



Image-Based Situation Awareness

Sakari Lampola

Sakari Lampola

Master of Science 1983 (TTKK, system theory)

TTKK 1980-1983 (assistant, researcher)

Tampereen Sähkölaitos 1980-1982 (student trainee)

Imatran Voima Oy 1984-1988 (team leader)

Kymenlaakso University of Applied Sciences 1988-1991 (teacher)

Oy Cybersoft Ab 1989-2012

Motivation: Understanding Image

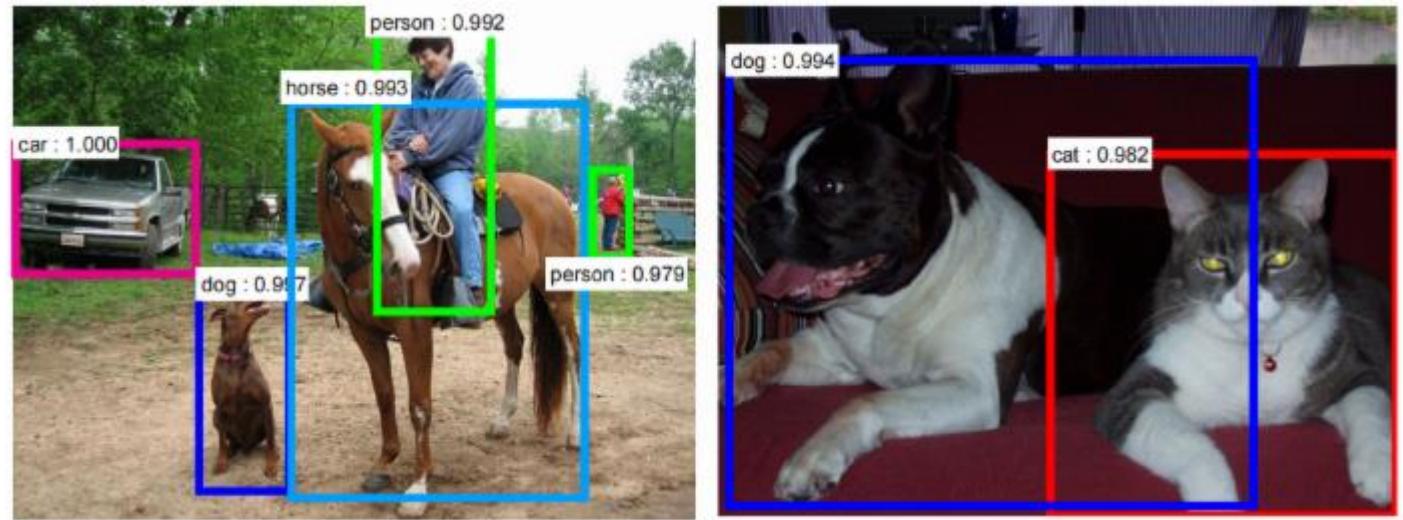
Understanding leads to action



Understanding
enables
prediction



Object detection helps understanding



Beyond state of the art



- External knowledge
- Expectations
- Probabilities
- Emotions
- ...

Understanding Image: Image Captioning

Image captioning examples

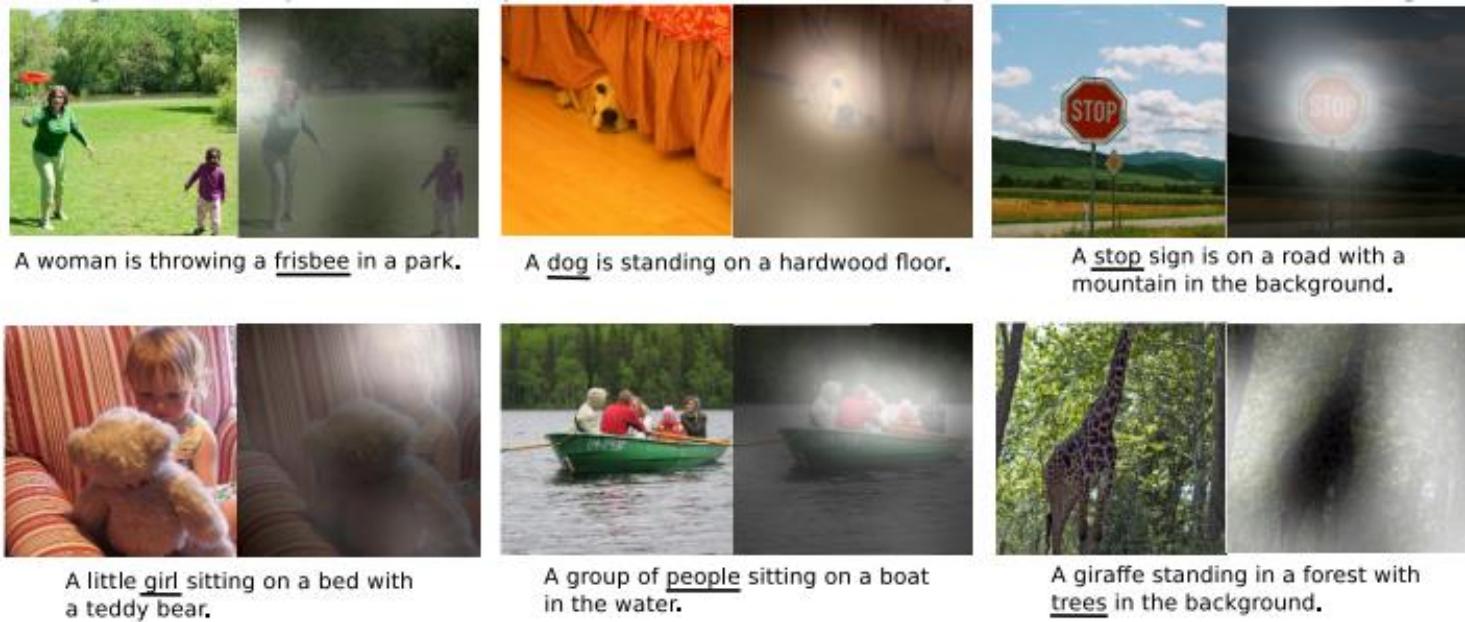


Image captioning models

Deep Visual-Semantic Alignments for Generating Image Descriptions

Andrej Karpathy Li Fei-Fei

Department of Computer Science, Stanford University

{karpathy, feifeili}@cs.stanford.edu

Abstract

We present a model that generates natural language descriptions of images and their regions. Our approach leverages datasets of images and their sentence descriptions to learn about the inter-modal correspondences between language and visual data. Our alignment model is based on a novel combination of Convolutional Neural Networks over image regions, bidirectional Recurrent Neural Networks over sentences, and a structured objective that aligns the two modalities through a multimodal embedding. We then describe a Multimodal Recurrent Neural Network architecture that uses the inferred alignments to learn to generate novel descriptions of image regions. We demonstrate that our alignment model produces state of the art results in retrieval experiments on Flickr8K, Flickr30K and MSCOCO datasets. We then show that the generated descriptions significantly outperform retrieval baselines on both full images and on a new dataset of region-level annotations.

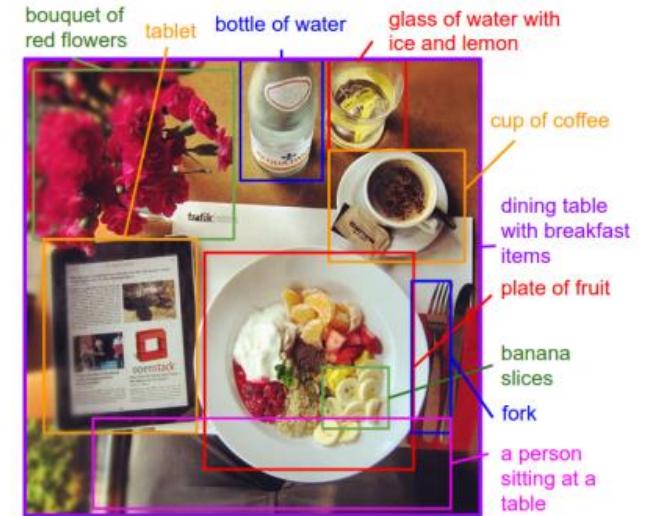


Figure 1. Motivation/Concept Figure: Our model treats language as a rich label space and generates descriptions of image regions.

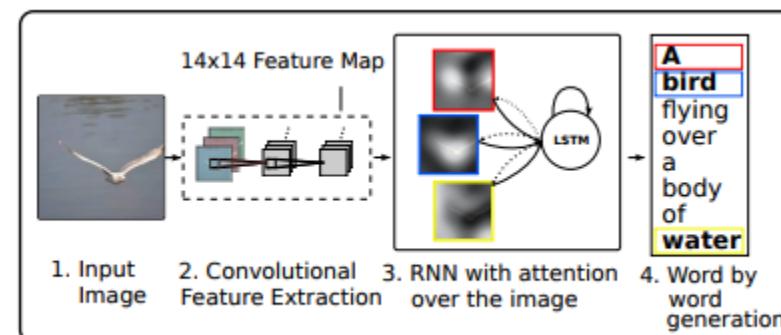
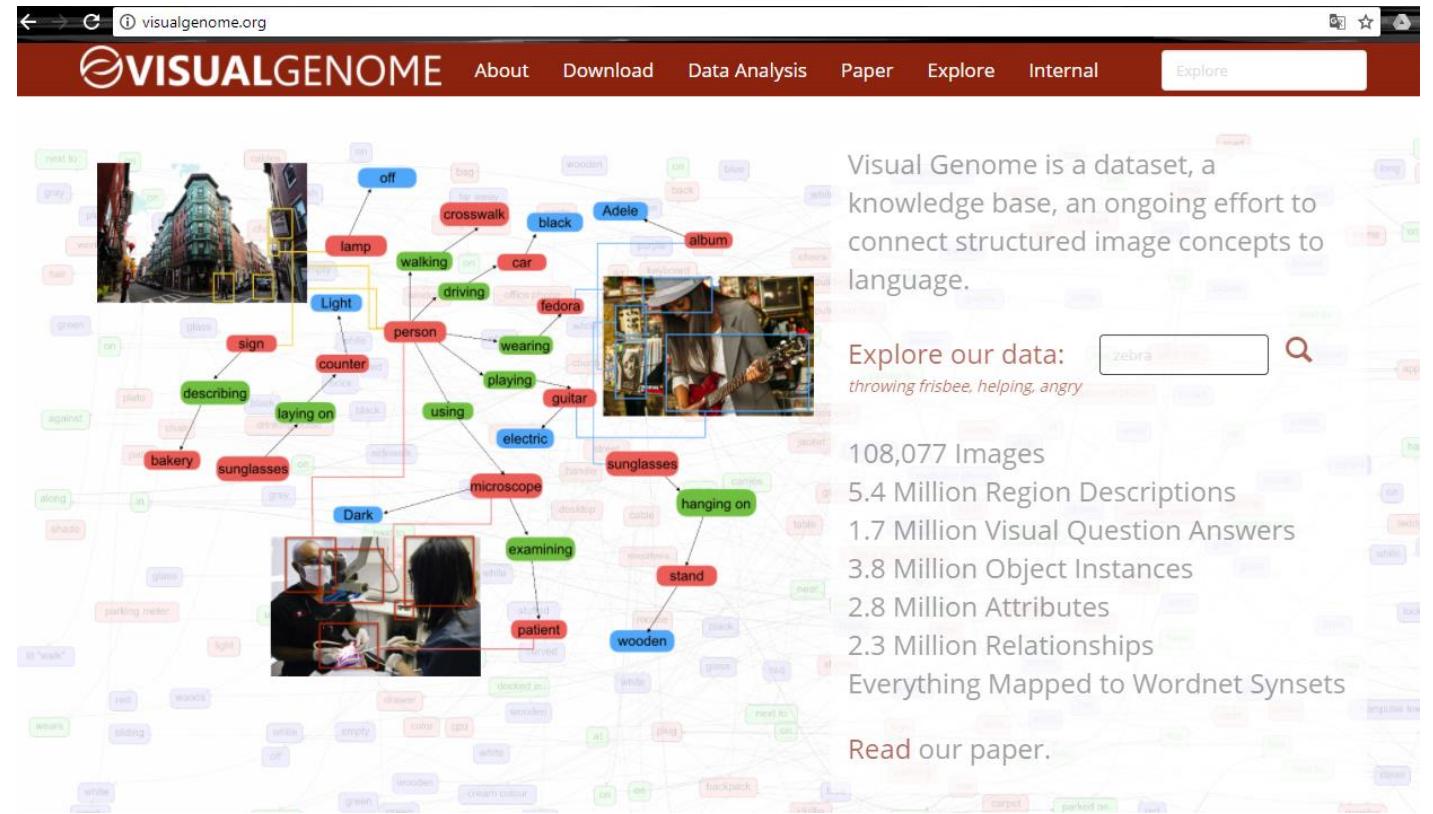


Image captioning datasets



Understanding Image: Visual Question Answering (VQA)

VQA examples



What is on the right side of the cabinet?

Neural-Image-QA: bed 3 bed

Language only: bed 6 table

How many drawers are there?

Table 7. Examples of questions and answers. Correct predictions are colored in green, incorrect in red.



What is on the refrigerator?

Neural-Image-QA: magnet, paper blue, white bed sheets, pillow

Language only: magnet, paper blue, green, red, yellow doll, pillow



What is the colour of the comforter?

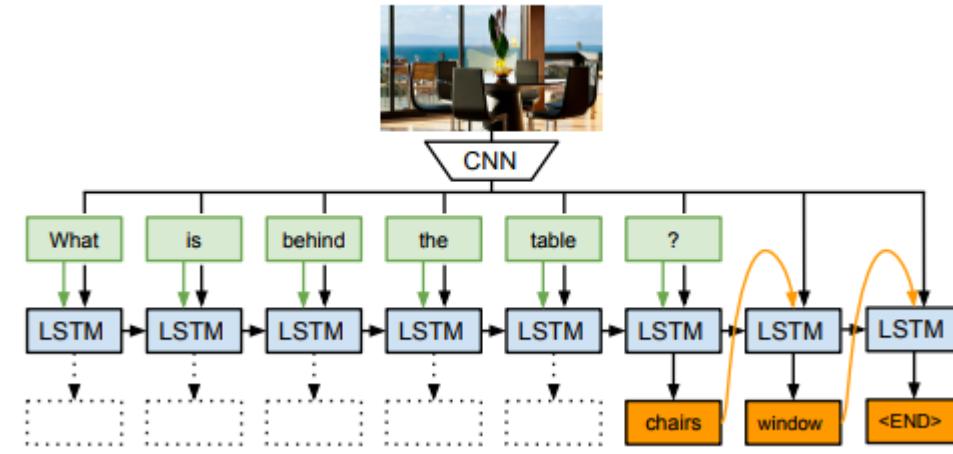


What objects are found on the bed?

VQA questions categories

Object Presence Is there a traffic light in the photo?	Subordinate Object Recognition What animal is in the picture?	Scene Classification What is the weather like?
Sport Recognition What sport are they playing?	 A photograph showing a woman in a dark t-shirt and shorts standing on a grassy field, throwing a yellow ball towards a black and white dog. A white fence is visible in the background.	Activity Recognition What is the dog doing?
Other Attributes What is the fence made of?		Counting How many dogs are there?
Positional Reasoning What is to the left of the woman?		Absurd What color is the couch?
Sentiment Understanding How is the woman feeling?	Color Attributes What color are the woman's shorts?	Utility/Affordance What object can be thrown?

VQA models and datasets



	Images	Questions	Annotation Source	Question Types	Unique Answers	MC/OE
DAQUAR	1,449	16,590	Both	3	968	OE
COCO-QA	123,287	117,684	Auto	4	430	OE
COCO-VQA	204,721	614,163	Manual	3	145,172	Both
Visual7W	47,300	327,939	Manual	7	25,553	MC
Visual Genome	108,000	1,773,358	Manual	6	207,675	OE
TDIUC (This Paper)	167,437	1,654,167	Both	12	1,618	OE

Understanding Image: Visual Dialog

Visual dialog example

Visual Dialog

A cat drinking water out of a coffee mug.

What color is the mug?

White and red

Are there any pictures on it?

No, something is there can't tell what it is

Is the mug and cat on a table?

Yes, they are

Are there other items on the table?

