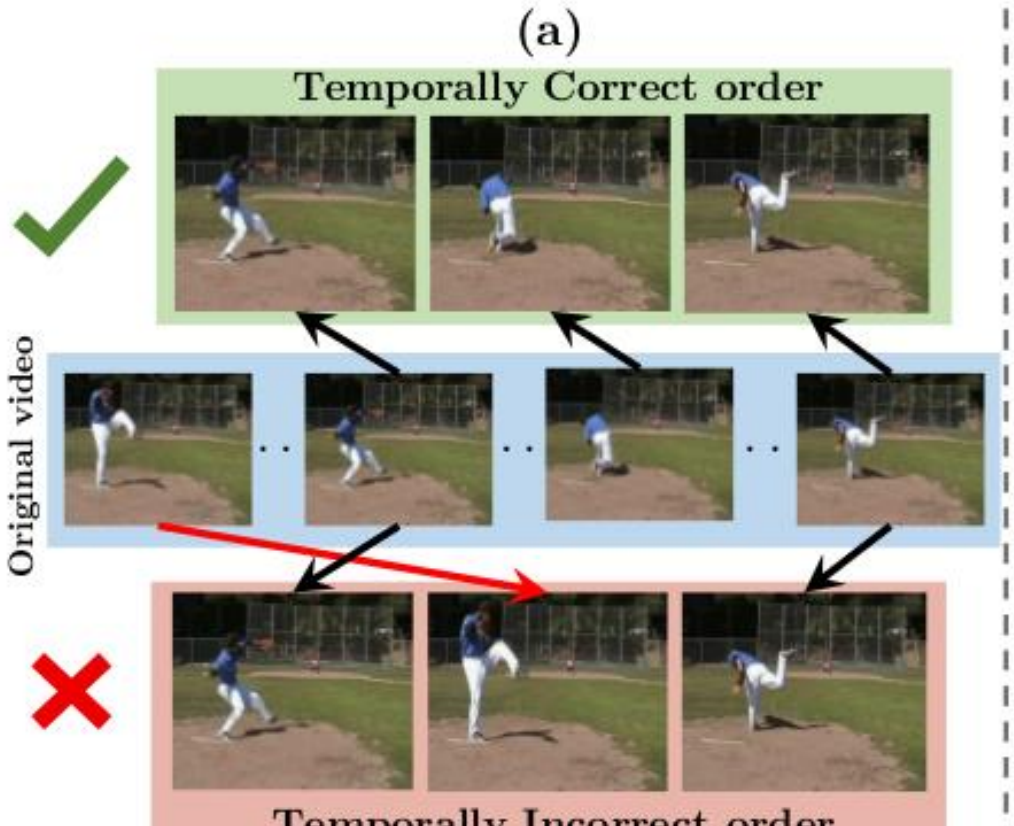


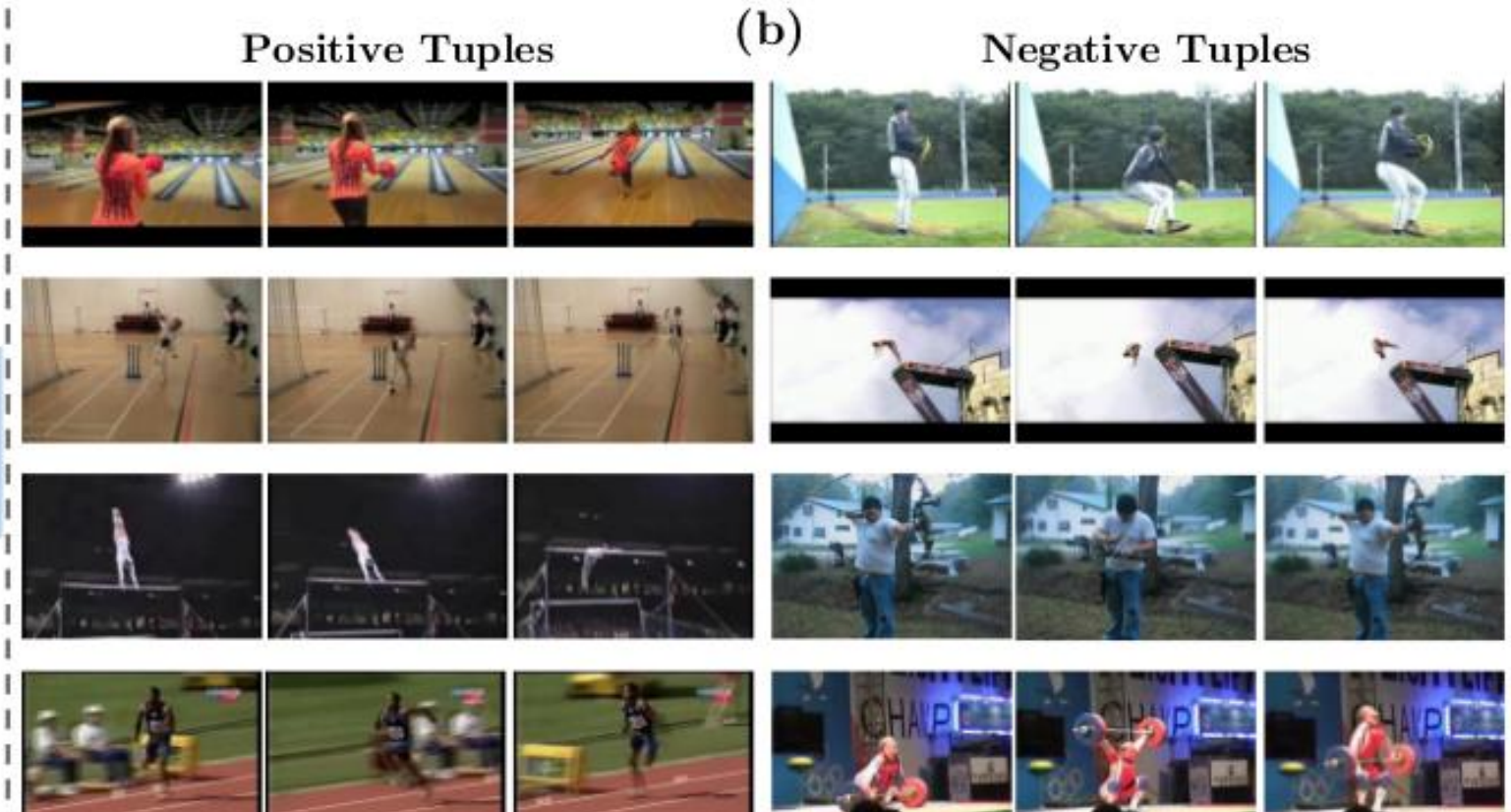



Pretext problem (classification): are these frames in the correct order?

Pretext task desiderata:

