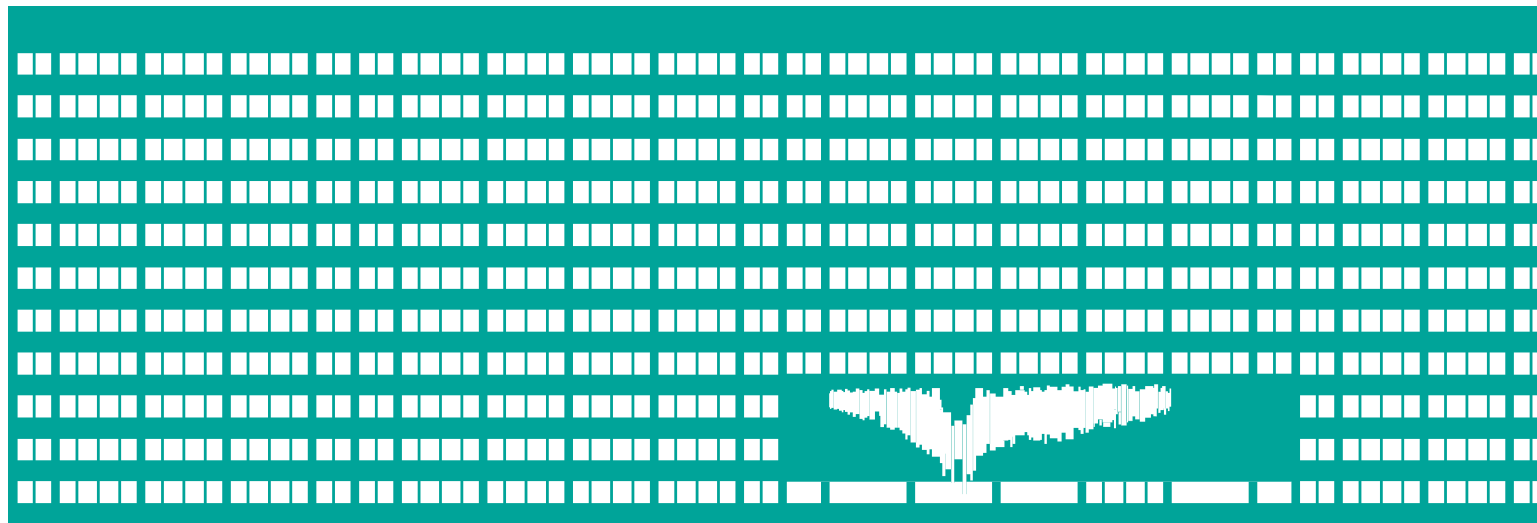




EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education



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



www.vsb.cz



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)



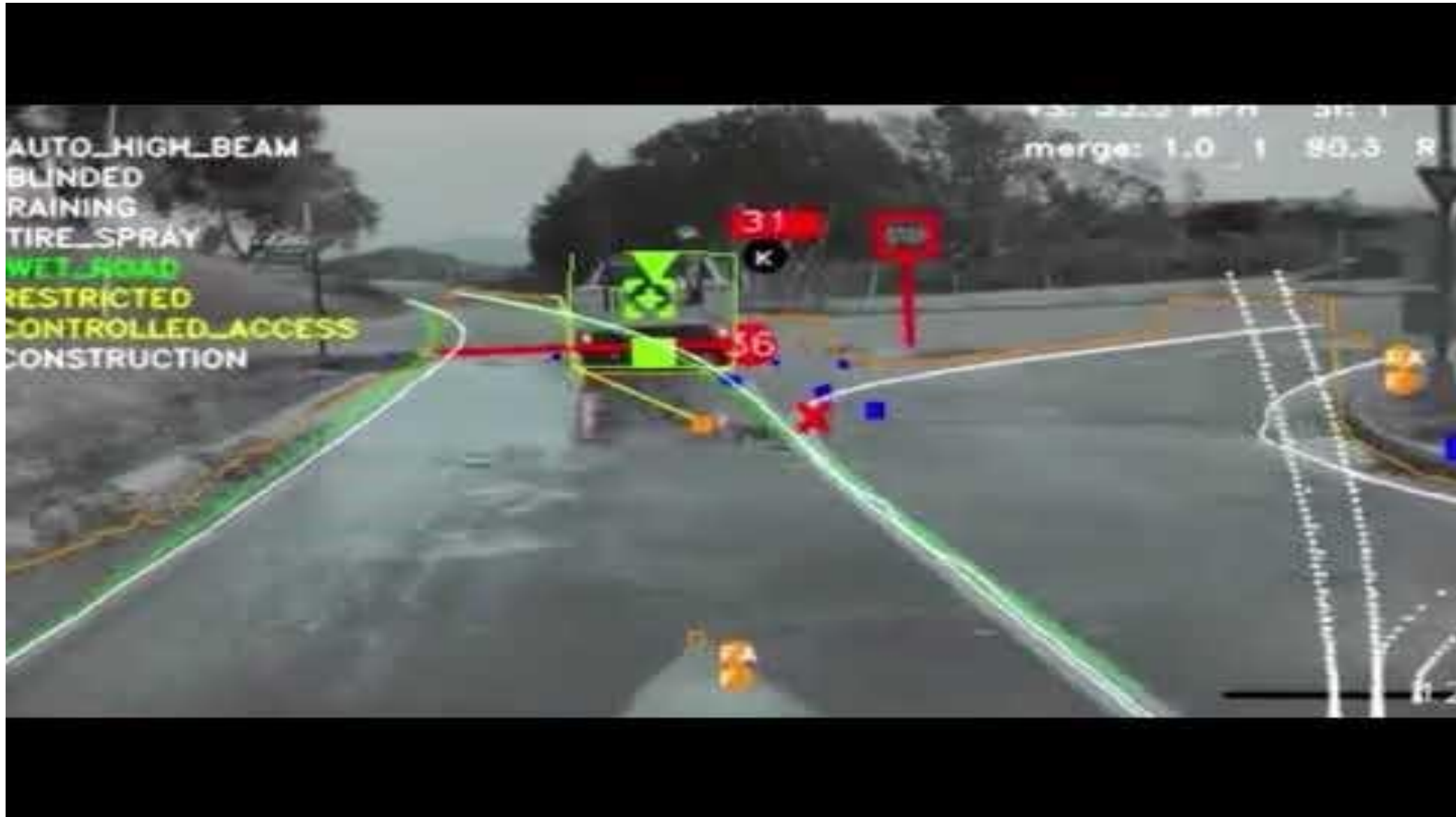


What Is AV (Autonomous Vehicle)?





What Tesla See





What Is AV (Autonomous Vehicle)?

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



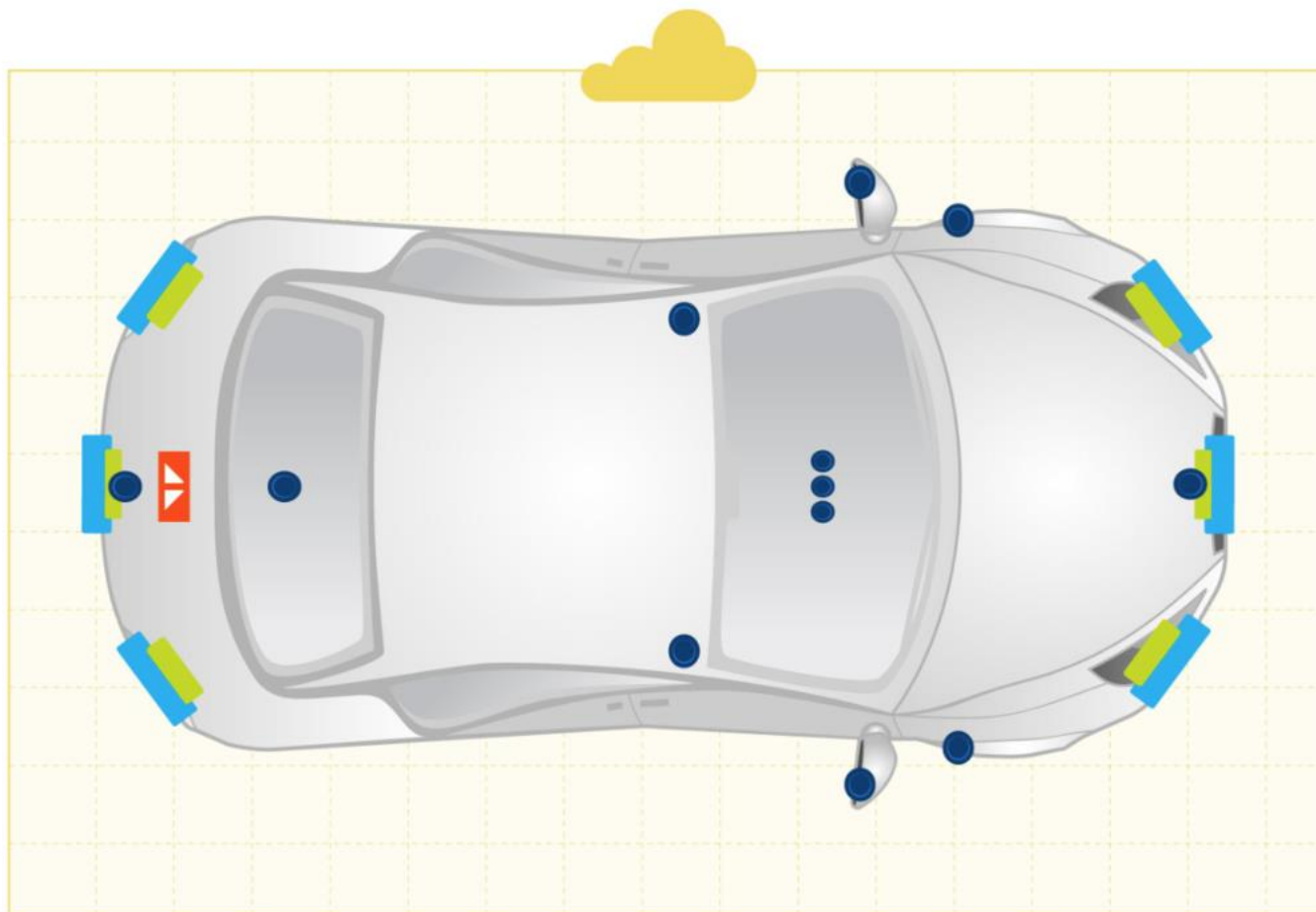
An MQ-9 Reaper unmanned aerial vehicle



AUTONOMOUS VEHICLE PLATFORM

The sensors, hardware and software provided by **Intel** and **Mobileye** give autonomous vehicles their ability to recognize the environment around them. This technology creates the building blocks for autonomous vehicles (AV) and includes a suite of cameras, lidar, radar, and computing and mapping technologies.

Click on an autonomous tool below to find out more information





- Cameras



- Lidars



- Radars



- Maps



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.

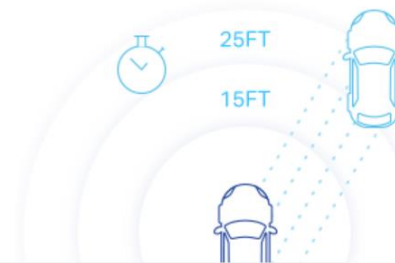
A typical lidar sensor emits pulsed light waves into the surrounding environment.



These pulses bounce off surrounding objects and return to the sensor.



The sensor uses the time it took for each pulse to return to the sensor to calculate the distance it traveled.



Repeating this process millions of times per second creates a precise, real-time 3D map of the environment. An onboard computer can utilize this map for safe navigation.



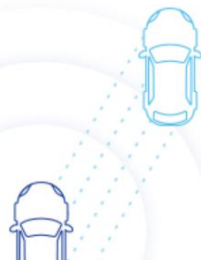
What is Lidar?

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.

A typical lidar sensor emits pulsed light waves into the surrounding environment.



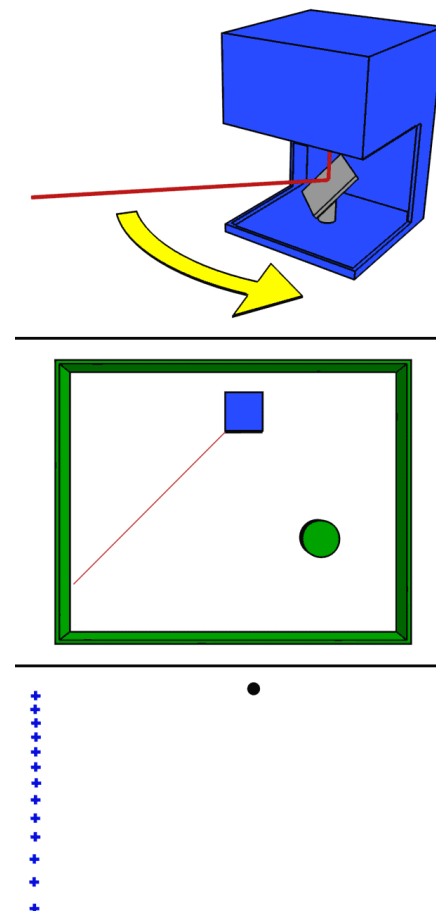
These pulses bounce off surrounding objects and return to the sensor.



The sensor uses the time it took for each pulse to return to the sensor to calculate the distance it traveled.



Repeating this process millions of times per second creates a precise, real-time 3D map of the environment. An onboard computer can utilize this map for safe navigation.





Lidars





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





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



Lidars





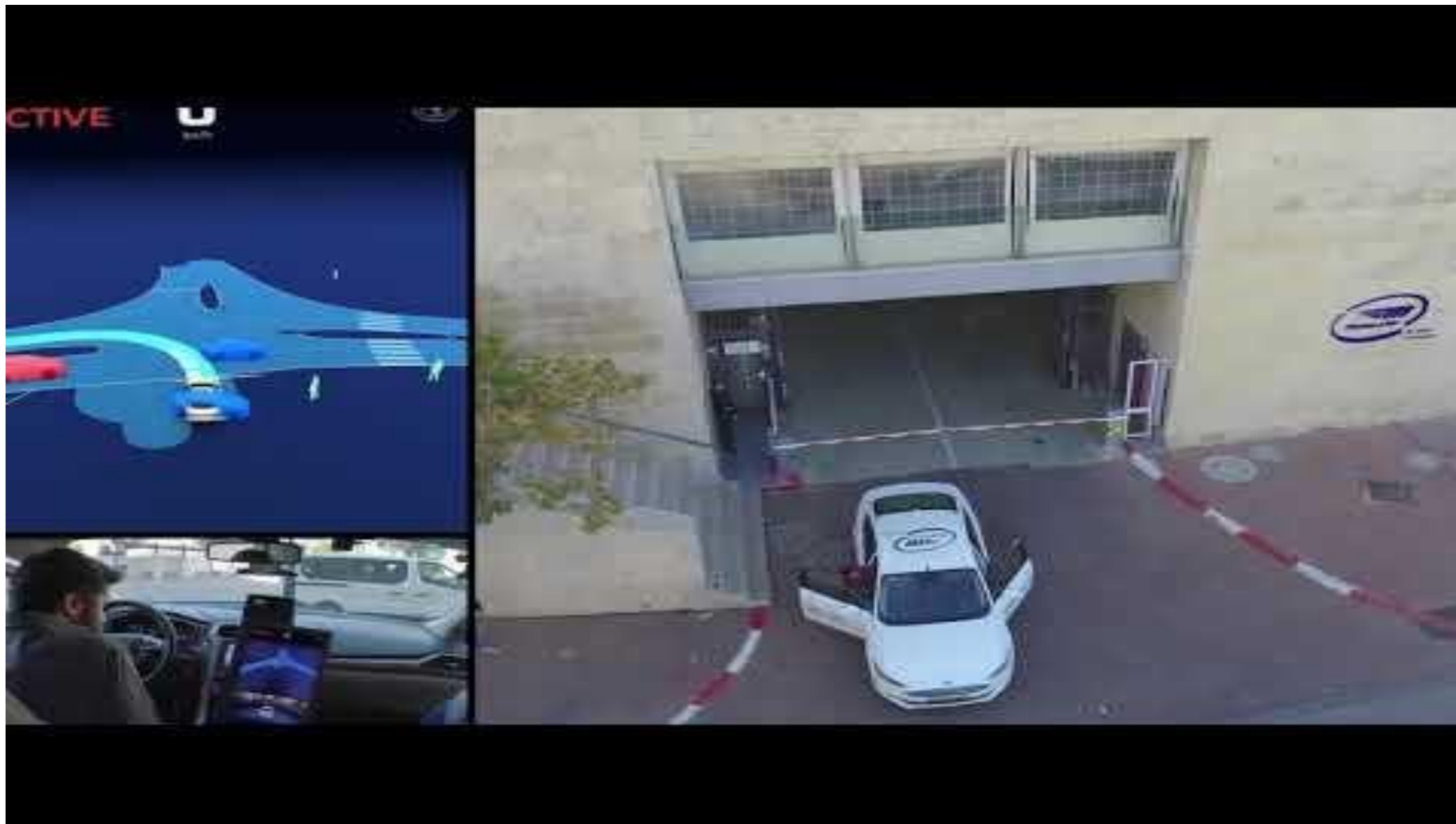
Lidars vs. Cameras

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





Lidars vs. Cameras

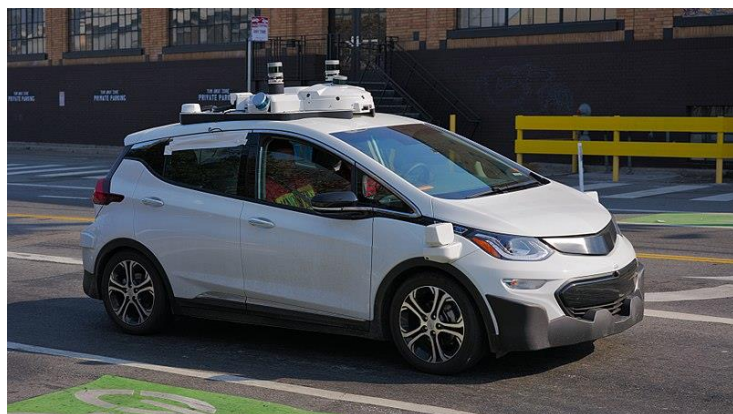




Lidar vs Camera

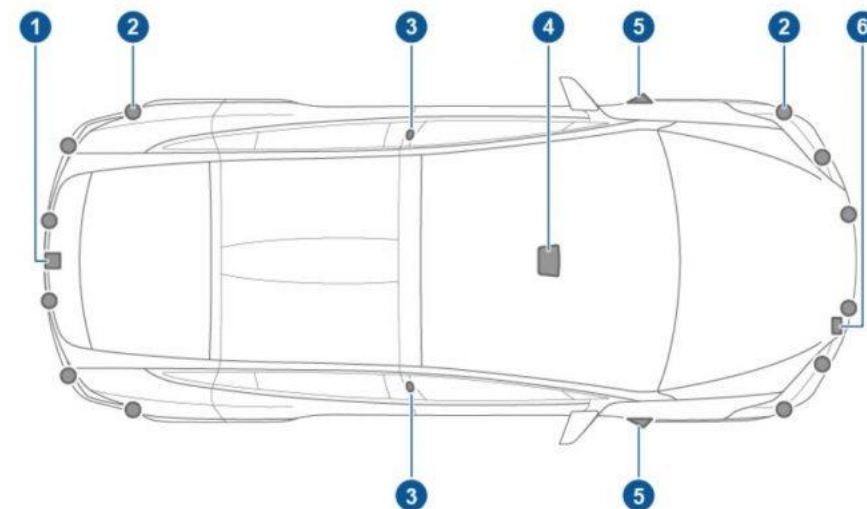
LiDAR

- 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

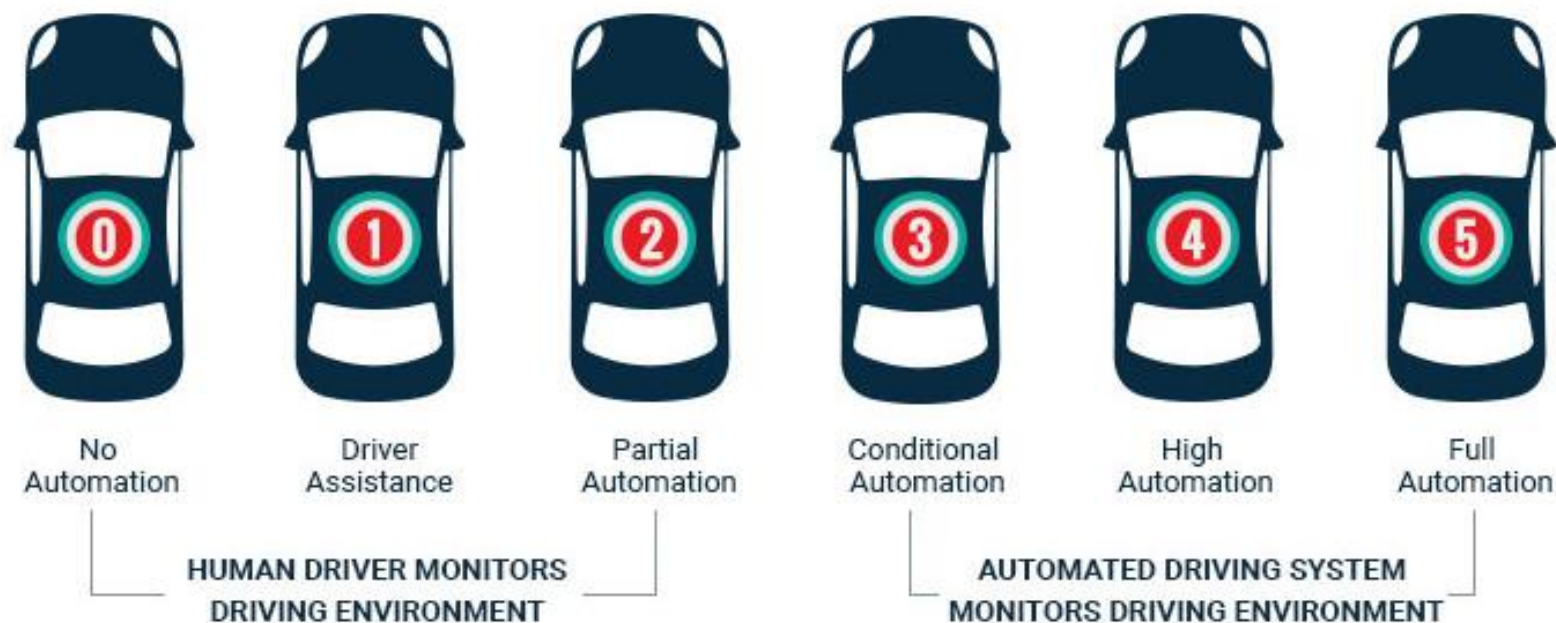


1. A camera is mounted above the rear license plate.
2. Ultrasonic sensors are located in the front and rear bumpers.
3. A camera is mounted in each door pillar.
4. Three cameras are mounted to the windshield above the rear view mirror.
5. A camera is mounted to each front fender.
6. Radar is mounted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

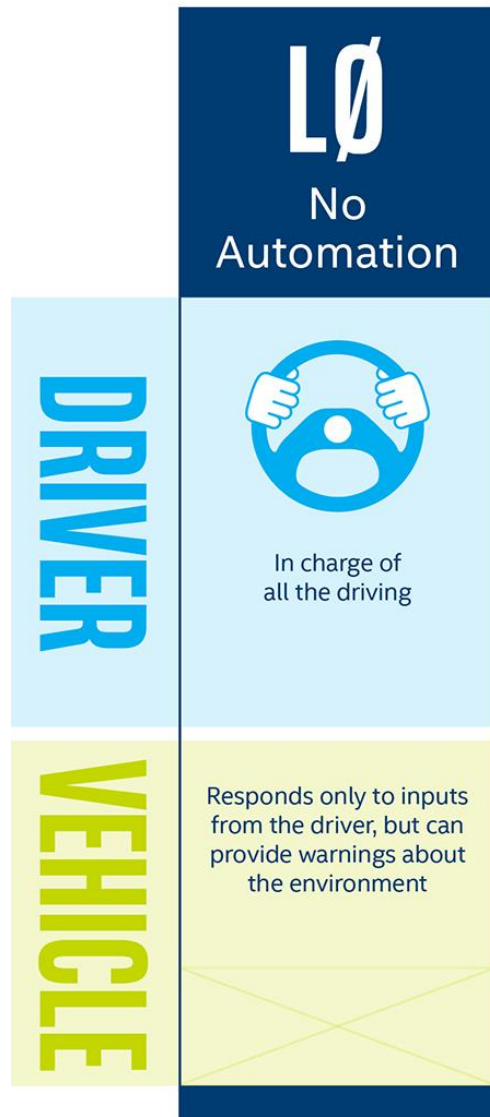


Levels of Autonomous Cars

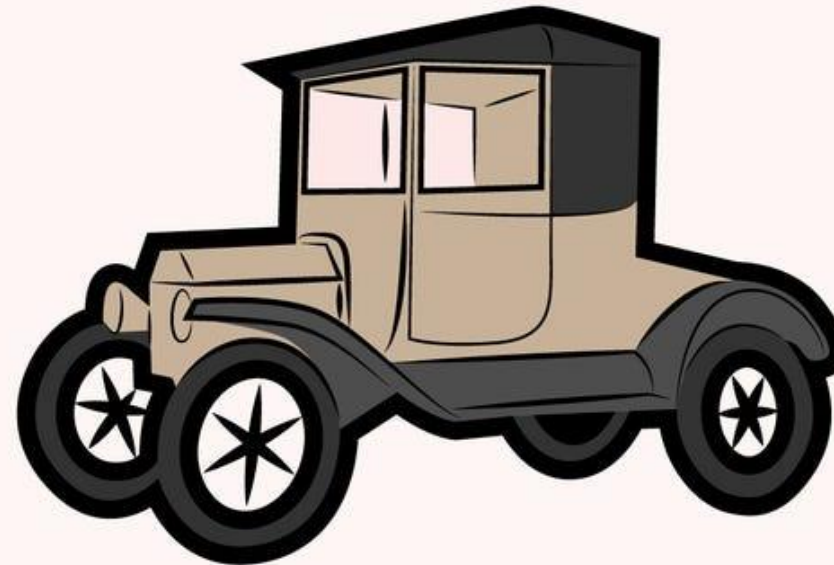




Levels of Autonomous Cars



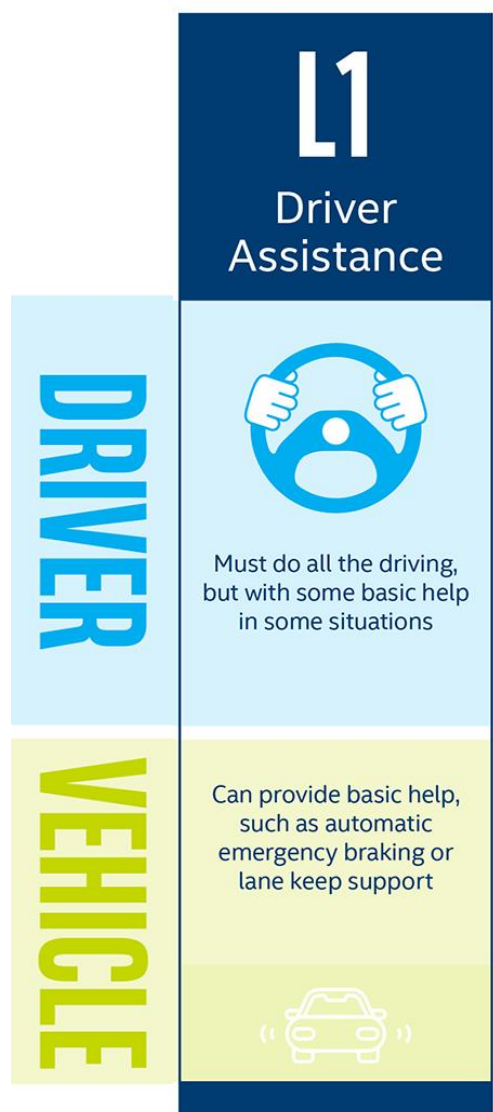
LEVEL 0



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>

Levels of Autonomous Cars

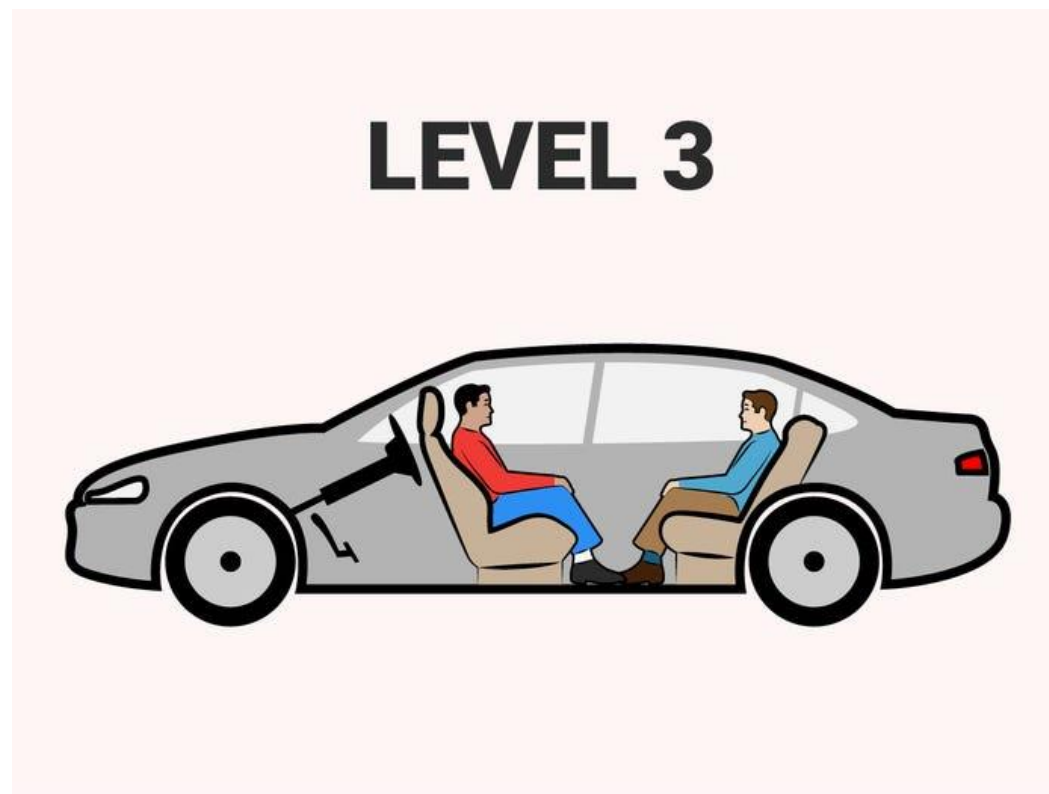


The driver still must be alert and monitor the environment at all times, but driving assist features that control acceleration, braking and steering may work together in unison so the driver does not need to provide any input in certain situations. Such automated functions available today include self-parking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