Yes, magazines, books, toaster and basket, and a plate

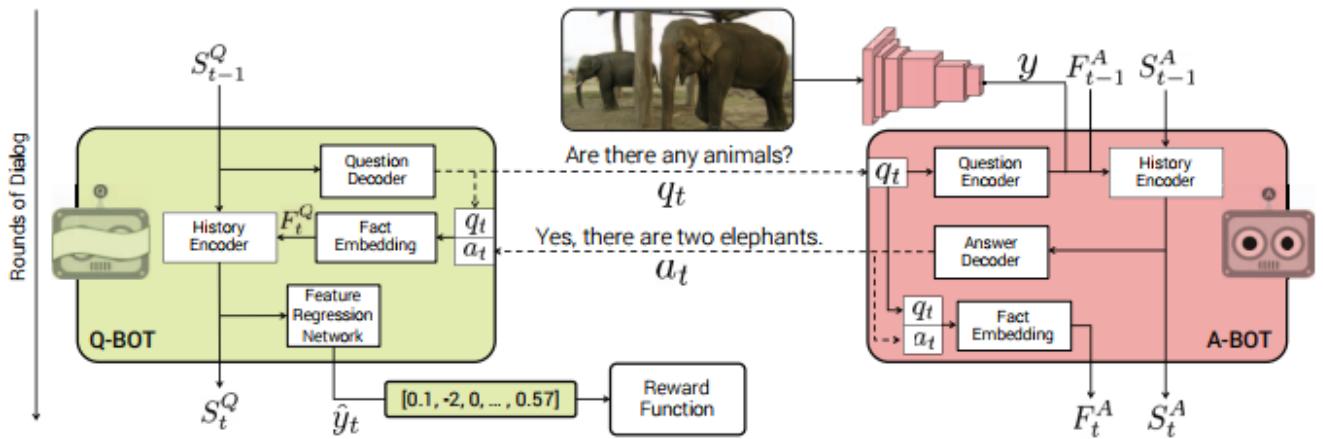
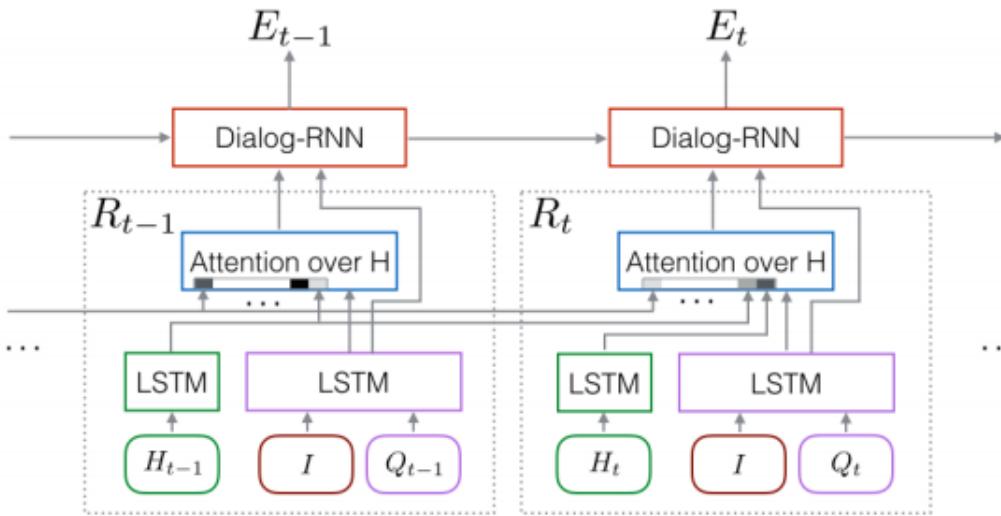
C Start typing question here ... >

Visual dialog example

Image + Caption	Human Dialog	SL-pretrained	RL-full-QAF
 A couple of people in the snow on skis.	<p>What are their genders? 1 man 1 woman Are they both adults? Yes Do they wear goggles? Looks like sunglasses Do they have hats on? Man does Are there any other people? No What color is man's hat? Black Is it snowing now? No What is woman wearing? Blue jacket and black pants Are they smiling? Yes Do you see trees? Yes</p>	<p>How many people? 2 Male or female? I can't tell What are they wearing? I can't tell, they are far away What color are skis? I can't tell Are they wearing goggles? Yes Are they wearing goggles? Yes Are they wearing goggles? Yes Are they wearing hats? Yes Are they wearing goggles? Yes</p>	<p>Are people male or female or both? I think 1 is male, can't see other 1 ... What are they wearing? They are all wearing snow pants and jackets Is it snowing? It does not appear to be snowing at moment Can you see any trees? Yes, there are lot of trees in background Can you see any buildings in background? No, I can not Can you see any trees? Yes, there are lot of trees in background Can you see any buildings in background? No, I do not see any buildings Can you see any trees? Yes, there are lot of trees in background Can you see any buildings in background? No, I do not see any buildings Does it look like they are in park? It does not appear to be</p>
 A man riding a skateboard down the side of a ramp.	<p>How old is man? In his teens Is he wearing helmet? No Is he wearing any protective gear? No What color is his hair? Dirty blonde or dark brown What color is his shirt? Black with red letters Is he wearing shorts? Yes What color are his shorts? Denim Are there other people visible? No What color are his shoes? I can't see them Is this at skate park? Yes</p>	<p>How old is man? I can't see his face, I don't know What is he wearing? shorts and t shirt What color is his shirt? white what color are his pants? black what color is his shirt? blue What color are his shoes? Black What color are his shoes? Black What color are his shoes? Black What color are his shoes? Black</p>	<p>What color is skateboard? It is hard to tell but I think it's black Is he wearing helmet? Yes Is he wearing knee pads? No, no knee pads Is it day or night? It is daytime Is it sunny? Yes, it is sunny Can you see any other people? There are 2 other people in background Is it day or night? It is daytime Is it sunny? Yes, it is sunny What color is skateboard? It is hard to tell but I think it's black Is he wearing helmet? Yes, he is wearing helmet</p>

Table 1: Selected examples of Q-BOT-A-BOT interactions for SL-pretrained and RL-full-QAF. RL-full-QAF interactions are diverse, less prone to repetitive and safe exchanges (“can’t tell”, “don’t know” etc.), and more image-discriminative.

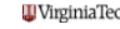
Visual dialog models



Visual dialog datasets

C  Turvallinen | <https://visuality.org/data>

Visual Dialog Overview People Data Bibtex Acknowledgements



VisDial Dataset

[Code for the real-time chat interface used to collect the VisDial dataset on Amazon Mechanical Turk](#)

VisDial v0.9

[Training set \(235M\)](#)
82,783 images

[Validation set \(108M\)](#)
40,504 images

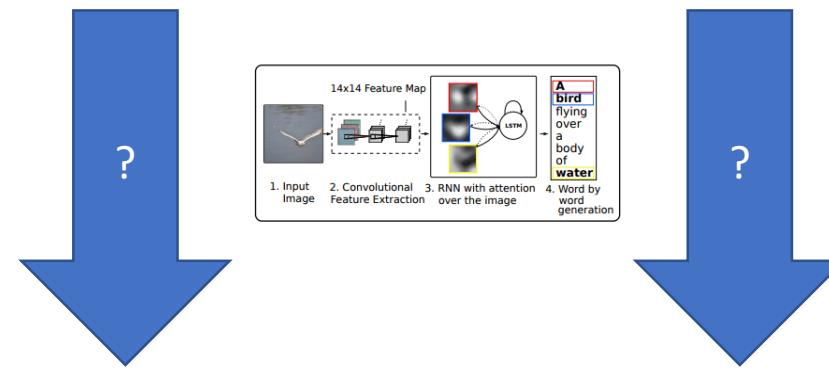
Understanding Image: The Big Problem

Clever Hans



(by Larry Zitnick, 29.10.2017)

Current
methods are
inexplicable
mappings from
image to
language



construction worker in orange safety vest is working on road.

one man and two women sitting in living room
man and woman are playing wii game while woman sits on couch with wine glass in her hand
group of people sitting on couch with their laptops

Understanding Image: Visual Reasoning

Visual reasoning example



How many chairs are at the table?



Is there a pedestrian in my lane?



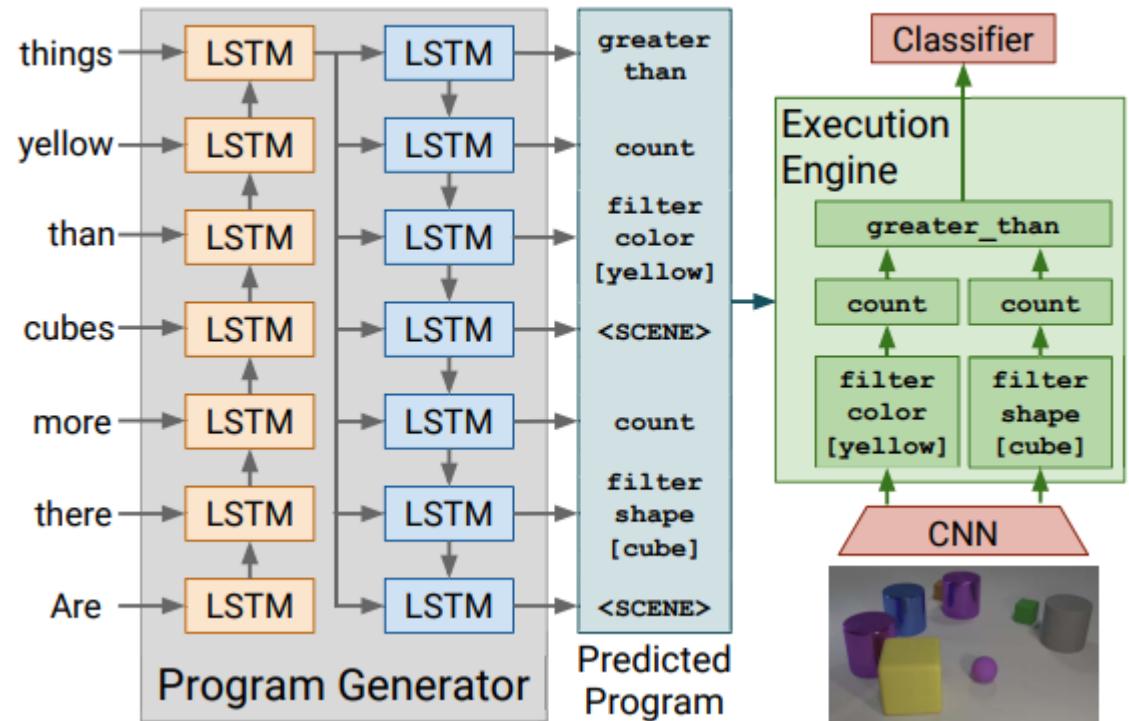
Is the person with the blue hat touching the bike in the back?



Is there a matte cube that has the same size as the red metal object?

Visual reasoning models

Question: Are there more cubes than yellow things? Answer: Yes



Visual reasoning datasets

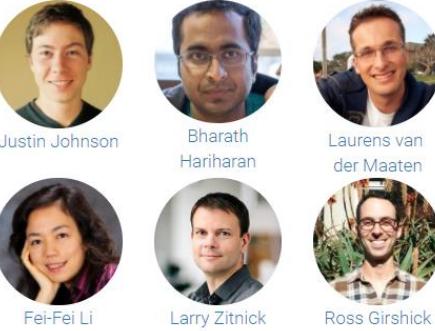
← → C cs.stanford.edu/people/jcjohns/clevr/ 



CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning

Abstract

When building artificial intelligence systems that can reason and answer questions about visual data, we need diagnostic tests to analyze our progress and discover shortcomings. Existing benchmarks for visual question answering can help, but have strong biases that models can exploit to correctly answer questions without reasoning. They also conflate multiple sources of error, making it hard to pinpoint model weaknesses. We present a diagnostic dataset that tests a range of visual reasoning abilities. It contains minimal biases and has detailed annotations describing the kind of reasoning each question requires. We use this dataset to analyze a variety of modern visual reasoning systems, providing novel insights into their abilities and limitations.



Justin Johnson
Bharath Hariharan
Laurens van der Maaten
Fei-Fei Li
Larry Zitnick
Ross Girshick



Download paper (arXiv)

Presented at CVPR 2017

Thesis

Preliminary work 2017

- Richard Szeliski: "Computer Vision: Algorithms and Applications"
- Sebastian Thrun, Wolfram Burgard, Dieter Fox: "Probabilistic Robotics"
- Kevin P. Murphy: "Machine Learning. A Probabilistic Perspective"
- Stanford CS231n: "Convolutional Neural Networks for Visual Recognition" (YouTube)
- Udacity: "Introduction to Computer Vision" (YouTube)
- Udacity: "Intro to Machine Learning" (YouTube)
- Udacity: "Machine Learning" (YouTube)
- Udacity: "Computational Photography" (YouTube)
- Daphne Koller: "Probabilistic Graphical Models" (YouTube)
- Reviews on calculus, linear algebra, statistics, probability, ...
- Kaggle competition
- Matlab, Python
- ICCV 2017

General information

Title:
Image-Based Situation Awareness

Doctoral Programme:
TUT Engineering Sciences, TTITO

Faculty:
Department of Automation Science and Engineering

Supervisor:
Professor Risto Ritala

Approved:
8.12.2017

Schedule:
2018-2021

Goal

“Develop methods which can perceive the elements in the environment within a volume of time and space, comprehend their meaning and project their status in the near future. The methods will be tested on a system which will help visually impaired people to understand their environment better. The system will be demonstrated on existing hardware.”



Research questions

- 1) By which methods an image-based perception of the environment can be produced, including object detection and localization in the field of view context?
- 2) By which methods a comprehension of the situation can be produced? This includes estimating object locations and their velocities in a 3d coordinate system, also for currently hidden objects.
- 3) By which methods a future projection of the situation can be created?
- 4) By which methods the comprehension and the future projection can be translated into human language?
- 5) How are uncertainty and reliability considered and represented?

Prototype



OrCam MyEye 2.0

"For blind and partially sighted people, an artificial vision device with a lightweight smart camera that instantly reads text aloud - from any surface – and recognizes faces, products, and money notes in real time."

Thesis Prototype

A reliable and fast system which can help a visually impaired person to understand her environment better by accurate and rich description. This is accomplished by

- capturing images
- detecting objects
- comprehending situation
- describing environment
- projecting future
- implying warnings

The system will be demonstrated on physical hardware, preferably mobile.

