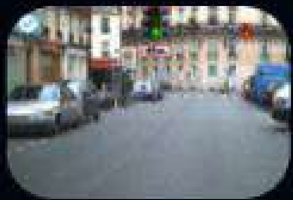

Car 5
Lane line 3
Person 27
Traffic light 74

- Describe Advanced Driving Assistance Systems (ADAS)
- List the most common examples of ADAS
- Review the principles of main visual object detection methods
- List the most commonly used visual features

- Motivations for visual objects detection
- Methods for visual detections
- Visual Features and Objections Recognition and Characterization

Traffic Sign Detection and Recognition (TSR)



Traffic Lights Detection

Cars Visual Detection

Pedestrians Visual Detection

Continue

All videos result from research conducted at Center for Robotics of MINES ParisTech


Advanced Driving Assistance Systems (ADAS)
Examples of ADAS are:
• Lane Departure Warning
• Forward Collision Warning
• Pedestrian Collision Warning
• Blind Spot Monitoring
• Speed Limit Assistant
• Driver Attention Warning
• Night vision
120
MPB

- Adaptive Cruise Control (ACC)
- Lane Keeping (LK)
- Autonomous Emergency Braking
- Automated Parking


Correct answer
Click to continue
Drag each Active ADAS to its corresponding picture.
A
B
3) Autonomous Emergency Breaking
4) Automated Parking
C
D
2) Lane Keeping
1) Adaptive Cruise Controls

Main Goal: Localize and Categorize "Objects"

- Intelligent functions for safer and/or easier driving are called **ADAS**, or Advanced Driving Assistance Systems

- There are several different types of ADAS, such as Forward Collision Warning (FCW), Blind Spot Monitoring (BSM), Lane Keeping, Adaptive Cruise Control (ACC), and Automated Parking to name a few

- ADAS require real-time, on-board analysis of video footage from a camera, to interpret the visual scene correctly

Objects visual **DETECTION**

For **objects**, visual scene analysis often performed in TWO (or three) STEPS:

Detection

Recognition

Temporal Tracking

Detection = find WHERE in the image are (maybe) located interesting objects

Candidate locations for searched objects

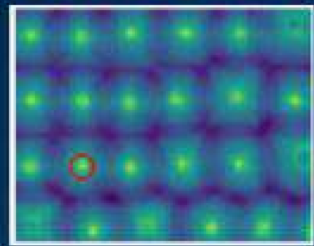
Recognized objects

Template Matching: Compares a reference image or template of an object with sub-images corresponding to all possible positions and sizes.

