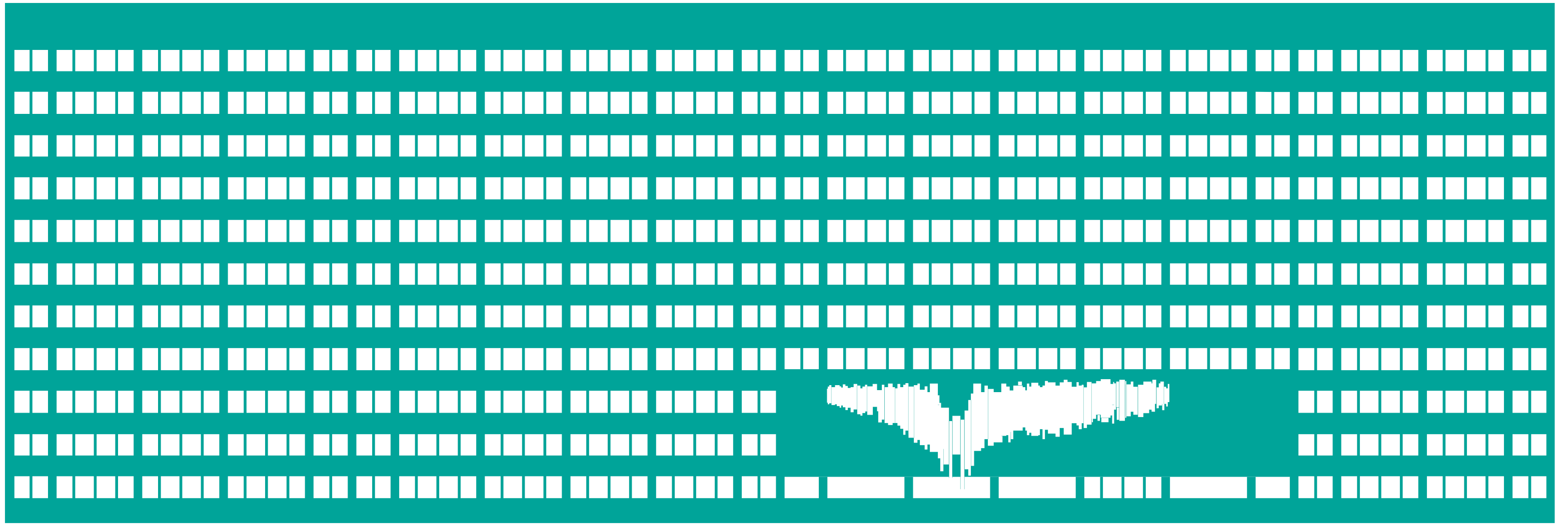
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MŠMT
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE



www.vsb.cz

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Optické systémy pro autonomní jízdu

Optical Systems for Self Driving Cars

Radovan Fusek

What Is AV (Autonomous Vehicle)?

What Is AV (Autonomous Vehicle)?

A **self-driving car**, **also known** as an **autonomous vehicle** (AV), connected and autonomous vehicle (CAV), driverless car, robo-car, or robotic car, is a vehicle that is capable of sensing its environment and moving safely with little or no human input. (Wikipedia)

A Waymo self-driving car. *Wikipedia* [online]. [cit. 2020-01-25]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/c/cf/Waymo_self-driving_car_front_view.gk.jpg

What Is AV (Autonomous Vehicle)?

- Ground vehicles
- Autonomous aerial vehicles (drone)
- Autonomous surface vehicles

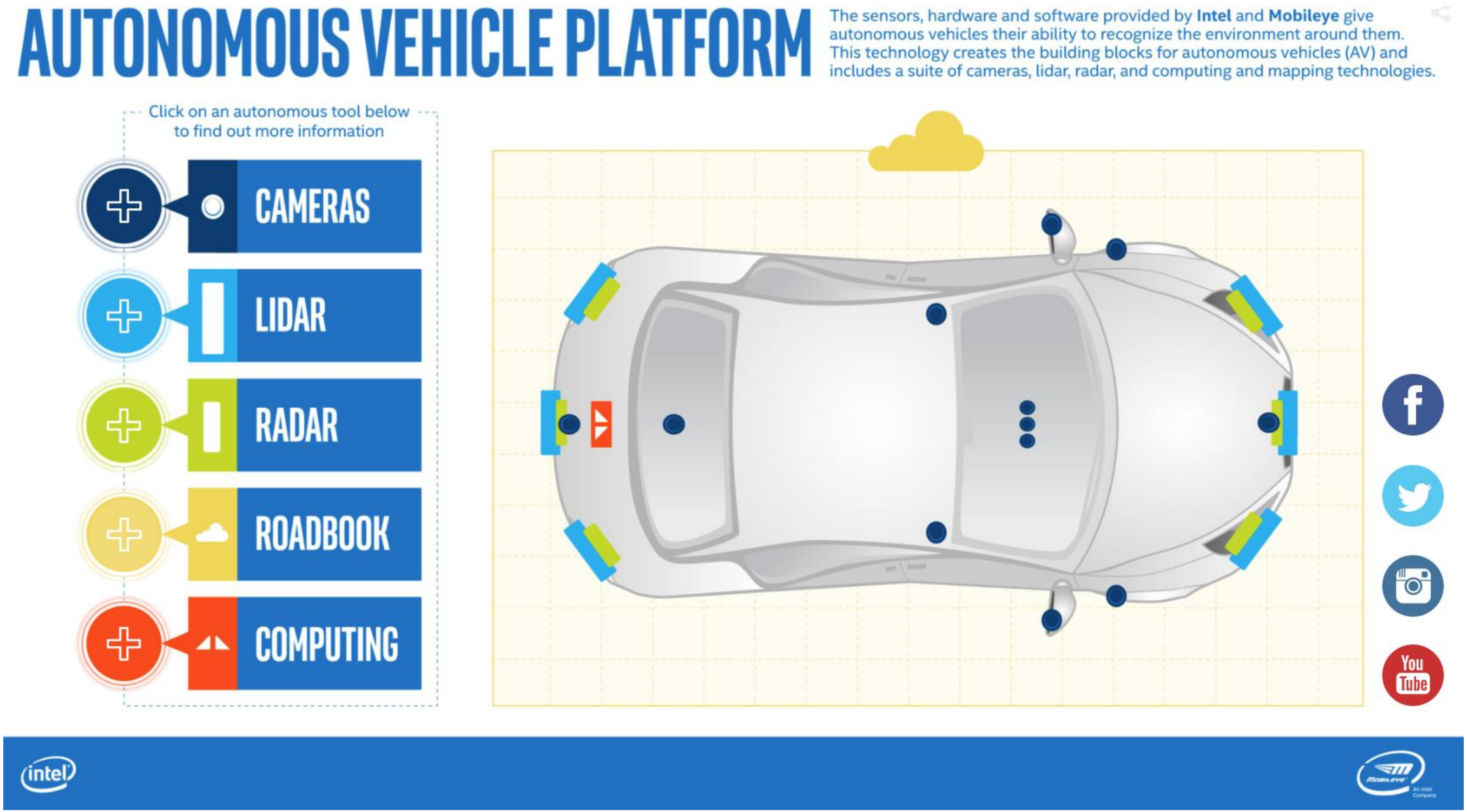
An MQ-9 Reaper unmanned aerial vehicle

An MQ-9 Reaper. *Wikipedia* [online]. [cit. 2020-01-25]. Dostupné z: https://en.wikipedia.org/wiki/File:MQ-9_Reaper_UAV_(cropped).jpg

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE



Intel Explainer: Sensors. *Businesswire* [online]. [cit. 2020-01-25]. Dostupné z: https://www.businesswire.com/news/home/20180816005108/en/Intel-Explainer-Sensors-%E2%80%93-Eyes-Ears-Autonomous https://www.youtube.com/watch?v=x7_GRigShUM

- Cameras
- Lidars
- Radars
- Maps

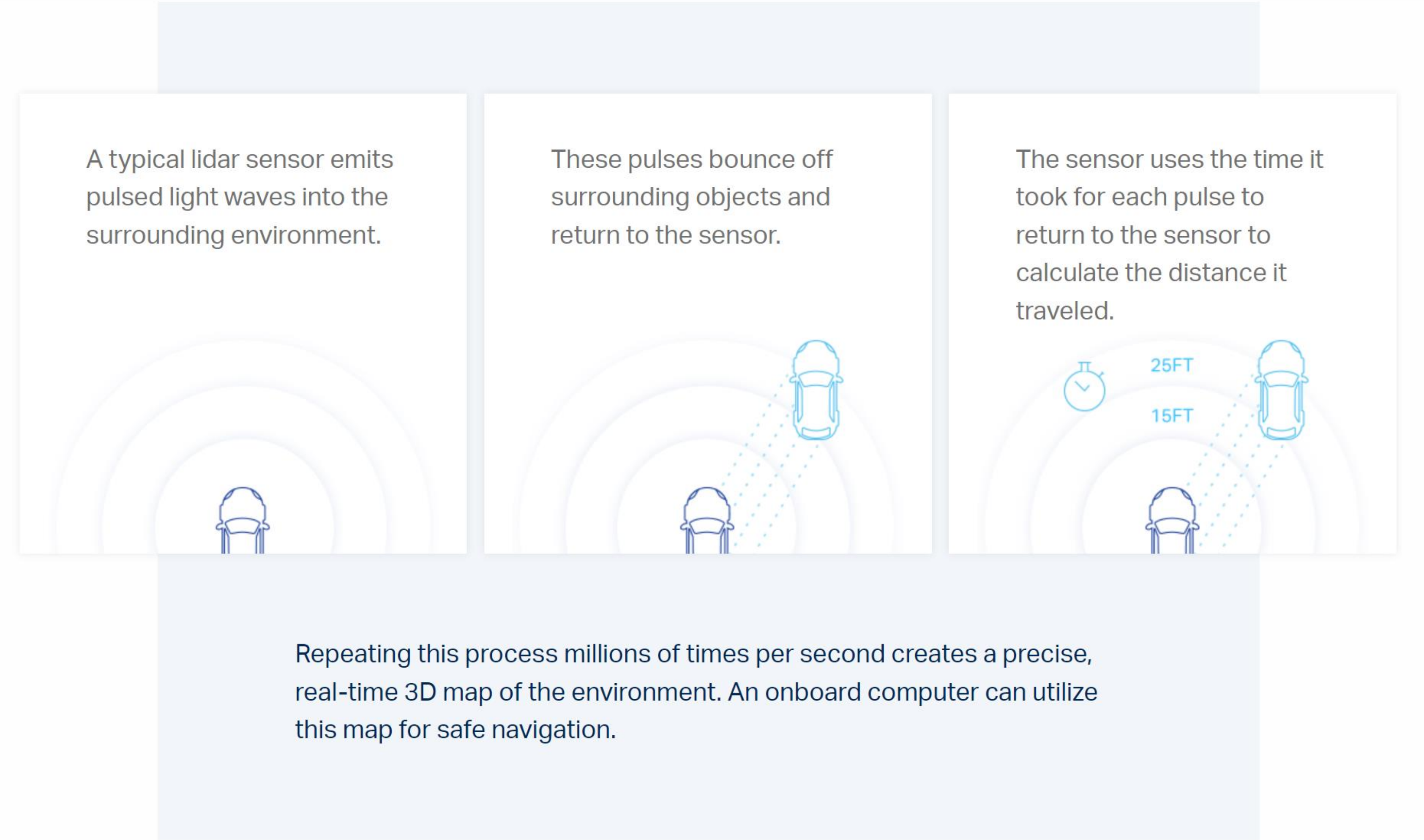
Puck Lidar Sensor. *Velodynelidar* [online]. [cit. 2020-01-25]. Dostupné z: https://eak2mvmpt4a.exactdn.com/wp-content/uploads/2019/08/Velodyne_Puck600.png?strip=all&lossy=1&ssl=1

Intel® RealSense™ Technology. *Intel* [online]. [cit. 2021-01-25]. Dostupné z: https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

https://velodynelidar.com/products/puck/

What is Lidar?

Lidar (“light detection and ranging”) uses eye-safe laser beams to “see” the world in 3D, providing machines and computers an accurate representation of the surveyed environment.

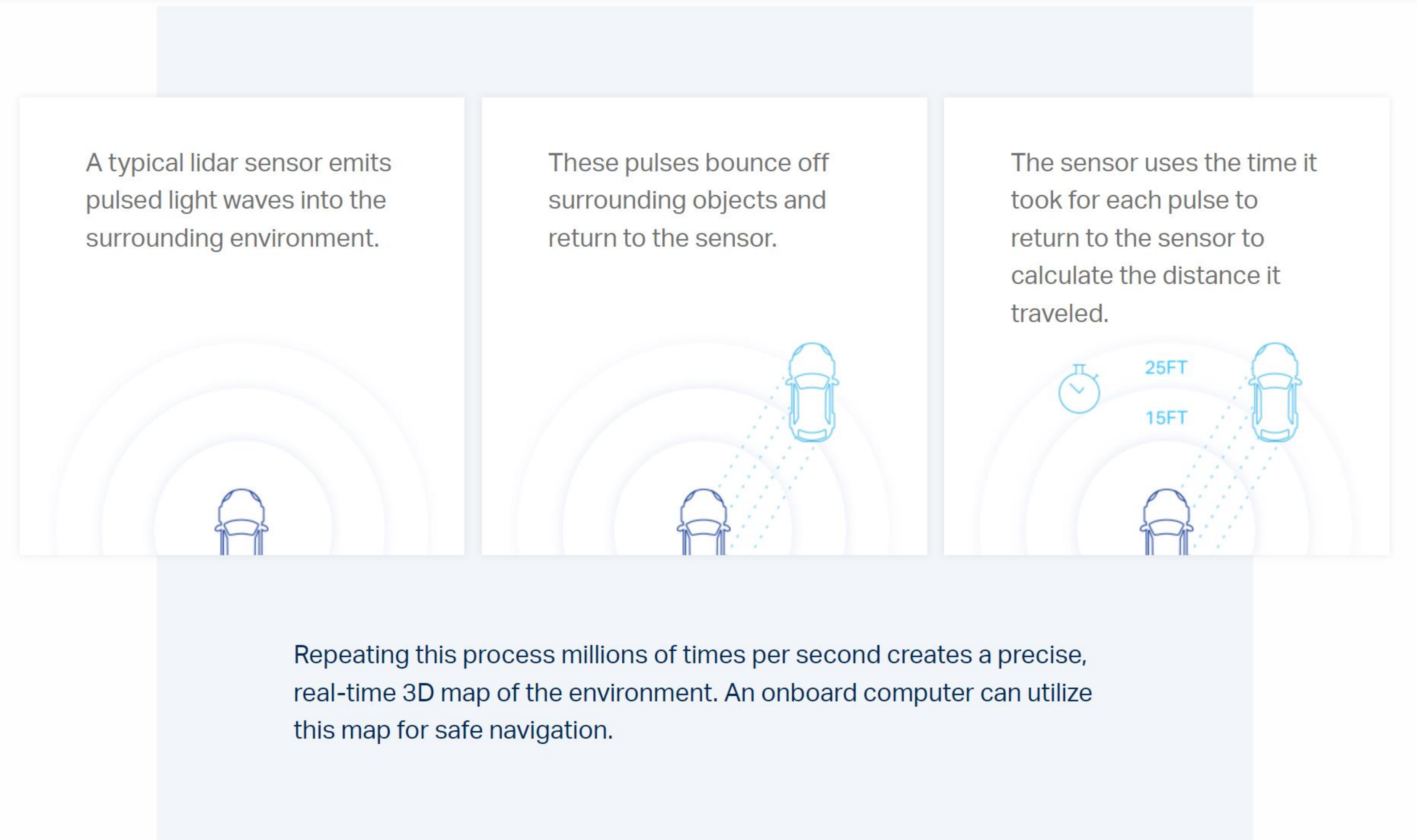


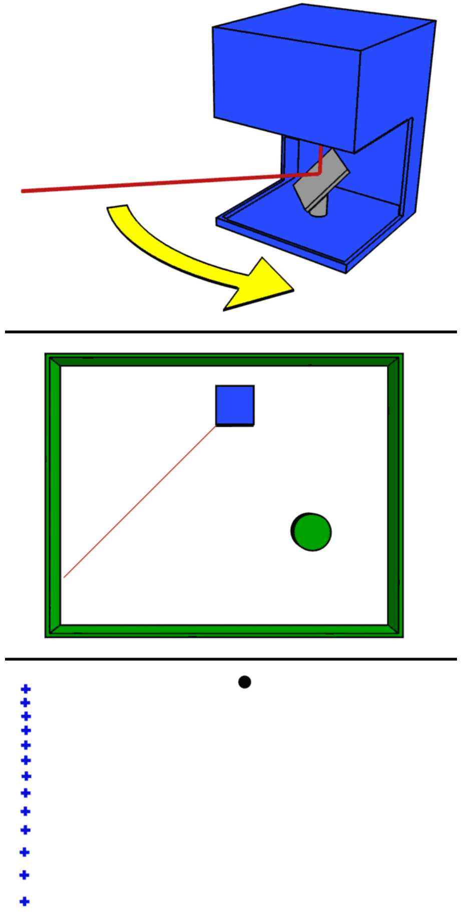
https://velodynelidar.com/what-is-lidar/

https://velodynelidar.com/products/puck/

What is Lidar?

Lidar (“light detection and ranging”) uses eye-safe laser beams to “see” the world in 3D, providing machines and computers an accurate representation of the surveyed environment.





https://velodynelidar.com/what-is-lidar/ https://upload.wikimedia.org/wikipedia/commons/c/c0/LIDAR-scanned-SICK-LMS-animation.gif https://velodynelidar.com/products/puck/

Lidars

https://velodynelidar.com/what-is-lidar/

https://velodynelidar.com/products/puck/

Lidars

https://www.youtube.com/channel/UCKyHFNyJ3QV-1rjogCYHkRA

https://velodynelidar.com/products/puck/

Lidars

https://www.youtube.com/c/ANYbotics/videos

https://velodynelidar.com/products/puck/

Lidars vs. Cameras

Tesla CEO Elon Musk: “Anyone relying on LiDAR is doomed”

https://www.youtube.com/watch?v=HM23sjhtk4Q

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Lidars vs. Cameras

https://youtu.be/HLczErXDujI

What Tesla See

https://www.youtube.com/watch?v=zRnSmw1i_DQ

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

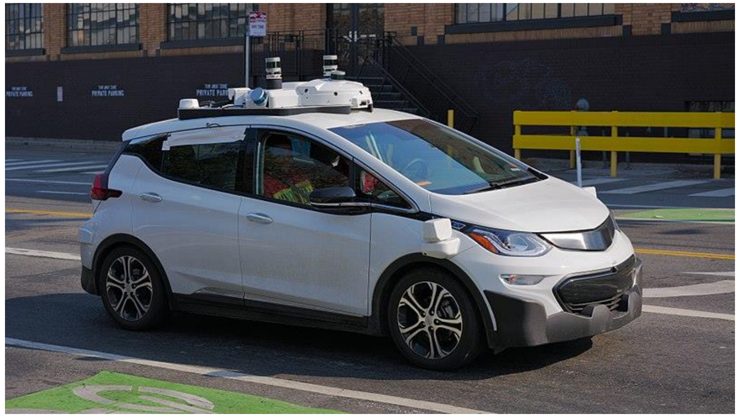
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Lidar vs Camera

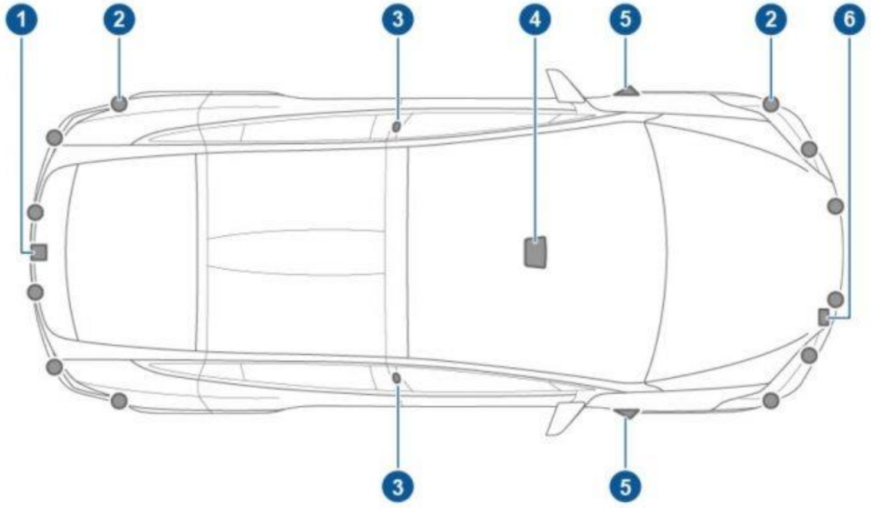
LiDAR

- cannot detect colors
- cannot interpret the text
- Impossible to identify traffic lights or road signs
- can achieve good results day and night
- high level of accuracy
- is more expensive
- requires more space
- gives self-driving cars a three-dimensional image



Camera

- can recognize colors and read road signs
- many modern AI methods to identify objects or distances
- require significantly more computing power
- camera systems are almost invisible
- challenging low-light conditions

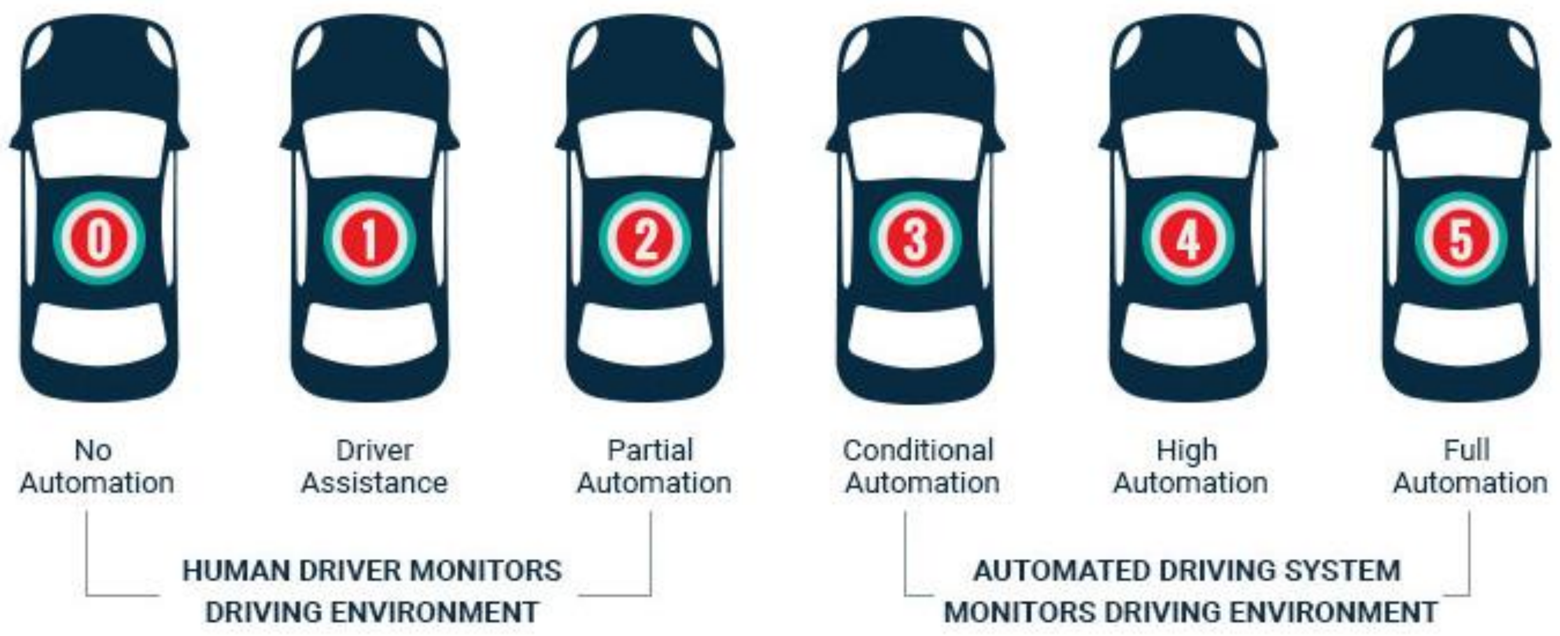


1. A camera is mounted above the rear license plate.
2. Ultrasonic sensors are located in the front and rear bumpers.
3. A camera is mounted in each door pillar.
4. Three cameras are mounted to the windshield above the rear view mirror.
5. A camera is mounted to each front fender.
6. Radar is mounted behind the front bumper on the right side of the vehicle.

Model X is also equipped with high precision electrically-assisted braking and steering systems.

https://www.autopilotreview.com/lidar-vs-cameras-self-driving-cars/
https://medium.com/0xmachina/lidar-vs-camera-which-is-the-best-for-self-driving-cars-9335b684f8d
https://leddartech.com/lidar-radar-camera-demystifying-adas-ad-technology-mix/

Levels of Autonomous Cars



Levels of autonomous cars [online]. [cit. 2020-01-25]. Dostupné z: https://geospatialmedia.s3.amazonaws.com/wp-content/uploads/2018/02/bi-graphics_autonomous-cars.png

Levels of Autonomous Cars

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

Zero autonomy; the driver performs all the driving, but the vehicle can aid with blind spot detection, forward collision warnings and lane departure warnings.

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

Levels of Autonomous Cars

The vehicle may have some active driving assist features, but the driver is still in charge. Such assist features available in today’s vehicles include adaptive cruise control, automatic emergency braking and lane keeping.

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

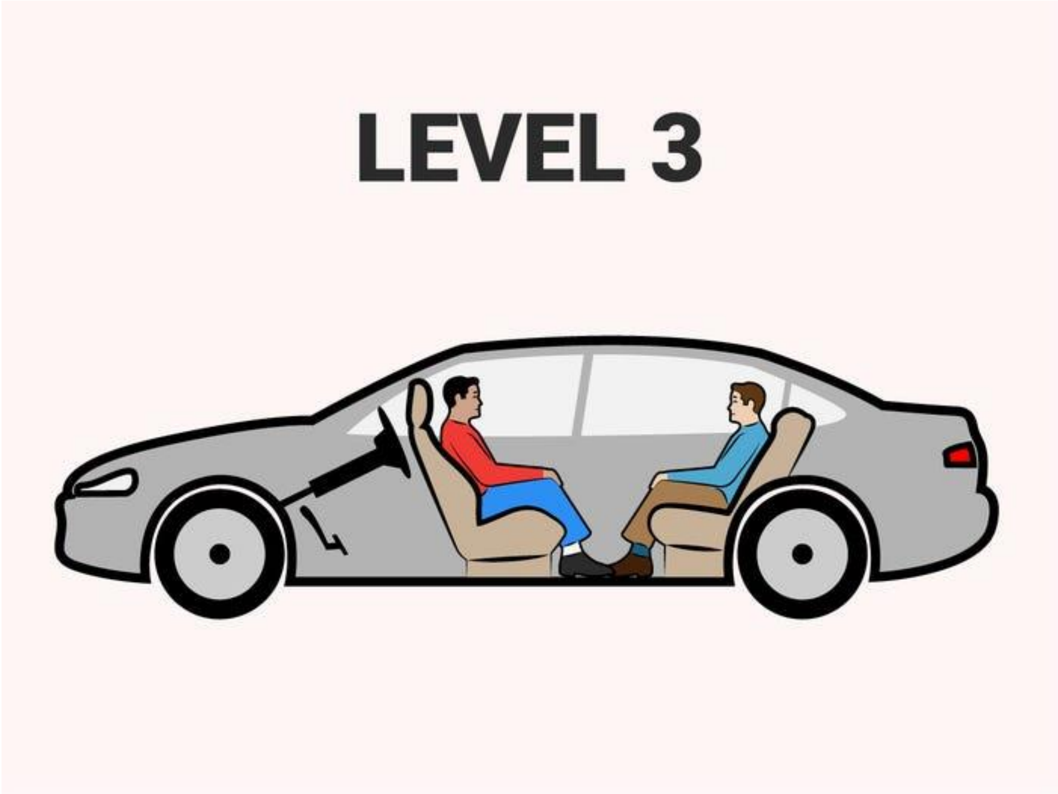
Levels of Autonomous Cars

The driver still must be alert and monitor the environment at all times, but driving assist features that control acceleration, braking and steering may work together in unison so the driver does not need to provide any input in certain situations. Such automated functions available today include self-parking and traffic jam assist (stop-and-go traffic driving).

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

Levels of Autonomous Cars

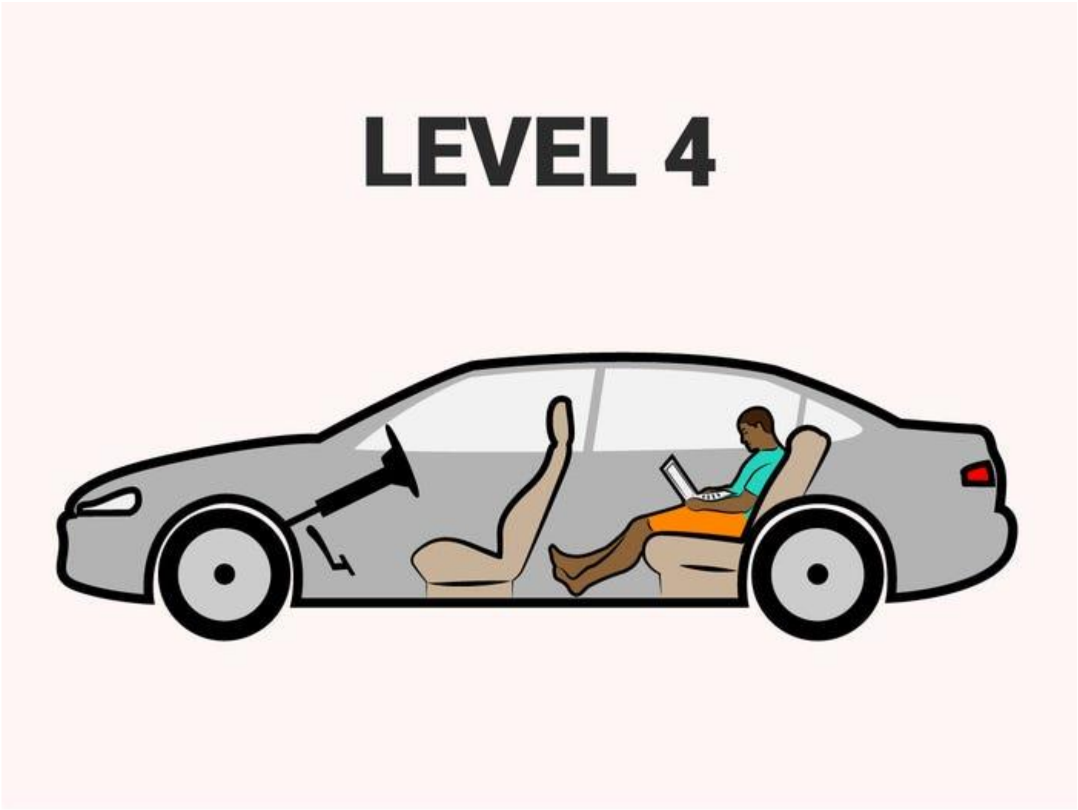


The vehicle can itself perform all aspects of the driving task under some circumstances, but the human driver must always be ready to take control at all times within a specified notice period. In all other circumstances, the human performs the driving.

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

Levels of Autonomous Cars



This is a self-driving vehicle. But it still has a driver’s seat and all the regular controls. Though the vehicle can drive and “see” all on its own, circumstances such as geographic area, road conditions or local laws might require the person in the driver’s seat to take over.

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

Levels of Autonomous Cars

The vehicle is capable of performing all driving functions under all environmental conditions and can operate without humans inside. The human occupants are passengers and need never be involved in driving. A steering wheel is optional in this vehicle.

https://newsroom.intel.com/news/autonomous-driving-hands-wheel-no-wheel-all

https://www.businessinsider.com/what-are-the-different-levels-of-driverless-cars-2016-10

Levels of Autonomous Cars in 2022 ???

“Mercedes-Benz is expected to launch the first mass-production **Level 3** car in 2022 using its Drive Pilot technology."

“BMW is widely expected to roll out **Level 3** technology in the new 7 Series”

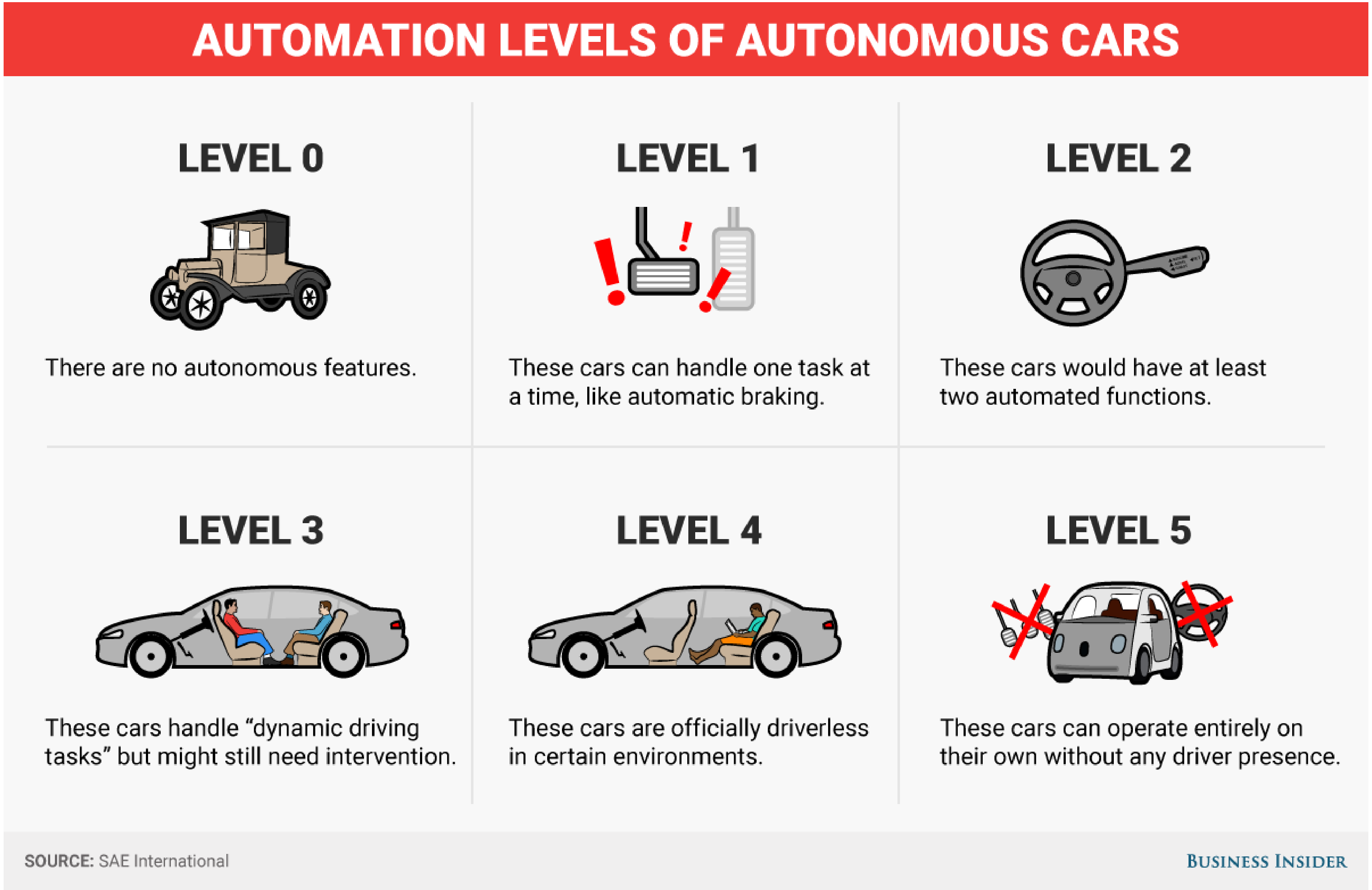
“Alphabet's Waymo recently unveiled a **Level 4** self-driving taxi service in Arizona, where they had been testing driverless cars—without a safety driver in the seat—for more than a year and over 10 million miles..”

“Tesla is likely to achieve **Level 4** autonomy in 2022, says Elon Musk, when certain milestones in the development of full self-driving (FSD) are achieved. The data show that Tesla's system performs better than a human driver for preventing accidents.”

https://www.cnet.com/roadshow/news/the-most-important-self-driving-cars-of-2022/

https://www.tesmanian.com/blogs/tesmanian-blog/tesla-likely-to-achieve-level-4-autonomy-in-2022-says-elon-musk

https://www.synopsys.com/automotive/autonomous-driving-levels.html



Levels of autonomous cars [online]. [cit. 2020-01-25]. Dostupné z: https://geospatialmedia.s3.amazonaws.com/wp-content/uploads/2018/02/bi-graphics_autonomous-cars.png

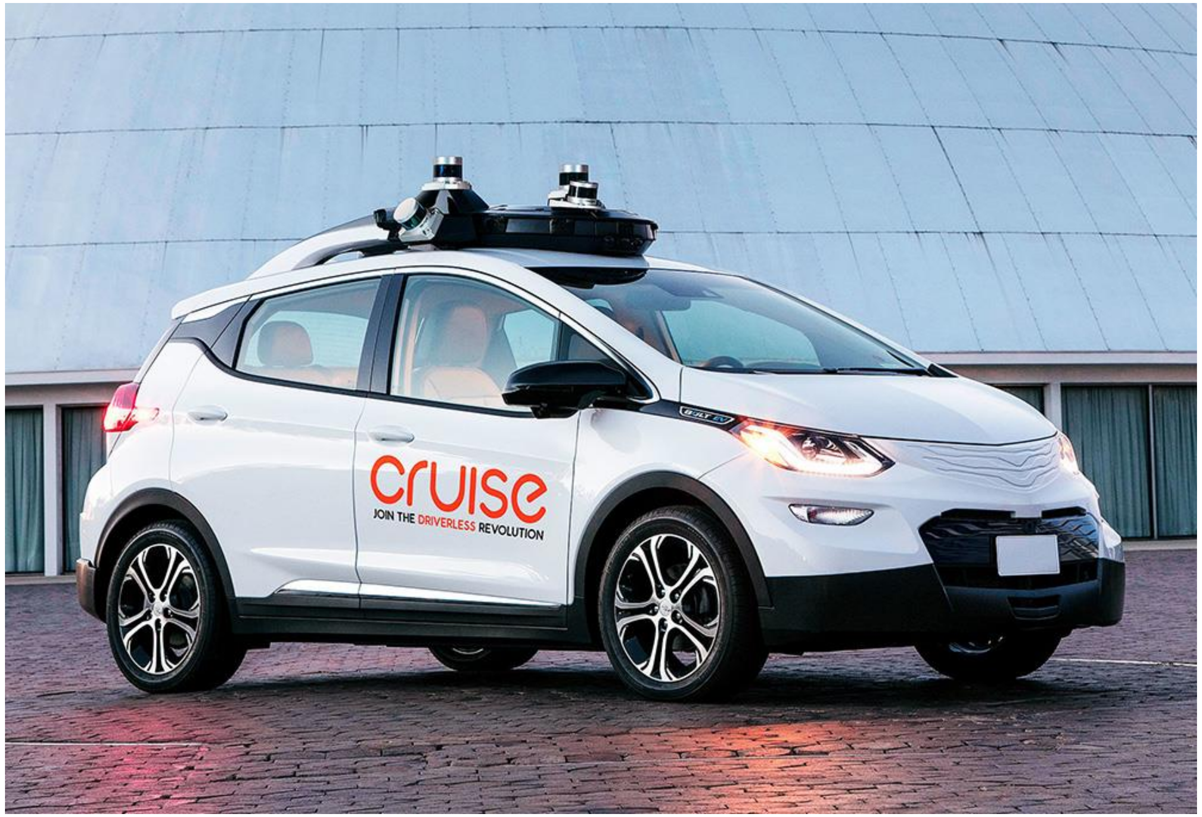
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Company Scores (2019)

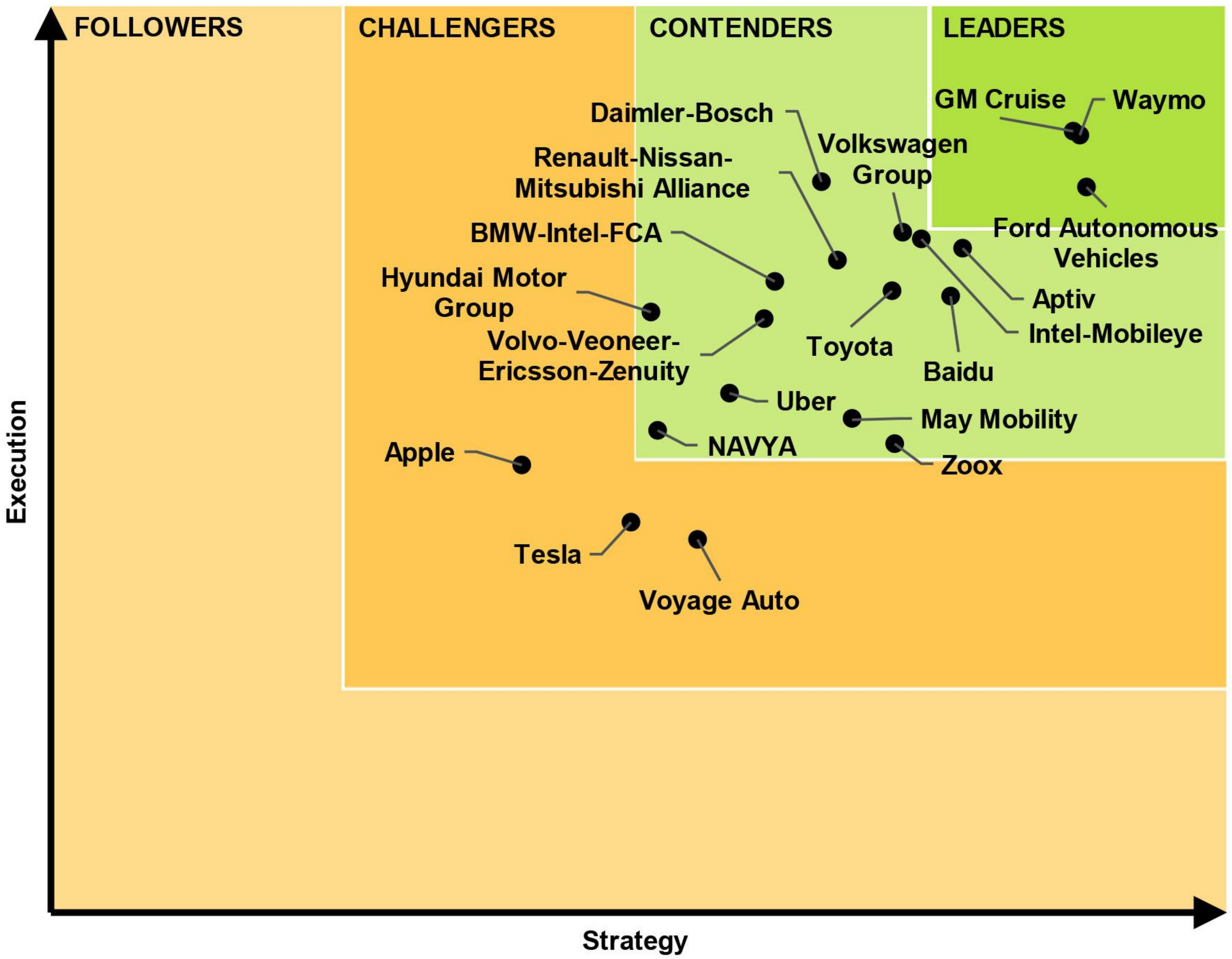
- Waymo (Google)
- GM
- Ford



Automated Driving Leaderboard. *Sae* [online]. [cit. 2020-01-25]. Dostupné z: https://www.sae.org/news/2019/03/2019-navigant-autonomous-leaderboard

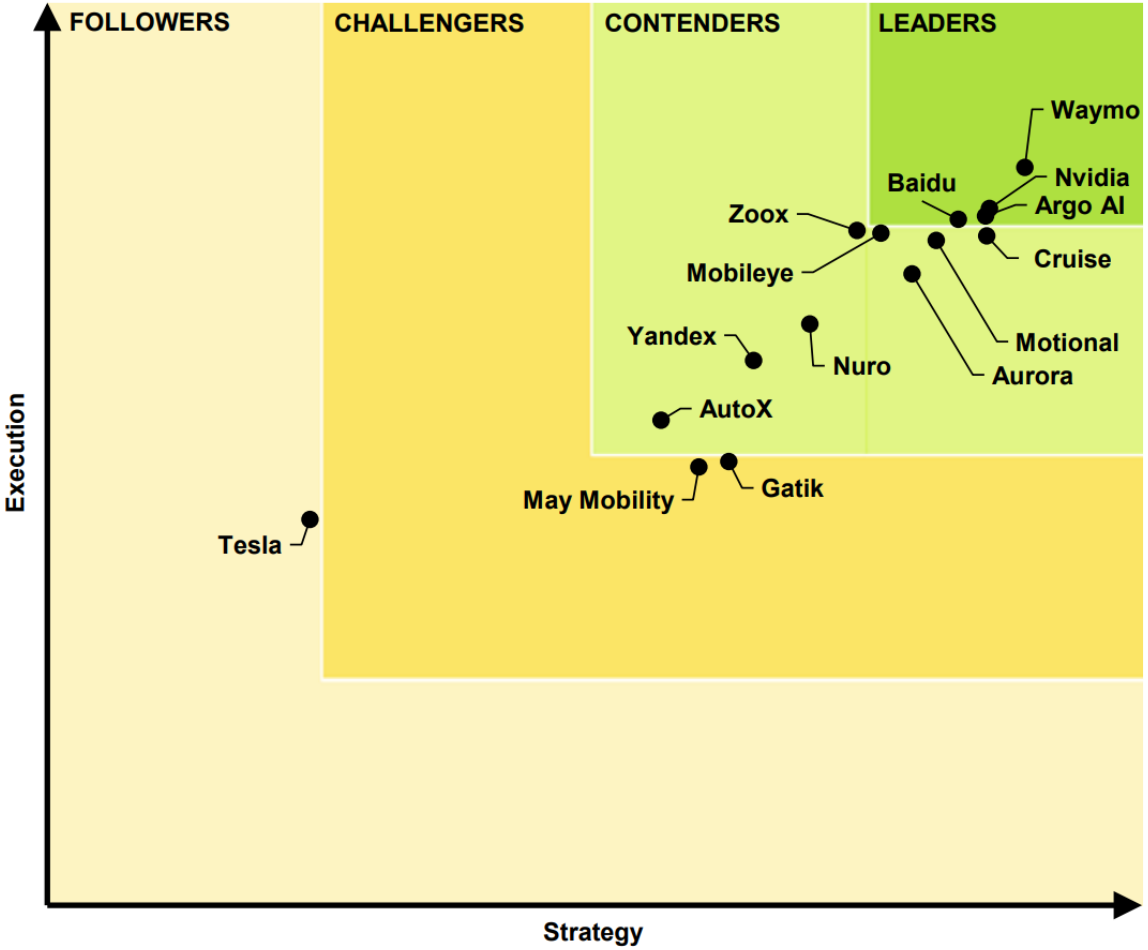
Company Scores (2019)

The Navigant Research Leaderboard Grid



Automated Driving Leaderboard. *Sae* [online]. [cit. 2020-01-25]. Dostupné z: https://www.sae.org/news/2019/03/2019-navigant-autonomous-leaderboard

Company Scores (2020)



(Source: Guidehouse Insights)

Chart 1-1 shows the ranking of each company. This year, four companies had scores that earned a place in the Leaders group: Waymo, Nvidia, Argo AI, and Baidu. Several others, including Cruise, Motional, Mobileye, Zoox, and Aurora, fell just outside of this group among the eight companies in the Contenders group. Notably, Tesla continues to rank at the bottom of this list despite getting significant press attention for its full self-driving (FSD) beta software release. Although several of the companies ranked this year have close affiliations with automakers, Tesla is the only automaker on the list and has made marketing FSD a key feature in selling vehicles. Tesla has made significant progress in strengthening several areas including staying power thanks to the runup in its stock price in the second half of 2020, but its technology is still lacking.

Guidehouse Insights Leaderboard report. *Guidehouseinsights* [online]. [cit. 2022-02-25]. Dostupné z: https://guidehouseinsights.com/reports/guidehouse-insights-leaderboard-automated-driving-systems

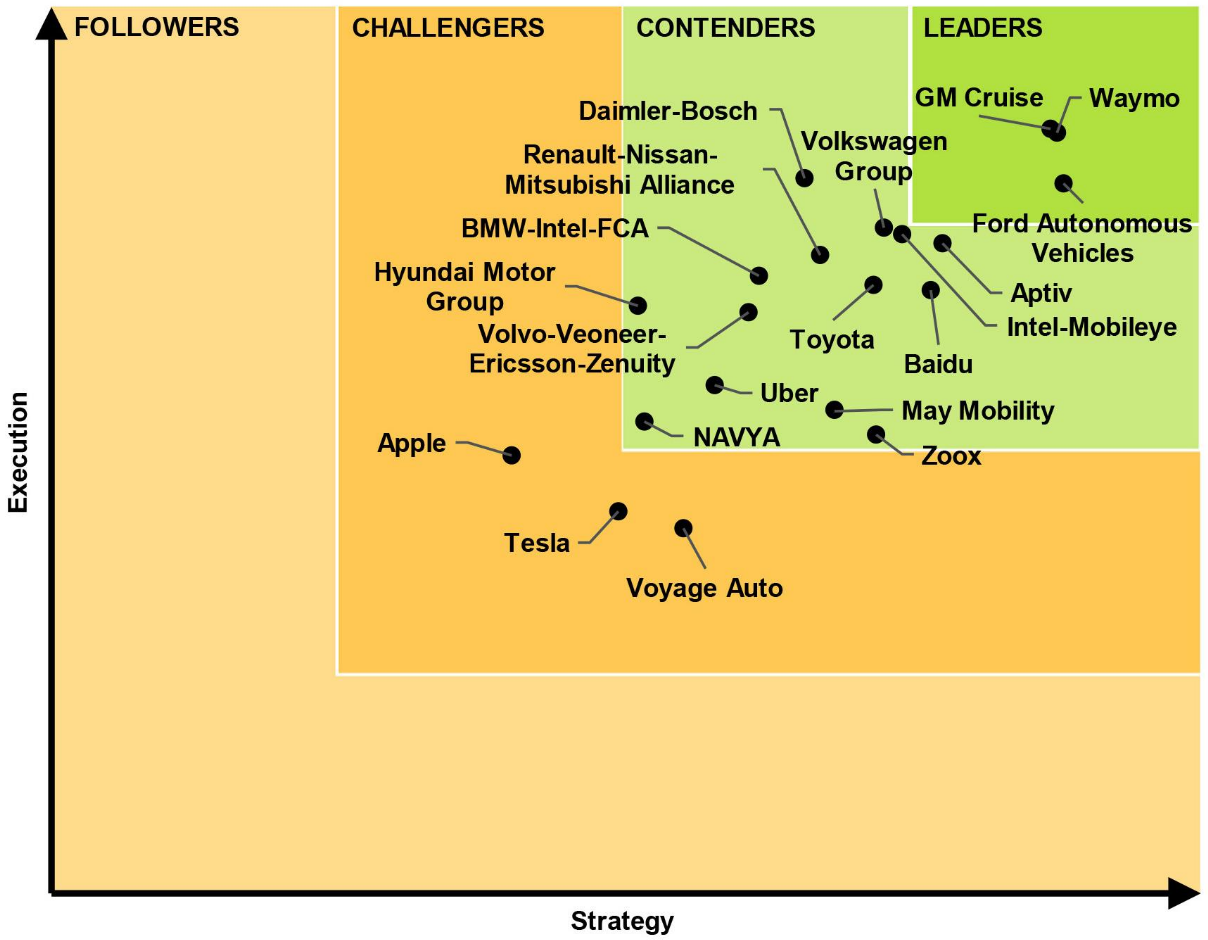
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

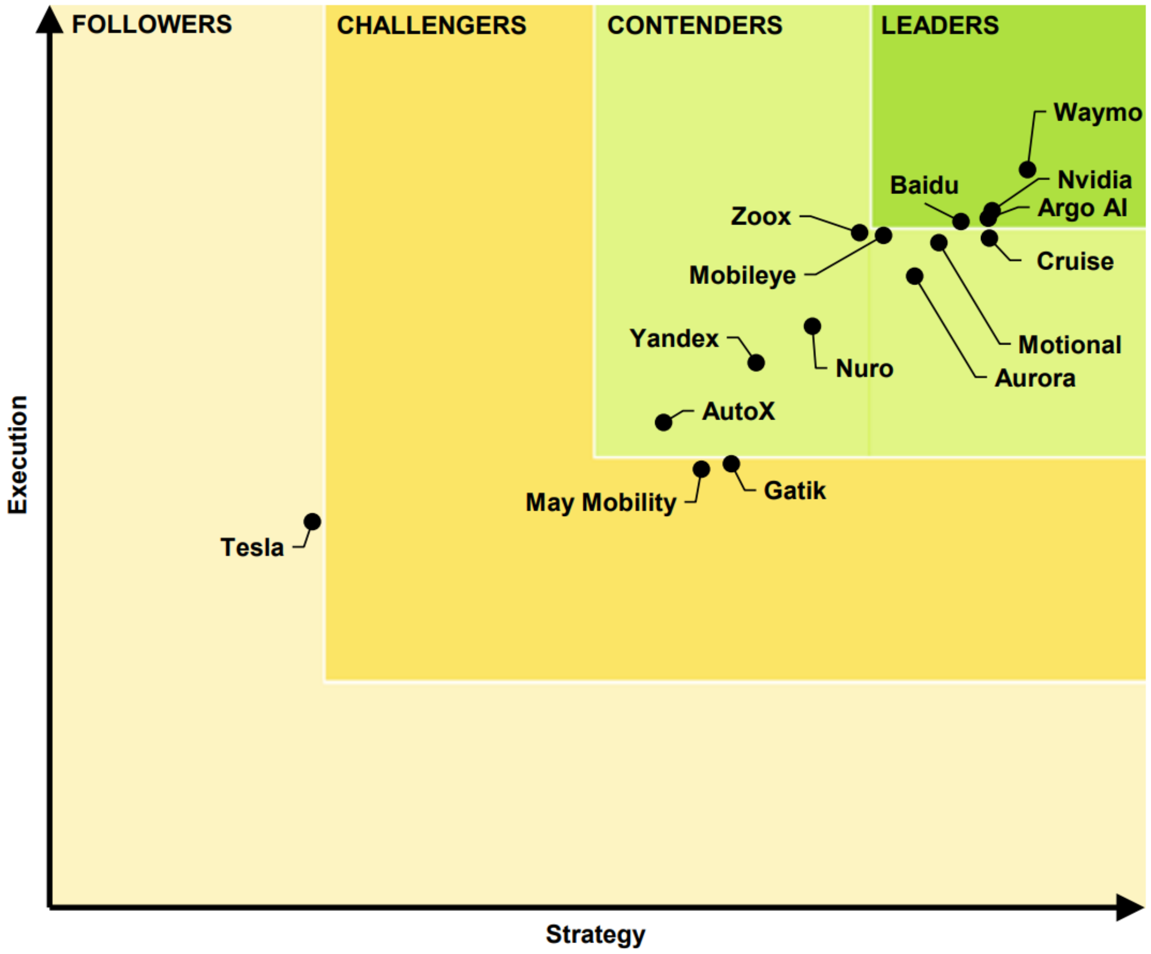
MINISTRY OF EDUCATION, YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Company Scores (2019 vs 2020)

The Navigant Research Leaderboard Grid





(Source: Guidehouse Insights)

Automated Driving Leaderboard. *Sae* [online]. [cit. 2020-01-25]. Dostupné z: https://www.sae.org/news/2019/03/2019-navigant-autonomous-leaderboard
Guidehouse Insights Leaderboard report. *Guidehouseinsights* [online]. [cit. 2022-02-25]. Dostupné z: https://guidehouseinsights.com/reports/guidehouse-insights-leaderboard-automated-driving-systems

Identifying the Waymo Fully Self-Driving Vehicle

The Waymo fully self-driving Chrysler Pacifica Hybrid minivans can be easily identified by the white color with Waymo logos, roof assembly, front fender additions, or rear roof additions below.