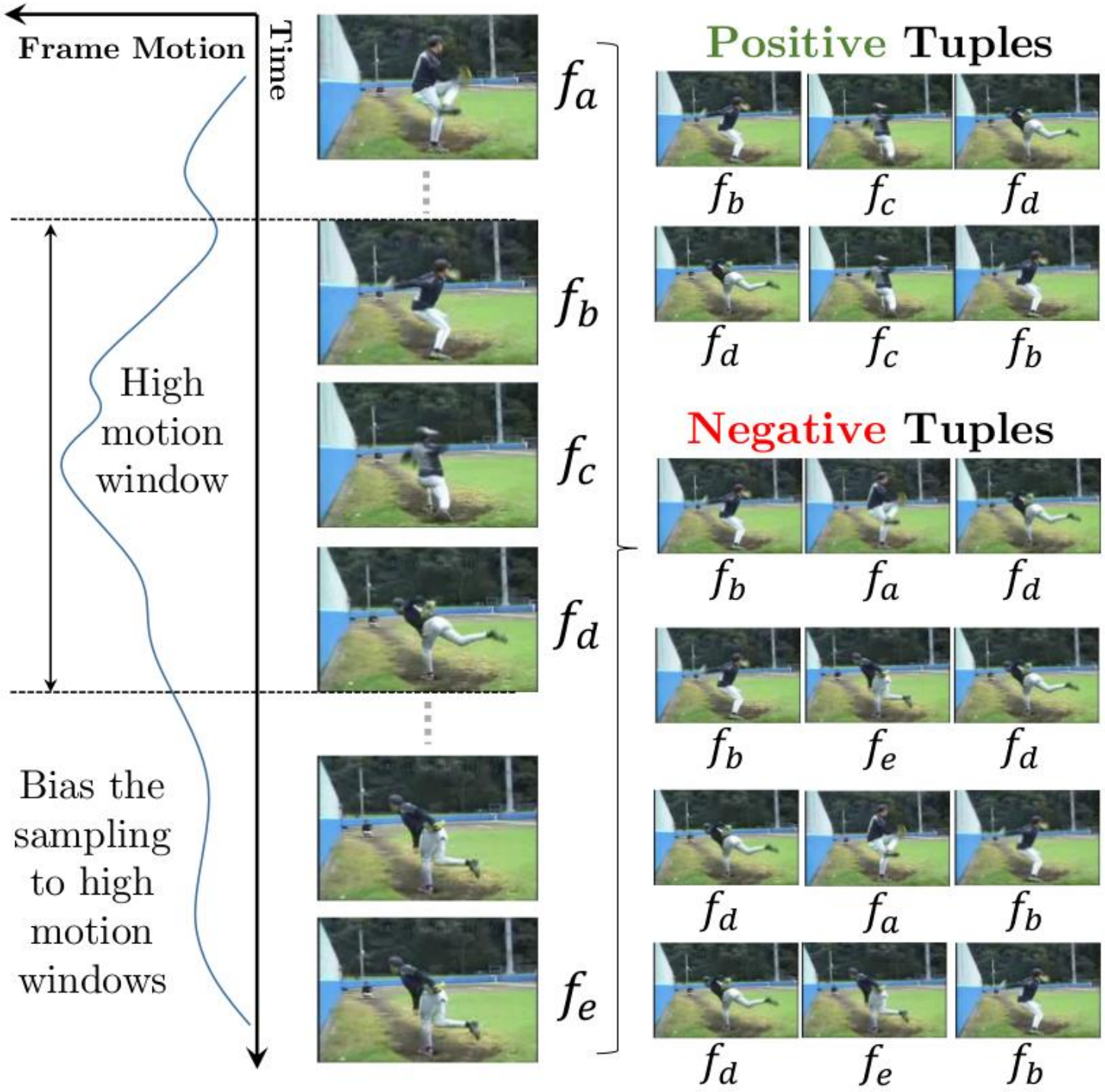
- you can get a ground truth for free from the data itself
- you need to "**understand**" the data in order to solve it