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



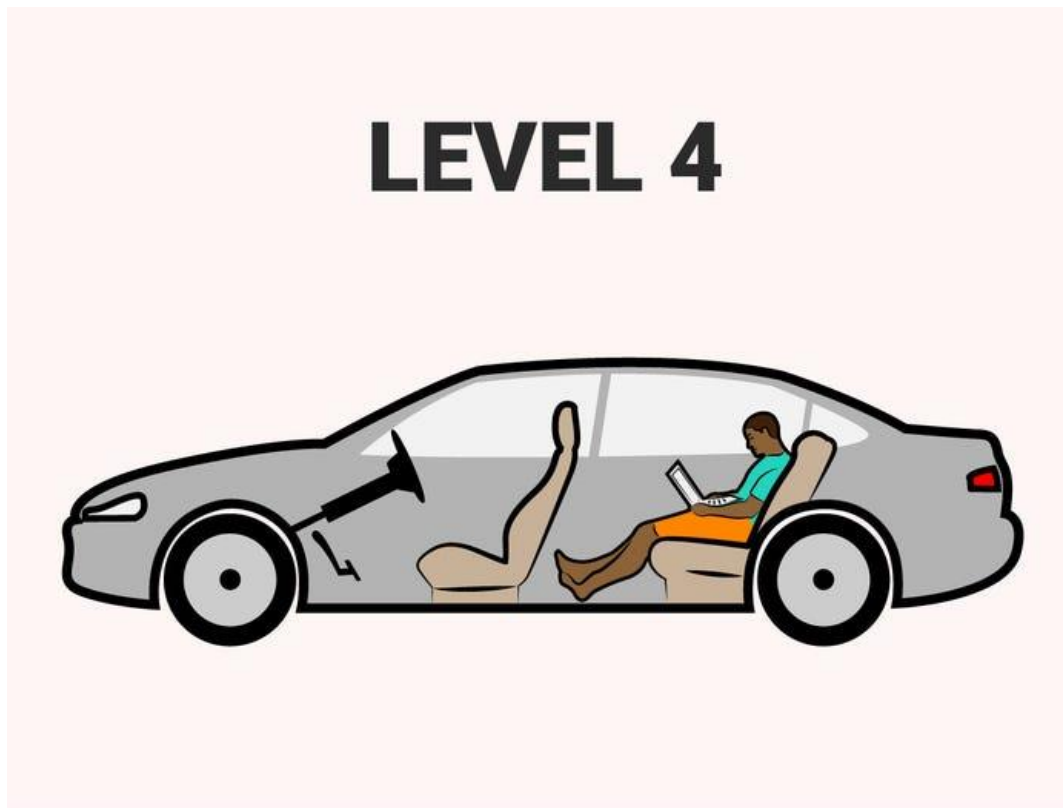
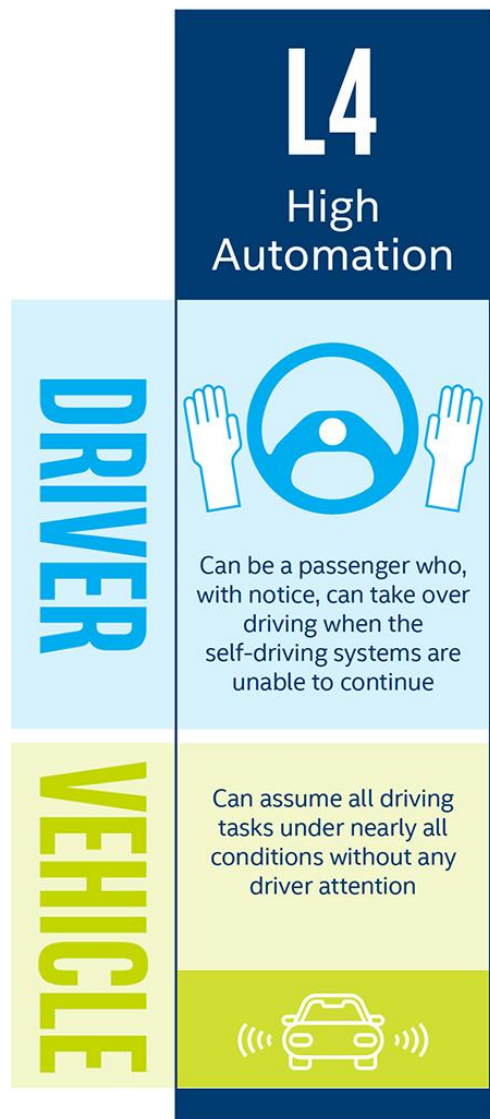
Levels of Autonomous Cars



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>

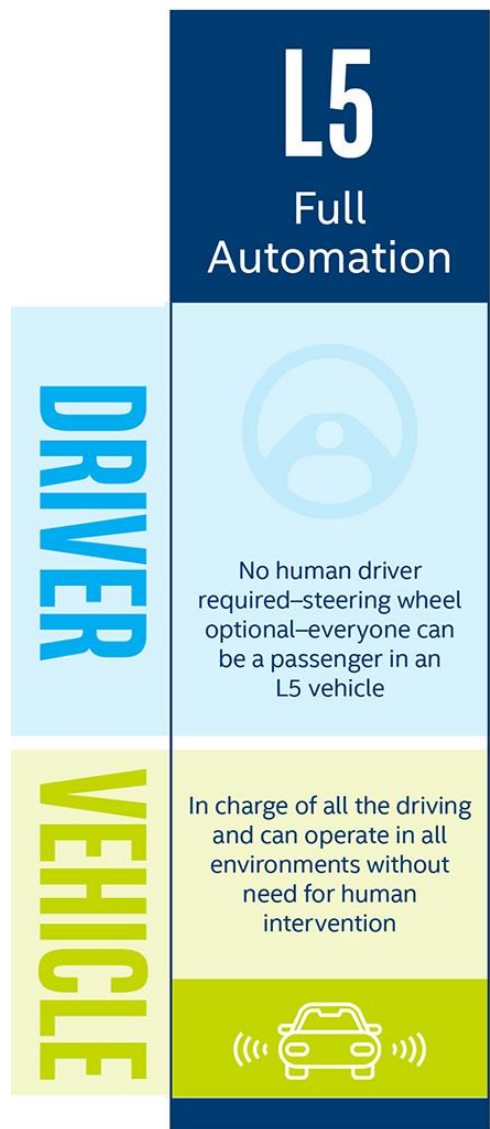
Levels of Autonomous Cars



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>

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>



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."

AUTOMATION LEVELS OF AUTONOMOUS CARS

LEVEL 0



There are no autonomous features.

LEVEL 1



These cars can handle one task at a time, like automatic braking.

LEVEL 2



These cars would have at least two automated functions.

LEVEL 3



These cars handle “dynamic driving tasks” but might still need intervention.

LEVEL 4



These cars are officially driverless in certain environments.

LEVEL 5



These cars can operate entirely on their own without any driver presence.



Company Scores (2019)

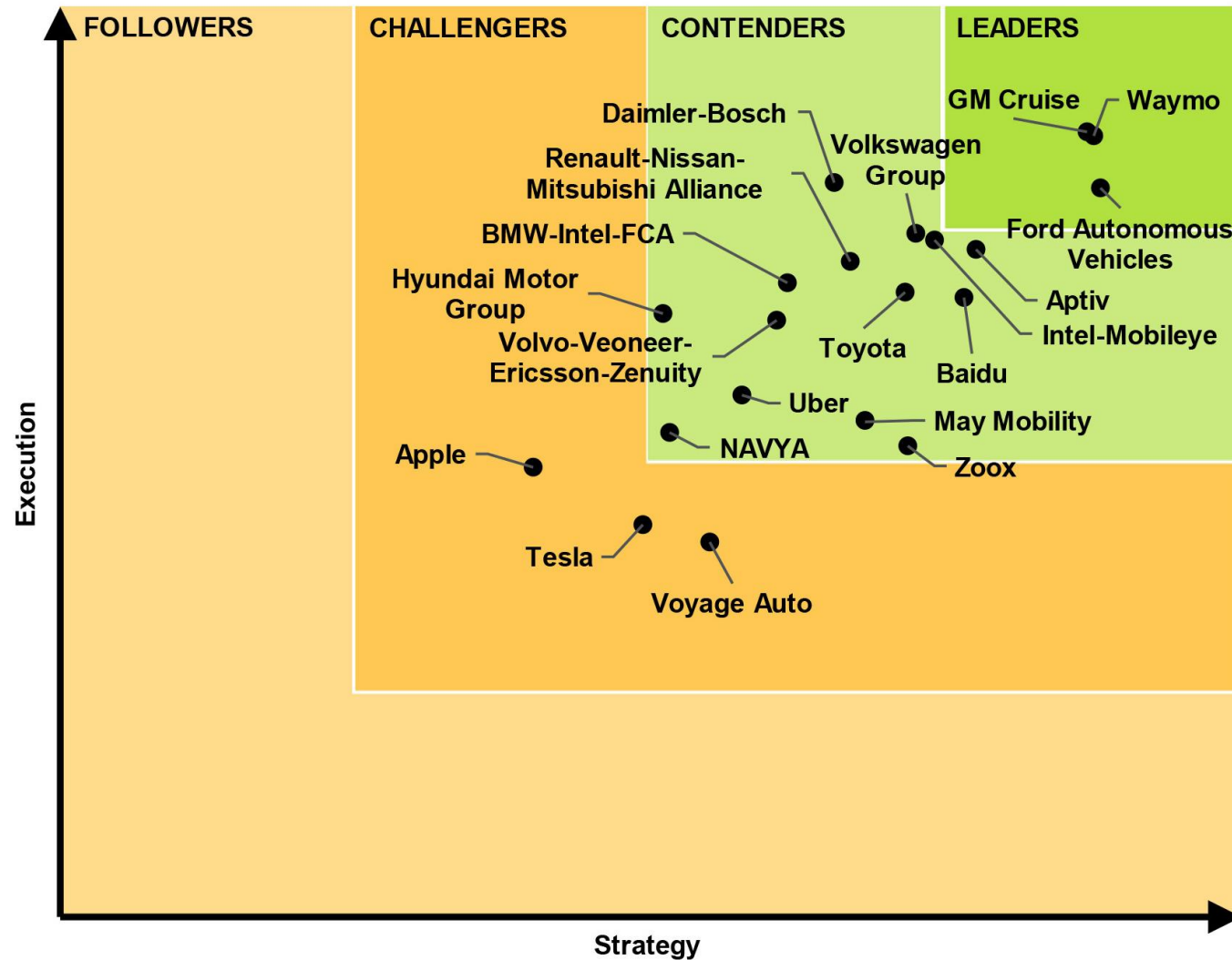
- Waymo (Google)
- GM
- Ford



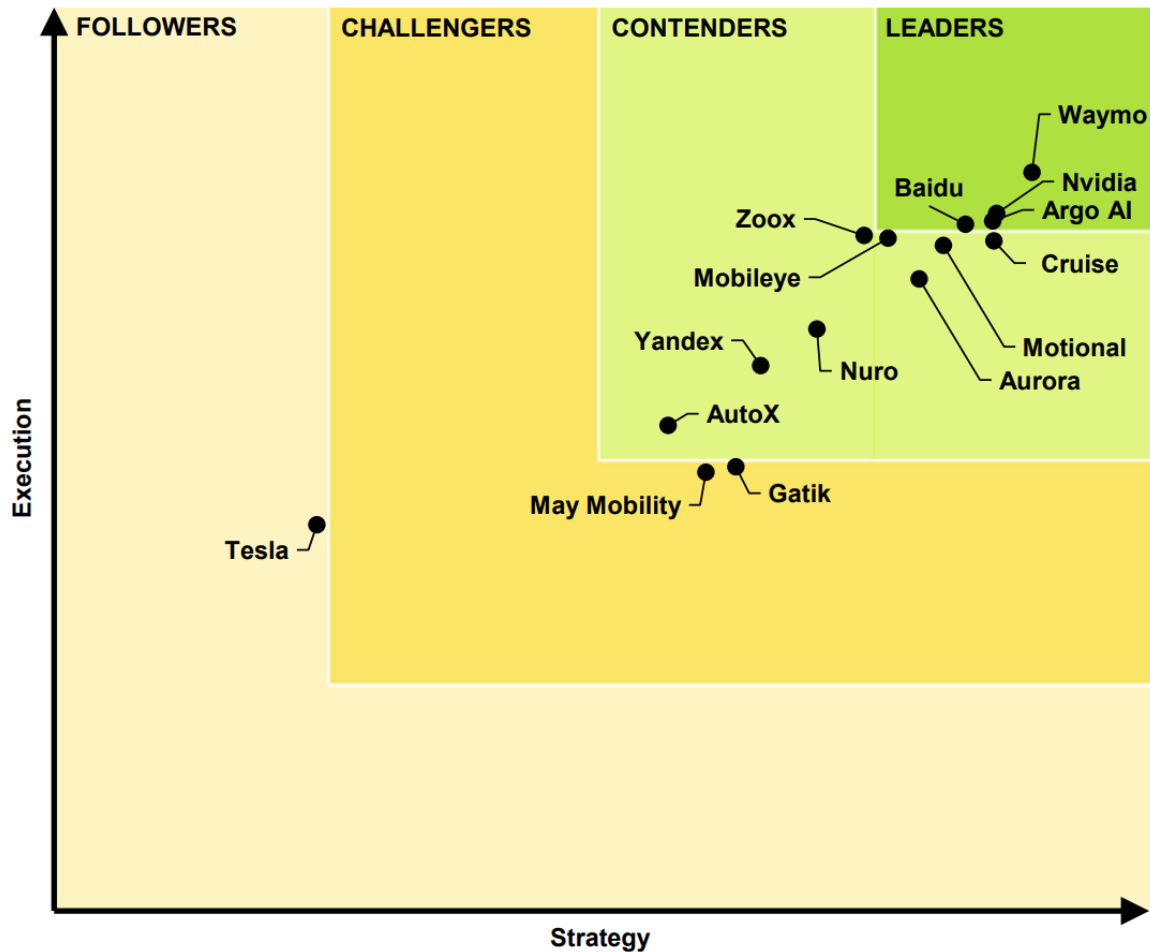


Company Scores (2019)

The Navigant Research Leaderboard Grid



Company Scores (2020)



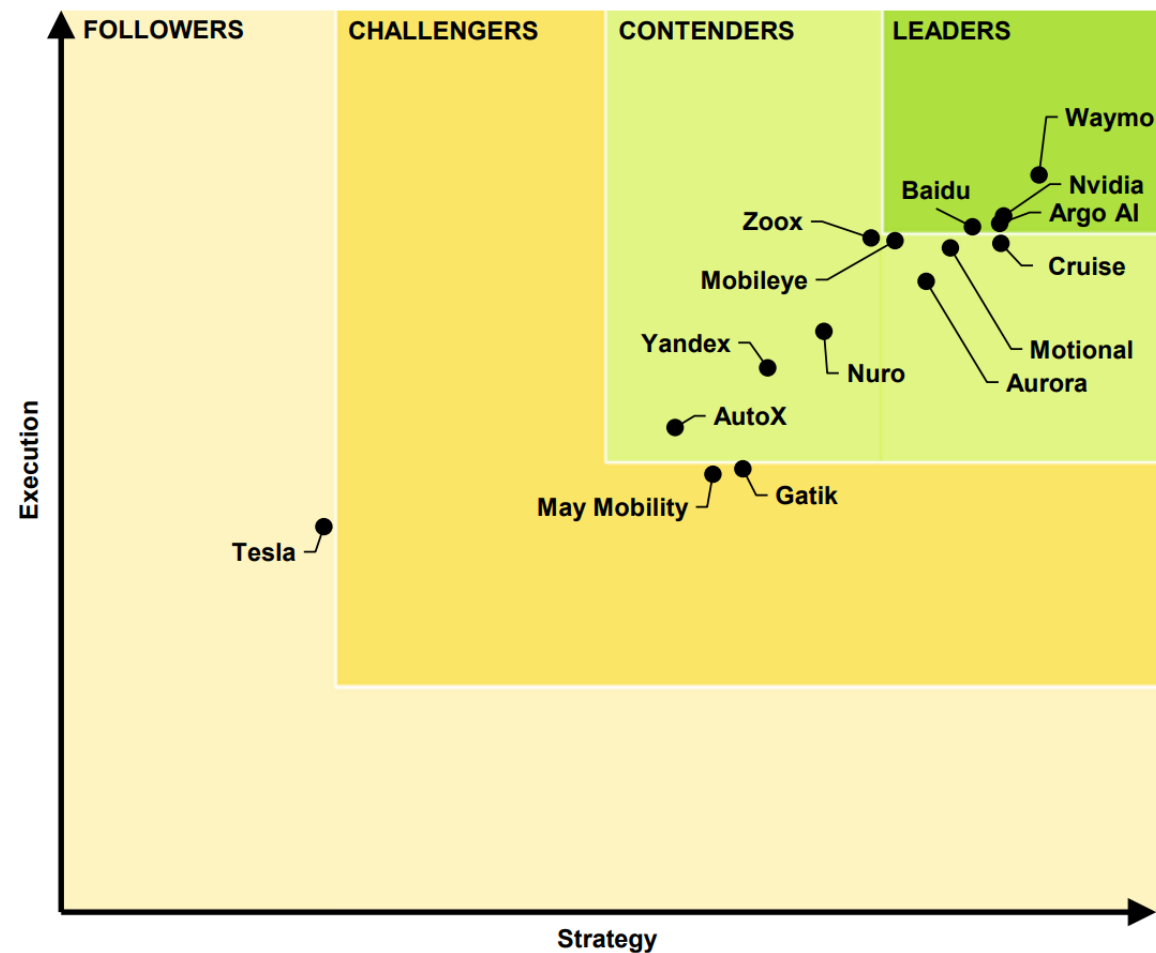
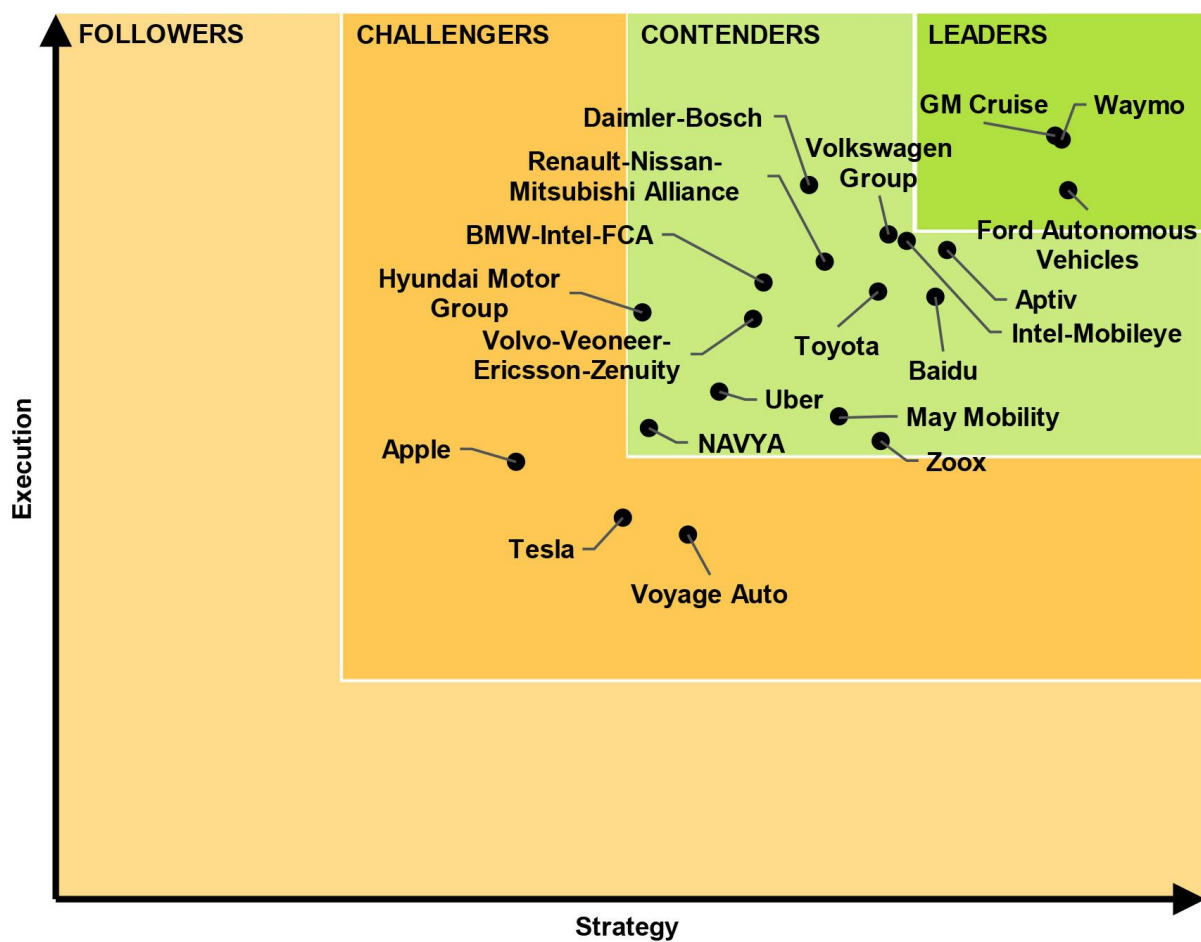
(Source: Guidehouse Insights)

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.



Company Scores (2019 vs 2020)

The Navigant Research Leaderboard Grid





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.



©2018 Waymo LLC

Page 5

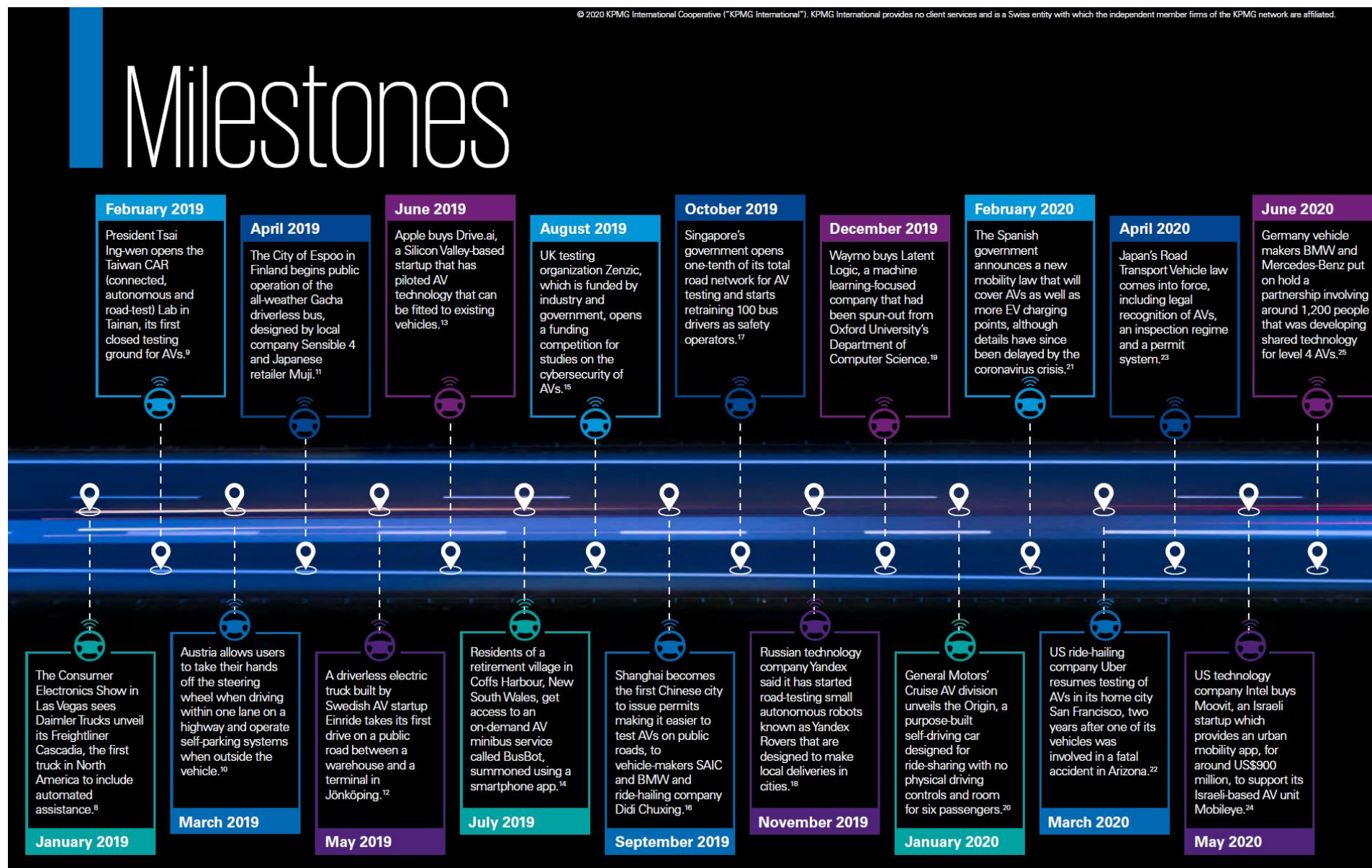






The Autonomous Vehicles Readiness Index

Index results			
Country or jurisdiction	Rank		2020 score
	2020	2019	
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









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

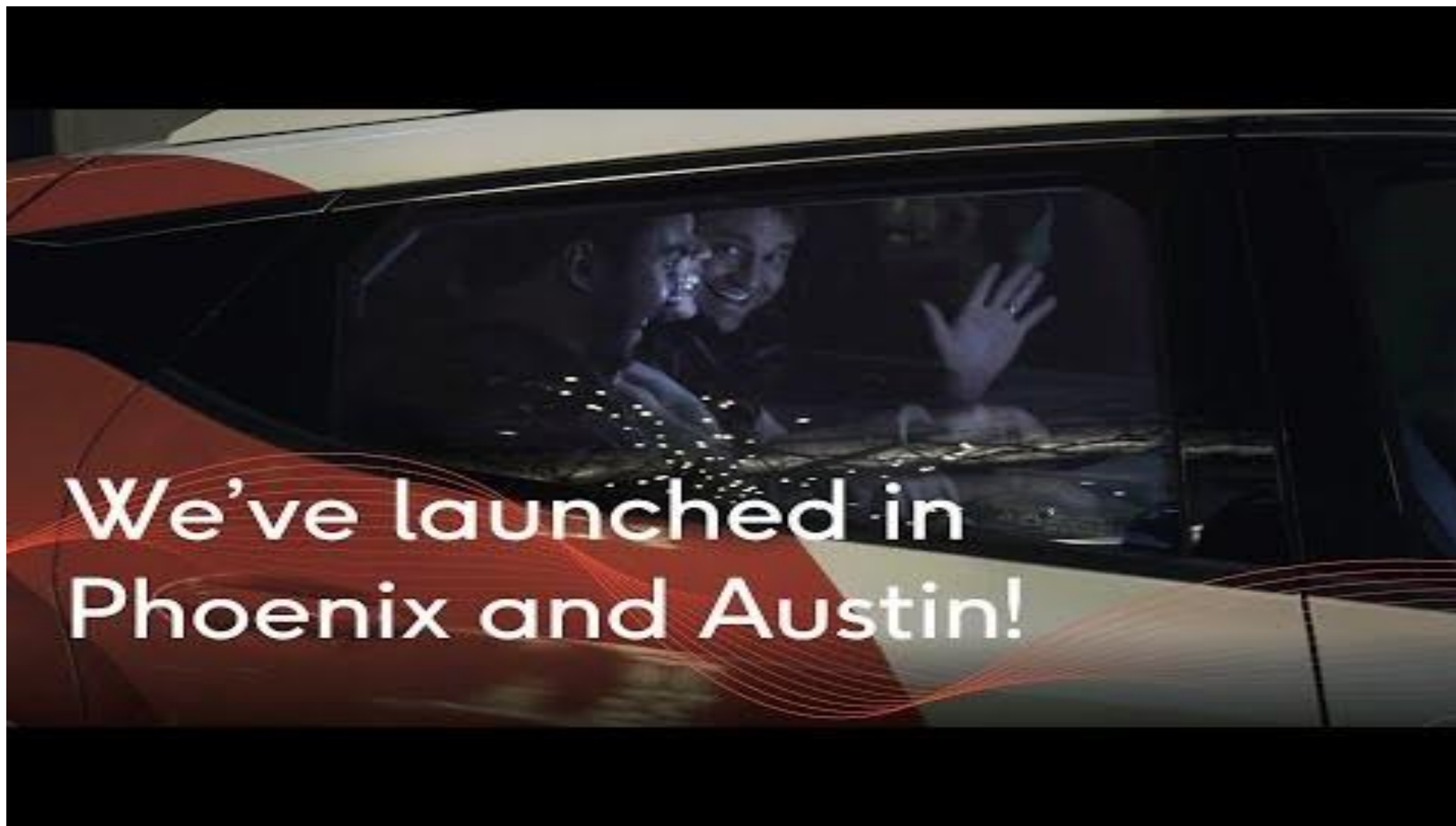


MINISTRY OF EDUCATION,
YOUTH AND SPORTS





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



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

- 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

- 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

- 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.



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



4th

- 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



5th

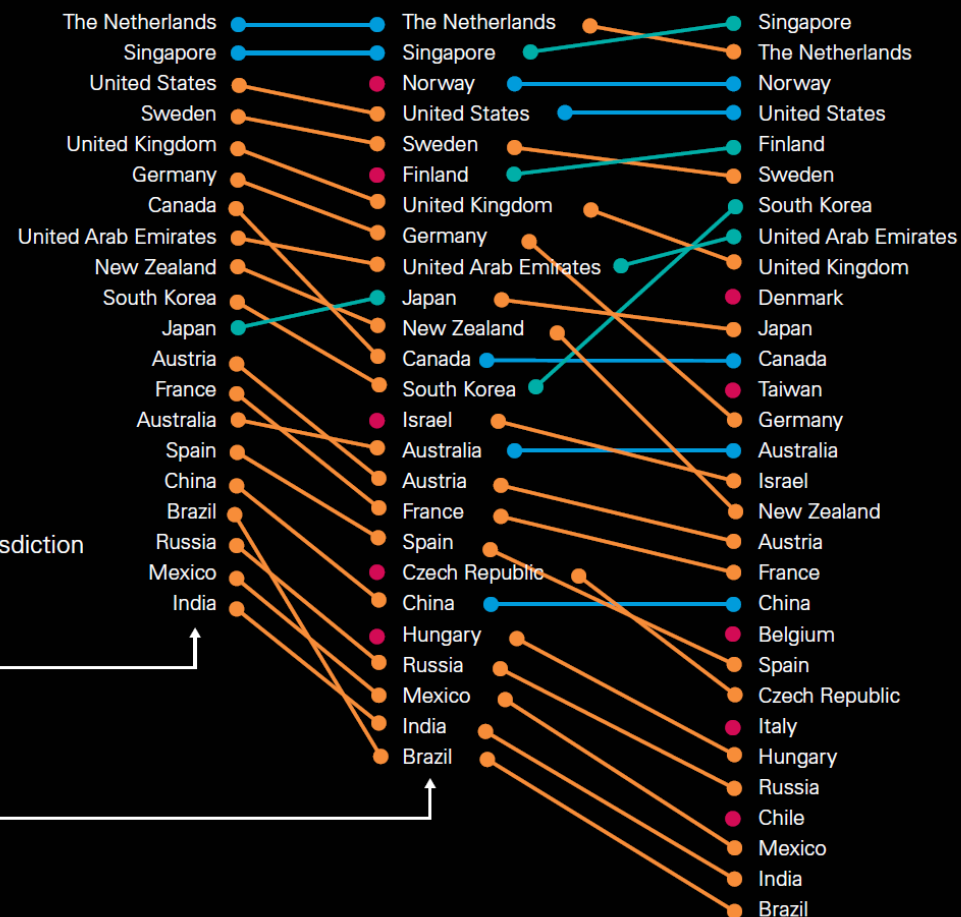
- Finland has the highest ratings for AV-specific regulations and for the efficiency of its legal system in challenging regulations, and its entire road network is open for AV trials.
- Helsinki (profiled in [Cities to watch](#)) and its neighbor Espoo both run public AV bus services, with the latter 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.