Schedule

	2018	2019	2020	2021
Methodology				
Preparation of research infra				
Method survey				
Building test cases				
Testing and comparison				
Prototype				
Definition				
Planning				
Implementation				
Testing and fixing				
Method follow-up				
Writing thesis				
Dissertation				

Supporting studies

2018-2019

TTI-90006 Orientation to Doctoral Studies, 3 cr

BMT-51506 Human Visual System, 5 cr

ASE-7516 Dynamic Planning with Incomplete Information, 5 cr

2019-2020

SGN-31007 Advanced Image Processing, 5 cr

ASE-7536 Model-Based Estimation, 5-7 cr

SGN-25006 Vector Space Methods for Signal and Image Processing, 5 cr

2020-2021

MAT-75006 Artificial Intelligence, 7 cr

SGN-43006 Knowledge Mining and Big Data, 5 cr

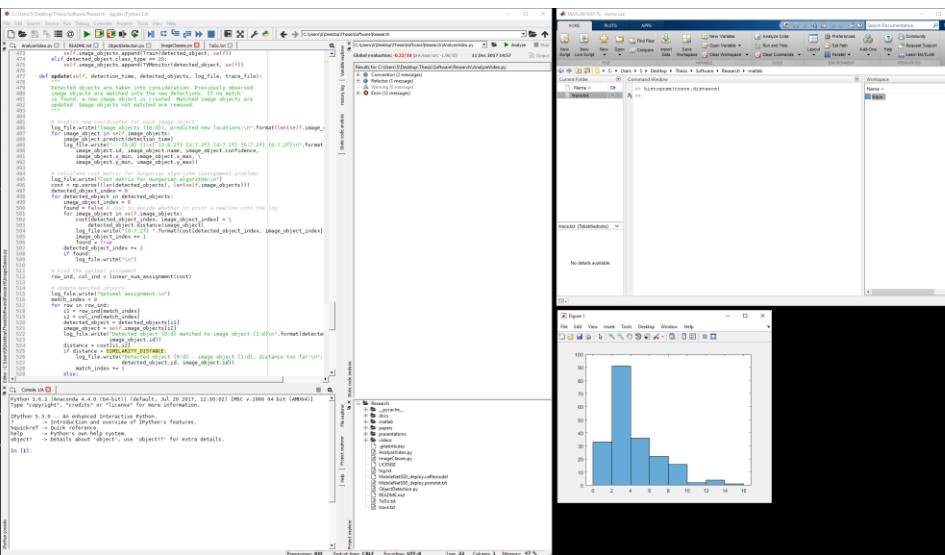
TTI-90016 Essay on Philosophy of Science/Research Ethics, 2 cr

Goodfellow, Bengio, Courville: “Deep Learning”

Udacity, Youtube courses on machine learning and computer vision

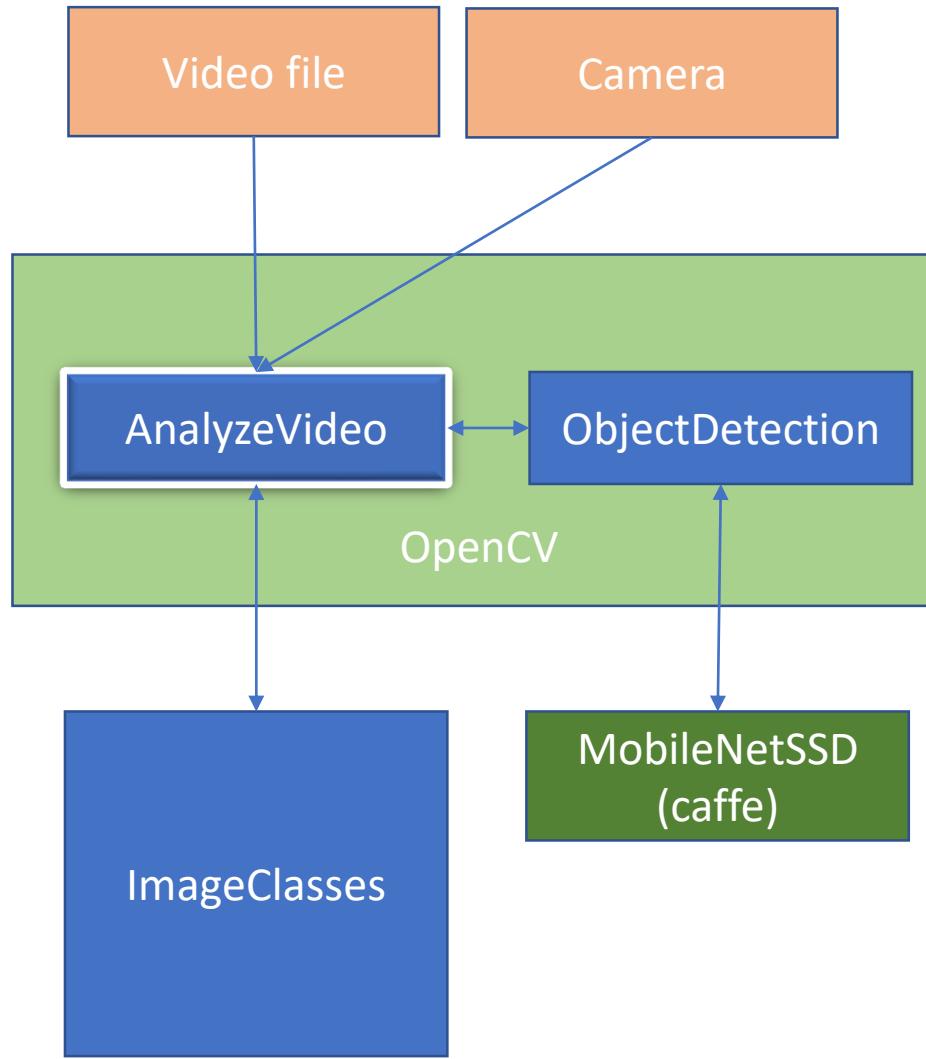
Research environment

- Software
 - Matlab
 - Python, Anaconda, Spyder
 - OpenCV
 - Keras, TensorFlow, (TensorFlow Lite, if required)
 - NumPy, Pandas, Matplotlib, SkLearn, SciPy, Seaborn,...
 - (PyTorch, Caffe2Go, if required)
 - Visual Studio - prototype user interface
- Hardware
 - Intel i7-4790 4GHz / 16Gt Ram
 - NVIDIA GeForce GTX 780 Ti (CUDA)
 - (Google Cloud Platform, if required)



Research environment

Software architecture



Research environment



- Video processing, frame by frame
- Python 3.6.2, OpenCV 3.3.0
- Single Shot Detector + MobileNets (Google)

Research environment

```

second = 25.00
width = 1280
height = 720
-----
0, frame 1
objects (6):
  1. 0.00 569 811 573 721 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04
  1. 0.96 804 986 298 440 0.18 0.03 0.00 0.01 0.11 0.35 0.13 0.19
  1. 0.78 520 730 384 531 0.02 0.00 0.00 0.01 0.01 0.29 0.06 0.50
  1. 0.78 593 741 205 304 0.12 0.03 0.00 0.00 0.06 0.20 0.11 0.47
  1. 0.64 415 585 267 382 0.02 0.00 0.00 0.00 0.10 0.56 0.20 0.11
  1. 0.39 311 429 200 286 0.05 0.00 0.00 0.01 0.15 0.57 0.15 0.06
objects (0), predicted new locations:
fix for Hungarian algorithm:
assignment:
object 1 created: car  1.00 569 811 573 721
object 2 created: car  0.96 804 986 298 440
object 3 created: car  0.94 520 730 379 522
object 4 created: car  0.78 593 741 205 304
object 5 created: car  0.64 415 585 267 382
object 6 created: car  0.39 311 429 200 286
-----
1, frame 2
objects (6):
  1. 0.00 569 811 573 721 0.00 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04
  1. 0.96 804.00 986.00 298.00 440.00 0.18 0.03 0.00 0.01 0.11 0.35 0.13 0.19
  1. 0.94 520.00 724.00 379.00 522.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09 0.50
  1. 0.78 593.00 741.00 205.00 304.00 0.12 0.03 0.00 0.00 0.06 0.20 0.11 0.47
  1. 0.64 415.00 585.00 267.00 382.00 0.02 0.00 0.00 0.00 0.10 0.56 0.20 0.11
  1. 0.39 311.00 429.00 200.00 286.00 0.05 0.00 0.00 0.01 0.15 0.57 0.15 0.06
objects (6), predicted new locations:
fix for Hungarian algorithm:
assignment:
object 1 matched to image object 1
image object 1 updated:
object 2 created: car  1.00 569 811 573 721 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04
object 3 created: car  1.00 573.00 811.00 572.00 721.00 0.02 0.00 0.01 0.03 0.19 0.65 0.05 0.03
object 4 created: car  1.00 573.00 811.00 572.00 721.00 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04
object 5 matched to image object 3
image object 3 updated:
object 6 created: car  0.94 520.00 724.00 379.00 522.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09 0.50
object 7 created: car  0.96 804.00 986.00 298.00 440.00 0.18 0.03 0.00 0.01 0.11 0.35 0.13 0.19
object 8 matched to image object 2
image object 2 updated:
object 9 created: car  0.95 511.00 730.00 384.00 531.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09 0.50
object 10 matched to image object 4
image object 4 updated:
object 11 created: car  0.78 593.00 741.00 205.00 304.00 0.12 0.03 0.00 0.00 0.06 0.20 0.11 0.47
object 12 created: car  0.77 595.00 743.00 204.00 306.00 0.09 0.02 0.00 0.00 0.06 0.21 0.15 0.47 0.01

```

- Detailed log for every frame
Trace file for external analysis

GitHub

SakariLampola/Thesis · <https://github.com/SakariLampola/Thesis>

This repository Search Pull requests Issues Marketplace Explore

SakariLampola / Thesis

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

SW for my thesis, Image-Based Situation Awareness

Add topics

17 commits 1 branch 0 releases 1 contributor GPL-3.0

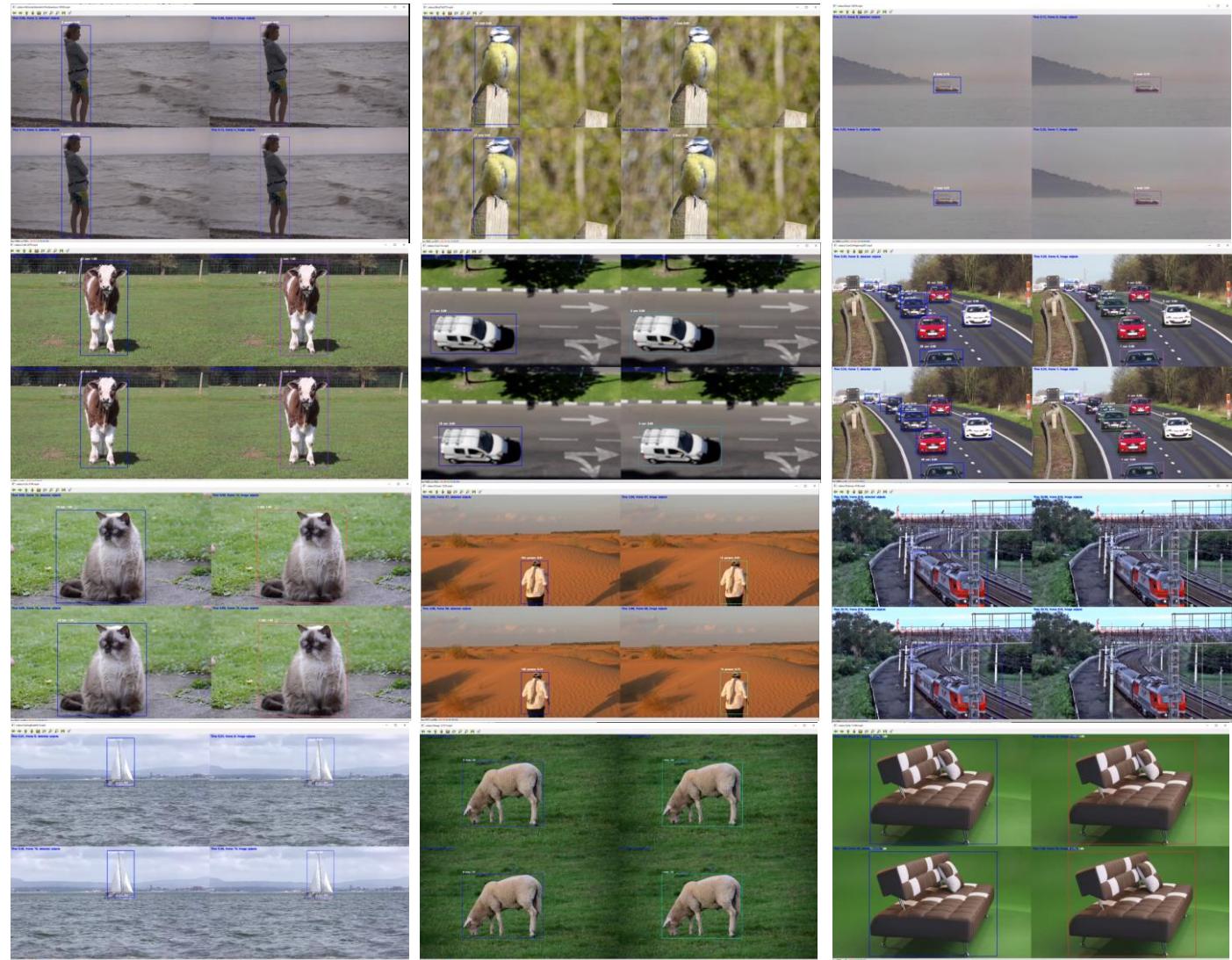
Branch: master New pull request Create new file Upload files Find file Clone or download

SakariLampola 20171220 ... Latest commit 9d5ca0e a day ago

File	Commit Date	Last Modified
.spyproject	20171211	10 days ago
pycache	20171220	a day ago
docs	20171218	3 days ago
matlab	20171220	a day ago
papers	20171129c	22 days ago
presentations	20171208	13 days ago
videos	20171220	a day ago
.gitattributes	Initial commit	23 days ago
AnalyzeVideo.py	20171220	a day ago
ImageClasses.py	20171220	a day ago
LICENSE	Initial commit	23 days ago
MobileNetSSD_deploy.caffemodel	Initial commit	23 days ago
MobileNetSSD_deploy.prototxt.txt	Initial commit	23 days ago
ObjectDetection.py	20171211	10 days ago
README.md	20171211	10 days ago
ToDo.txt	20171213	8 days ago
log.txt	20171220	a day ago
trace.txt	20171220	a day ago

<https://github.com/SakariLampola/Thesis>

Demos



YouTube channel: Sakari Lampola

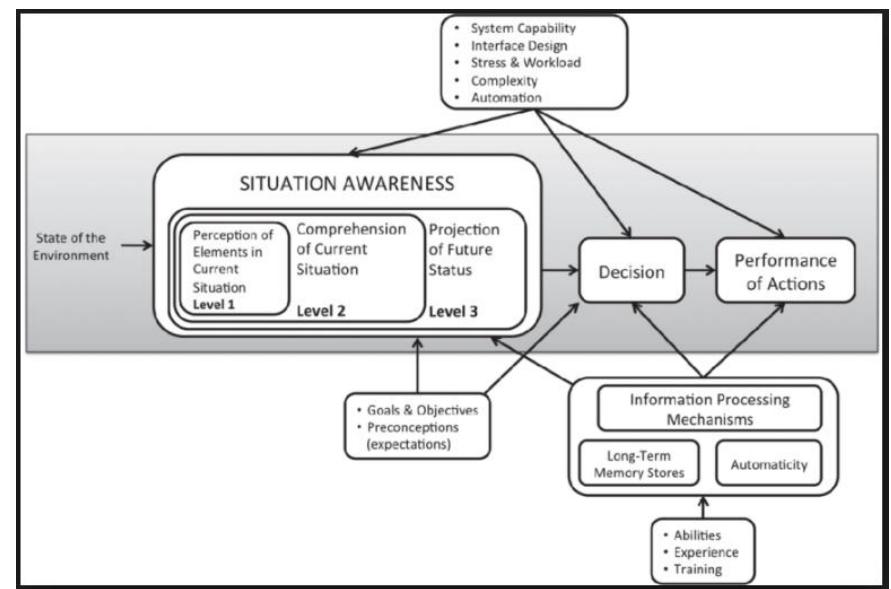
Situation Awareness

Definition

Wikipedia:

Situational awareness or **situation awareness** (SA) is the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their status after some variable has changed, such as time, or some other variable, such as a predetermined event.

It is also a field of study concerned with understanding of the environment critical to decision-makers in complex, dynamic areas from aviation, air traffic control, ship navigation, power plant operations, military command and control, and emergency services such as fire fighting and policing; to more ordinary but nevertheless complex tasks such as driving an automobile or riding a bicycle.



Levels

Endsley's model:

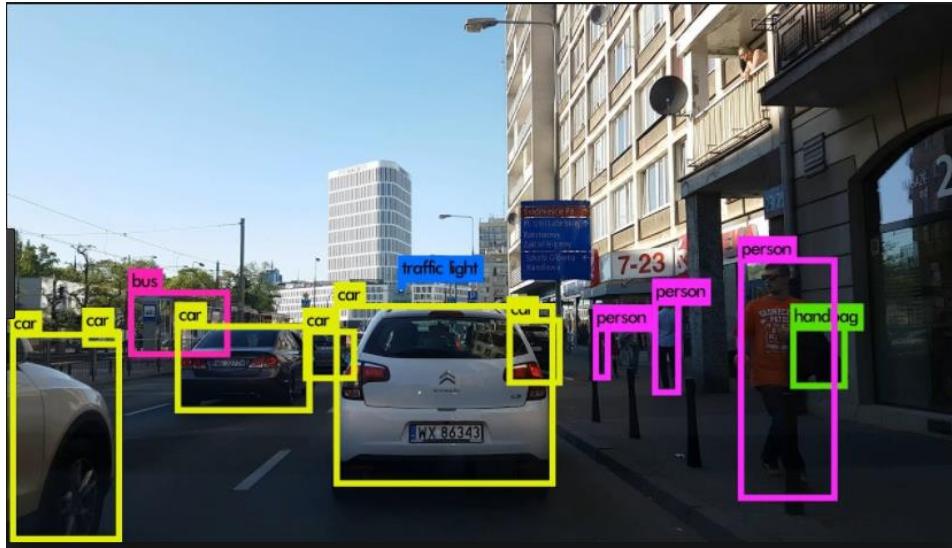
1. Perception of the elements in the current situation
2. Comprehension of the current situation
3. Projection of future status of the situation



N. A. Stanton, P. R. G. Chambers, and J. Piggott, "Situational awareness and safety," Safety Sci., vol. 39, no. 3, pp. 189–204, 2001

Perception

Perception of the elements in the current situation



Object detection

Algorithms:

- Faster R-CNN
- YOLO
- SSD
- ...