During driverless testing and operation, Waymo's vehicles are fully self-driving at all times, and will not have any person in the driver's seat either steering or otherwise controlling the vehicle.

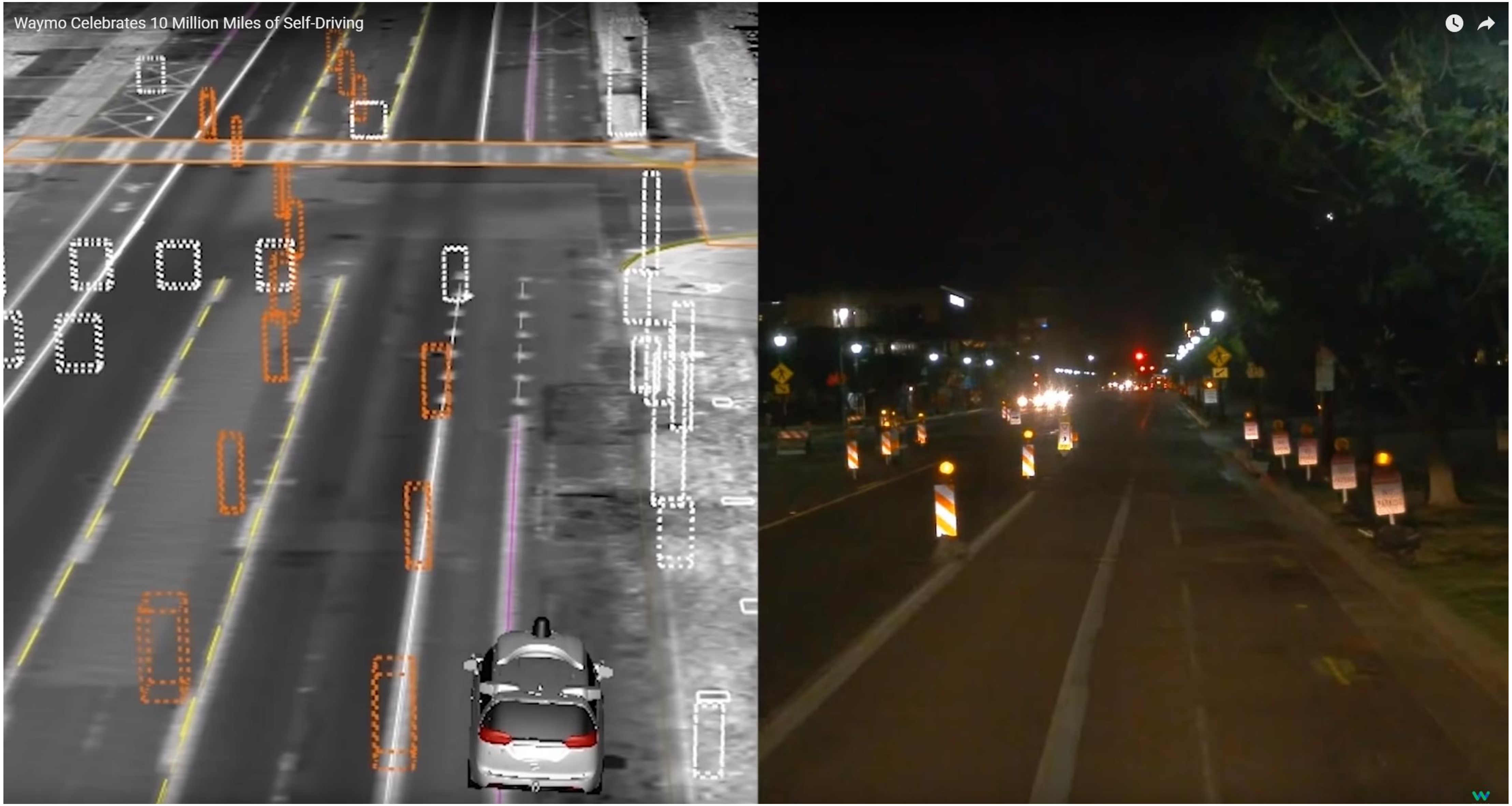
Waymo Fully Self-Driving. *Thedrive* [online]. [cit. 2022-02-25]. Dostupné z: https://s3.amazonaws.com/the-drive-staging/message-editor%2F1540065806515-waymo1.jpg

Waymo Fully Self-Driving. *Thedrive* [online]. [cit. 2020-01-25]. Dostupné z: https://guidehouseinsights.com/reports/guidehouse-insights-leaderboard-automated-driving-systems

https://waymo.com/

https://www.youtube.com/watch?v=DJlcdH6iuAk

https://www.youtube.com/watch?v=O8TSA-X9UIU



Waymo Celebrates 10 Million Miles of Autonomous Driving [online]. [cit. 2020-01-25]. Dostupné z: https://www.youtube.com/watch?v=ROAwXEqDk7k&t=32s

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

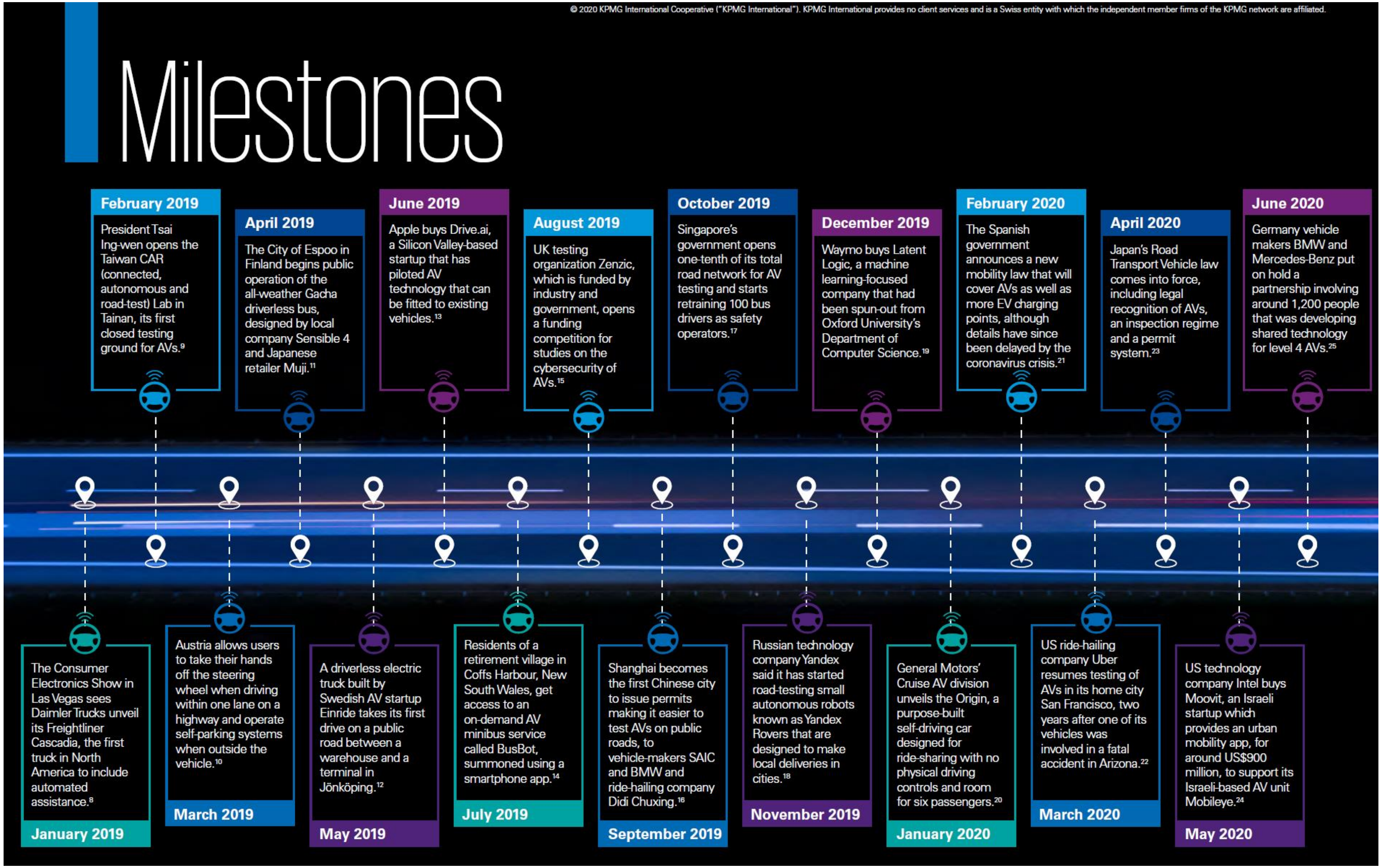
VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Full Self-Driving: Tesla [online]. [cit. 2020-01-25]. Dostupné z: https://www.youtube.com/watch?time_continue=1&v=tlThdr3O5Qo

The Autonomous Vehicles Readiness Index

Country or jurisdiction	Rank 2020	Rank 2019	2020 score
Singapore	1	2	25.45
The Netherlands	2	1	25.22
Norway	3	3	24.25
United States	4	4	23.99
Finland	5	6	23.58
Sweden	6	5	23.17
South Korea	7	13	22.71
United Arab Emirates	8	9	22.23
United Kingdom	9	7	21.36
Denmark	10	n/a	21.21
Japan	11	10	20.88
Canada	12	12	20.68
Taiwan	13	n/a	19.97
Germany	14	8	19.88
Australia	15	15	19.70
Israel	16	14	19.40
New Zealand	17	11	19.19
Austria	18	16	19.16
France	19	17	18.59
China	20	20	16.42
Belgium	21	n/a	16.23
Spain	22	18	16.15
Czech Republic	23	19	13.99
Italy	24	n/a	12.70
Hungary	25	21	11.66
Russia	26	22	11.45
Chile	27	n/a	11.28
Mexico	28	23	7.42
India	29	24	6.95
Brazil	30	25	5.49

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html



Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

https://www.daimlertruck.com/innovation/safe-automated/autonomous-driving-daimler-trucks.html

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

https://en.wikipedia.org/wiki/Einride

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

https://www.youtube.com/c/Einride

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

https://www.youtube.com/watch?v=Vf44Pw3BqUI

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Executive summary

Methodology

The 2020 edition of the AVRI assesses 30 countries and jurisdictions. This includes the addition of five new countries and jurisdictions to the roster from 2019, and can explain some of the downward movement of some countries as a result. The AVRI uses 28 different measures, organized into four pillars: policy and legislation, technology and innovation, infrastructure and consumer acceptance. Four of the variables are scored for this index by KPMG International and ESI ThoughtLab and 24 draw on existing research by KPMG International and other organizations. Full details are in the Appendix.

P. 12 Singapore

1st

— For the first time Singapore leads the AVRI, overtaking the Netherlands for the top-ranked position and leading on both the consumer acceptance and policy and legislation pillars.

— The city-state has expanded AV testing to cover all public roads in western Singapore and aims to serve three areas with driverless buses from 2022.

— The number of charging points will increase from 1,600 to 28,000 by 2030 with incentives for buying EVs, although the government is also phasing in a usage tax to compensate for loss of fuel excise duties. Given they will be mostly electric, such moves are vital in enabling AV implementation.

P. 13 The Netherlands

2nd

— The Netherlands retains top ranking on the infrastructure pillar, leading on EV charging stations per capita and second only to Singapore on road quality.

— An extensive series of pilots means that 81 percent of people live near AV testing sites. However, tests on truck platooning in July 2019 found challenges in keeping vehicles connected at all times.

— 2019 saw the Netherlands extending its use of smart road furniture, including traffic lights that send their statuses wirelessly to AVs in 60 new areas of the country.

P. 14 Norway

3rd

— Norway extended its use of AVs in 2019, with several bus routes in Oslo now driverless, and the speed limit for driverless vehicles on roads increasing from 16kph to 20kph.

— A majority of passenger vehicles bought in Norway in 2019 were battery or plug-in hybrids, as a result of high taxes on internal combustion vehicles and fuels and subsidies for EVs.

— The country is testing AVs in extreme weather, with pilots of driverless trucks, cars and buses on the snow-bound Svalbard islands in the Arctic Circle.

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

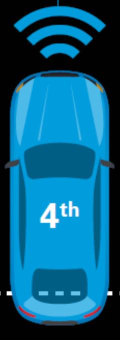
Also noted

- South Korea climbs six places to 7th in this edition of the AVRI, the biggest rise of any country. The government published a national strategy for AVs in October 2019, with the goal of reducing road deaths by three-quarters.
- The UK leads on a new AVRI measure of cybersecurity, with AV testing body Zenzic funding seven projects in this field.
- Israel retains its leadership of the technology pillar, leading on both AV-related companies and investments scaled by population.

New to AVRI

- Denmark is the highest-rated of the five countries and jurisdictions joining this edition of the AVRI, occupying the 10th spot. It allows AV tests on any public road and its first driverless bus service started running in March 2020 in Aalborg.
- Taiwan, the second highest at 13th, has a focus on testing AVs on its challenging mixed-use roads. Taipei is planning to start a night-time trial of driverless buses partly to tackle a shortage of drivers.
- Belgium, entering at 21st, ran its first demonstration of an AV bus at Brussels airport in May 2019, operated by Flemish regional transport authority De Lijn.
- Italy, placed 24th, introduced rules and an observatory for AV testing in 2018, with tests beginning in Parma and Turin in 2019.
- Chile, at 27th, has made use of AVs in mining for several years and in January 2020 started Latin America's first public pilot in a park in central Santiago.

P. 15 United States

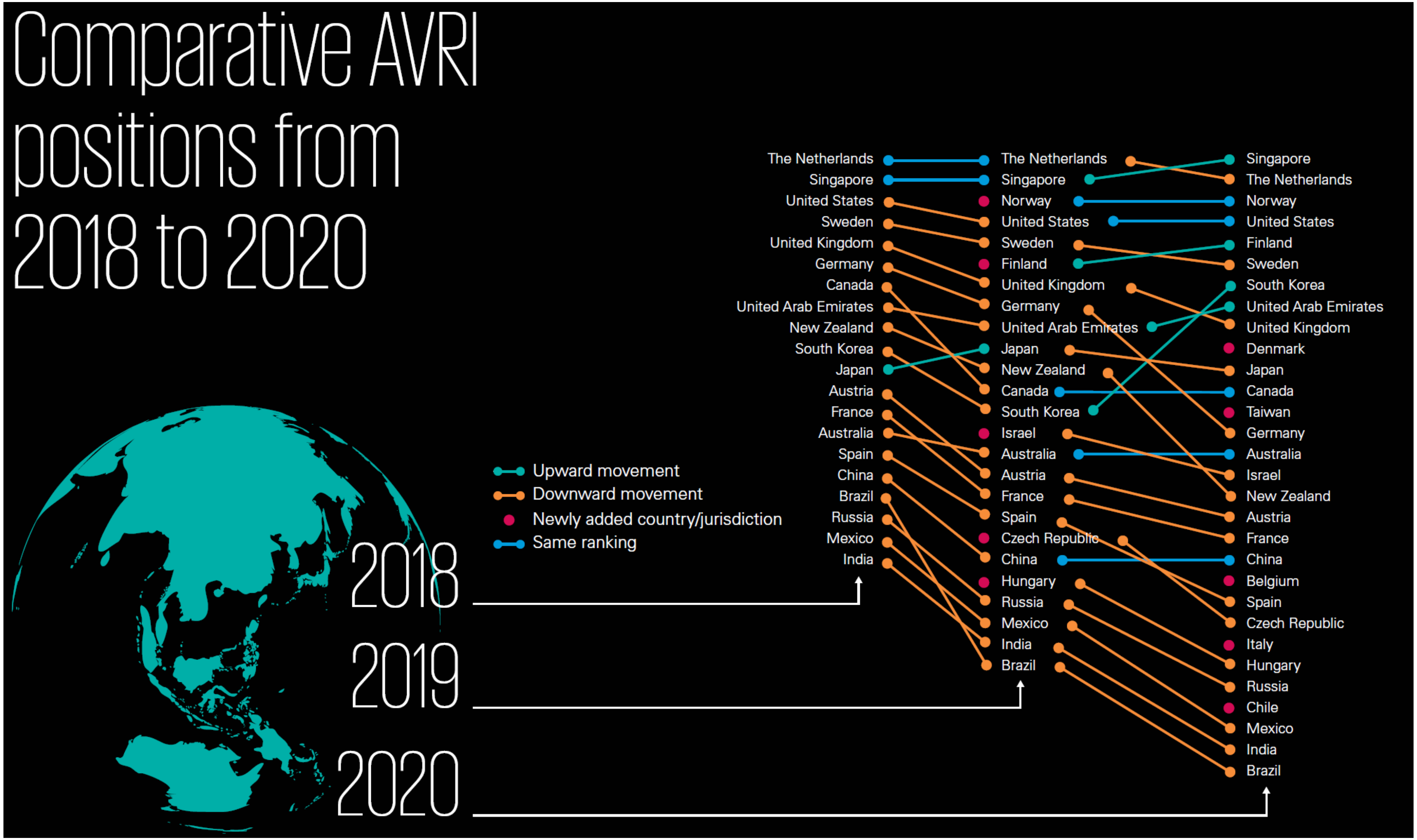


- The US is second only to Israel on technology and innovation, with 420 AV company headquarters, 44 percent of all of those tracked in this research.
- American technology companies, including Apple and Google's Waymo unit, and vehicle makers such as General Motors and Ford, continue to dominate AV development. GM's Cruise division unveiled the Origin, a purpose-built self-driving car designed for ride-sharing.
- Cities including Detroit and Pittsburgh are undertaking innovative work to introduce and promote AVs (both are profiled in the Cities to watch section).

P. 16 Finland

- Finland has the highest ratings for AV-specific regulations and for the efficiency of its legal system in challenging regulations, and its entire road network is open for AV trials.
- Helsinki (profiled in Cities to watch) and its neighbor Espoo both run public AV bus services, with the latter using an all-weather vehicle designed by local company Sensible 4.
- Finland also leads on measures of digital skills, benefiting from a breadth of talented engineers, many of whom have notable experience having been part of Nokia's legacy. It also makes the greatest use of ride-hailing services.

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html



Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

23 | Czech Republic

Policy and legislation	Technology and innovation	Infrastructure	Consumer acceptance
22	25	24	22

The Czech Republic is one of the five countries receiving the top rating for government-funded AV pilots, and testing is the country's main area of strength. 2020 should see construction start on German vehicle maker BMW's EUR300 million (US$340 million) AV test site at Sokolov, around 300km (190 miles) from the company's main development site in Munich. BMW plans to open the site, which will have around 100km of road allowing tests of city, highway and rural roads, in the second half of 2022. It will create around 700 jobs and has established a cooperation agreement with the University of West Bohemia.[103]

The country has several other test facilities under development. Czech investment group Accolade is planning to build on a site near Stříbro, which is similarly near the German border, to be used by companies developing AV technologies. It plans to open in 2022 at a cost of EUR180 million (US$200 million), which will also offer a range of road environments including European cities that do not use right-angled grids of roads.[104] Czech-based vehicle maker Skoda, part of Germany's Volkswagen, is working on a site while German safety company TÜV and French vehicle part maker Valeo Group are both looking to convert disused airfields.

"Our strength is that the automotive industry is already here," says Pavel Kliment, Partner, KPMG in the Czech Republic, with the country making vehicles for a number of companies. "That's why there is the focus on test sites." There is less research and development work, although there are good examples, such as German vehicle maker Porsche, another Volkswagen unit, and Italian parts maker Marelli have research partnerships with the Czech Technical University in Prague.[105]

Aside from testing, Kliment says that the Czech Republic lacks a legal framework for the use of AVs. The technology attracts attention when there is a significant announcement, such as when BMW detailed its test site plans in January 2020. "There are a lot of positive things happening, but it's not a strategic issue," he says. "I expect the importance will gradually grow over time, particularly when the test sites are completed."

Government-funded AV pilots

Top performing countries and jurisdictions

- Canada
- Czech Republic
- Singapore
- South Korea
- Taiwan

Source: KPMG International (2020)

> "Our strength is that the automotive industry is already here. That's why there is the focus on test sites."
>
> **Pavel Kliment**
> Partner
> KPMG in the Czech Republic

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

The Autonomous Vehicles Readiness Index

- Policy and legislation
- Technology and innovation
- Infrastructure
- Consumer acceptance

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

Policy and legislation pillar scores breakdown by variable

	Position	AV Regulations	Government-funded AV Pilots	AV-focused agency	Future orientation of government	Efficiency of legal system in challenging regulations	Government readiness for change	Data-sharing environment	Pillar 1 score (unadjusted)
1	**Singapore**	1.000	1.000	1.000	1.000	0.673	1.000	0.411	**6.084**
2	**United Kingdom**	0.929	0.857	0.857	0.534	0.668	0.780	1.000	**5.626**
3	**The Netherlands**	1.000	0.929	0.714	0.639	0.825	0.780	0.688	**5.576**
4	**Finland**	1.000	0.857	0.714	0.718	1.000	0.780	0.451	**5.521**
5	**New Zealand**	0.929	0.714	0.929	0.573	0.792	0.829	0.743	**5.509**
6	**United States**	0.857	0.929	0.714	0.763	0.792	0.634	0.771	**5.461**
7	**Germany**	0.786	0.857	0.857	0.604	0.747	0.829	0.621	**5.301**
8	**United Arab Emirates**	0.857	0.714	0.929	0.880	0.865	0.951	0.081	**5.278**
9	**Canada**	0.786	1.000	0.714	0.502	0.614	0.756	0.870	**5.242**
10	**Norway**	0.929	0.857	0.643	0.575	0.629	0.854	0.674	**5.161**
11	**Austria**	0.857	0.857	0.929	0.502	0.568	0.610	0.629	**4.952**
12	**Denmark**	0.714	0.643	0.857	0.589	0.666	0.829	0.633	**4.931**
13	**Taiwan**	0.857	1.000	0.786	0.334	0.425	0.659	0.860	**4.920**
14	**France**	0.786	0.857	0.714	0.481	0.615	0.585	0.815	**4.854**
15	**Sweden**	0.714	0.714	0.714	0.564	0.624	0.878	0.625	**4.834**
16	**South Korea**	0.857	1.000	0.857	0.488	0.346	0.463	0.766	**4.777**
17	**Australia**	1.000	0.571	0.714	0.409	0.516	0.707	0.765	**4.683**
18	**Japan**	0.571	0.857	0.571	0.505	0.642	0.659	0.691	**4.496**
19	**Israel**	0.714	0.786	0.643	0.532	0.603	0.488	0.331	**4.097**
20	**Belgium**	0.929	0.714	0.714	0.271	0.565	0.512	0.319	**4.024**
21	**China**	0.786	0.929	0.643	0.490	0.535	0.561	0.000	**3.944**
22	**Czech Republic**	0.857	1.000	0.714	0.186	0.222	0.512	0.309	**3.800**
23	**Spain**	0.857	0.571	0.714	0.163	0.322	0.317	0.668	**3.614**
24	**Chile**	0.429	0.571	0.429	0.435	0.435	0.439	0.346	**3.083**
25	**Hungary**	0.643	0.857	1.000	0.266	0.000	0.244	0.046	**3.056**
26	**Russia**	0.571	0.286	0.857	0.367	0.240	0.293	0.360	**2.973**
27	**Italy**	0.857	0.643	0.643	0.000	0.056	0.293	0.452	**2.943**
28	**India**	0.000	0.000	0.000	0.536	0.514	0.341	0.288	**1.679**
29	**Mexico**	0.143	0.143	0.143	0.168	0.194	0.098	0.670	**1.557**
30	**Brazil**	0.286	0.143	0.143	0.011	0.119	0.000	0.488	**1.190**

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

Technology and innovation pillar scores breakdown by variable

	Position	Industry partnerships	AV technology firm headquarters	AV-related patents	Industry investments in AV	Availability of the latest technologies	Innovation capability	Cybersecurity	Assessment of cloud computing, AI and IoT	Market share of electric cars	Pillar 2 score (unadjusted)
1	**Israel**	0.750	1.000	0.052	1.000	0.946	0.716	0.679	0.551	0.029	**5.722**
2	**United States**	1.000	0.122	0.298	0.370	0.931	0.939	0.989	1.000	0.033	**5.681**
3	**Japan**	0.917	0.022	1.000	0.055	0.843	0.808	0.889	0.707	0.017	**5.258**
4	**Germany**	1.000	0.078	0.849	0.124	0.751	1.000	0.822	0.574	0.052	**5.250**
5	**Norway**	0.917	0.053	0.012	0.000	0.971	0.576	0.915	0.764	1.000	**5.209**
6	**Sweden**	0.833	0.203	0.352	0.051	0.937	0.826	0.737	0.805	0.201	**4.946**
7	**South Korea**	1.000	0.026	0.856	0.023	0.633	0.826	0.874	0.551	0.043	**4.832**
8	**Finland**	0.833	0.171	0.017	0.035	1.000	0.752	0.837	0.705	0.123	**4.475**
9	**United Kingdom**	0.833	0.104	0.113	0.011	0.855	0.806	1.000	0.676	0.057	**4.456**
10	**The Netherlands**	0.667	0.066	0.032	0.103	0.907	0.763	0.900	0.701	0.265	**4.403**
11	**Singapore**	0.833	0.133	0.020	0.004	0.771	0.738	0.928	0.717	0.085	**4.230**
12	**France**	0.833	0.043	0.116	0.029	0.735	0.783	0.972	0.567	0.049	**4.127**
13	**Canada**	1.000	0.085	0.012	0.073	0.782	0.711	0.915	0.488	0.047	**4.114**
14	**Taiwan**	0.833	0.007	0.094	0.000	0.551	0.851	0.856	0.736	0.018	**3.946**
15	**Denmark**	0.667	0.015	0.011	0.000	0.740	0.761	0.829	0.800	0.074	**3.896**
16	**Austria**	0.667	0.087	0.036	0.044	0.685	0.722	0.772	0.450	0.065	**3.527**
17	**Australia**	0.500	0.034	0.045	0.007	0.576	0.609	0.911	0.545	0.020	**3.248**
18	**Belgium**	0.417	0.032	0.007	0.001	0.808	0.652	0.746	0.521	0.057	**3.242**
19	**New Zealand**	0.667	0.019	0.010	0.000	0.743	0.409	0.692	0.567	0.048	**3.155**
20	**China**	1.000	0.002	0.045	0.014	0.023	0.503	0.777	0.446	0.103	**2.913**
21	**Italy**	0.833	0.008	0.012	0.000	0.330	0.519	0.796	0.360	0.016	**2.875**
22	**United Arab Emirates**	0.833	0.008	0.000	0.005	0.787	0.221	0.731	0.193	0.085	**2.864**
23	**Spain**	0.500	0.015	0.013	0.000	0.462	0.492	0.924	0.338	0.025	**2.769**
24	**Hungary**	0.667	0.037	0.006	0.026	0.371	0.111	0.742	0.103	0.033	**2.095**
25	**Czech Republic**	0.583	0.007	0.008	0.000	0.543	0.325	0.215	0.335	0.009	**2.025**
26	**Russia**	0.333	0.004	0.007	0.001	0.000	0.235	0.794	0.058	0.001	**1.432**
27	**Chile**	0.333	0.004	0.001	0.000	0.554	0.000	0.000	0.190	0.001	**1.084**
28	**India**	0.167	0.001	0.001	0.000	0.122	0.190	0.540	0.000	0.000	**1.020**
29	**Mexico**	0.000	0.000	0.001	0.000	0.269	0.025	0.345	0.122	0.001	**0.763**
30	**Brazil**	0.167	0.001	0.001	0.000	0.046	0.144	0.232	0.144	0.001	**0.736**

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

Infrastructure pillar scores breakdown by variable

	Position	EV charging stations	4G coverage	Quality of roads	Technology infrastructure change readiness	Mobile connection speed (0.5 weight)	Broadband (0.5 weight)	Pillar 3 score (unadjusted)
1	**The Netherlands**	1.000	0.832	0.993	0.622	0.755	0.792	**4.221**
2	**South Korea**	0.060	1.000	0.838	0.689	0.959	0.917	**3.525**
3	**Norway**	0.808	0.929	0.448	0.467	0.728	0.958	**3.495**
4	**United Arab Emirates**	0.010	0.636	0.869	1.000	1.000	0.833	**3.431**
5	**Singapore**	0.095	0.739	1.000	0.756	0.578	1.000	**3.379**
6	**Japan**	0.078	0.957	0.894	0.689	0.272	0.958	**3.233**
7	**Austria**	0.166	0.611	0.871	0.844	0.498	0.708	**3.095**
8	**Sweden**	0.290	0.771	0.669	0.578	0.473	0.958	**3.023**
9	**United States**	0.070	0.839	0.714	0.600	0.393	0.917	**2.878**
10	**Denmark**	0.158	0.682	0.744	0.556	0.491	0.875	**2.823**
11	**Finland**	0.068	0.714	0.653	0.644	0.483	0.833	**2.738**
12	**Australia**	0.010	0.743	0.557	0.578	0.693	1.000	**2.735**
13	**Canada**	0.074	0.689	0.587	0.378	0.788	0.917	**2.580**
14	**Taiwan**	0.024	0.588	0.754	0.533	0.453	0.865	**2.558**
15	**Spain**	0.062	0.639	0.782	0.533	0.327	0.750	**2.555**
16	**United Kingdom**	0.141	0.543	0.538	0.689	0.313	0.750	**2.442**
17	**France**	0.150	0.364	0.704	0.533	0.467	0.792	**2.381**
18	**Belgium**	0.192	0.746	0.399	0.333	0.516	0.708	**2.283**
19	**Germany**	0.165	0.264	0.666	0.600	0.328	0.667	**2.192**
20	**New Zealand**	0.021	0.250	0.420	0.711	0.522	0.917	**2.121**
21	**Hungary**	0.023	0.782	0.293	0.333	0.433	0.542	**1.919**
22	**China**	0.079	0.581	0.456	0.267	0.751	0.250	**1.884**
23	**Italy**	0.024	0.339	0.406	0.622	0.318	0.625	**1.863**
24	**Czech Republic**	0.033	0.754	0.261	0.289	0.496	0.542	**1.856**
25	**Israel**	0.108	0.000	0.537	0.578	0.146	0.833	**1.712**
26	**Chile**	0.002	0.257	0.638	0.422	0.117	0.542	**1.648**
27	**Russia**	0.001	0.157	0.136	0.622	0.117	0.625	**1.287**
28	**Mexico**	0.007	0.368	0.434	0.200	0.218	0.333	**1.284**
29	**India**	0.000	0.764	0.437	0.000	0.000	0.000	**1.202**
30	**Brazil**	0.001	0.089	0.000	0.311	0.171	0.417	**0.695**

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

Consumer acceptance pillar scores breakdown by variable

	Position	Population living near test areas	Civil society technology use	Consumer ICT adoption	Digital skills	Individual readiness	Online ride-hailing market penetration	Pillar 4 score (unadjusted)
1	**Singapore**	1.000	0.514	0.906	0.910	0.715	0.828	**4.873**
2	**Finland**	0.364	0.886	0.796	1.000	0.673	1.000	**4.718**
3	**Sweden**	0.353	1.000	0.918	0.941	0.641	0.524	**4.377**
4	**United Arab Emirates**	0.210	0.543	0.985	0.814	1.000	0.719	**4.271**
5	**Norway**	0.342	0.857	0.840	0.805	0.705	0.528	**4.078**
6	**United States**	0.324	0.914	0.695	0.818	0.636	0.682	**4.069**
7	**The Netherlands**	0.811	0.814	0.728	0.926	0.624	0.131	**4.034**
8	**Denmark**	0.574	0.729	0.843	0.849	0.734	0.199	**3.927**
9	**Australia**	0.365	0.786	0.684	0.705	0.719	0.412	**3.670**
10	**South Korea**	0.216	0.514	1.000	0.694	0.690	0.483	**3.597**
11	**Israel**	0.562	0.643	0.585	0.880	0.472	0.412	**3.553**
12	**United Kingdom**	0.305	0.714	0.674	0.674	0.541	0.607	**3.515**
13	**Canada**	0.477	0.729	0.629	0.724	0.444	0.453	**3.457**
14	**New Zealand**	0.342	0.757	0.751	0.672	0.583	0.315	**3.420**
15	**Taiwan**	0.465	0.171	0.827	0.748	0.749	0.435	**3.396**
16	**China**	0.043	0.571	0.764	0.573	0.419	0.993	**3.364**
17	**Spain**	0.000	0.329	0.759	0.457	0.676	0.539	**2.761**
18	**Japan**	0.302	0.286	0.891	0.490	0.709	0.000	**2.678**
19	**France**	0.284	0.386	0.685	0.512	0.407	0.348	**2.622**
20	**Russia**	0.000	0.329	0.740	0.678	0.394	0.442	**2.583**
21	**Germany**	0.096	0.529	0.624	0.722	0.483	0.127	**2.581**
22	**Czech Republic**	0.000	0.500	0.598	0.617	0.364	0.416	**2.494**
23	**Belgium**	0.000	0.657	0.575	0.635	0.485	0.116	**2.468**
24	**Austria**	0.000	0.457	0.552	0.617	0.504	0.333	**2.463**
25	**Chile**	0.000	0.257	0.511	0.429	0.612	0.367	**2.176**
26	**Italy**	0.125	0.271	0.534	0.396	0.464	0.000	**1.790**
27	**Mexico**	0.000	0.100	0.377	0.245	0.375	0.464	**1.561**
28	**Hungary**	0.000	0.114	0.529	0.322	0.221	0.266	**1.451**
29	**Brazil**	0.104	0.000	0.428	0.000	0.306	0.476	**1.314**
30	**India**	0.000	0.157	0.000	0.490	0.000	0.427	**1.074**

Autonomous Vehicles Readiness Index: AVRI [online]. [cit. 2020-01-25]. Dostupné z: https://home.kpmg/xx/en/home/insights/2020/06/autonomous-vehicles-readiness-index.html

What is Object Detection?

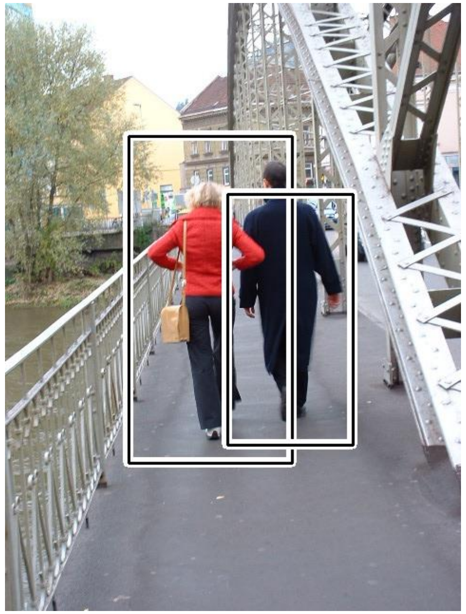
- It is clear that the images contain many objects of interest. The goal of the object detection
systems is to find the location of these objects in the images (e.g. cars, faces, pedestrians).

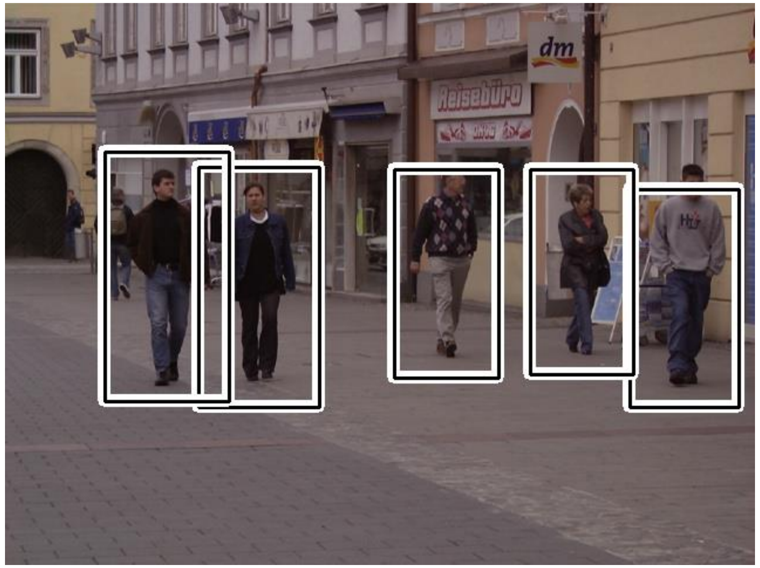
- For example, the vehicle detection systems are crucial for traffic analysis or intelligent scheduling, the people detection systems can be useful for automotive safety, and the face detection systems are a key part of face recognition systems.

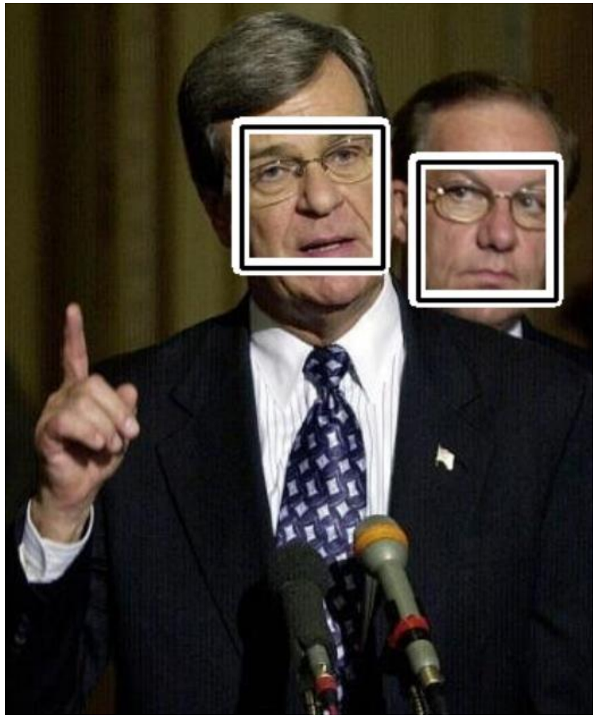
What is Object Detection?

What is Object Detection?

- Output?
  - position of the objects
  - scale of the objects







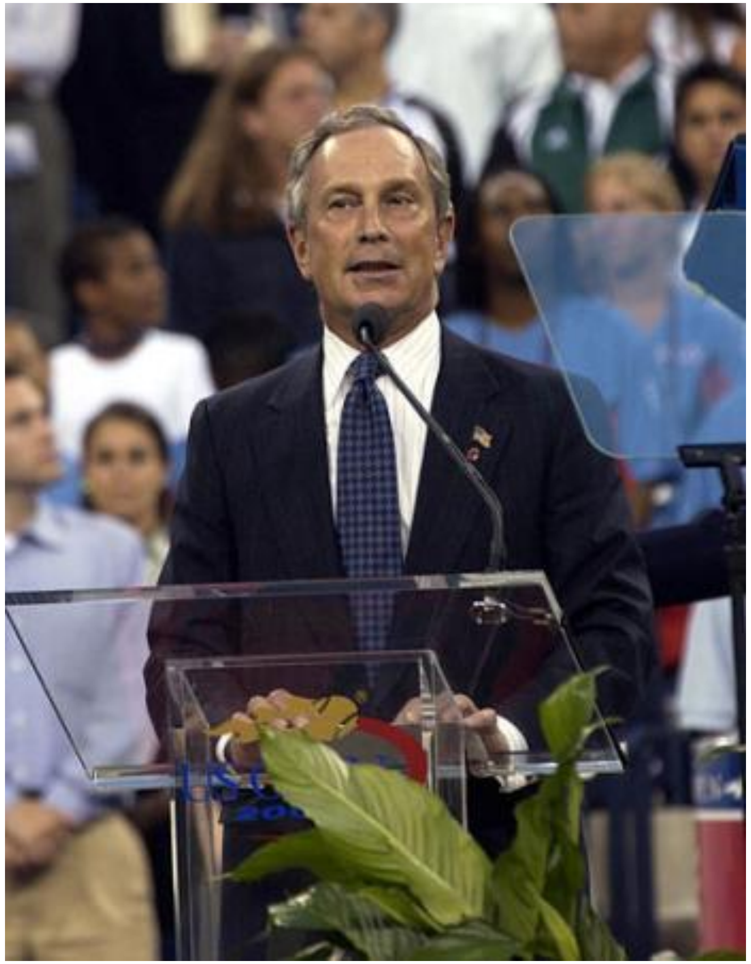
Multi-Class Vehicle Detection with dlib's MMOD+CNN Detector [online]. [cit. 2020-01-25]. Dostupné z: https://youtu.be/OHbJ7HhbG74

Problems (Challenges)

- different views
- Illumination challenges
- occlusion
- different backgrounds
- shadows
- …
- …

Problems (Challenges)

- low quality images



Face Detection Data Set and Benchmark Home [online]. [cit. 2020-01-25]. Dostupné z: http://vis-www.cs.umass.edu/fddb/

Problems (Challenges)

- illumination + low quality

Face Detection Data Set and Benchmark Home [online]. [cit. 2020-01-25]. Dostupné z: http://vis-www.cs.umass.edu/fddb/

Image Features

- The objects of interest can be described using various image information (e.g. shape, texture, colour). In the area of feature based detectors the image features are the carries of this information.

- Many methods for extracting the image features that are able to describe the appearance of objects were presented, especially, the detectors that are based on the histograms of oriented gradients (HOG), Haar features, or local binary patterns (LBP) are dominant and they are considered as the state-of-the-art methods.

Object Detection/Recognition

- Haar
- HOG
- LBP

Traditional Approaches

- SIFT, SURF

KeyPoints

- CNNs

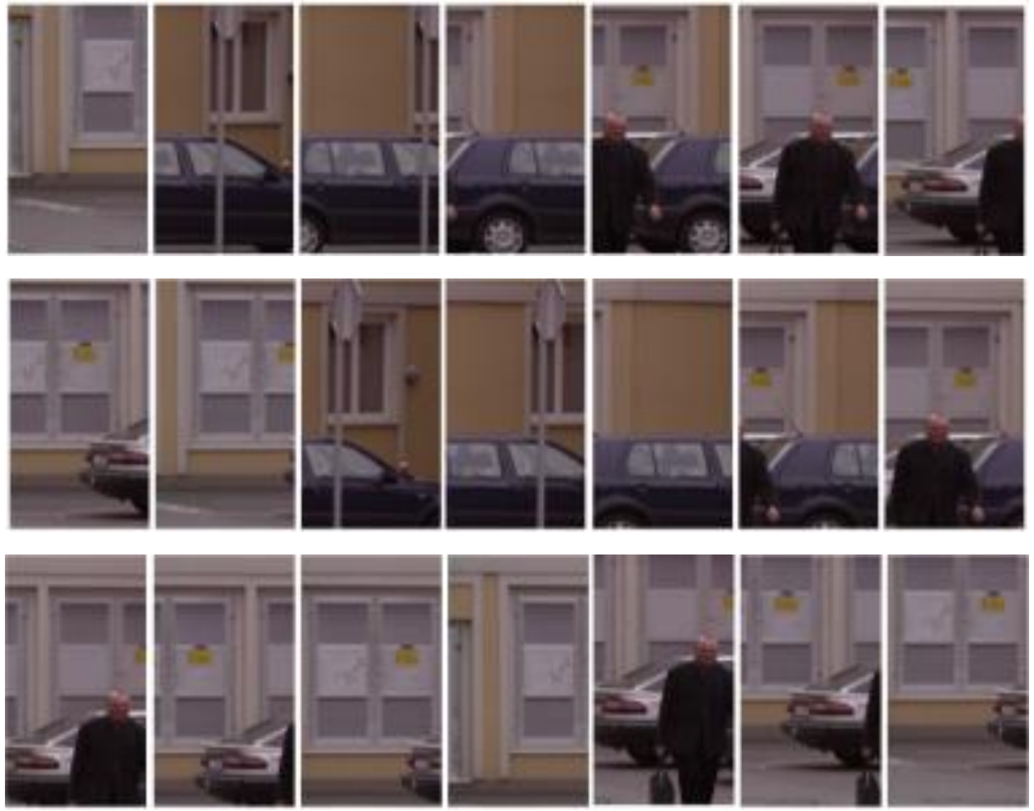
Deep Learning Approach

- Practical examples using OpenCV + Dlib (https://opencv.org/, http://dlib.net/)

Sliding Window - Main Idea

Constantine Papageorgiou and Tomaso Poggio: A Trainable System for Object Detection. *Int. J. Comput. Vision* 38, pp. 15-33. (2000)

Sliding Window - Main Idea



Constantine Papageorgiou and Tomaso Poggio: A Trainable System for Object Detection. *Int. J. Comput. Vision* 38, pp. 15-33. (2000)

Sliding Window - Main Idea

- In general, the sliding window technique represents the popular and successful approach for object detection. The main idea of this approach is that the input image is scanned by a rectangular window at multiple scales. The result of the scanning process is a large number of various sub-windows. A vector of features is extracted from each sub-window. The vector is then used as an input for the classifier (e.g. SVM classifer).

- During the classification process, some sub-windows are marked as the objects. Using the sliding window approach, the multiple positive detections may appear, especially around the objects of interest

Sliding Window - Main Idea

- These detections are merged to the final bounding box that represents the resulting detection.
- The classifer that determines each sub-window is trained over the training set that consists of positive and negative images.
- The key point is to find what values (features) should be used to effectively encode the image inside the sliding window.

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Detection Process

Feature Vector

(gradient, HOG, LBP, …)

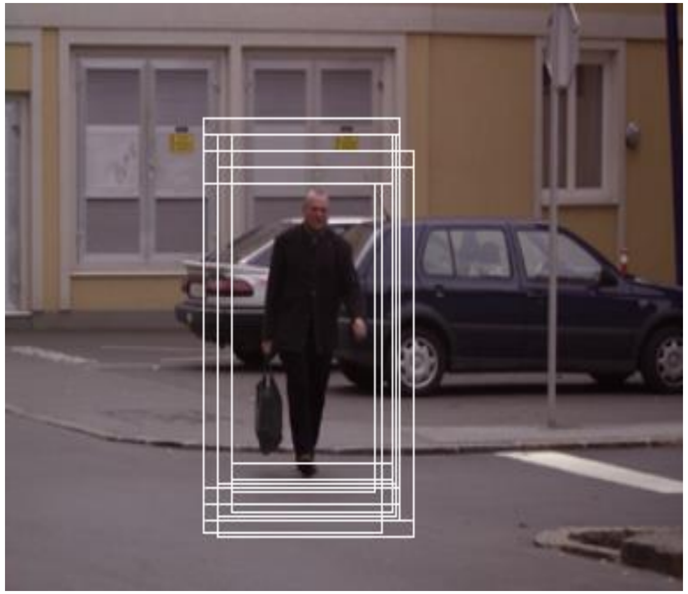
Detection Process

Feature Vector

(gradient, HOG, LBP, …)

Trainable Classifier

(SVM, ANNs, …)



Detection Process

- Typically, in the area of feature-based detectors, the detection algorithms consist of two main parts. The extraction of image features is the first part. The second part is created by the trainable classifiers that handle a final classification (object/non-object).

- The extraction of relevant features has a significant influence on the successfulness of detectors. The large number of features slows down the training and detection phases; on the other hand a very small number of features may not be able to describe the properties of object of interest. The quality of training set is also equally important.