Comparative AVRI positions from 2018 to 2020



- Upward movement
- Downward movement
- Newly added country/jurisdiction
- Same ranking





© 2020 KPMG International Cooperative ("KPMG International"). KPMG International provides no client services and is a Swiss entity with which the independent member firms of the KPMG network are affiliated.

23 | Czech Republic



Policy and
legislation



Technology
and innovation



Infrastructure



Consumer
acceptance

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



The Autonomous Vehicles Readiness Index

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



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



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



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



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



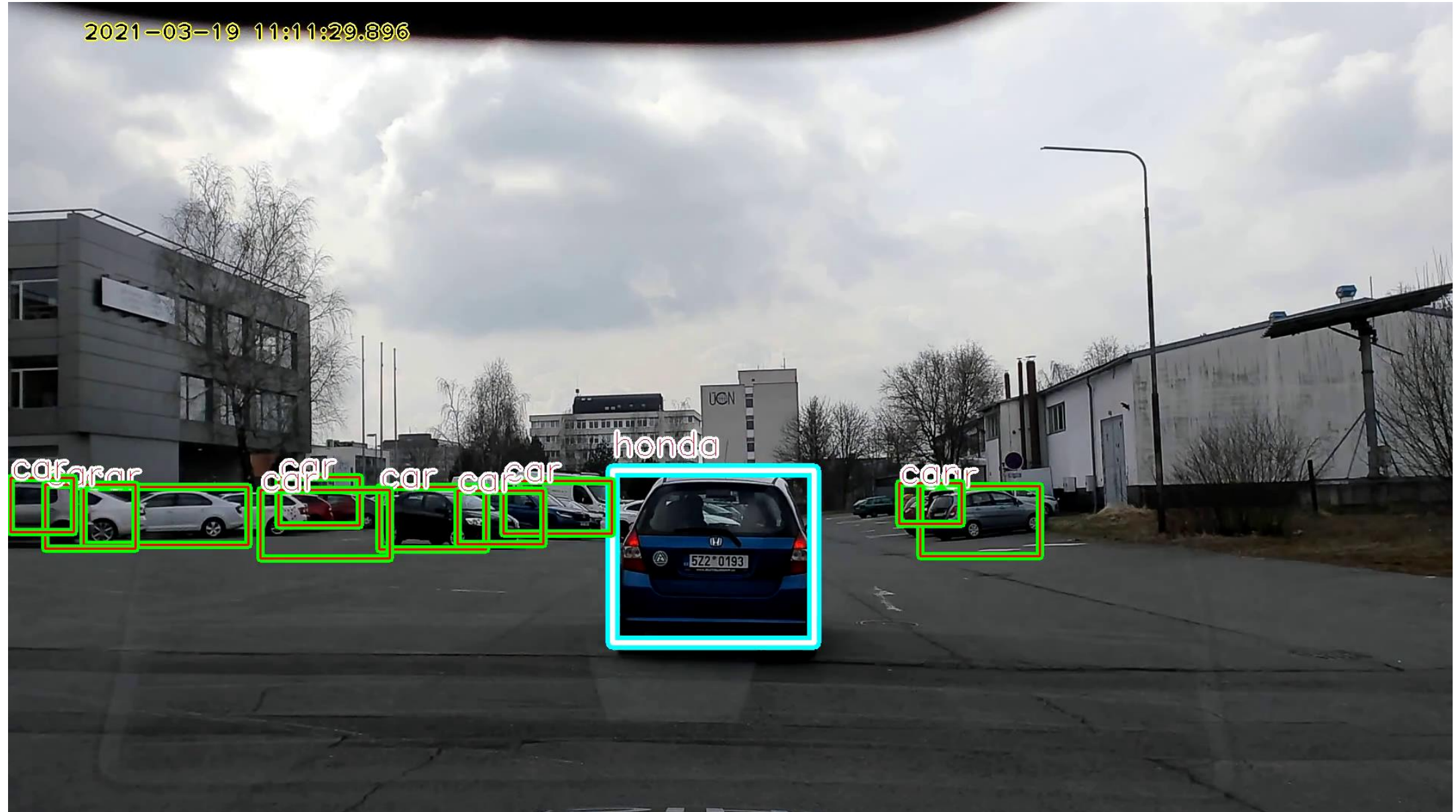
What is Object Detection?

- 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?

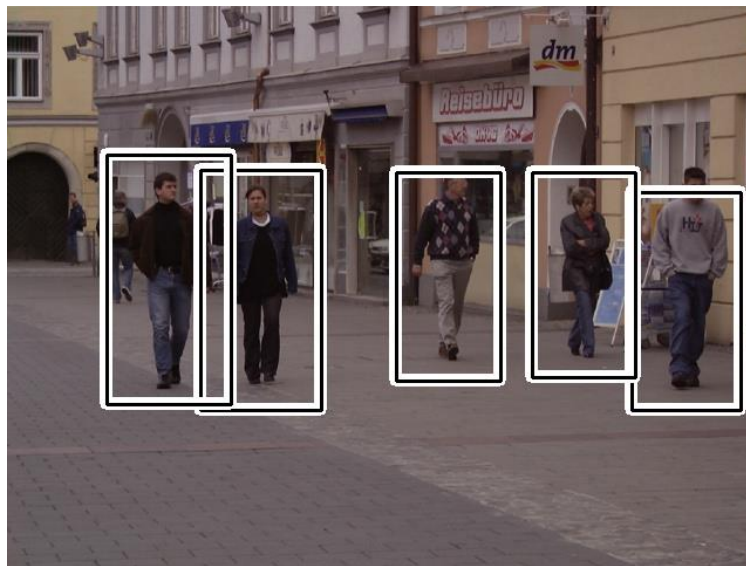
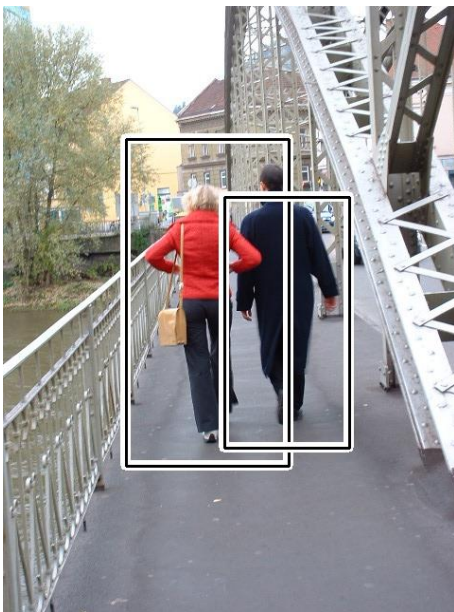






What is Object Detection?

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





Problems (Challenges)

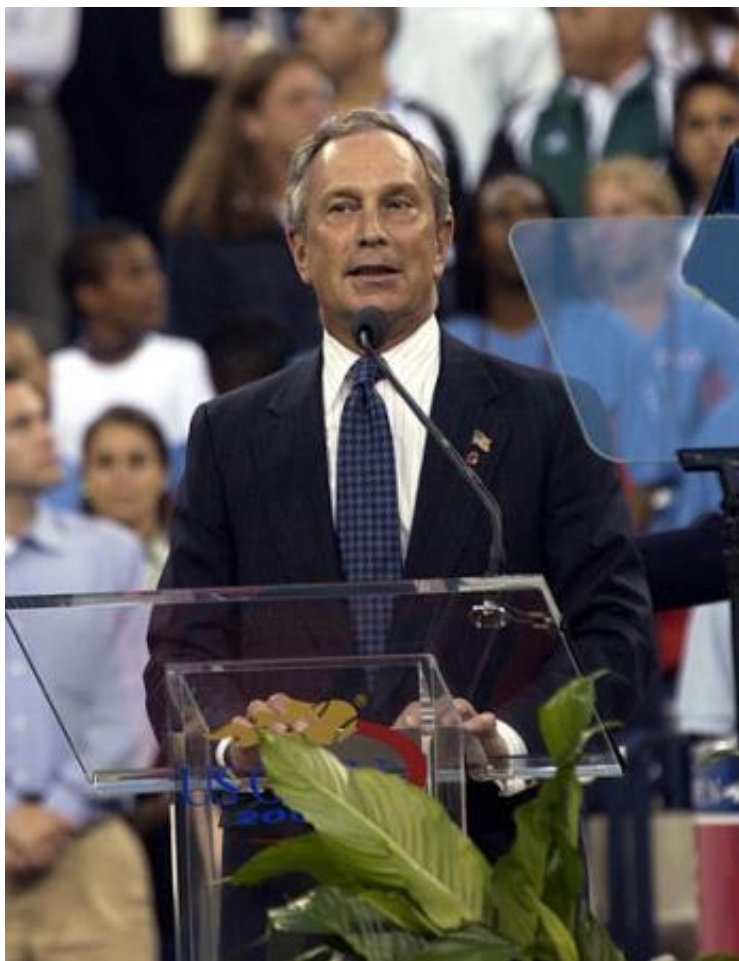
- different views
- Illumination challenges
- occlusion
- different backgrounds
- shadows
- ...
- ...





Problems (Challenges)

- low quality images





Problems (Challenges)

- illumination + low quality





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/>)

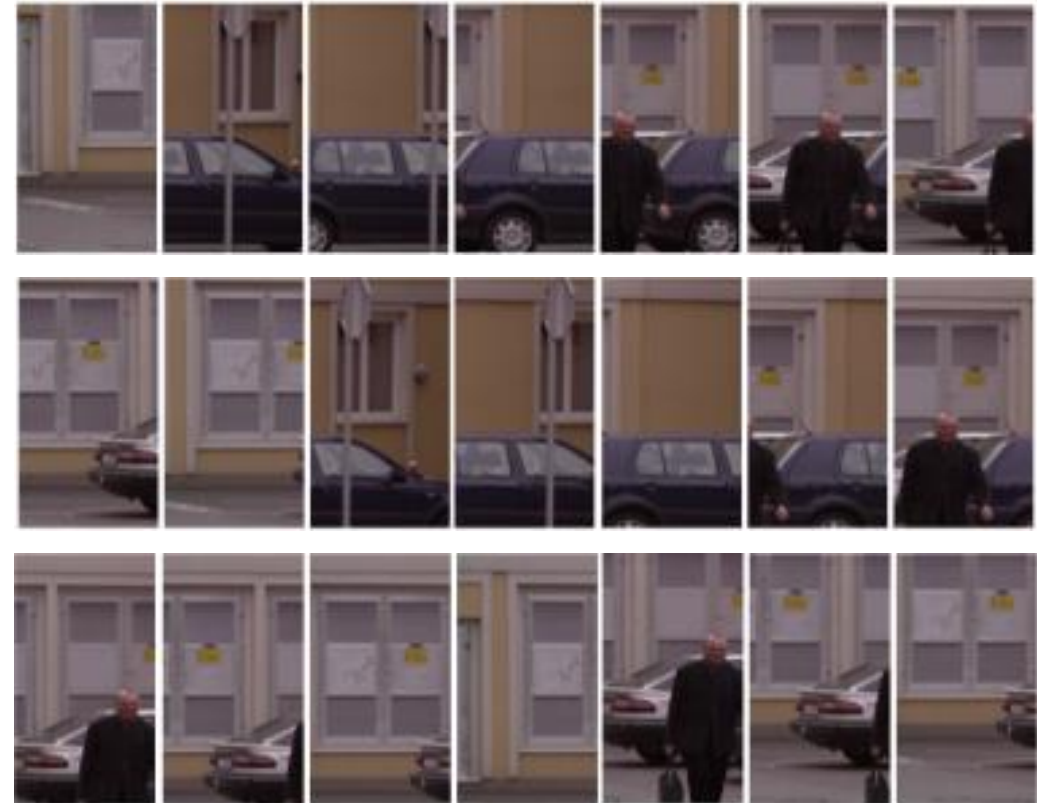


Sliding Window - Main Idea





Sliding Window - Main Idea





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 classifier).
- 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

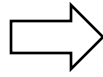


Sliding Window - Main Idea

- These detections are merged to the final bounding box that represents the resulting detection.
- The classifier 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.



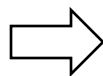
Detection Process



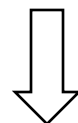
Feature Vector
(gradient, HOG, LBP, ...)



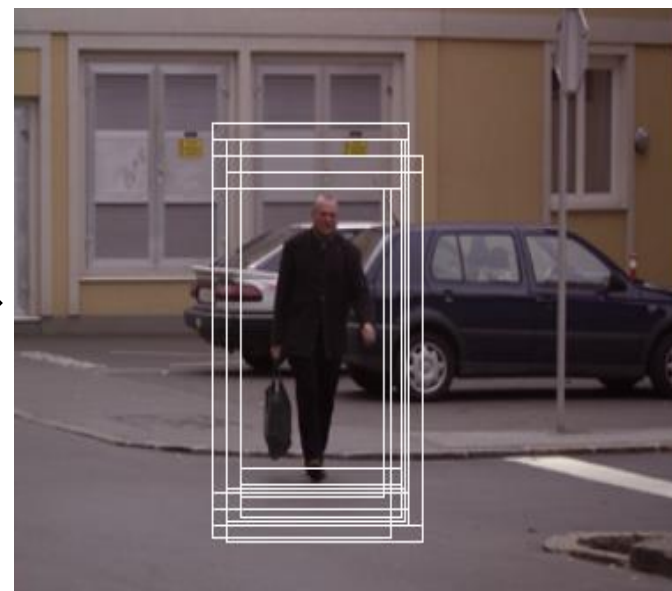
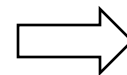
Detection Process



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)



Training Set – Traffic Lights

Bosch Small Traffic Lights Dataset (Germany)

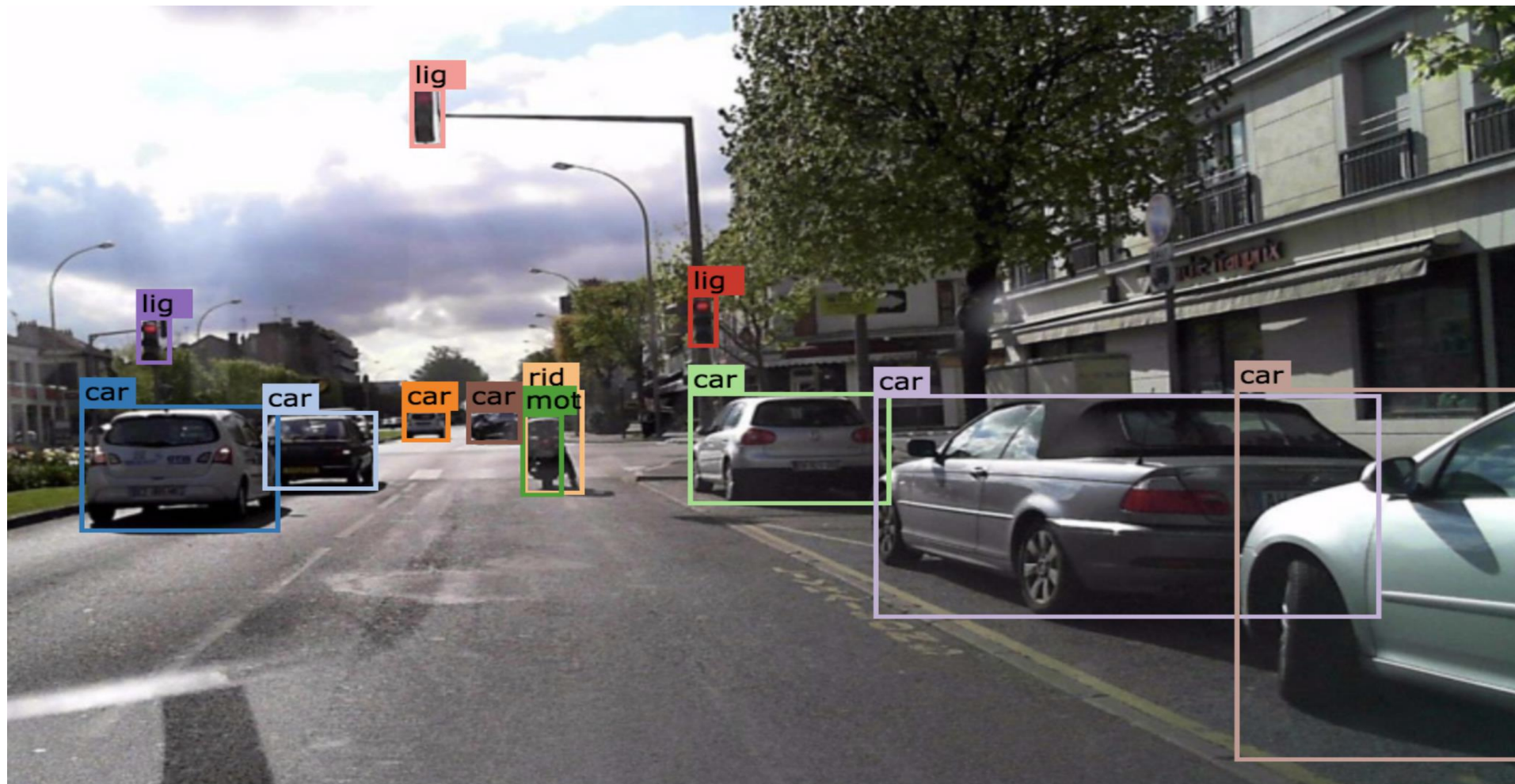
- Training set:
 - 5093 images
 - Annotated about every 2 seconds
 - 10756 annotated traffic lights
 - Median traffic lights width: ~8.6 pixels
 - 15 different labels
 - 170 lights are partially occluded
- Test set:
 - 8334 consecutive images
 - Annotated at about 15 fps
 - 13486 annotated traffic lights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





Training Set – Road Objects

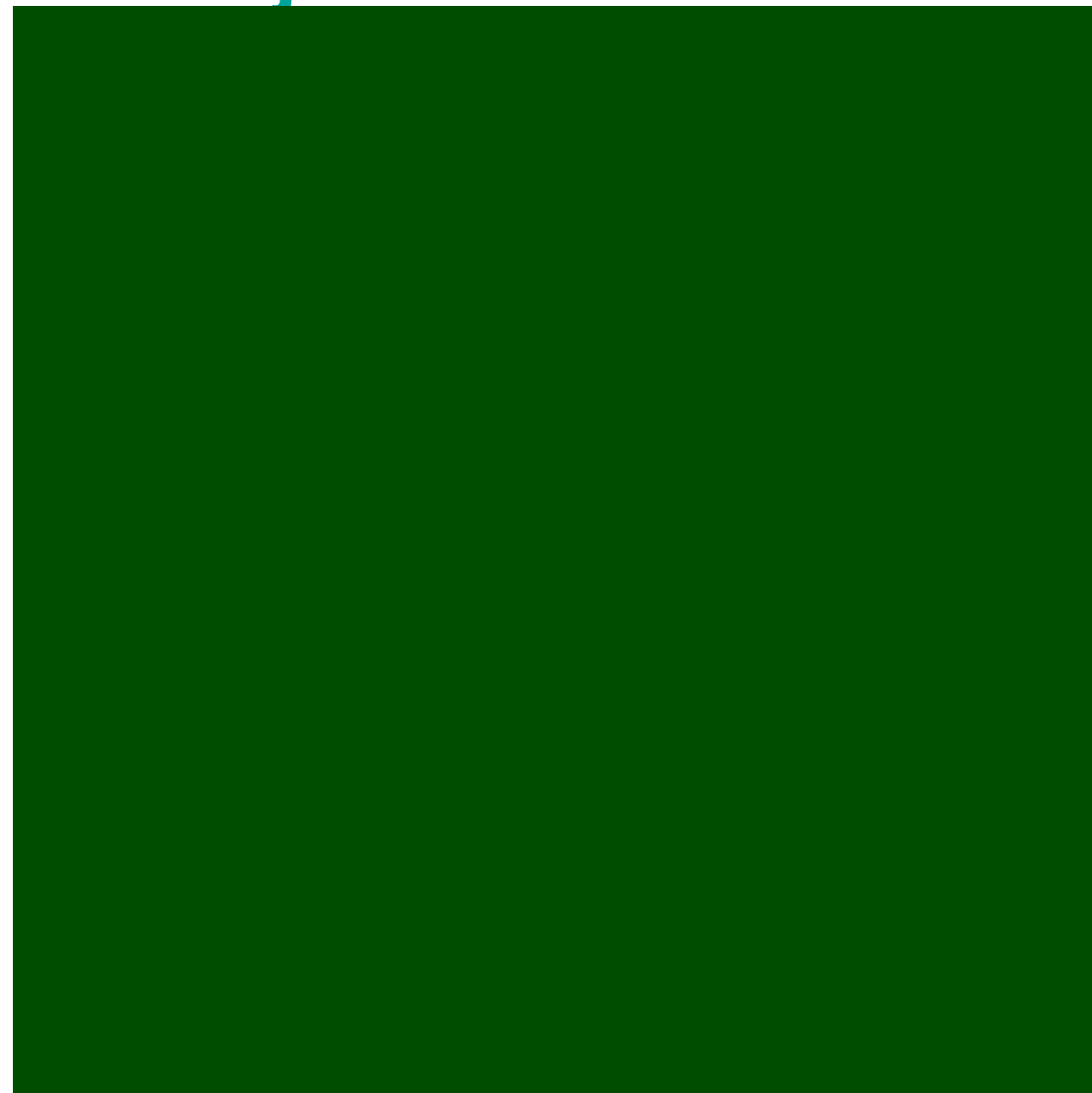
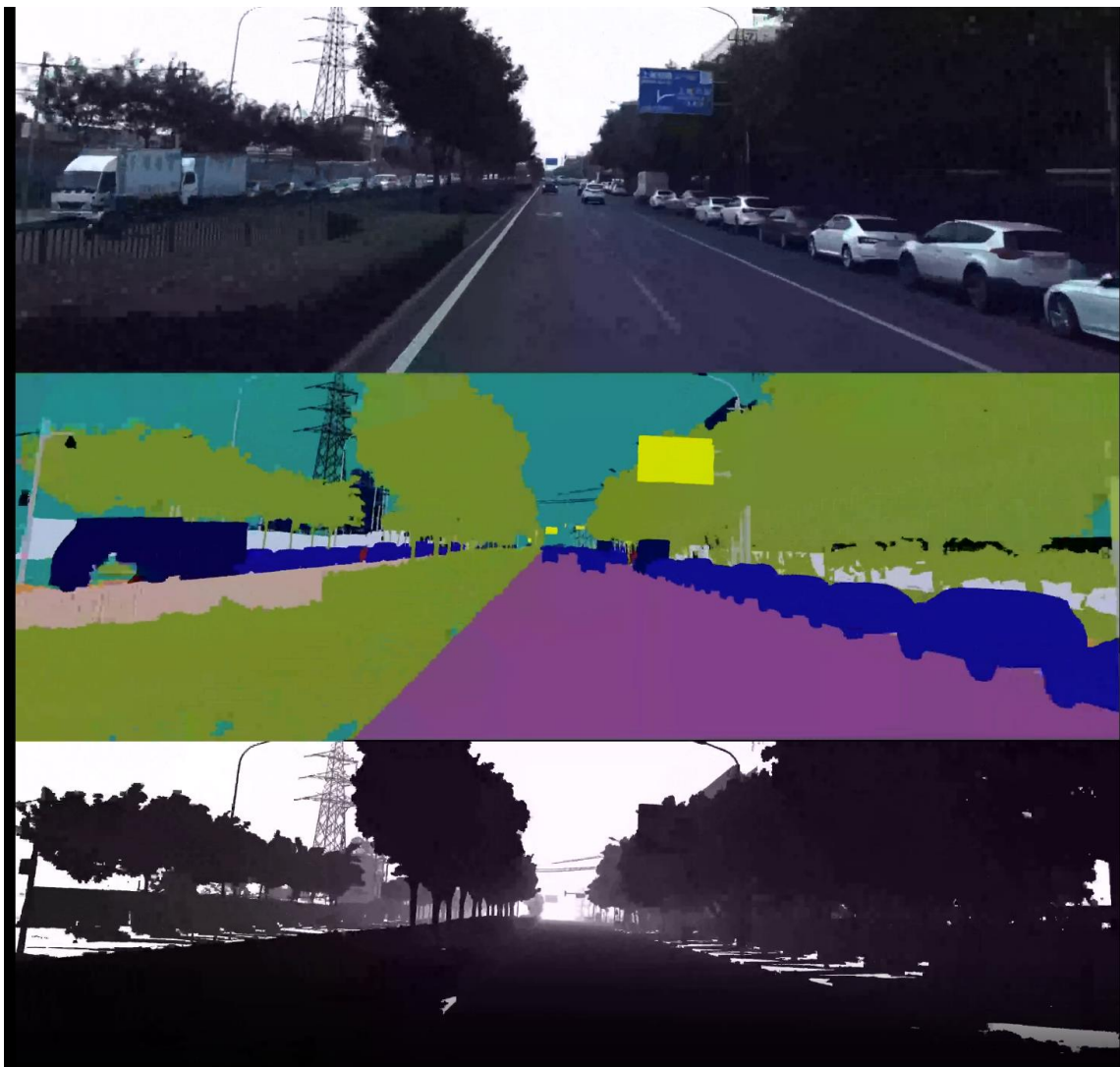
Berkeley Deep Drive





<http://apolloscape.auto/>

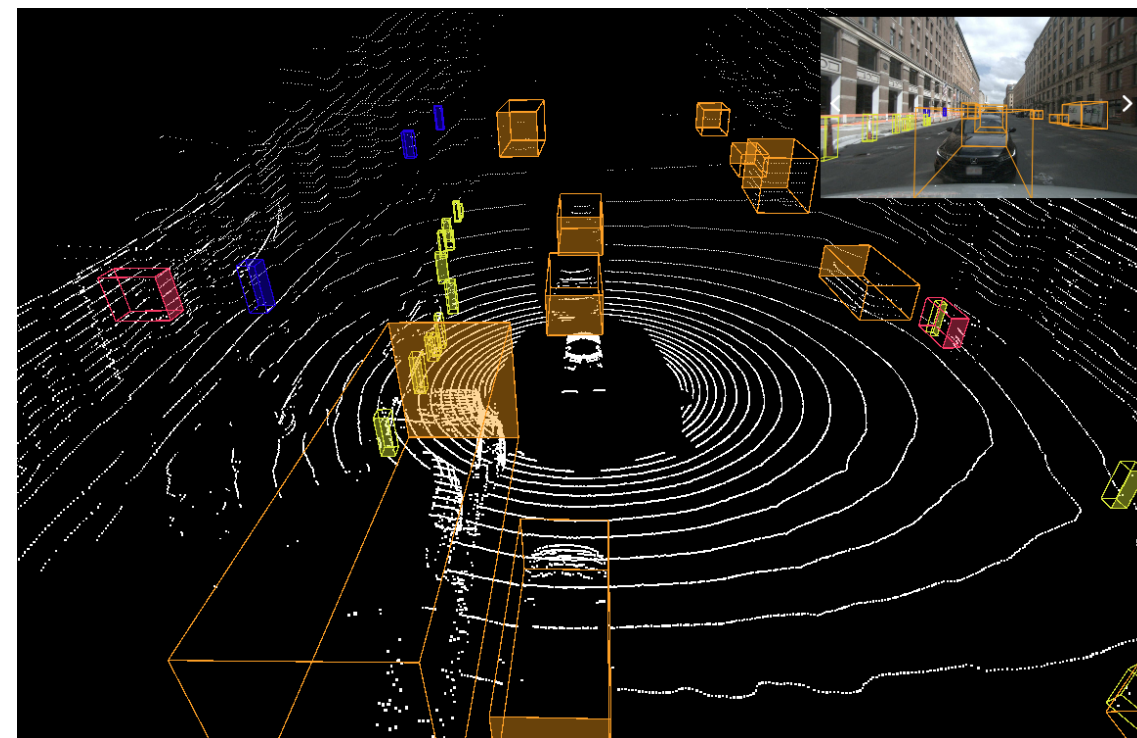
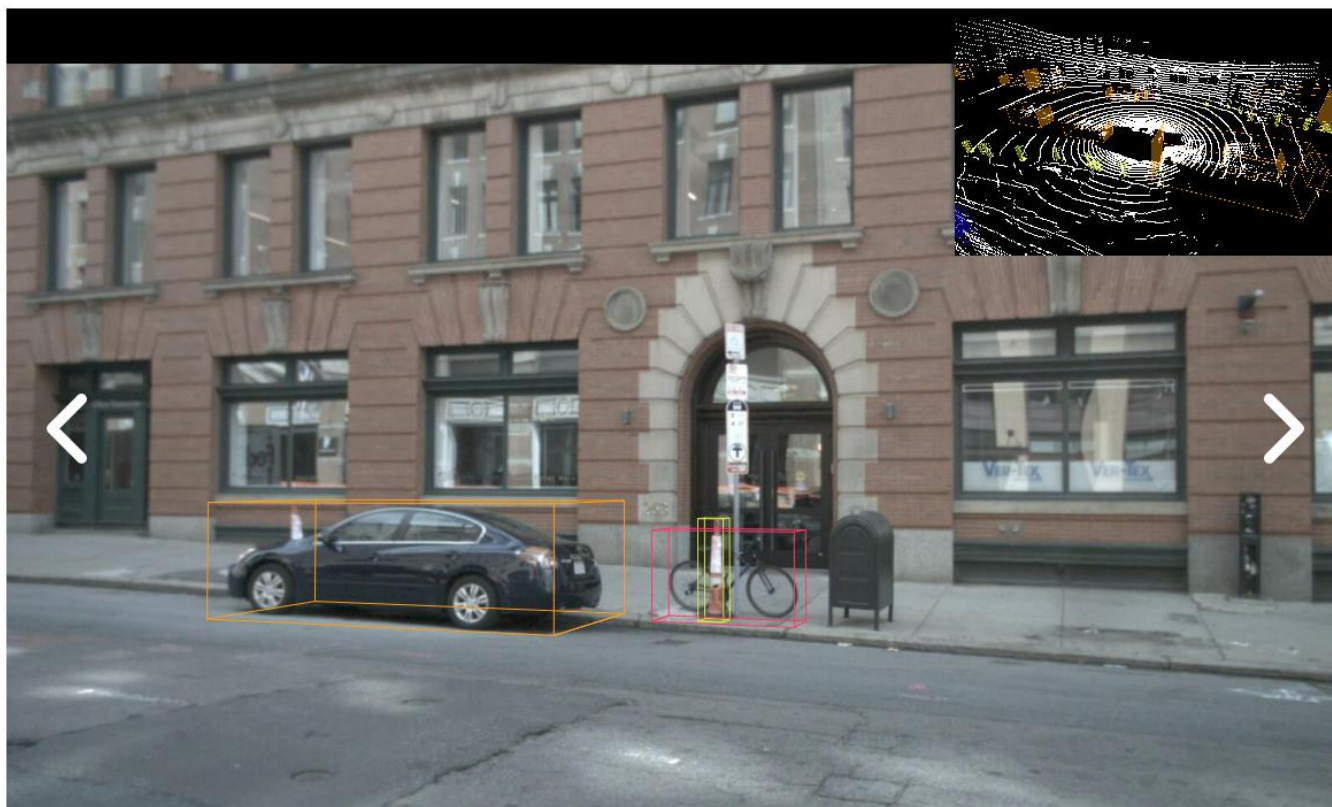
Training Set – Road Objects





Training Set – Road Objects

nuScenes





<https://boxy-dataset.com/boxy/>

Training Set – Road Objects





- Haar

- HOG

- LBP

- SIFT, SURF

- CNNs



Traditional Approaches



KeyPoints



Deep Learning Approach



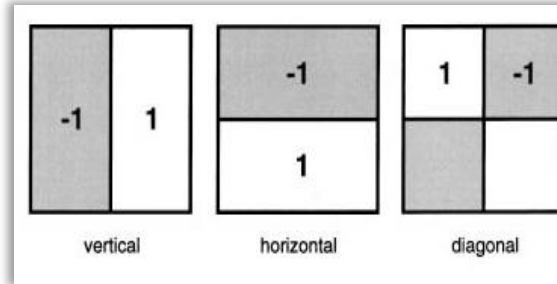
Related Works

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

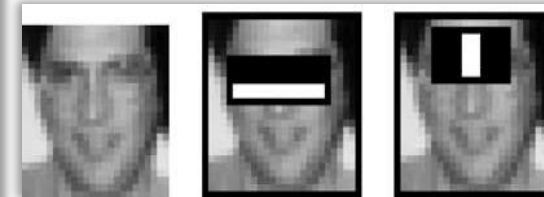


Viola, Jones
(2001,2004)
cit. > 10000

Robust Real-Time Face Detection

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

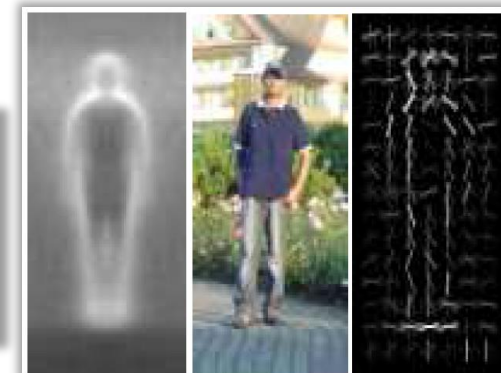
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs
(2005)
cit. > 10000

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alpes, 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 below or on top of eyes have different intensities than 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.