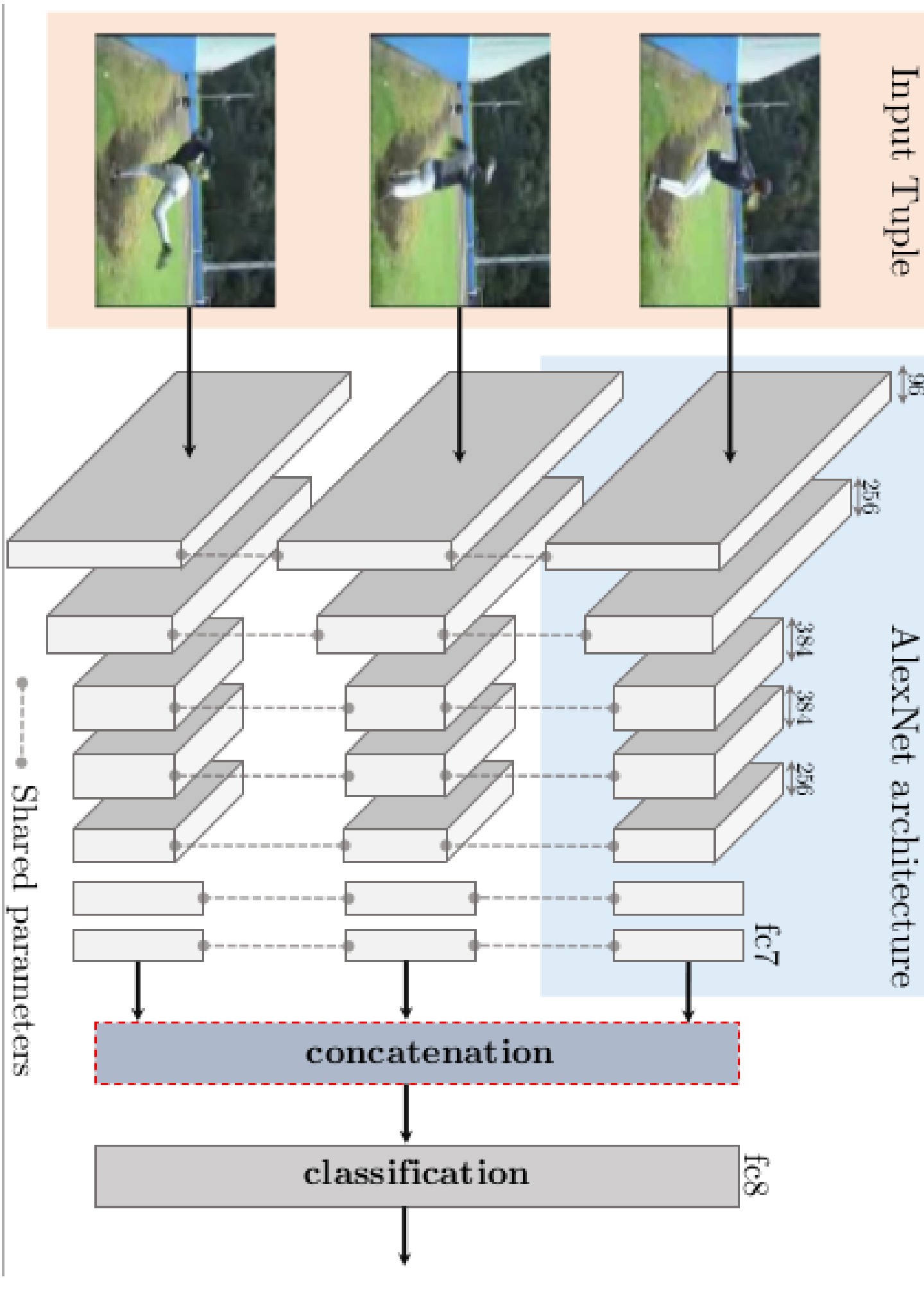


Sampling reasonable instances

- What is the problem if you sample frames from any video?
- That most samples will be impossible to predict due to almost no motion
- Then, only sample from high-motion windows