Training Sets

- negative set - without the object of interest
- positive set
  - rotation
  - noise
  - Illumination
  - scale

Training Set – Traffic Sign

The German Traffic Sign Recognition/Detection Benchmark

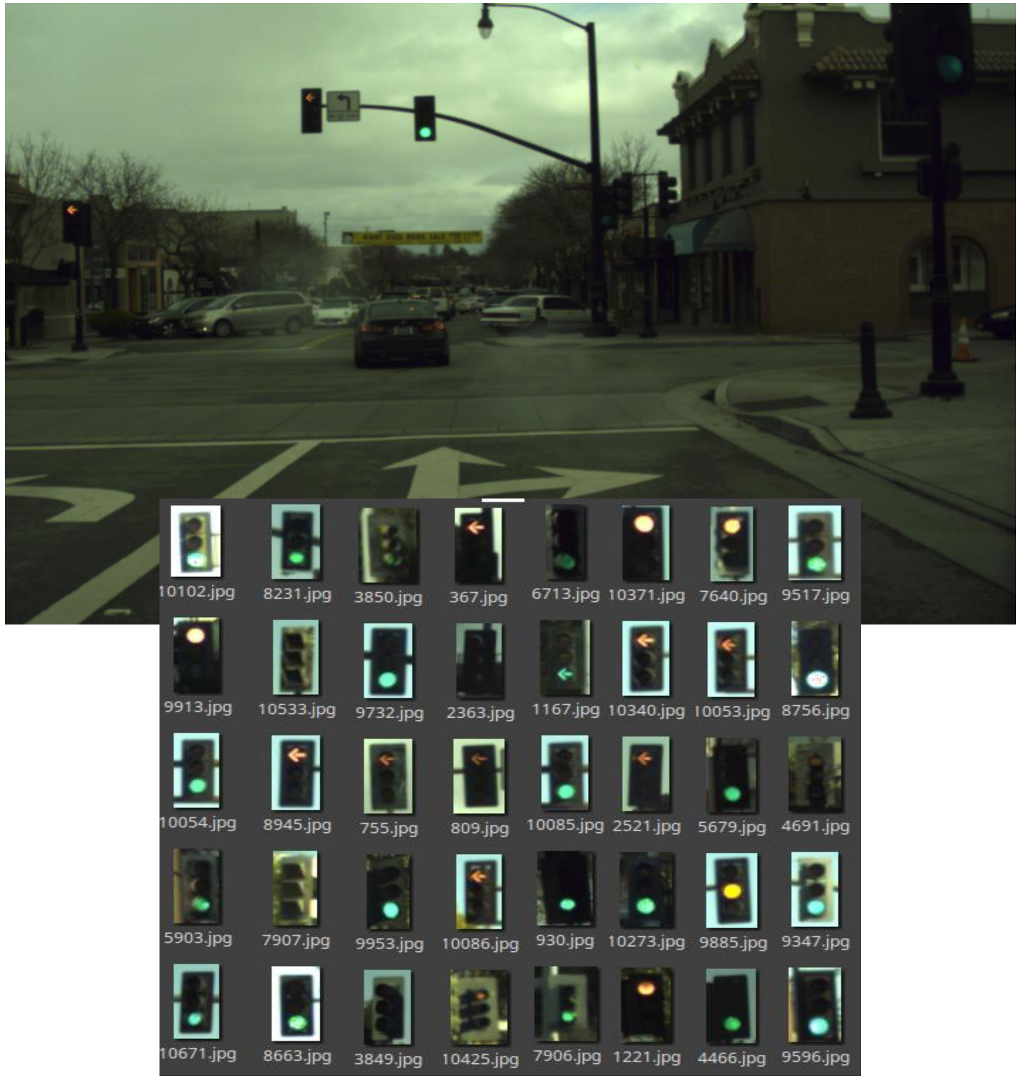
- Single-image, multi-class classification problem
- More than 40 classes
- More than 50,000 images in total
- Large, lifelike database
- Reliable ground-truth data due to semi-automatic annotation
- Physical traffic sign instances are unique within the dataset
  (i.e., each real-world traffic sign only occurs once)

The German Traffic Sign Recognition Benchmark [online]. [cit. 2020-01-25]. Dostupné z: https://benchmark.ini.rub.de/gtsrb_news.html

Training Set – Traffic Lights

Bosch Small Traffic Lights Dataset (Germany)

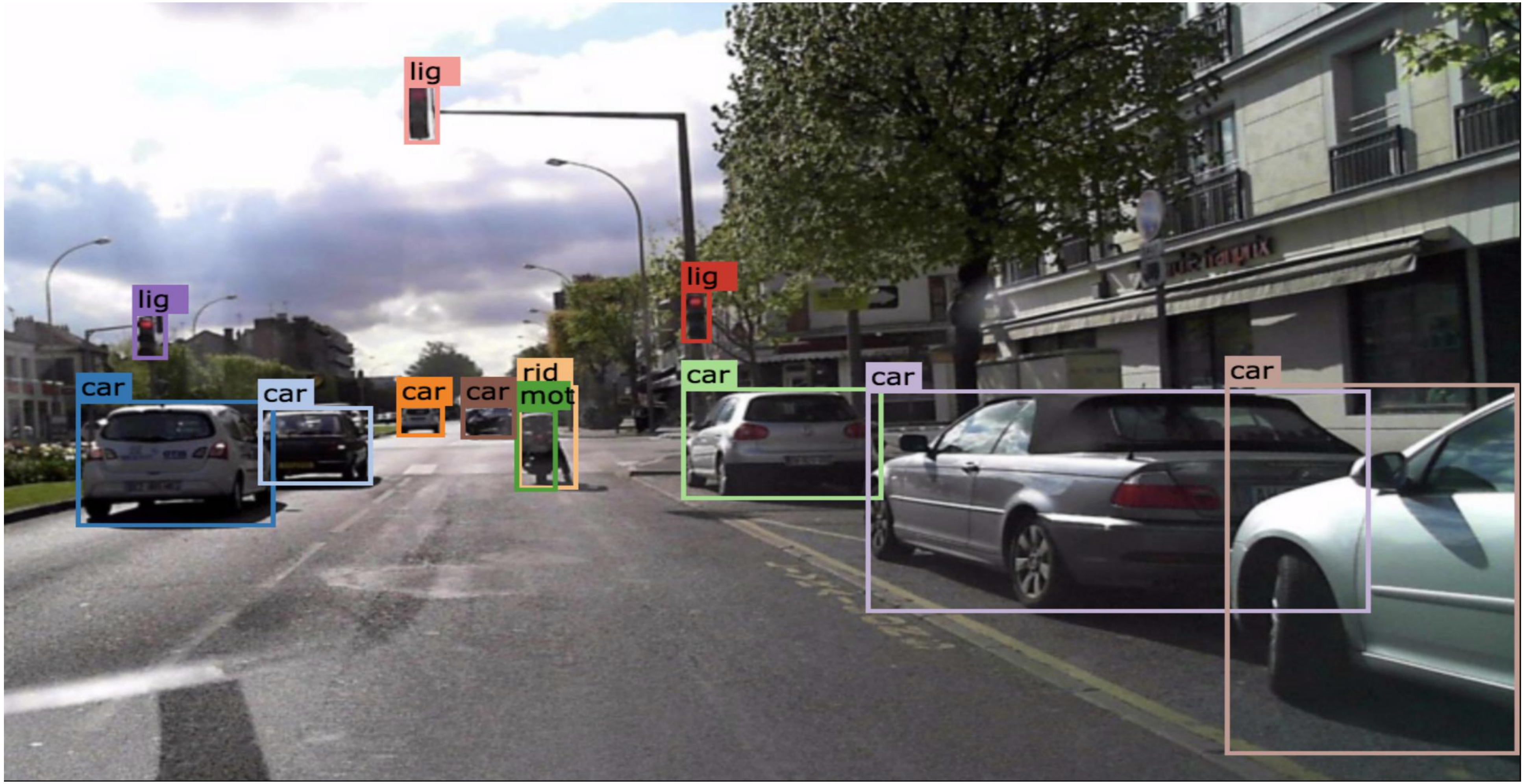
- Training set:
  - 5093 images
  - Annotated about every 2 seconds
  - 10756 annotated traffic lights
  - Median traffic lights width: ~8.6 pixels
  - 15 different labels
  - 170 lights are partially occluded
- Test set:
  - 8334 consecutive images
  - Annotated at about 15 fps
  - 13486 annotated traffic lights
  - Median traffic light width: 8.5 pixels
  - 4 labels (red, yellow, green, off)
  - 2088 lights are partially occluded



Bosch Small Traffic Lights Dataset [online]. [cit. 2020-01-25]. Dostupné z: https://hci.iwr.uni-heidelberg.de/content/bosch-small-traffic-lights-dataset

Training Set – Road Objects

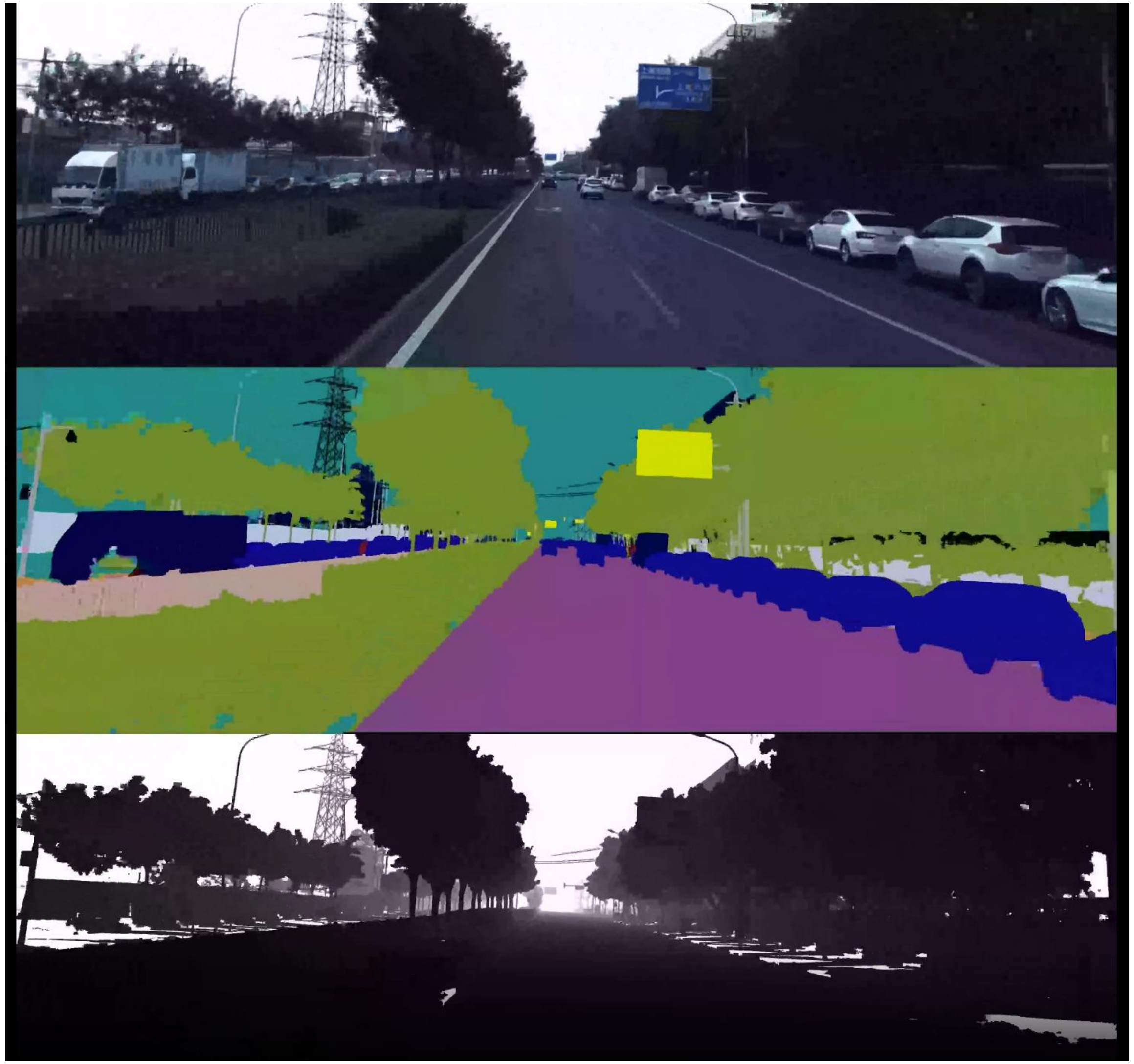
Berkeley Deep Drive



Berkeley DeepDrive [online]. [cit. 2020-01-25]. Dostupné z: https://bdd-data.berkeley.edu/

Training Set – Road Objects

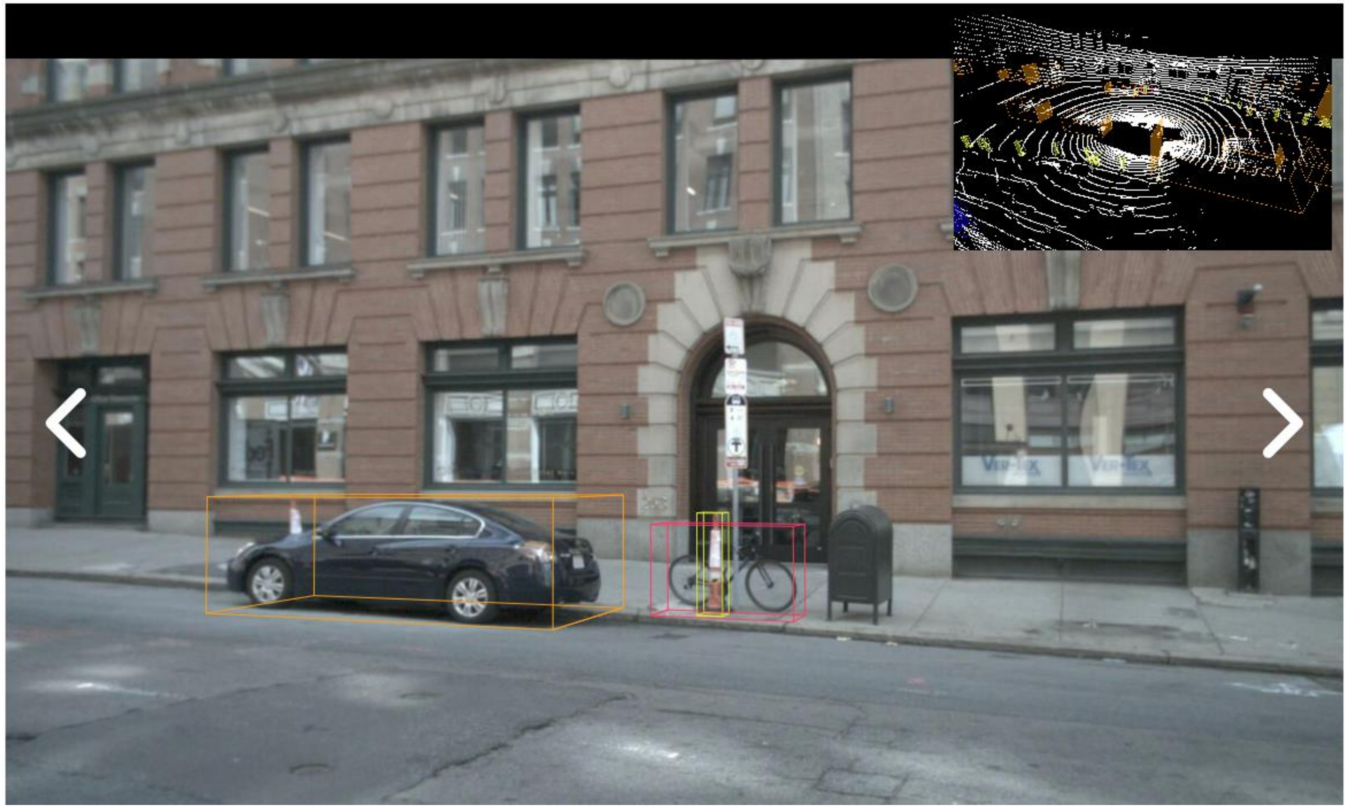
http://apolloscape.auto/

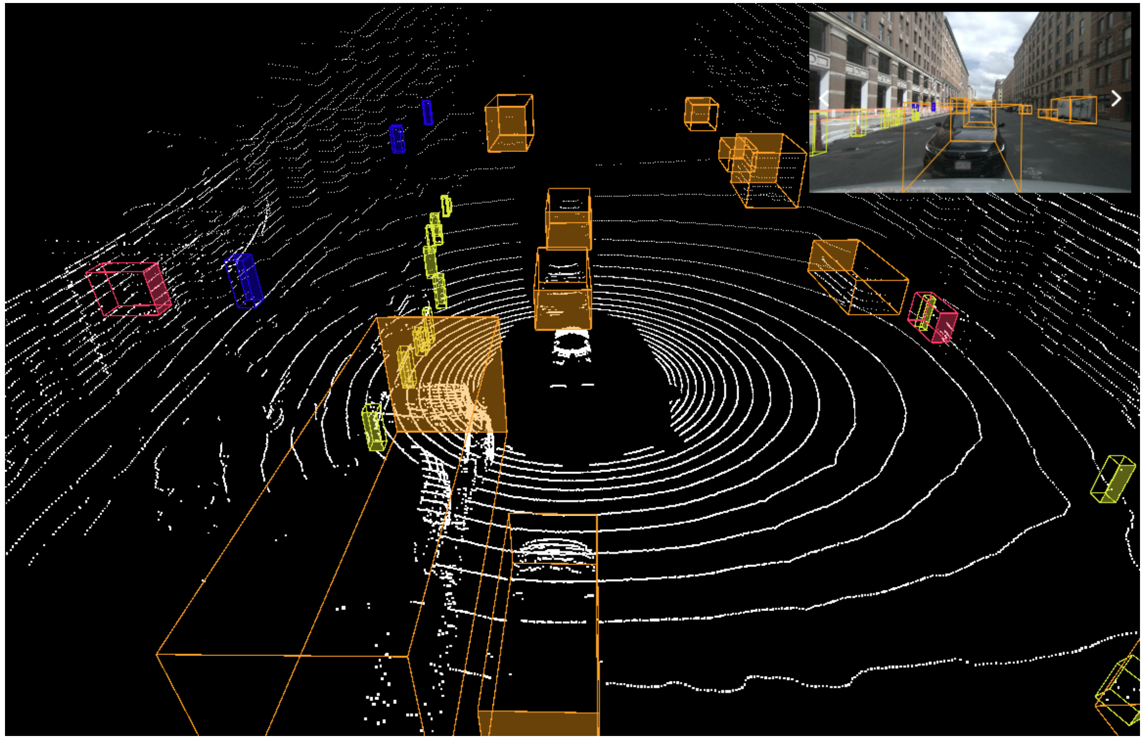


ApolloScape Dataset [online]. [cit. 2020-01-25]. Dostupné z: http://apolloscape.auto/

Training Set – Road Objects

nuScenes





NuScenes dataset [online]. [cit. 2020-01-25]. Dostupné z: https://www.nuscenes.org/nuscenes https://boxy-dataset.com/boxy/

Training Set – Road Objects

THE BOXY VEHICLES DATASET [online]. [cit. 2020-01-25]. Dostupné z: https://boxy-dataset.com/boxy/

The Boxy Vehicle Dataset and Baselines [online]. [cit. 2020-01-25]. Dostupné z: https://www.youtube.com/watch?v=8kwvPJczhd4&t=27s

- Haar
- HOG
- LBP

$\Big\}$ Traditional Approaches

- SIFT, SURF

$\Big\}$ KeyPoints

- CNNs

$\Big\}$ Deep Learning Approach

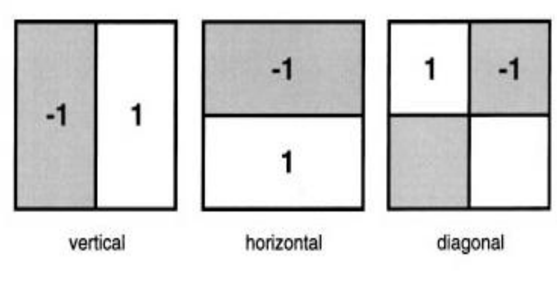
Related Works

2000

Papageorgiou (2000)

A Trainable System for Object Detection

CONSTANTINE PAPAGEORGIOU AND TOMASO POGGIO
Center for Biological and Computational Learning, Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
cpapa@ai.mit.edu
tp@ai.mit.edu

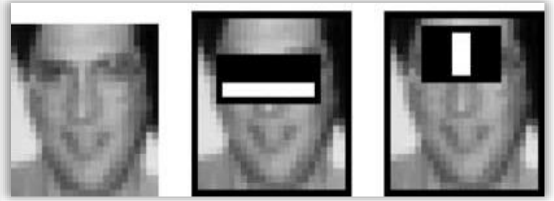


Viola, Jones (2001,2004) cit. > 6500

Robust Real-Time Face Detection

PAUL VIOLA
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
viola@microsoft.com

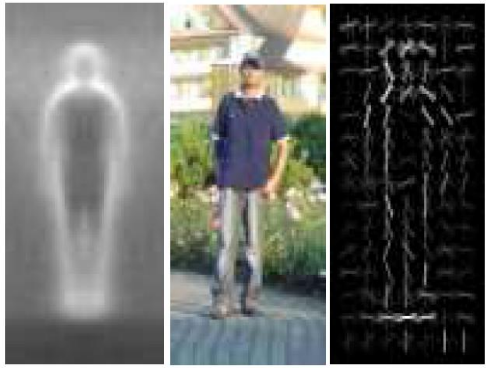
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs (2005) cit. > 10000

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr



2005

Haar Wavelet-based Descriptors

- The main idea behind the Haar-like features is that the features can encode the differences of mean intensities between the rectangular areas. For instance, in the problem of face detection, the regions around the eyes are lighter than the areas of the eyes; the regions bellow or on top of eyes have different intensities that the eyes themselves.

- These specific characteristics can be simply encoded by one two-rectangular feature, and the value of this feature can be calculated as the difference between the sum of the intensities inside the rectangles.

Haar Wavelet-based Descriptors

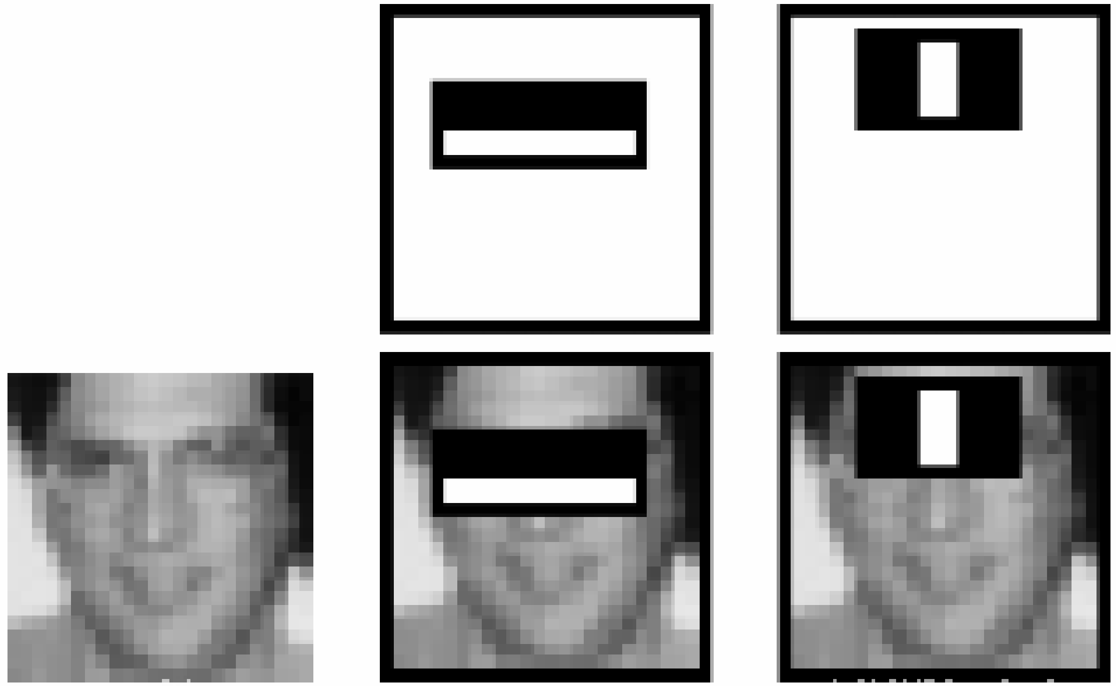
- The paper of Viola and Jones contributed to the popularity of Haar-like features. The authors proposed the object detection framework based on the image representation called the integral image combined with the rectangular features, and the AdaBoost algorithm.

- With the use of integral image, the rectangular features are computed very quickly. The AdaBoost algorithm helps to select the most important features.

- The features are used to train classifers and the cascade of classifers is used for reducing the computational time.

Haar Wavelet-based Descriptors

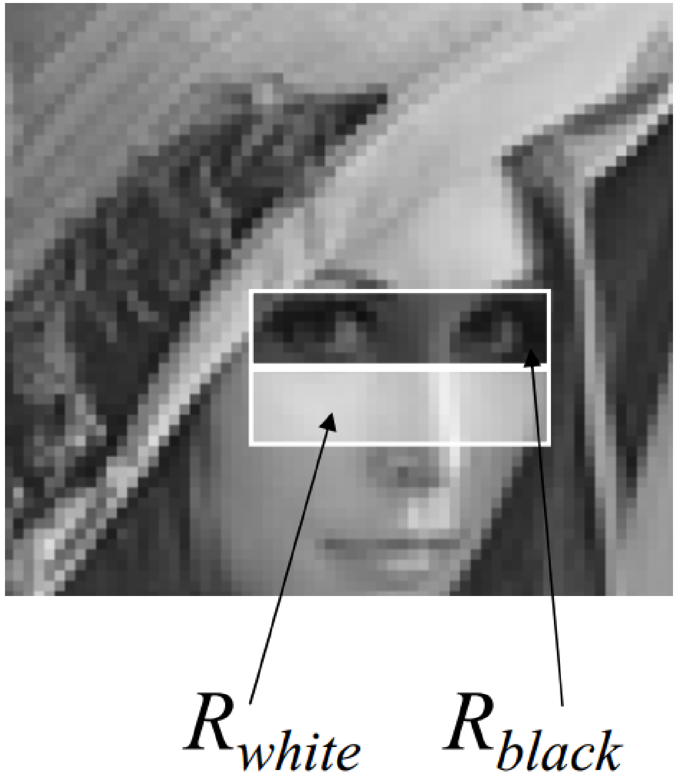
- faces have similar properties
  - eye regions are darker than the upper-cheeks
  - the nose bridge region is brighter than the eyes

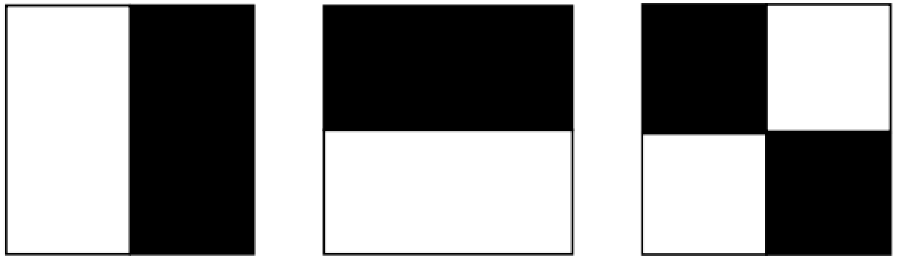


Face Detection using Haar Cascades [online]. [cit. 2020-01-25]. Dostupné z: https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

Features

- Rectangular features

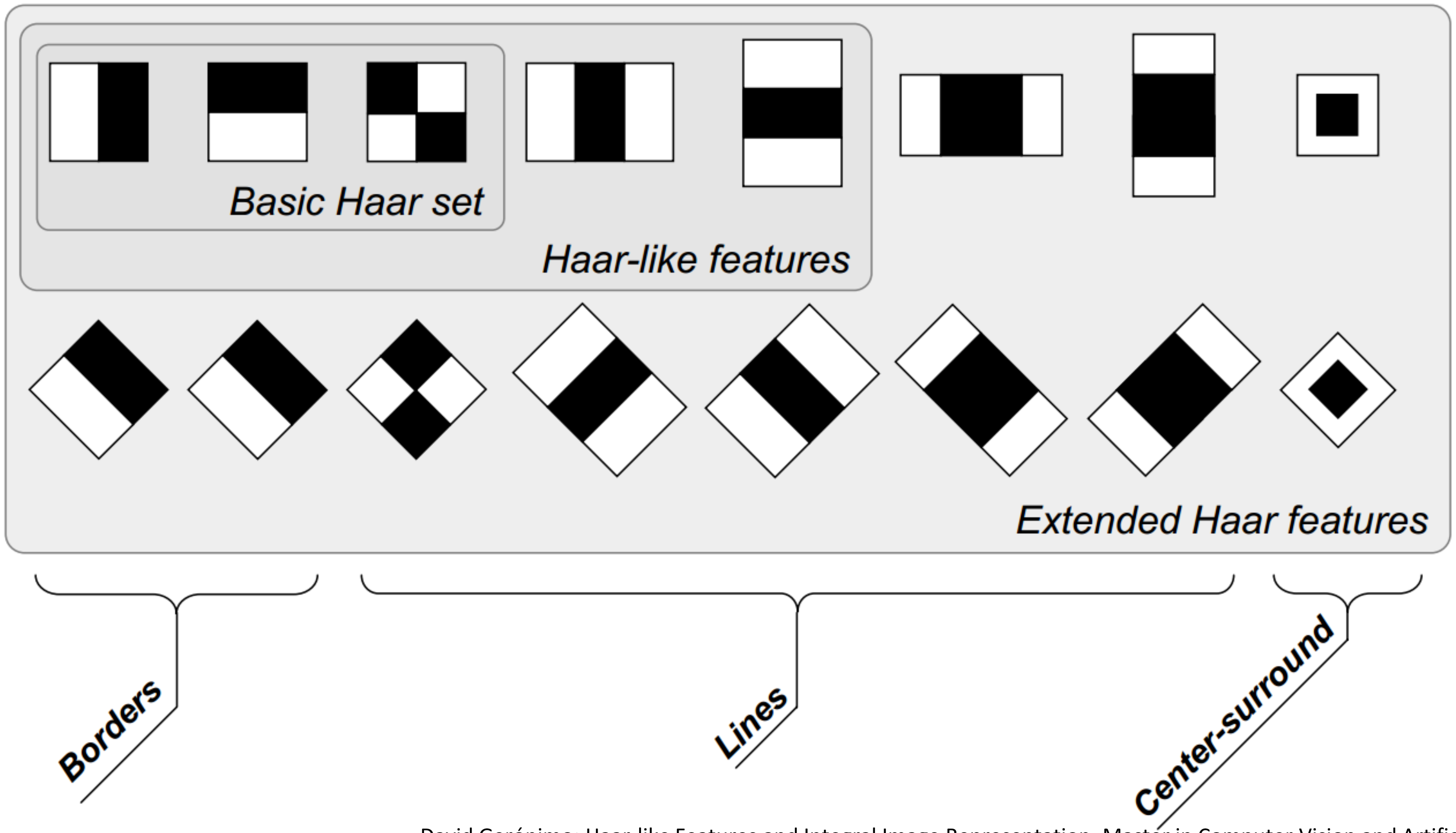




$$F_{Haar} = E(R_{white}) - E(R_{black})$$

Features

Different sets



David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009

Face Detection

- 24x24 sub-window aprox. 160,000 rectangular features
- How speed the computational speed?
  - decrease memory accesses

David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009

Integral Image

1	1	1
1	1	1
1	1	1

Original image *(i)*

1	2	3
2	4	6
3	6	9

Integral image (ii)

Integral Image

Original image (I)

7	7	7	7	4	8	4	1
7	7	7	8	4	4	1	1
7	7	8	4	8	4	4	1
7	8	4	7	**8**	4	1	1
7	8	7	7	4	8	1	1
7	8	7	7	8	8	1	1

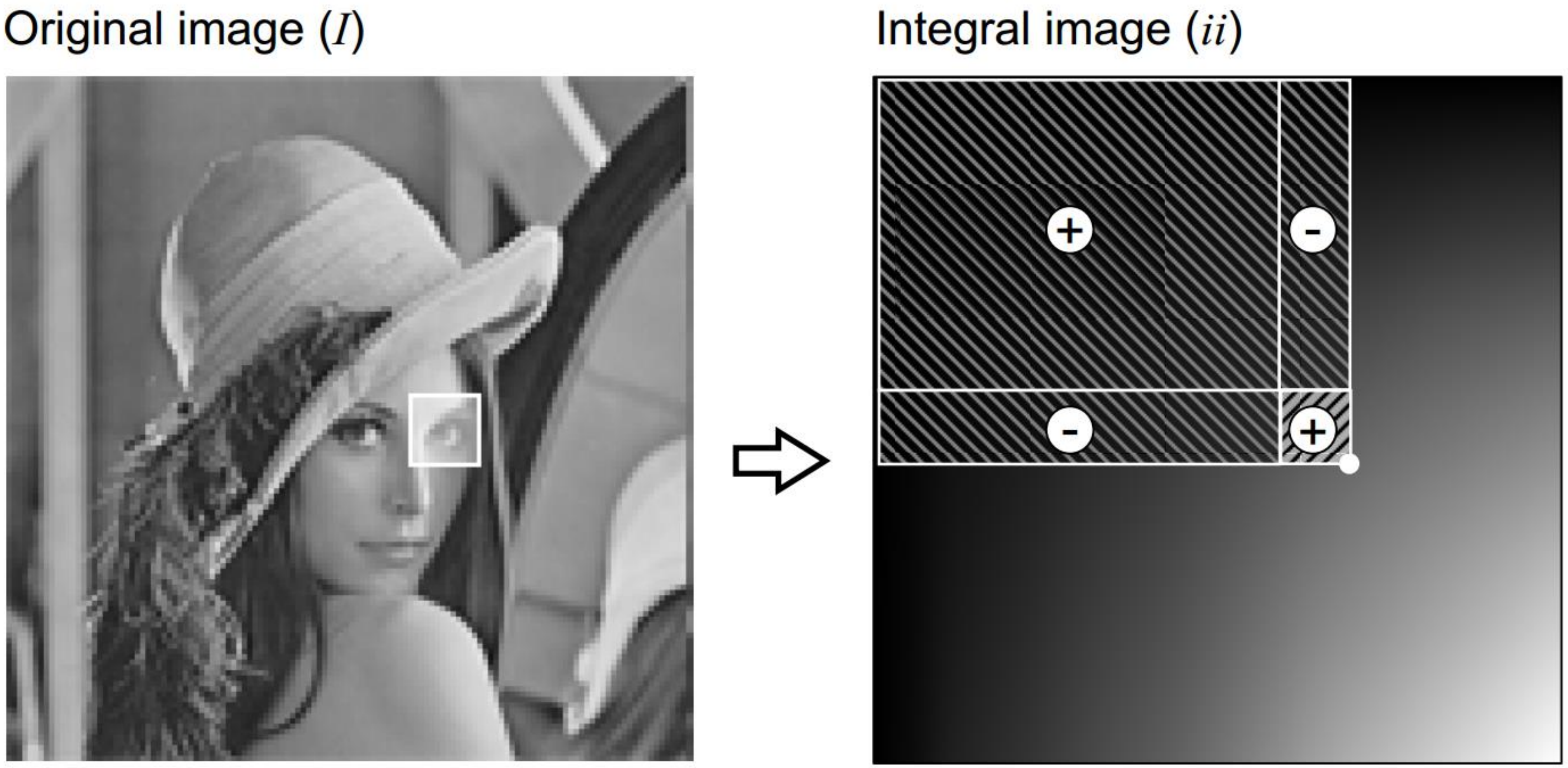
Integral image (ii)

7	14	21	28	32	40	44	45
14	28	42	57	65	77	82	84
21	42	64	83	99	119	124	127
28	57	83	109	**133**			

$$133 = 99 + 109 - 83 + 8$$

David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009

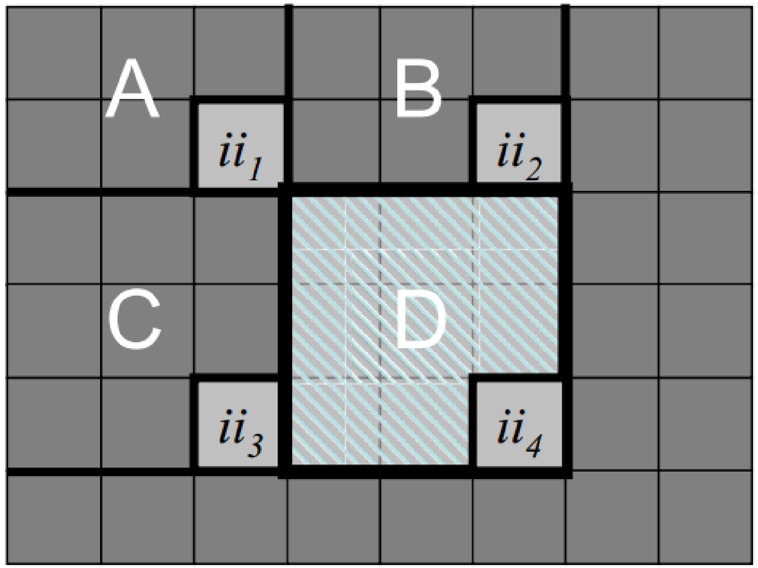
Integral Image



David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009

Integral Image

Integral image (ii)

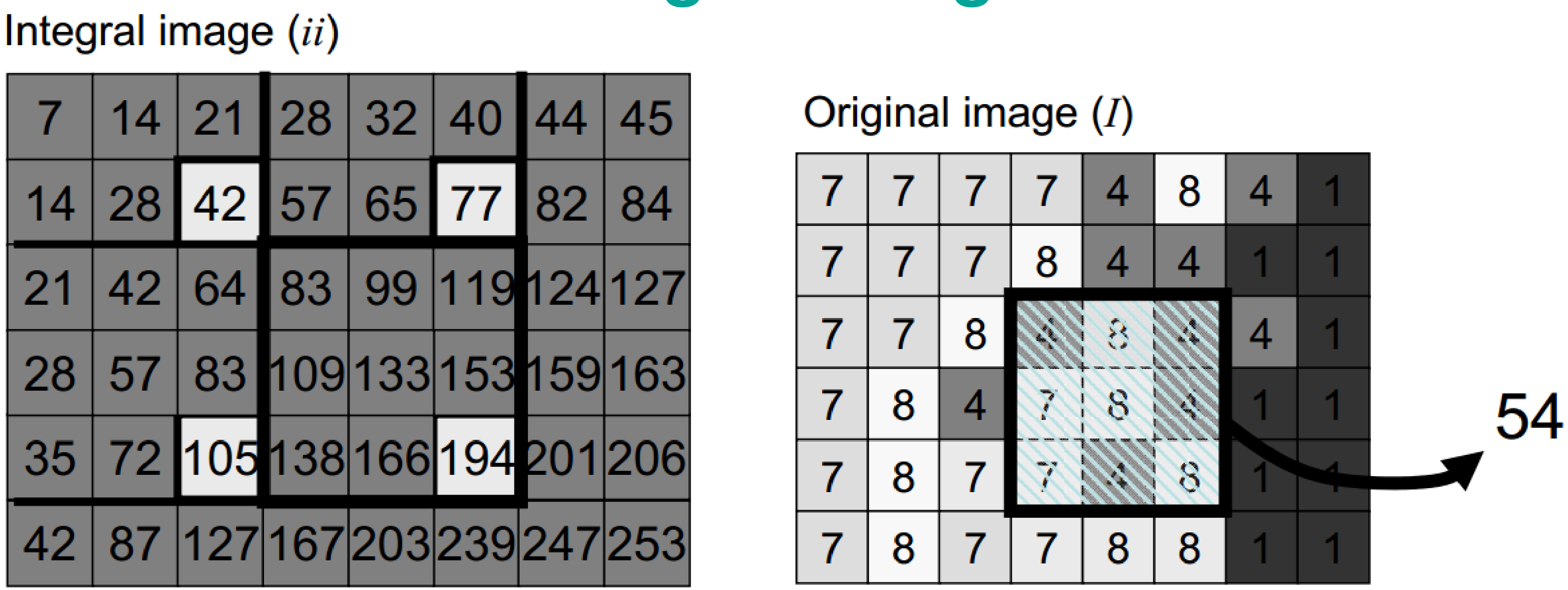


$$ii_1 = \mathrm{sum}(A)$$
$$ii_2 = \mathrm{sum}(A) + \mathrm{sum}(B)$$
$$ii_3 = \mathrm{sum}(A) + \mathrm{sum}(C)$$
$$ii_4 = \mathrm{sum}(A) + \mathrm{sum}(B) + \mathrm{sum}(C) + \mathrm{sum}(D)$$

$$\mathrm{sum}(D) = ii_4 + ii_1 - ii_2 - ii_3$$

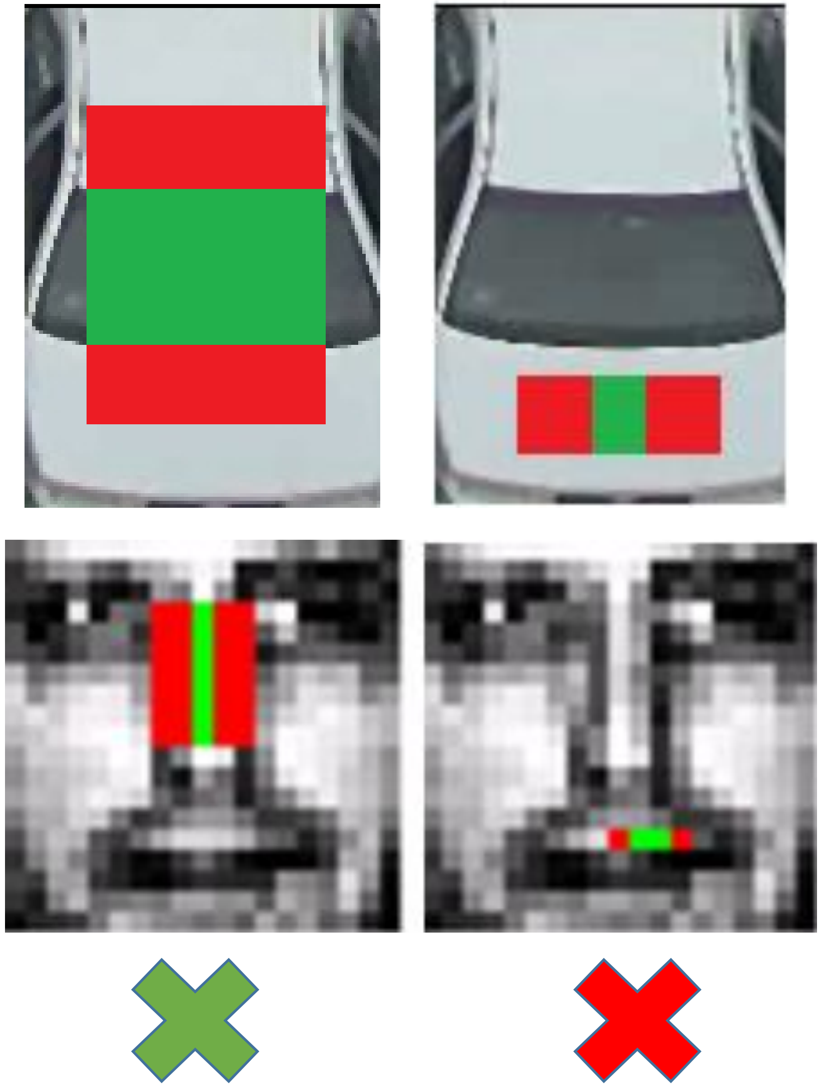
David Gerónimo: Haar-like Features and Integral Image Representation, Master in Computer Vision and Artificial Intelligence, 18th December 2009

Integral Image



$$54 = 194 + 42 - 77 - 105$$

Feature Selection



Feature Selection

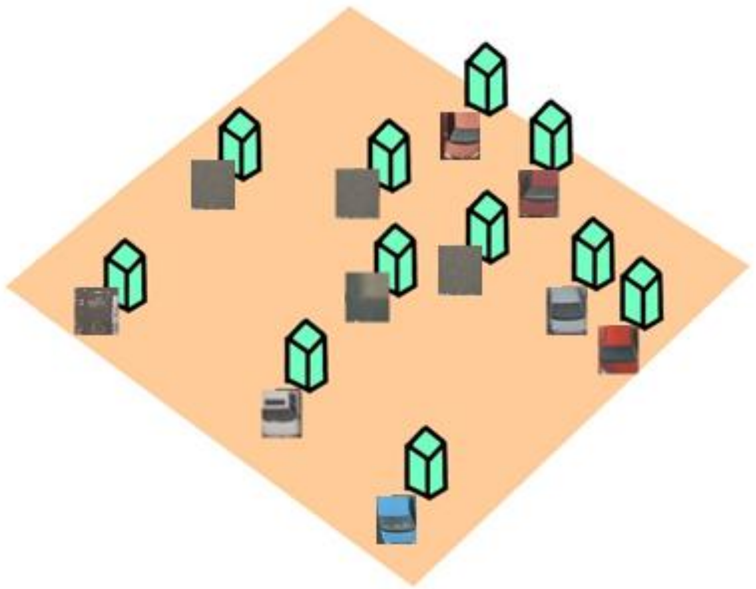
- AdaBoost (Adaptive Boost) is an iterative learning algorithm to construct a “strong” classifier as a linear combination of weighted simple “weak” classifiers

- weak classifier - each single rectangle feature (features as weak classifiers)

- during each iteration, each example/image receives a weight determining its importance

Feature Selection

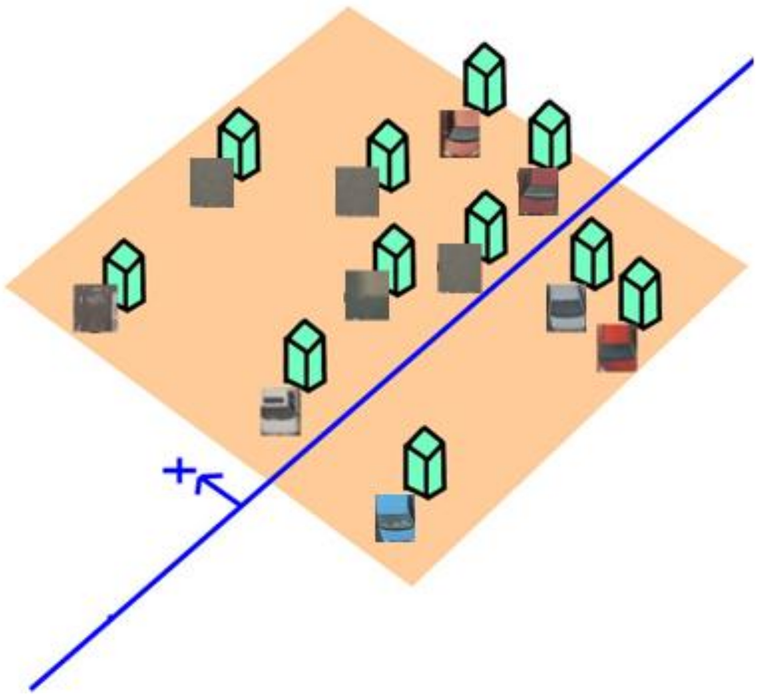
- AdaBoost starts with a uniform distribution of “weights” over training examples.



Feature Selection

- AdaBoost starts with a uniform distribution of “weights” over training examples.

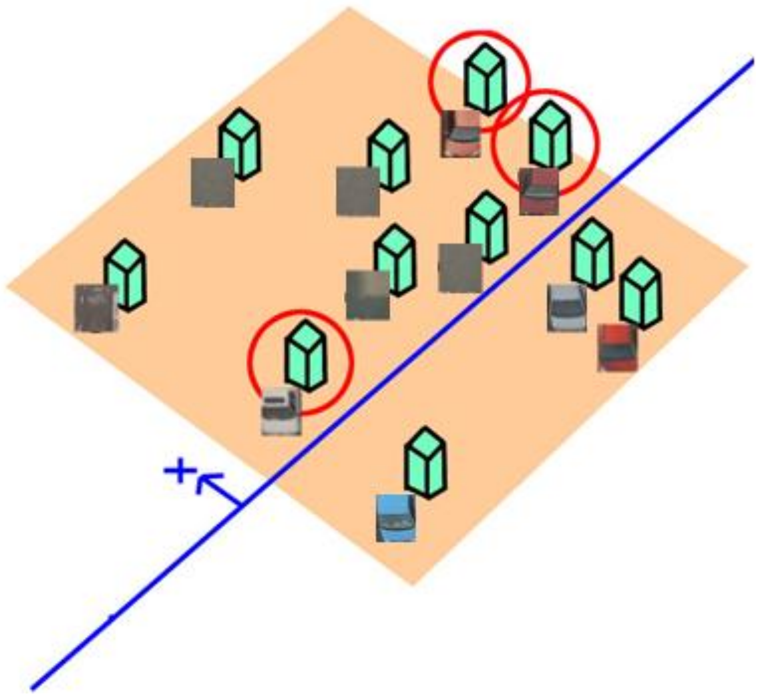
- Select the classifier with the lowest weighted error (i.e. a “weak” classifier)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection

- AdaBoost starts with a uniform distribution of “weights” over training examples.
- Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- Increase the weights on the training examples that were misclassified.



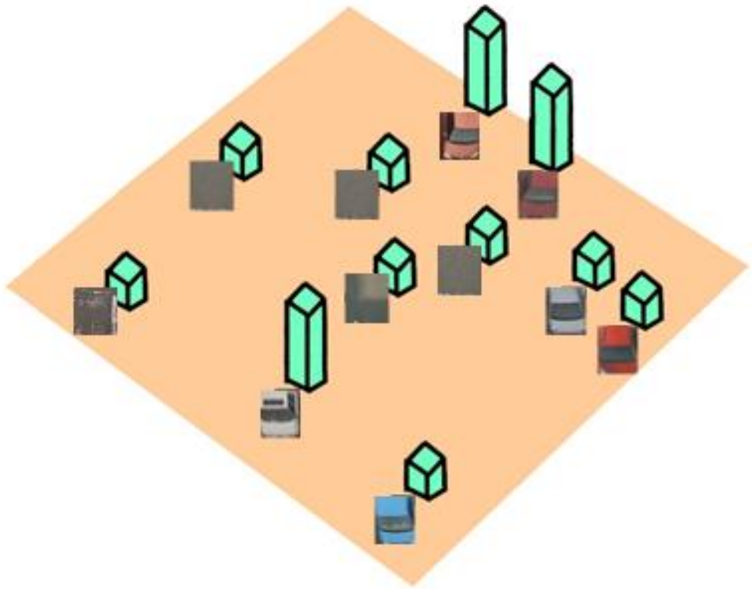
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Feature Selection

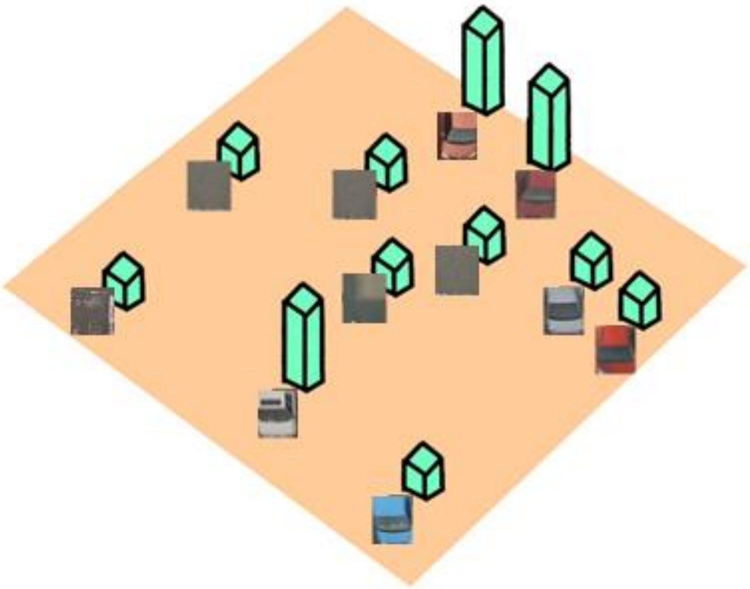
- AdaBoost starts with a uniform distribution of “weights” over training examples.
- Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- Increase the weights on the training examples that were misclassified.