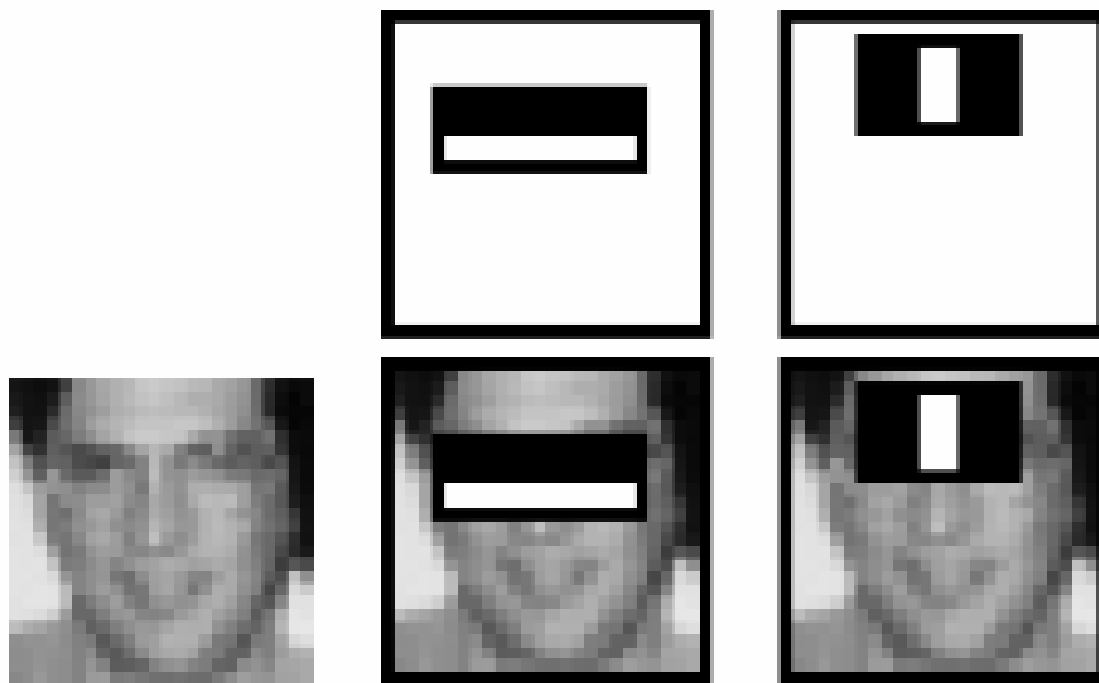
Haar Wavelet-based Descriptors

- 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 classifiers and the cascade of classifiers 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



Haar Wavelet-based Descriptors

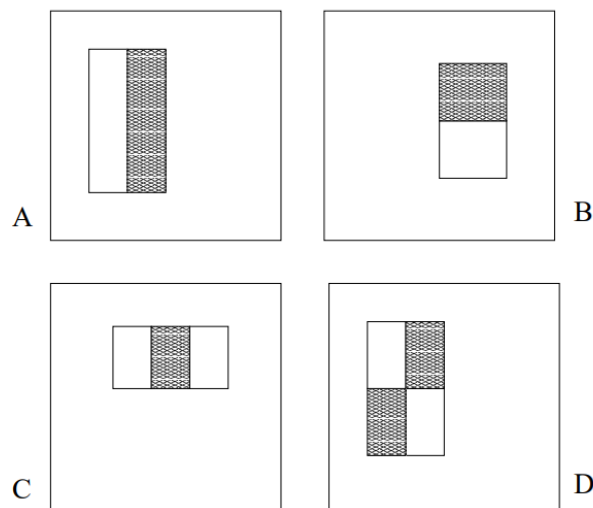


Figure 1: Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

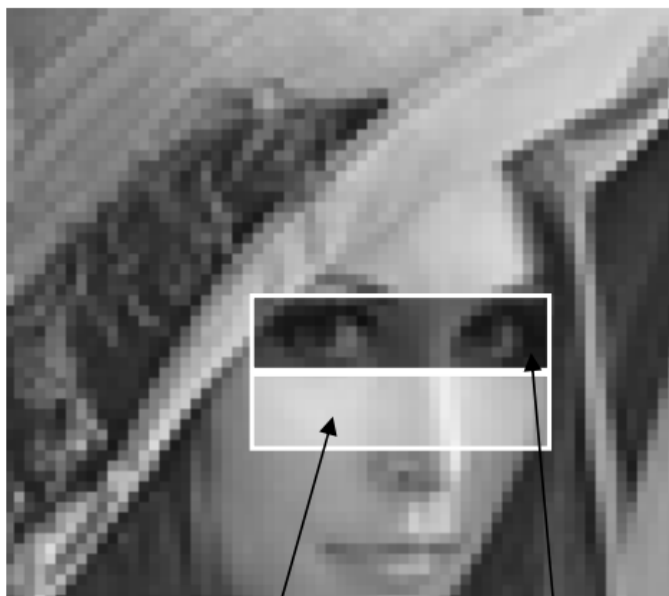
There are many motivations for using features rather than the pixels directly. The most common reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. For this system there is also a second critical motivation for features: the feature based system operates much faster than a pixel-based system.

The simple features used are reminiscent of Haar basis functions which have been used by Papageorgiou et al. [10]. More specifically, we use three kinds of features. The value of a *two-rectangle feature* is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent (see Figure 1). A *three-rectangle feature* computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a *four-rectangle feature* computes the difference between diagonal pairs of rectangles.



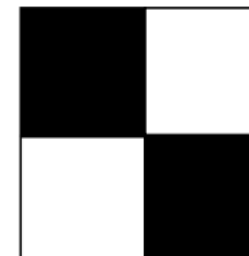
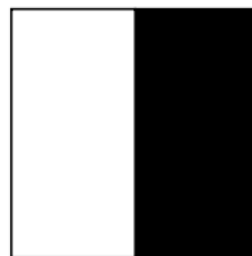
Features

- Rectangular features



R_{white}

R_{black}

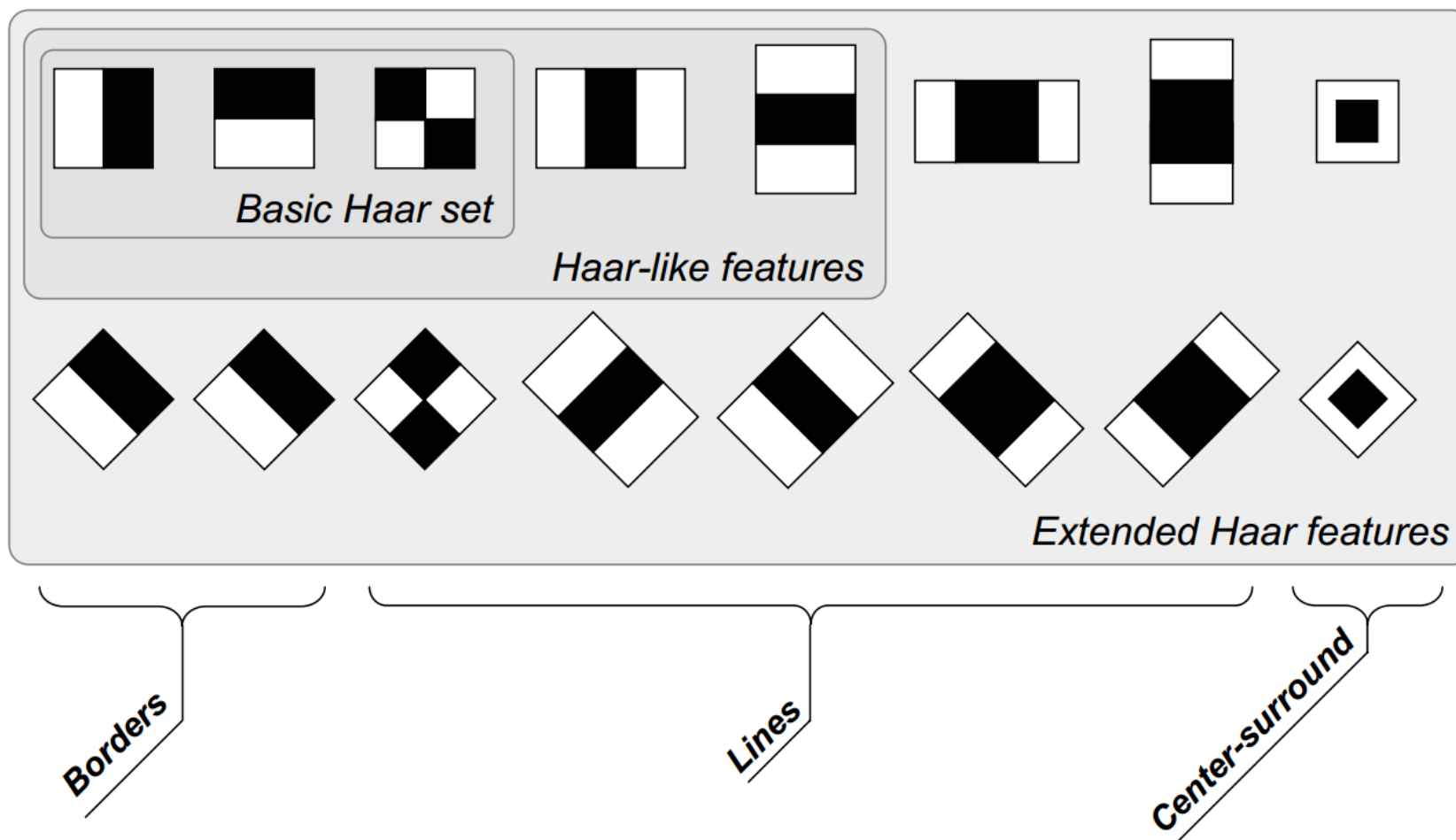


Haar-Feature = white region - black region



Features

Different sets





Face Detection

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





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



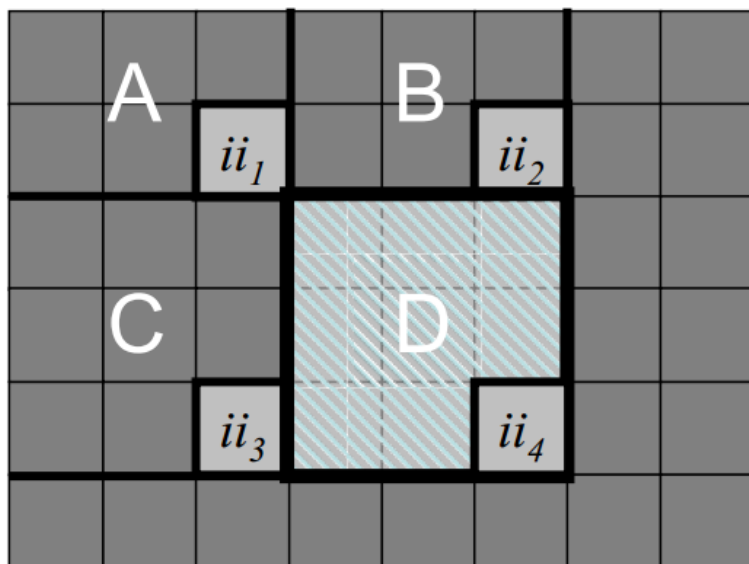
Integral image (ii)

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			

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

Integral Image

Integral image (ii)



$$ii_1 = \text{sum}(A)$$

$$ii_2 = \text{sum}(A) + \text{sum}(B)$$

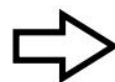
$$ii_3 = \text{sum}(A) + \text{sum}(C)$$

$$ii_4 = \text{sum}(A) + \text{sum}(B) + \text{sum}(C) + \text{sum}(D)$$

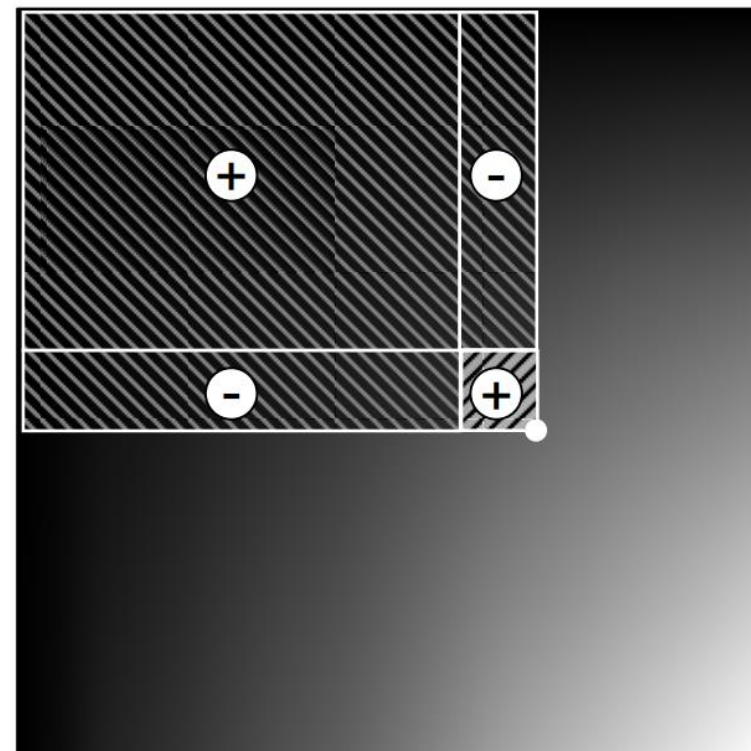
$$\text{sum}(D) = ii_4 + ii_1 - ii_2 - ii_3$$

Integral Image

Original image (I)



Integral image (ii)



Integral Image

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	153	159	163
35	72	105	138	166	194	201	206
42	87	127	167	203	239	247	253

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

54

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



1.5. Stochastic Gradient Descent

1.6. Nearest Neighbors

1.7. Gaussian Processes

1.8. Cross decomposition

1.9. Naive Bayes

1.10. Decision Trees

1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking

1.12. Multiclass and multioutput algorithms

1.13. Feature selection

1.14. Semi-supervised learning

1.15. Isotonic regression

1.16. Probability calibration

1.17. Neural network models (supervised)

2. Unsupervised learning ▾

3. Model selection and evaluation ▾

4. Inspection ▾

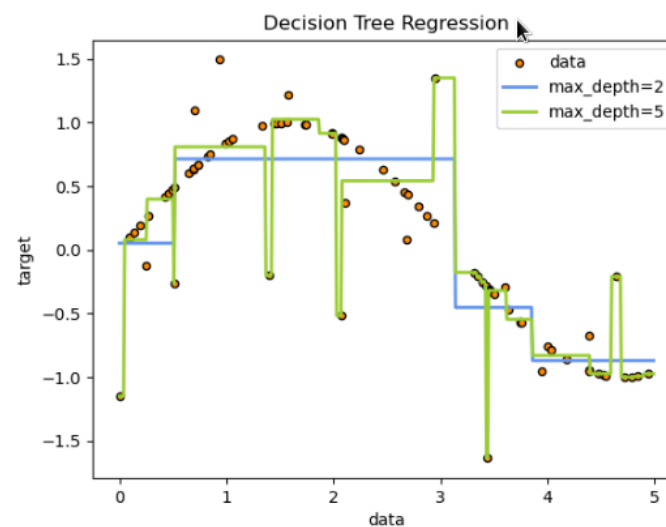
5. Visualizations

[learn.org/stable/auto_examples/tree/plot_tree_regression.html](https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html)

1.10. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for [classification](#) and [regression](#). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.





Install User Guide API Examples Community ↗ More ▾

- sklearn.multioutput ▾
- sklearn.naive_bayes ▾
- sklearn.neighbors ▾
- sklearn.neural_network ▾
- sklearn.pipeline ▾
- sklearn.preprocessing ▾
- sklearn.random_projection ▾
- sklearn.semi_supervised ▾
- sklearn.svm ▾
- sklearn.tree** ^
- DecisionTreeClassifier**
- DecisionTreeRegressor
- ExtraTreeClassifier
- ExtraTreeRegressor
- export_graphviz
- export_text
- plot_tree

🏠 > API Reference > sklearn.tree > DecisionTree...

DecisionTreeClassifier

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini',  
splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1,  
min_weight_fraction_leaf=0.0, max_features=None, random_state=None,  
max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None,  
ccp_alpha=0.0, monotonic_cst=None)
```

[\[source\]](#)

A decision tree classifier.

Read more in the [User Guide](#).

Parameters:

criterion : {"gini", "entropy", "log_loss"}, default="gini"

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see [Mathematical formulation](#).



AdaBoostRegressor

BaggingClassifier

BaggingRegressor

ExtraTreesClassifier

ExtraTreesRegressor

GradientBoostingClassifier

GradientBoostingRegressor

HistGradientBoostingClassifier

HistGradientBoostingRegressor

IsolationForest

RandomForestClassifier

RandomForestRegressor

RandomTreesEmbedding

StackingClassifier

StackingRegressor

[🏠](#) > [API Reference](#) > [sklearn.ensemble](#) > [RandomForest...](#)

RandomForestClassifier

```
class sklearn.ensemble.RandomForestClassifier(n_estimators=100, *,  
criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1,  
min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None,  
min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None,  
random_state=None, verbose=0, warm_start=False, class_weight=None,  
ccp_alpha=0.0, max_samples=None, monotonic_cst=None)
```

[\[source\]](#)

A random forest classifier.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Trees in the forest use the best split strategy, i.e. equivalent to passing `splitter="best"` to the underlying [DecisionTreeRegressor](#). The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree.



sklearn.datasets



sklearn.decomposition



sklearn.discriminant_analysis



sklearn.dummy



sklearn.ensemble



AdaBoostClassifier

AdaBoostRegressor

BaggingClassifier

BaggingRegressor

ExtraTreesClassifier

ExtraTreesRegressor

GradientBoostingClassifier

GradientBoostingRegressor

HistGradientBoostingClassifier

HistGradientBoostingRegressor

IsolationForest

RandomForestClassifier

RandomForestRegressor

AdaBoostClassifier

```
class sklearn.ensemble.AdaBoostClassifier(estimator=None, *,  
n_estimators=50, learning_rate=1.0, algorithm='SAMME.R', random_state=None)
```

An AdaBoost classifier.

[\[source\]](#)

An AdaBoost [\[1\]](#) classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

This class implements the algorithm based on [\[2\]](#).

Read more in the [User Guide](#).

! Added in version 0.14.

Parameters:

estimator : *object, default=None*

The base estimator from which the boosted ensemble is built. Support for sample weighting is required, as well as proper `classes_` and `n_classes_` attributes. If `None`, then the base estimator is [DecisionTreeClassifier](#) initialized with `max_depth=1`.



Topics to be discussed:

- What is Gini index?
- Can we use Random Forest or AdaBoost for feature selection?
- What can be used as estimator in AdaBoost?
- What is decision stump?