Template

Image

“Match_template” result

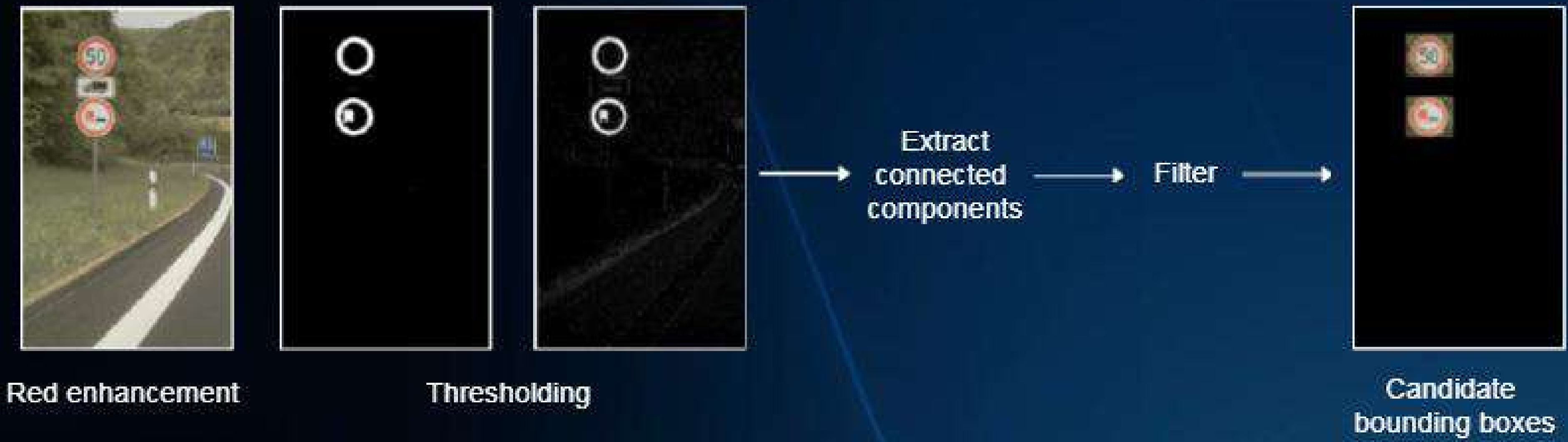


For each position, a similarity measure is computed.

$$SAD(x,y) = \sum_{i=0}^{T_{\text{rows}}} \sum_{j=0}^{T_{\text{cols}}} \text{Diff}(x+i, y+j, i, j)$$

Drawbacks:

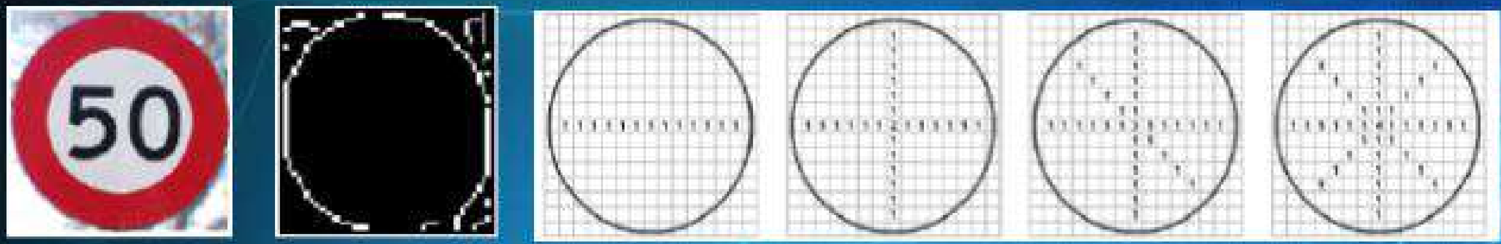
- High computation time
- Handling of variations in luminosity, contrast, orientation
- Handling of deformations



Potential Problems:

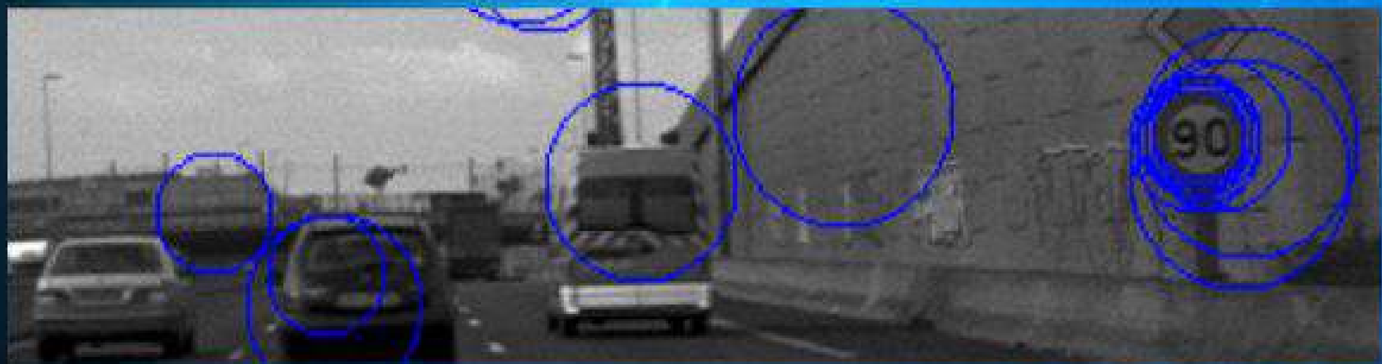
- “Parasite” detection (similar color on totally different object)
- High variability of color appearance (especially in RGB!)

General case: template-matching on contours image



Problems:

- Rather computer-intensive
- Some shape are not so rare (rectangles!!)