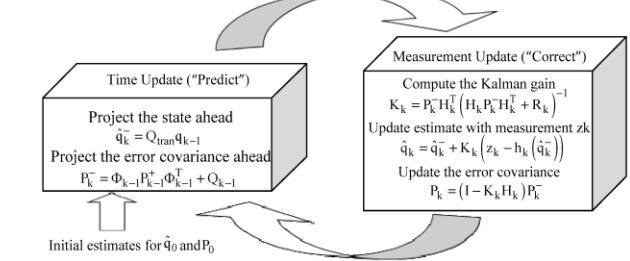
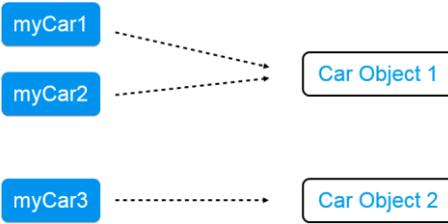
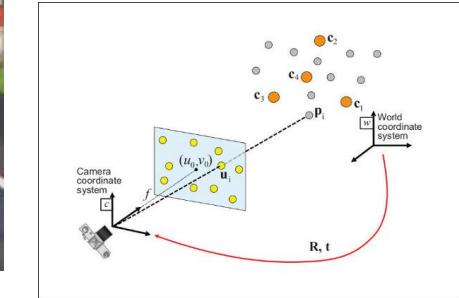
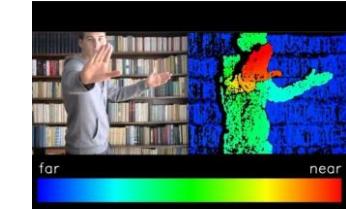
Output

- Classification
- Bounding box
- Reliability index

"The first step in achieving SA is to perceive the status, attributes, and dynamics of relevant elements in the environment. Thus, Level 1 SA, the most basic level of SA, involves the processes of monitoring, cue detection, and simple recognition, which lead to an awareness of multiple situational elements (objects, events, people, systems, environmental factors) and their current states (locations, conditions, modes, actions)."

Comprehension

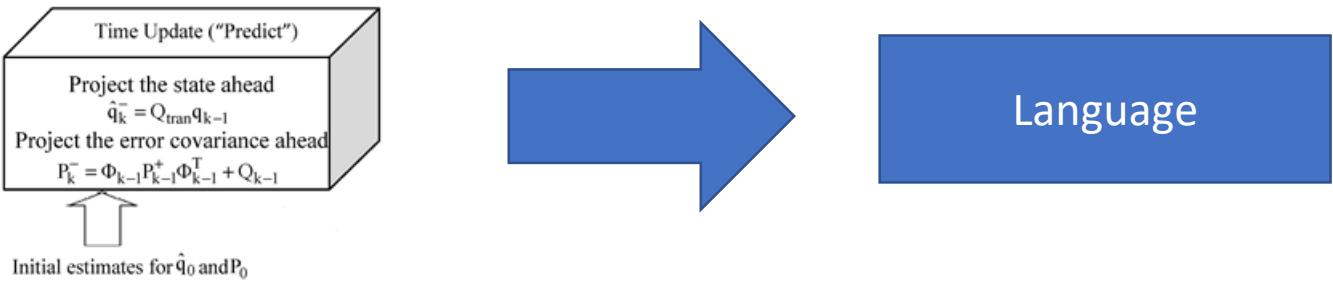
Comprehension of the current situation



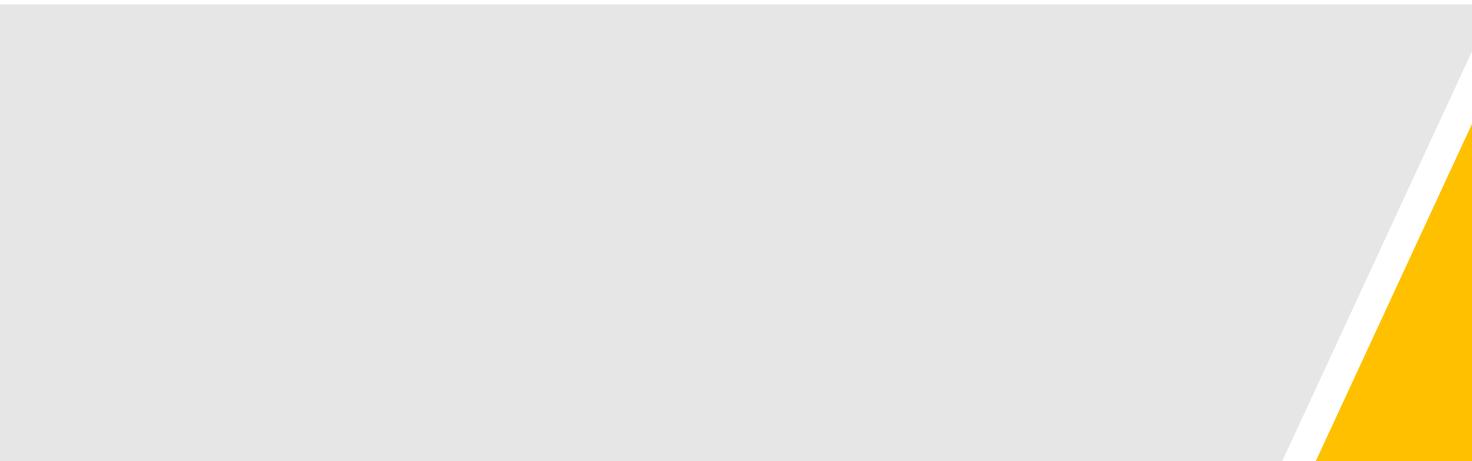
"The next step in SA formation involves a synthesis of disjointed Level 1 SA elements through the processes of pattern recognition, interpretation, and evaluation. Level 2 SA requires integrating this information to understand how it will impact upon the individual's goals and objectives. This includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual."

Projection

Projection of future status of the situation



"The third and highest level of SA involves the ability to project the future actions of the elements in the environment. Level 3 SA is achieved through knowledge of the status and dynamics of the elements and comprehension of the situation (Levels 1 and 2 SA), and then extrapolating this information forward in time to determine how it will affect future states of the operational environment."



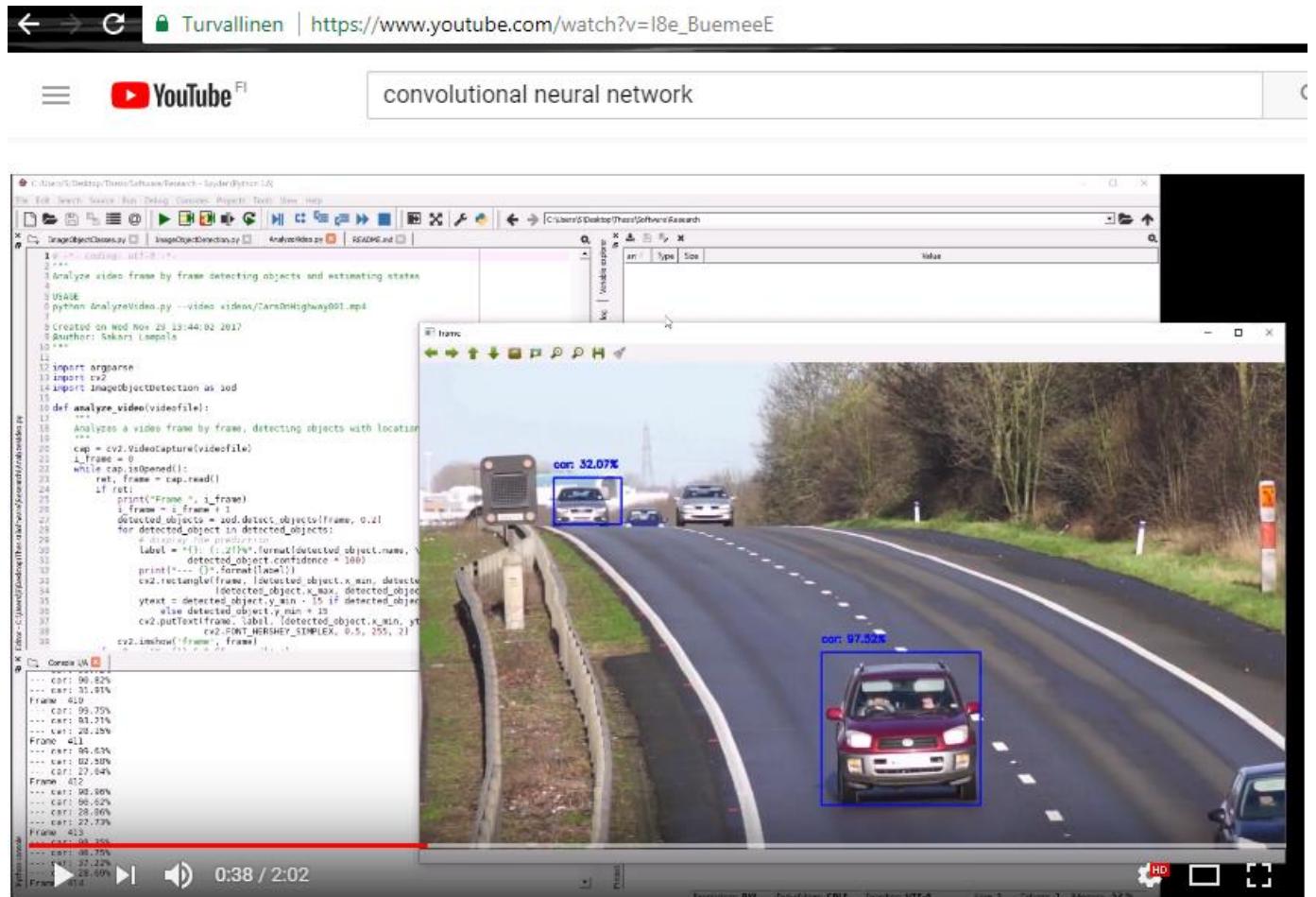
Work in Progress

Perception

“The first step in achieving SA is to perceive the status, attributes, and dynamics of relevant elements in the environment. Thus, Level 1 SA, the most basic level of SA, involves the processes of monitoring, cue detection, and simple recognition, which lead to an awareness of multiple situational elements (objects, events, people, systems, environmental factors) and their current states (locations, conditions, modes, actions).”

Perception

Object detection



Object Detection Using SSD and MobileNets

Perception

SSD + MobileNet

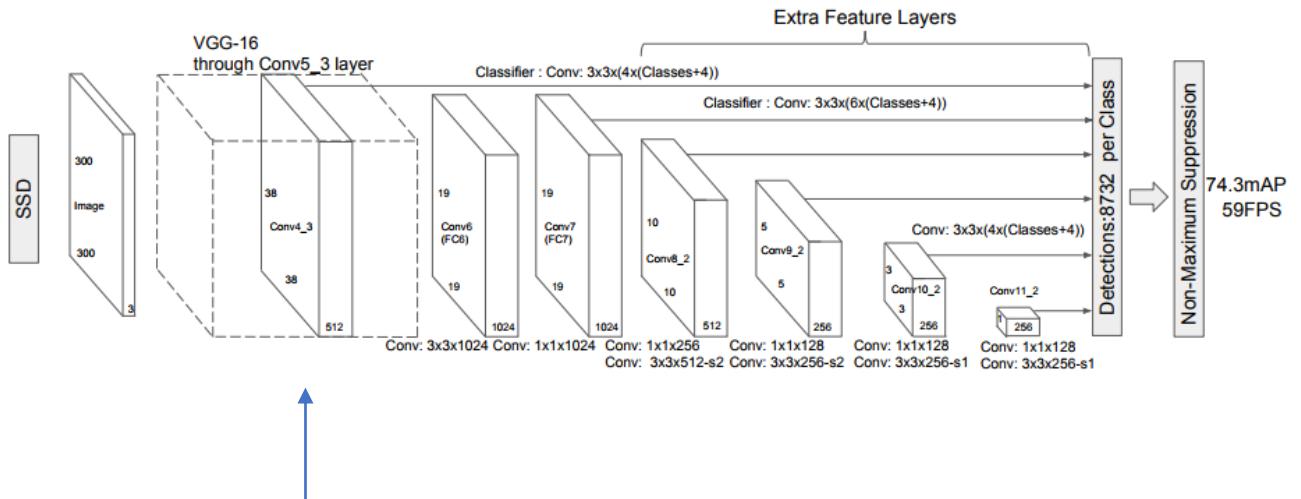


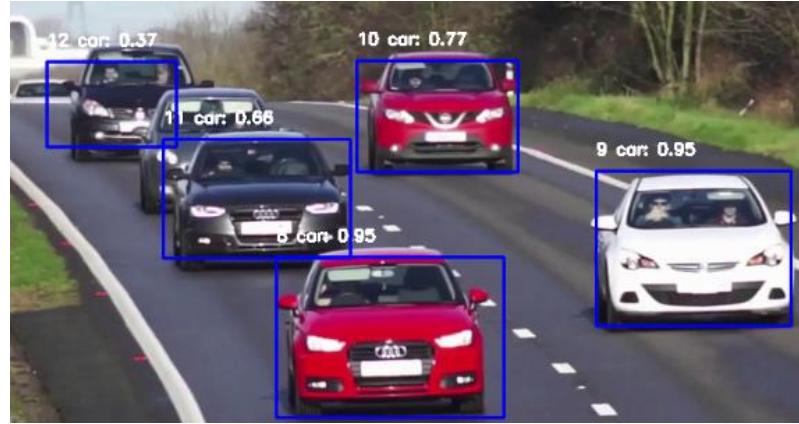
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

```
CLASS_NAMES = ["background", "aeroplane", "bicycle", "bird", "boat",  
"bottle", "bus", "car", "cat", "chair", "cow", "diningtable",  
"dog", "horse", "motorbike", "person", "pottedplant", "sheep",  
"sofa", "train", "tvmonitor"]
```

Perception

Result



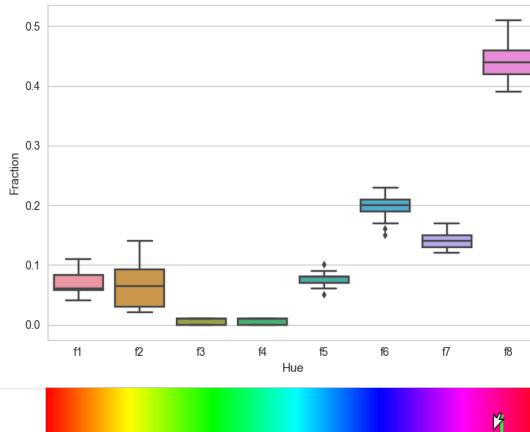
Detected object

- x_min
- x_max
- y_min
- y_max
- class
- confidence
- +- appearance

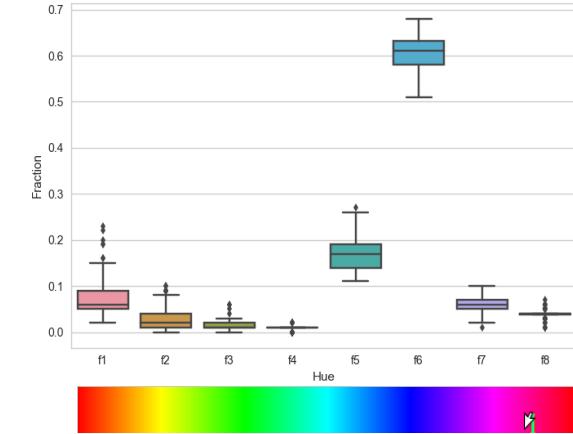
Perception

Appearance : hue histogram, 8 bins

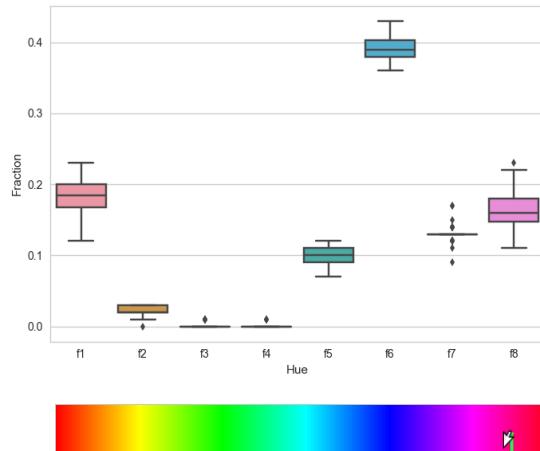
Red car



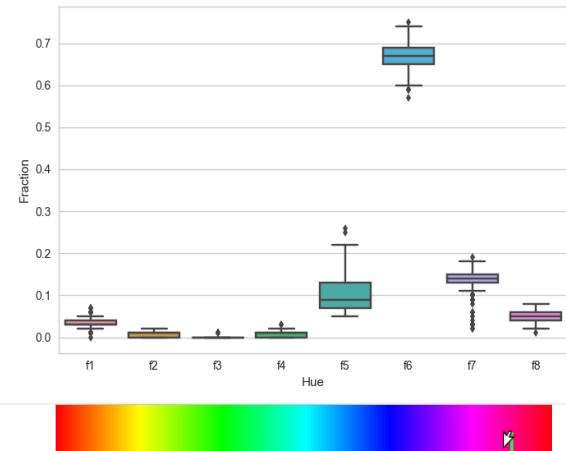
Blue car



White car



Black car



Perception

Appearance : hue histogram, 8 bins

RGB → HSV

Why hue?