```
# Create and fit an AdaBoosted decision tree  
bdt = AdaBoostClassifier(  
    DecisionTreeClassifier(max_depth=1), algorithm="SAMME", n_estimators=200  
)
```

- **let's see next slides with visualization for better understanding**

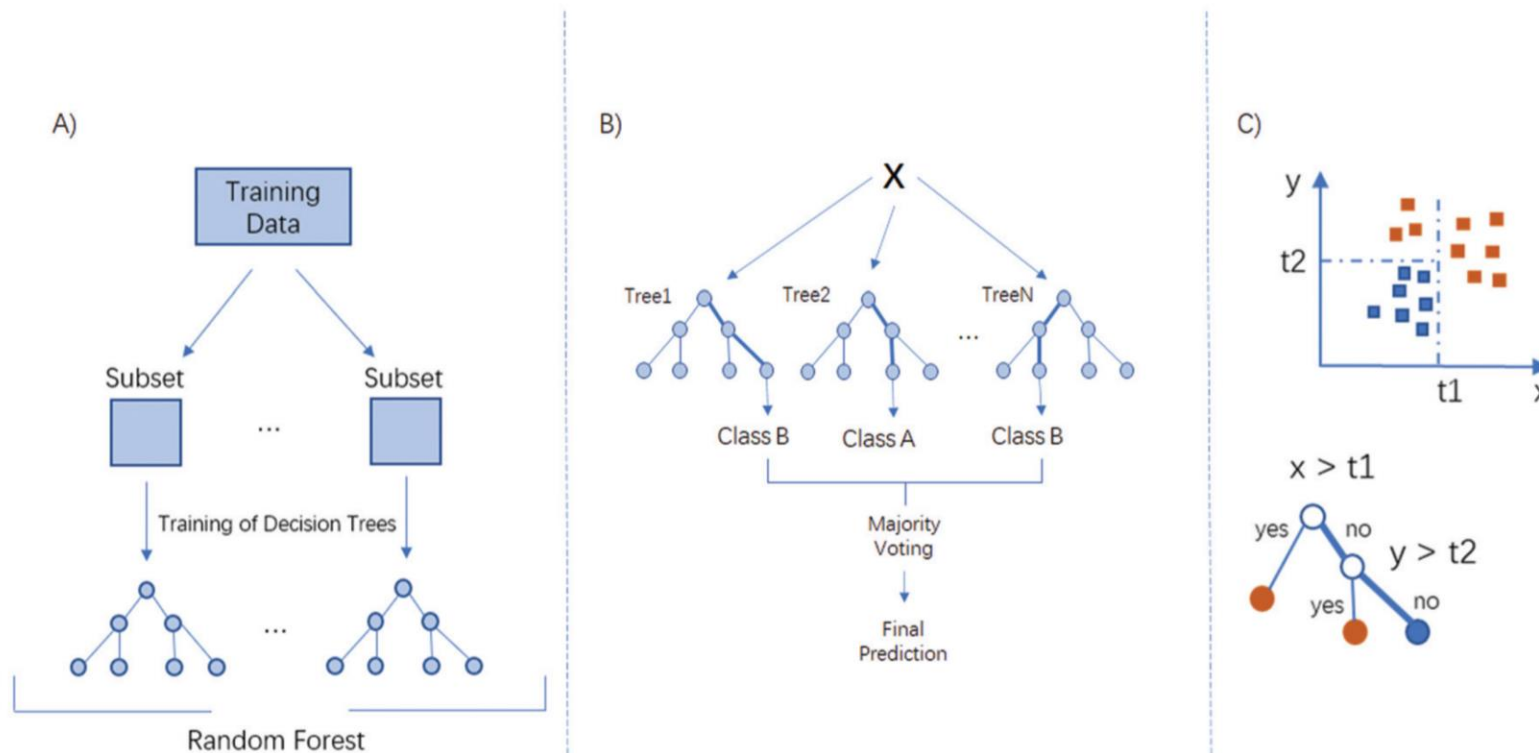


Fig. 1 Simplified Random Forests method. (A) Features in the training data are divided into multiple subsets and used to train individual decision trees in the forest. (B) When an unknown input X is introduced to the trained forest, each tree will make its own prediction. The final prediction is decided by majority voting of all trees. (C) An example of how a decision tree makes a prediction, where blue squares represent samples in class A while orange squares represent samples in class B, and $t1$ and $t2$ are two example features used at nodes.

Topics to be discussed:

- What is Gini index (Gini impurity)?

Gini impurity [\[edit\]](#)

Gini impurity, **Gini's diversity index**,^[26] or **Gini-Simpson Index** in biodiversity research, is named after Italian mathematician **Corrado Gini** and used by the CART (classification and regression tree) algorithm for classification trees. Gini impurity measures how often a randomly chosen element of a set would be incorrectly labeled if it were labeled randomly and independently according to the distribution of labels in the set. It reaches its minimum (zero) when all cases in the node fall into a single target category.

For a set of items with J classes and relative frequencies p_i , $i \in \{1, 2, \dots, J\}$, the probability of choosing an item with label i is p_i , and the probability of miscategorizing that item is $\sum_{k \neq i} p_k = 1 - p_i$.

The Gini impurity is computed by summing pairwise products of these probabilities for each class label:

$$I_G(p) = \sum_{i=1}^J \left(p_i \sum_{k \neq i} p_k \right) = \sum_{i=1}^J p_i (1 - p_i) = \sum_{i=1}^J (p_i - p_i^2) = \sum_{i=1}^J p_i - \sum_{i=1}^J p_i^2 = 1 - \sum_{i=1}^J p_i^2.$$

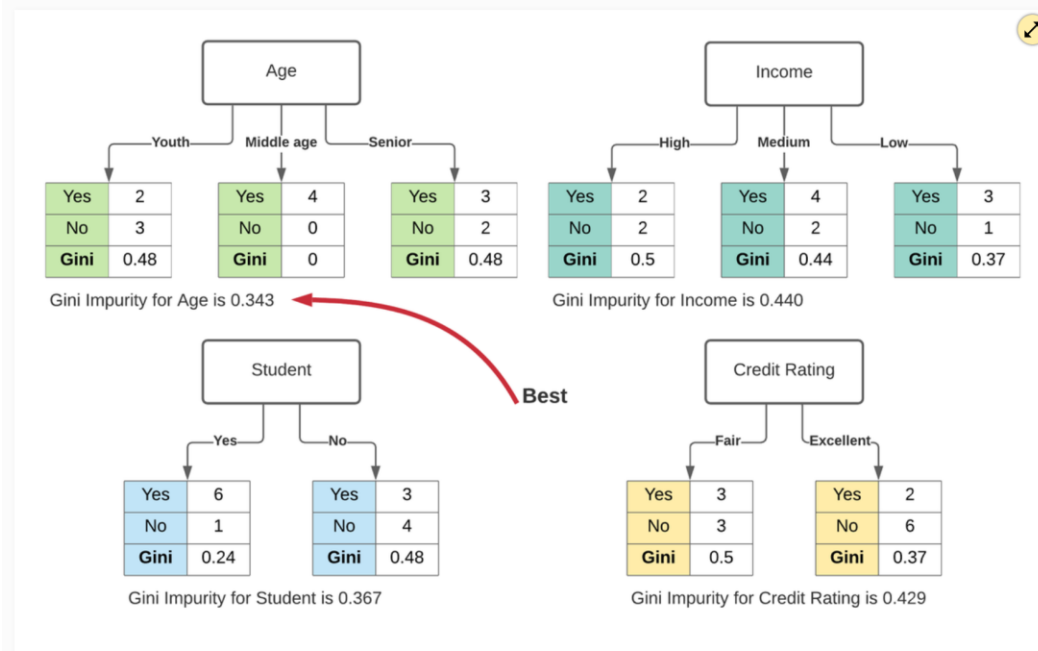
The Gini impurity is also an information theoretic measure and corresponds to **Tsallis Entropy** with deformation coefficient $q = 2$, which in physics is associated with the lack of information in out-of-equilibrium, non-extensive, dissipative and quantum systems. For the limit $q \rightarrow 1$ one recovers the usual Boltzmann-Gibbs or Shannon entropy. In this sense, the Gini impurity is nothing but a variation of the usual entropy measure for decision trees.

https://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity

<https://www.learndatasci.com/glossary/gini-impurity/>

What is Gini Impurity?

Gini Impurity is a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree. More precisely, the Gini Impurity of a dataset is a number between 0-0.5, which indicates the likelihood of new, random data being misclassified if it were given a random class label according to the class distribution in the dataset.



For example, say you want to build a classifier that determines if someone will default on their credit card. You have some labeled data with features, such as bins for age, income, credit rating, and whether or not each person is a student. To find the best feature for the first split of the tree – the root node – you could calculate how poorly each feature divided the data into the correct class, default ("yes") or didn't default ("no"). This calculation would measure the **impurity** of the split, and the feature with the lowest impurity would determine the best feature for splitting the current node. This process would continue for each subsequent node using the remaining features.

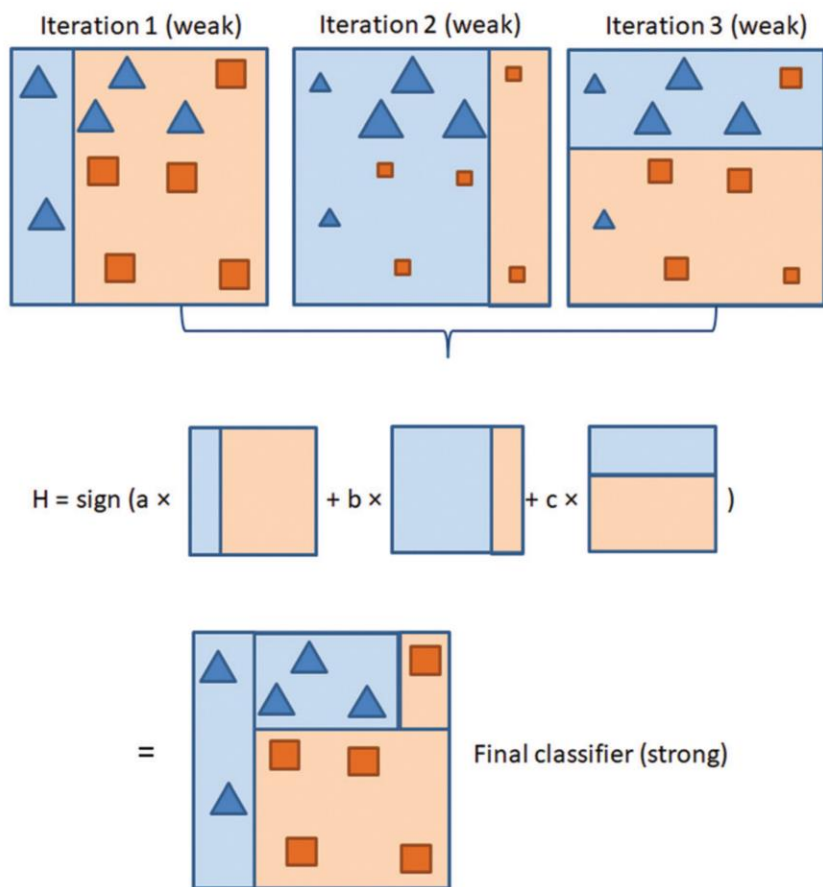


Fig. 2 A schematic of the AdaBoost process. Blue triangles and orange squares represent features, with the size of the features representing weighting. Iteration 1, all features carry equal weight. Iteration 2, correctly classified features (in iteration 1) down-weighted, incorrectly classified feature up-weighted. Iteration 3, correctly classified features (in iteration 2) down-weighted, incorrectly classified feature up-weighted. The combination of the iterations produces a final classifier that is strong.

Procedure: AdaBoost Learning Algorithm

- 1 **Given:** $(x_1, y_1) \dots (x_m, y_m), x_i \in X, y_i \in \{-1, 1\}$