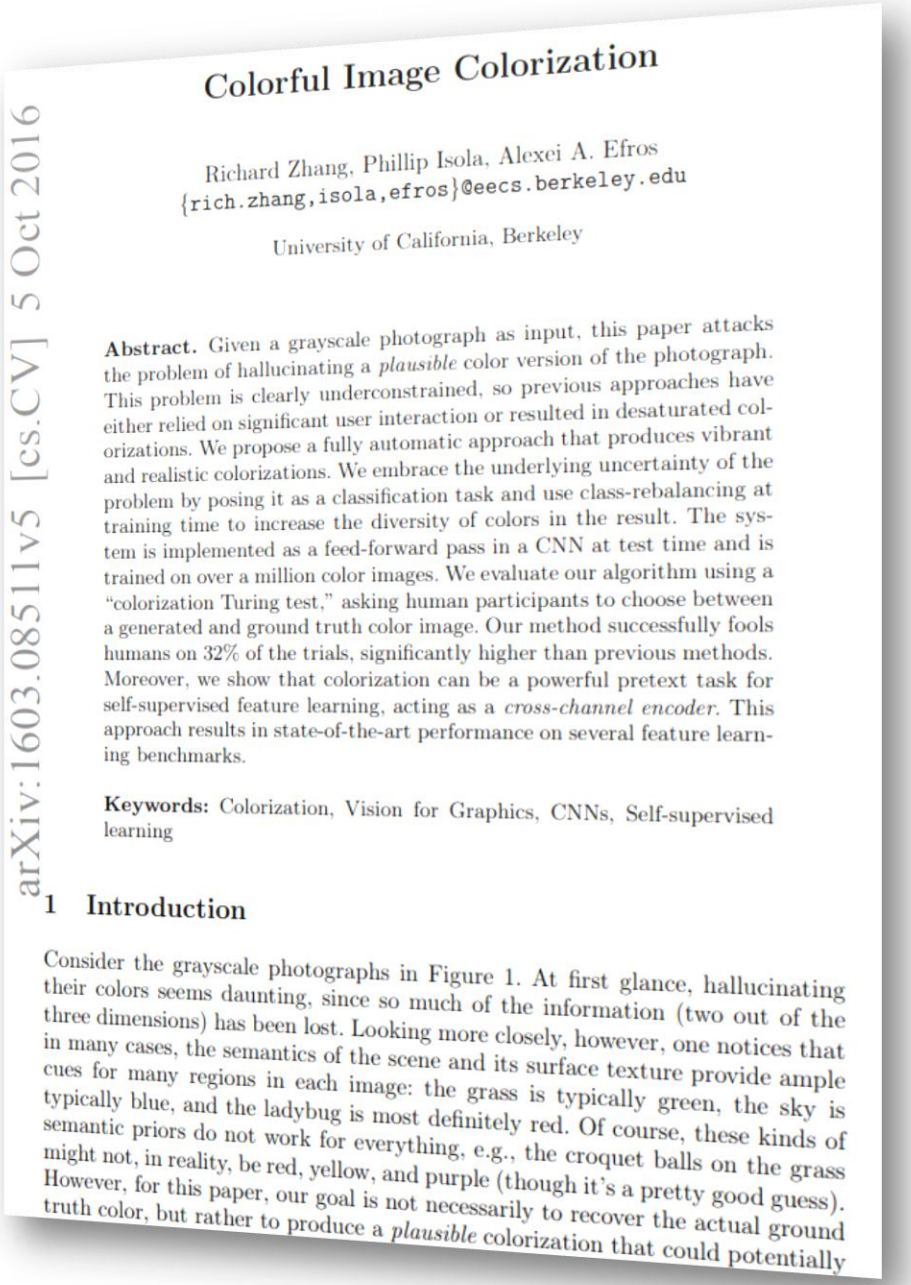


NN architecture



Colorful image colorization

2016



Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei A. Efros
{rich.zhang,isola,efros}@eecs.berkeley.edu

University of California, Berkeley

Abstract. Given a grayscale photograph as input, this paper attacks the problem of hallucinating a *plausible* color version of the photograph. This problem is clearly underconstrained, so previous approaches have either relied on significant user interaction or resulted in desaturated colorizations. We propose a fully automatic approach that produces vibrant and realistic colorizations. We embrace the underlying uncertainty of the problem by posing it as a classification task and use class-rebalancing at training time to increase the diversity of colors in the result. The system is implemented as a feed-forward pass in a CNN at test time and is trained on over a million color images. We evaluate our algorithm using a "colorization Turing test," asking human participants to choose between a generated and ground truth color image. Our method successfully fools humans on 32% of the trials, significantly higher than previous methods. Moreover, we show that colorization can be a powerful pretext task for self-supervised feature learning, acting as a *cross-channel encoder*. This approach results in state-of-the-art performance on several feature learning benchmarks.