- V (value) is sensitive to lighting, shadows
- S (saturation) is sensitive to dust, fog, distance

Why 8 bins?

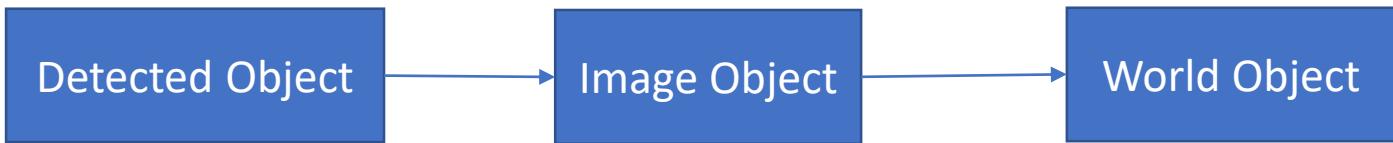
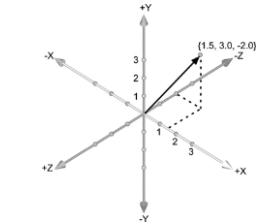
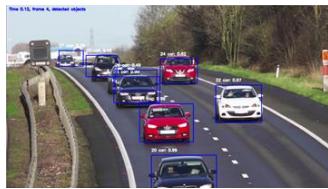
- Arbitrary decision
- Value range 0...179
- Used in other CV algorithms (Sift)

Comprehension

“The next step in SA formation involves a synthesis of disjointed Level 1 SA elements through the processes of pattern recognition, interpretation, and evaluation. Level 2 SA requires integrating this information to understand how it will impact upon the individual's goals and objectives. This includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual.”

Comprehension

Entity definitions



Detected object:

Output from an object detector, no persistency. Measurement.

Image object:

Persistent object created and updated by detected objects.

World object:

Object living in a 3d world with state: including location, velocity and acceleration. Updated by projected image object.

Comprehension

Image object identity



Perception

Image object

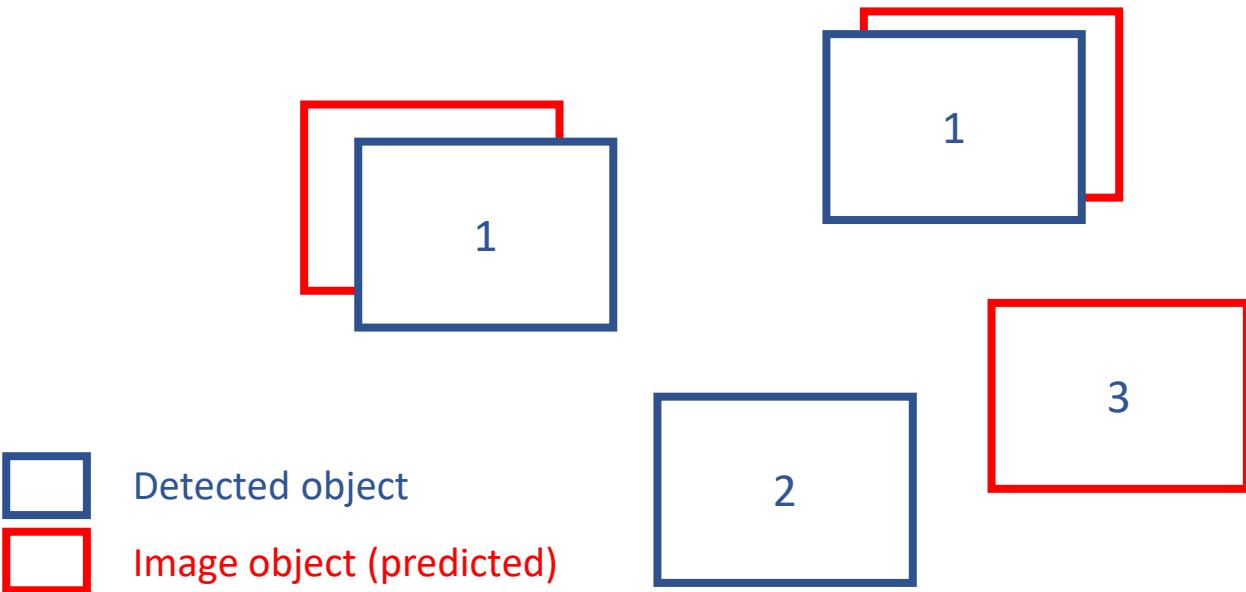


Image object

- id
- status
- x_min
- x_max
- y_min
- y_max
- vx_min
- vx_max
- vy_min
- vy_max
- class
- confidence
- appearance

Comprehension

Matching detected and image objects



1. Detected object and image object are "near" each other. Image object updated according to the detected object.
2. Detected object without any nearby image objects. A new image object is created according to the detected object.
3. Image object without any nearby detected objects. The image object status is changed to 'Hidden'. Movement prediction continues until out of frame.

Comprehension

Distance metrics



- Detected object
- Image object (predicted)

```
def distance(self, image_object):
    """
    Calculates distance (4 corners) to a predicted image object
    """
    dx_min = abs(self.x_min - image_object.x_min)
    dx_max = abs(self.x_max - image_object.x_max)
    dy_min = abs(self.y_min - image_object.y_min)
    dy_max = abs(self.y_max - image_object.y_max)

    horizontal_size = max(image_object.x_max - image_object.x_min, 1)
    vertical_size = max(image_object.y_max - image_object.y_min, 1)

    dx_min /= horizontal_size
    dx_max /= horizontal_size
    dy_min /= vertical_size
    dy_max /= vertical_size

    return (dx_min + dx_max + dy_min + dy_max) / 4.0 # Average of 4 corners
```

Comprehension

Probabilistic approach

Feature vector f

- x_{\min}
- x_{\max}
- y_{\min}
- y_{\max}
- class ??
- confidence
- appearance

$$P(\text{do} = \text{io}_i) \sim N(f_{\text{io}} - f_{\text{do},i} | 0, V_f)$$

- V_f estimated using comprehensive simulation
- $-\log(P)$ used as distance (cost)

Detected object (do)

- x_{\min}
- x_{\max}
- y_{\min}
- y_{\max}
- class
- confidence
- +- appearance

Image object 1 (io_1)

- id
- status
- x_{\min}
- x_{\max}
- y_{\min}
- y_{\max}
- vx_{\min}
- vx_{\max}
- vy_{\min}
- vy_{\max}
- class
- confidence
- appearance

Image object 2 (io_2)

- id
- status
- x_{\min}
- x_{\max}
- y_{\min}
- y_{\max}
- vx_{\min}
- vx_{\max}
- vy_{\min}
- vy_{\max}
- class
- confidence
- appearance



Comprehension

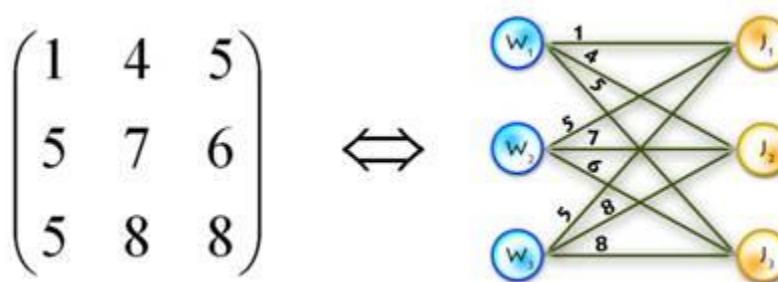
Matching algorithm

Hungarian algorithm

From Wikipedia, the free encyclopedia

The **Hungarian method** is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal-dual methods. It was developed and published in 1955 by Harold Kuhn, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Dénes König and Jenő Egerváry.^{[1][2]}

James Munkres reviewed the algorithm in 1957 and observed that it is (strongly) polynomial.^[3] Since then the algorithm has been known also as the **Kuhn–Munkres algorithm** or **Munkres assignment algorithm**. The time complexity of the original algorithm was $O(n^4)$, however Edmonds and Karp, and independently Tomizawa noticed that it can be modified to achieve an $O(n^3)$ running time. Ford and Fulkerson extended the method to general transportation problems.^[citation needed] In 2006, it was discovered that Carl Gustav Jacobi had solved the assignment problem in the 19th century, and the solution had been published posthumously in 1890 in Latin.^[4]



scipy.optimize.linear_sum_assignment

`scipy.optimize.linear_sum_assignment(cost_matrix)`

[\[source\]](#)

Solve the linear sum assignment problem.

The linear sum assignment problem is also known as minimum weight matching in bipartite graphs. A problem instance is described by a matrix C , where each $C[i,j]$ is the cost of matching vertex i of the first partite set (a "worker") and vertex j of the second set (a "job"). The goal is to find a complete assignment of workers to jobs of minimal cost.

Formally, let X be a boolean matrix where $X[i, j] = 1$ iff row i is assigned to column j . Then the optimal assignment has cost

$$\min \sum_i \sum_j C_{i,j} X_{i,j}$$

Comprehension

Matching algorithm

```
-----  
Time 0.04, frame 2  
Detected objects (6):  
---7 car 1.00 573 811 572 721 0.02 0.00 0.01 0.03 0.19 0.65 0.05 0.03  
---8 car 0.95 521 730 384 531 0.02 0.00 0.00 0.01 0.06 0.31 0.11 0.49  
---9 car 0.95 815 994 305 447 0.16 0.03 0.01 0.00 0.08 0.37 0.18 0.17  
---10 car 0.77 595 743 204 306 0.08 0.02 0.00 0.00 0.06 0.21 0.15 0.47  
---11 car 0.66 417 589 276 385 0.01 0.00 0.00 0.00 0.08 0.59 0.23 0.09  
---12 car 0.37 310 429 205 283 0.05 0.00 0.00 0.00 0.15 0.58 0.16 0.05  
Image objects (6), predicted new locations:  
---1 car 1.00 569.00 811.00 573.00 721.00 0.02 0.01 0.01 0.03 0.20 0.64 0.06  
---2 car 0.96 804.00 986.00 298.00 440.00 0.18 0.03 0.00 0.01 0.11 0.35 0.13  
---3 car 0.94 520.00 724.00 379.00 522.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09  
---4 car 0.78 593.00 741.00 205.00 304.00 0.12 0.03 0.00 0.00 0.06 0.20 0.11  
---5 car 0.64 415.00 585.00 267.00 382.00 0.02 0.00 0.00 0.00 0.10 0.56 0.20  
---6 car 0.39 311.00 429.00 200.00 286.00 0.05 0.00 0.00 0.00 0.01 0.15 0.57 0.15  
Cost matrix for Hungarian algorithm:  
 0.01 1.53 0.86 2.13 1.96 3.71  
 0.77 1.05 0.03 1.17 0.95 2.33  
 1.36 0.05 0.95 1.42 1.41 3.04  
 1.42 1.02 0.80 0.01 0.80 1.34  
 1.46 1.21 0.71 0.94 0.03 1.07  
 2.02 1.88 1.34 1.06 0.73 0.03  
Optimal assignment:  
Detected object 7 matched to image object 1  
Existing image object 1 updated:  
---predicted: car 1.00 569.00 811.00 573.00 721.00 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04  
---measured: car 1.00 573.00 811.00 572.00 721.00 0.02 0.00 0.01 0.03 0.19 0.65 0.05 0.03 0.01  
---corrected: car 1.00 573.00 811.00 572.00 721.00 0.02 0.01 0.01 0.03 0.20 0.64 0.06 0.04  
Detected object 8 matched to image object 3  
Existing image object 3 updated:  
---predicted: car 0.94 520.00 724.00 379.00 522.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09 0.50  
---measured: car 0.95 521.00 730.00 384.00 531.00 0.02 0.00 0.00 0.01 0.06 0.31 0.11 0.49 0.03  
---corrected: car 0.95 521.00 730.00 384.00 531.00 0.02 0.00 0.00 0.02 0.07 0.29 0.09 0.50  
Detected object 9 matched to image object 2  
Existing image object 2 updated:  
---predicted: car 0.96 804.00 986.00 298.00 440.00 0.18 0.03 0.00 0.01 0.11 0.35 0.13 0.19
```

Comprehension

Matching algorithm

Open question: Role of classical algorithms?