- 2 **Initialize weight** $D_1(i) = \frac{1}{m}$
- 3 **For** $t = 1 \dots T$
Call weak learn which returns weak classifier $h_t : X \in \{-1, 1\}$ with minimum error w.r.t D_t
Choose $\alpha_t \in R$
Update weight

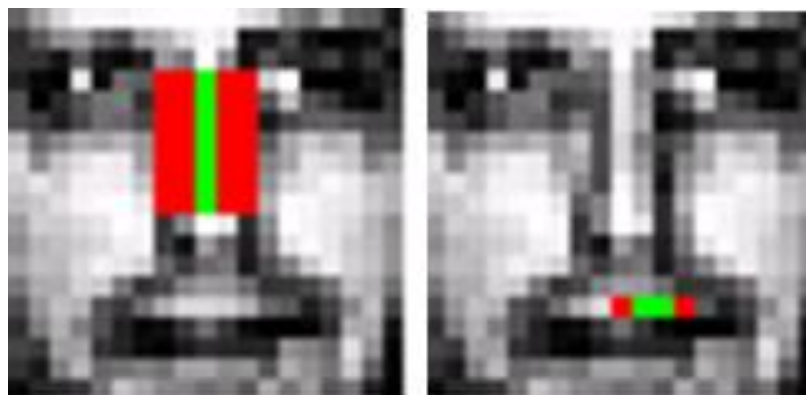
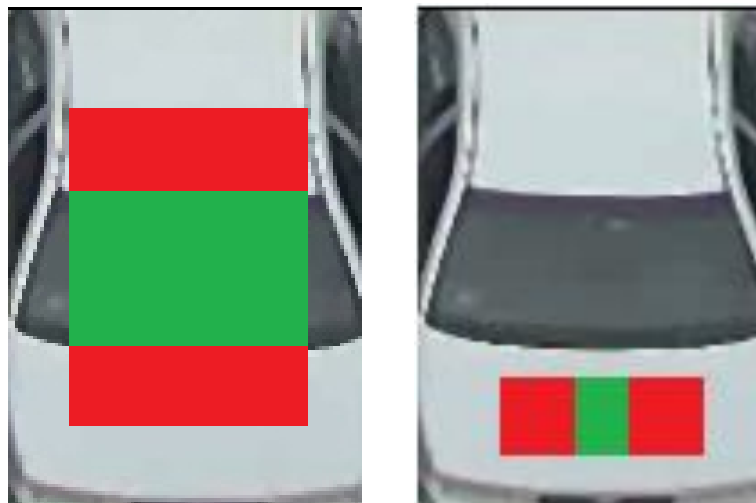
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

When Z_t is a normalization factor chosen so that D_{t+1} is a distribution
- 4 **Produce the strong classifier:**

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$



Feature Selection





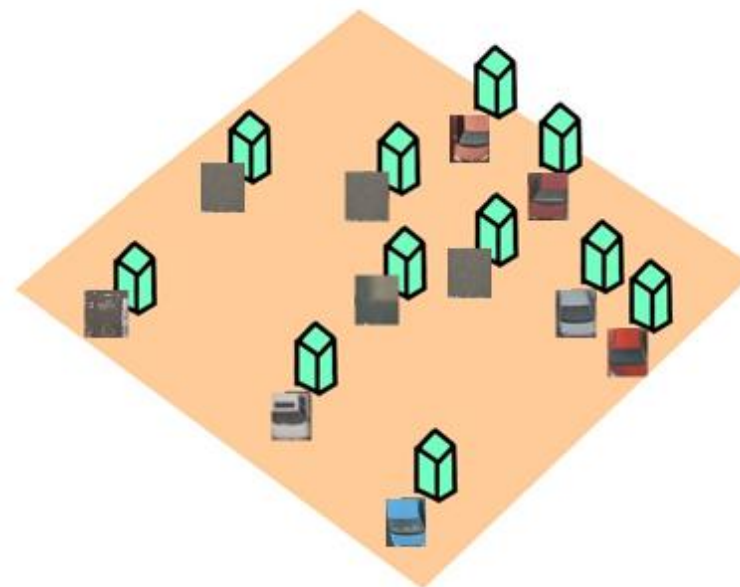
Feature Selection

- 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

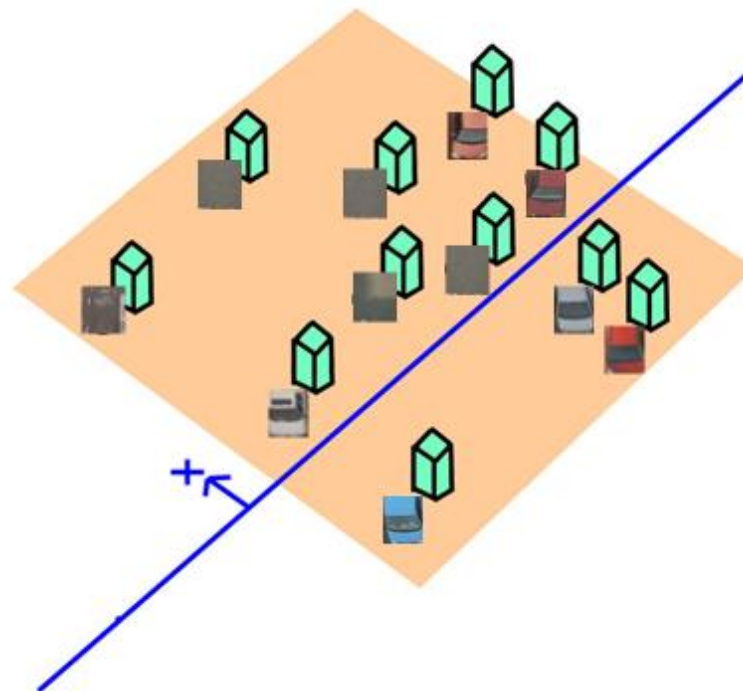
□ AdaBoost starts with a uniform distribution of “weights” over training examples.





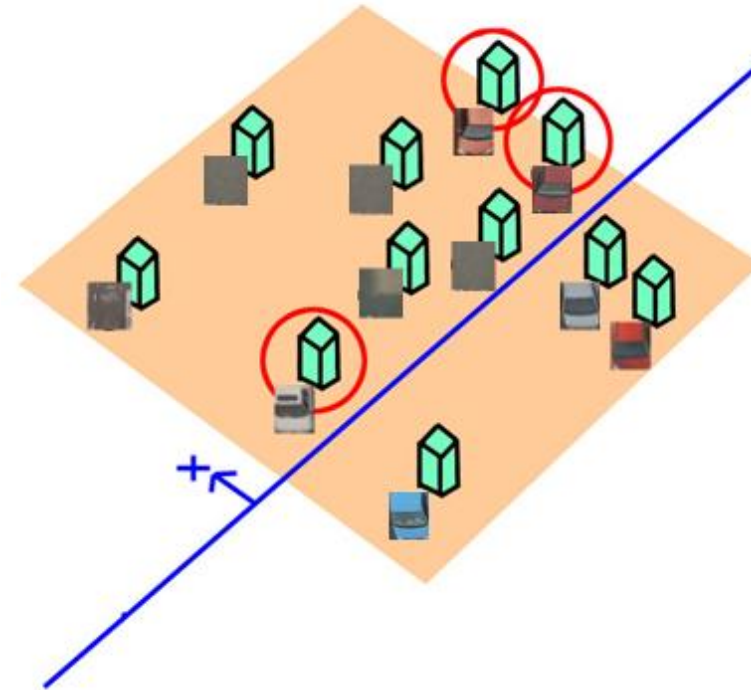
Feature Selection

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



Feature Selection

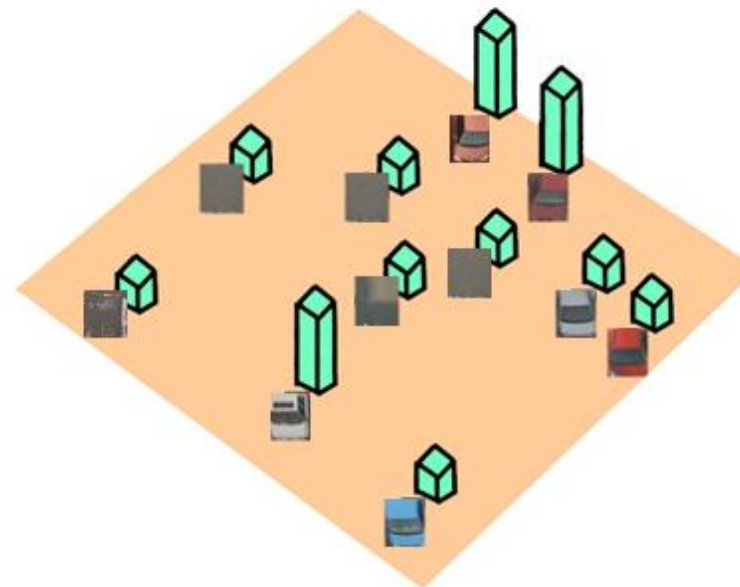
- ❑ AdaBoost starts with a uniform distribution of “weights” over training examples.
- ❑ Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- ❑ Increase the weights on the training examples that were misclassified.





Feature Selection

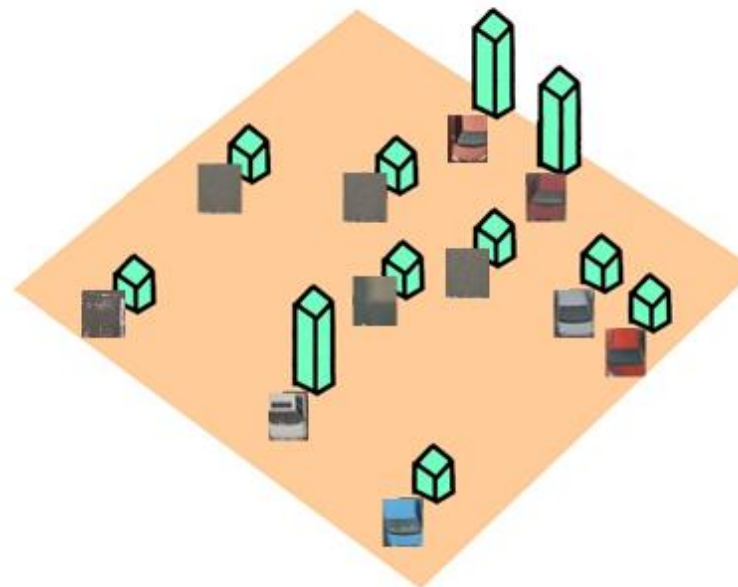
- ❑ AdaBoost starts with a uniform distribution of “weights” over training examples.
- ❑ Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- ❑ Increase the weights on the training examples that were misclassified.





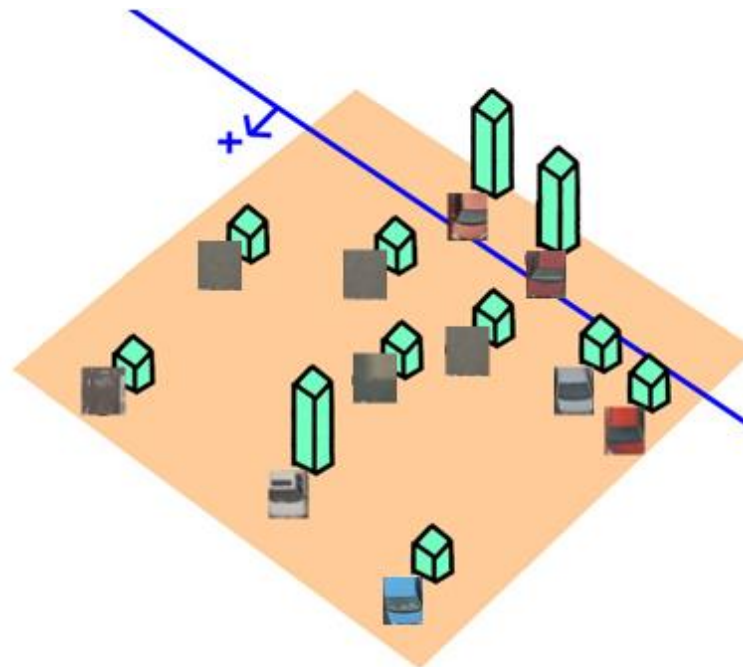
Feature Selection

- ❑ 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)



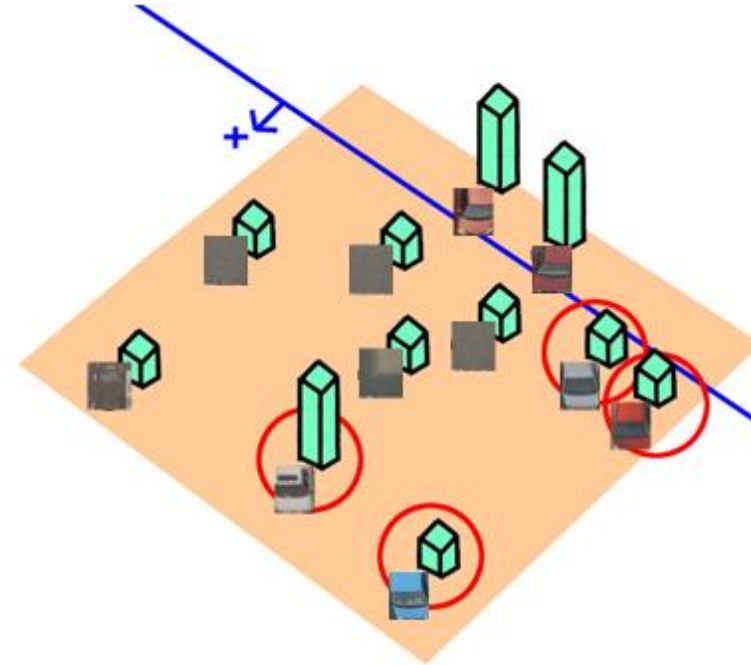
Feature Selection

- ❑ 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)



Feature Selection

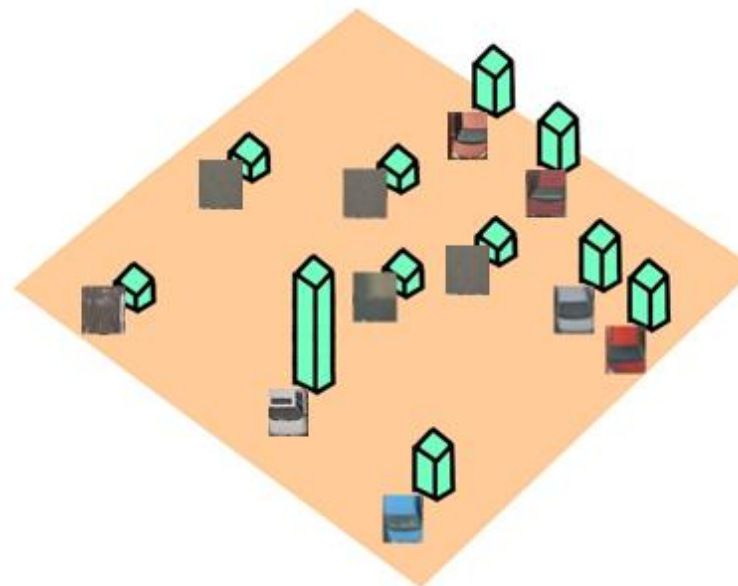
- ❑ AdaBoost starts with a uniform distribution of “weights” over training examples.
- ❑ Select the classifier with the lowest weighted error (i.e. a “weak” classifier)
- ❑ Increase the weights on the training examples that were misclassified.
- ❑ (Repeat)





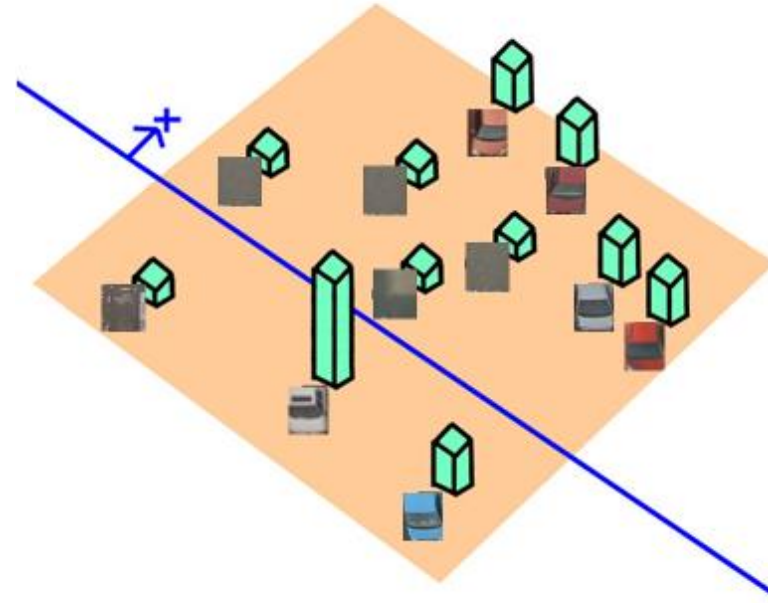
Feature Selection

- ❑ 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)



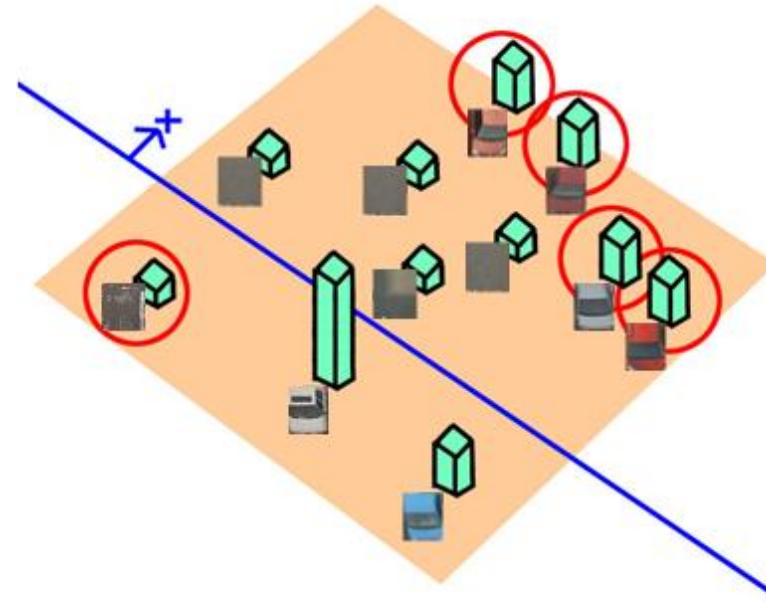
Feature Selection

- ❑ 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)



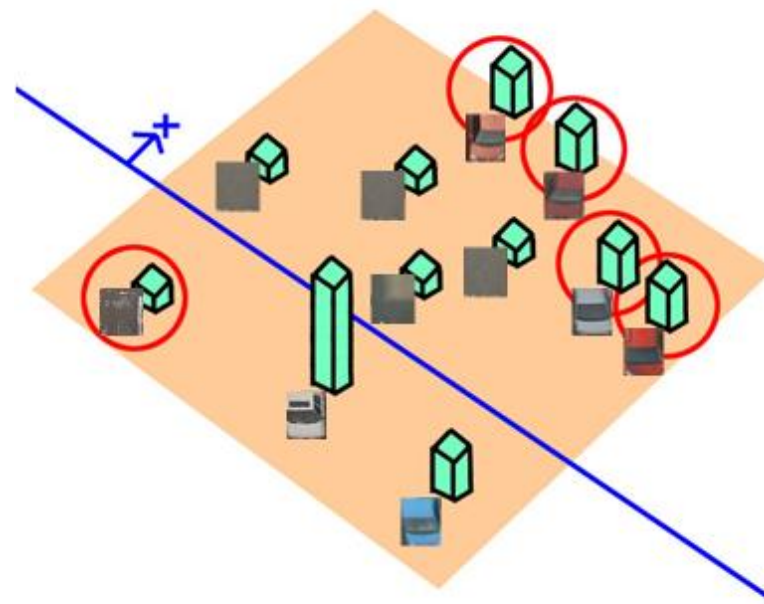
Feature Selection

- ❑ 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)





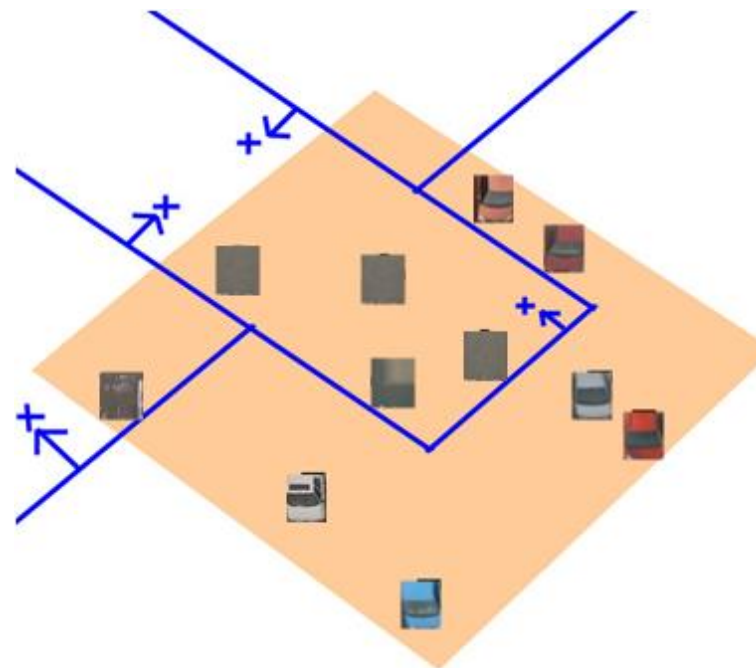
Feature Selection



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



Feature Selection



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



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

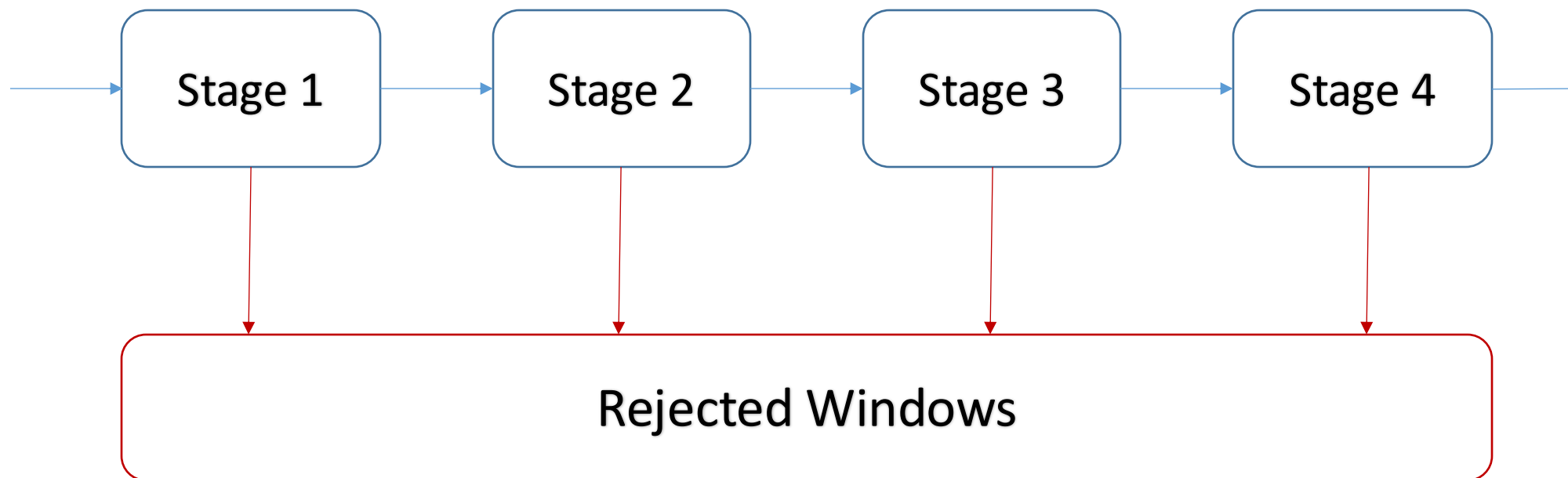
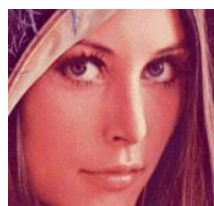


MINISTRY OF EDUCATION,
YOUTH AND SPORTS



Cascade of Classifier

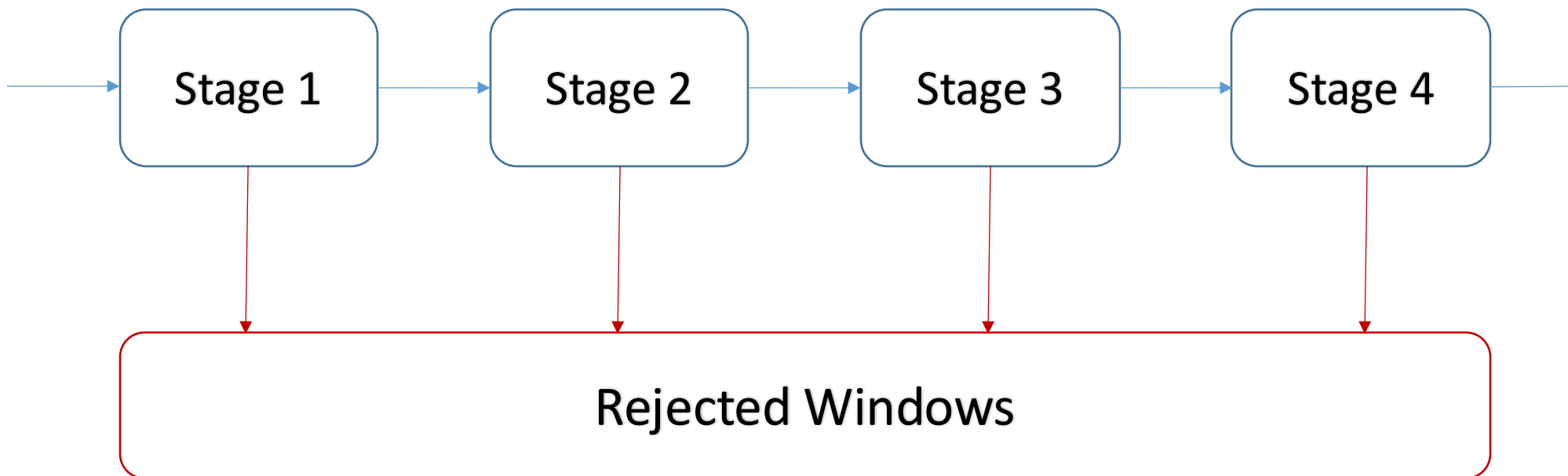
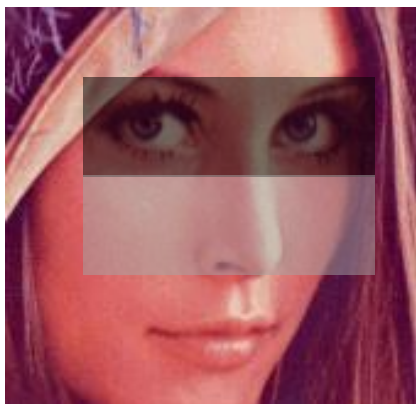
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

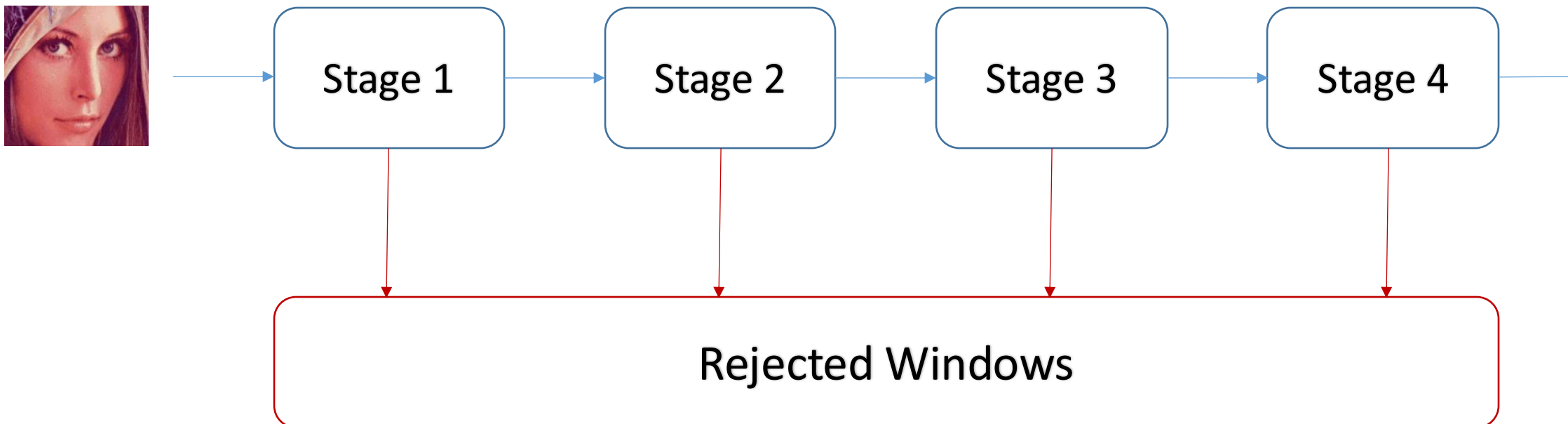
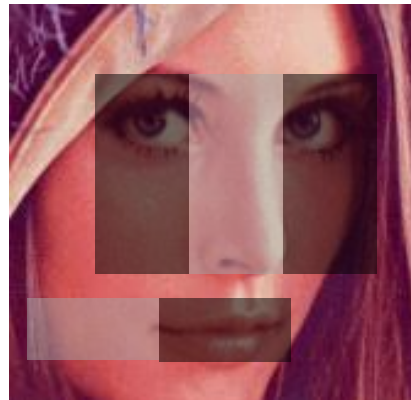
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

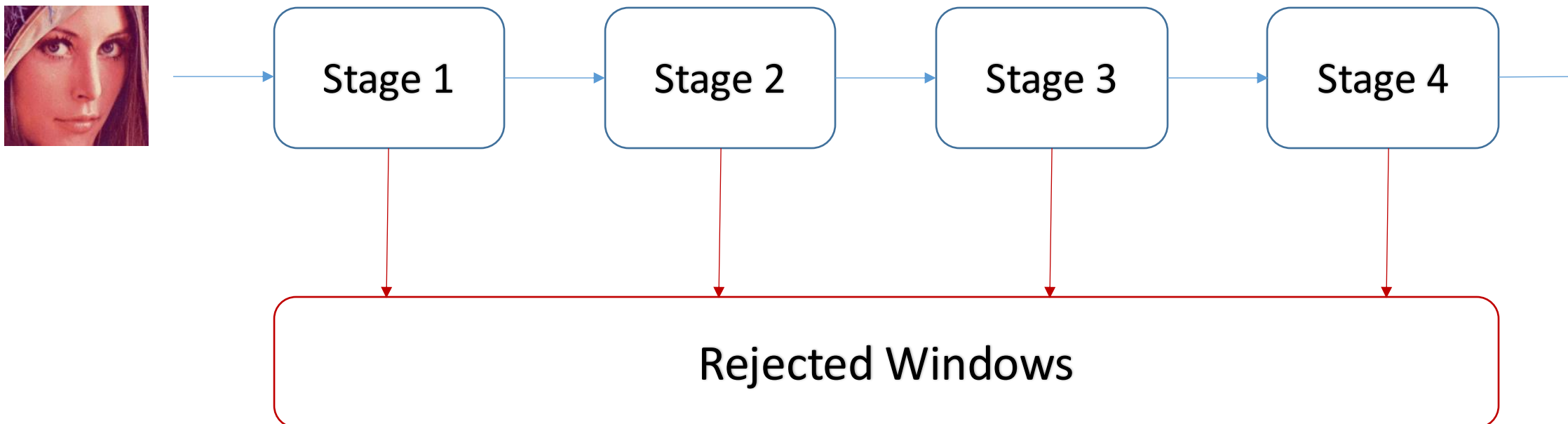
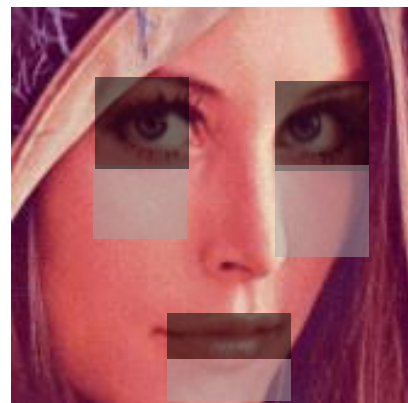
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

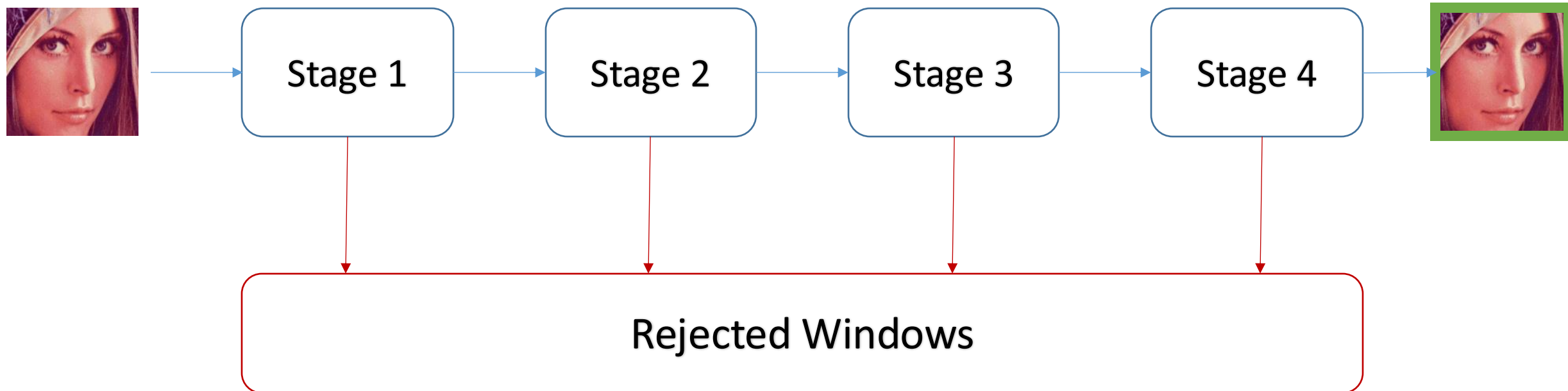
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

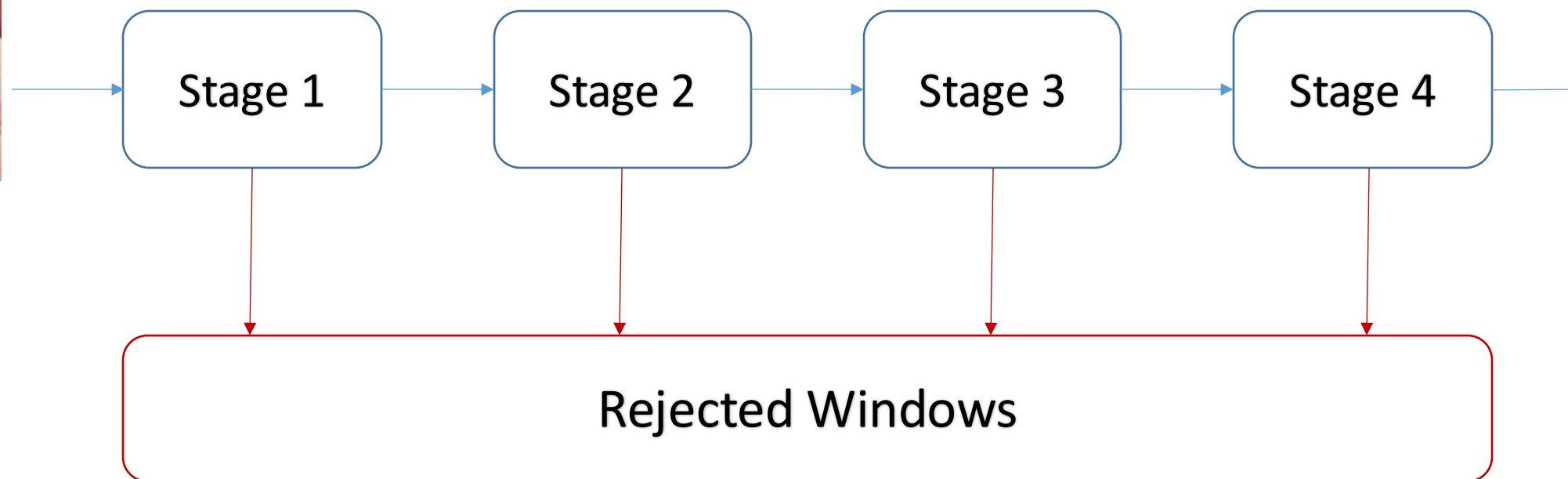
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

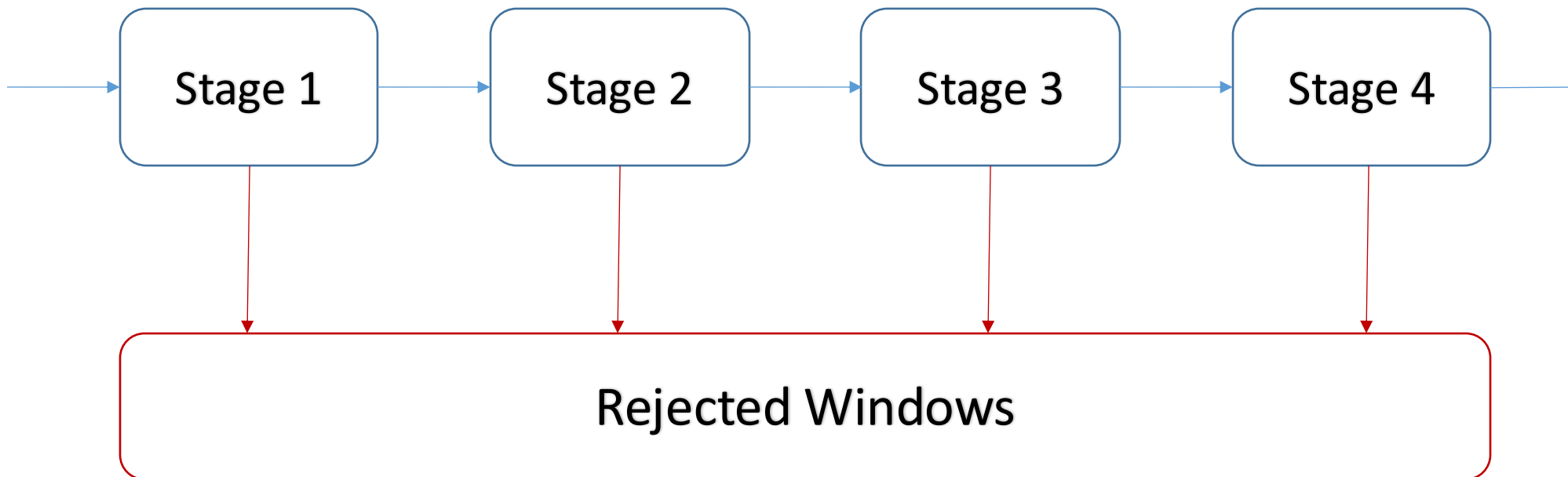
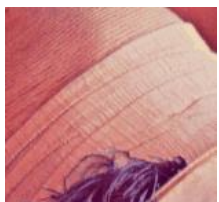
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

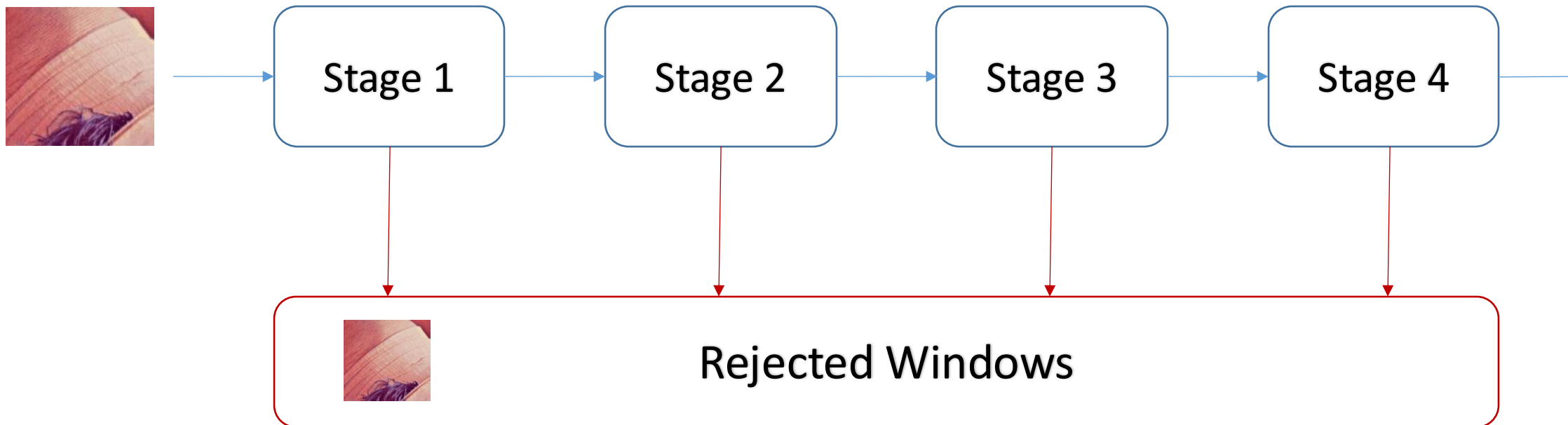
The idea of cascade classifier is reject the non-face region as soon as possible





Cascade of Classifier

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



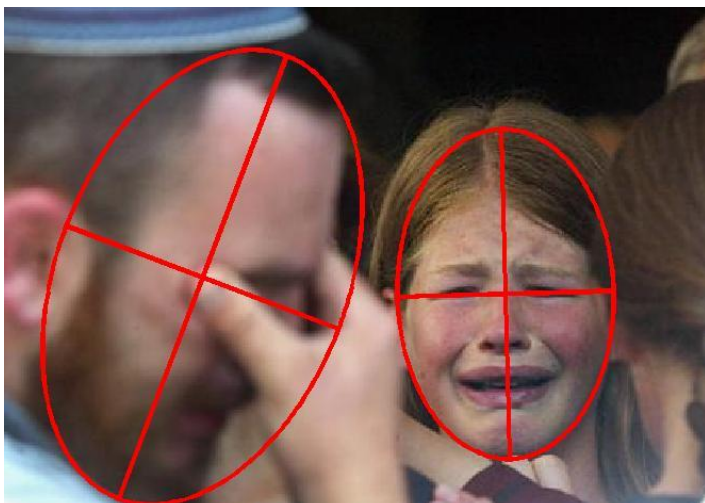


Haar Features





Face Detection (Object Detection) - Evaluation





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) = $2 \times \text{precision} \times \text{sensitivity} / (\text{precision} + \text{sensitivity})$

What else?



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.

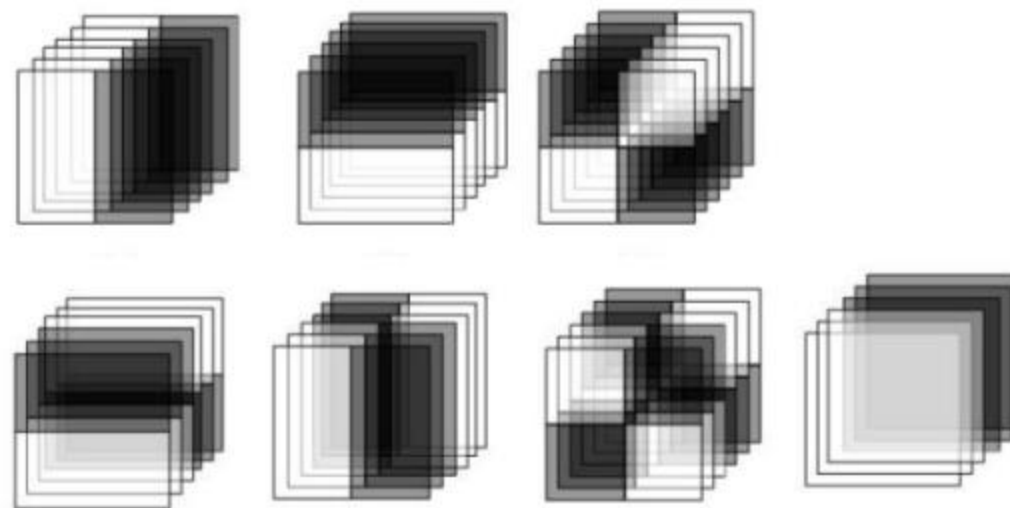


Haar Features