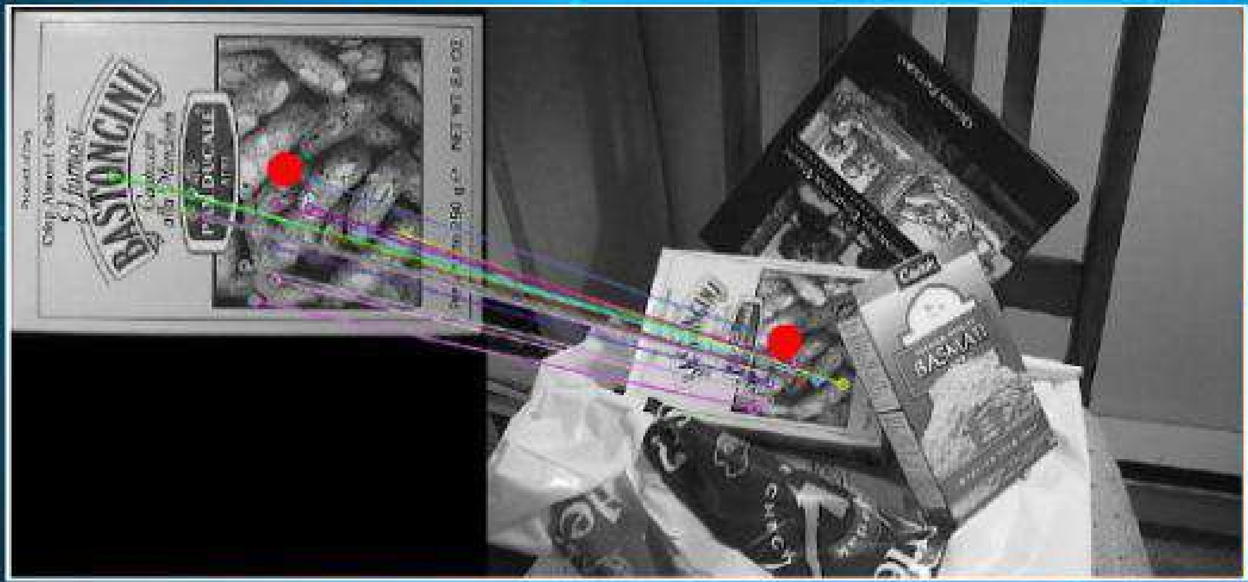


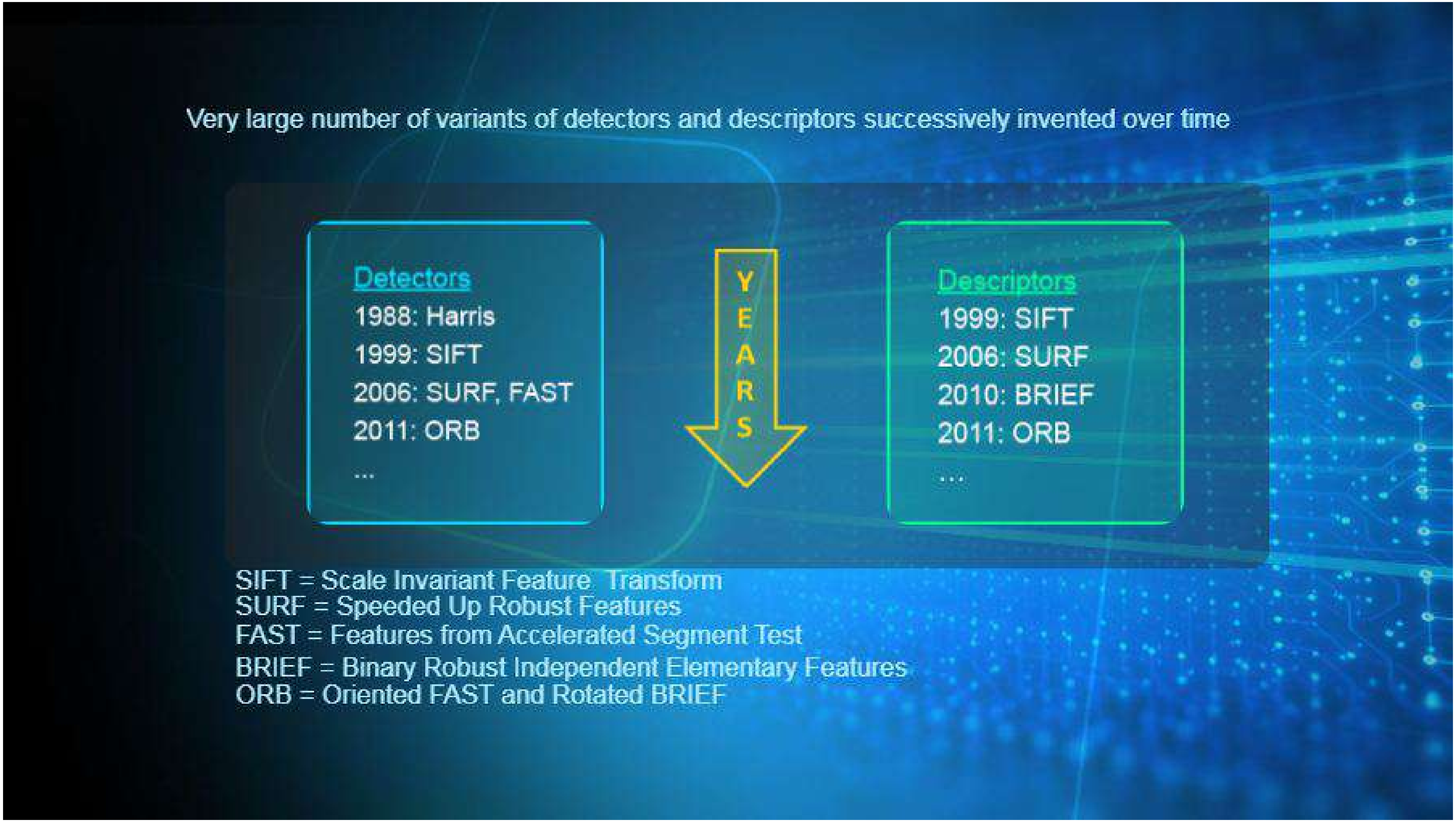
Scan the image with a sliding window:

- Build a pyramid of down-sampled images
- Scan each level of pyramid with a sliding fixed-size detection window
- Apply a single common classifier on all sub-images to determine if it is a bounding-box

Detector should ideally be repeatable, i.e. select same points whatever the scale, rotation, lighting

Descriptor should ideally be invariant under change of scale/rotation/lighting