- MeanShift
- CamShift
- Optical flow (Lucas-Kanade)
- Background Subtraction

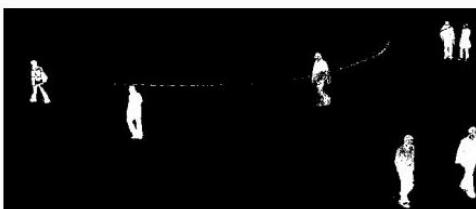
Possible added value:

- Refined object region
- Support for occasionally missing bounding boxes

Below image shows the 200th frame of a video



Result of BackgroundSubtractorMOG



Comprehension

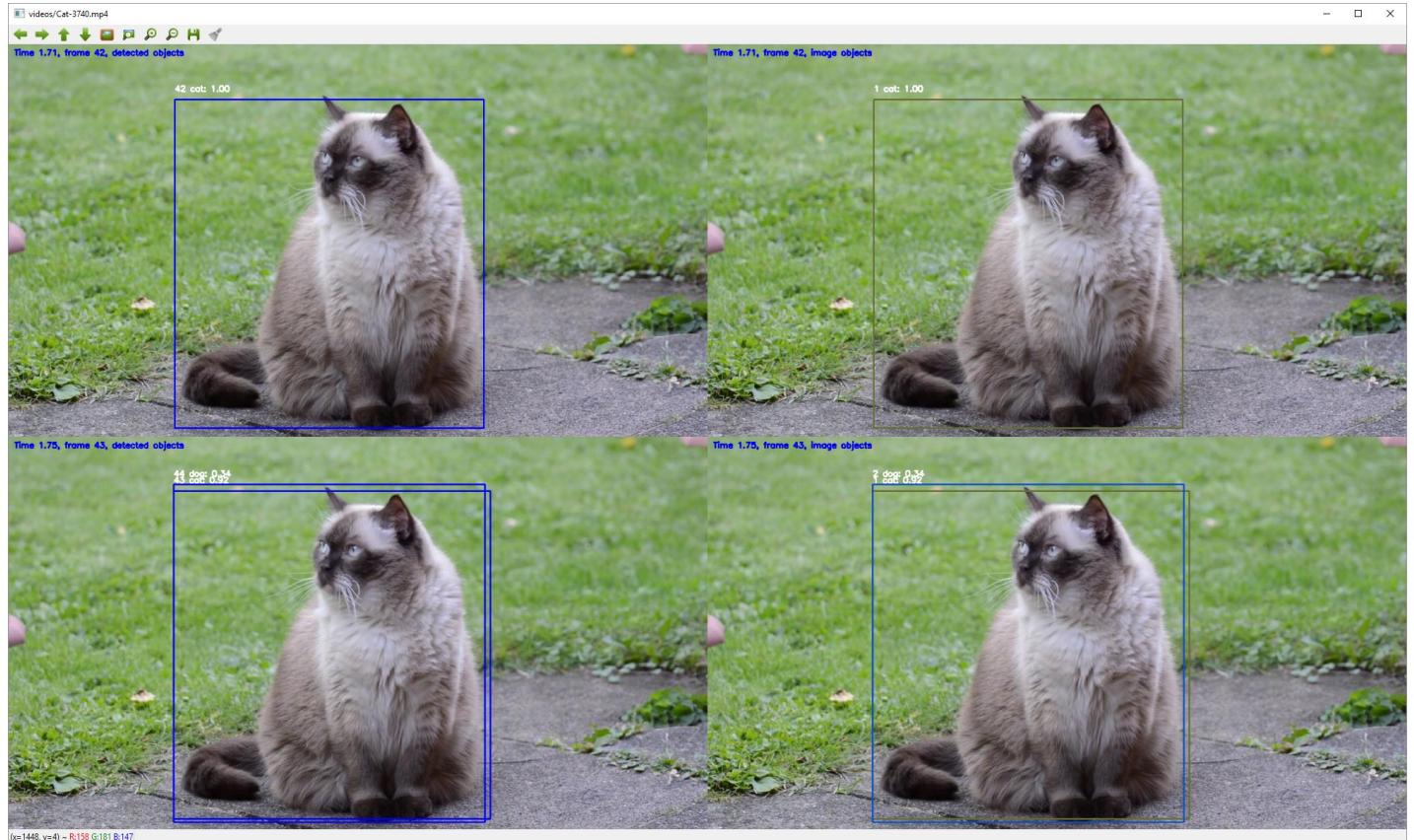
Special cases / Vanishing detected objects



Solution: Image objects have statuses. Their location is predicted according to the previous location and velocity. When no matching detected objects exist, image objects are not immediately removed, but their status is changed ('detected'->'hidden'). Prediction continues in following frames.

Comprehension

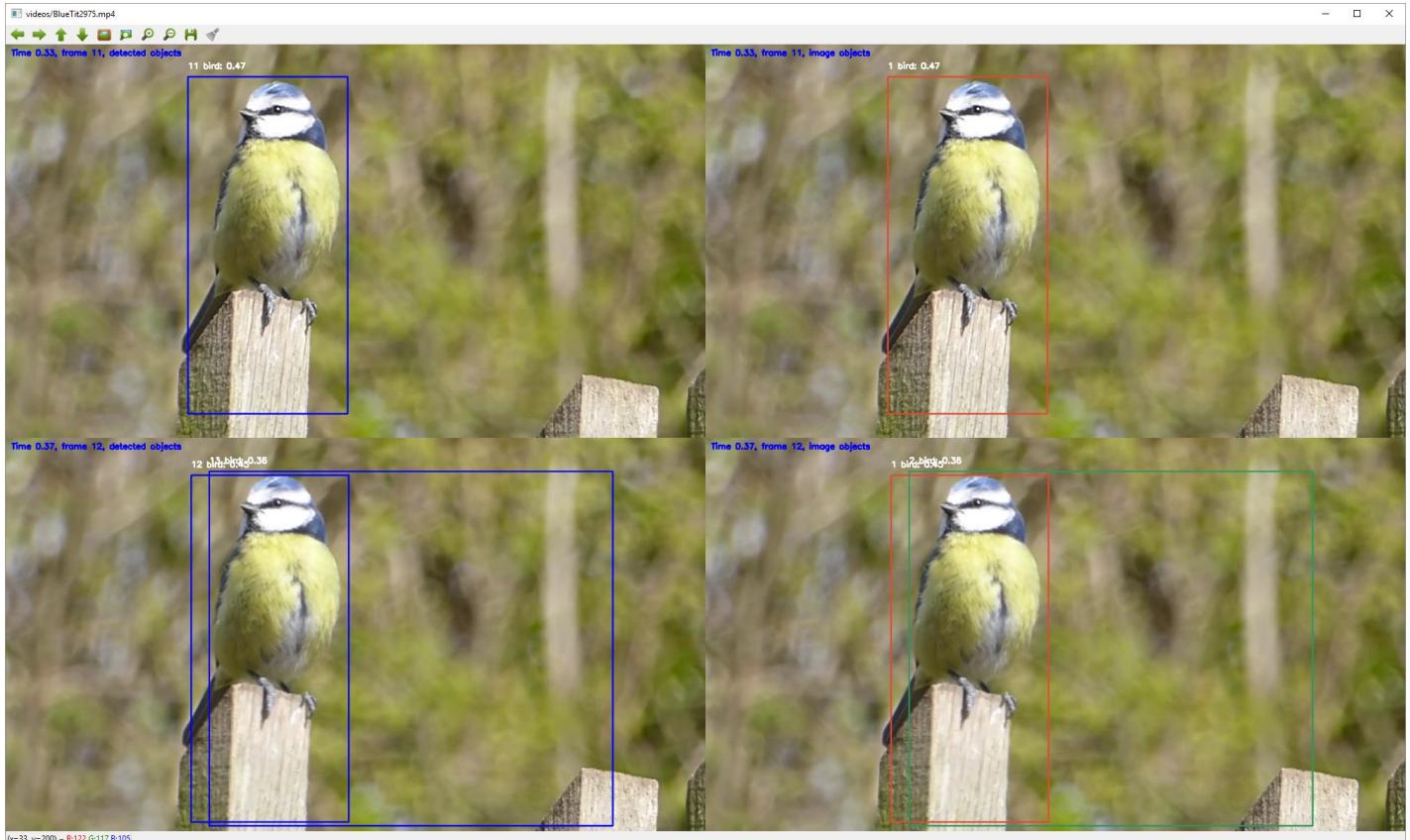
Special cases / Multiple classifications



Solution: The classification with highest confidence is used if multiple detected objects correspond to an image object and no other image object is nearby. The information is available in the Hungarian algorithm cost matrix.

Comprehension

Special cases / Multiple bounding boxes (a)

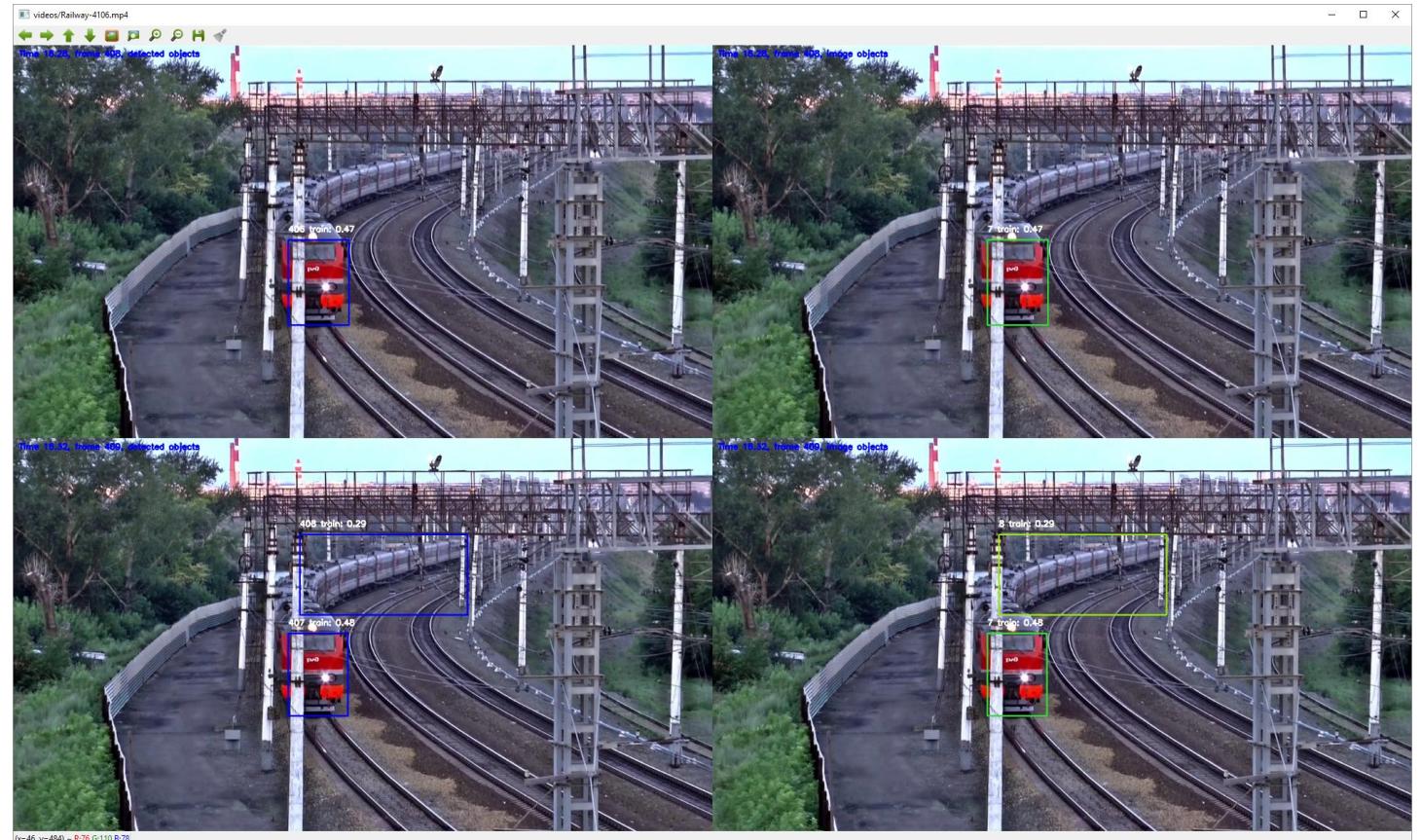


Solution: ?

Open question: What to do?

Comprehension

Special cases / Multiple bounding boxes (b)



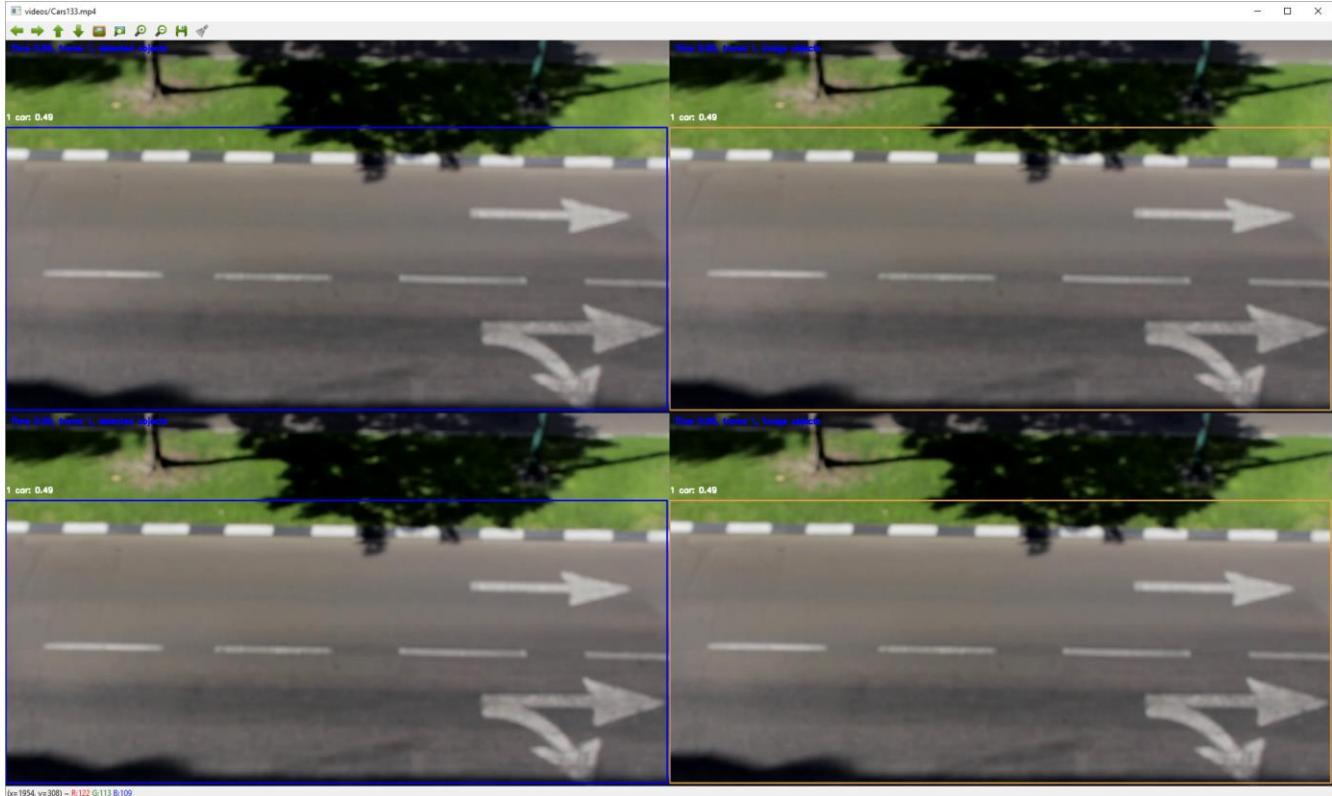
Comprehension

Special cases / Multiple bounding boxes (c)



Comprehension

Special cases / Typical background elements are messed up with object in the classification. (a)

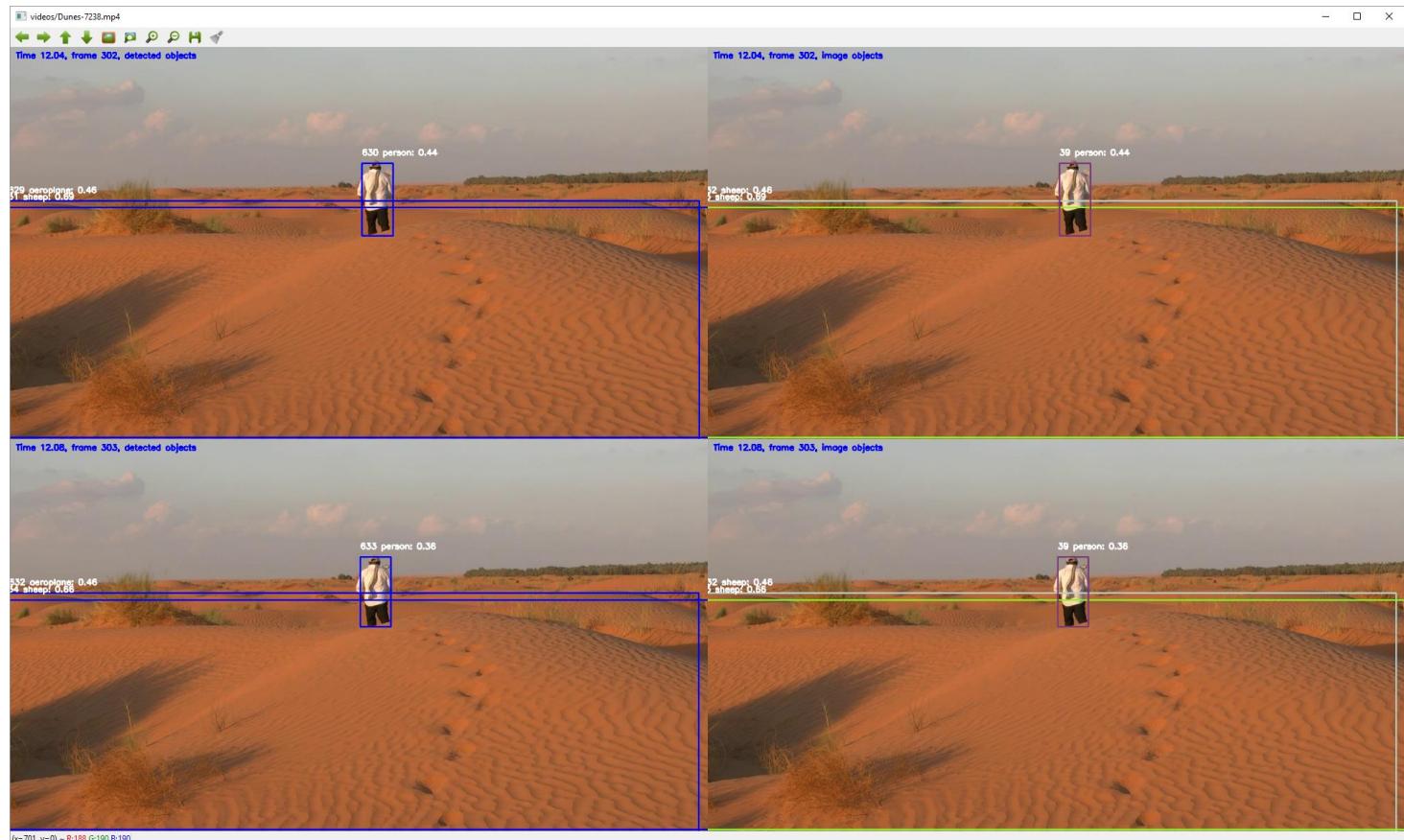


Solution: Classification confidence must be high enough for creating a new image object. Probabilistic estimate based on distance and a priori size.