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection

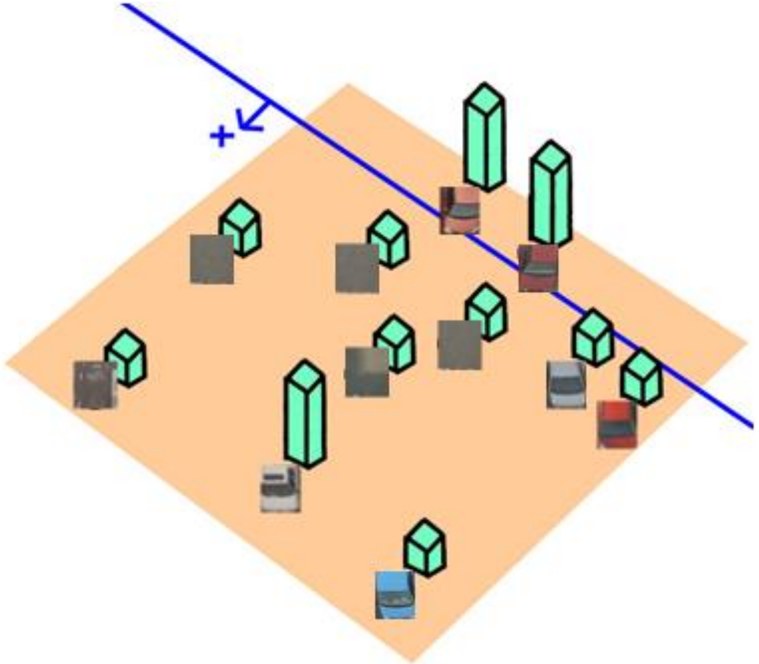
- AdaBoost starts with a uniform distribution of “weights” over training examples.
- Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- Increase the weights on the training examples that were misclassified.
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection

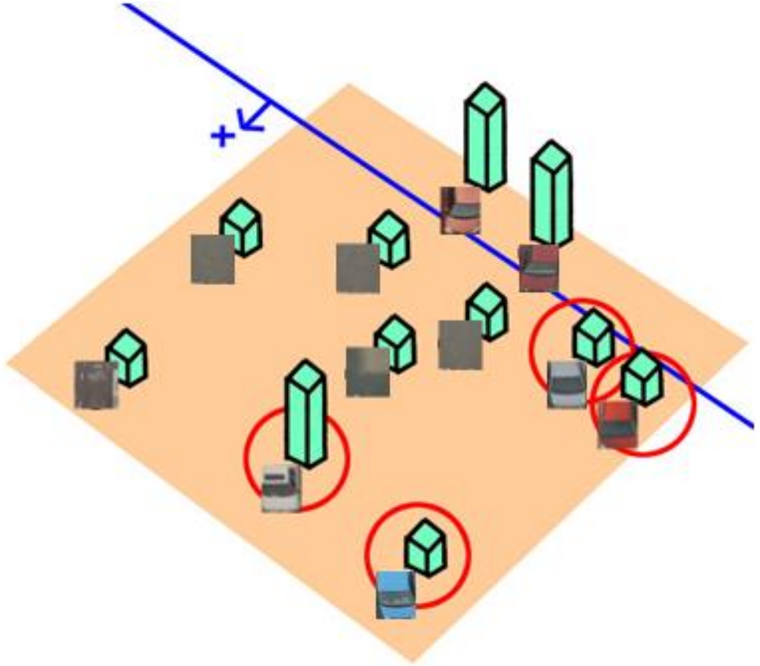
- AdaBoost starts with a uniform distribution of "weights" over training examples.
- **Select the classifier with the lowest weighted error (i.e. a "weak" classifier)**
- Increase the weights on the training examples that were misclassified.
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection

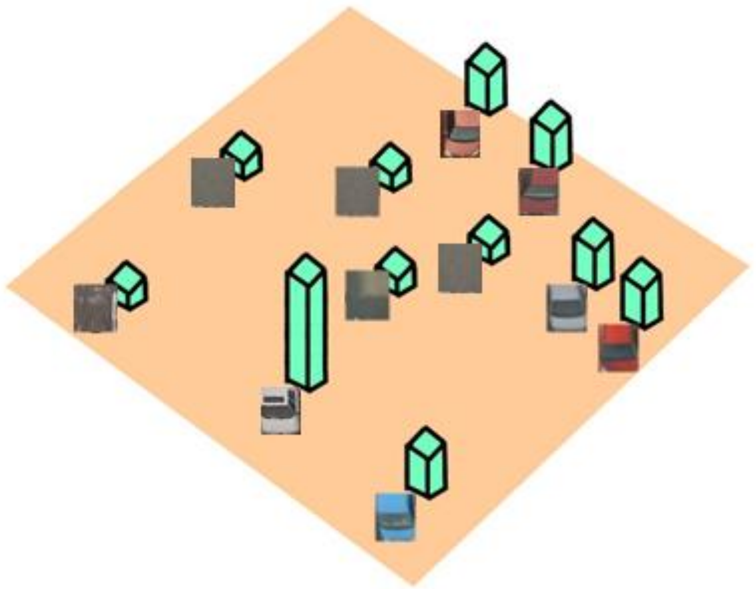
- AdaBoost starts with a uniform distribution of "weights" over training examples.
- **Select the classifier with the lowest weighted error (i.e. a "weak" classifier)**
- **Increase the weights on the training examples that were misclassified.**
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection

- AdaBoost starts with a uniform distribution of “weights” over training examples.
- **Select the classifier with the lowest weighted error (i.e. a “weak” classifier)**
- **Increase the weights on the training examples that were misclassified.**
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

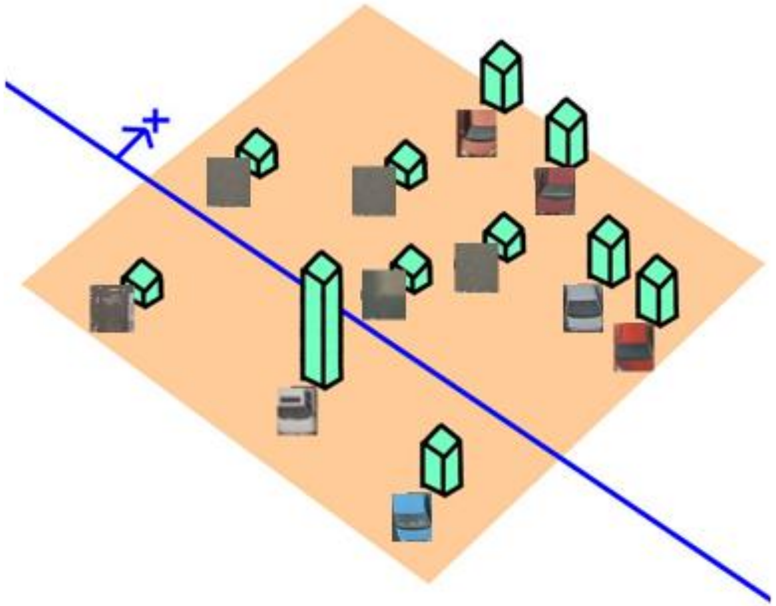
Feature Selection

- AdaBoost starts with a uniform distribution of “weights” over training examples.

- **Select the classifier with the lowest weighted error (i.e. a “weak” classifier)**

- **Increase the weights on the training examples that were misclassified.**

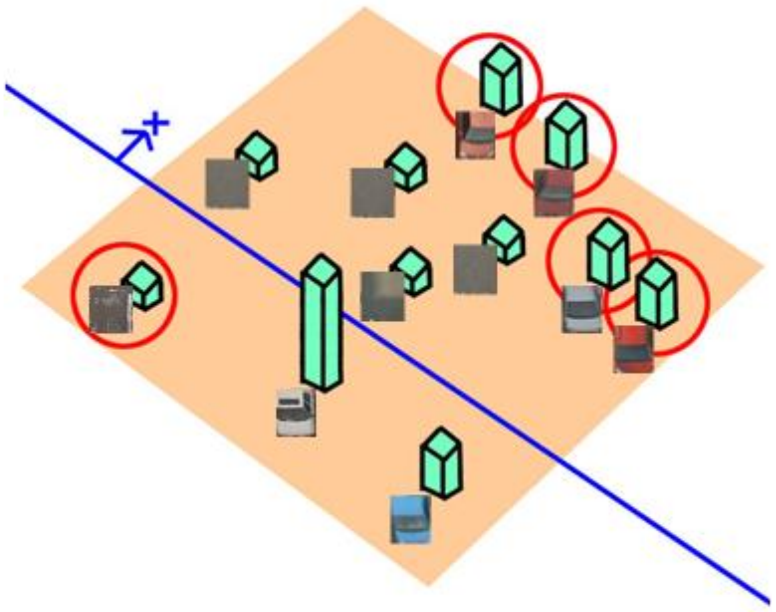
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

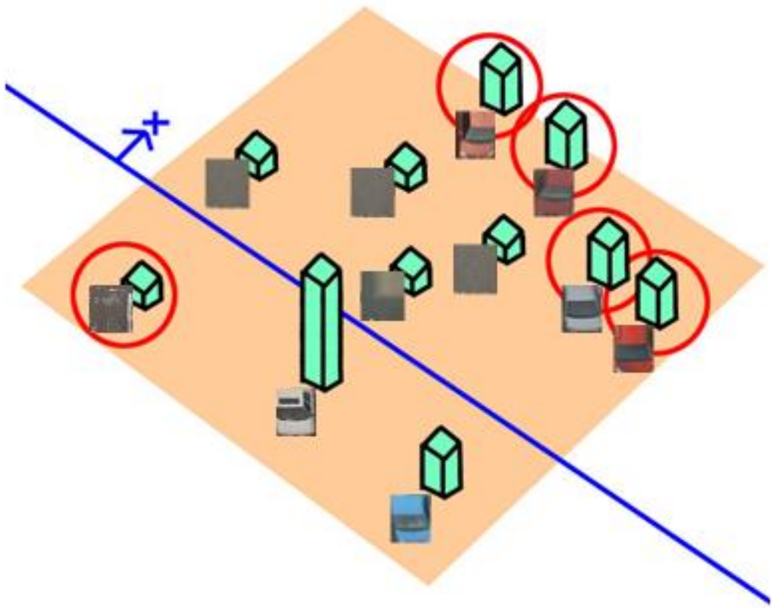
Feature Selection

- AdaBoost starts with a uniform distribution of "weights" over training examples.
- **Select the classifier with the lowest weighted error (i.e. a "weak" classifier)**
- **Increase the weights on the training examples that were misclassified.**
- (Repeat)



Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

Feature Selection



- At the end, carefully make a linear combination of the weak classifiers obtained at all iterations.

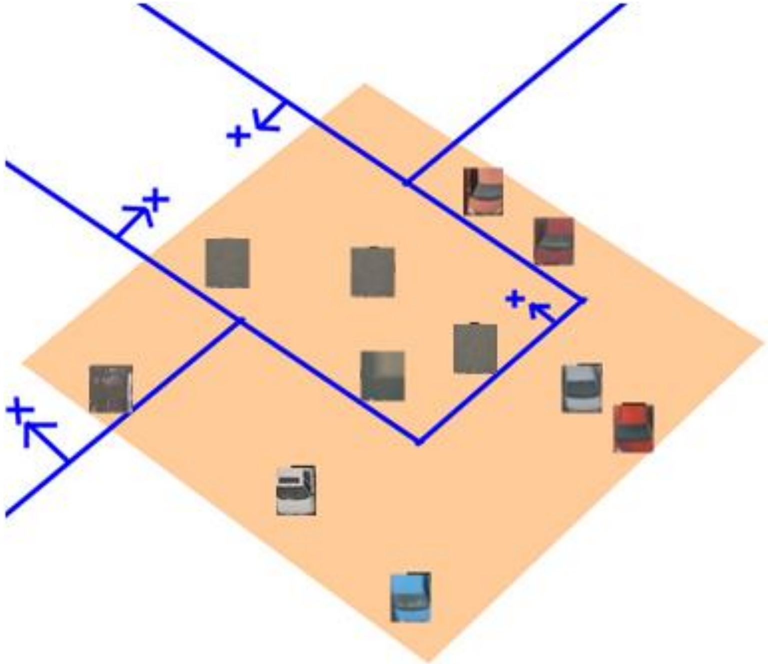
Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Feature Selection



- At the end, carefully make a linear combination of the weak classifiers obtained at all iterations.

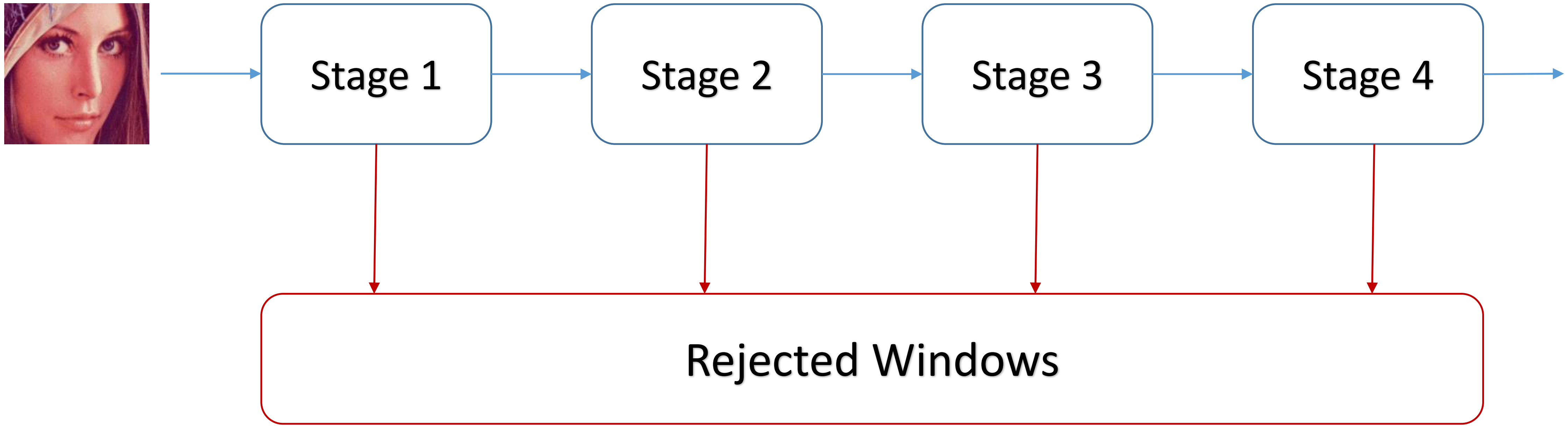
Slide taken from a presentation by Qing Chen, Discover Lab, University of Ottawa

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

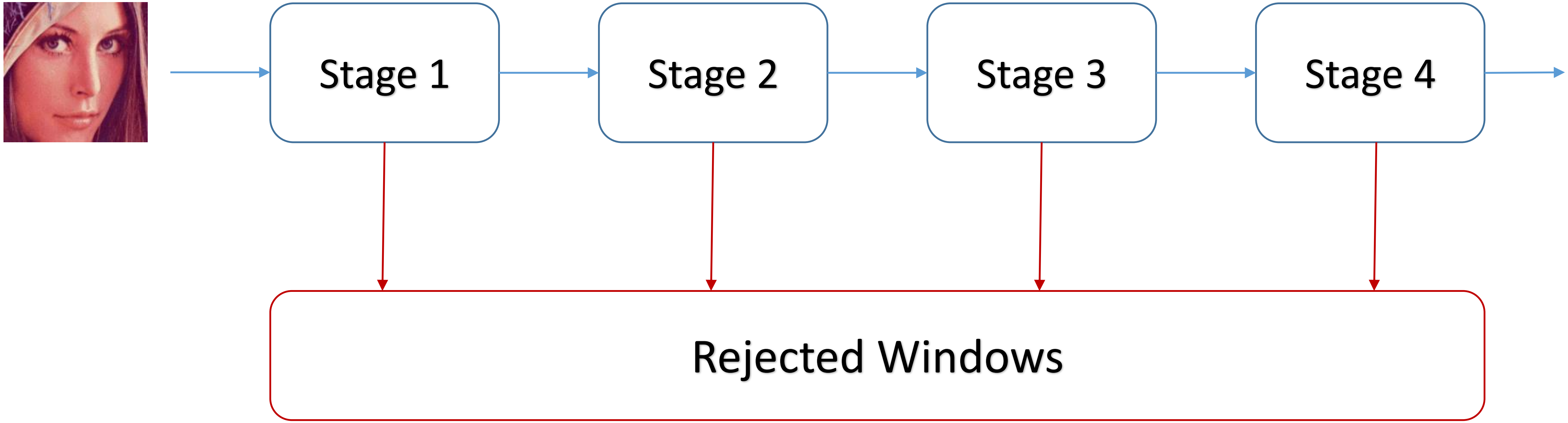
Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



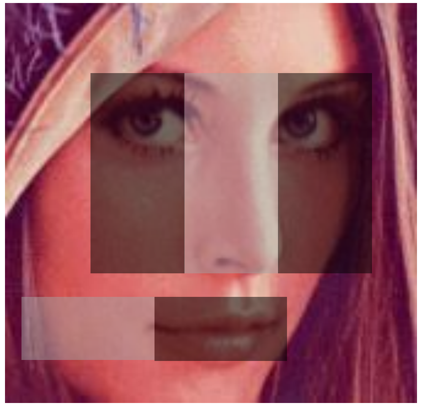
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Stage 1 → Stage 2 → Stage 3 → Stage 4 →

Rejected Windows

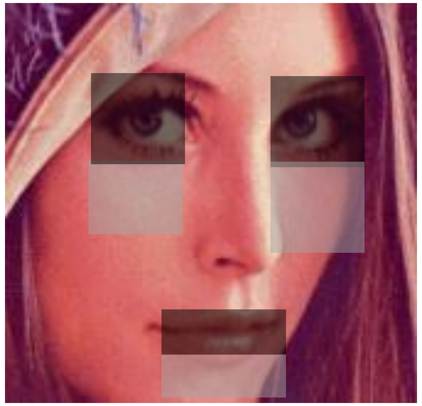
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Stage 1 → Stage 2 → Stage 3 → Stage 4 →

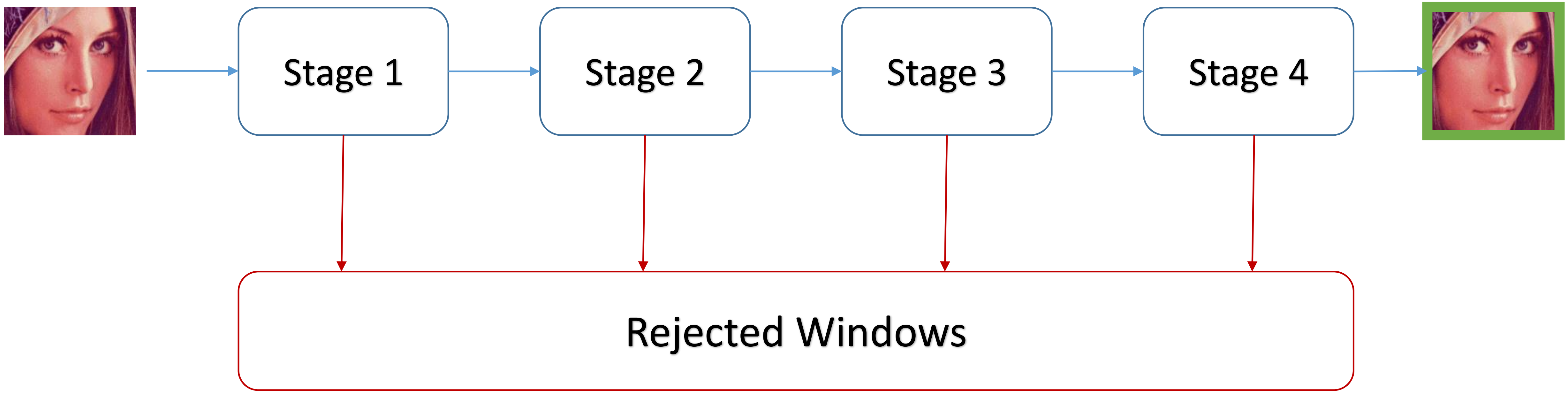
Rejected Windows

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

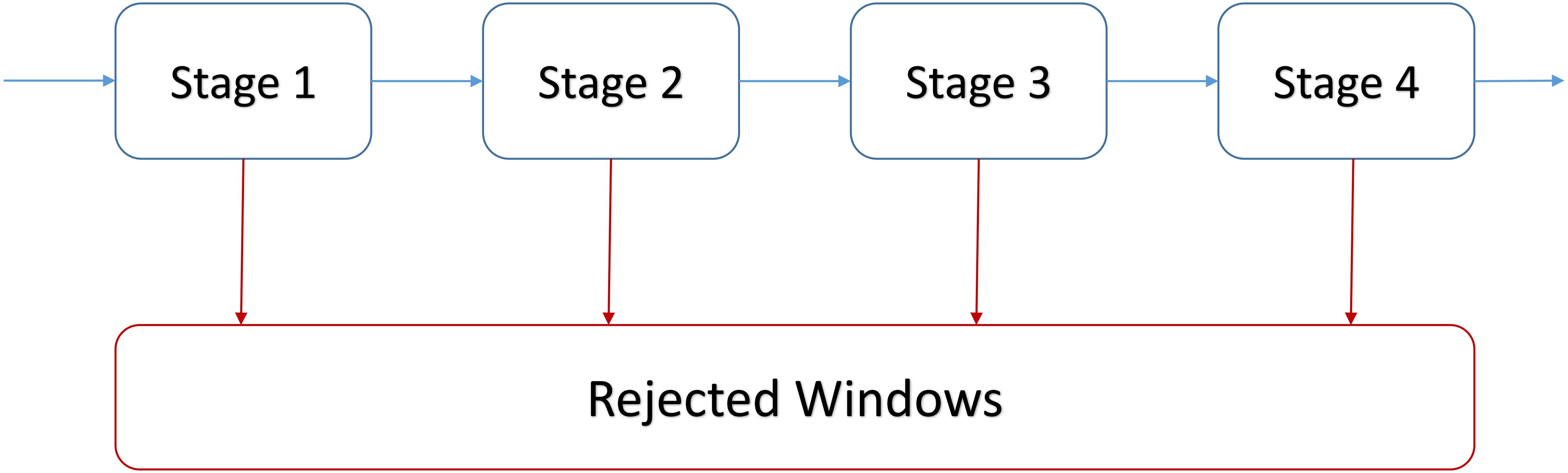
Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



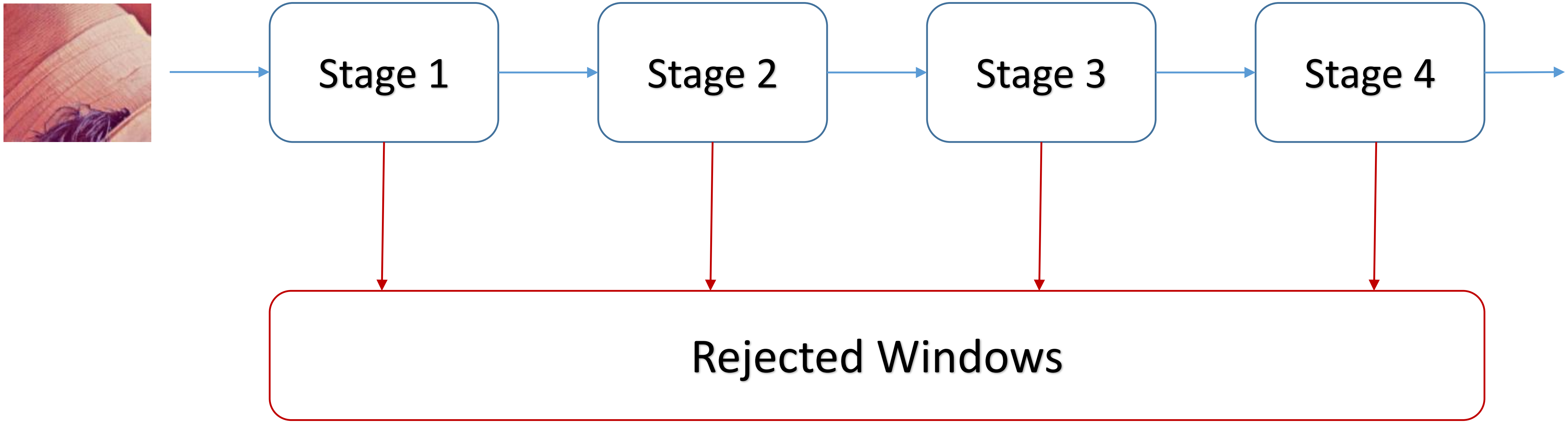
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

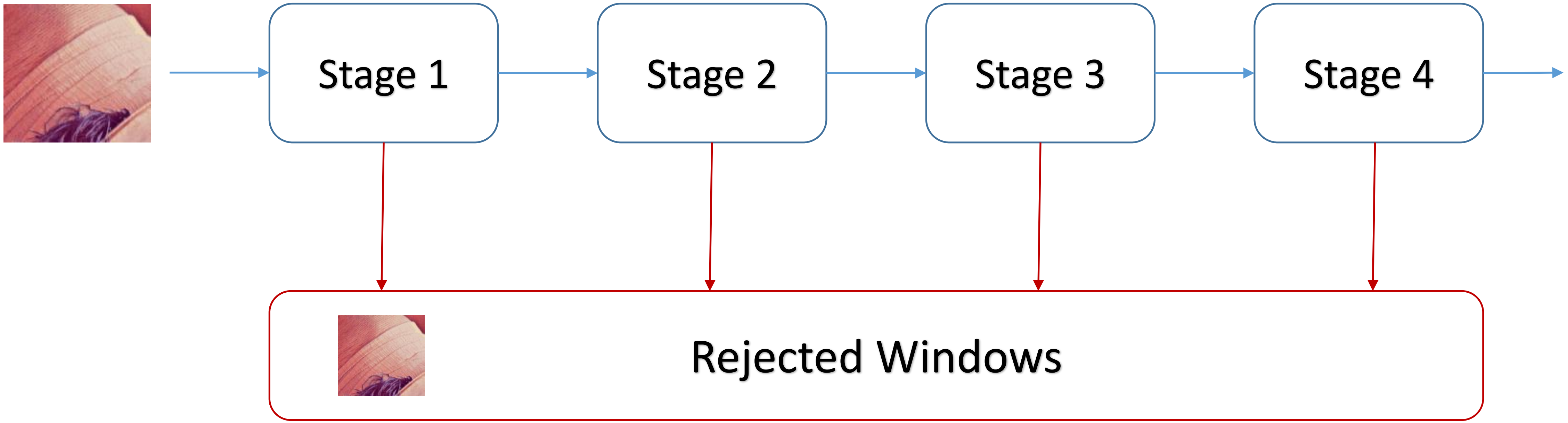
Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Cascade of Classifier

The idea of cascade classifier is reject the non-face region as soon as possible



Haar Features

OpenCV Face Detection: Visualized [online]. [cit. 2020-01-25]. Dostupné z: https://vimeo.com/12774628

Face Detection - Evaluation

Face Detection Data Set and Benchmark Home [online]. [cit. 2020-01-25]. Dostupné z: http://vis-www.cs.umass.edu/fddb/

Face Detection - Evaluation

TP = number of true positives
FP = number of false positives
FN = number of false negatives
TN = number of true negatives

precision = TP/(TP+FP)
sensitivity = TP/(TP+FN)
F1 score (harmonic mean of precision and sensitivity) = 2 × precision × sensitivity/(precision + sensitivity)

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Face Detection - Evaluation

Figure 8. *Matching detections and annotations*. In this image, the ellipses specify the face annotations and the five rectangles denote a face detector's output. Note that the second face from left has two detections overlapping with it. We require a valid matching to accept only one of these detections as the true match, and to consider the other detection as a false positive. Also, note that the third face from the left has no detection overlapping with it, so no detection should be matched with this face. The blue rectangles denote the true positives and yellow rectangles denote the false positives in the desired matching.

Face Detection Data Set and Benchmark Home [online]. [cit. 2020-01-25]. Dostupné z: http://vis-www.cs.umass.edu/fddb/

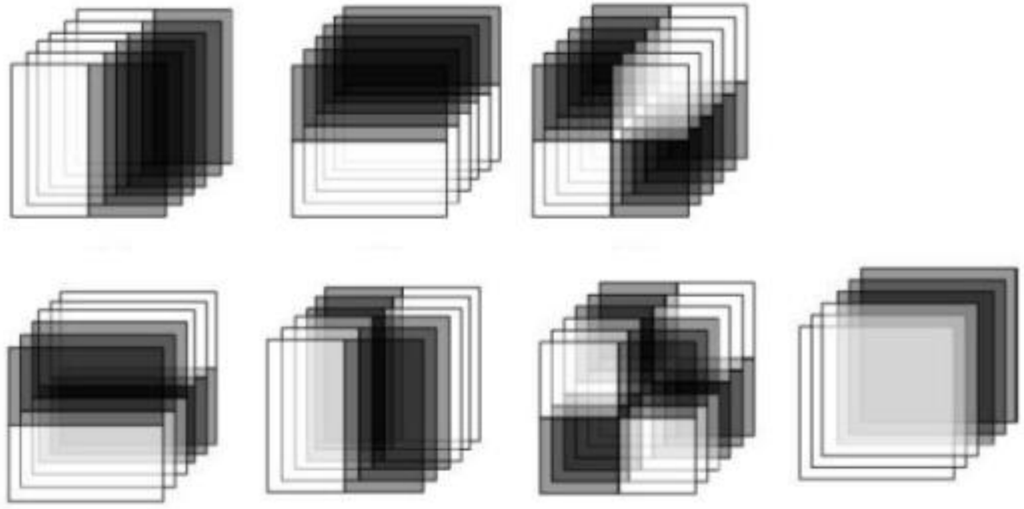
Haar Features

- Since Viola and Jones popularized the Haar-like features for face detection, the Haarlike features and their modifcations were used in many detection tasks (e.g. pedestrian, eye, vehicle).

- In the area of pedestrian detection, in [1], the authors presented the component-based person detector that is able to detect the occluded people in clustered scenes in static images. The detector uses the Haar-like features to describe the components of people (heads, legs, arms) combined with the SVM classifer. The Viola and Jones detection framework was successfully extended for moving-human detection in [2]. In [3], the authors proposed the method for estimating the walking direction of pedestrian.

Haar Features

- The 3D Haar-like features for pedestrian detection were presented in [4]. The authors extend the classical Haar-like features using the volume flters in 3D space (instead of using rectangle flters in 2D space) to capture motion information. The 3D features are then combined with the SVM classifer. To compute the 3D Haar-like features using the integral image like the classical 2D features, the authors introduced Integral Volume that extends 2D integral image to the three dimensions.



Radovan Fusek: Descriptors for Object Detection in Image Recognition, Philosophiæ Doctor Thesis, 2016

Haar Features

[1] Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Trans. Pattern Anal. Mach. Intell. 23(4), 349–361 (Apr 2001), http://dx.doi.org/10.1109/34.917571

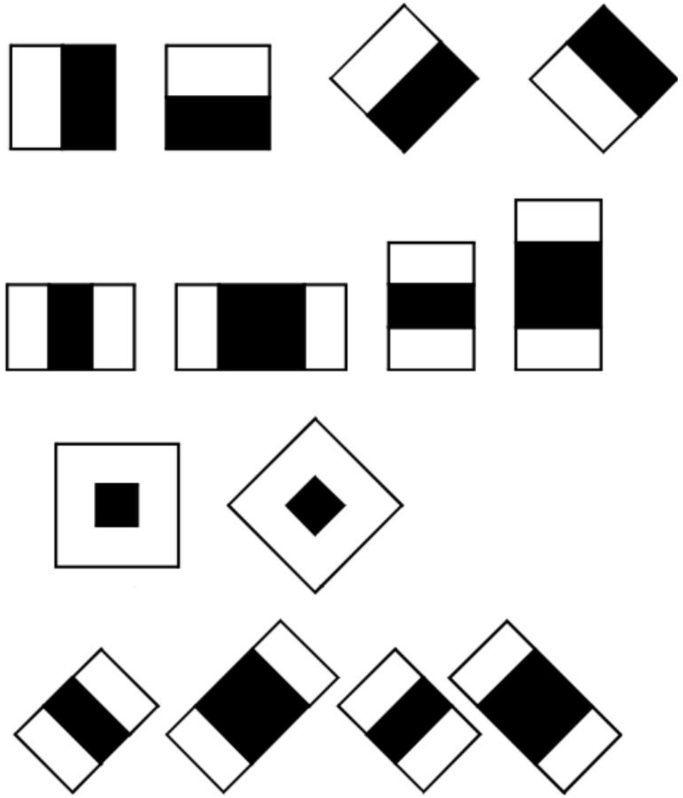
[2] Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. pp. 734 –741 vol.2 (oct 2003)

[3] Shimizu, H., Poggio, T.: Direction estimation of pedestrian from multiple still images. In: Intelligent Vehicles Symposium, 2004 IEEE. pp. 596–600 (2004)

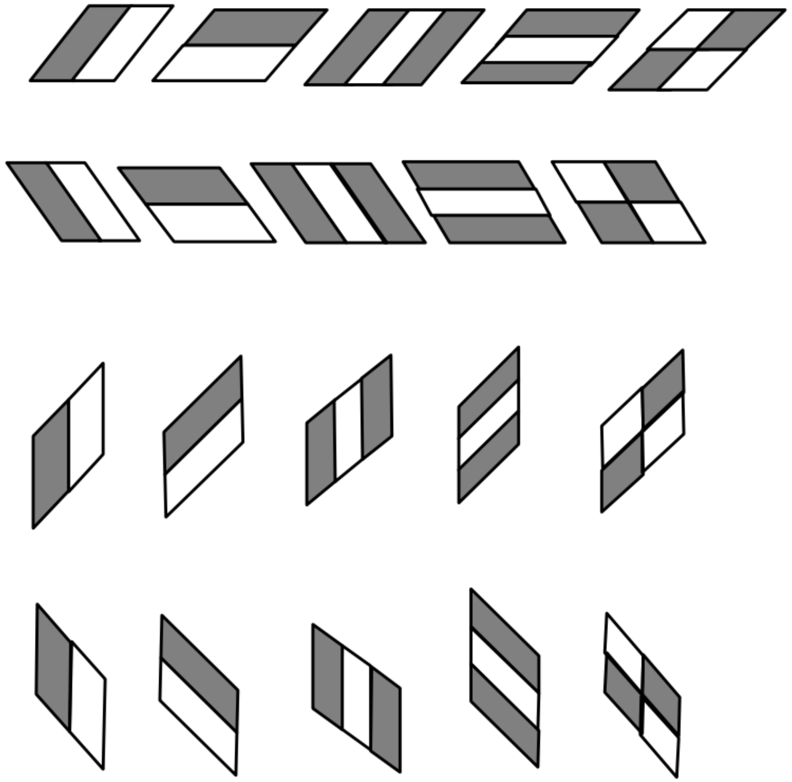
[4] Cui, X., Liu, Y., Shan, S., Chen, X., Gao, W.: 3d haar-like features for pedestrian detection. In: Multimedia and Expo, 2007 IEEE International Conference on. pp. 1263–1266 (July 2007)

Haar Features

The modified version of Haar-like features that more properly reflect the shape of the pedestrians than the classical Haar-like features.



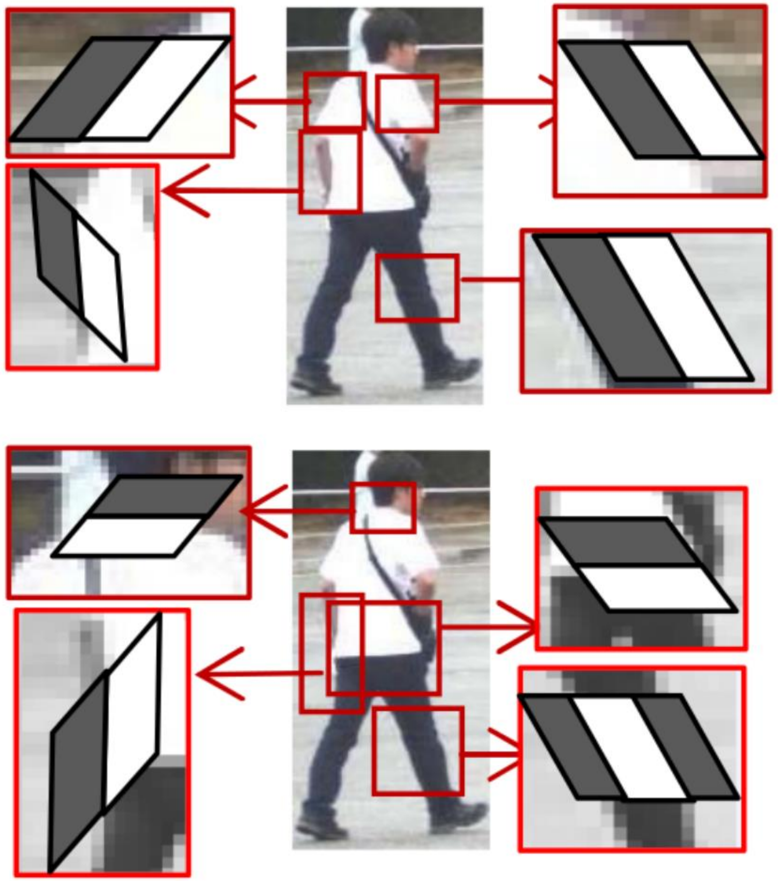
Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Image Processing. 2002. Proceedings. 2002 International Conference on. vol. 1, pp. I–900–I–903 vol.1 (2002)



Hoang, V.D., Vavilin, A., Jo, K.H.: Pedestrian detection approach based on modified haar-like features and adaboost. In: Control, Automation and Systems (ICCAS), 2012 12th International Conference on. pp. 614-618 (Oct 2012)

Haar Features

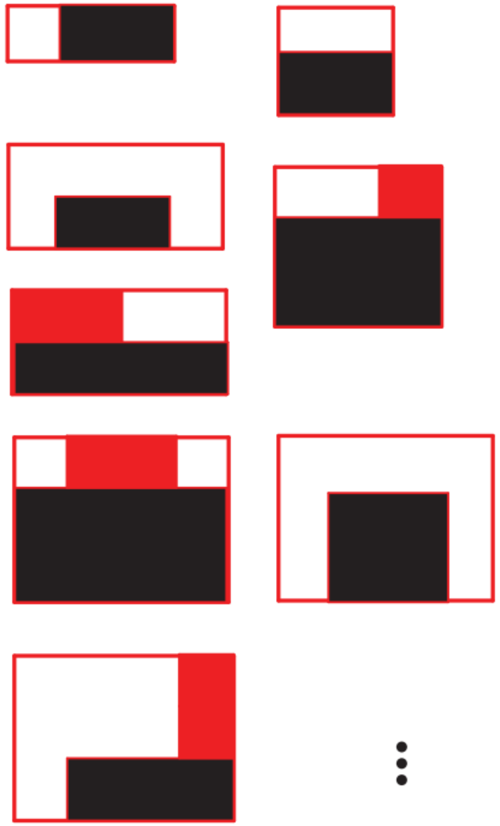
The modified version of Haar-like features that more properly reflect the shape of the pedestrians than the classical Haar-like features.

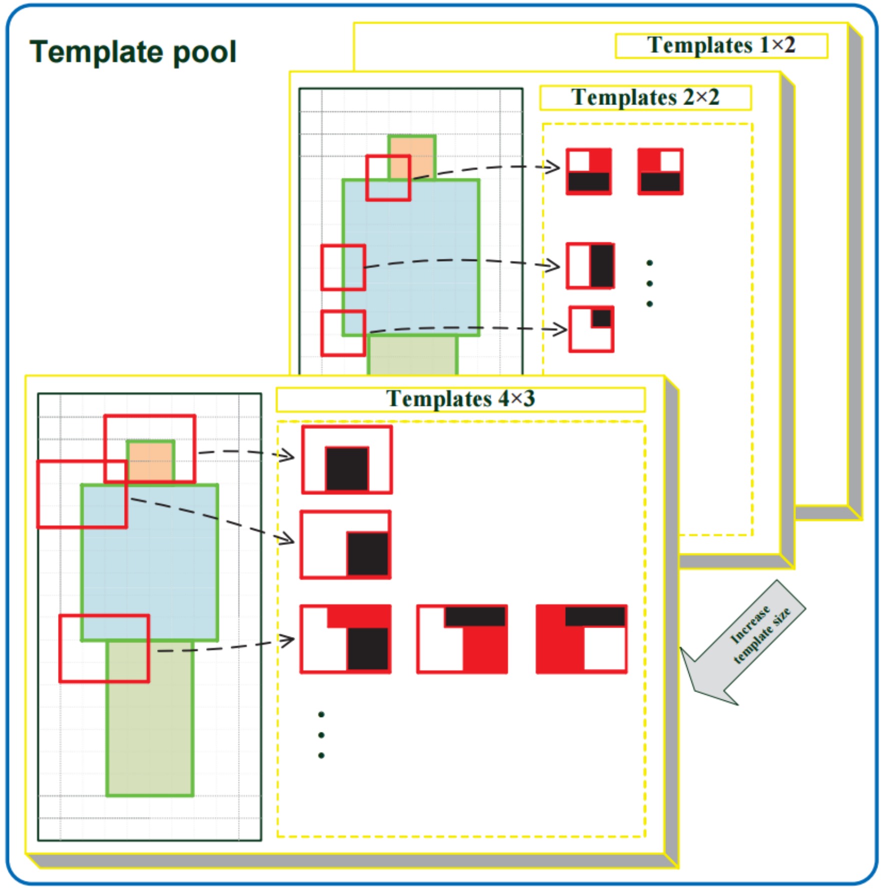


Hoang, V.D., Vavilin, A., Jo, K.H.: Pedestrian detection approach based on modified haar-like features and adaboost. In: Control, Automation and Systems (ICCAS), 2012 12th International Conference on. pp. 614-618 (Oct 2012)

Haar Features

The modified version of Haar-like features that more properly reflect the shape of the pedestrians than the classical Haar-like features.

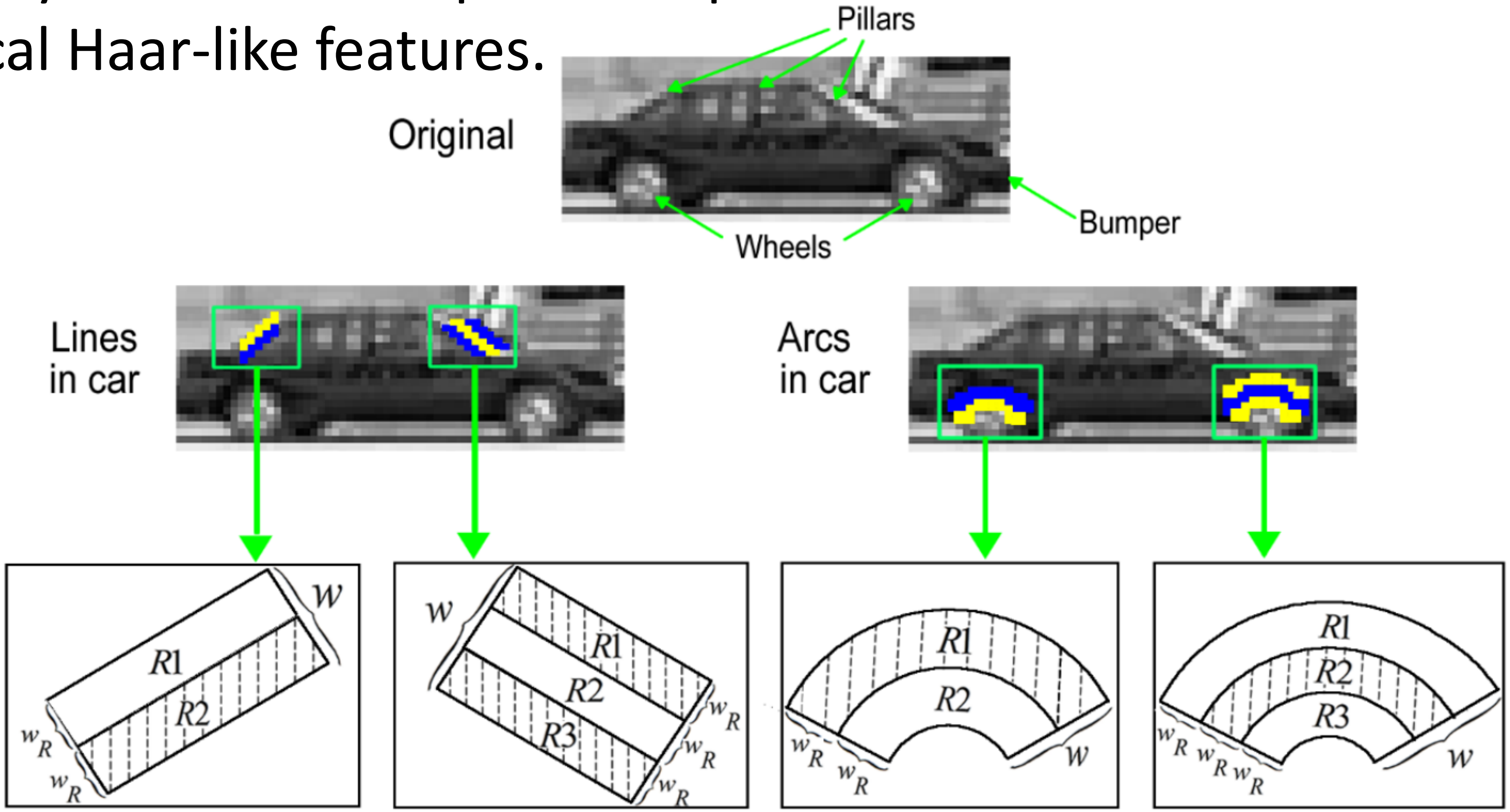




S. Zhang, C. Bauckhage, and A. B. Cremers. Informed haar-like features improve pedestrian detection. In CVPR, 2014.

Haar Features

The modified version of Haar-like features that more properly reflect the shape of the pedestrians than the classical Haar-like features.



Zheng, W., Liang, L.: Fast car detection using image strip features. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 2703–2710 (2009)

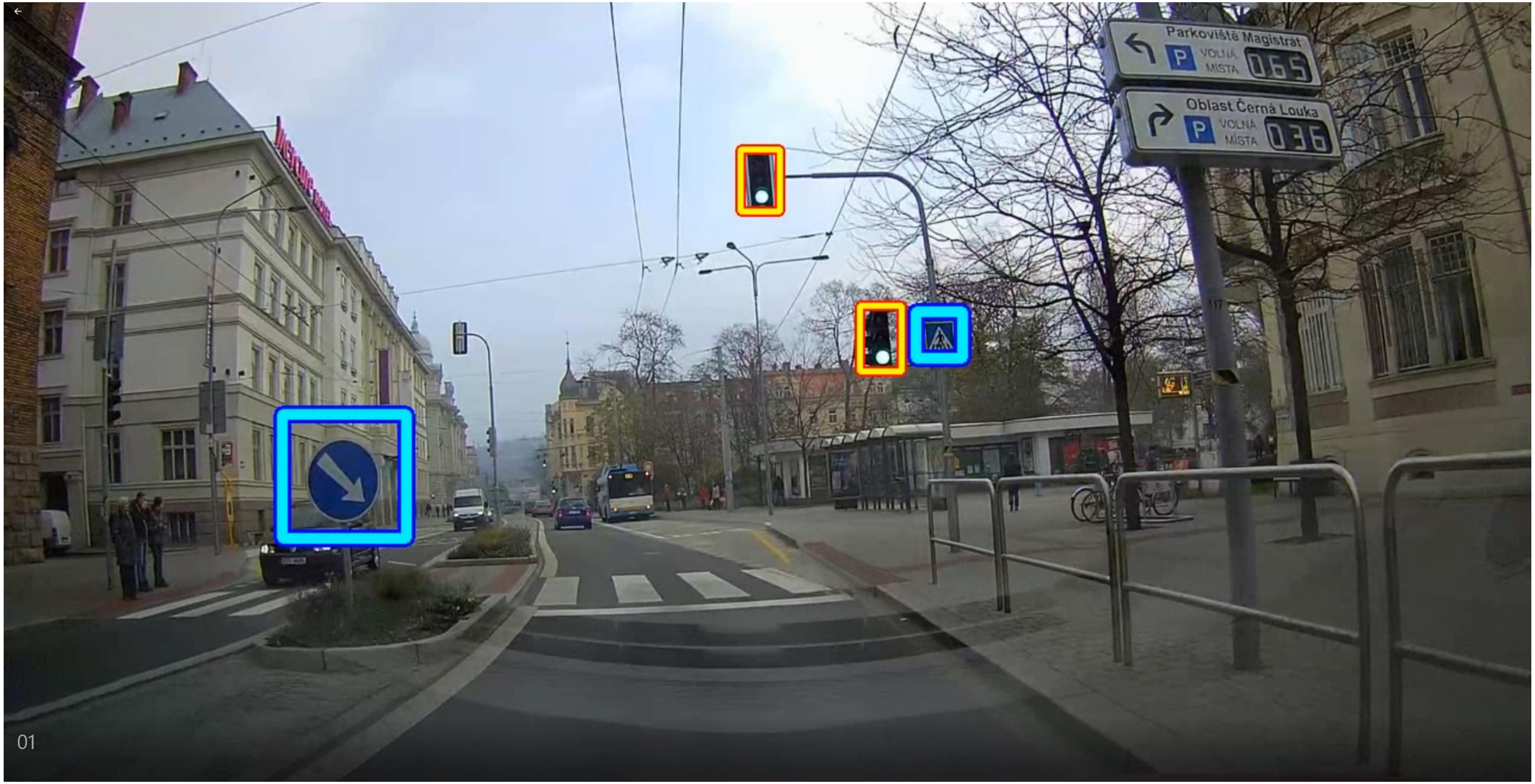
Haar Features

Five most significant Haar features selected for the first stage of each of the 3 trained detectors

https://ieeexplore.ieee.org/abstract/document/8050341

A. Suleiman, Y. Chen, J. Emer and V. Sze, "Towards closing the energy gap between HOG and CNN features for embedded vision," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017, pp. 1-4.
doi: 10.1109/ISCAS.2017.8050341

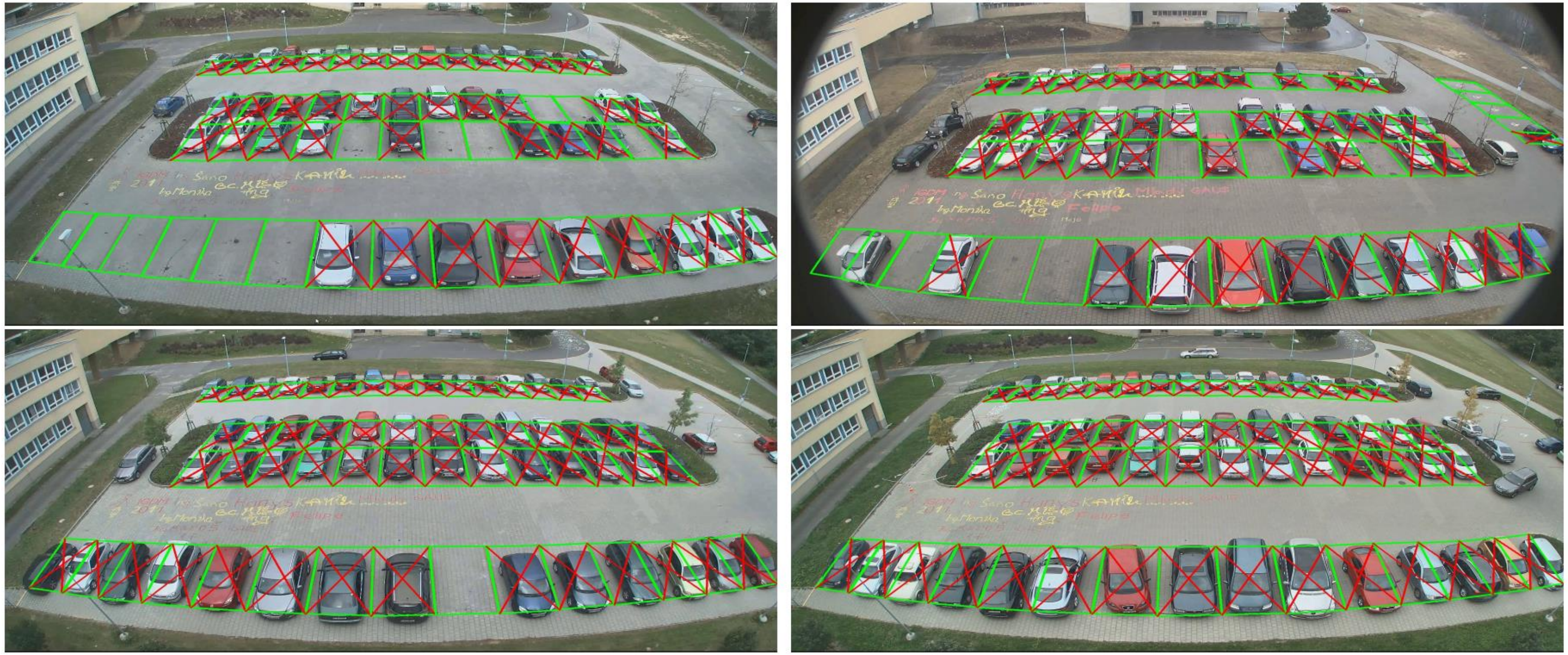
Haar Features



Haar Features

- Fusek, R., Mozdřeň, K., Šurkala, M., Sojka, E.: **AdaBoost for Parking Lot Occupation Detection**. Advances in Intelligent Systems and Computing, vol. 226, pp. 681-690 (2013)

http://mrl.cs.vsb.cz/



Parking Lot Occupation

- Haar
- HOG
- LBP

Traditional Approaches

- SIFT, SURF

KeyPoints

- CNNs

Deep Learning Approach

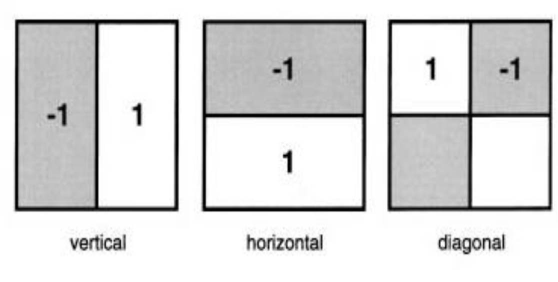
Related Works

2000

Papageorgiou (2000)

A Trainable System for Object Detection

CONSTANTINE PAPAGEORGIOU AND TOMASO POGGIO
Center for Biological and Computational Learning, Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
cpapa@ai.mit.edu
tp@ai.mit.edu

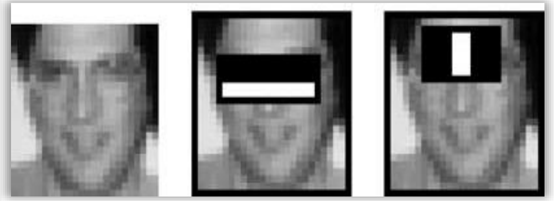


Viola, Jones (2001,2004)

Robust Real-Time Face Detection

PAUL VIOLA
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
viola@microsoft.com

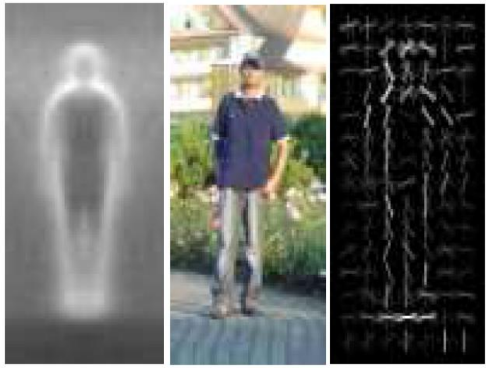
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs (2005) cit. 10947

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr



2005

Object Detection (Analysis)
HOG

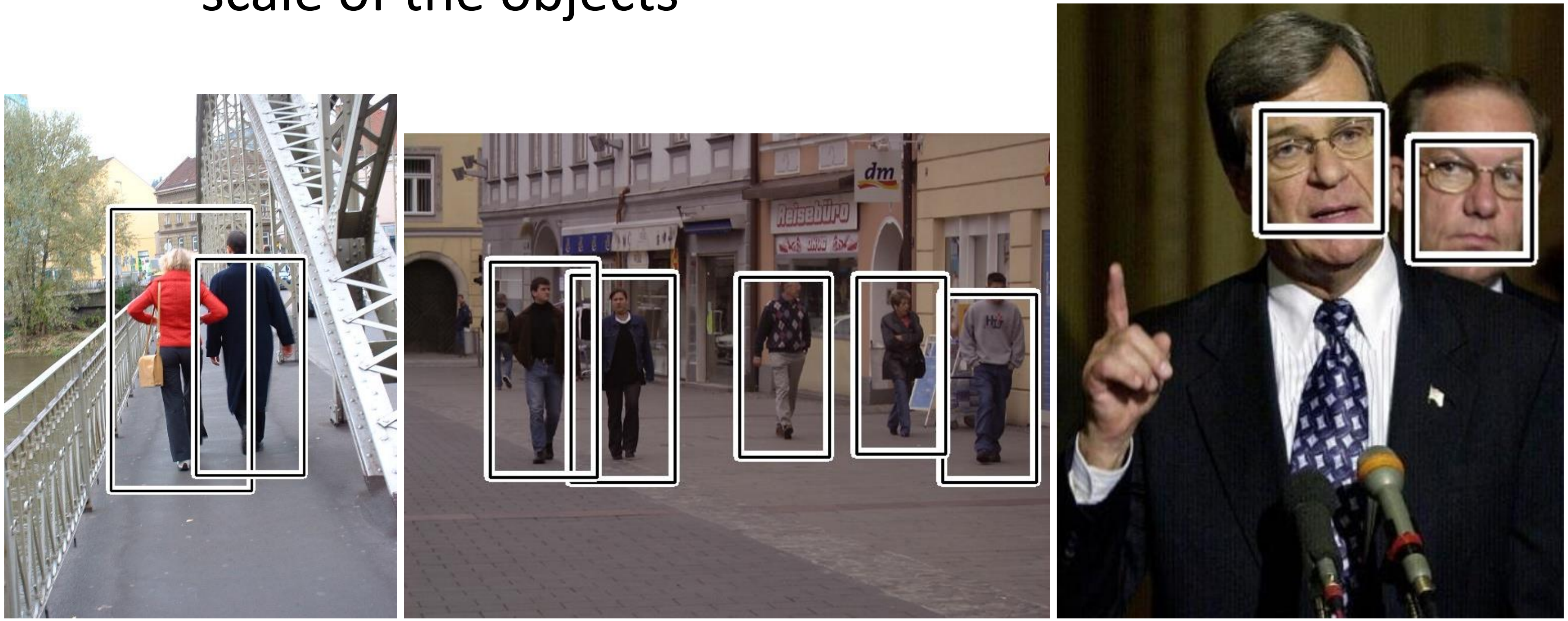
What is Object Detection?