Very large number of variants of detectors and descriptors successively invented over time
Detectors
1988: Harris
1999: SIFT
2006: SURF, FAST
2011: ORB
...
YEARS
Descriptors
1999: SIFT
2006: SURF
2010: BRIEF
2011: ORB
...
SIFT = Scale Invariant Feature Transform
SURF = Speeded Up Robust Features
FAST = Features from Accelerated Segment Test
BRIEF = Binary Robust Independent Elementary Features
ORB = Oriented FAST and Rotated BRIEF

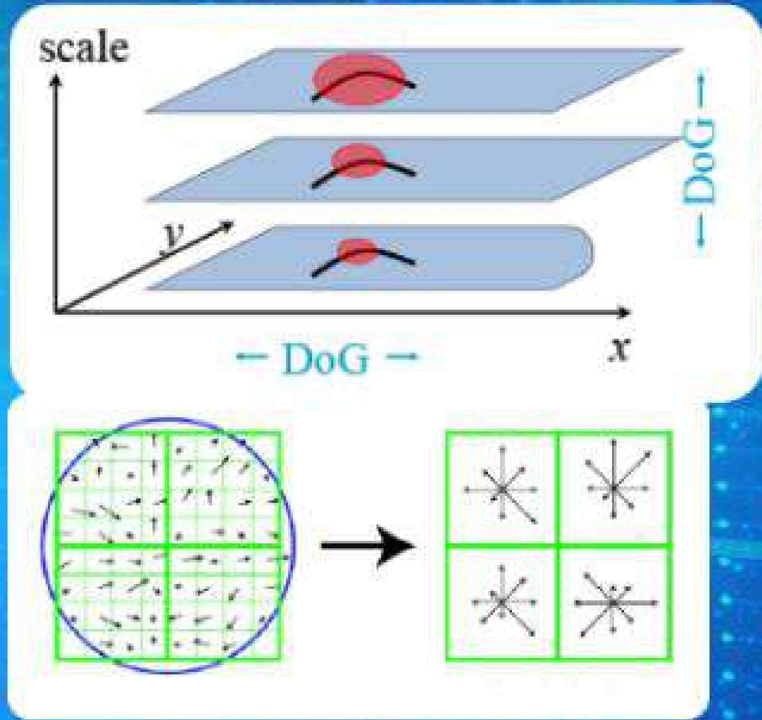
Scale Invariant Feature Transform [proposed by Lowe in 1999]