Keywords: Colorization, Vision for Graphics, CNNs, Self-supervised learning

1 Introduction

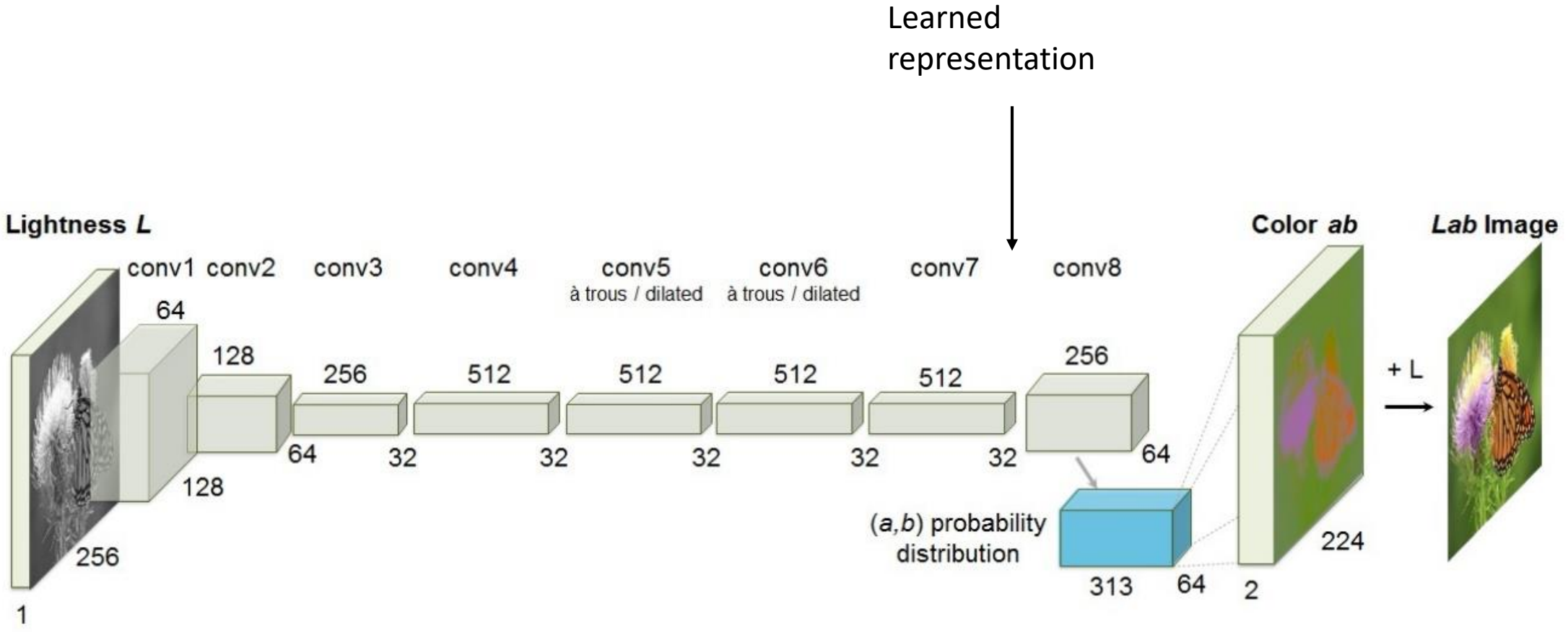
Consider the grayscale photographs in Figure 1. At first glance, hallucinating their colors seems daunting, since so much of the information (two out of the three dimensions) has been lost. Looking more closely, however, one notices that in many cases, the semantics of the scene and its surface texture provide ample cues for many regions in each image: the grass is typically green, the sky is typically blue, and the ladybug is most definitely red. Of course, these kinds of semantic priors do not work for everything, e.g., the croquet balls on the grass might not, in reality, be red, yellow, and purple (though it's a pretty good guess). However, for this paper, our goal is not necessarily to recover the actual ground truth color, but rather to produce a *plausible* colorization that could potentially

Image colorization (hallucinate colors)

Pretext task desiderata:

- you can get a ground truth for free from the data itself
- you need to "**understand**" the data in order to solve it

Main idea


Learned representation
Lightness L
conv1
conv2
conv3
conv4
conv5
à trous / dilated
conv6
à trous / dilated
conv7
conv8
64
128
256
512
512
512
512
256
64
32
32
32
32
32
64
128
256
1
(a,b) probability distribution
313
64
Color ab
2
224
+ L
Lab Image

Self-supervised deep learning conclusions

- You are interested in solving problem A
- Take a lot of data similar to the one you'll use, without labels
  (of course: you are lazy)
- Invent a problem B (*pretext task*) on the data for which
  - you can get a ground truth for free from the data itself
  - you need to "**understand**" the data in order to solve it
- Train a network for B