Open question: How to determine the appropriate minimum confidence level?

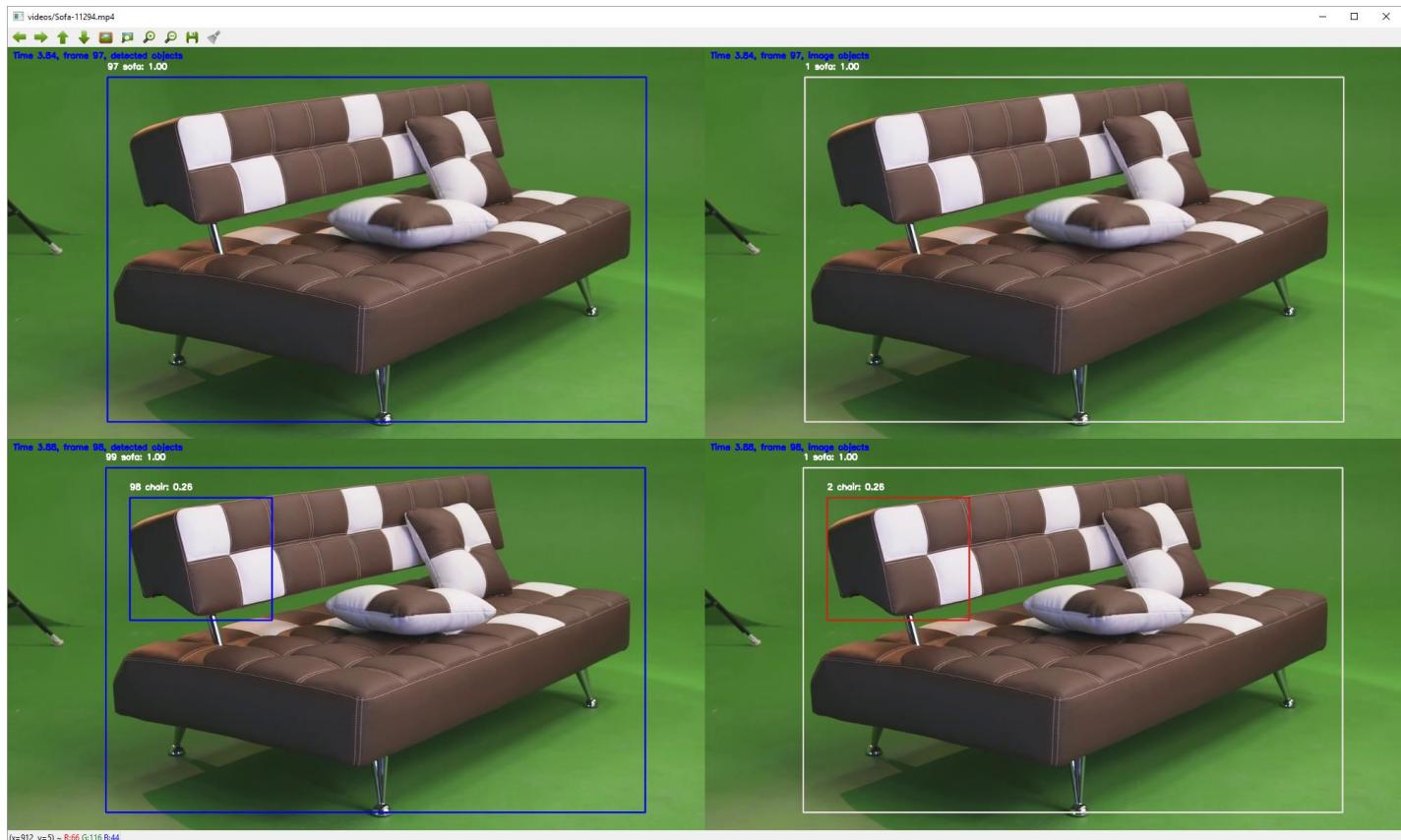
Comprehension

Special cases / Typical background elements are messed up with object in the classification. (b)



Comprehension

Special cases / False detections inside other objects



Solution: ?

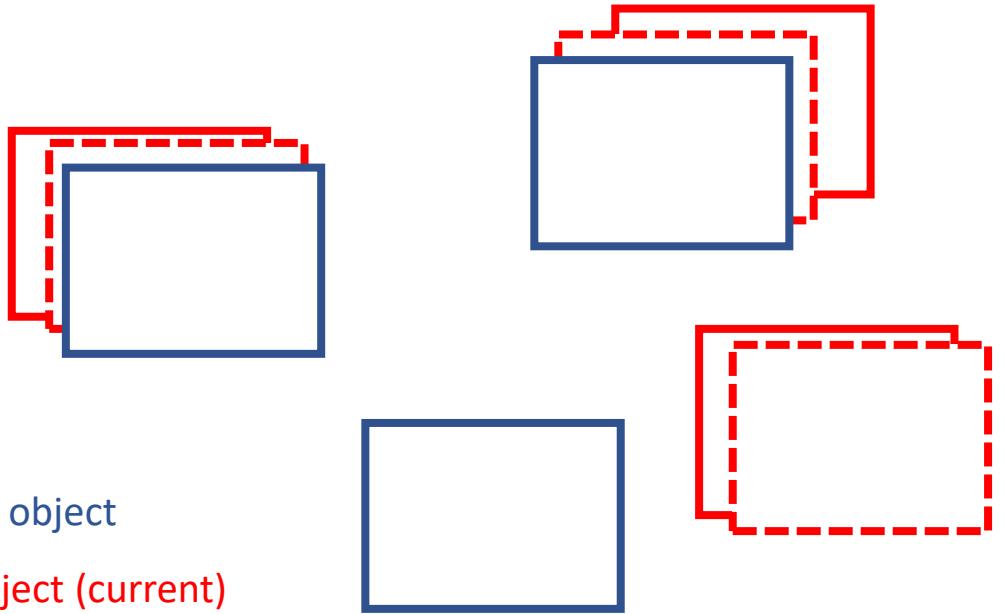
Open question: What to do?

Comprehension

Image object location and velocity prediction

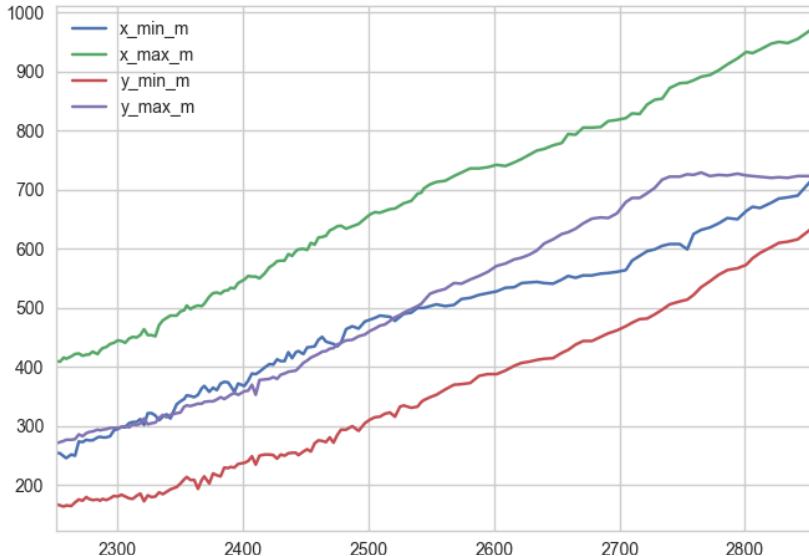


- Detected object
- Image object (current)
- Image object (predicted)

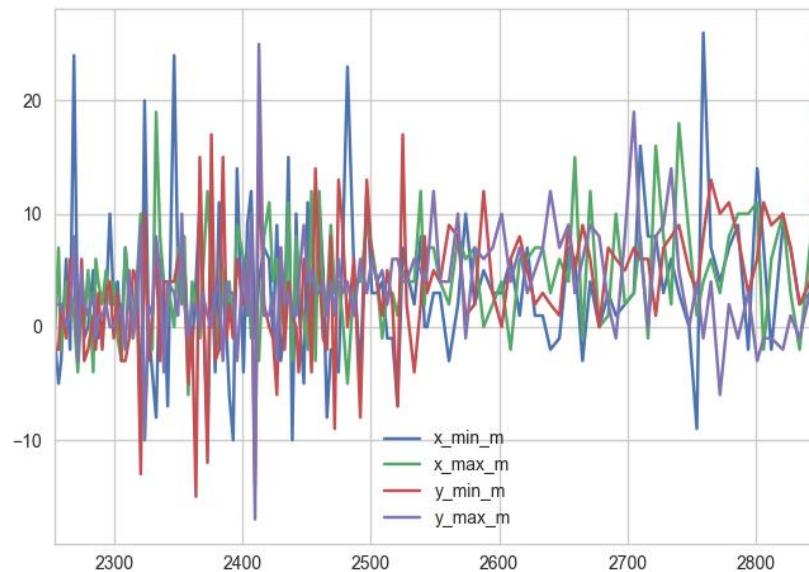


Comprehension

Image object location and velocity prediction



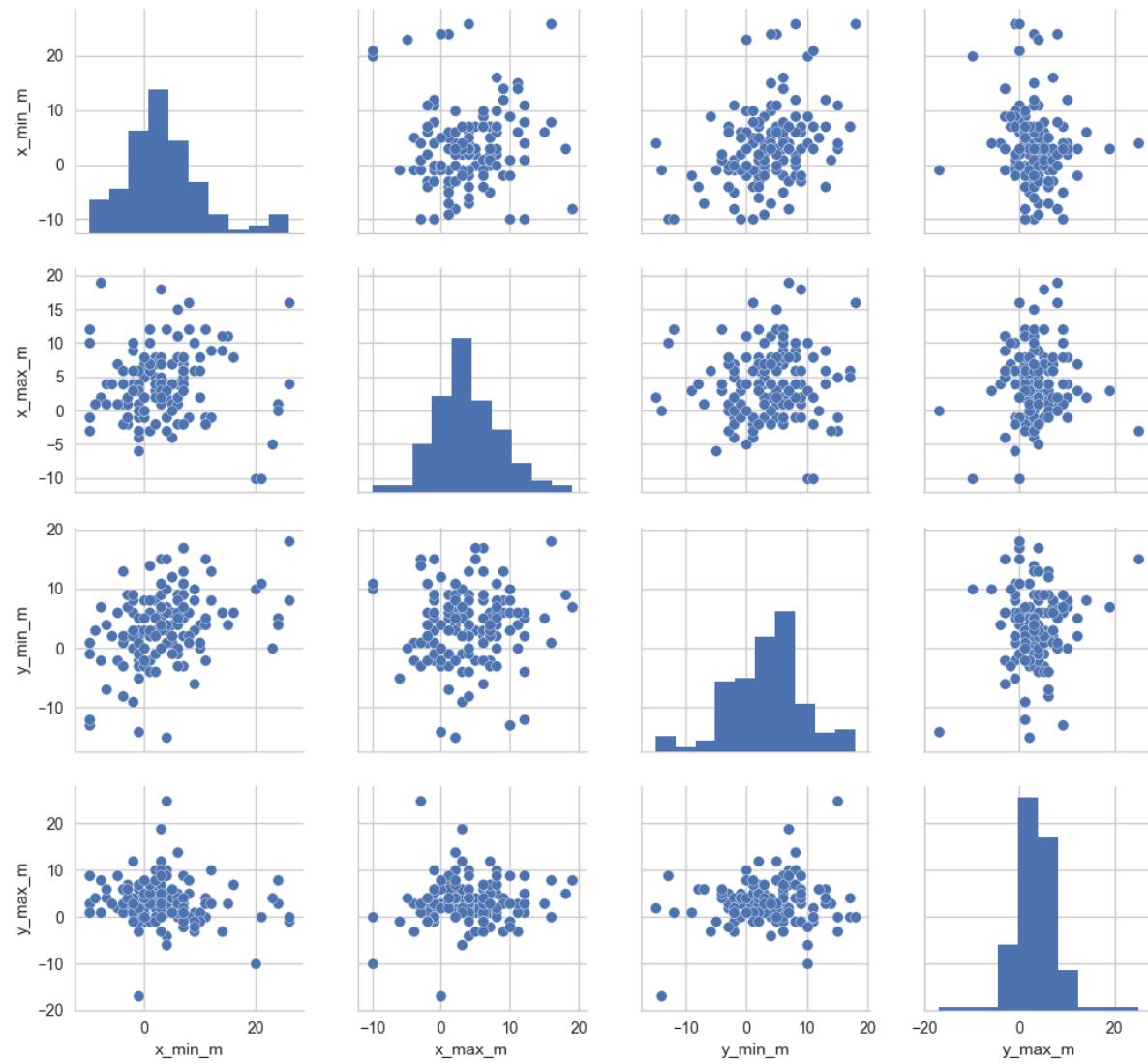
location



velocity

Comprehension

Image object location and velocity prediction



Comprehension

Image object location and velocity prediction

Bounding box corner location

State vector s :

$$s = \begin{bmatrix} l \\ v \end{bmatrix}$$

where

l = location coordinate ($x_{\min}, x_{\max}, y_{\min}, y_{\max}$) of the bounding box corner in the image

v = velocity ($v_{x\min}, v_{x\max}, v_{y\min}, v_{y\max}$) of the bounding box corner in the image

State equation in differential form:

$$\frac{ds(t)}{dt} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} * s(t) + \epsilon(t) = A_1 * s$$

State equation in difference form:

$$s(k+1) = (I + \Delta * A_1) * s(k) + \epsilon(k)$$

$$= \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} * s(k) + \epsilon(k) = A * s(k) + \epsilon(k)$$

where Δ is the time increment and ϵ Gaussian noise with covariance R .

Measurement equation

$$z(k) = [1 \ 0] * s(k) + \delta(k) = C * s(k) + \delta(k)$$

Where δ is Gaussian noise with covariance matrix Q .

Comprehension

Image object location and velocity prediction

Kalman filter initialization:

$$\mu(0) = \begin{bmatrix} l(0) \\ 0 \end{bmatrix}$$

where $l(0)$ is the first location measurement.

$$\Sigma(0) = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$$

where α , β and γ are believed variances of location and velocity.

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

where r_1 , r_2 and r_3 are believed state equation variances of location and velocity.

$$Q = [q]$$

Where q is the believed measurement variance, for example 1.0.