- It is clear that the images contain many objects of interest. The goal of the object detection
  systems is to find the location of these objects in the images (e.g. cars, faces, pedestrians).

- For example, the vehicle detection systems are crucial for traffic analysis or intelligent scheduling, the people detection systems can be useful for automotive safety, and the face detection systems are a key part of face recognition systems.

What is Object Detection?

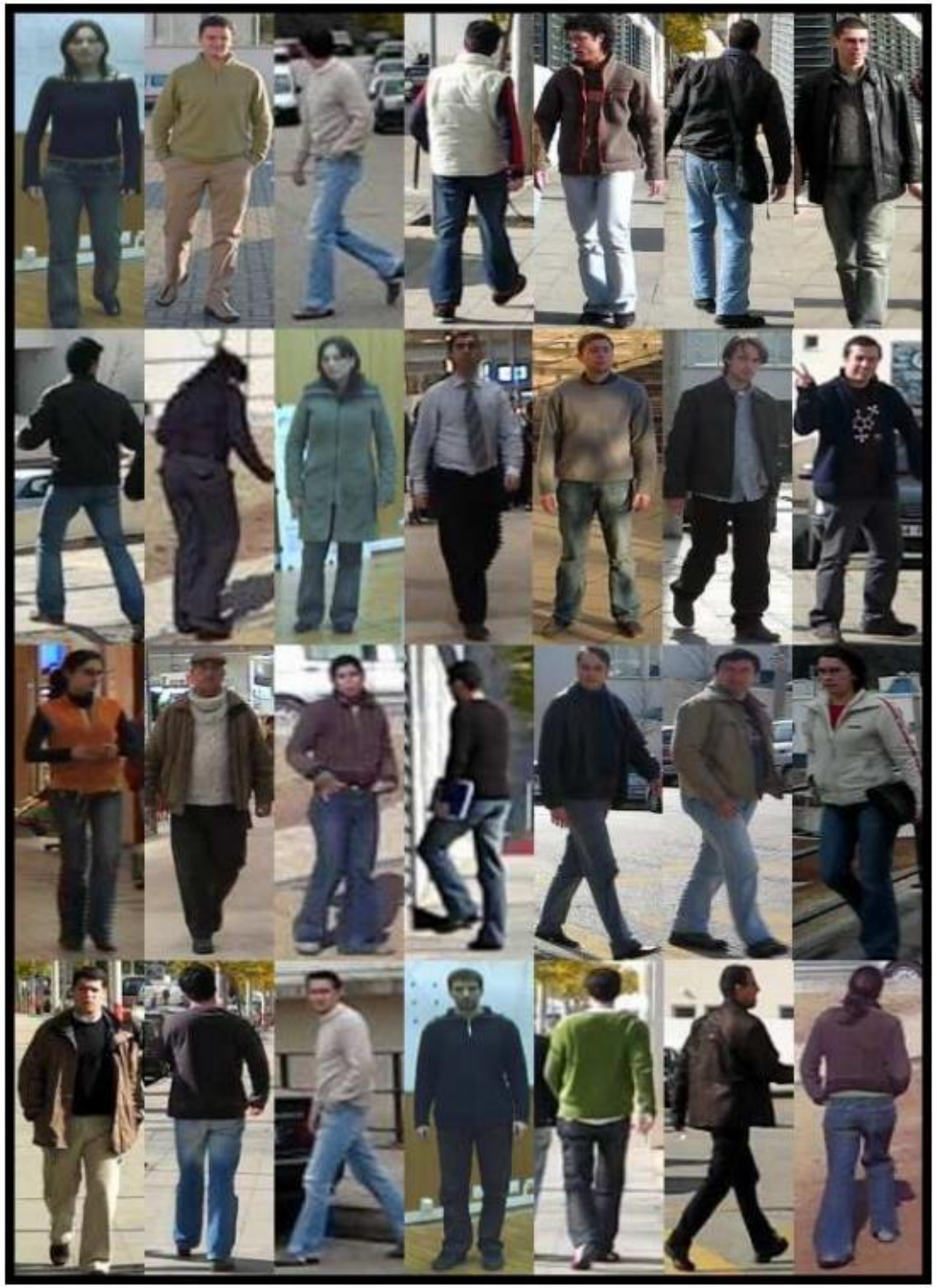
- Output?
  - position of the objects
  - scale of the objects



EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Pedestrian Detection - Challenges?



EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

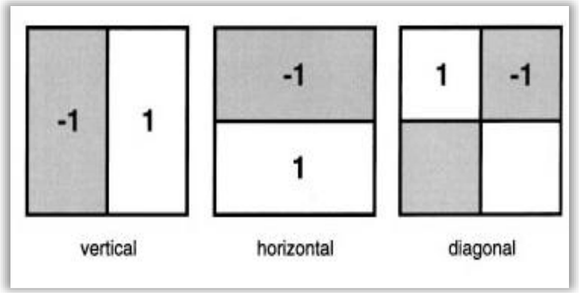
Related Works

2000

Papageorgiou (2000)

A Trainable System for Object Detection

CONSTANTINE PAPAGEORGIOU AND TOMASO POGGIO
Center for Biological and Computational Learning, Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
cpapa@ai.mit.edu
tp@ai.mit.edu

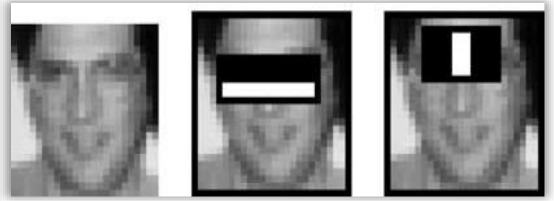


Viola, Jones (2001,2004)

Robust Real-Time Face Detection

PAUL VIOLA
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
viola@microsoft.com

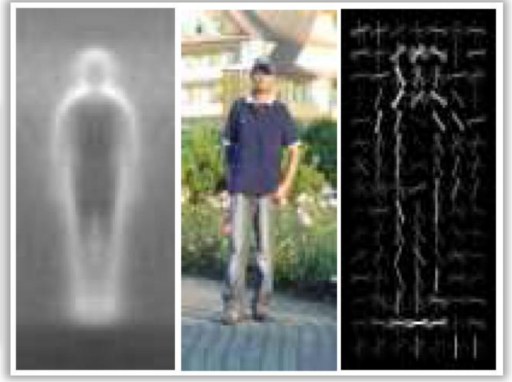
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs (2005)

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr



2005

Histograms of Oriented Gradients (HOG)

- In recent years, the object detectors that are based on edge analysis that provides valuable information about the objects of interest were used in many detection tasks. In this area, the histograms of oriented gradients (HOG) [1] are considered as the state-of-theart method.

- In HOG, a sliding window is used for detection. The window is divided into small connected cells in the process of obtaining HOG descriptors. The histograms of gradient orientations are calculated in each cell. It is desirable to normalize the histograms across a large block of image. As a result, a vector of values is computed for each position of window. This vector is then used for recognition, e.g. by the Support Vector Machine classifier.

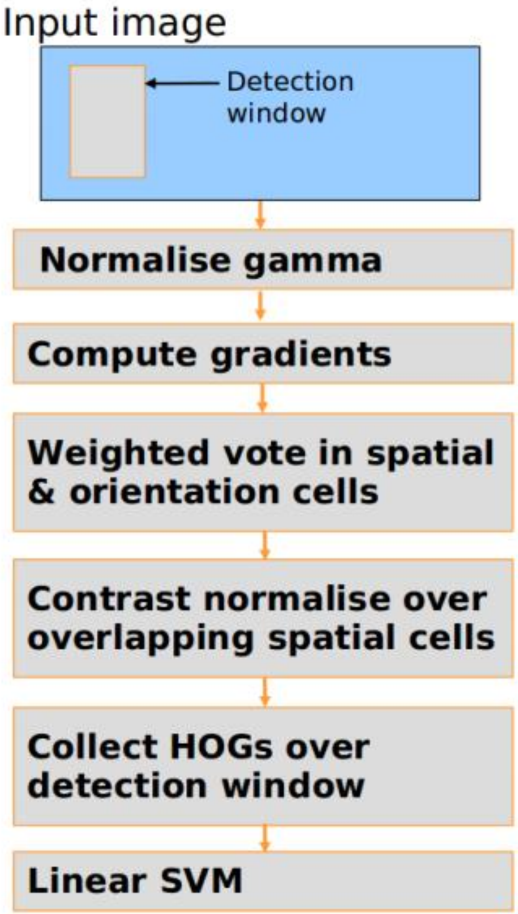
Histograms of Oriented Gradients (HOG)

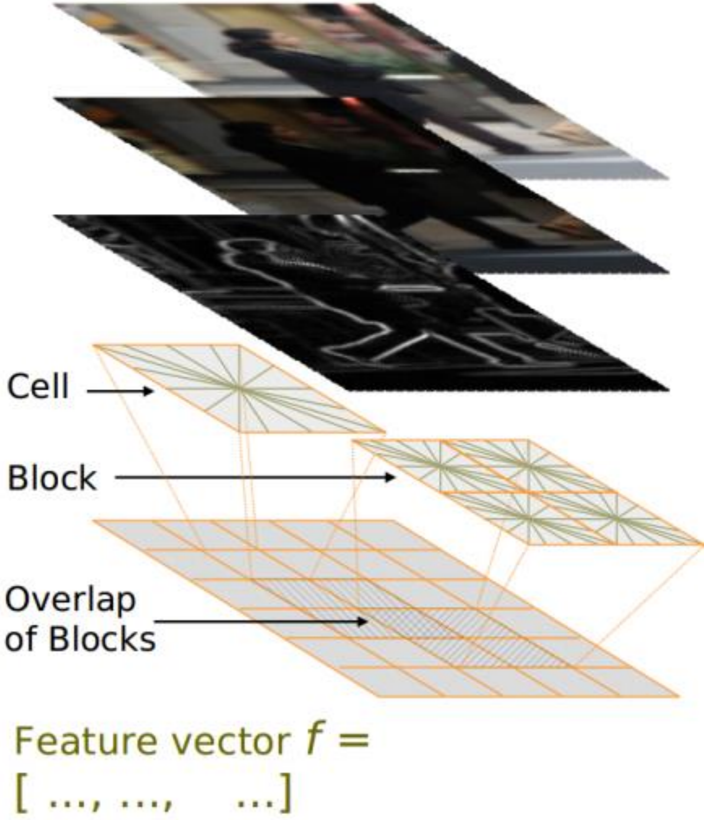
- Dalal and Trigs experimented with the size of detection window and they suggested the rectangular window with the size 64 × 128 pixels. They also tried to reduce the size of the window to 48 × 112 pixels. Nevertheless, they obtained the best detection result with the size 64 × 128 pixels.

Histograms of Oriented Gradients (HOG)

Basic Steps:

- In HOG, a sliding window is used for detection.
- The window is divided into small connected cells.
- The histograms of gradient orientations are calculated in each cell.
- Support Vector Machine (SVM) classifier.





Navneet Dalal, Bill Triggs : Object Detection using Histograms of Oriented Gradients [online]. [cit. 2020-01-25]. Dostupné z: http://host.robots.ox.ac.uk/pascal/VOC/voc2006/slides/dalal.ppt

- For gradient computation, the image without Gaussian smoothing is filtered with the [1, 0, -1] kernel to compute the horizontal and vertical derivatives.
- Then the derivatives are used to compute the magnitude of the gradient and orientation .

$$D_X = [-1 \quad 0 \quad 1] \text{ and } D_Y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \qquad I_X = I * D_X \text{ and } I_Y = I * D_Y$$

magnitude of the gradient is $|G| = \sqrt{I_X^2 + I_Y^2}$

orientation of the gradient is given by: $\theta = \arctan \dfrac{I_Y}{I_X}$

- In the next step, the image is divided into the cells and the cell histograms are constructed. The histogram bins are spread over 0 to 180 degrees or 0 to 360 degrees. The corresponding histogram bin is found for each pixel inside the cell. Each pixel contributes a weighted vote for its corresponding bin. The pixel contribution can be the gradient magnitude.

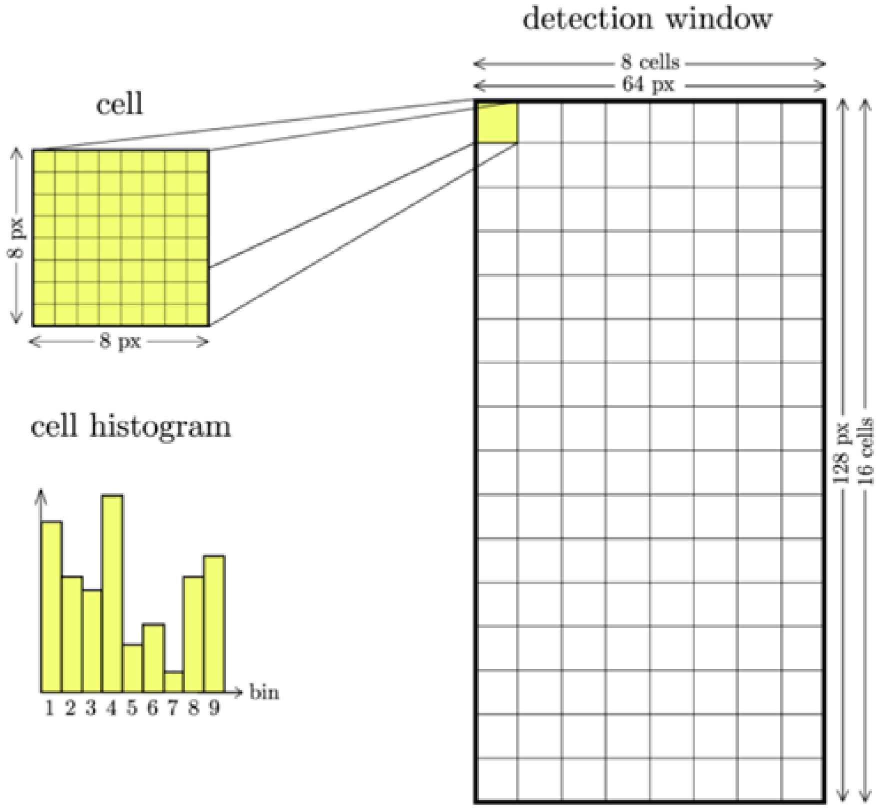
- Next step represents contrast normalization. For this purpose, the cells are grouped into the large blocks (i.e. 2×2 cells are considered as blocks). The histograms are normalized within the blocks (e.g. using L2-norm). In the paper, the two main block geometries are presented; rectangular and circular.

- The final HOG descriptor is represented by histogram vectors of all blocks within the detection window

Radovan Fusek: Descriptors for Object Detection in Image Recognition, Philosophiæ Doctor Thesis, 2016

Dalal and Trigs experimented with the size of detection window and they suggested the rectangular window with the size 64 × 128 pixels.

They also tried to reduce the size of the window to 48 × 112 pixels. Nevertheless, they obtained the best detection result with the size 64 × 128 pixels.



Blocks, Cells:

- 8 x 8 cell
- 16 x 16 block – overlap
- normalization within the blocks

- **<u>Final Vector</u>: Collect HOG blocks into vector**

Bauer,S., Brunsmann, U., FPGA Implementation of a HOG-based Pedestrian Recognition System, Stefan Schlotterbeck-Macht Faculty of Engineering Aschaffenburg University of Applied Sciences, Aschaffenburg, Germany

- The classical HOG descriptors suffer from the large number of features, which causes that the training and detection phases can be time consuming. The sufficient amount of training data is also needed to find a separating hyperplane by the SVM classifier.

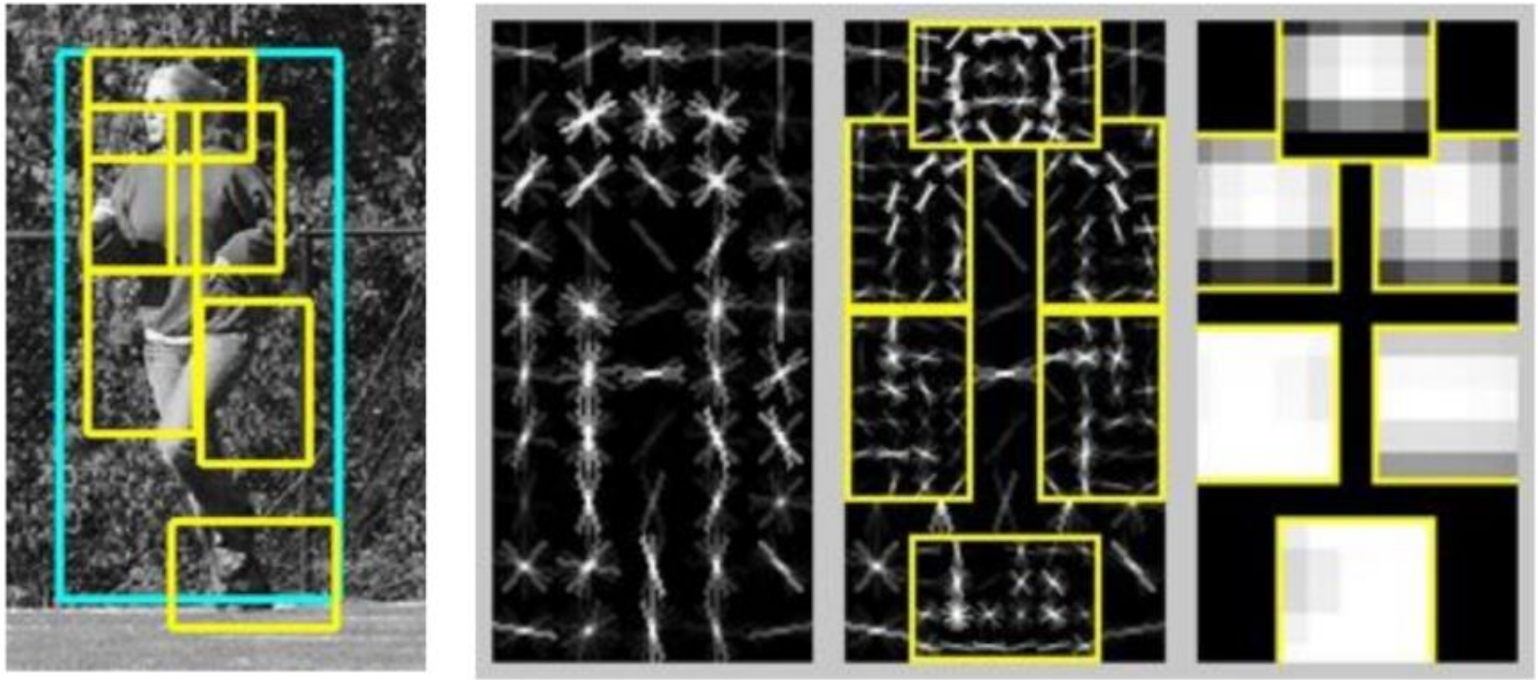
- Sometimes, it is desirable to use the methods for the dimensionality reduction of feature vector. In addition the that, the classical HOG descriptors are not rotation invariant.

- These shortcomings became the motivation for creating many variations of HOG-based detectors. Many methods and applications based on HOG were presented in recent years.

In [1], the authors applied the principal component analysis (PCA) to the HOG feature vector to obtain the PCA-HOG vector. This vector contains the subset of HOG features and the vector is used as an input for the SVM classifier. Their method was used for pedestrian detection with the satisfactory results.

Felzenszwalb et al. proposed the part-based detector that is based on HOG. In this method, the objects are represented using the mixtures of deformable HOG part models and these models are trained using a discriminative method (see following image). This method obtained excellent performance for object detection tasks [2, 3].

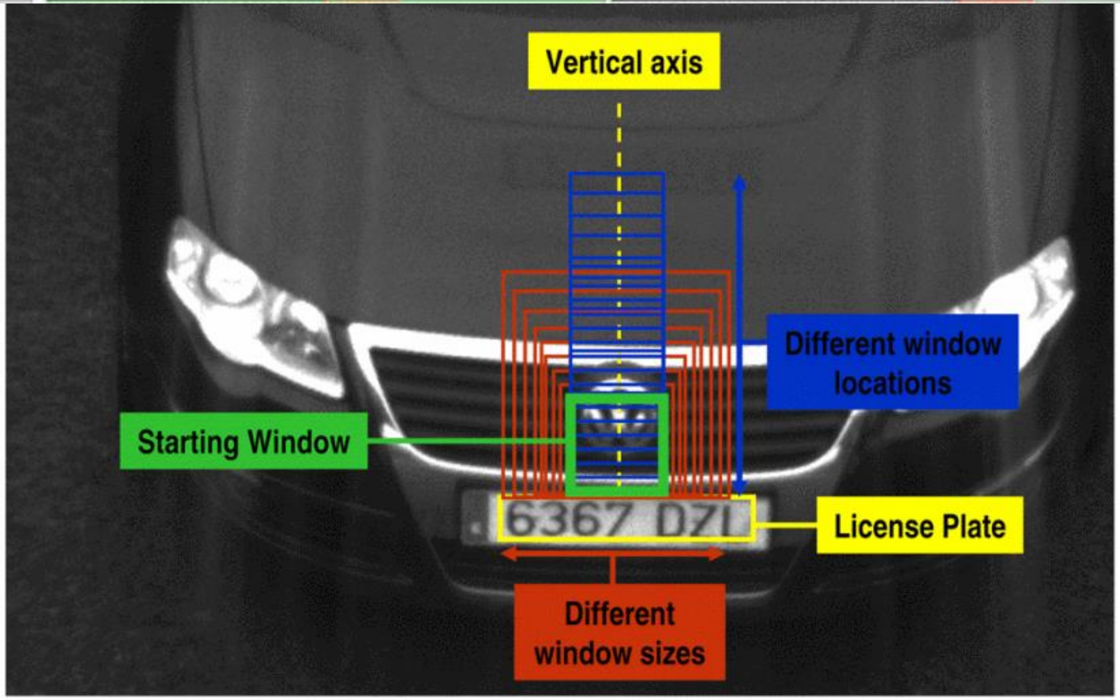
[1] Kobayashi, T., Hidaka, A., Kurita, T.: Neural information processing. chap. Selection of Histograms of Oriented Gradients Features for Pedestrian Detection, pp. 598–607. Springer-Verlag, Berlin, Heidelberg (2008)
[2] Felzenszwalb, P.F., McAllester, D.A., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
[3] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(9), 1627–1645 (2010)



An example of person detection using a part model. The model is defined by the coarse global template that covers the entire object and higher resolution part templates. The templates represent the histogram of oriented gradient [2].

[1] Kobayashi, T., Hidaka, A., Kurita, T.: Neural information processing. chap. Selection of Histograms of Oriented Gradients Features for Pedestrian Detection, pp. 598–607. Springer-Verlag, Berlin, Heidelberg (2008)
[2] Felzenszwalb, P.F., McAllester, D.A., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
[3] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(9), 1627–1645 (2010)

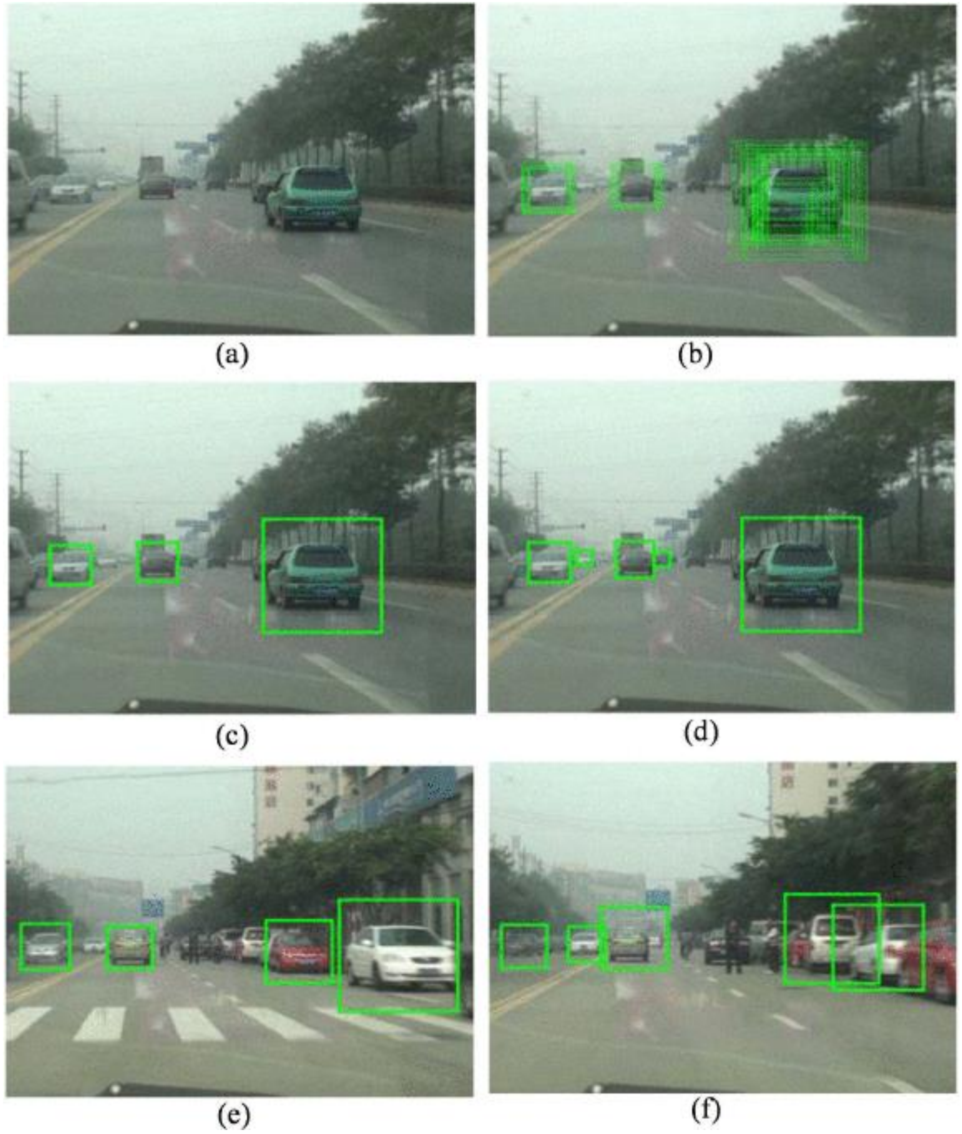
Histograms of Oriented Gradients (HOG

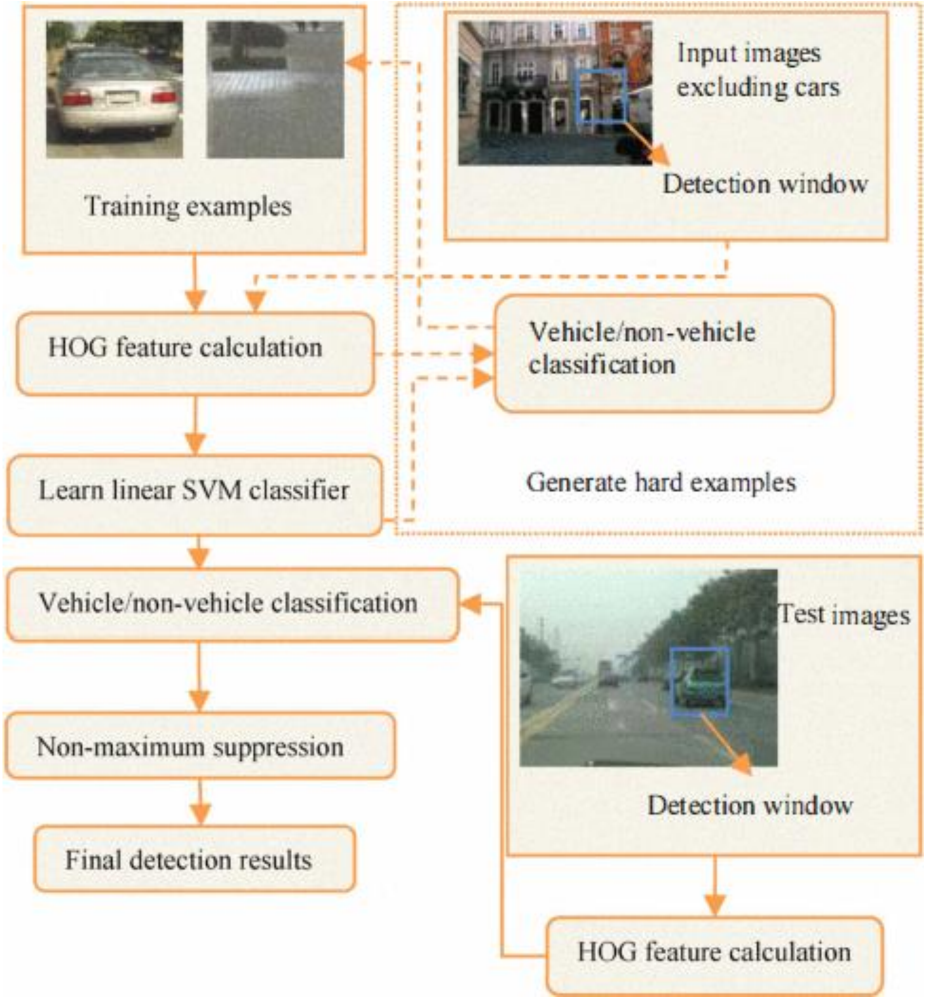


https://ieeexplore.ieee.org/abstract/document/6728559

D. F. Llorca, R. Arroyo and M. A. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, 2013, pp. 2229-2234.
doi: 10.1109/ITSC.2013.6728559

Histograms of Oriented Gradients (HOG) and Automotive



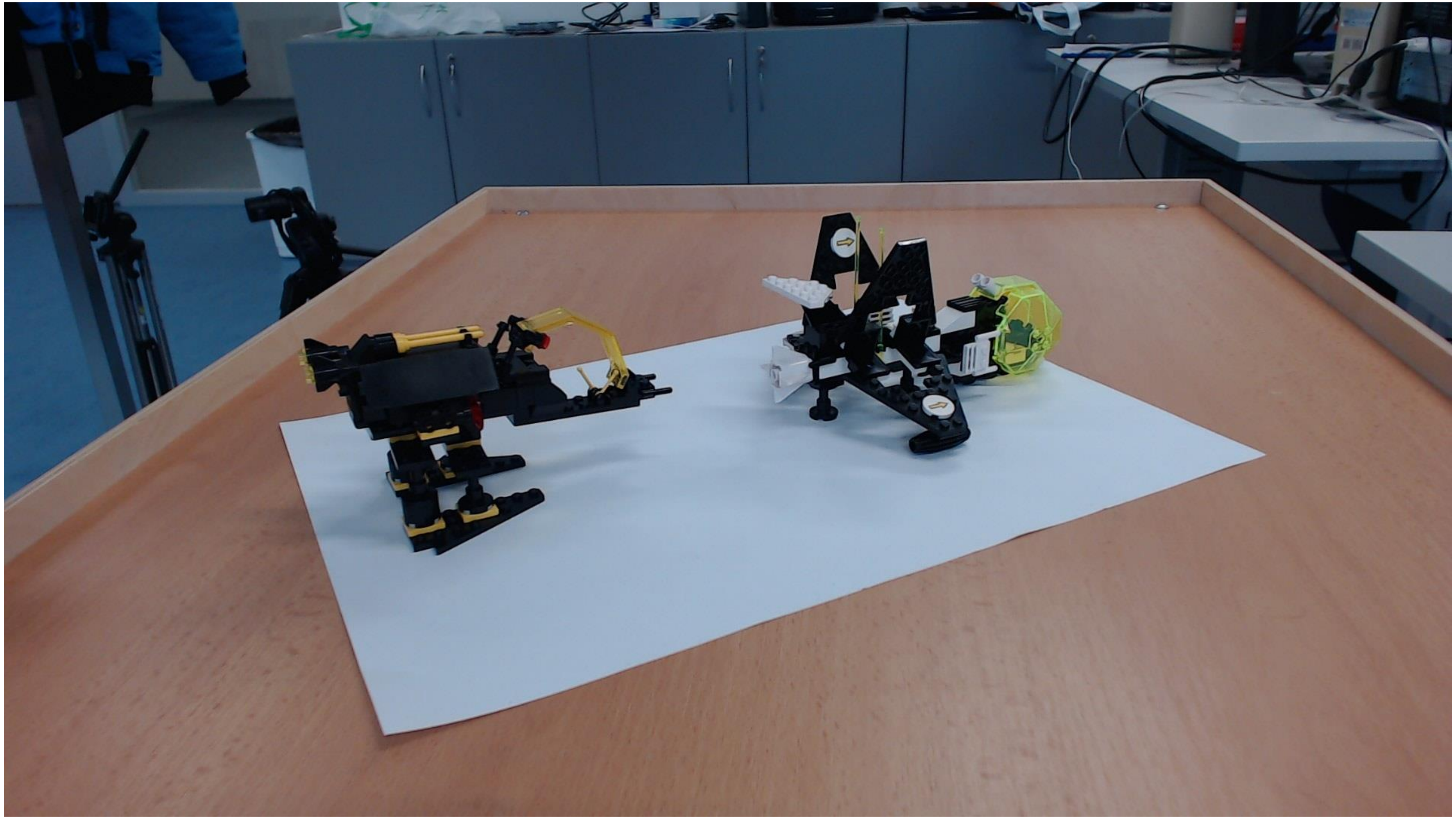


https://ieeexplore.ieee.org/abstract/document/5581983

L. Mao, M. Xie, Y. Huang and Y. Zhang, "Preceding vehicle detection using Histograms of Oriented Gradients," *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, Chengdu, 2010, pp. 354-358. doi: 10.1109/ICCCAS.2010.5581983

Practical Example – Detection + Recognition

Consider the following problem: Find and recognize two following lego kits



OpenCV - http://opencv.org/

OpenCV 3.3.0
Open Source Computer Vision

Main Page | Related Pages | Modules | Namespaces | Classes | Files | Examples

Introduction

OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposite to the C-based OpenCV 1.x API. The latter is described in opencv1x.pdf.

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

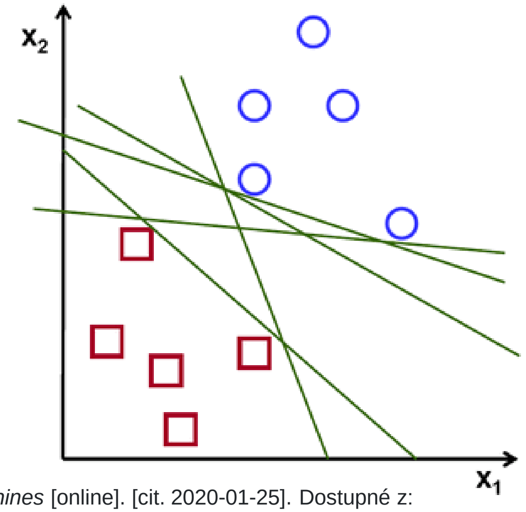
- **Core functionality** - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- **Image processing** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **features2d** - salient feature detectors, descriptors, and descriptor matchers.
- **objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **highgui** - an easy-to-use interface to simple UI capabilities.
- **Video I/O** - an easy-to-use interface to video capturing and video codecs.
- **gpu** - GPU-accelerated algorithms from different OpenCV modules.
- ... some other helper modules, such as FLANN and Google test wrappers, Python bindings, and others.

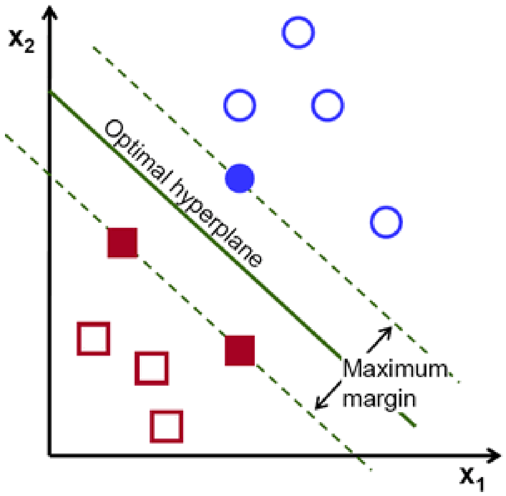
The further chapters of the document describe functionality of each module. But first, make sure to get familiar with the common API concepts used thoroughly in the library.

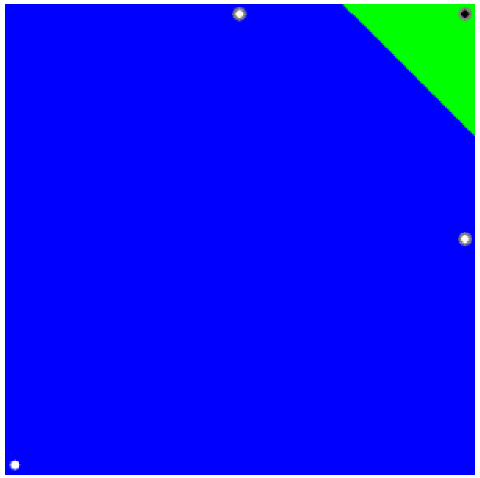
http://opencv.org/

Detection step HOG+SVM (OpenCV)

```cpp
// Set up training data
int labels[4] = {1, -1, -1, -1};
Mat labelsMat(4, 1, CV_32SC1, labels);

float trainingData[4][2] = { {501, 10}, {255, 10}, {501, 255}, {10, 501} };
Mat trainingDataMat(4, 2, CV_32FC1, trainingData);

// Set up SVM's parameters
SVM::Params params;
params.svmType    = SVM::C_SVC;
params.kernelType = SVM::LINEAR;
params.termCrit   = TermCriteria(TermCriteria::MAX_ITER, 100, 1e-6);

// Train the SVM
Ptr<SVM> svm = StatModel::train<SVM>(trainingDataMat, ROW_SAMPLE, labelsMat, params);
```







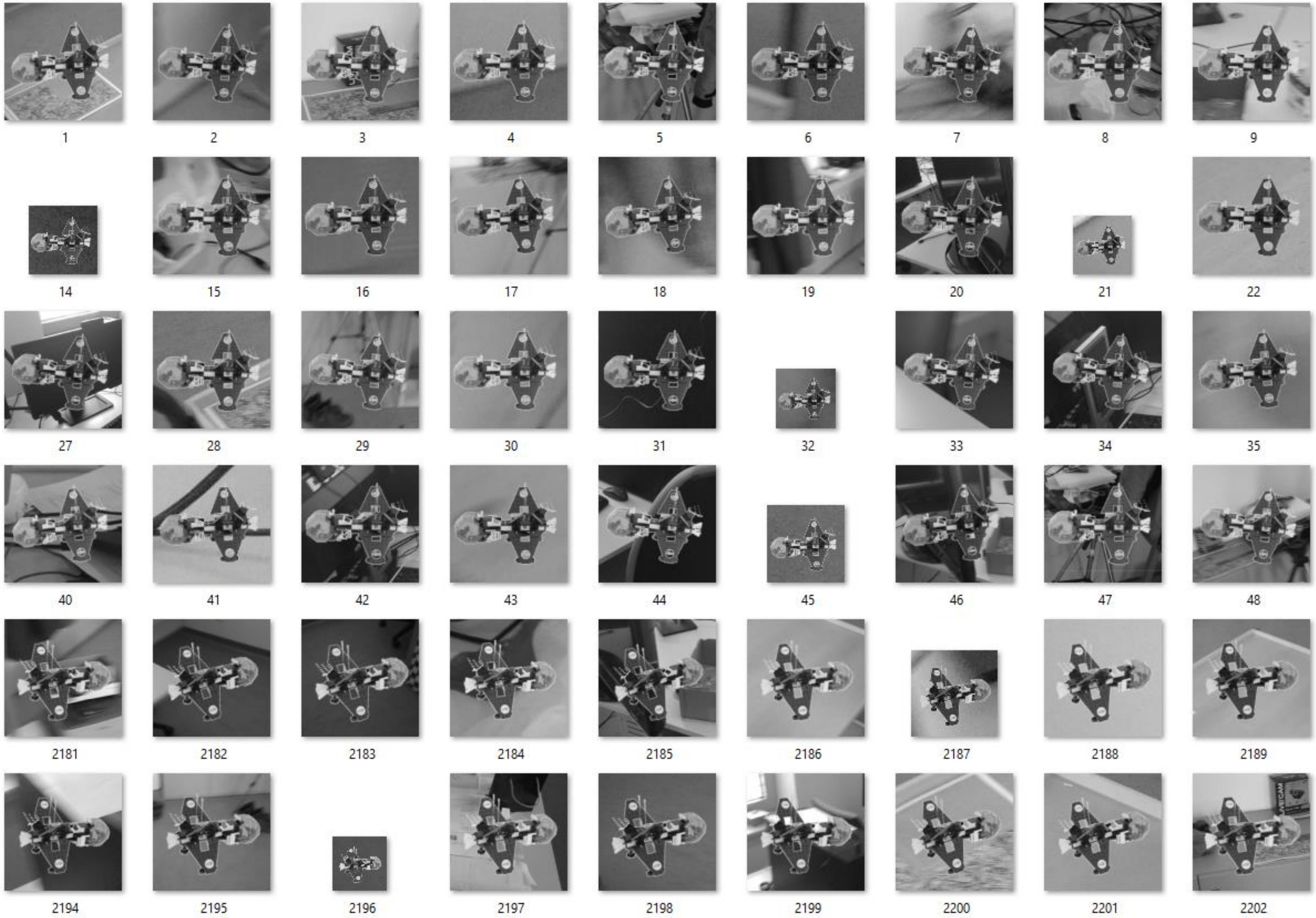
Introduction to Support Vector Machines [online]. [cit. 2020-01-25]. Dostupné z: https://docs.opencv.org/3.1.0/d1/d73/tutorial_introduction_to_svm.html

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Alien

Avenger

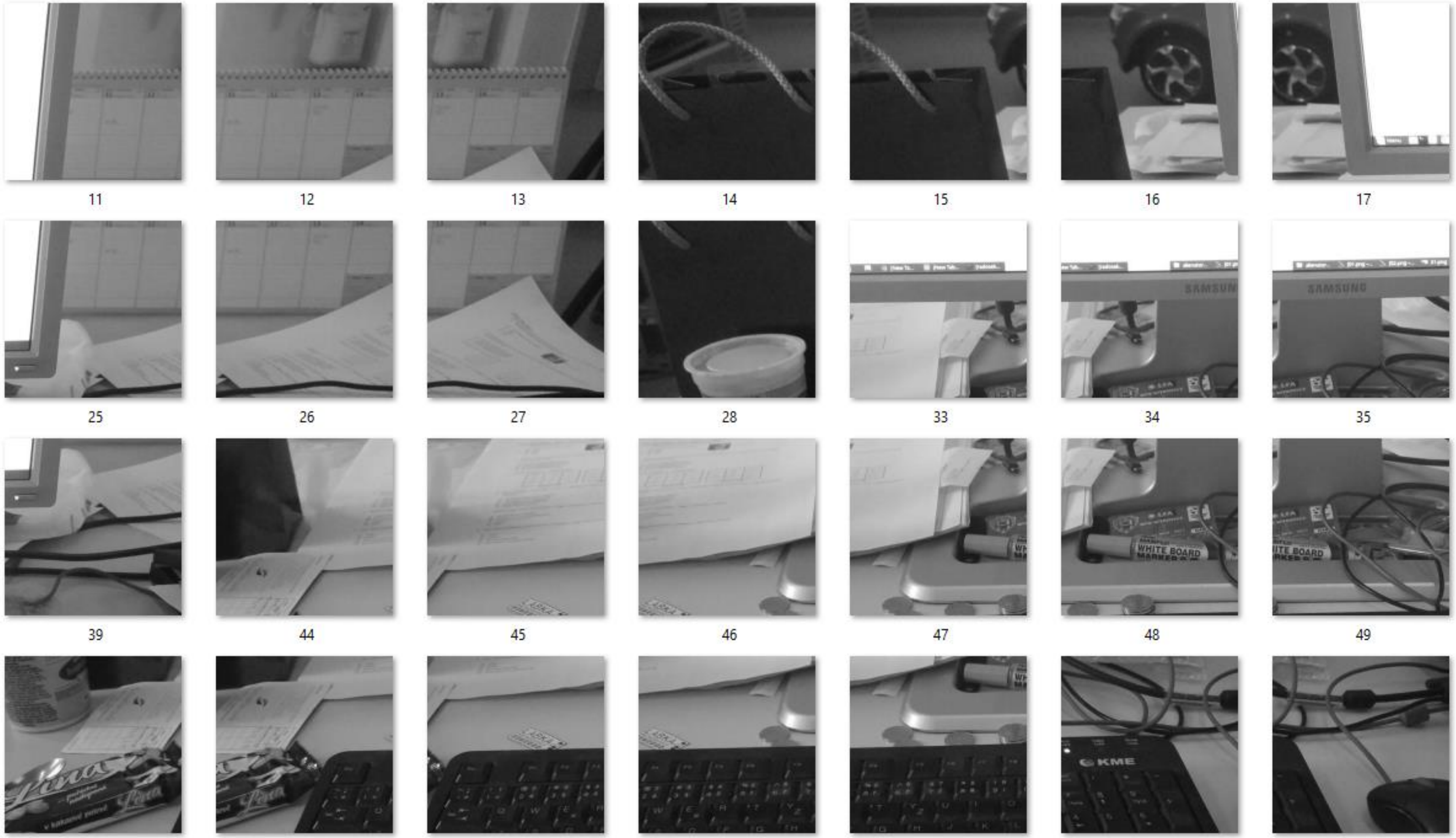

1 2 3 4 5 6 7 8 9
14 15 16 17 18 19 20 21 22
27 28 29 30 31 32 33 34 35
40 41 42 43 44 45 46 47 48
2181 2182 2183 2184 2185 2186 2187 2188 2189
2194 2195 2196 2197 2198 2199 2200 2201 2202

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Negative Set



11 12 13 14 15 16 17

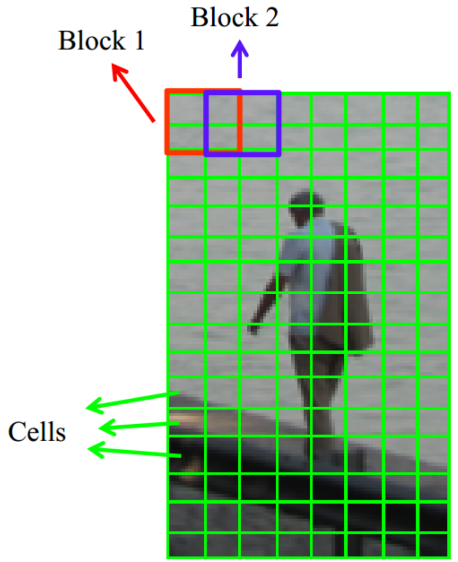
25 26 27 28 33 34 35

39 44 45 46 47 48 49

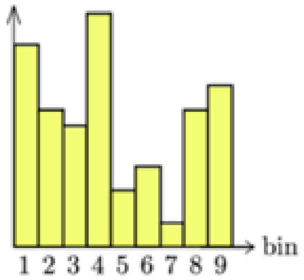
57 58 59 60 61 66 67

Sliding Window (detectMultiScale)

```
int blockSize = 16;
int cellSize = 8;
int strideSize = 8;
int winSize = 64;

//HOGDescriptor hog;
HOGDescriptor my_hog(
            cv::Size(winSize,winSize), //winSize
            cv::Size(blockSize,blockSize), //blocksize
            cv::Size(strideSize,strideSize), //blockStride,
            cv::Size(cellSize,cellSize), //cellSize,
            9, //nbins,
            );

//SVM
Ptr<SVM> svm = StatModel::load<SVM>( classifierName );
std::vector< float > hog_detector;
//get the support vectors
get_svm_detector( svm, hog_detector );
//set SVM
my_hog.setSVMDetector( hog_detector );

std::vector<Rect> positivesAll;
my_hog.detectMultiScale( frameGray, positivesAll, 0,
        Size(0,0), Size(0,0), 1.1, 4);
```



cell histogram



Train_HOG [online]. [cit. 2020-01-25]. Dostupné z: https://github.com/opencv/opencv/blob/master/samples/cpp/train_HOG.cpp

- Haar
- HOG
- LBP

Traditional Approaches

- SIFT, SURF

KeyPoints

- CNNs

Deep Learning Approach

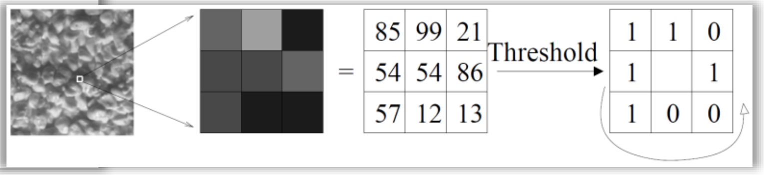
Related Works

2006

Ahonen at al.
(2006)
1300 cit. SCOPUS

Face Description with Local Binary Patterns: Application to Face Recognition

Timo Ahonen, *Student Member, IEEE,* Abdenour Hadid, and Matti Pietikäinen, *Senior Member, IEEE*

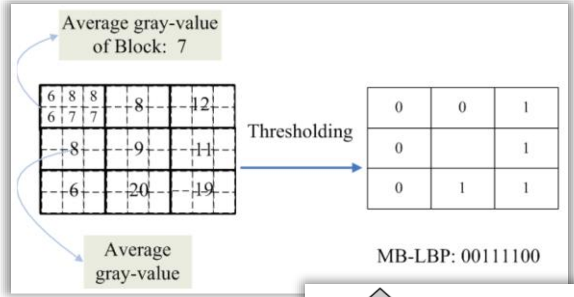


Zhang at al.
(2007)

Face Detection Based on Multi-Block LBP Representation

Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, Stan Z. Li

Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun Donglu Beijing 100080, China

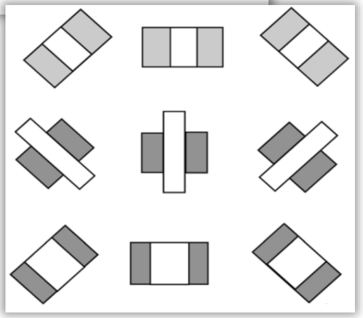


Xiaohua at al.
(2009)

Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers

Li Xiaohua [a,b], Kin-Man Lam [b,*], Shen Lansun [c], Zhou Jiliu [a]

[a] *Department of Computer Science, Sichuan University, Chengdu 610064, China*
[b] *Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[c] *Signal and Information Processing Lab., Beijing University of Technology, Beijing 100022, China*



2009

Object Detection (Analysis)
LBP

LBP - Local Binary Patterns

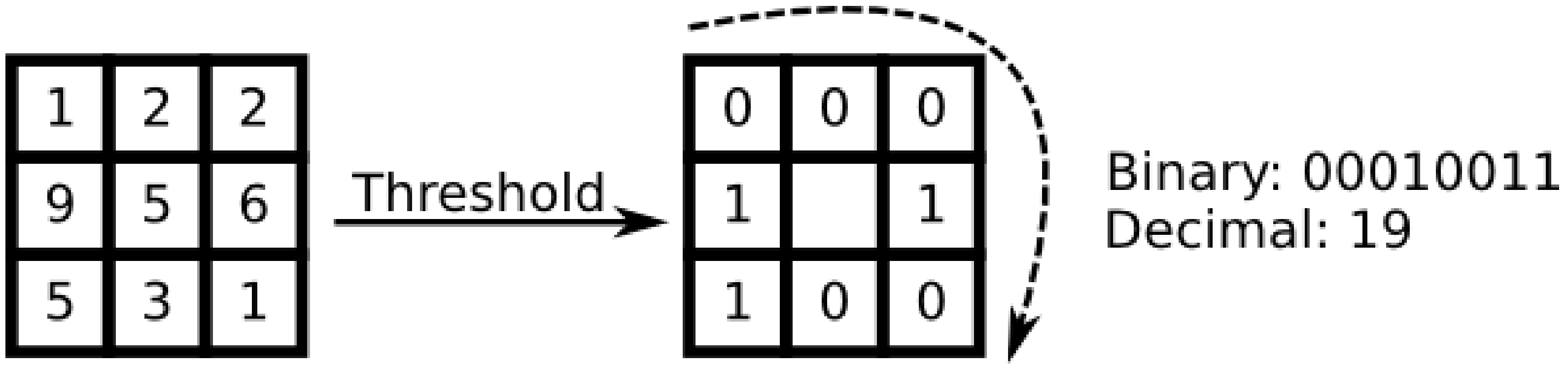
- Were introduced by Ojala et al. for the texture analysis.