→ The network has learned something valuable for A, i.e. to understand the data

What we have seen so far

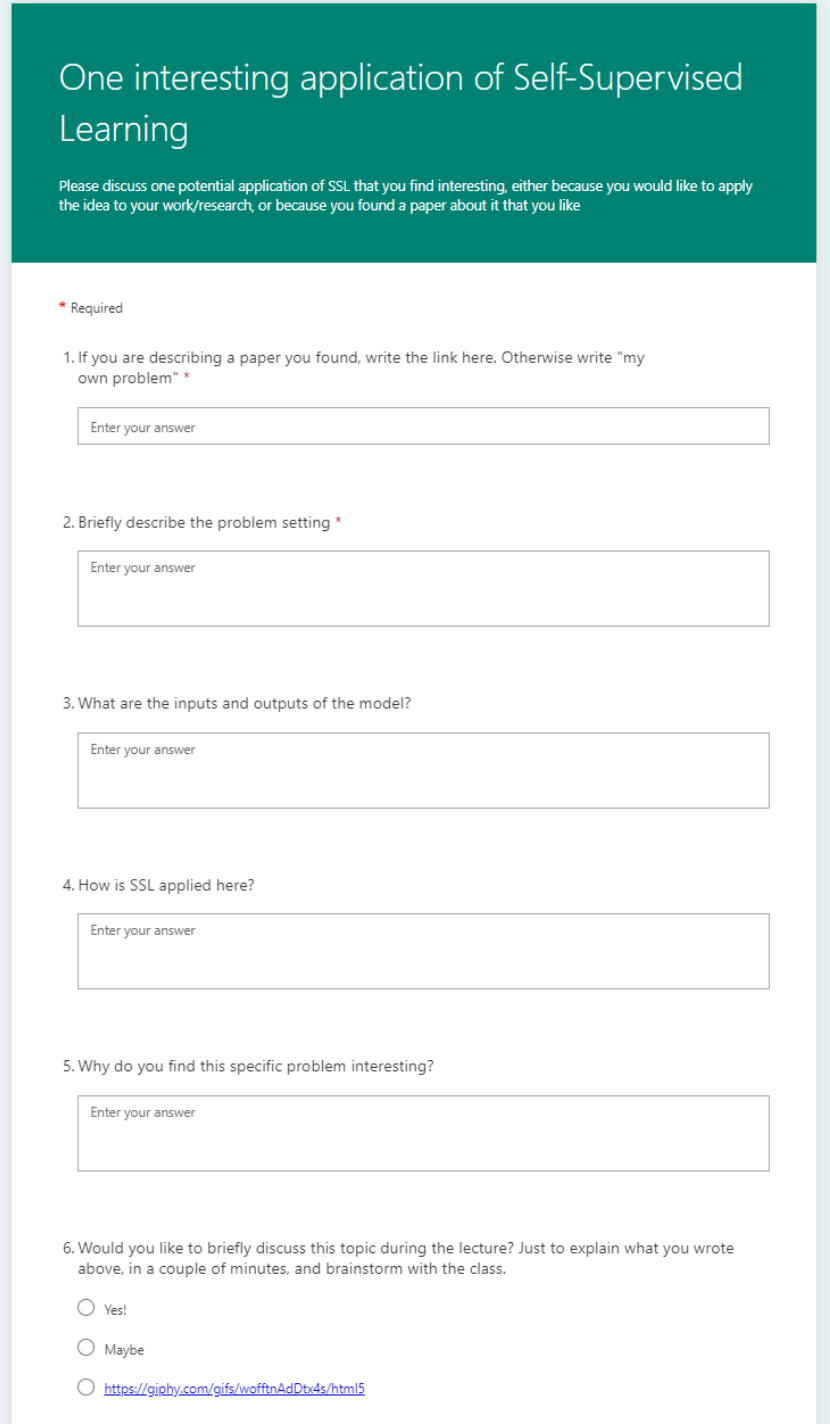
- Systems (typically robots) that collect their own training data but then solve a standard supervised learning task
- Systems that learn to extract meaningful representations from the data itself

Now…

- Take a coffee
- Fill this form:

  https://forms.office.com/Pages/ResponsePage.aspx?id=pDglg56TEk-VLuM8-eVonUcsmPZBUUVBgjoOf_oZabpUNVlMMEtQS1JBUFY4WUxUNVVOTTZBWE5DUS4u

- See you at 17:25 for discussion



One interesting application of Self-Supervised Learning

Please discuss one potential application of SSL that you find interesting, either because you would like to apply the idea to your work/research, or because you found a paper about it that you like

* Required

1. If you are describing a paper you found, write the link here. Otherwise write "my own problem" *

Enter your answer

2. Briefly describe the problem setting *

Enter your answer

3. What are the inputs and outputs of the model?

Enter your answer

4. How is SSL applied here?

Enter your answer

5. Why do you find this specific problem interesting?

Enter your answer

6. Would you like to briefly discuss this topic during the lecture? Just to explain what you wrote above, in a couple of minutes, and brainstorm with the class.

- Yes!
- Maybe
- https://giphy.com/gifs/wofftnAdDtx4s/html5