Kalman filter update:

$$\mu_1(k) = A * \mu(k - 1)$$

$$\Sigma_1(k) = A * \Sigma(k - 1) * A^T + R$$

$$K(k) = \Sigma_1(k) * C^T * (C * \Sigma_1(k) * C^T + Q)^{-1}$$

$$\mu(k) = \mu_1(k) + K(k) * (z(k) - C * \mu_1(k))$$

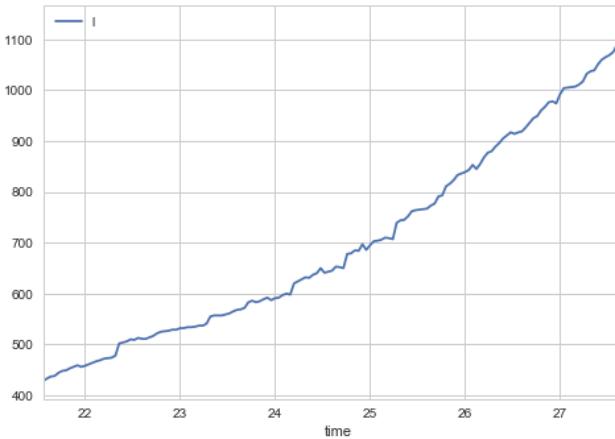
$$\Sigma(k) = (I - K(k) * C) * \Sigma_1(k)$$

Comprehension

Image object location and velocity prediction

```
In [39]: x_min.plot()
```

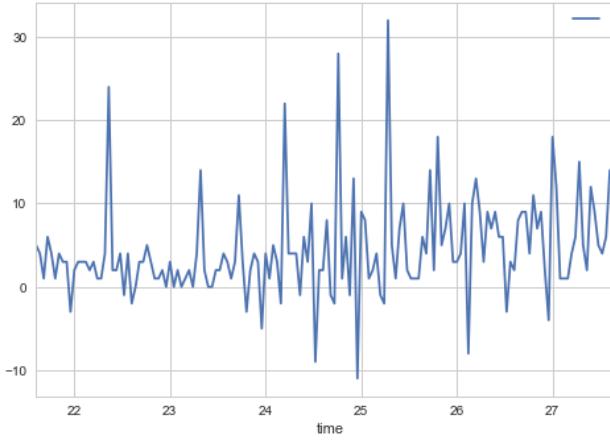
```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x20472a436d8>
```



location

```
In [40]: x_min_diff = x_min.diff(1)  
x_min_diff = x_min_diff.dropna()  
x_min_diff.plot()
```

```
Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x20472a5ee80>
```



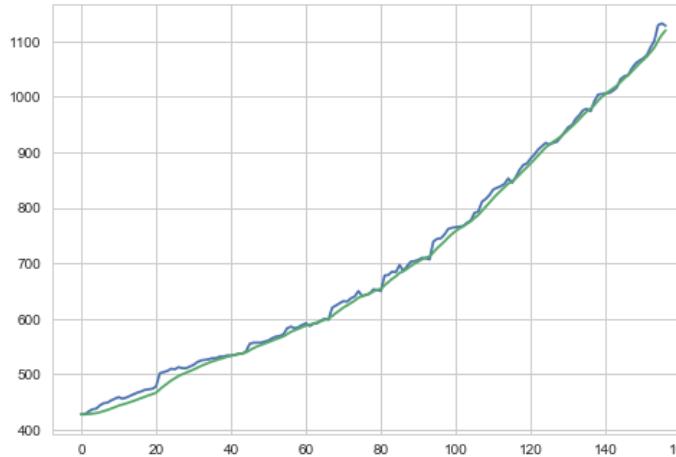
velocity

Comprehension

Image object location and velocity prediction

```
In [50]: plt.plot(x_min_m)
plt.plot(x_min_f)
```

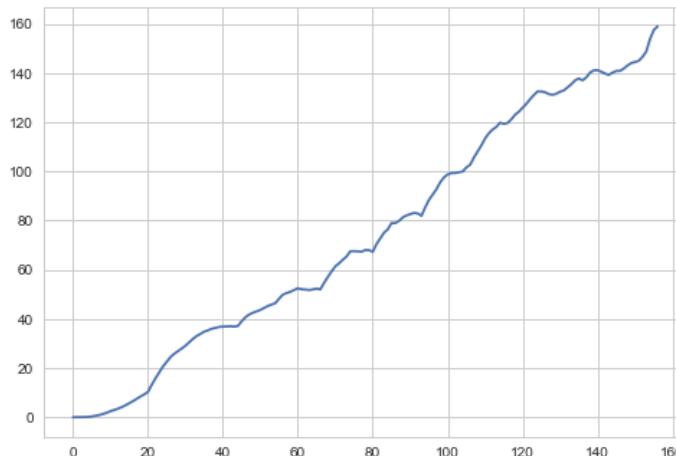
```
Out[50]: [<matplotlib.lines.Line2D at 0x204728d0f98>]
```



Location, measured and filtered

```
In [51]: plt.plot(vx_min_f)
```

```
Out[51]: [<matplotlib.lines.Line2D at 0x20472dfb978>]
```



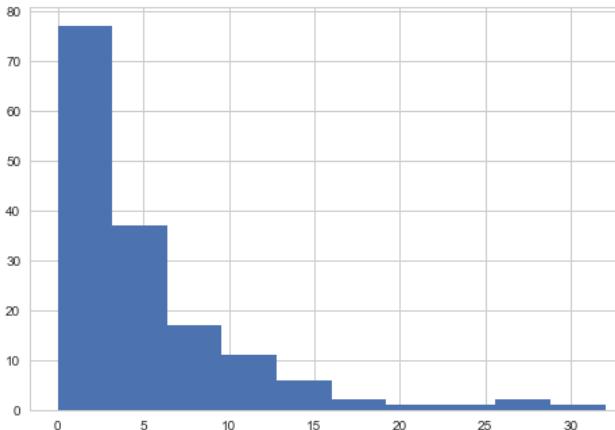
Estimated velocity

Comprehension

Image object location and velocity prediction

```
In [57]: plt.hist(x_min_error_prev)  
print(np.mean(x_min_error_prev))
```

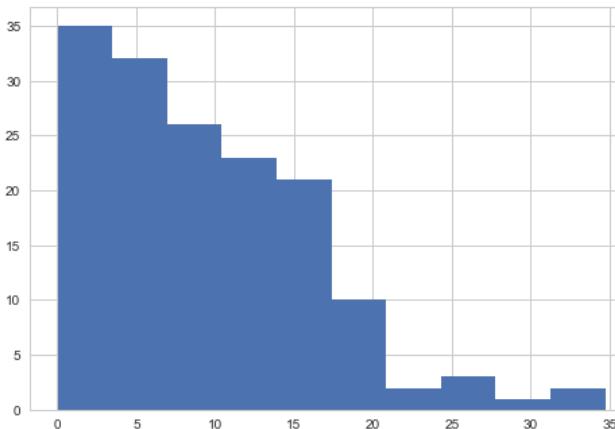
5.32903225806



Prediction error using previous measurement

```
In [58]: plt.hist(x_min_error_filter)  
print(np.mean(x_min_error_filter))
```

9.44186196322



Prediction error using filtered prediction

Comprehension

Image object location and velocity prediction

Open questions:

- How to set covariances and initialize sigma?
- How to predict when using previous measurement gives smaller average error? Bias!

(Notebook demo)

Next Steps

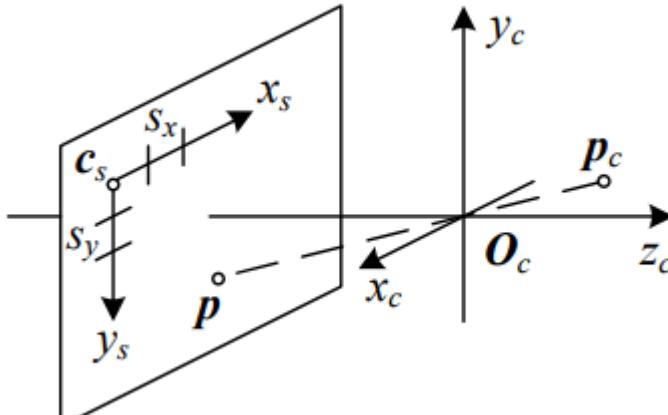
Next steps

Comprehension:

1. Closing the open questions
2. 2d -> 3d transformation
3. World object state estimation

Next steps

2d -> 3d transformation



$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\tilde{\mathbf{x}}_s = \alpha \mathbf{M}_s^{-1} \mathbf{p}_c = \mathbf{K} \mathbf{p}_c. \quad (2.54)$$

The 3×3 matrix \mathbf{K} is called the *calibration matrix* and describes the camera *intrinsics* (as opposed to the camera's orientation in space, which are called the *extrinsics*).

$$\tilde{\mathbf{x}}_s = \mathbf{K} \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right] \mathbf{p}_w = \mathbf{P} \mathbf{p}_w,$$

Open question: How to create the reverse transformation?

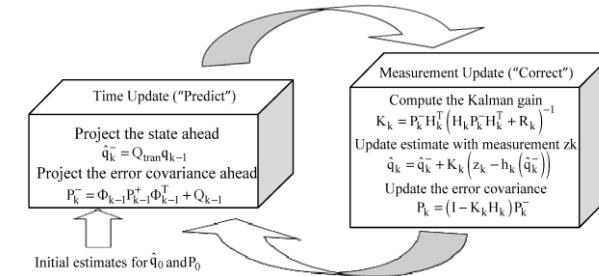
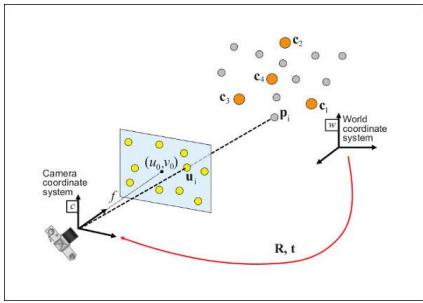
Next steps

2d -> 3d transformation (a priori size for distance estimation)

```
199 class Aeroplane(ImageObject):
200     """
201     Aeroplane - derived class
202     """
203     name = 'aeroplane'
204     class_type = 1
205     height_min, height_mean, height_max = 1.5, 2.0, 6.0
206     width_min, width_mean, width_max = 2.0, 20.0, 80.0
207     length_min, length_mean, length_max = 3.0, 20.0, 80.0
208     velocity_max, acceleration_max = 300.0, 10.0
209
210 class Bicycle(ImageObject):
211     """
212     Bicycle - derived class
213     """
214     name = 'bicycle'
215     class_type = 2
216     height_min, height_mean, height_max = 0.6, 1.1, 1.4
217     width_min, width_mean, width_max = 0.4, 0.6, 0.8
218     length_min, length_mean, length_max = 1.2, 1.5, 1.7
219     velocity_max, acceleration_max = 13.0, 1.0
220
221 class Bird(ImageObject):
222     """
223     Bird - derived class
224     """
225     name = 'bird'
226     class_type = 3
227     height_min, height_mean, height_max = 0.1, 0.25, 2.8
228     width_min, width_mean, width_max = 0.05, 0.5, 1.2
229     length_min, length_mean, length_max = 0.1, 0.2, 1.2
230     velocity_max, acceleration_max = 80.0, 100.0
231
```

Next steps

World object state estimation



State vector s :

$$s = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ a_x \\ a_y \\ a_z \end{bmatrix}$$

where

- (x, y, z) = location of the world object center point
- (v_x, v_y, v_z) = velocity of the object
- (a_x, a_y, a_z) = acceleration of the object

Next steps

World object state estimation

State equation in differential form:

$$\frac{ds(t)}{dt} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} * s(t) + \epsilon(t) = A_1 * s$$

State equation in difference form:

$$s(k+1) = (I + \Delta * A_1) * s(k) + \epsilon(k)$$

$$= \begin{bmatrix} 1 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * s(k) + \epsilon(k) = A * s(k) + \epsilon(k)$$

where Δ is the time increment and ϵ Gaussian noise with covariance R.

Measurement equation

$$z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} * s(k) + \delta(k) = C * s(k) + \delta(k)$$

Where δ is Gaussian noise with covariance matrix Q.

Next steps

World object state estimation

Kalman filter initialization:

$$\mu(0) = \begin{bmatrix} x(0) \\ 0 \\ 0 \\ y(0) \\ 0 \\ 0 \\ z(0) \\ 0 \\ 0 \end{bmatrix}$$

where $x(0), y(0), z(0)$ is the first location measurement.

$$\Sigma(0) = \begin{bmatrix} \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma \end{bmatrix}$$

where α, β and γ are believed variances of location, velocity and acceleration.

$$R = \begin{bmatrix} r_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_3 \end{bmatrix}$$

where r_1, r_2 and r_3 are believed variances of location, velocity and acceleration.

$$Q = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix}$$

Where q is the believed measurement variance.

Next steps

World object state estimation

Kalman filter update:

$$\mu_1(k) = A * \mu(k - 1)$$

$$\Sigma_1(k) = A * \Sigma(k - 1) * A^T + R$$

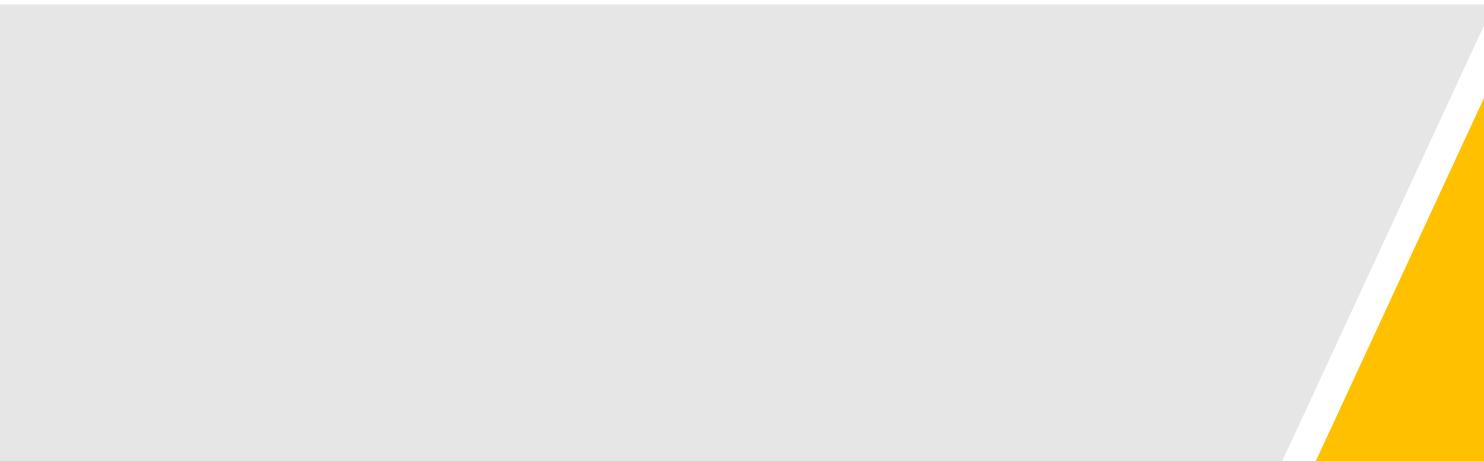
$$K(k) = \Sigma_1(k) * C^T (C * \Sigma_1(k) * C^T + Q)^{-1}$$

$$\mu(k) = \mu_1(k) + K(k) * (z(k) - C * \mu_1(k))$$

$$\Sigma(k) = (I - K(k) * C) * \Sigma_1(k)$$

Open questions:

1. Equations ok?
2. Initial values?
3. Different time step for prediction, not equal to inverse video frame rate?



To Be Discussed

Method follow-up

- Google search enough?
- Good way of following new papers?

Actions

- The role of actions, should it be considered?
- Navigation assistance?

Probability representation

- Probability there is something in the image
 - $P(S)$
- Probability of class given that there is something
 - $P(C|S)$
- Probability of location (or state) given the class
 - $P(X|C,S)$

What is the goal?

“Develop methods which can perceive the elements in the environment within a volume of time and space, comprehend their meaning and project their status in the near future. The methods will be tested on a system which will help visually impaired people to understand their environment better. The system will be demonstrated on existing hardware.”

Sources of uncertainty

Ingredients:

- Classification belief
- Bounding box noise
- Pixel discretization
- Distance estimation

1. Inherent stochasticity in the system being modeled.
 - ?
2. Incomplete observability.
 - ?
3. Incomplete modeling.
 - ?

Goal Fine Tuning

“Develop methods which can perceive the elements in the environment within a volume of time and space, comprehend their meaning and project their status in the near future. The methods will be tested on a system which will help visually impaired people to understand their environment better. The system will be demonstrated on existing hardware.”

What is the goal in practice?

What is the output of the system?

Thank you!

lampola@student.tut.fi

<https://github.com/SakariLampola/Thesis>