Detector

Max and mins of Difference of Gaussians (DoG) applied in scale space to a series of smoothed and resampled images

Descriptor

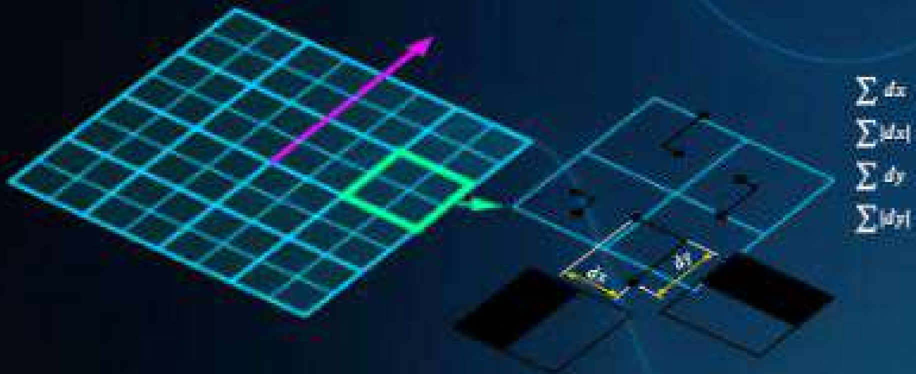
Summarizes spatial distribution of gradient orientations around keypoint in a 128D vector



Speeded Up Robust Features proposed by Bay et al. in 2006

Detector: approximation with Haar filters of blob detection by determinant of Hessian (speed-up with integral image)

Descriptor: based on Haar filters responses around keypoint



The general process for objects detection with keypoints consists of the following successive steps:

- Precompute once for all keypoints' locations and descriptors on object to find
- Compute keypoints' locations and descriptors on « query » (image where we search object)
- Find keypoints in query with descriptors similar to a keypoint in object
- Filter false matches by geometric checking using Random Sample Consensus (RANSAC)

Advantage: intrinsically multi-scale search, thanks to scale invariance of keypoint detector and descriptor

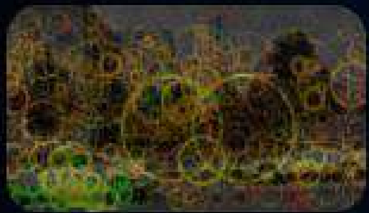
Problem: can search/find only a specific image pattern

Correct answer
Click to continue

Drag each term related to Keypoints to the correct space in the sentences provided to make the sentences correct.

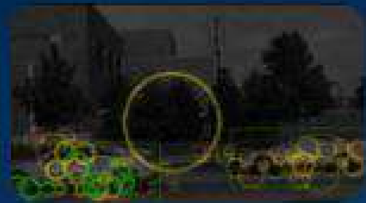
- Pre-compute once for all keypoints' **Locations** and descriptors on object to find.
- **Compute** keypoints' locations and descriptors on « query », an image where we search object
- Find keypoints in **Query** with descriptors similar to that of a keypoint in object
- Filter false matches by **Geometric** checking using Random Sample Consensus, or RANSAC.