- The local binary patterns (LBP) were introduced by Ojala et al. [2, 3] for the texture analysis. The main idea behind LBP is that the local image structures (micro patterns such as lines, edges, spots, and flat areas) can be effciently encoded by comparing every pixel with its neighboring pixels. In the basic form, every pixel is compared with its neighbors in the 3 × 3 region. The result of comparison is the 8-bit binary number for each pixel; in the 8-bit binary number, the value 0 means that the value of center pixel is greater than the neighbor and vice versa. The histogram of these binary numbers (that are usually converted to decimal) is then used to encode the appearance of region.

LBP - Local Binary Patterns

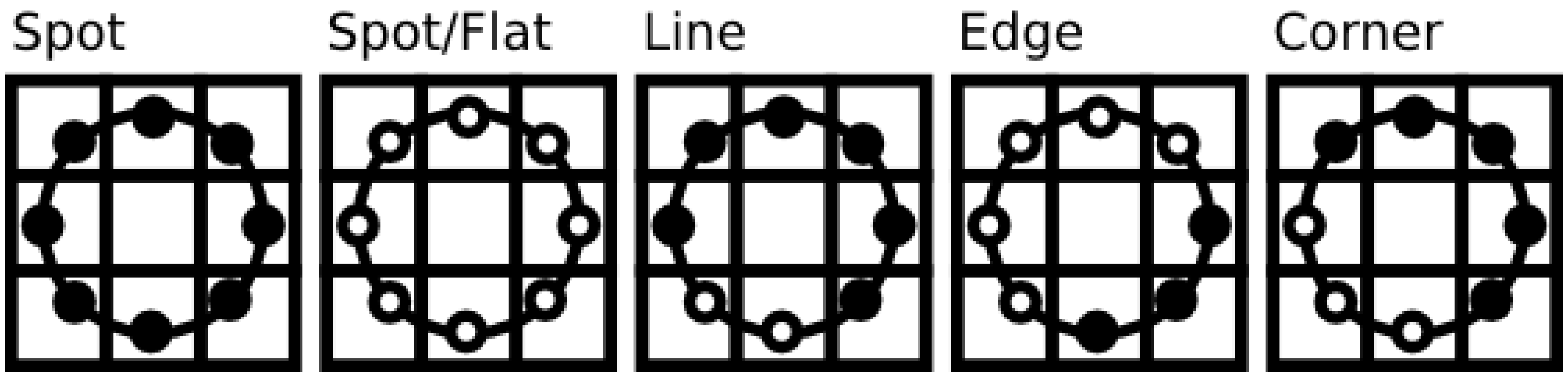
- The important properties of LBP are the resistance to the lighting changes and a low computational complexity.

- Duo to their properties, LBP were used in many detection tasks, especially in facial image analysis [1, 4].



Local Binary Patterns Histograms [online]. [cit. 2020-01-25]. Dostupné z: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

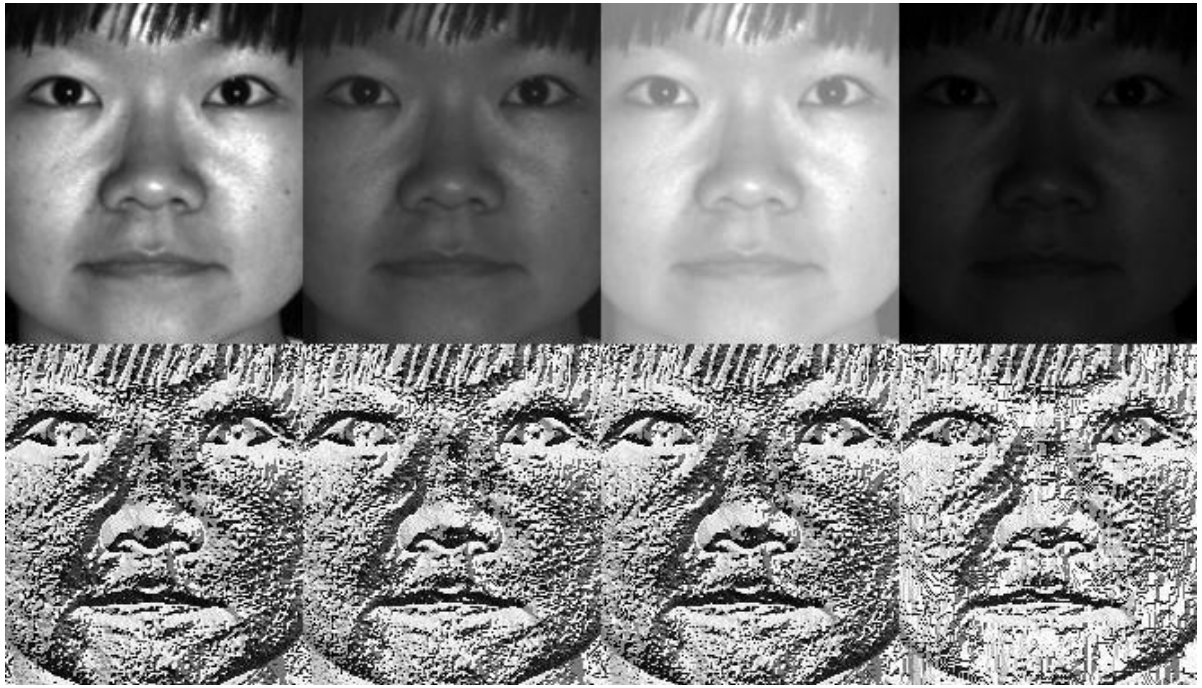
LBP - Local Binary Patterns



Local Binary Patterns Histograms [online]. [cit. 2020-01-25]. Dostupné z:
https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

LBP - Local Binary Patterns

- Robust to monotonic changes in illumination

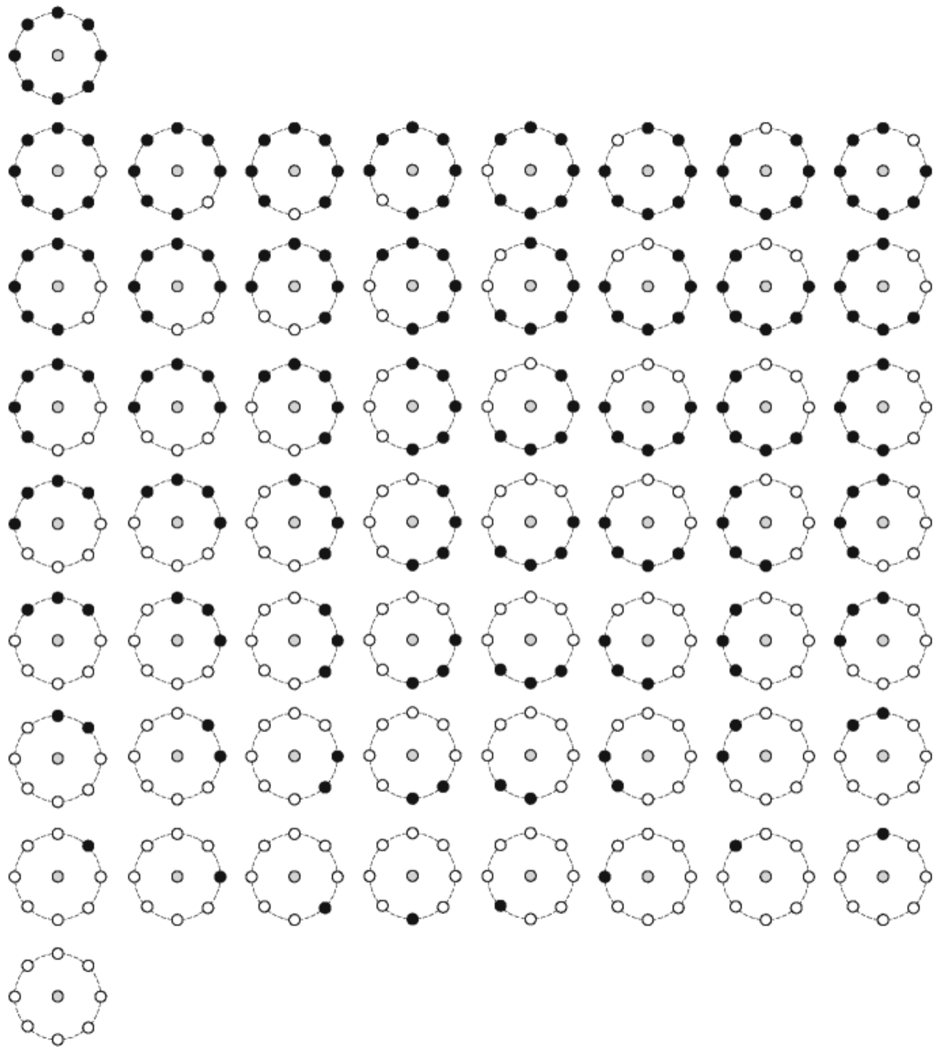


Local Binary Patterns Histograms [online]. [cit. 2020-01-25]. Dostupné z:
https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

LBP - Local Binary Patterns (uniform)

- **Uniform** patterns example:
- 1 1 1 1 1 1 1 1 (0 transitions)
- 1 1 0 0 1 1 1 1 (2 transitions)

- **Non-uniform** patterns example:
- more than two transitions
- 0 0 1 0 0 1 0 1
- 0 0 0 0 0 1 0 1



Feature Extraction [online]. [cit. 2020-01-25]. Dostupné z: https://ebrary.net/37818/engineering/feature_extraction

LBP - Local Binary Patterns

[1] Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(12), 2037–2041 (2006)

[2] Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classifcation based on featured distributions. Pattern Recognition 29(1), 51–59 (Jan 1996), http://dx.doi.org/10.1016/0031-3203(95)00067-4

[3] Ojala, T., Pietikainen, M., Maenpaa, T.: A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classifcation. In: Proceedings of the Second International Conference on Advances in Pattern Recognition. pp. 397–406. ICAPR '01, Springer-Verlag, London, UK, UK (2001), http://dl.acm.org/citation.cfm?id=646260.685274

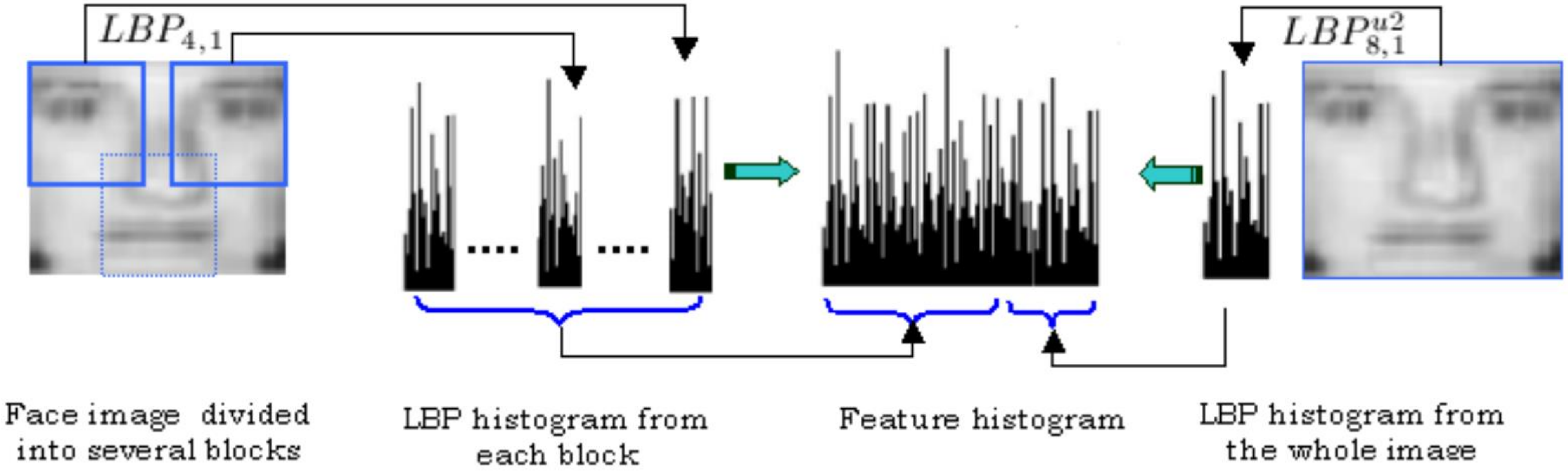
[4] Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J. (eds.) Computer Vision - ECCV 2004, Lecture Notes in Computer Science, vol. 3021, pp. 469–481. Springer Berlin Heidelberg (2004)

LBP - Local Binary Patterns

In [1], LBP were used for solving the face detection problem in low-resolution images. In this approach, the 19 × 19 face images are divided into the 9 overlapping regions in which the LBP descriptors are computed. Additionally, the LBP descriptors are extracted from the whole 19 × 19 image. The descriptors are then used to create the feature vector, and the SVM classifer with a polynomial kernel is used for the fnal classifcation.

[1] Hadid, A., Pietikainen, M., Ahonen, T.: A discriminative feature space for detecting and recognizing faces. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 2, pp. II–797–II–804 Vol.2 (2004)

LBP - Local Binary Patterns



[1] Hadid, A., Pietikainen, M., Ahonen, T.: A discriminative feature space for detecting and recognizing faces. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 2, pp. II–797–II–804 Vol.2 (2004)

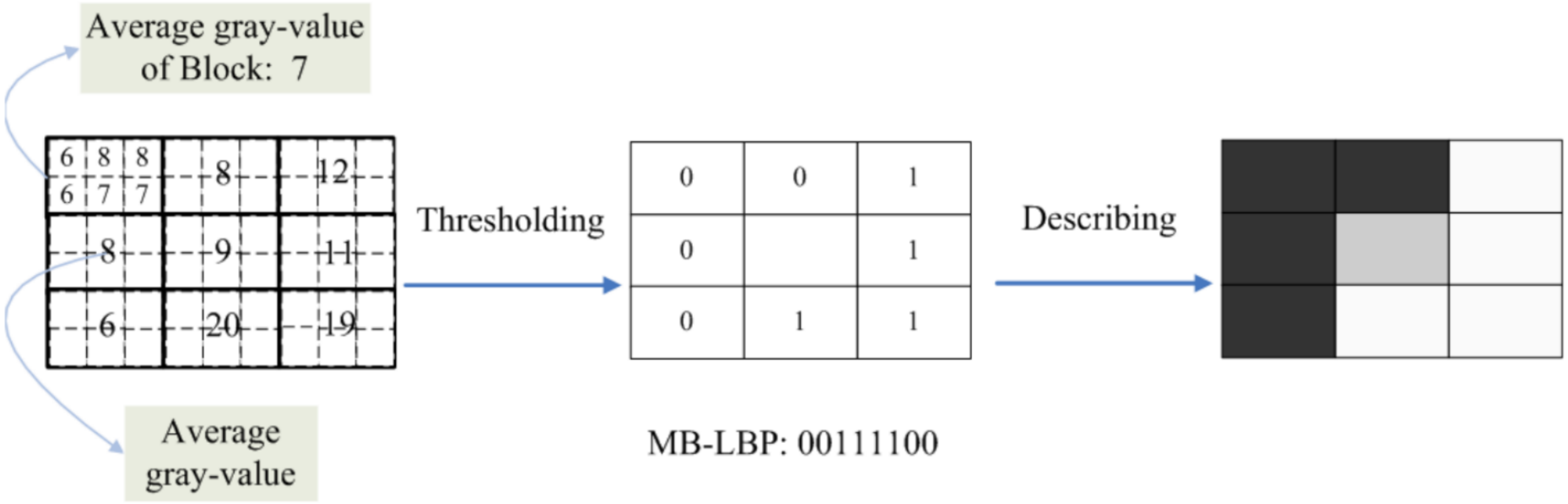
LBP - Local Binary Patterns

Multi-block local binary patterns (MB-LBP) for face detection and recognition were proposed in [1, 2]. In this method, the authors encode the rectangular regions by the local binary pattern operator and the Gentle AdaBoost is used for feature selection. Their results showed that MBLBP are more distinctive than the Haar-like features and the original LBP features.

[1] Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: Proceedings of the 2007 international conference on Advances in Biometrics. pp. 11–18. ICB'07, Springer-Verlag, Berlin, Heidelberg (2007)

[2] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB. pp. 828–837 (2007)

LBP - Local Binary Patterns



[1] Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: Proceedings of the 2007 international conference on Advances in Biometrics. pp. 11–18. ICB’07, Springer-Verlag, Berlin, Heidelberg (2007)

[2] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB. pp. 828–837 (2007)

LBP - Local Binary Patterns

The paper of Tan and Triggs [2] proposed the face recognition method with robust preprocessing based on the difference of Gaussian image filter combined with LBP in which the binary LBP code is replaced by the ternary code to create local ternary patterns (LTP).

LBP were also successfully used for the facial expression analysis. The coarse-tofine classification scheme with LBP combined with the k-nearest neighbor classifier that carries out the final classification was proposed in [1].

The comprehensive study of facial expression recognition using LBP was proposed in [78], the survey of facial image analysis using LBP was presented in [38].

[1] Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under diffcult lighting conditions. Image Processing, IEEE Transactions on 19(6), 1635–1650 (2010)
[2] Feng, X., Hadid, A., Pietikainen, M.: A coarse-to-fne classifcation scheme for facial expression recognition. In: Campilho, A., Kamel, M. (eds.) Image Analysis and Recognition. Lecture Notes in Computer Science, vol. 3212, pp. 668–675. Springer Berlin Heidelberg (2004)
[3] Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image Vision Comput. 27(6), 803–816 (May 2009)
[4] Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L.: Local binary patterns and its application to facial image analysis: A survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41(6), 765–781 (Nov 2011)

- Haar
- HOG
- LBP

Traditional Approaches

- SIFT, SURF

KeyPoints

- CNNs

Deep Learning Approach

KeyPoints

The most of the previously mentioned methods for object description were based on the fact that the descriptors were extracted over the whole image (sliding window) that was usually divided into the overlap or non-overlap regions. Inside these regions, the descriptors were calculated and combined to the final feature vector that was used as an input for the classifier.

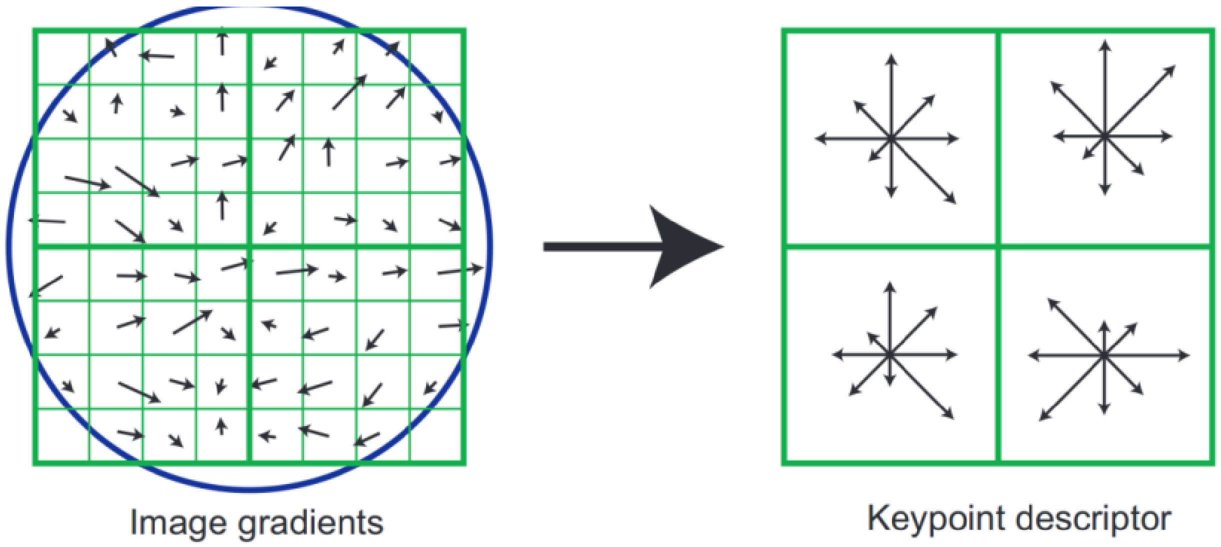
In this lecture, we present the state-of-the-art descriptors that are based on the fact that the regions (within which the descriptors are extracted) are selected using the keypoint detectors.

KeyPoints - SIFT

One of the most popular descriptors based on the interest points was proposed by David Lowe [1, 2, 3]. The method is called scale invariant feature transform (SIFT).

The idea of the SIFT descriptor is that the interesting points (keypoints) of the objects can be extracted to provide the key information about the objects. The gradient magnitude and orientation are computed around the keypoint location; the histograms are then summarized over subregions (see following image). The keypoints are extracted from the reference image (that contains the object of interest) and also from the target image (that possibly contains the object of interest). The extracted keypoints are matched to find similarity between the images.

KeyPoints - SIFT



An example of SIFT keypoint descriptor in which the gradient orientation and gradient magnitude around each interest point are used [3].

[1] Brown, M., Lowe, D.: Invariant features from interest point groups. In: In British Machine Vision Conference. pp. 656–665 (2002)

[2] Lowe, D.: Object recognition from local scale-invariant features. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. vol. 2, pp. 1150–1157 vol.2 (1999)

[3] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (Nov 2004), http://dx.doi.org/10.1023/B: VISI.0000029664.99615.94

KeyPoints - SURF

The speeded up robust feature (SURF) descriptor by Bay et al. [1, 2] is also one of the widely used keypoint descriptors. In this method, the Hessian matrix-based measure is used to find the points of interest. The sum of the Haar-wavelet responses within the neighborhood of interest point is calculated. The authors also use the fast calculation via the integral image thanks to which SURF is faster than SIFT.

[1] Bay, H., Tuytelaars, T., Gool, L.J.V.: Surf: Speeded up robust features. In: ECCV (1). pp. 404–417 (2006)

[2] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. 110(3), 346–359 (Jun 2008)

KeyPoints – BRIEF/ORB

A very fast method called binary robust independent elementary features (BRIEF) was proposed by Calonder et al. [1]. The authors reported that the method outperforms SURF in the term of speed, and the recognition rate in many cases. In BRIEF, a binary string that contains the results of intensity differences of pixels are used and the descriptor similarity is evaluated using the Hamming distance. In [2], the authors proposed another binary descriptor with rotation and noise invariant properties called oriented fast and rotated BRIEF (ORB).

[1] Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: binary robust independent elementary features. In: Proceedings of the 11th European conference on Computer vision: Part IV. pp. 778–792. ECCV’10, Springer-Verlag, Berlin, Heidelberg (2010)

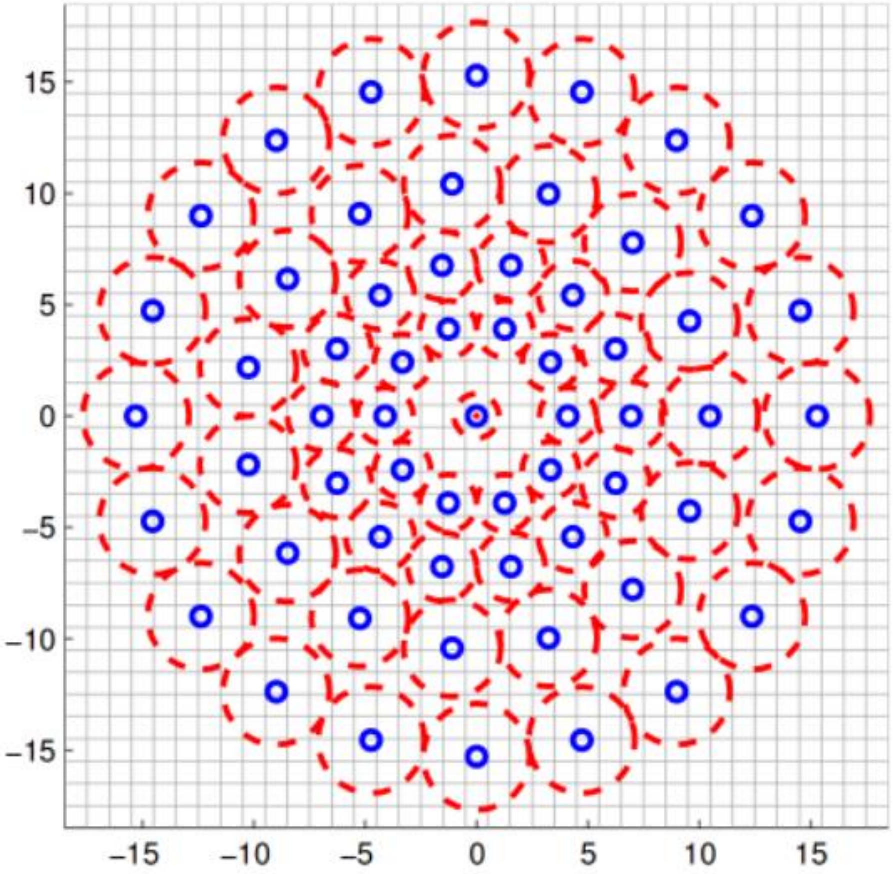
[2] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An effcient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2564–2571 (2011)

KeyPoints – BRISK

Leutenegger et al. [1] proposed binary robust invariant scalable keypoints (BRISK). The method provides both scale and rotation invariance. BRISK is a binary descriptor like BRIEF and ORB, it means that the binary string that represents a region around the keypoint is composed. In BRISK, a concentric circle pattern of points near to the keypoint is used (see following image). In this pattern, the blue circles represent the sampling locations and Gaussian blurring is computed to be less sensitive to noise; the radius of red circles denotes a standard deviation of blurring kernel. The standard deviation of the Gaussian kernel is increased with the increasing distance from the feature center to avoid aliasing effects. The final descriptor is determined by the comparison of sample points.

Radovan Fusek: Descriptors for Object Detection in Image Recognition, Philosophiæ Doctor Thesis, 2016

KeyPoints – BRISK



BRISK sampling pattern [1]

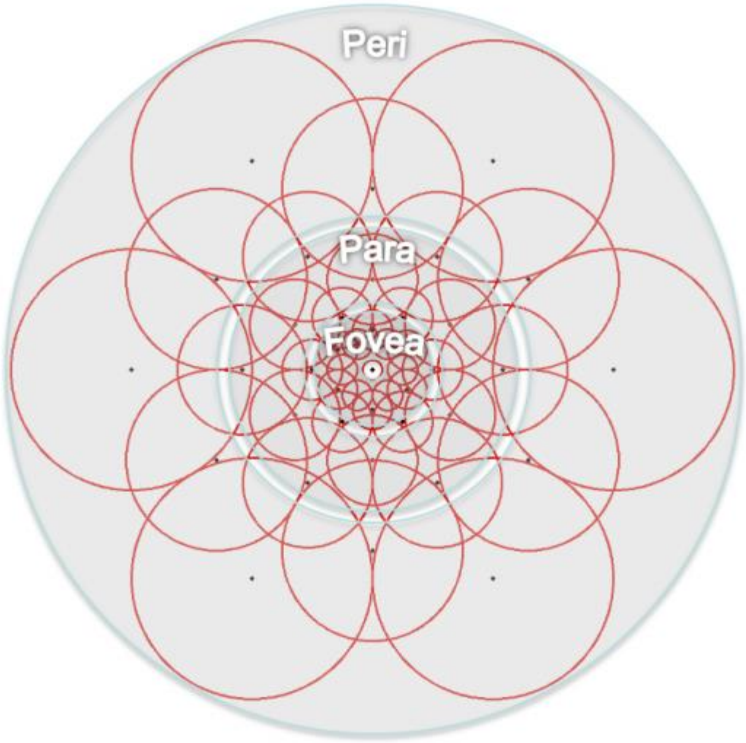
[1] Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2548–2555 (2011)

KeyPoints – FREAK

In [1], the authors proposed the fast retina keypoint (FREAK) descriptor that also uses the binary strings. The method is biologically inspired by a human visual system; more exactly by the retina. In this paper, the authors proposed a retinal sampling pattern; The following image shows the topology of this pattern. The pattern is divided into the areas (foveal, fovea, parafoveal, and perifoveal) similar to the human retina. In this pattern, the pixels are overlapped and concentrated near to the center. The binary strings is computed by comparing the point pairs of image intensities within the pattern.

[1] Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition, Ieee, New York (2012)

KeyPoints – FREAK

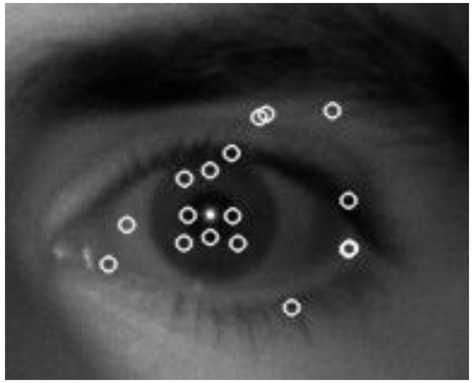


FREAK sampling pattern [1]

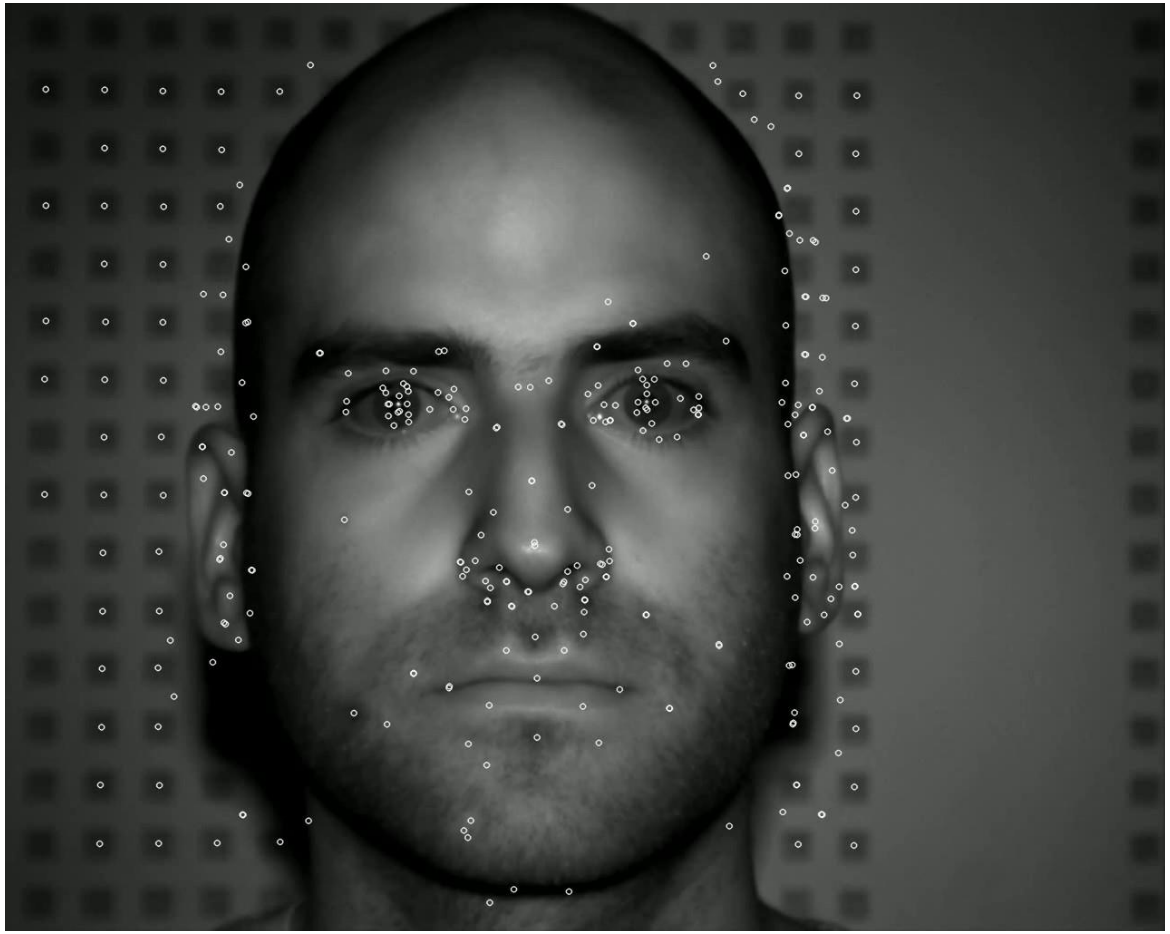
[1] Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition, Ieee, New York (2012)

KeyPoints - Example

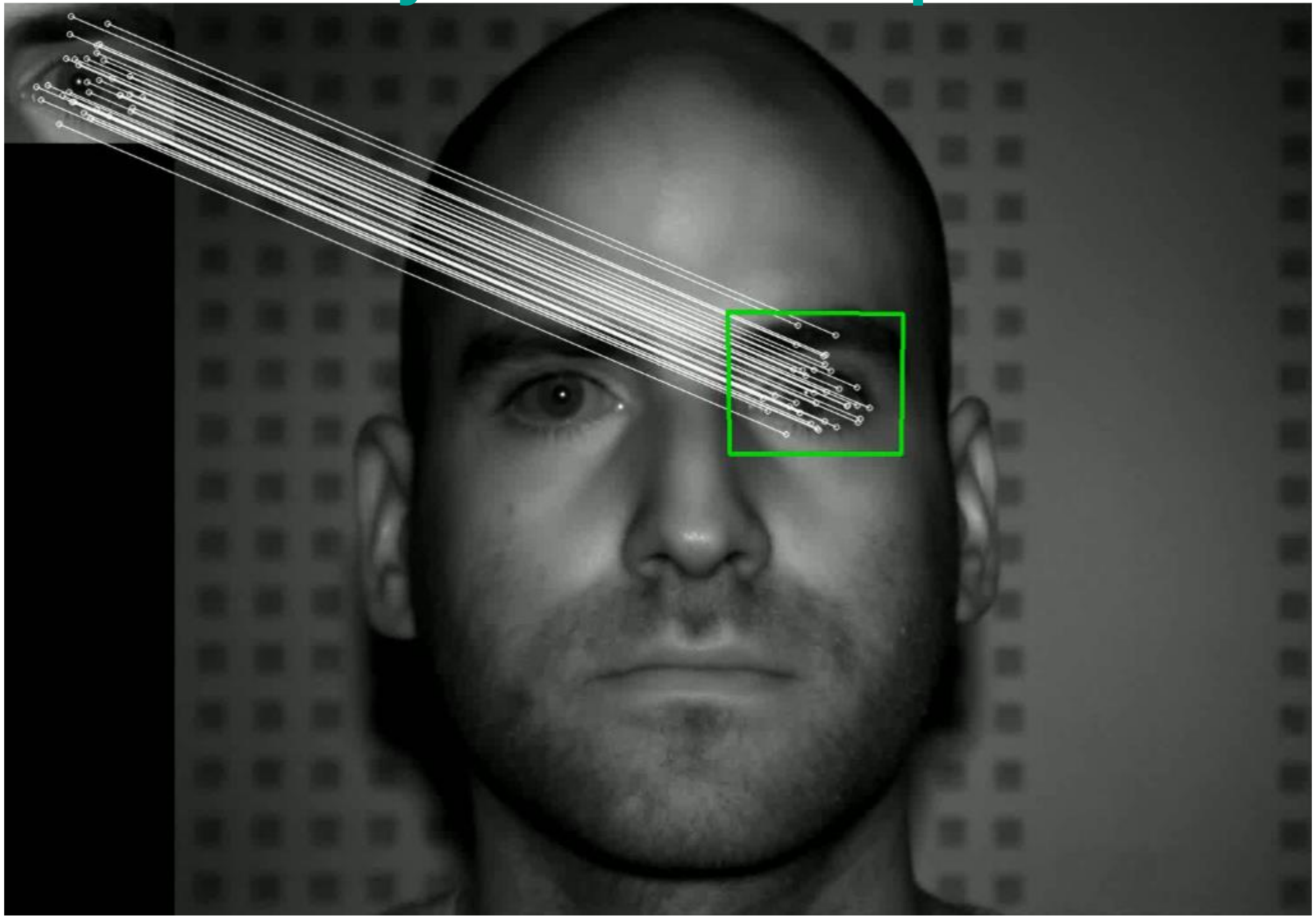
The goal is to find image KeyPoints that are invariant in the terms of scale, orientation, position, illumination, partially occlusion.



template



KeyPoints - Example



https://docs.opencv.org/3.1.0/d5/d6f/tutorial_feature_flann_matcher.html

- Haar
- HOG
- LBP

Traditional Approaches

- SIFT, SURF

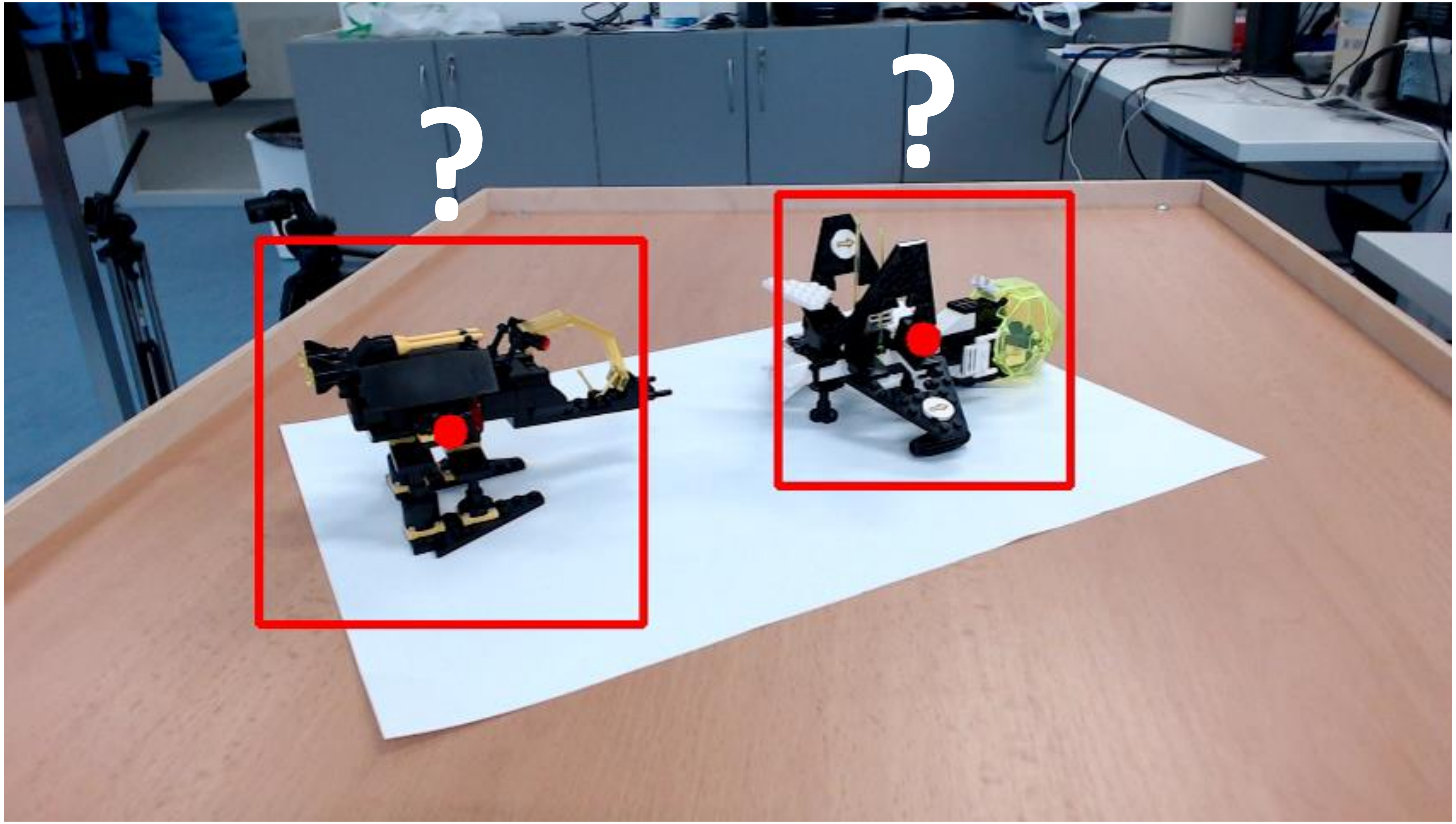
KeyPoints

- CNNs

Deep Learning Approach

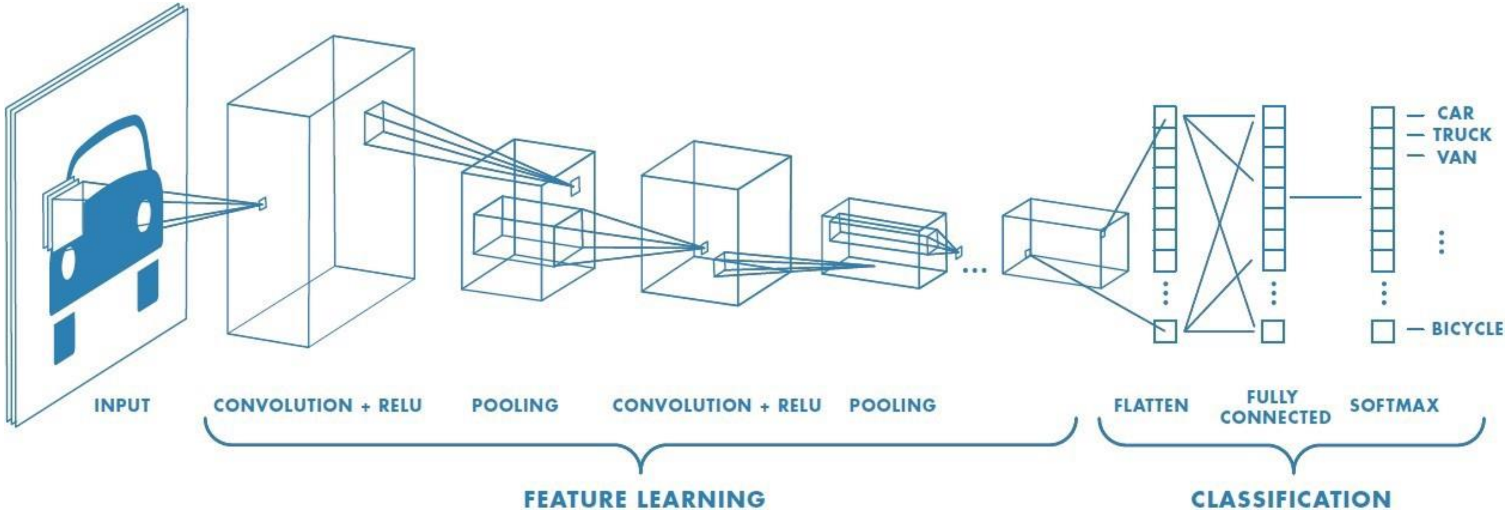
Recognition

Alien vs. Avenger



CNNs – Main Steps (LeNet)

1. Convolution
2. Non Linearity (ReLU)
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)



A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

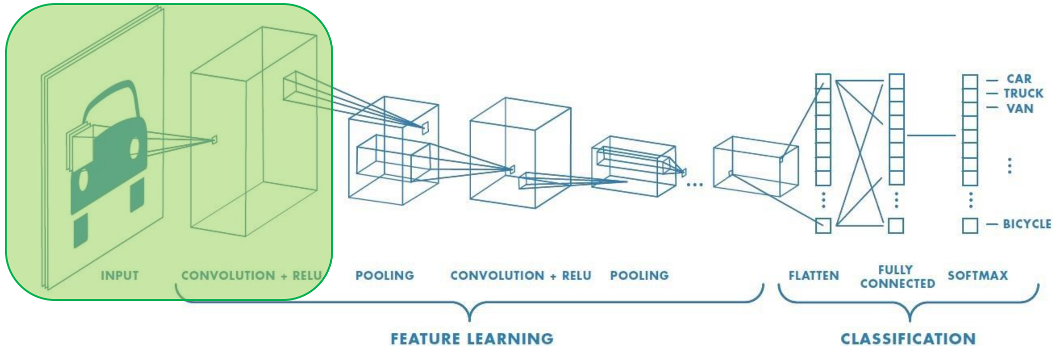
1. Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input Image

1	0	1
0	1	0
1	0	1

Mask/Filter



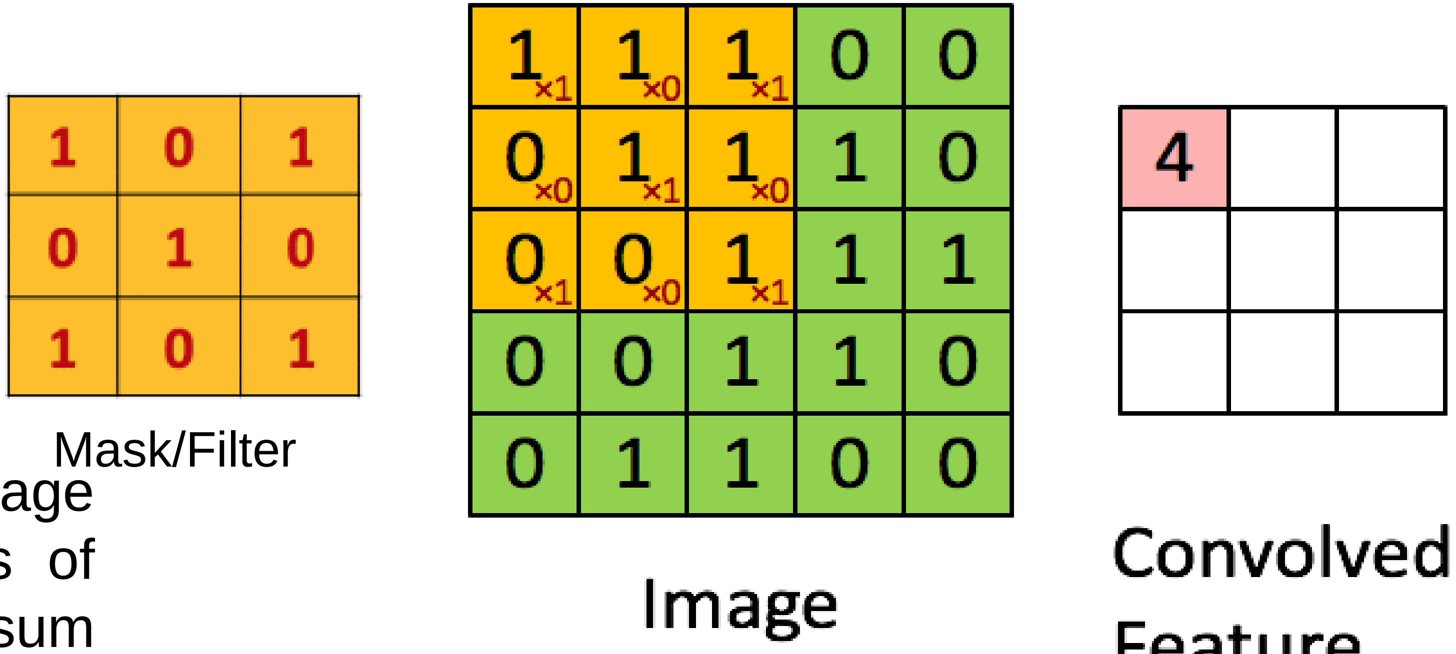
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

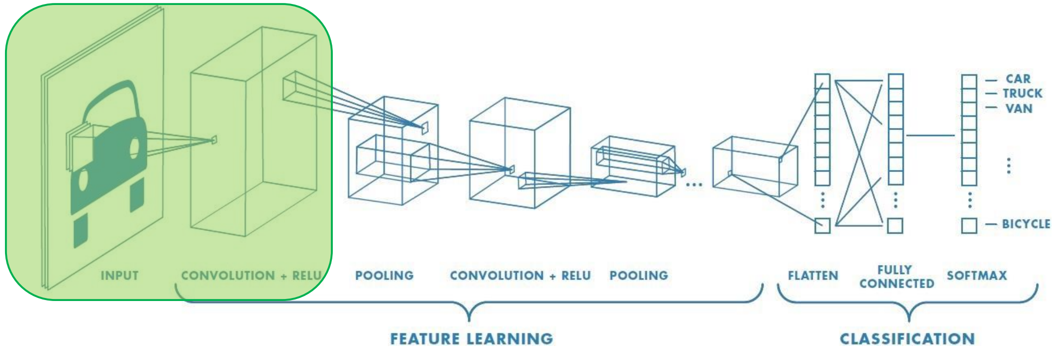
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

1. Convolution



Multiply the image pixels by pixels of the filter, then sum the results



A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

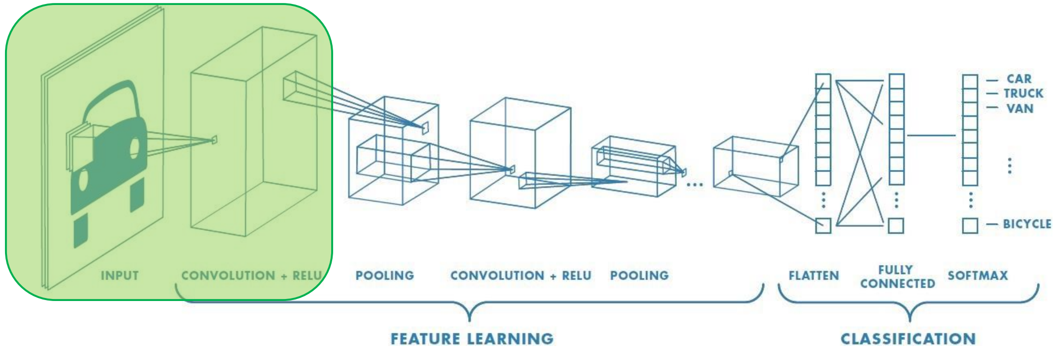
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

1. Convolution

Operation	Kernel ω	Image result g(x,y)
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Ridge detection	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

Depending on the element values, a kernel can cause a wide range of effects..