- Since Viola and Jones popularized the Haar-like features for face detection, the Haarlike features and their modifications 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 classifier. 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 filters in 3D space (instead of using rectangle filters in 2D space) to capture motion information. The 3D features are then combined with the SVM classifier. 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.



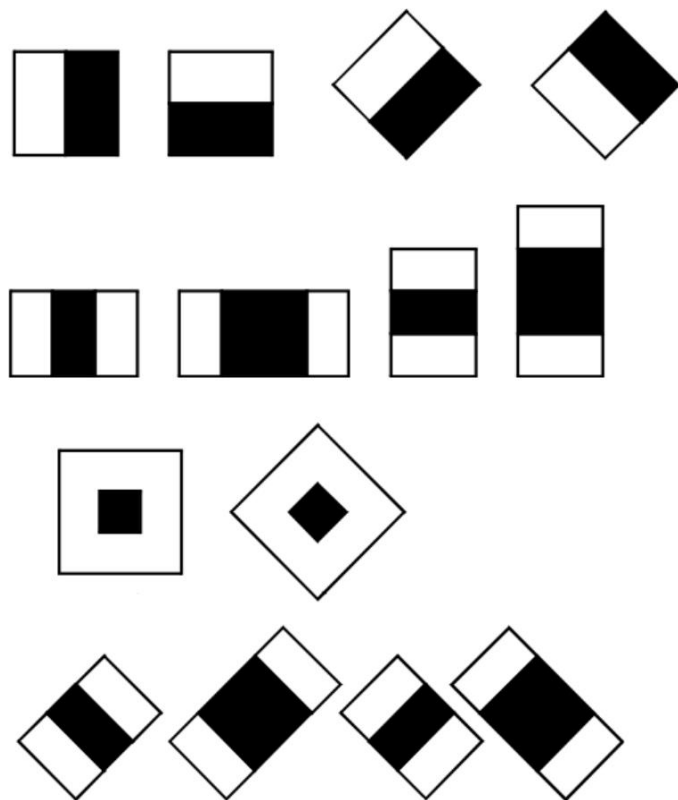


Haar Features

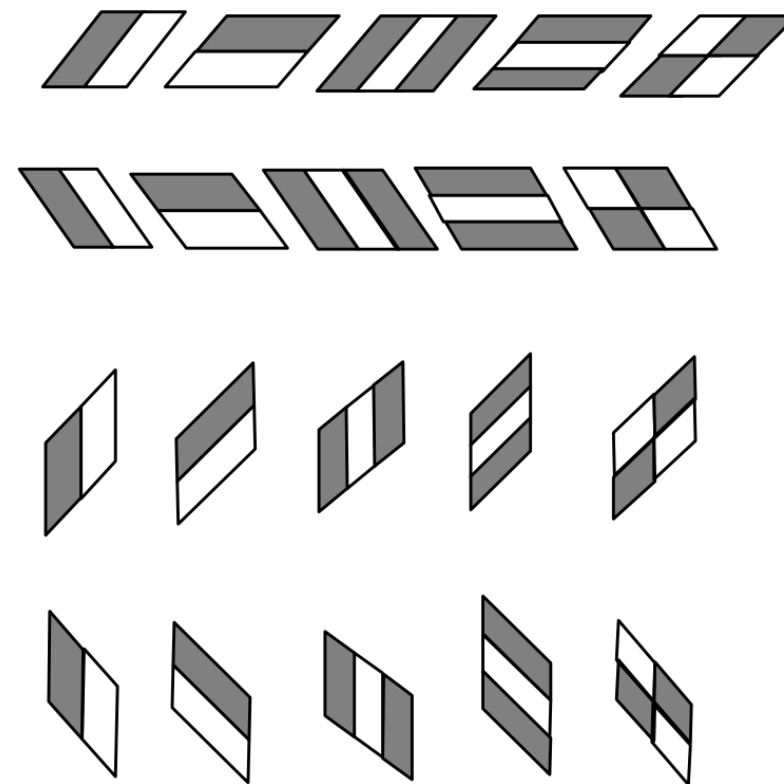
- [1] Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Trans. Pattern Anal. Mach. Intell. 23(4), 349–361 (Apr 2001), <http://dx.doi.org/10.1109/34.917571>
- [2] Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. 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.



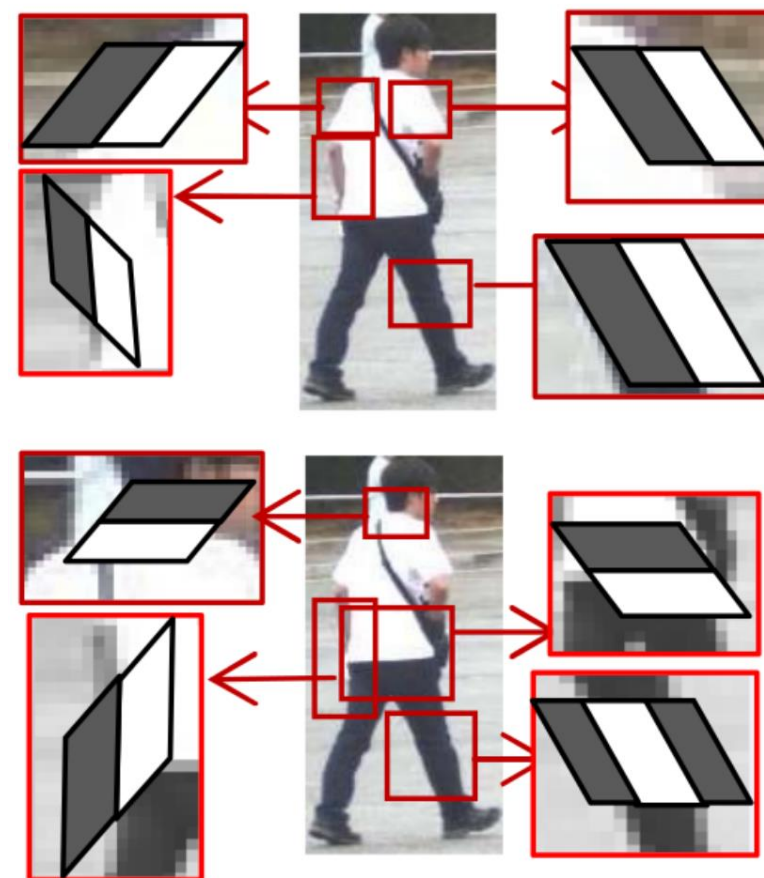
Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Image Processing. 2002. Proceedings. 2002 International Conference on. vol. 1, pp. 1-900-1-903 vol.1 (2002)



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)

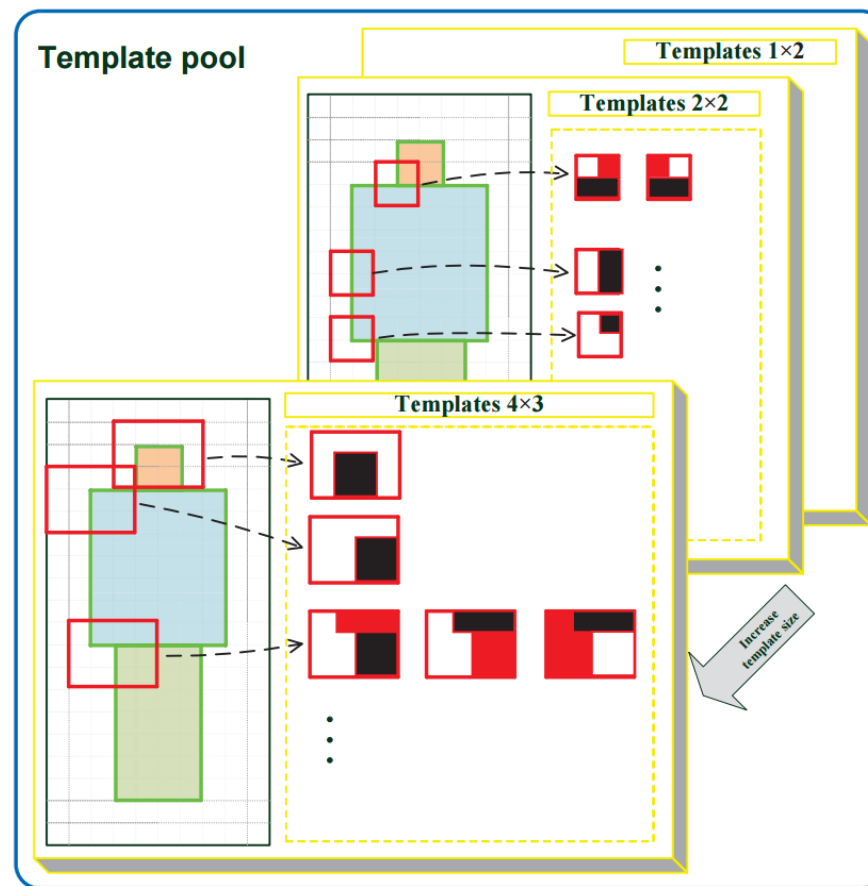
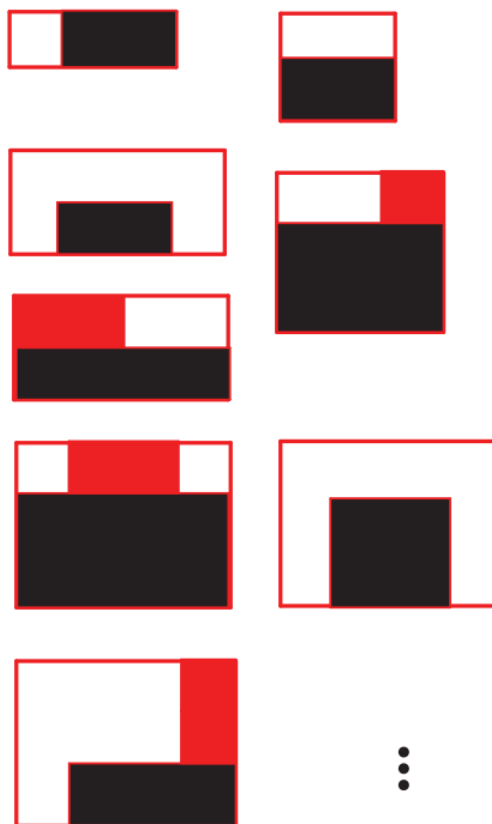
Haar Features

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



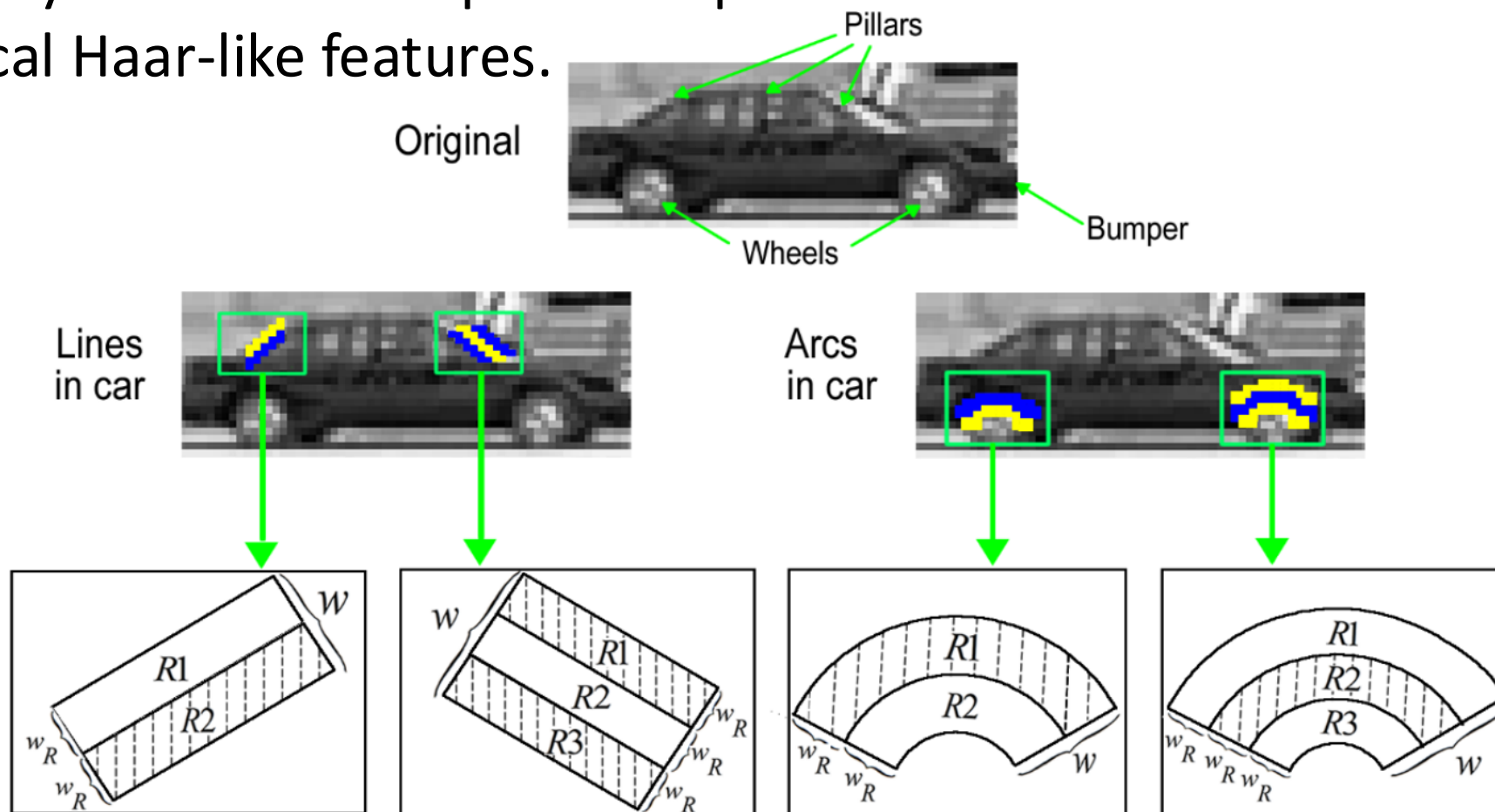
Haar Features

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



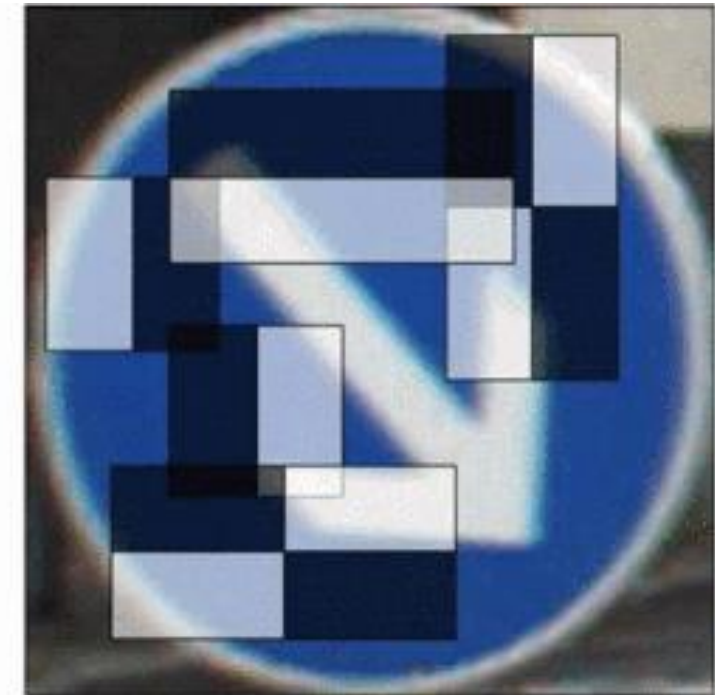
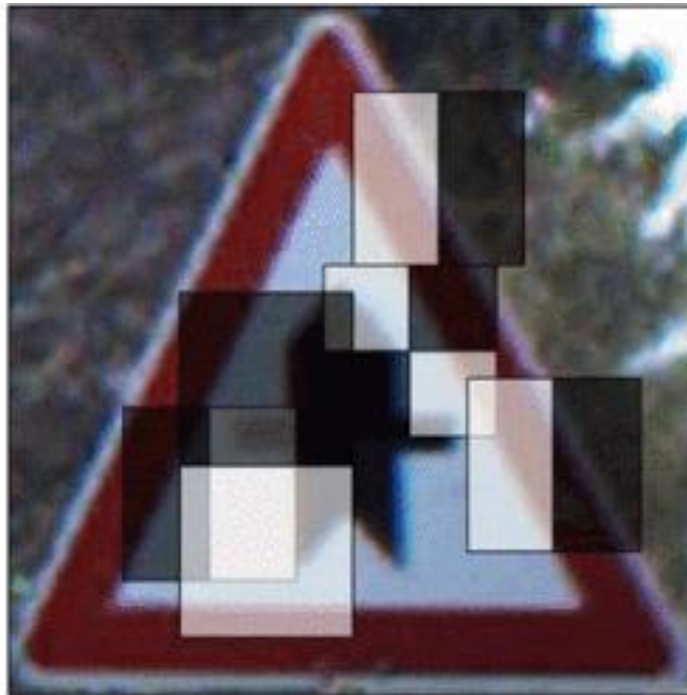
Haar Features

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





Haar Features



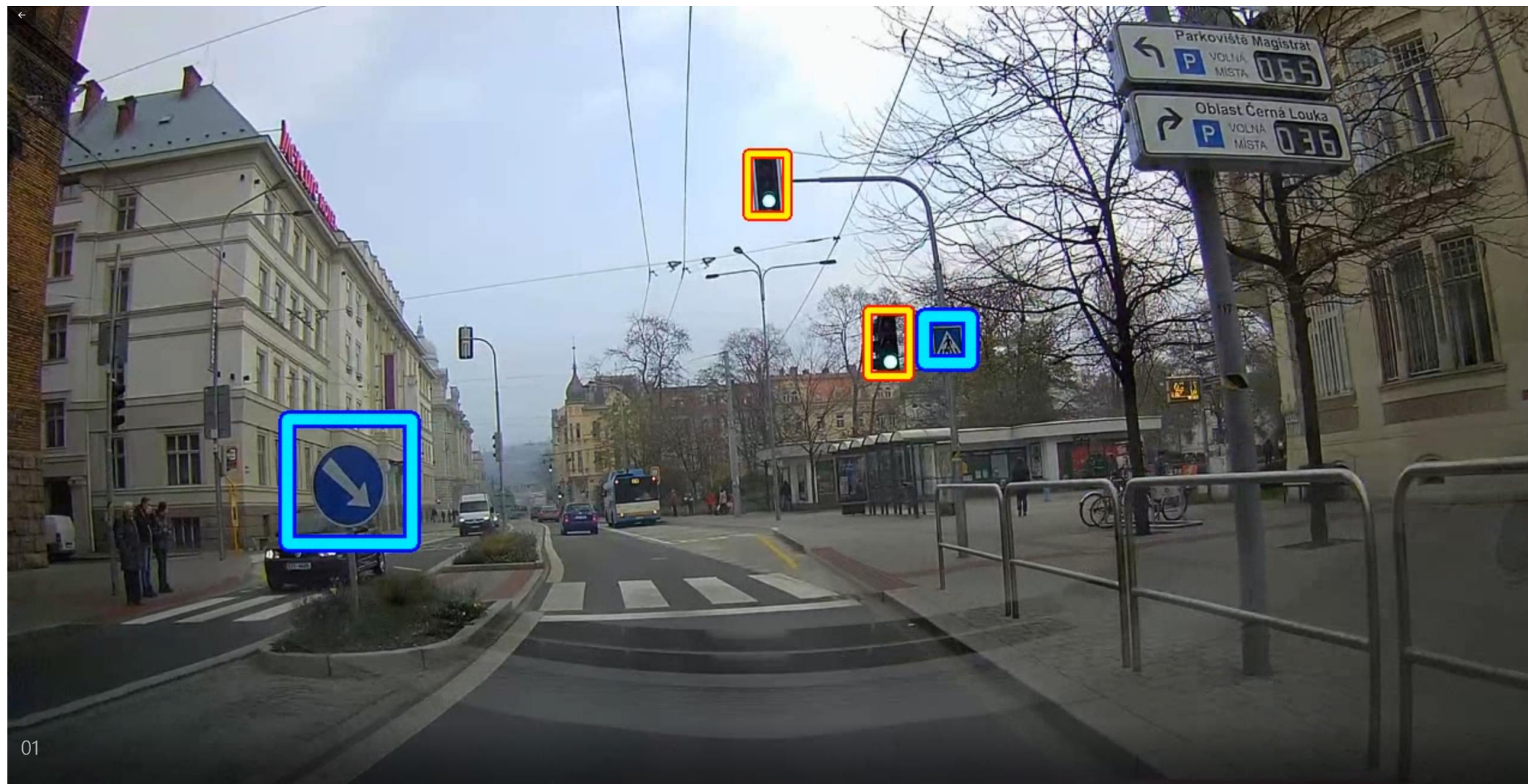
Five most significant Haar features selected for the first stage of each of the 3 trained detectors

<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



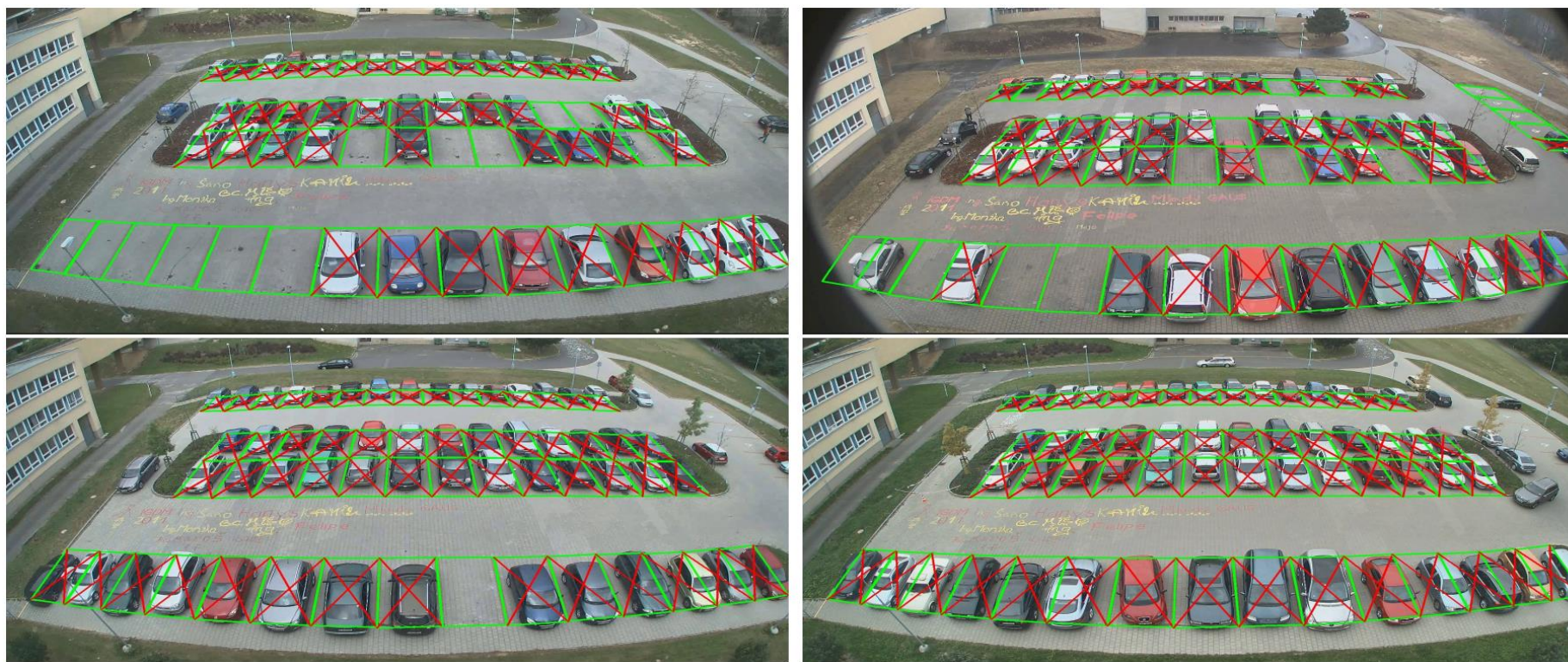
Haar Features



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

- SIFT, SURF

- CNNs



Traditional Approaches



KeyPoints



Deep Learning Approach



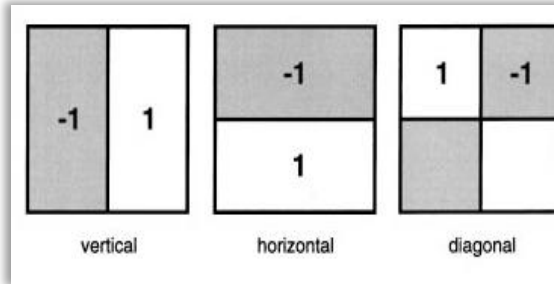
Related Works

2000

Papageorgiou
(2000)

A Trainable System for Object Detection

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

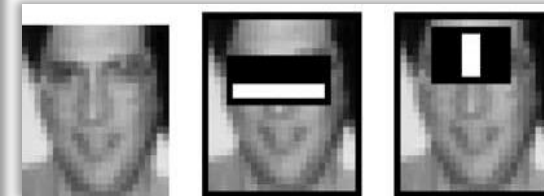


Viola, Jones
(2001,2004)

Robust Real-Time Face Detection

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

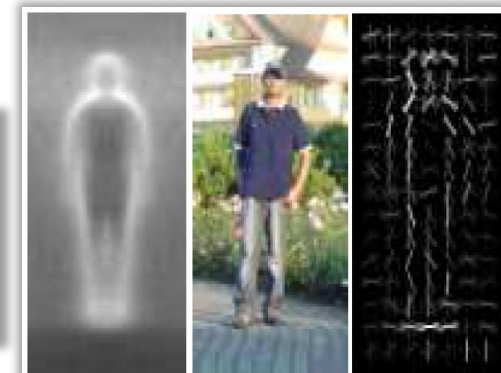
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs
(2005)
cit. 10947

Histograms of Oriented Gradients for Human Detection

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



2005



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



Object Detection (Analysis)

HOG



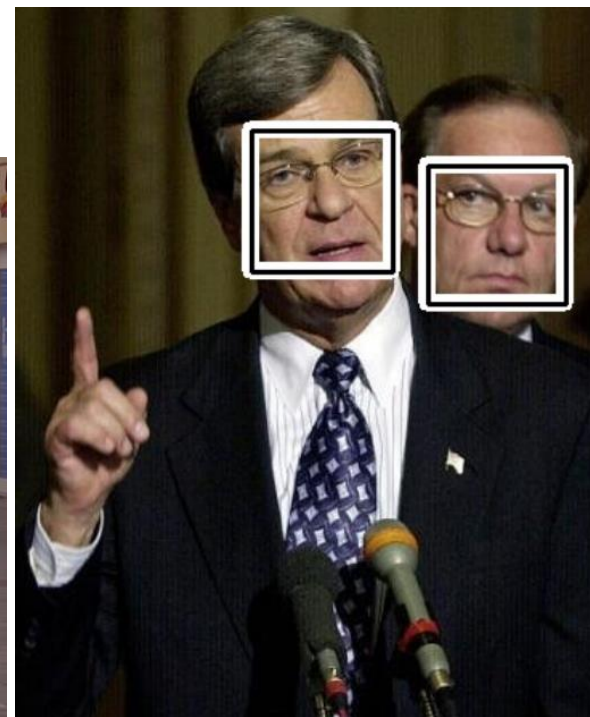
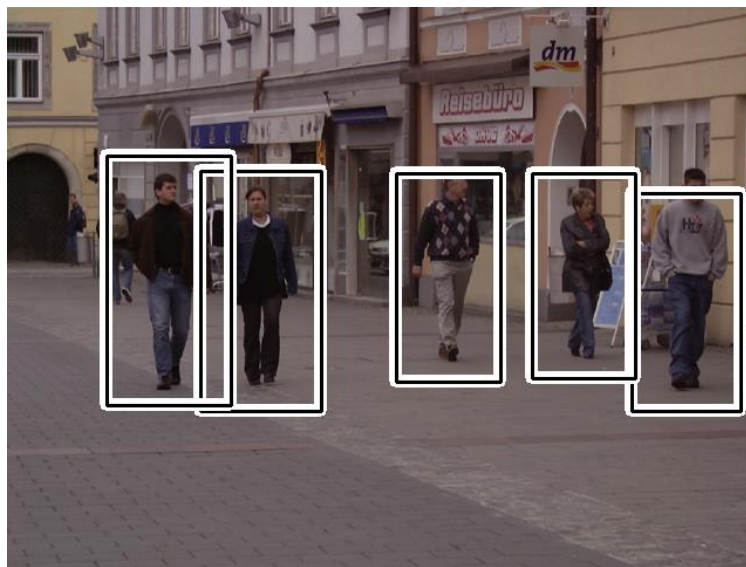
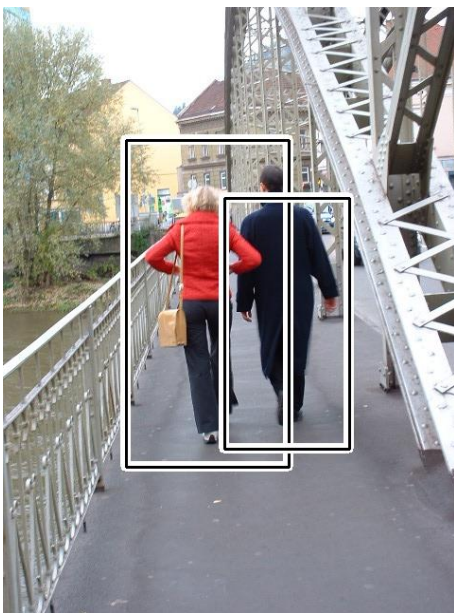
What is Object Detection?

- 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



Pedestrian Detection - Challenges?





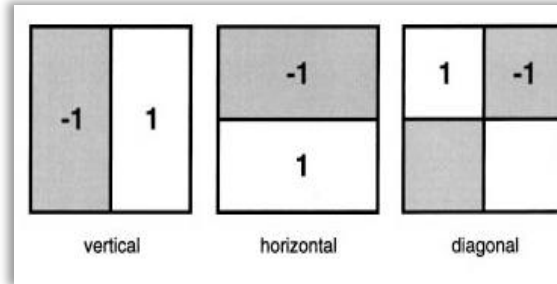
Related Works

2000

Papageorgiou
(2000)

A Trainable System for Object Detection

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

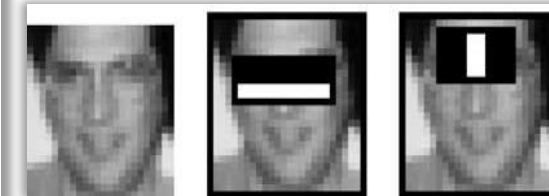


Viola, Jones
(2001,2004)

Robust Real-Time Face Detection

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

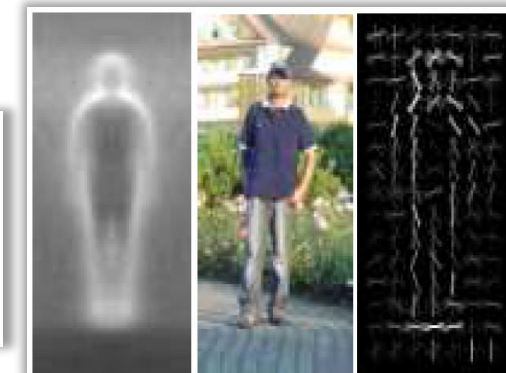
MICHAEL J. JONES
Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA
mjones@merl.com



Dalal, Triggs
(2005)

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alpes, 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-the-art 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.



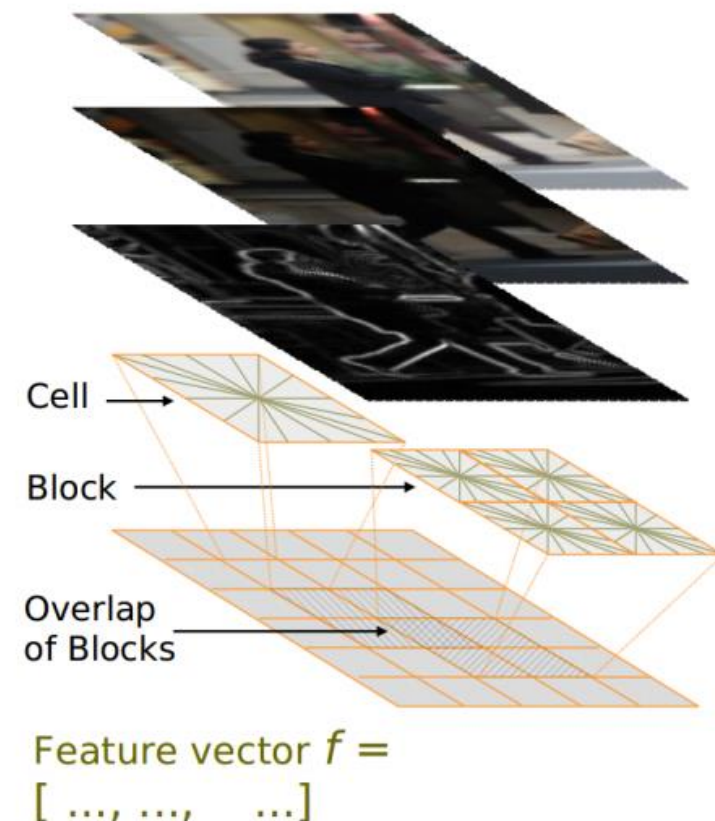
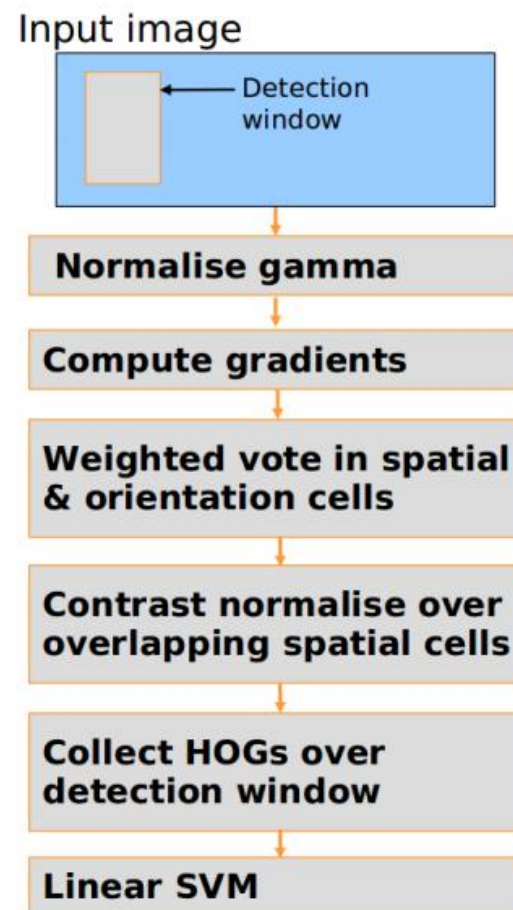
Histograms of Oriented Gradients (HOG)

- 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.



- 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 .

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

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

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



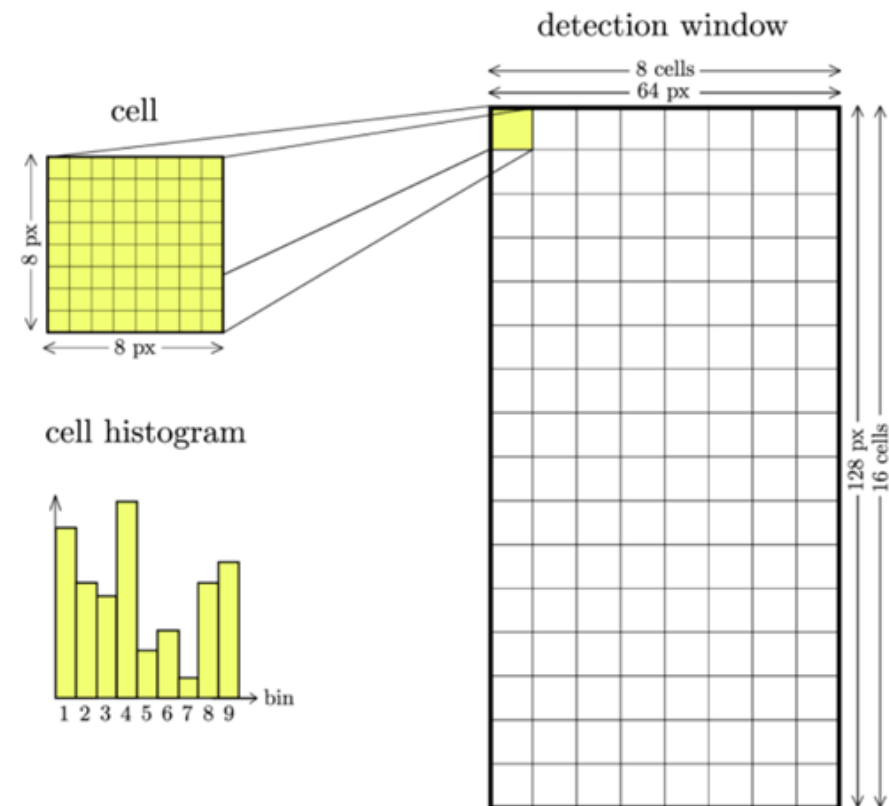
- In the next step, the image is divided into the cells and the cell histograms are constructed. The histogram bins are spread over 0 to 180 degrees or 0 to 360 degrees. The corresponding histogram bin is found for each pixel inside the cell. Each pixel contributes a weighted vote for its corresponding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

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
- Final Vector: Collect HOG blocks into vector





- 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 to 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.