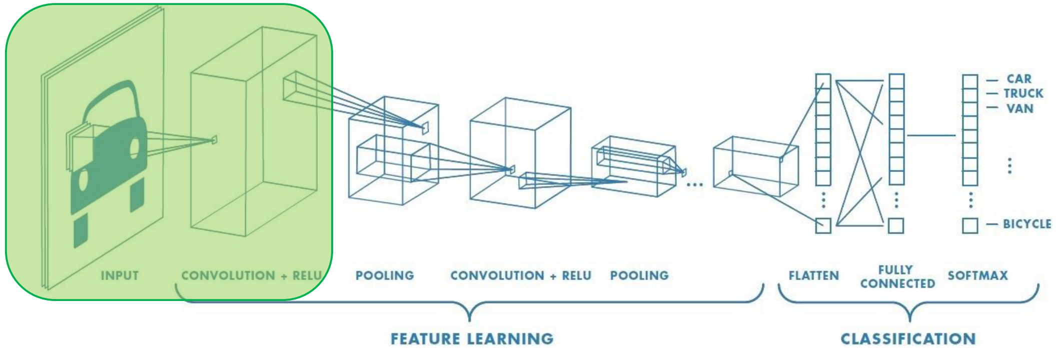
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Kernel (image processing) [online]. [cit. 2020-01-25]. Dostupné z: https://en.wikipedia.org/wiki/Kernel_(image_processing)

1. Convolution

- Before training, we have many filters/kernels
  - Filter values are randomized
- Depth of this conv. layer corresponds to the number of filters we use for the convolution operation

- The filters are learned during the training

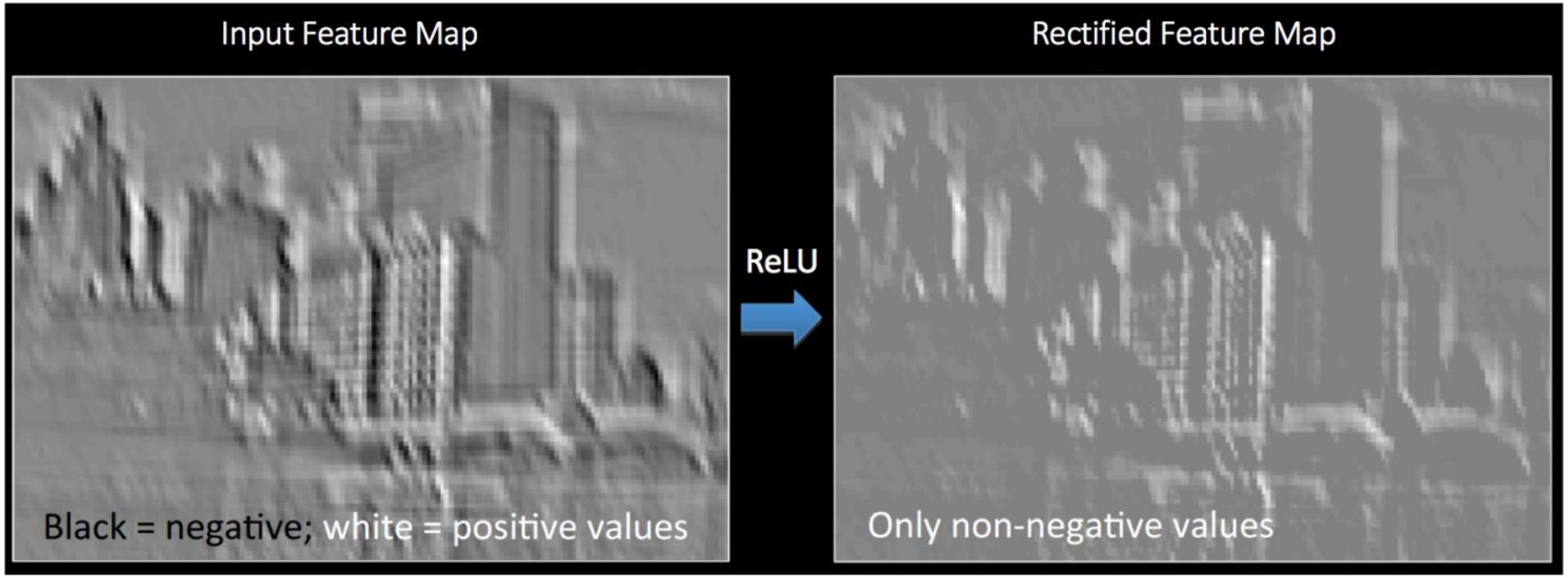


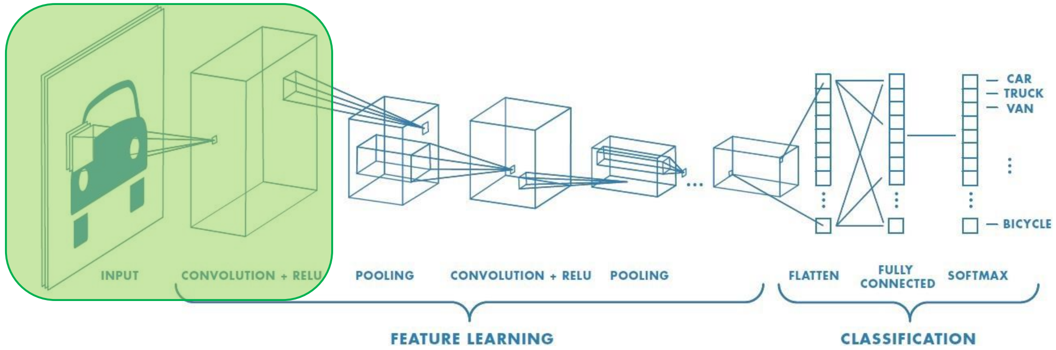
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

http://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

2. Non Linearity (ReLU)

- ReLU is used after every Convolution operation
- The goal of this step is to replace all negative pixels by zero in the feature map





A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

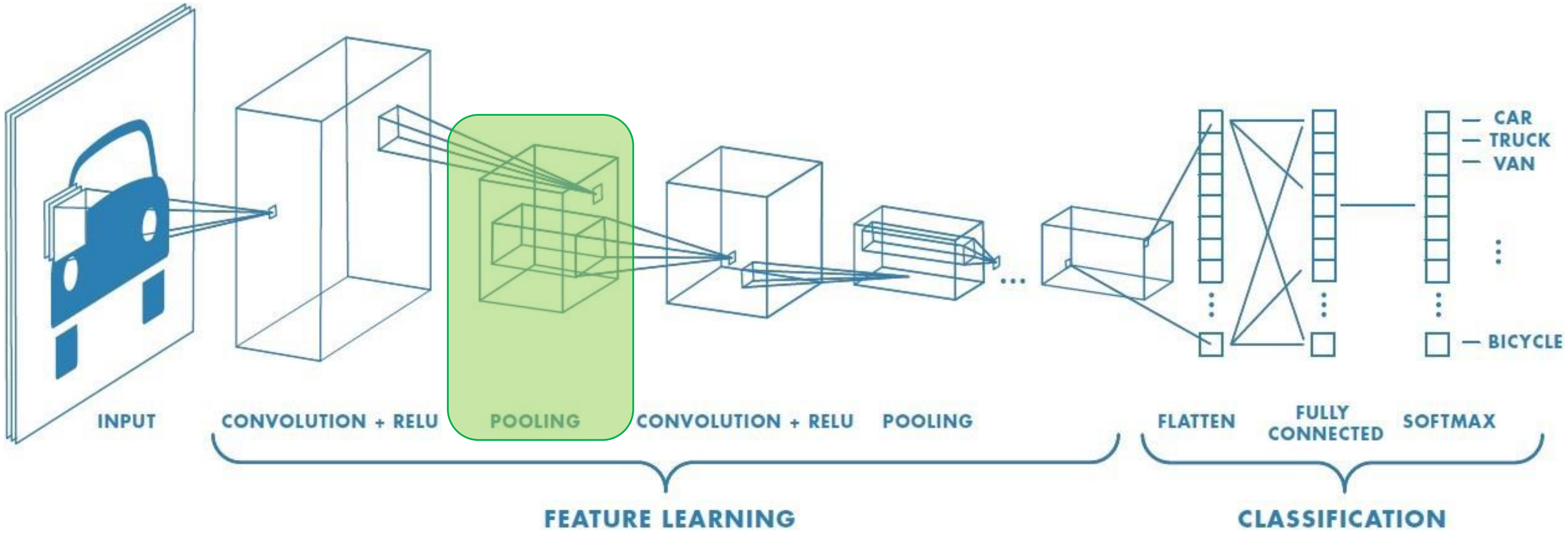
Deep Learning Methods for Vision [online]. [cit. 2020-01-25]. Dostupné z: https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

3. Pooling

(Subsampling or downsampling)

- The goal of this step is to reduce the dimensionality of each feature map but preserve important informations

- Operations: e.g. Sum, Average, Max

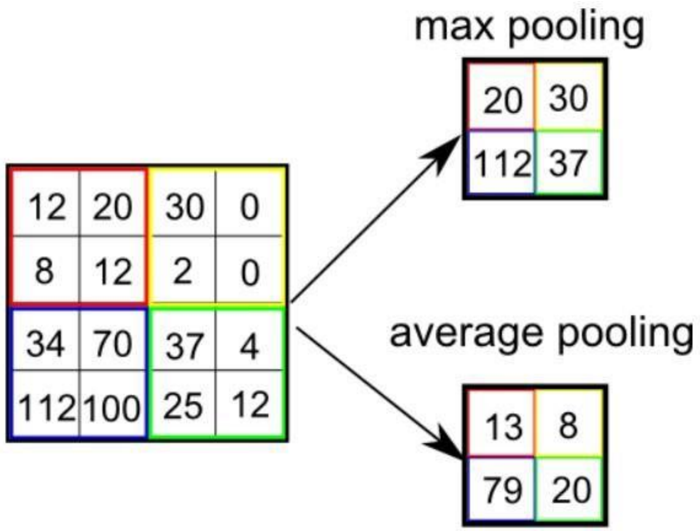


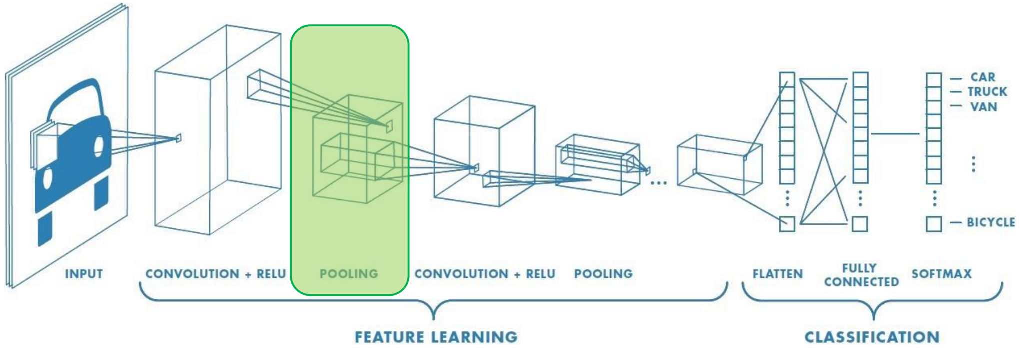
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3. Pooling

(Subsampling or downsampling)

- Common way is a pooling layer with filters of size 2x2 applied with a stride of 2

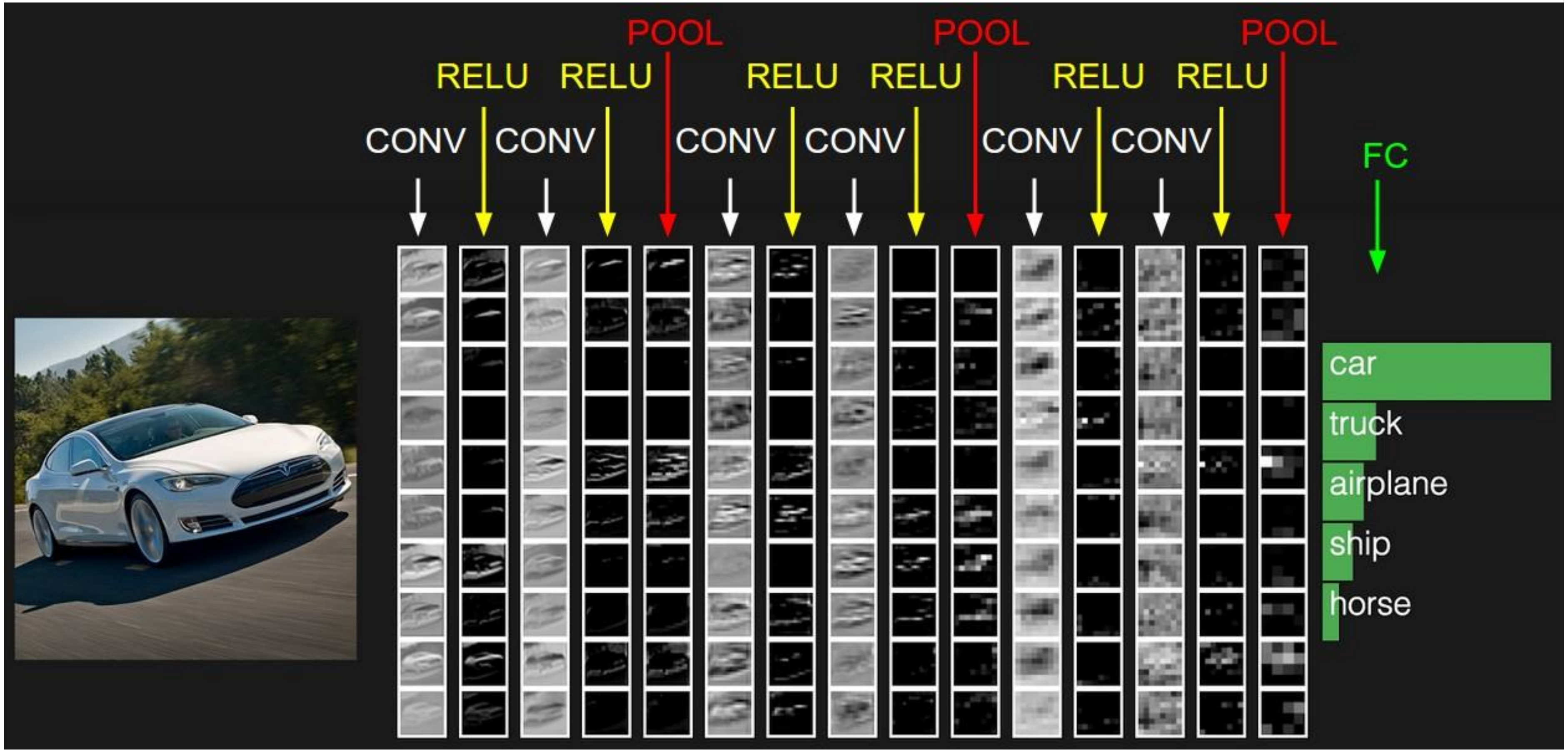




A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Conv. + ReLU + POOL

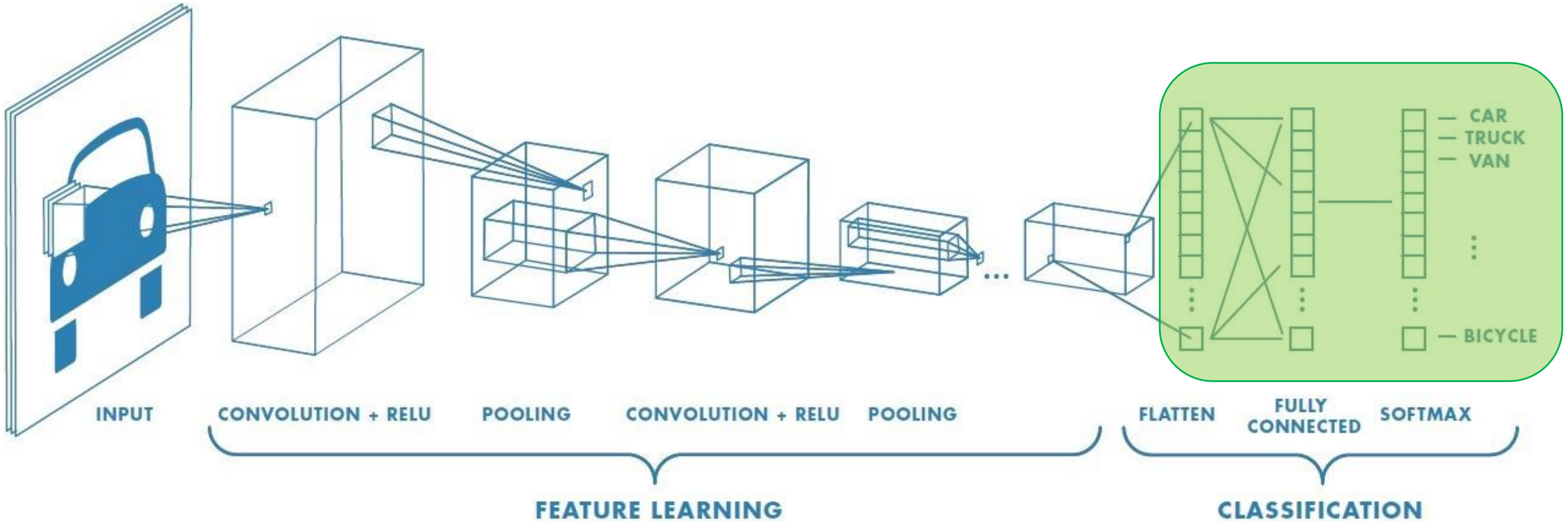
- Convolution layers and Pooling layers can be repeated any number of times in a single ConvNet.



Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://cs231n.github.io/convolutional-networks/

4. Classification

- Multi Layer Perceptron
- The number of filters, filter sizes, architecture of the network etc. are fixed and do not change during training process.
- Only the values of the filter matrix and connection weights get updated.



A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

CovNet Architectures

- LeNet (1990s)
- AlexNet (2012)
- ZF NET (2013)
- GoogLeNet (2014)
- VGGNet (2014)
- ResNets (2015)
- DenseNet (2016)

Dlib http://dlib.net

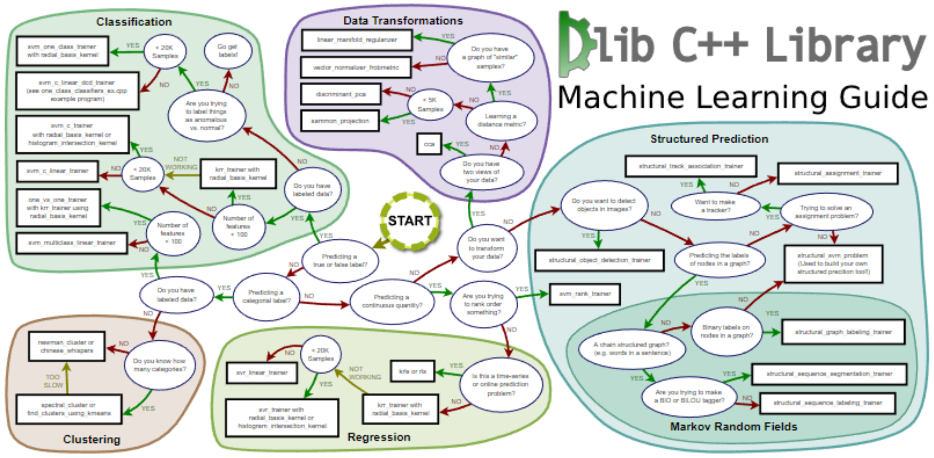
Dlib C++ Library

Google Custom Search

The Library
- Algorithms
- API Wrappers
- Bayesian Nets
- Compression
- Containers
- Graph Tools
- Image Processing
- Linear Algebra
- Machine Learning
- Metaprogramming
- Miscellaneous
- Networking
- Optimization
- Parsing

Help/Info
- Dlib Blog
- ⊞Examples: C++
- ⊞Examples: Python
- FAQ
- Home
- How to compile
- How to contribute
- Index
- Introduction
- License
- Python API

Machine Learning



Dlib contains a wide range of machine learning algorithms. All designed to be highly modular, quick to execute, and simple to use via a clean and modern C++ API. It is used in a wide range of applications including robotics, embedded devices, mobile phones, and large high performance computing environments. If you use dlib in your research please cite:

Primary Algorithms

Binary Classification
- rvm_trainer
- svm_c_ekm_trainer
- svm_c_linear_dcd_trainer
- svm_c_linear_trainer
- svm_c_trainer
- svm_nu_trainer
- svm_pegasos
- train_probabilistic_decision_function

Multiclass Classification
- one_vs_all_trainer
- one_vs_one_trainer
- svm_multiclass_linear_trainer

Regression
- krls
- krr_trainer
- mlp
- rbf_network_trainer
- rls
- rr_trainer
- rvm_regression_trainer
- svr_linear_trainer
- svr_trainer

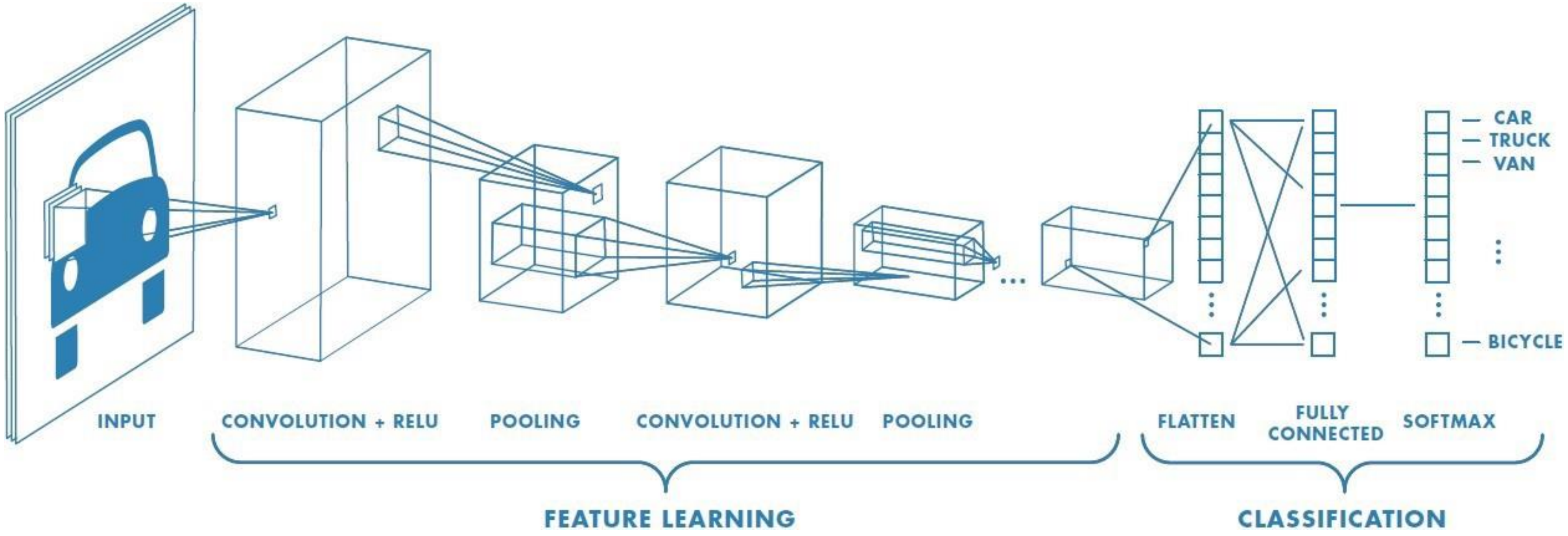
Dlib [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>;
```



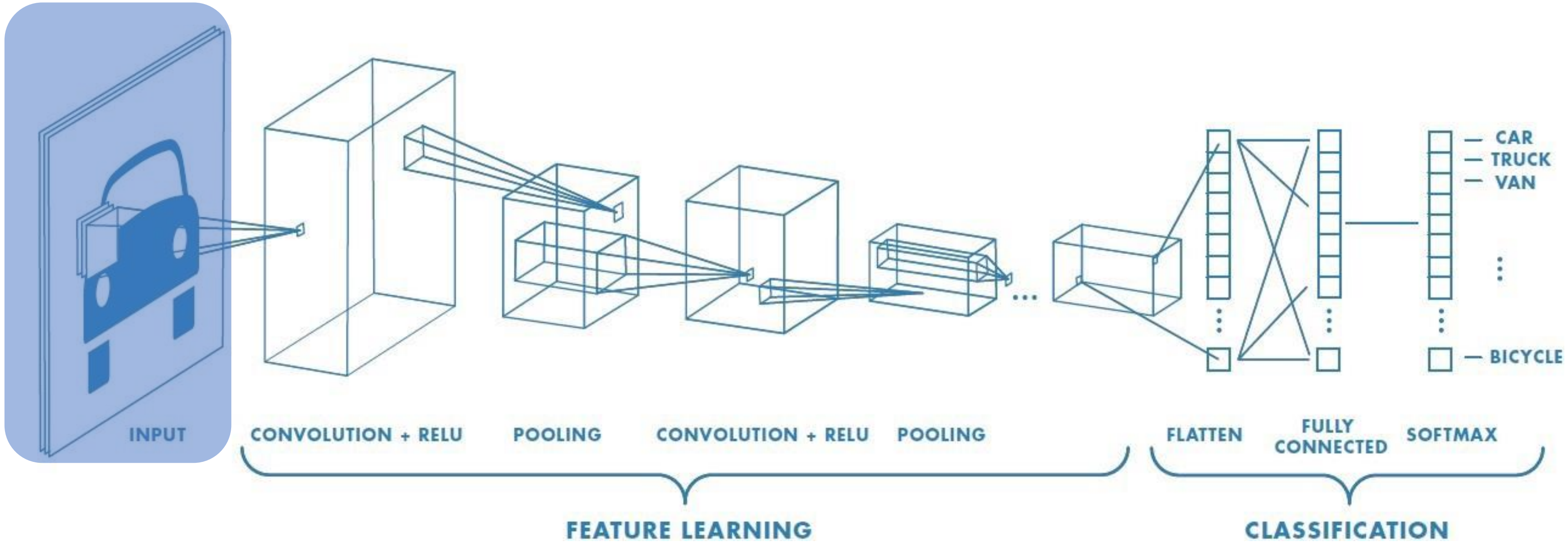
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>;
```

Input Image



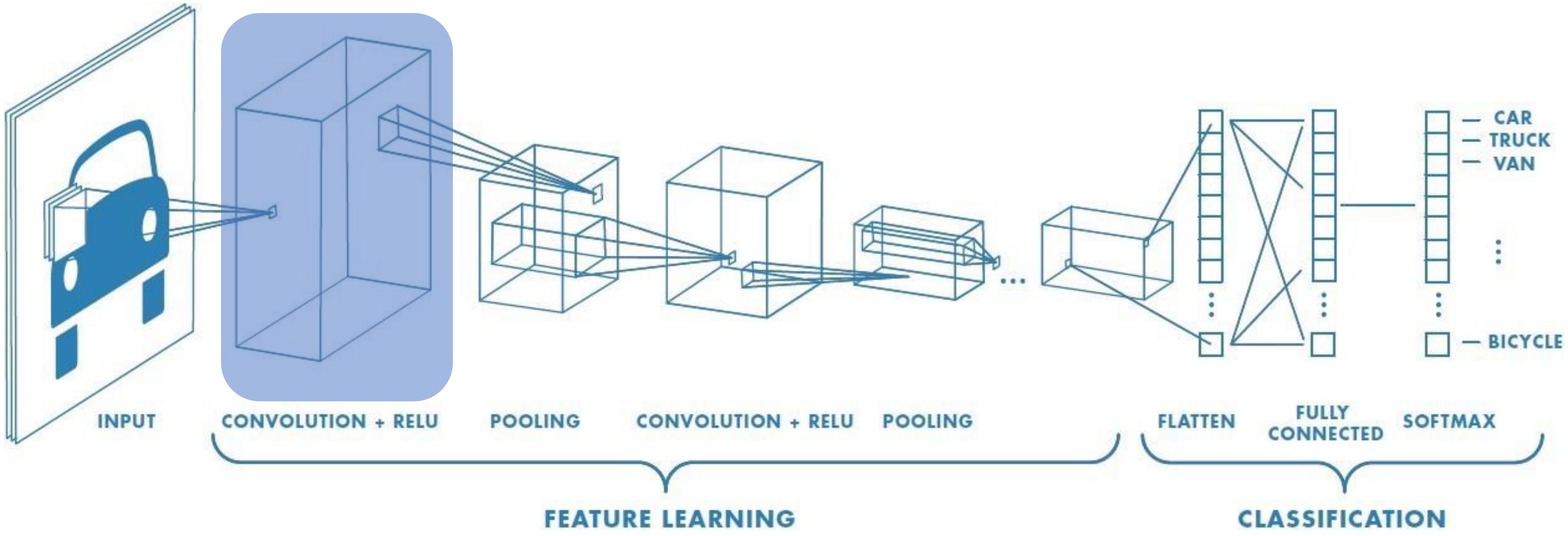
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>;
```

6 conv. filters
5x5 filter size
1x1 stride
+ReLU



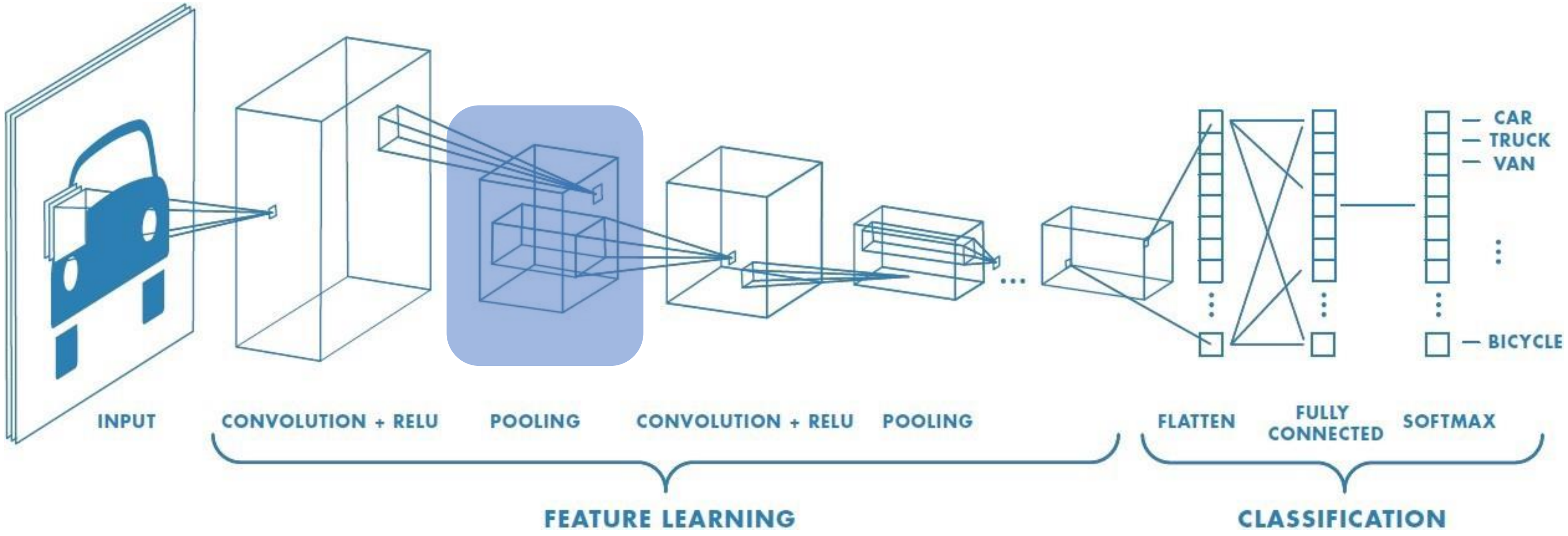
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>>;
```

MAX POOLING
2x2 window
2x2 stride



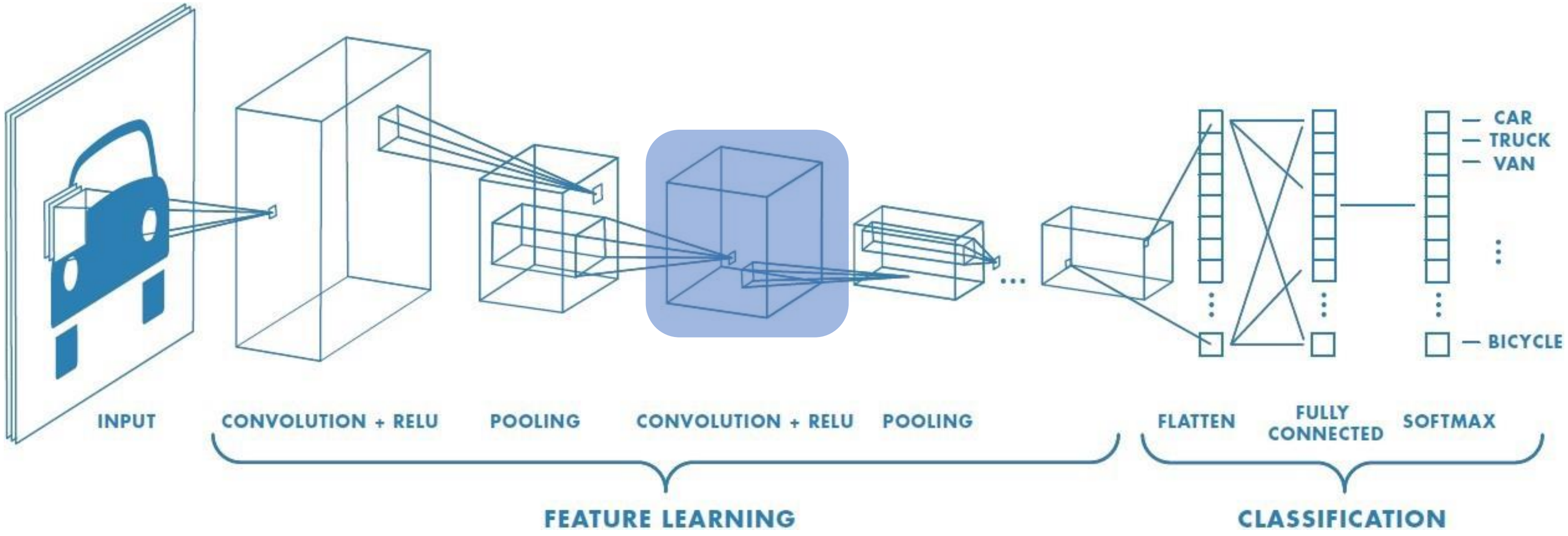
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>>;
```

16 conv. filters
5x5 filter size
1x1 stride
+ReLU



A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

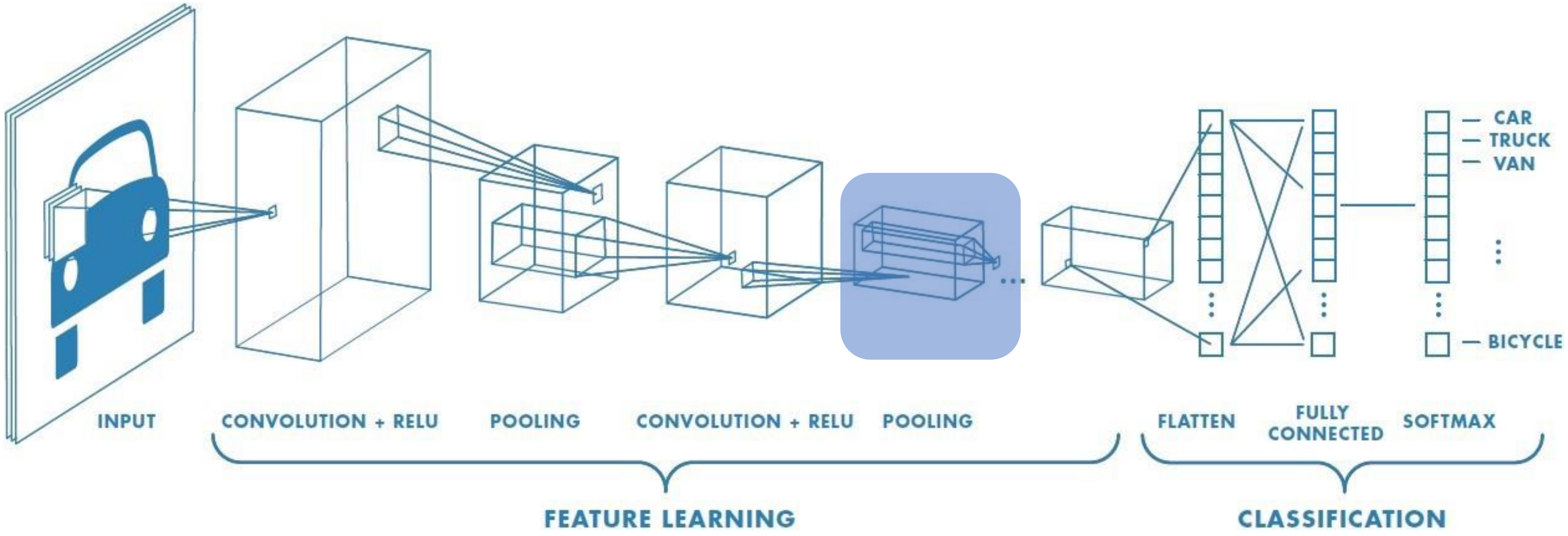
EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
                            fc<10,
                            relu<fc<84,
                            relu<fc<120,
                            max_pool<2,2,2,2,relu<con<16,5,5,1,1,
                            max_pool<2,2,2,2,relu<con<6,5,5,1,1,
                            input<matrix<unsigned char>>
                            >>>>>>>>>>>;
```

MAX POOLING
2x2 window
2x2 stride



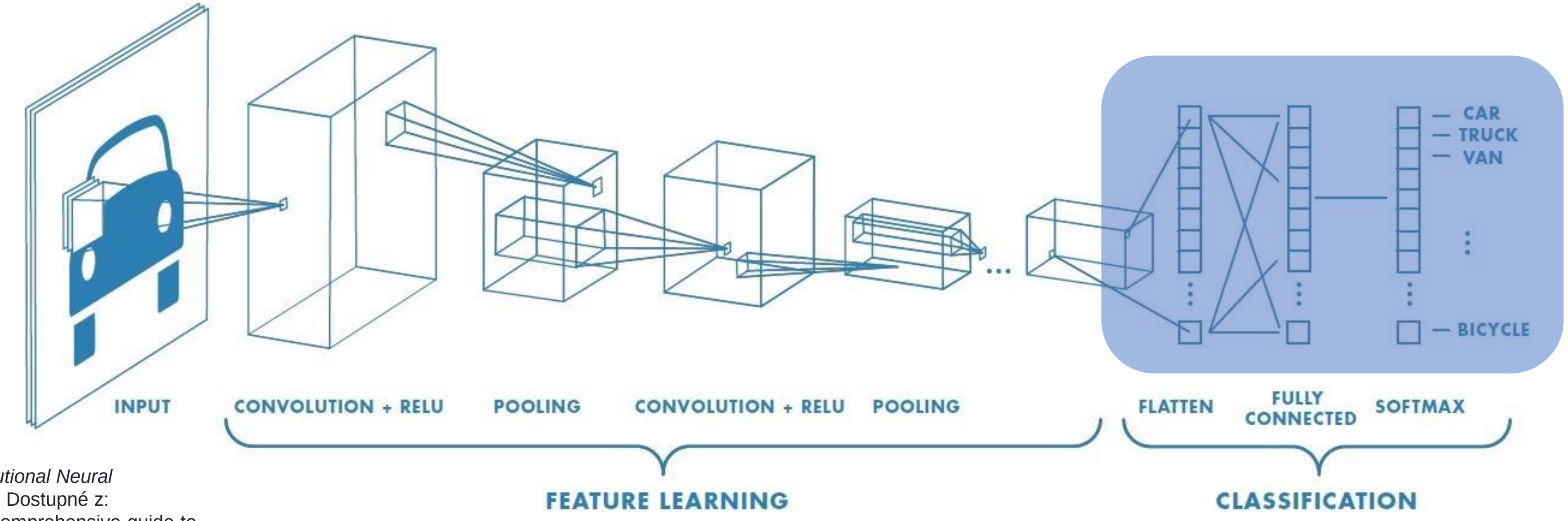
A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
using net_type = loss_multiclass_log<
    fc<10,
    relu<fc<84,
    relu<fc<120,
    max_pool<2,2,2,2,relu<con<16,5,5,1,1,
    max_pool<2,2,2,2,relu<con<6,5,5,1,1,
    input<matrix<unsigned char>>
    >>>>>>>>>>>;
```

Fully connected layer
120 neurons
84 neurons
10 outputs/classes
multiclass
classification



A Comprehensive Guide to Convolutional Neural Networks [online]. [cit. 2020-01-25]. Dostupné z: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
// network instance
net_type net;

// mini-batch stochastic gradient descent
//dnn_trainer<net_type> trainer(net, sgd(), {0,1}); //{0,1} - will use two GPU
dnn_trainer<net_type> trainer(net);
trainer.set_learning_rate(0.01);
trainer.set_min_learning_rate(0.0001);
trainer.set_mini_batch_size(160);
trainer.set_iterations_without_progress_threshold(500);
trainer.set_max_num_epochs(100);
trainer.be_verbose();
//train
trainer.train(train_images, train_labels);
// save
serialize("LeNet.dat") << net;
```

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Recognition step CNNs (Dlib)

```
//Load image using OpenCV
Mat frame;
frame = imread( "my_img.png", 1 );
cvtColor( frame, frame, COLOR_BGR2GRAY );
medianBlur(frame, frame, 5);

 //OpenCV Mat to Dlib
cv_image<unsigned char> cimg(frame);
matrix<unsigned char> dlibFrame = dlib::mat(cimg);

//prediction using CNN
unsigned long predict_label = net(frame);
```

Dnn_introduction_ex.cpp [online]. [cit. 2020-01-25]. Dostupné z: http://dlib.net/dnn_introduction_ex.cpp.html

Checking the driver state

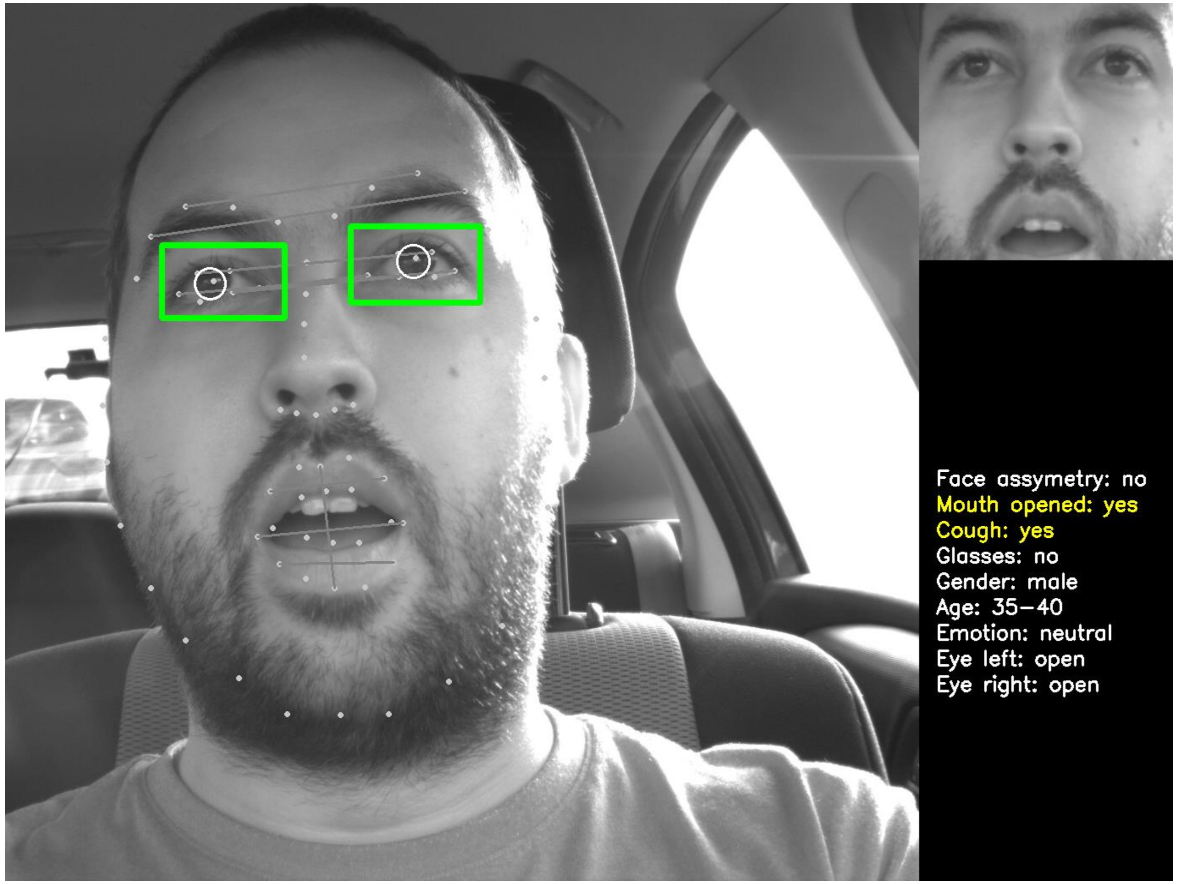
The contemporary level achieved in the area of hardware and software makes it possible to create the software for checking the driver state and his ability to drive the car. For monitoring the driver, the methods based on detecting various facial parts can be used. For obtaining a more complete information about driver's behaviour (e.g. fatigue, drowsiness, asthma, heart attack), a 3D model of the whole driver body can also be used.

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

Checking the driver state

The particular face parts and their states can be used as the features for detecting certain health problems. All these detected properties of human face may be used as features for recognizing certain health problems of the driver.



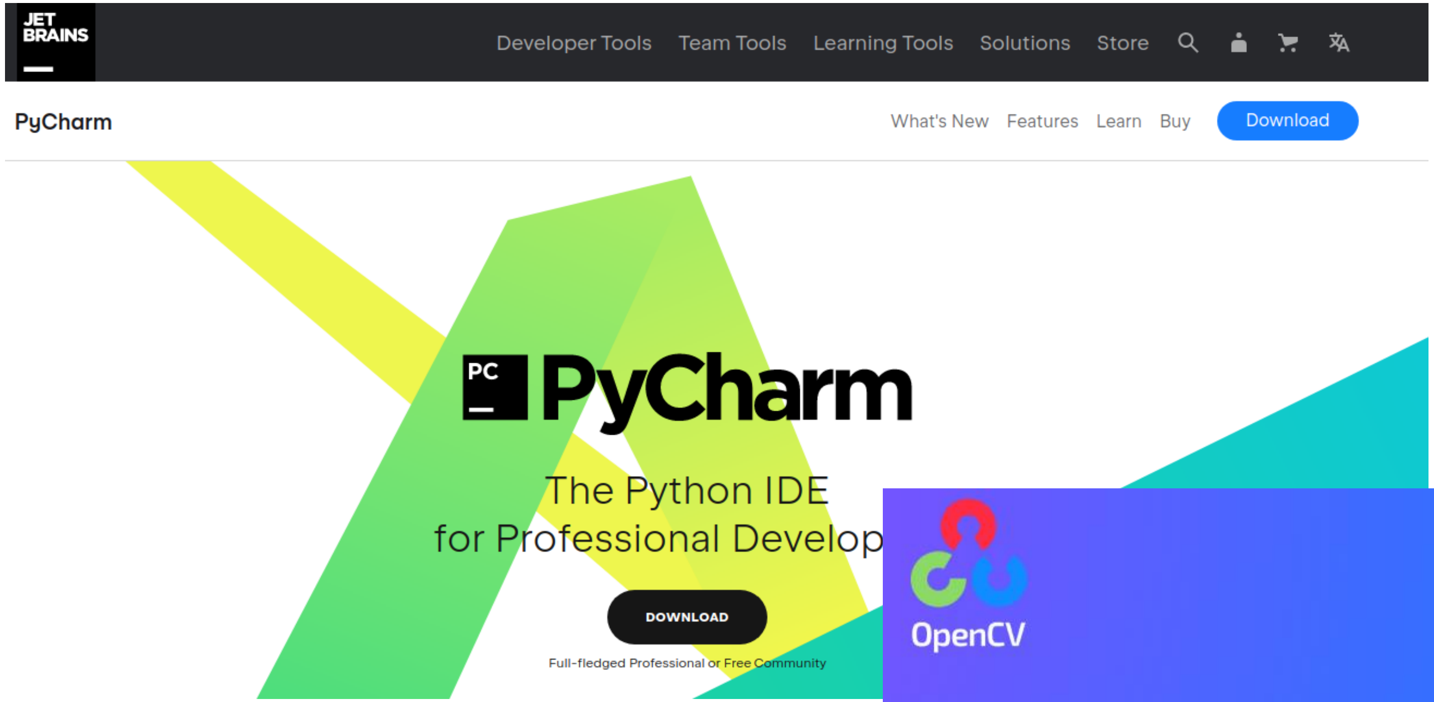
Checking the driver state

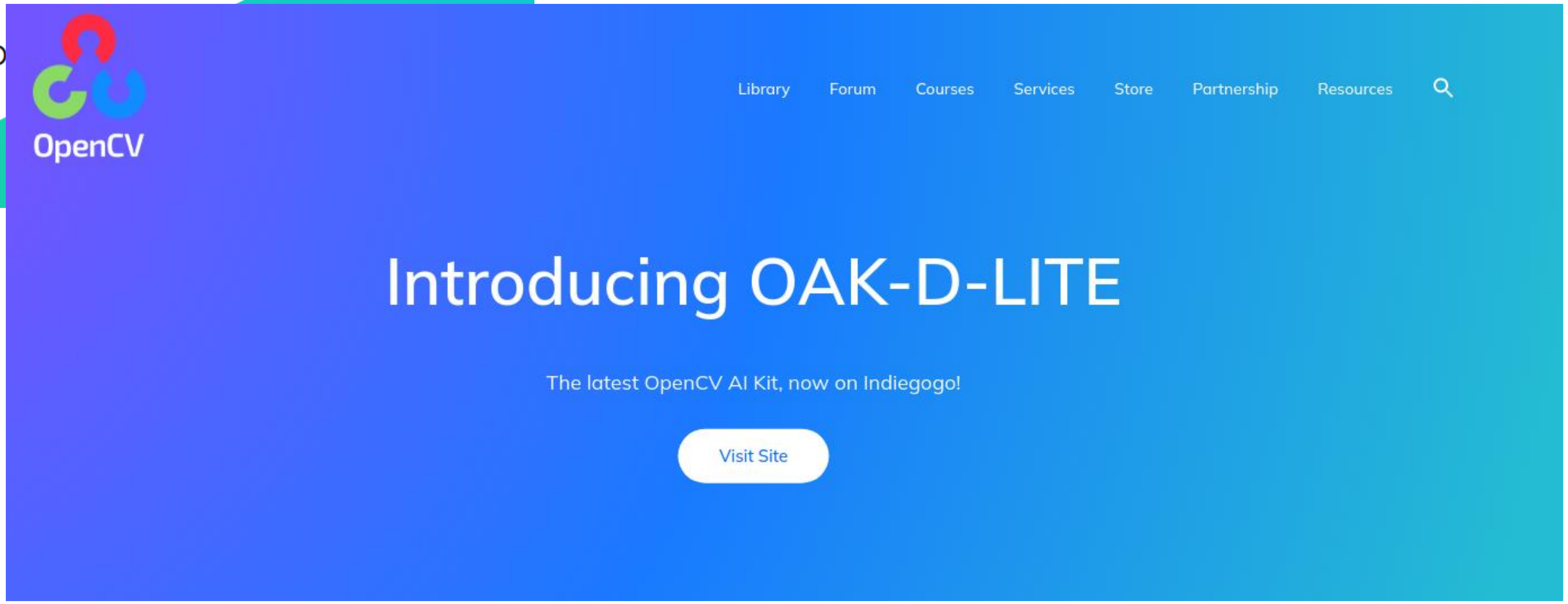
Openpose Library

Z. Cao, T. Simon, S. Wei and Y. Sheikh, "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1302-1310.

Practical Exercises

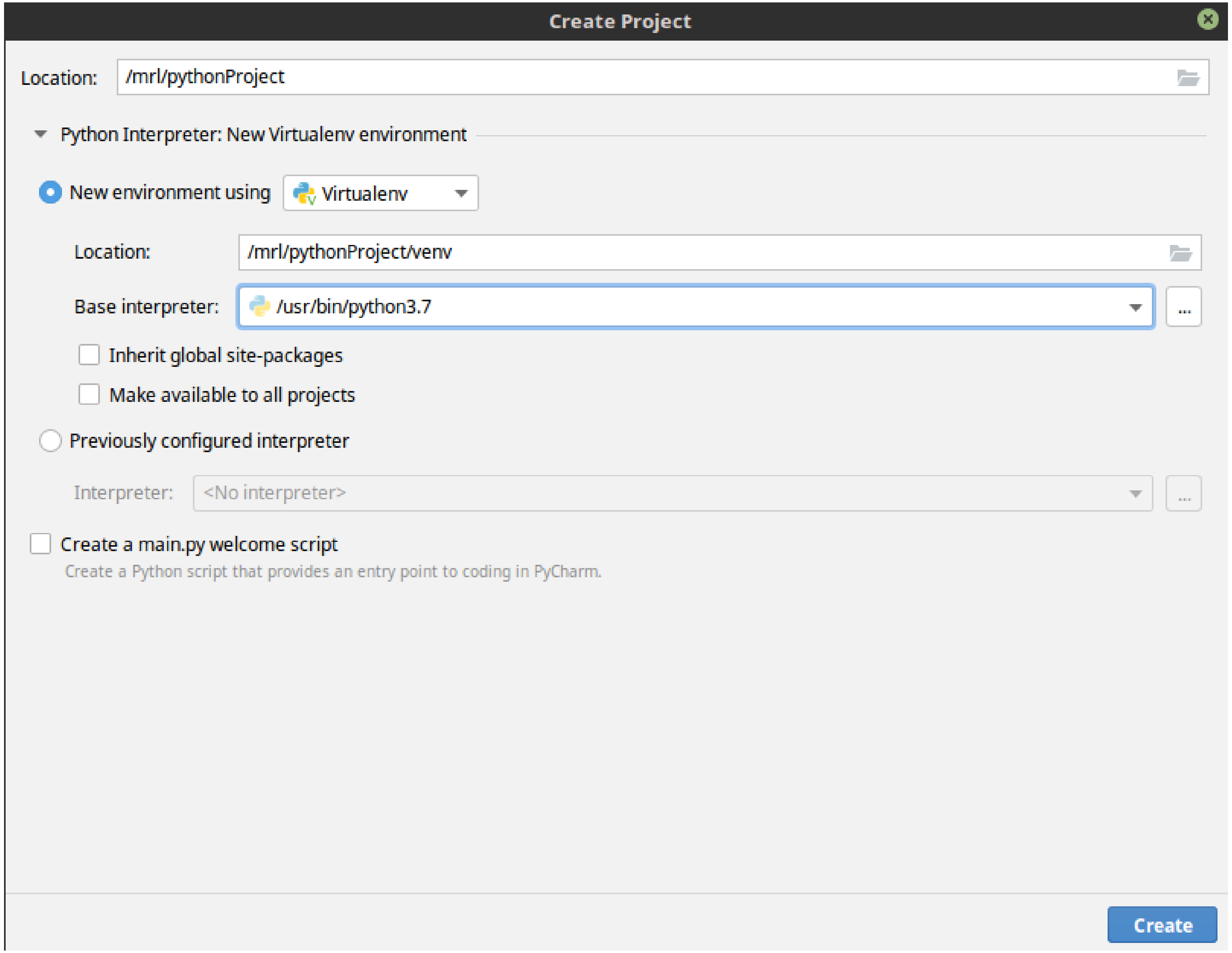
Opencv + Python





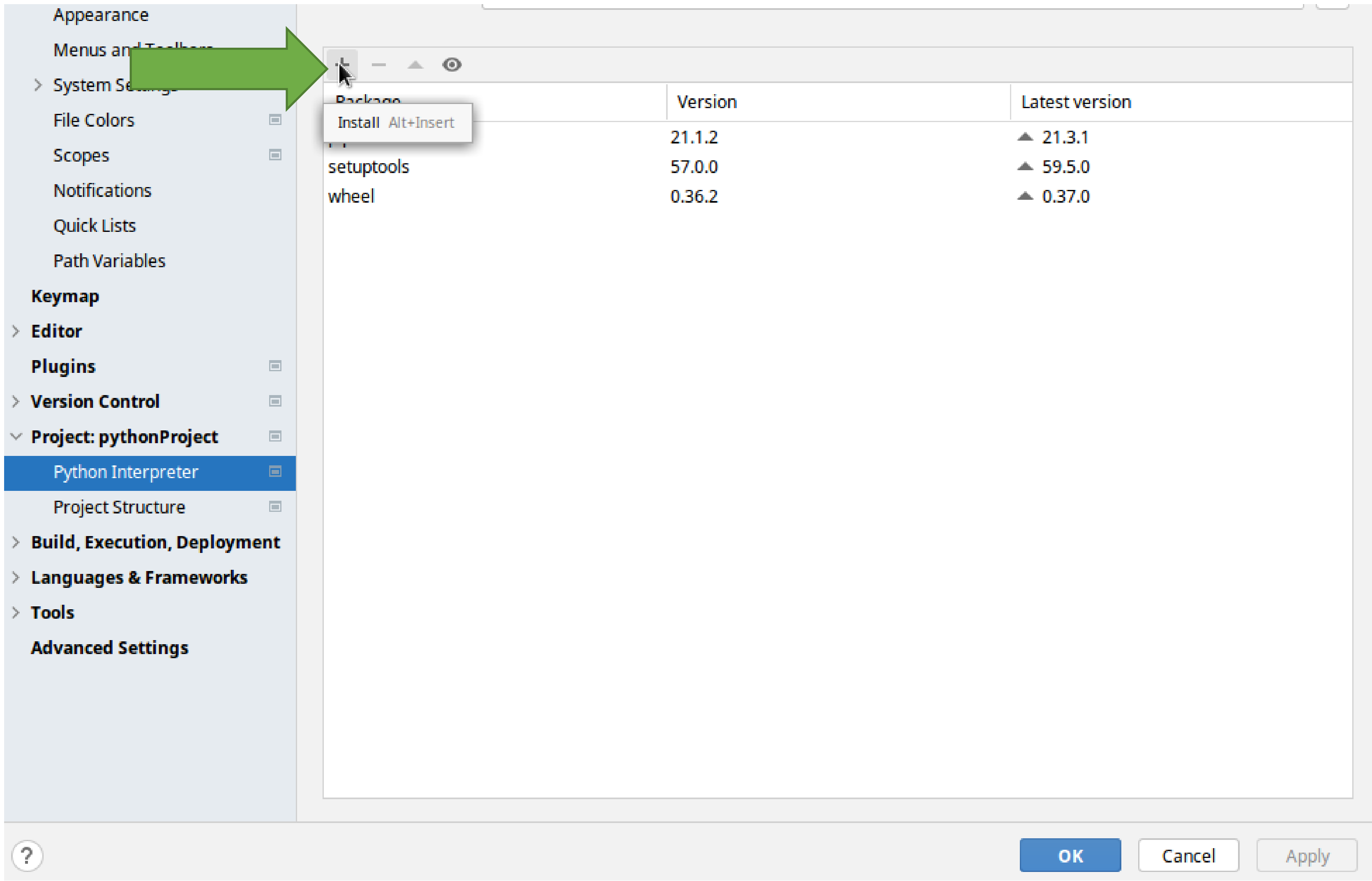
python-3.7.9-amd64
(OK with Dlib)

Opencv + Python

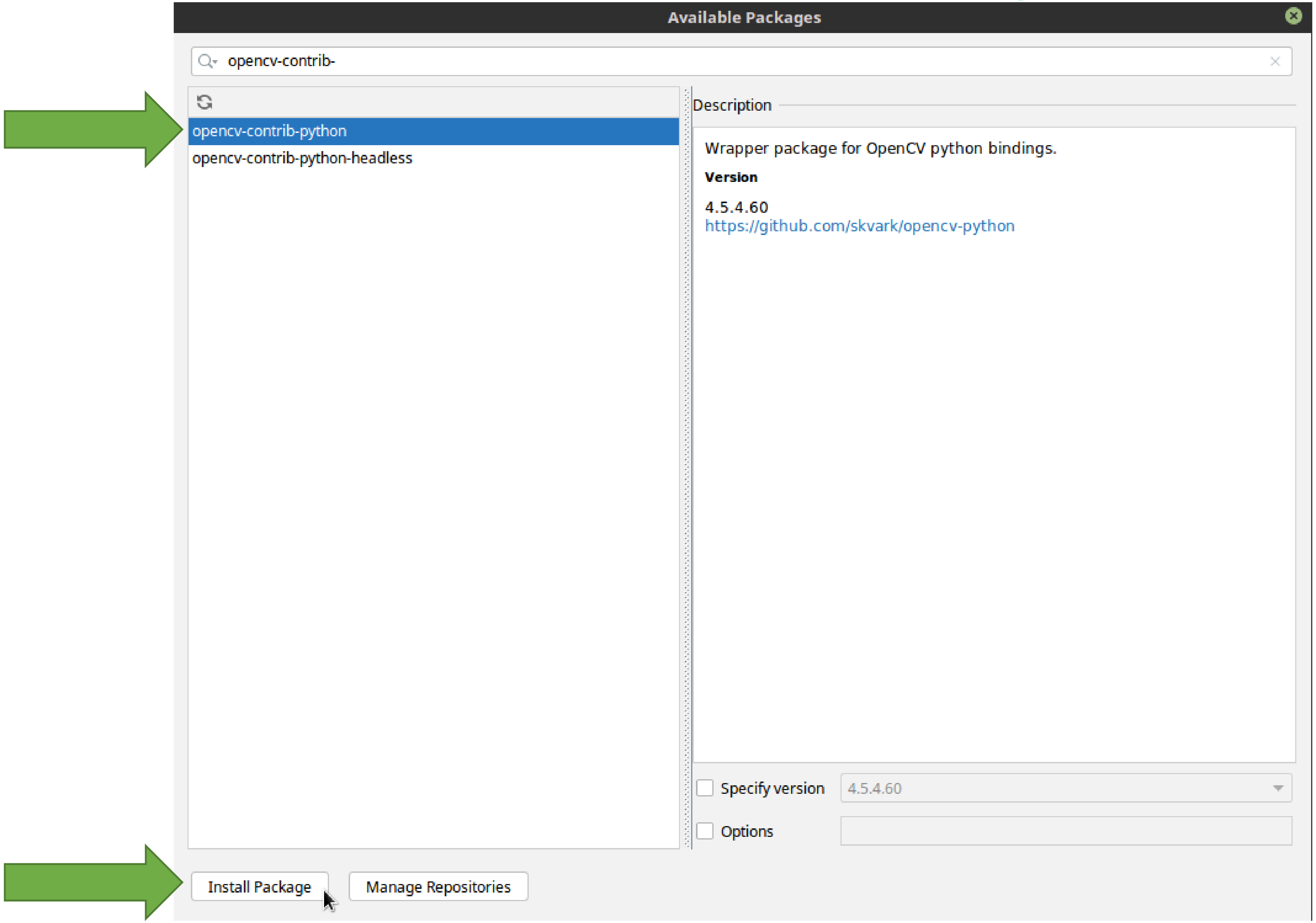


Opencv + Python

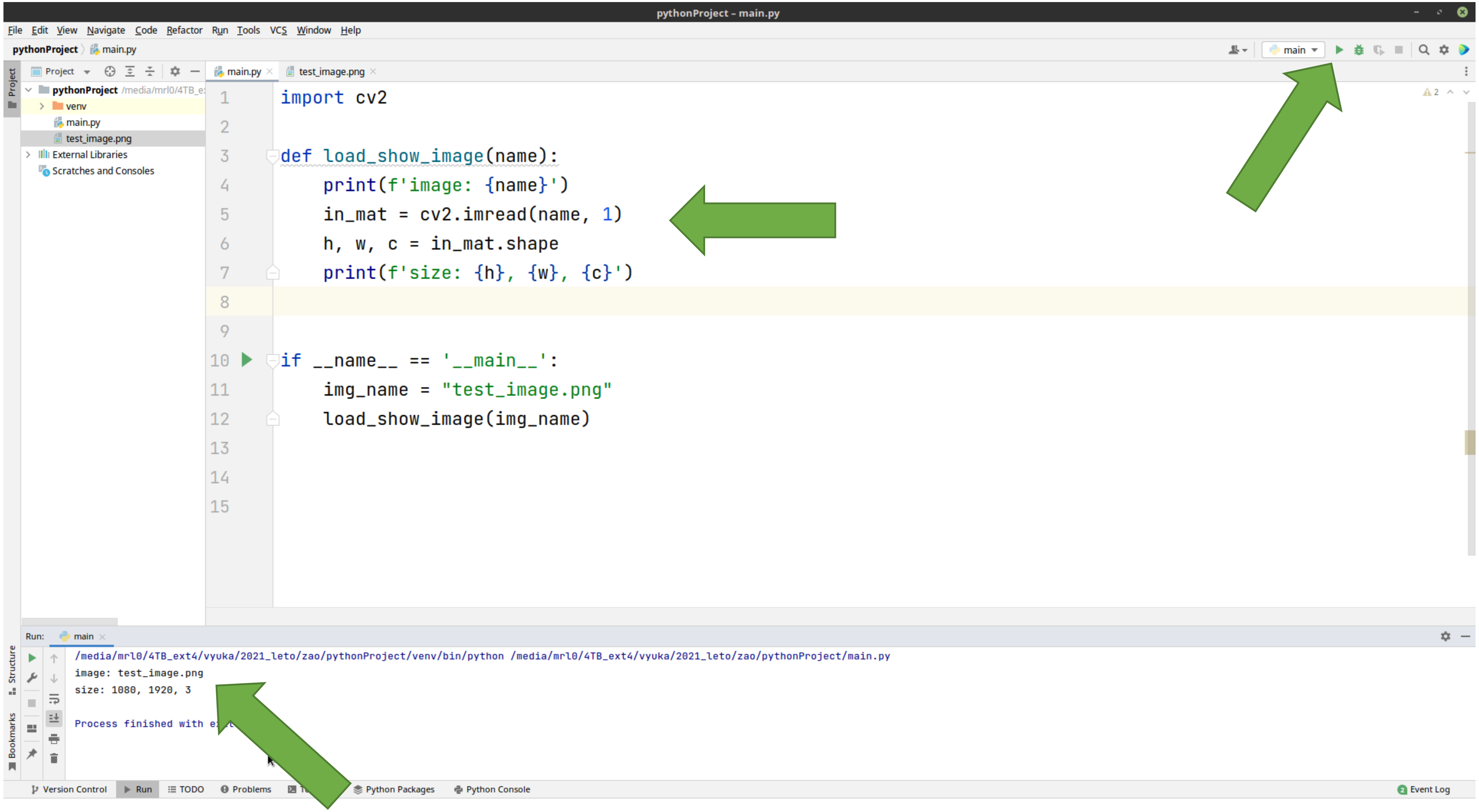
Ctrl+Alt+S



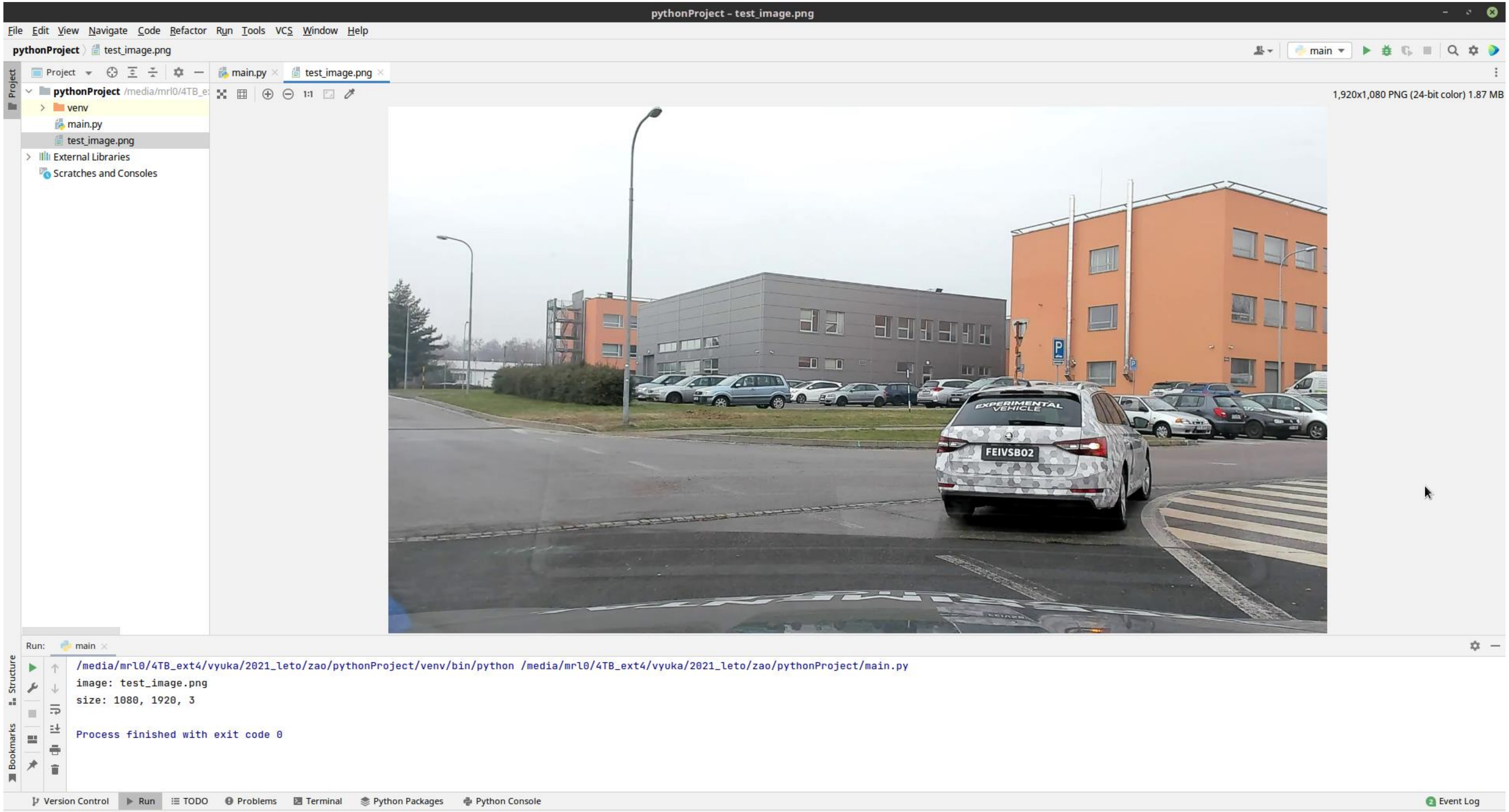
Opencv + Python



Opencv + Python



Opencv + Python



EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Opencv + Python

pythonProject – main.py

```python
import cv2


def load_show_image(name):
    print(f'image: {name}')
    in_mat = cv2.imread(name, 1)
    h, w, c = in_mat.shape
    print(f'size: {h}, {w}, {c}')

    win_name = "image"
    cv2.namedWindow(win_name, 0)
    cv2.resizeWindow(win_name, int(w/3), int(h/3))
    cv2.imshow(win_name, in_mat)
    cv2.waitKey(0)


if __name__ == '__main__':
    img_name = "test_image.png"
    load_show_image(img_name)
```

image

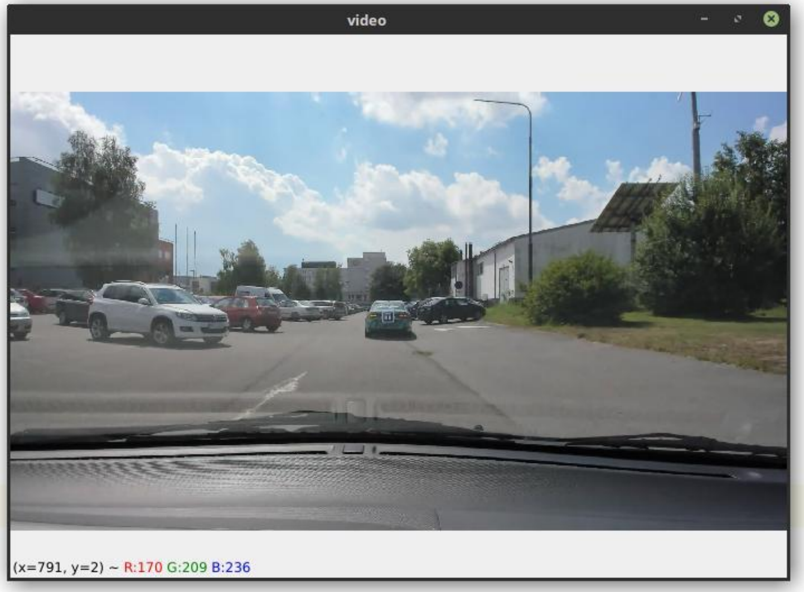
(x=1206, y=1059) ~ R:140 G:149 B:151

```
/media/mrl0/4TB_ext4/vyuka/2021_leto/zao/pythonProject/venv/bin/python /media/mrl0/4TB_ext4/vyuka/2021_leto/zao/pythonProject/main.py
image: test_image.png
size: 1080, 1920, 3
```

Opencv + Python

pythonProject – main.py

```python
def load_show_video(name):
    video_cap = cv2.VideoCapture(name)
    win_video = "video"
    cv2.namedWindow(win_video, 0)
    while True:
        ret, frame = video_cap.read()
        if ret is True:
            cv2.imshow(win_video, frame)
            if cv2.waitKey(1) == ord('q'):
                break


if __name__ == '__main__':
    img_name = "test_image.png"
    vid_name = "test_video.mp4"
    #load_show_image(img_name)
    load_show_video(vid_name)
```




```
QApplication: invalid style override 'gtk' passed, ignoring it.
    Available styles: Windows, Fusion
```

Opencv + Python

```python
import cv2


def convert_video_to_gray(in_video, out_video):
    video_cap = cv2.VideoCapture(in_video)
    # if not video_cap.isOpened():
    if video_cap.isOpened() == False:
        print("video_cap: False")

    fps = video_cap.get(cv2.CAP_PROP_FPS)
    frame_count = video_cap.get(cv2.CAP_PROP_FRAME_COUNT)
    frame_width = video_cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    frame_height = video_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

    print(f'fps: {fps}, frame_count: {frame_count}')
    print(f'frame_width: {frame_width}, frame_height: {frame_height}')


def main():
    in_video = "test_video.mp4"
    out_video = "test_video_gray.mp4"
    convert_video_to_gray(in_video, out_video)


if __name__ == "__main__":
    main()
```

Read video properties

```
/media/mrl/SSD_500GB/VSB/vyuka_2021_letni/PytorchProjects/venv_zao/bin/python /media/mrl/SSD_500GB/VSB/vyuka_2021_letni/PytorchProjects/01_zao_video/main.py
fps: 24.0, frame_count: 6748.0
frame_width: 1920.0, frame_height: 1080.0

Process finished with exit code 0
```

Opencv + Python

```python
def convert_video_to_gray(in_video, out_video):
    video_cap = cv2.VideoCapture(in_video)
    # if not video_cap.isOpened():
    if video_cap.isOpened() == False:
        print("video_cap: False")

    fps = video_cap.get(cv2.CAP_PROP_FPS)
    frame_count = video_cap.get(cv2.CAP_PROP_FRAME_COUNT)
    frame_width = video_cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    frame_height = video_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

    print(f'fps: {fps}, frame_count: {frame_count}')
    print(f'frame_width: {frame_width}, frame_height: {frame_height}')

    frame_size = (int(frame_width), int(frame_height))
    video_writer = cv2.VideoWriter(out_video,
                                   cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'),
                                   fps,
                                   frame_size,
                                   False)
    frame_num = 0
    while (video_cap.isOpened()):
        ret, frame = video_cap.read()
        if ret:
            frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            video_writer.write(frame_gray)
            frame_num = frame_num + 1
            print("frame_num: ", frame_num)
```

Save video

◆ VideoWriter() [2/3]

```
cv::VideoWriter::VideoWriter ( const String &  filename,
                               int             fourcc,
                               double          fps,
                               Size            frameSize,
                               bool            isColor = true
                             )
```

Python:

```
cv.VideoWriter(                                                      ) -> <VideoWriter object>
cv.VideoWriter( filename, fourcc, fps, frameSize[, isColor]          ) -> <VideoWriter object>
cv.VideoWriter( filename, apiPreference, fourcc, fps, frameSize[, isColor] ) -> <VideoWriter object>
```

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

filename	Name of the output video file.
fourcc	4-character code of codec used to compress the frames. For example, VideoWriter::fourcc('P','I','M','1') is a MPEG-1 codec, VideoWriter::fourcc('M','J','P','G') is a motion-jpeg codec etc. List of codes can be obtained at Video Codecs by FOURCC page. FFMPEG backend with MP4 container natively uses other values as fourcc code: see ObjectType, so you may receive a warning message from OpenCV about fourcc code conversion.
fps	Framerate of the created video stream.
frameSize	Size of the video frames.
isColor	If it is not zero, the encoder will expect and encode color frames, otherwise it will work with grayscale frames (the flag is currently supported on Windows only).

Tips:

- With some backends `fourcc=-1` pops up the codec selection dialog from the system.
- To save image sequence use a proper filename (eg. `img_%02d.jpg`) and `fourcc=0` OR `fps=0`. Use uncompressed image format (eg. `img_%02d.BMP`) to save raw frames.
- Most codecs are lossy. If you want lossless video file you need to use a lossless codecs (eg. FFMPEG FFV1, Huffman HFYU, Lagarith LAGS, etc...)
- If FFMPEG is enabled, using `codec=0; fps=0;` you can create an uncompressed (raw) video file.

MatchTemplate [online]. [cit. 2020-01-25]. Dostupné z:
https://docs.opencv.org/4.x/df/dfb/group__imgproc__object.html#ga586ebfb0a7fb604b35a23d85391329be

Template Matching

- basic (simple) method for object localization
- we need the template and the source (input) image
- we need compare a template vs. overlapped image regions

template

source

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

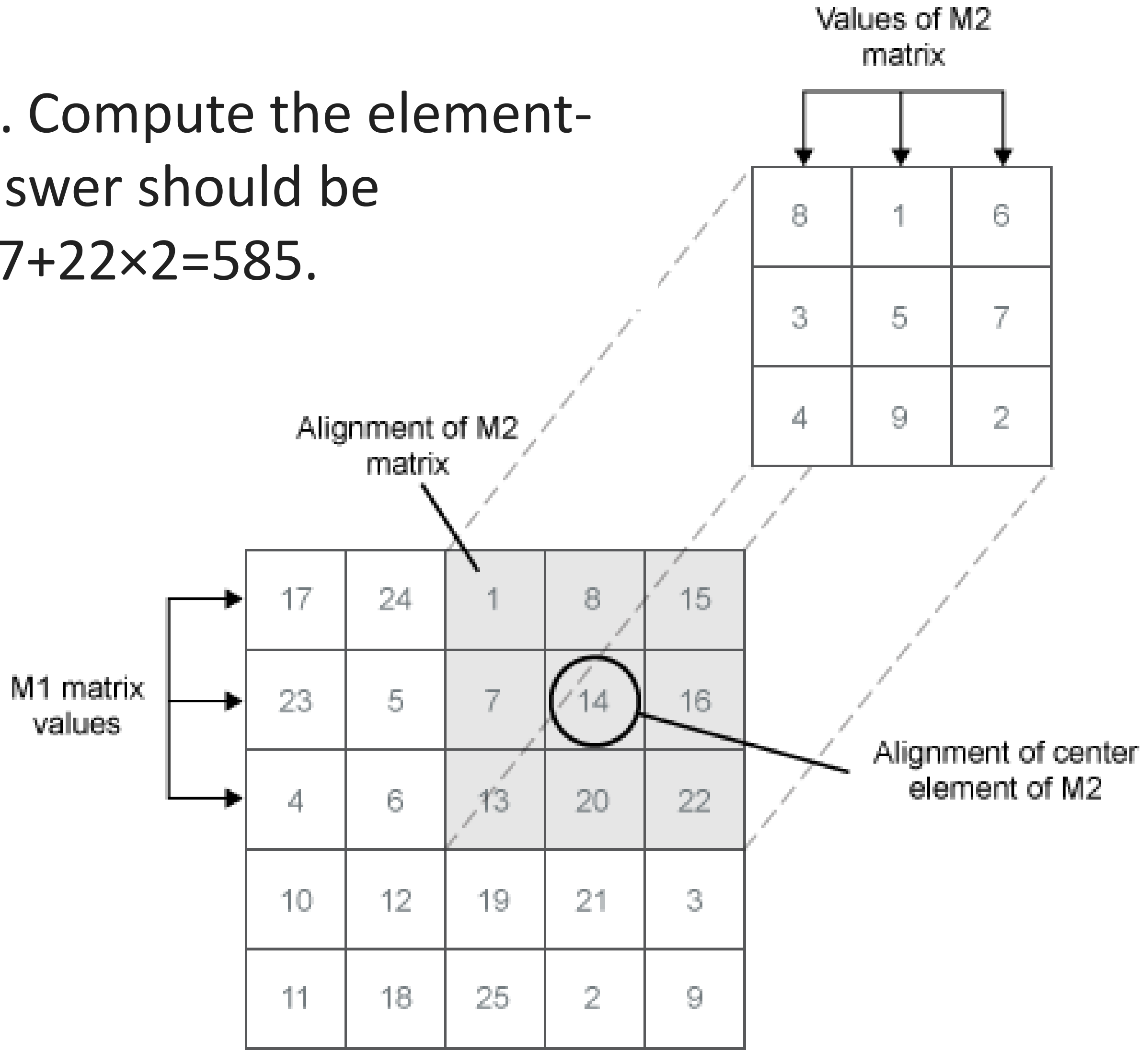
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Template Matching

Template Matching

Template Matching

Now M2 is on top of the matrix M1(1:3,3:5). Compute the element-by-element products and sum them. The answer should be 1×8+7×3+13×4+8×1+14×5+20×9+15×6+16×7+22×2=585.



2-D cross-correlation [online]. [cit. 2020-01-25]. Dostupné z: https://www.mathworks.com/help/signal/ref/xcorr2.html

Template Matching

- several comparison methods are implemented in OpenCV

- TM_SQDIFF $$R(x,y) = \sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2$$

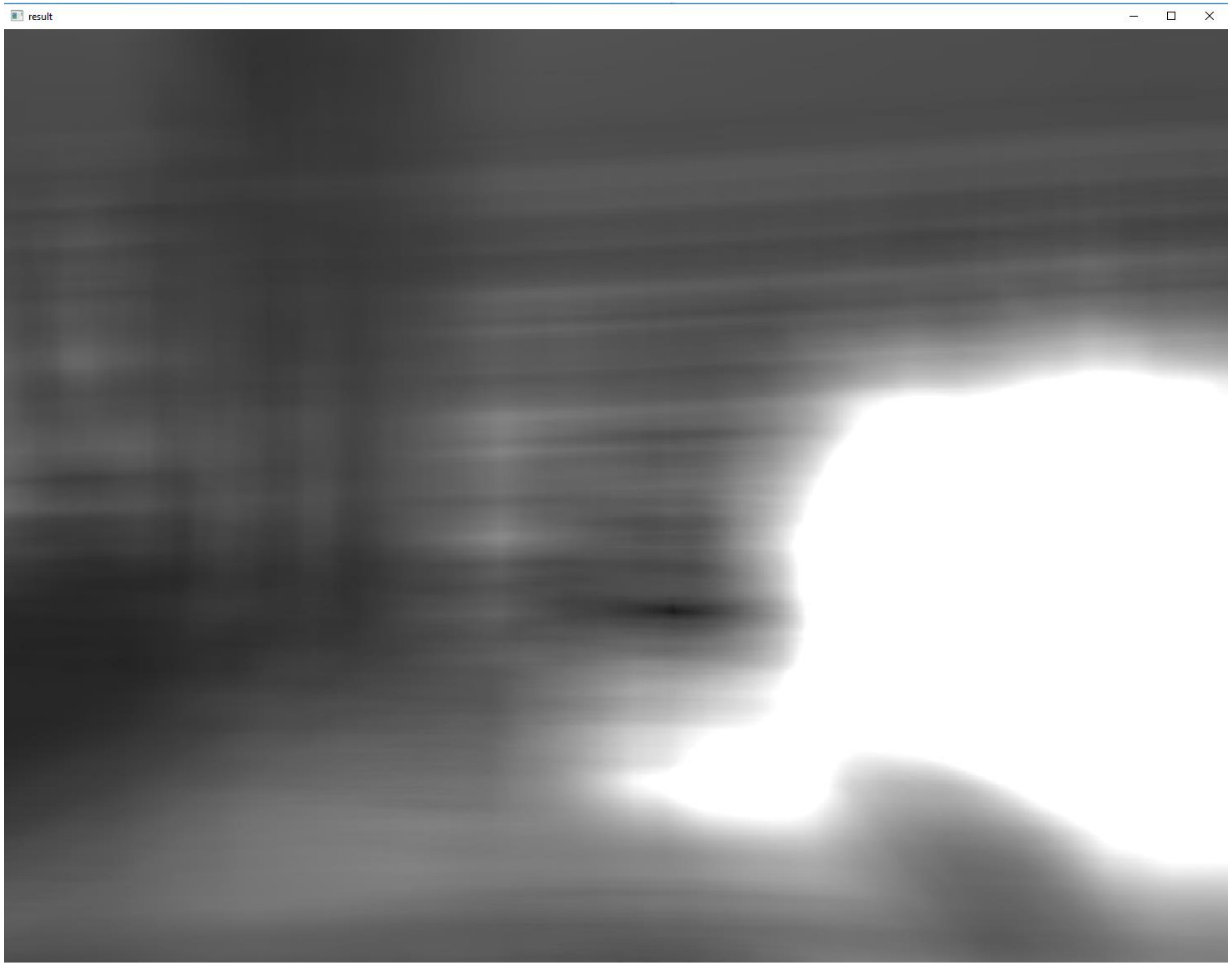
- TM_CCORR $$R(x,y) = \sum_{x',y'} (T(x',y') \cdot I(x+x',y+y'))$$

- there are also the normalized versions: https://docs.opencv.org/4.x/df/dfb/group__imgproc__object.html#ga586ebfb0a7fb604b35a23d85391329be

- after the function finishes the comparison, the best matches can be found as global minimums (when **TM_SQDIFF** was used) or maximums (when **TM_CCORR** or **TM_CCOEFF** was used) using the **minMaxLoc** function

Template Matching

TM_SQDIFF_NORMED



Template Matching

TM_CCORR_NORMED

Template Matching

- Example – preparing phase:

```python
def template_simple():

    cv2.namedWindow("src_mat", 0)
    cv2.namedWindow("temp_mat", 0)

    src_mat = cv2.imread("source.png")
    temp_mat = cv2.imread("template.png")

    src_mat = cv2.cvtColor(src_mat, cv2.COLOR_BGR2GRAY)
    temp_mat = cv2.cvtColor(temp_mat, cv2.COLOR_BGR2GRAY)

    src_mat_h = src_mat.shape[0]
    src_mat_w = src_mat.shape[1]
    print("src w h", src_mat_w, src_mat_h)

    temp_mat_h = temp_mat.shape[0]
    temp_mat_w = temp_mat.shape[1]
    print("temp w h", temp_mat_w, temp_mat_h)

    cv2.imshow("src_mat", src_mat)
    cv2.imshow("temp_mat", temp_mat)
    cv2.waitKey(0)
```

src_mat

temp_mat


```
src w h 1600 900
temp w h 131 136
```

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

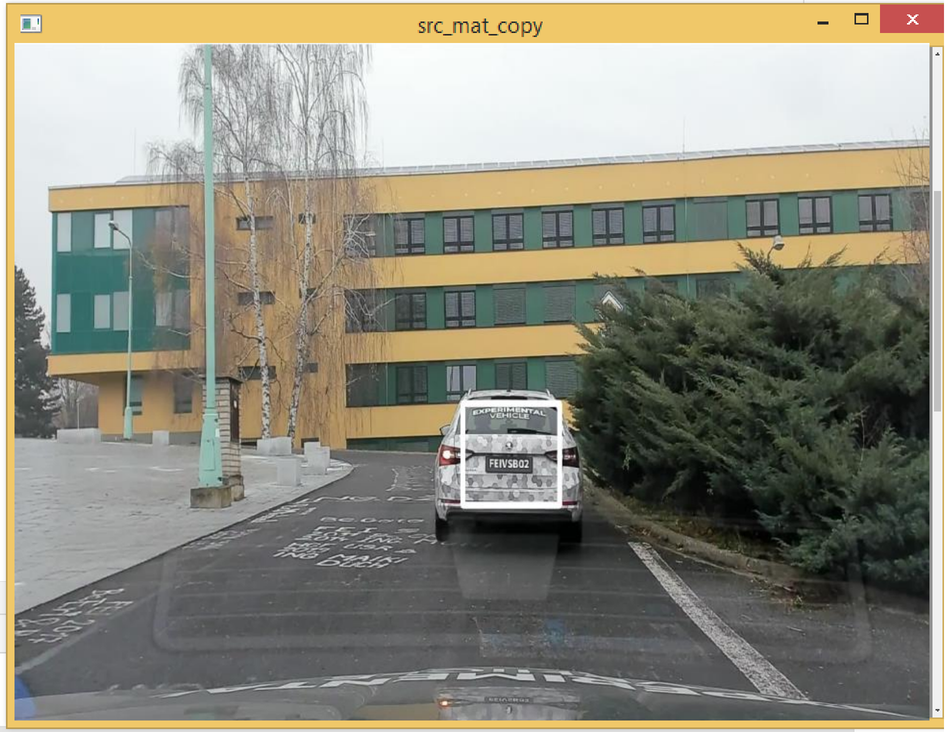
VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Template Matching

- Example – localization phase:

```python
    result = cv2.matchTemplate(src_mat, temp_mat, cv2.TM_CCORR_NORMED)

    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
    print("max_val", round(max_val, 2))

    top_left = max_loc
    bottom_right = (top_left[0] + temp_mat_w, top_left[1] + temp_mat_h)

    center_x = int((top_left[0] + bottom_right[0])/2)
    center_y = int((top_left[1] + bottom_right[1])/2)

    src_mat_copy = src_mat.copy()
    cv2.namedWindow("src_mat_copy", 0)

    cv2.rectangle(src_mat_copy, top_left, bottom_right, 255, 5)
    cv2.circle(src_mat_copy, (center_x, center_y), 5, 255, -1)

    cv2.imshow("src_mat_copy", src_mat_copy)
    cv2.waitKey(0)
```

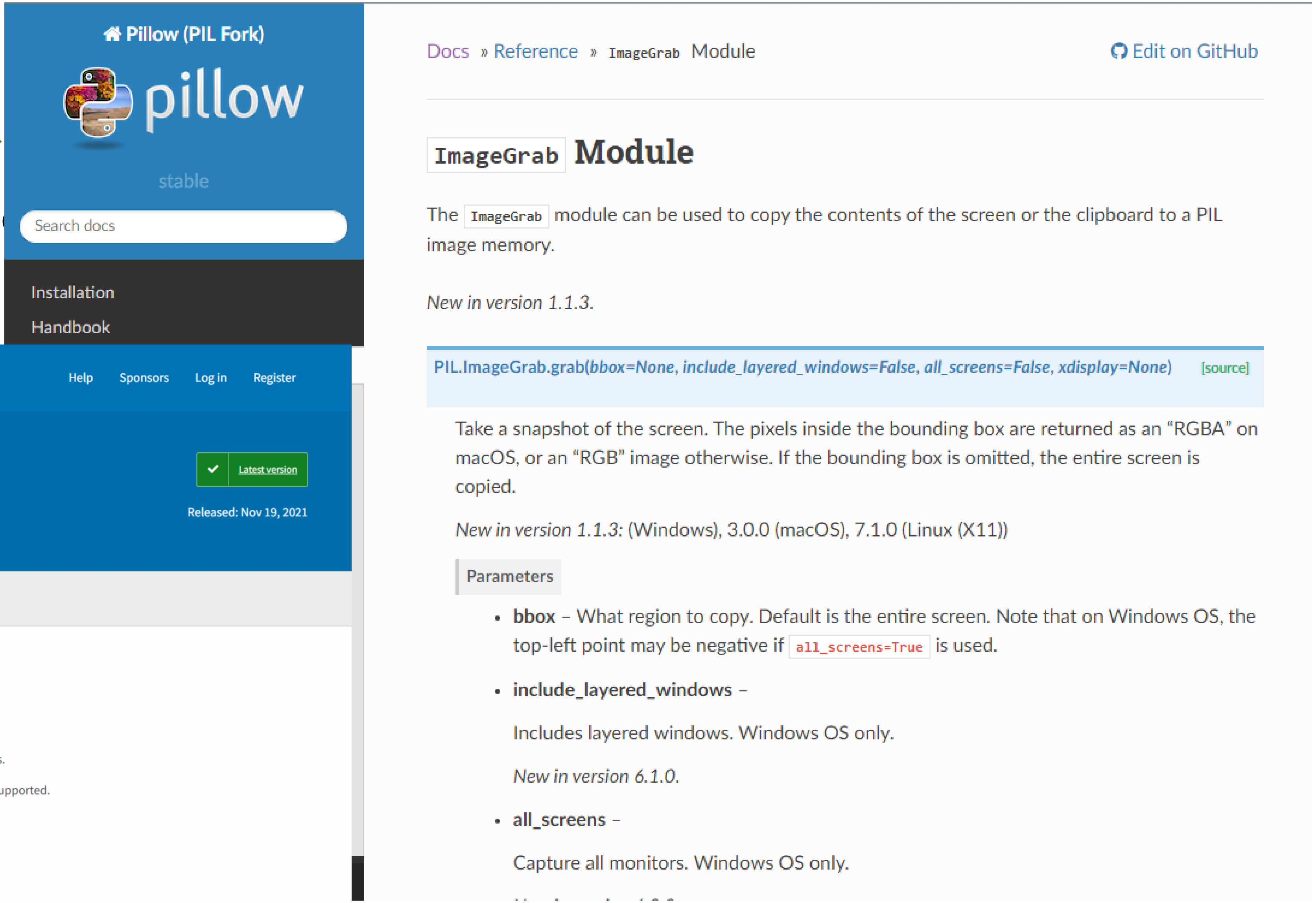
```
temp w h 131 136
max_val 1.0
```



Template Matching

- taking screenshots :

```
import cv2
from PIL import ImageGrab
import numpy as np
from pynput.mouse import Button, Controll
```


Pillow (PIL Fork)
pillow
stable
Search docs
Installation
Handbook

Docs » Reference » ImageGrab Module   Edit on GitHub

ImageGrab Module

The ImageGrab module can be used to copy the contents of the screen or the clipboard to a PIL image memory.

New in version 1.1.3.

PIL.ImageGrab.grab(*bbox=None, include_layered_windows=False, all_screens=False, xdisplay=None*) [source]

Take a snapshot of the screen. The pixels inside the bounding box are returned as an "RGBA" on macOS, or an "RGB" image otherwise. If the bounding box is omitted, the entire screen is copied.

New in version 1.1.3: (Windows), 3.0.0 (macOS), 7.1.0 (Linux (X11))

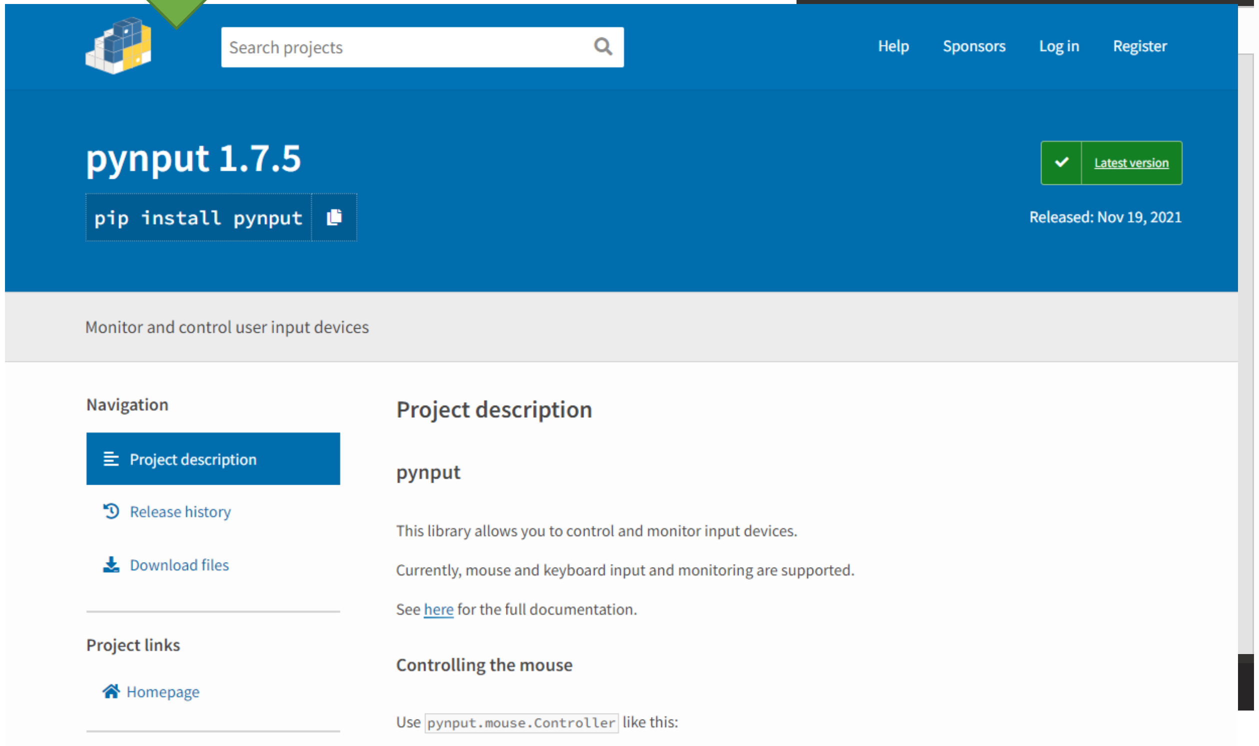
Parameters

- **bbox** – What region to copy. Default is the entire screen. Note that on Windows OS, the top-left point may be negative if `all_screens=True` is used.
- **include_layered_windows** –

  Includes layered windows. Windows OS only.

  New in version 6.1.0.
- **all_screens** –

  Capture all monitors. Windows OS only.


Search projects   Help Sponsors Log in Register

pynput 1.7.5
pip install pynput
Latest version
Released: Nov 19, 2021

Monitor and control user input devices

Navigation
Project description
Release history
Download files

Project links
Homepage

Project description

pynput

This library allows you to control and monitor input devices.

Currently, mouse and keyboard input and monitoring are supported.

See here for the full documentation.

Controlling the mouse

Use `pynput.mouse.Controller` like this:

Exercise – Parking

DETECTING FREE/OCCUPIED PLACES IN PARKING LOTS

Motivation:

The vehicle detection systems using images have been very useful in the recent years. Especially nowadays in the cities, the increasing number of vehicles brings a major problem. The car detection systems can be important, especially for drivers who are looking for vacant spaces in the parking lots, for traffic analysis, for intelligent scheduling, for smart cities and so on.

Input Data:

The training data with a basic template (Python/OpenCV) can be found in the following link:

http://mrl.cs.vsb.cz/data/vyuka/osaj/parking_template_python.zip

Exercise – Parking

description of template:

- training and testing data are in the “testImages” and “trainImages” folders
- each image is named as free_xx.png or full_xx.png (the name of the images represents the state of parking space)
- functions for loading training/testing images are already implemented - train_parking(), test_parking()
- **the training and prediction steps are missing** - You can use any available libraries to solve this detection task. The use of the provided main.cpp template is not required.

Exercise – Parking

Output:

If you successfully run the template, you obtain this output. It means that the accuracy of the detector is aprox. 32%. The accuracy is low because each parking space is labeled as occupied - line 82 in main.cpp. The goal is to implement better prediction approach.

```
Train OpenCV Start
Train images: 5656
Train labels: 5656
Train OpenCV End
Test OpenCV Start
testing num_right: 426
testing num_wrong: 918
testing accuracy:  0.316964
Test OpenCV End
```

Exercise – Parking

Hints:

Since we want to label each parking space as free (0) or occupied (1), this recognition problem can be solved using classical binary classifiers (SVM, neural networks). To train the classifiers, you can use the provided training data in the “trainImages” folder. As the input for the classifiers, you can use the whole image or you can use feature extraction approaches (e.g. histograms of oriented gradients, local binary patterns).
Alternatively, you can skip the training process and use simple color or gradient information for example. In that case, you can use only the test_parking() function without the training.
The provided template is based on the **OpenCV** library https://opencv.org/
Installation in Linux: https://www.learnopencv.com/install-opencv3-on-ubuntu/
Installation in Windows: https://www.learnopencv.com/install-opencv3-on-windows/
Installation in MacOS: https://www.learnopencv.com/install-opencv3-on-macos/
Simple install for Windows without cmake using NuGet:

http://funvision.blogspot.com/2017/04/simple-install-opencv-visual-studio.html
https://www.nuget.org/packages/opencv.win.native/320.1.1-vs141

Exercise – Parking

Hints:

Alternatively, you can install OpenCV from the Ubuntu or Debian repository:

    sudo apt-get install libopencv-dev python3-opencv

You can find the several tutorials in the following link: https://docs.opencv.org/3.4.2/d9/df8/tutorial_root.html

Dlib library represents another option how to solve this detection problem

- Installation in Windows https://www.learnopencv.com/install-dlib-on-windows/
- Installation in Linux https://www.learnopencv.com/install-dlib-on-ubuntu/
- Installation in MacOS: https://www.learnopencv.com/install-dlib-on-macos/
- You can follow this tutorial: http://dlib.net/dnn_introduction_ex.cpp.html

You can also use **Keras, Caffe, TensorFlow**, etc.

Examples of modern methods for object detection

- **R-CNN (Region based CNN, Faster R-CNN, Mask R-CNN)**
- **YOLO - You Only Look Once**
- **SSD – Single Shot Detector**

https://arxiv.org/abs/1512.02325

https://pytorch.org/vision/main/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn.html

https://github.com/ultralytics/yolov5

- **EXAMPLE Faster R-CNN – PyTorch**

https://pytorch.org/vision/main/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn.html

```
def main():

    cv2.namedWindow("detection", 0)
    print("main")

    test_images = [img for img in glob.glob("test_images/*.jpg")]
    test_images.sort()

    model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model.eval().to(device)

    transformRCNN = transforms.Compose([
        transforms.ToTensor(),
    ])
```

https://arxiv.org/abs/1506.01497
https://arxiv.org/abs/1703.06870

- **EXAMPLE Faster R-CNN – PyTorch**

coco_names = ['__background__', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table', 'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']

```
for img in test_images:
    one_img = cv2.imread(img)
    one_img_paint = one_img.copy()

    one_img_rgb = cv2.cvtColor(one_img, cv2.COLOR_BGR2RGB)
    img_pil = Image.fromarray(one_img_rgb)
    imageRCNN = transformRCNN(img_pil).to(device)
    imageRCNN = imageRCNN.unsqueeze(0)
    outputsRCNN = model(imageRCNN)
    pred_classes = [coco_names[i] for i in outputsRCNN[0]['labels'].cpu().numpy()]
    pred_scores = outputsRCNN[0]['scores'].detach().cpu().numpy()
    pred_bboxes = outputsRCNN[0]['boxes'].detach().cpu().numpy()

    print(pred_scores)
    print(pred_classes)
```

https://arxiv.org/abs/1506.01497
https://arxiv.org/abs/1703.06870