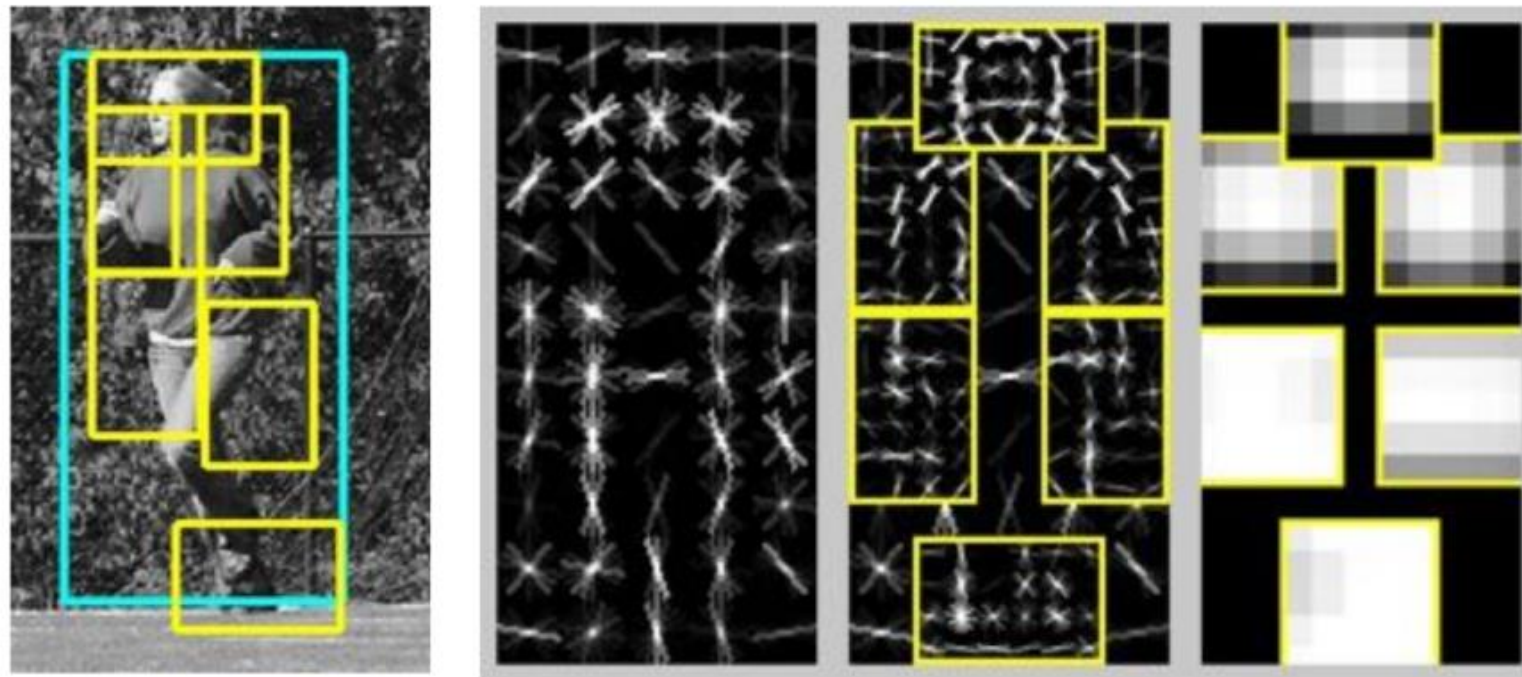
In [1], the authors applied the principal component analysis (PCA) to the HOG feature vector to obtain the PCA-HOG vector. This vector contains the subset of HOG features and the vector is used as an input for the SVM classifier. Their method was used for pedestrian detection with the satisfactory results.

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)

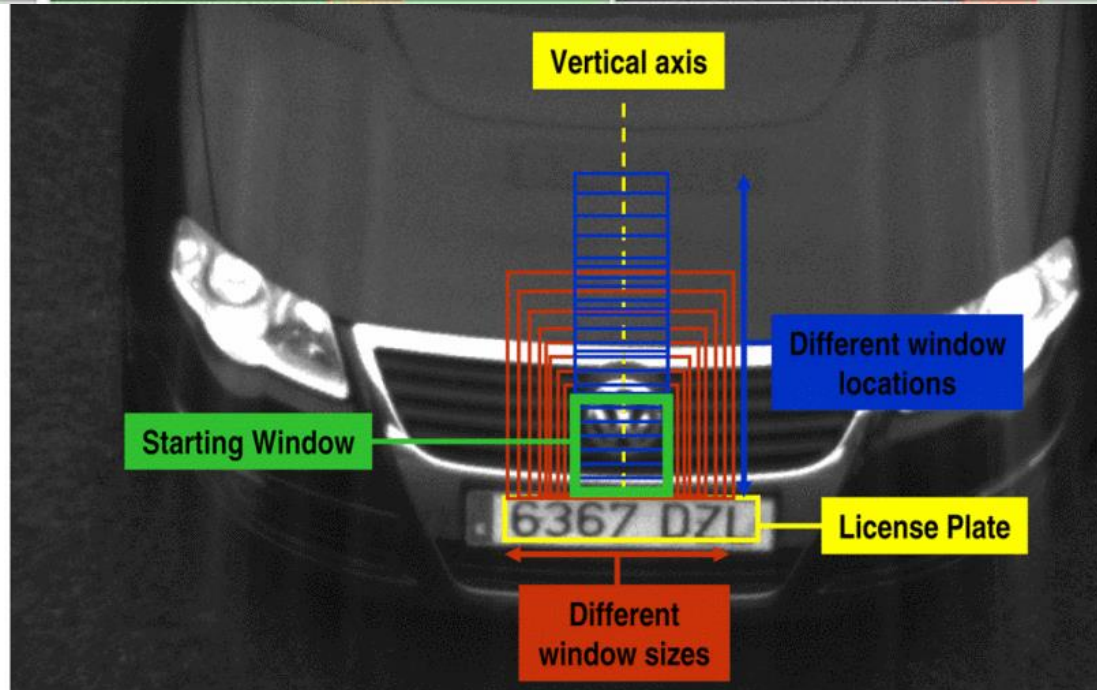


An example of person detection using a part model. The model is defined by the coarse global template that covers the entire object and higher resolution part templates. The templates represent the histogram of oriented gradient [2].

- [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)



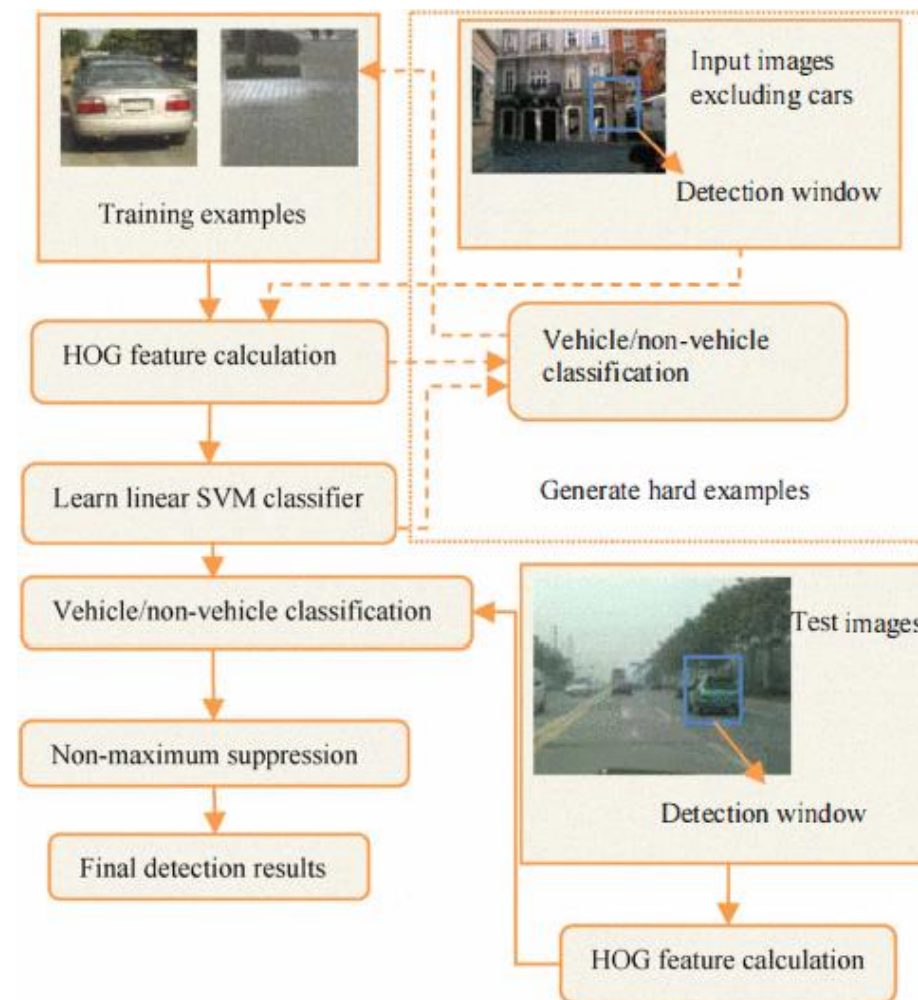
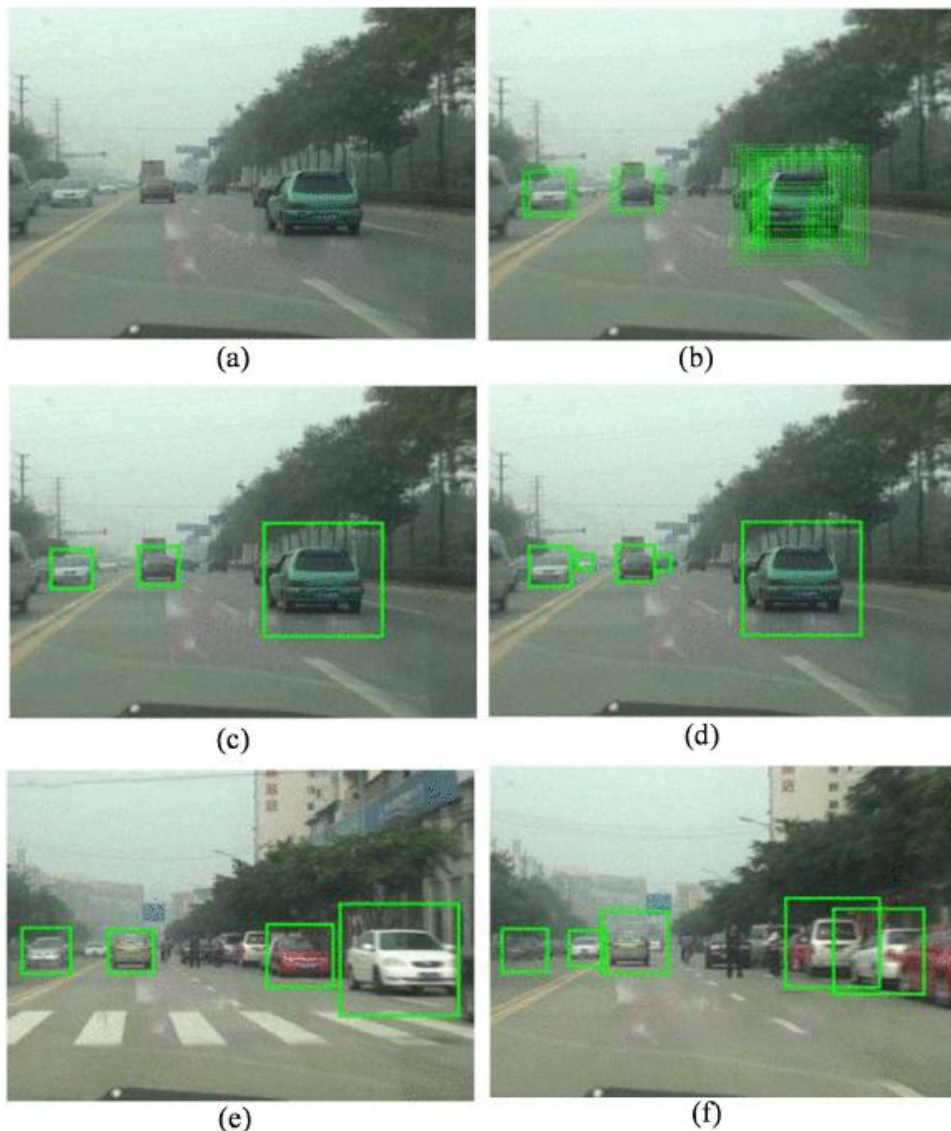
<https://ieeexplore.ieee.org/abstract/document/6728559>

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



- Haar

- HOG

- LBP

- SIFT, SURF

- CNNs



Traditional Approaches



KeyPoints



Deep Learning Approach

Related Works

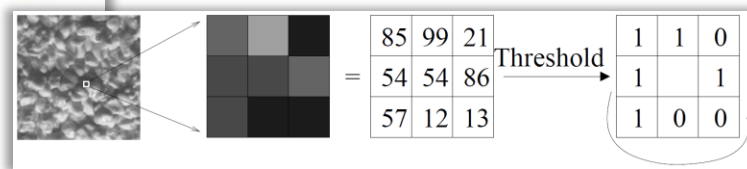
2006

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*

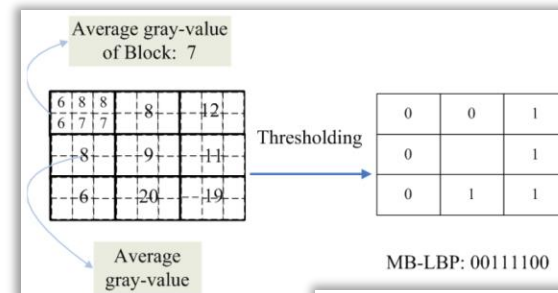


Zhang at al.
(2007)

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)

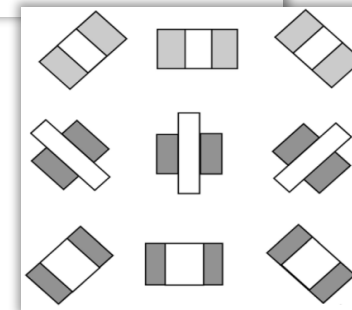
Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers

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



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



MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Object Detection (Analysis)

LBP

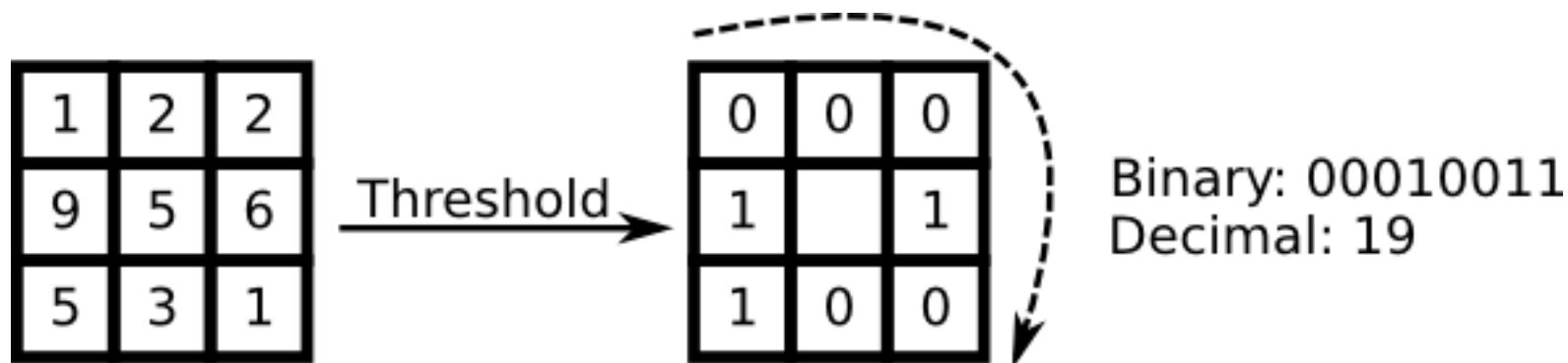


LBP - Local Binary Patterns

- 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 efficiently 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.

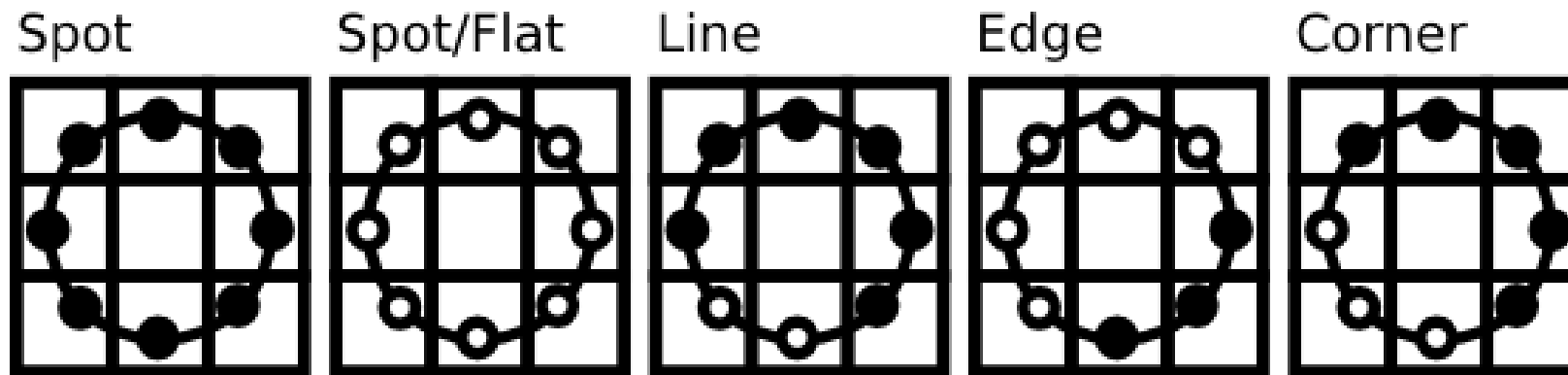
LBP - Local Binary Patterns

- The important properties of LBP are the resistance to the lighting changes and a low computational complexity.
- Due to their properties, LBP were used in many detection tasks, especially in facial image analysis [1, 4].





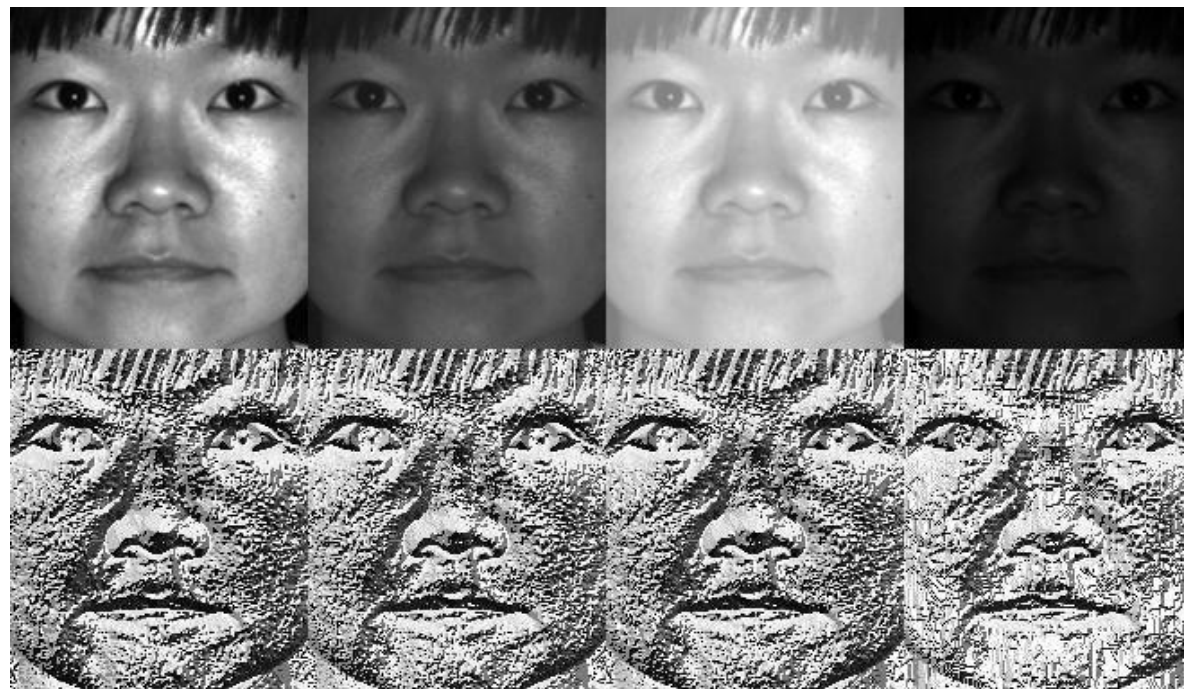
LBP - Local Binary Patterns





LBP - Local Binary Patterns

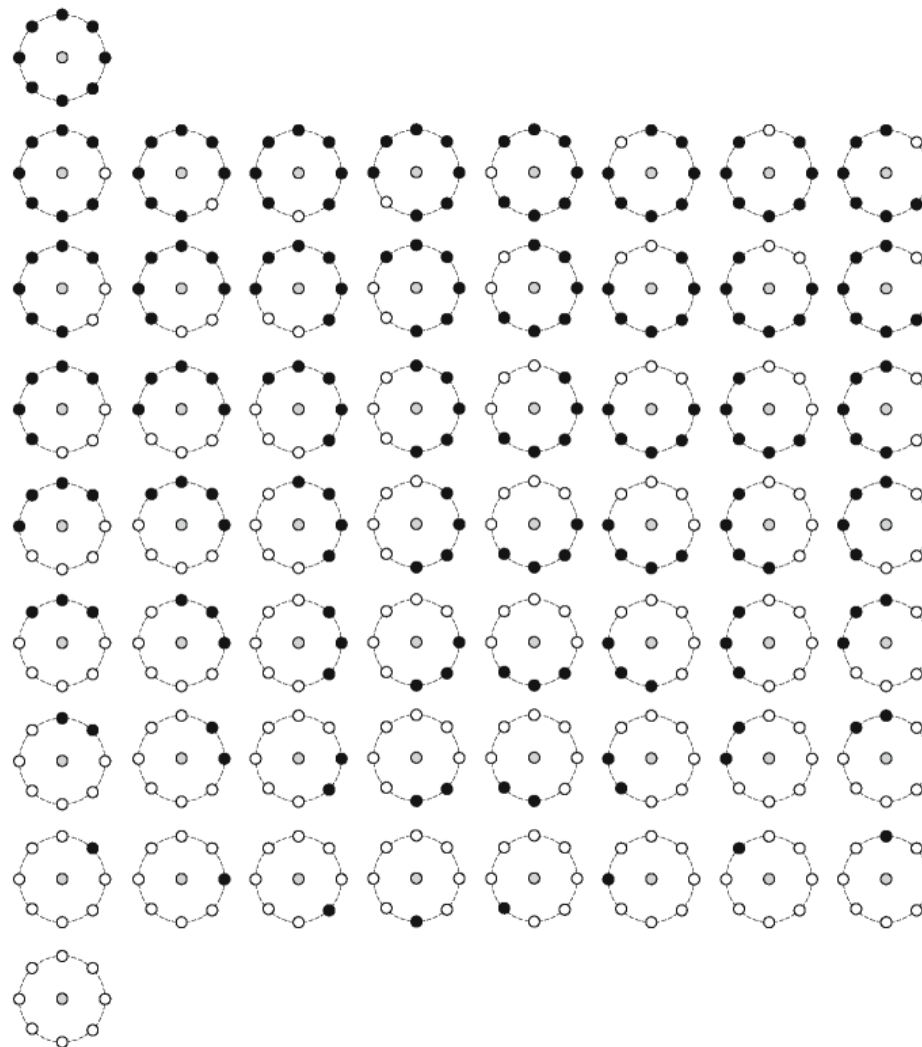
- Robust to monotonic changes in illumination





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





LBP - Local Binary Patterns

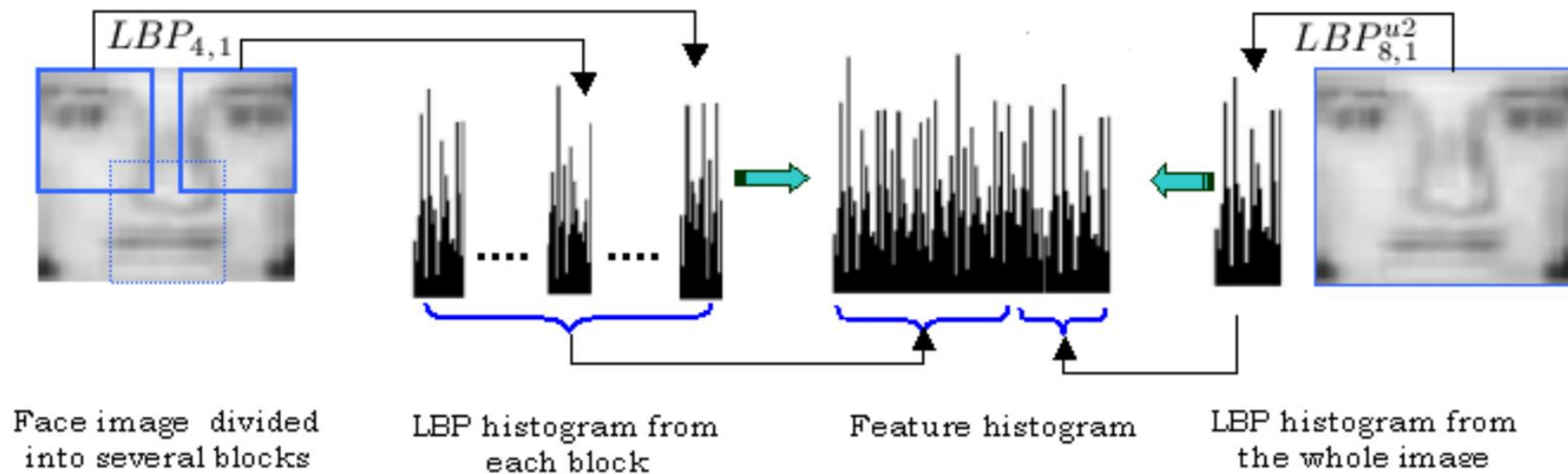
- [1] Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(12), 2037–2041 (2006)
- [2] Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (Jan 1996), [http://dx.doi.org/10.1016/0031-3203\(95\)00067-4](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 classification. 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 classifier with a polynomial kernel is used for the final classification.

LBP - Local Binary Patterns





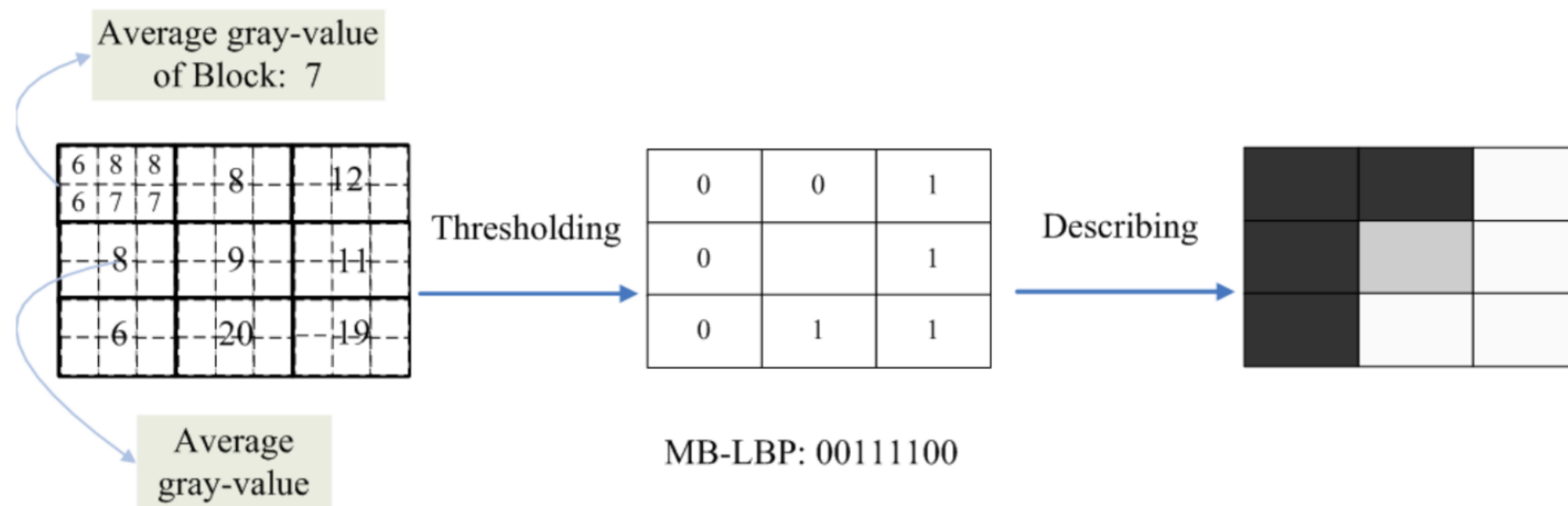
LBP - Local Binary Patterns

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-to-fine 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 difficult lighting conditions. *Image Processing, IEEE Transactions on* 19(6), 1635–1650 (2010)

[2] Feng, X., Hadid, A., Pietikainen, M.: A coarse-to-fine classification 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

- SIFT, SURF

- CNNs



Traditional Approaches



KeyPoints



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

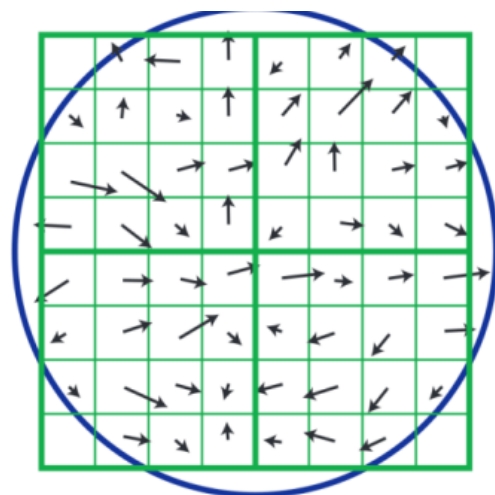
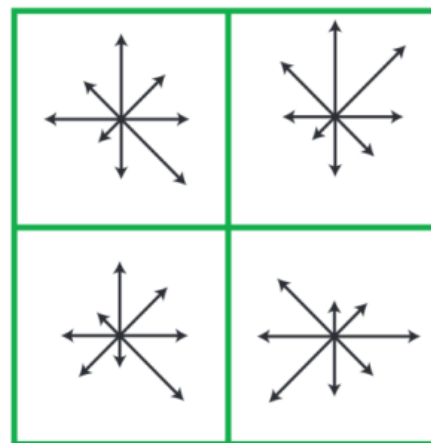


Image gradients



Keypoint descriptor

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 efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2564–2571 (2011)

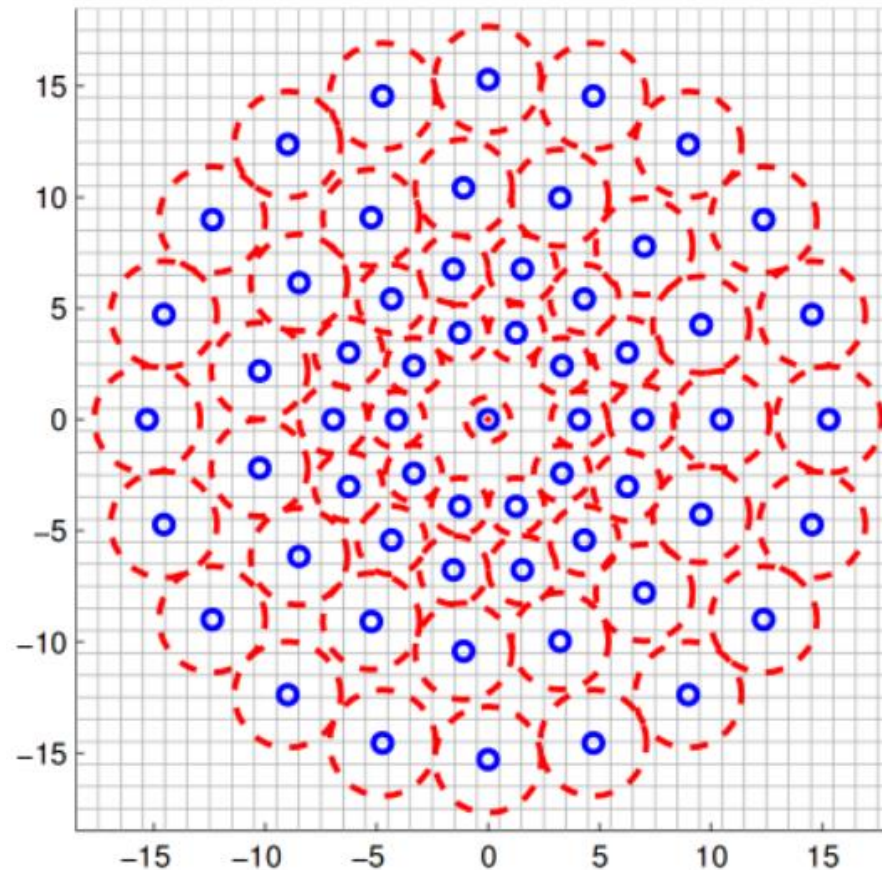


KeyPoints – BRISK

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.



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)

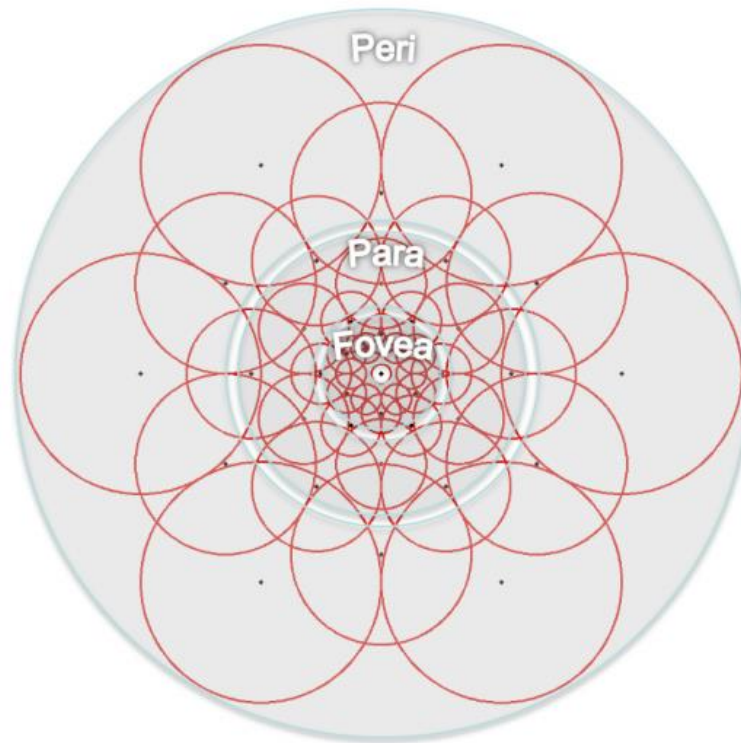


KeyPoints – FREAK

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.



KeyPoints – FREAK



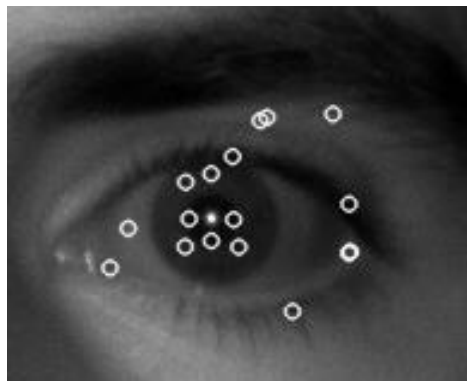
FREAK sampling pattern [1]

[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)

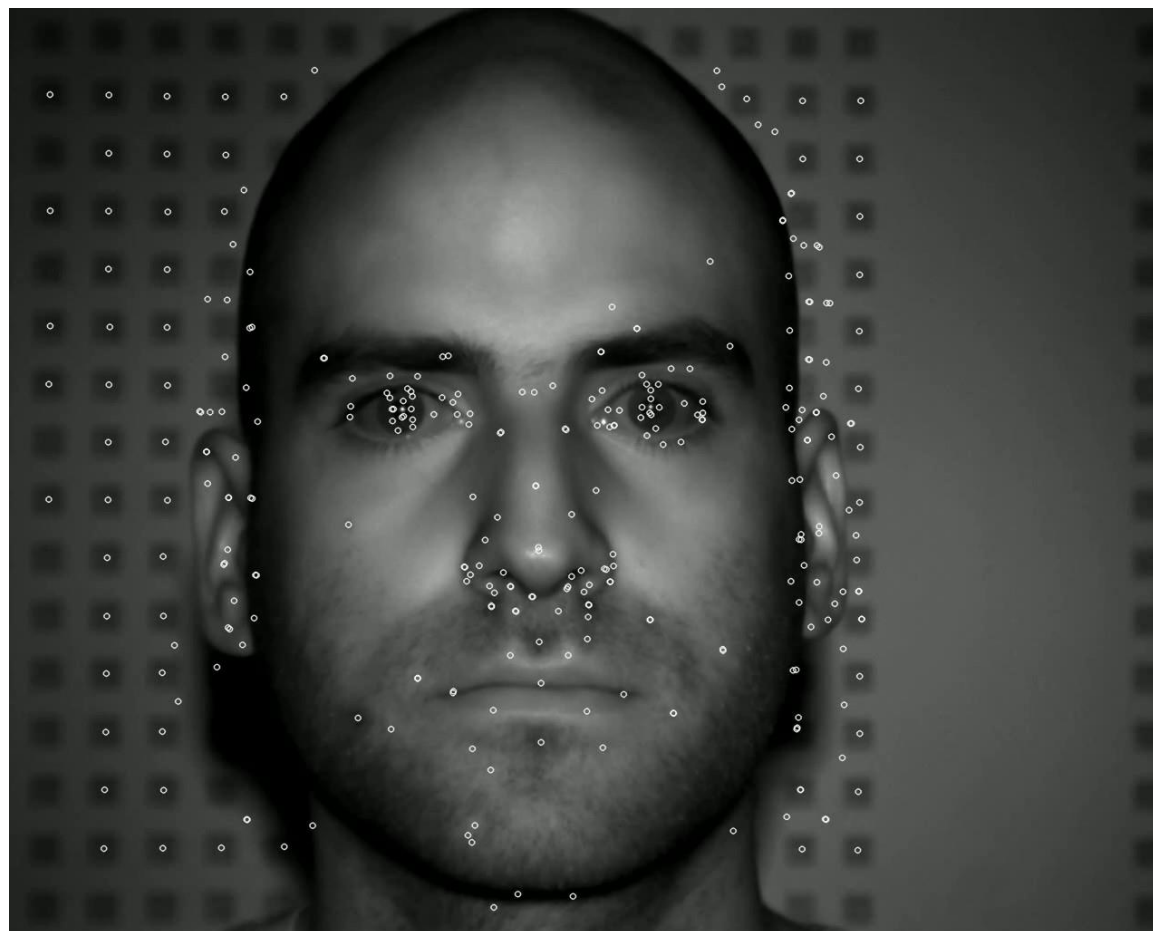


Keypoints - Example

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

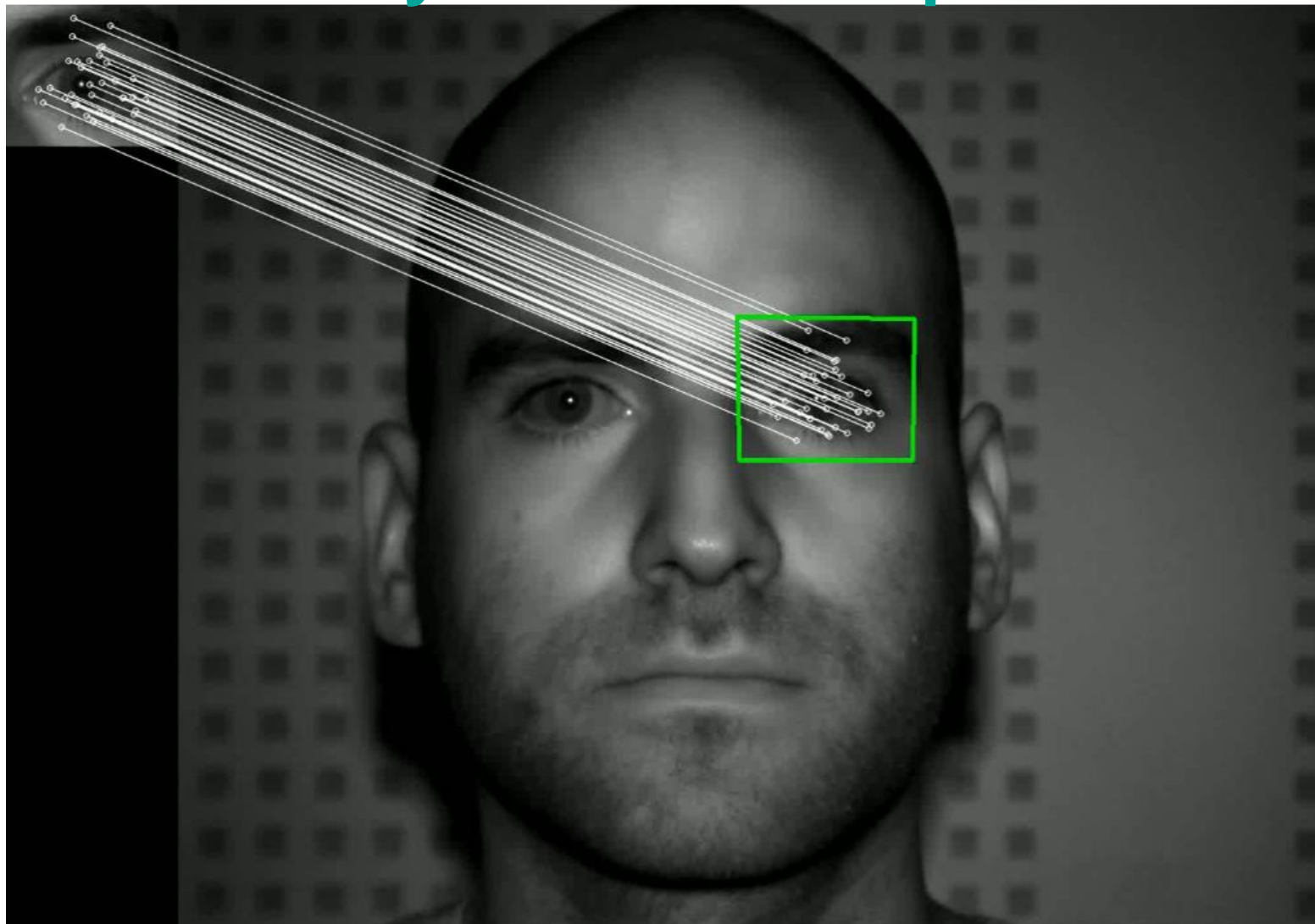


template





Keypoints - Example





- Haar

- HOG

- LBP

- SIFT, SURF

- CNNs



Traditional Approaches



KeyPoints



Deep Learning Approach