If looking for objects of a CATEGORY (rather than a particular pattern/sub-image), we first need to build a filter for discriminating keypoints that are specific of the type of searched objects, as illustrated in the upper line of images:



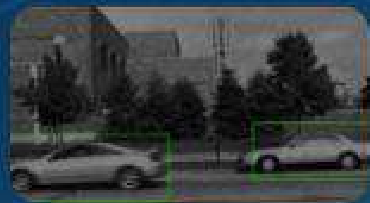
SURF keypoints

Filtering of CARS keypoints by classifier

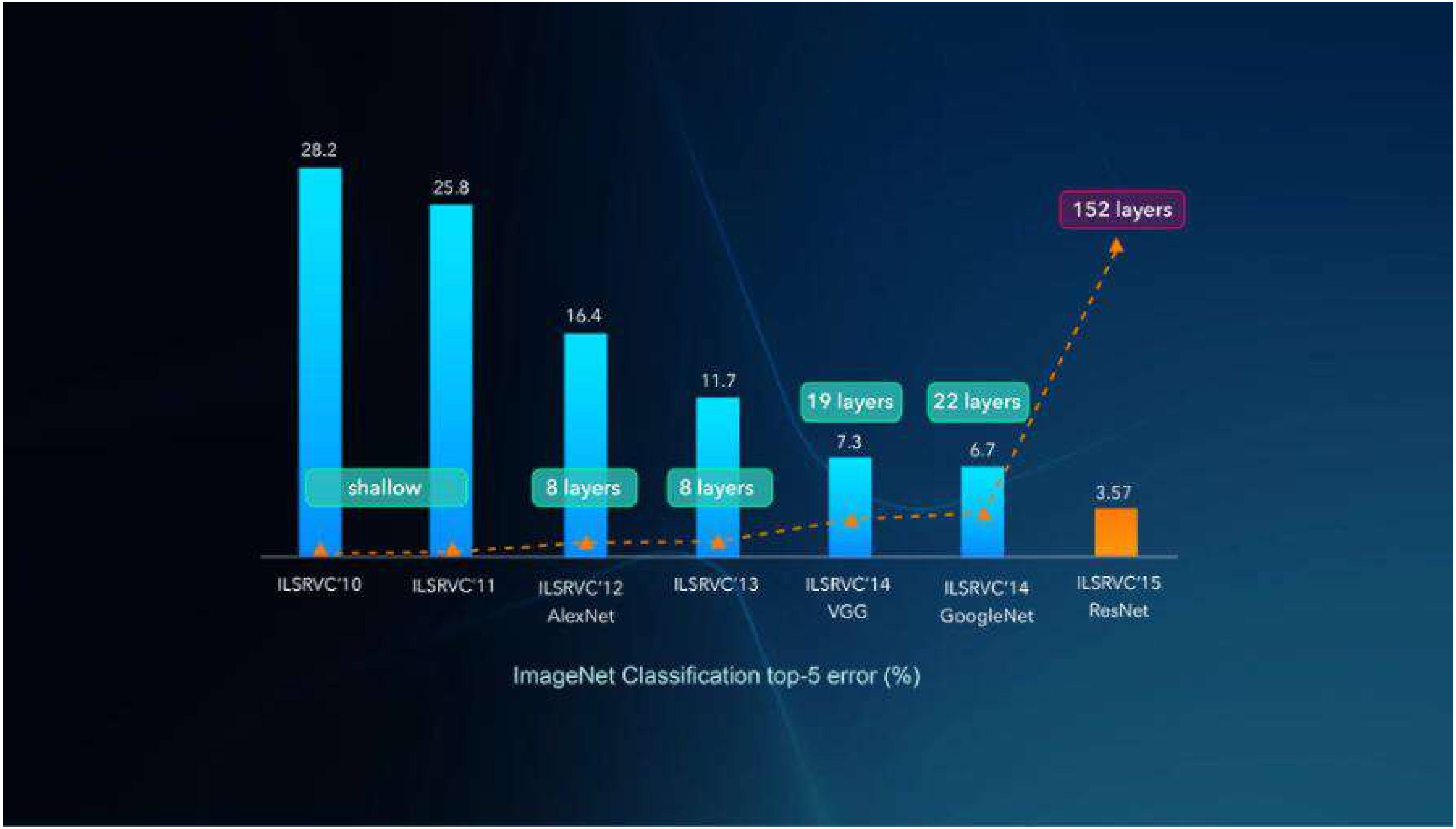


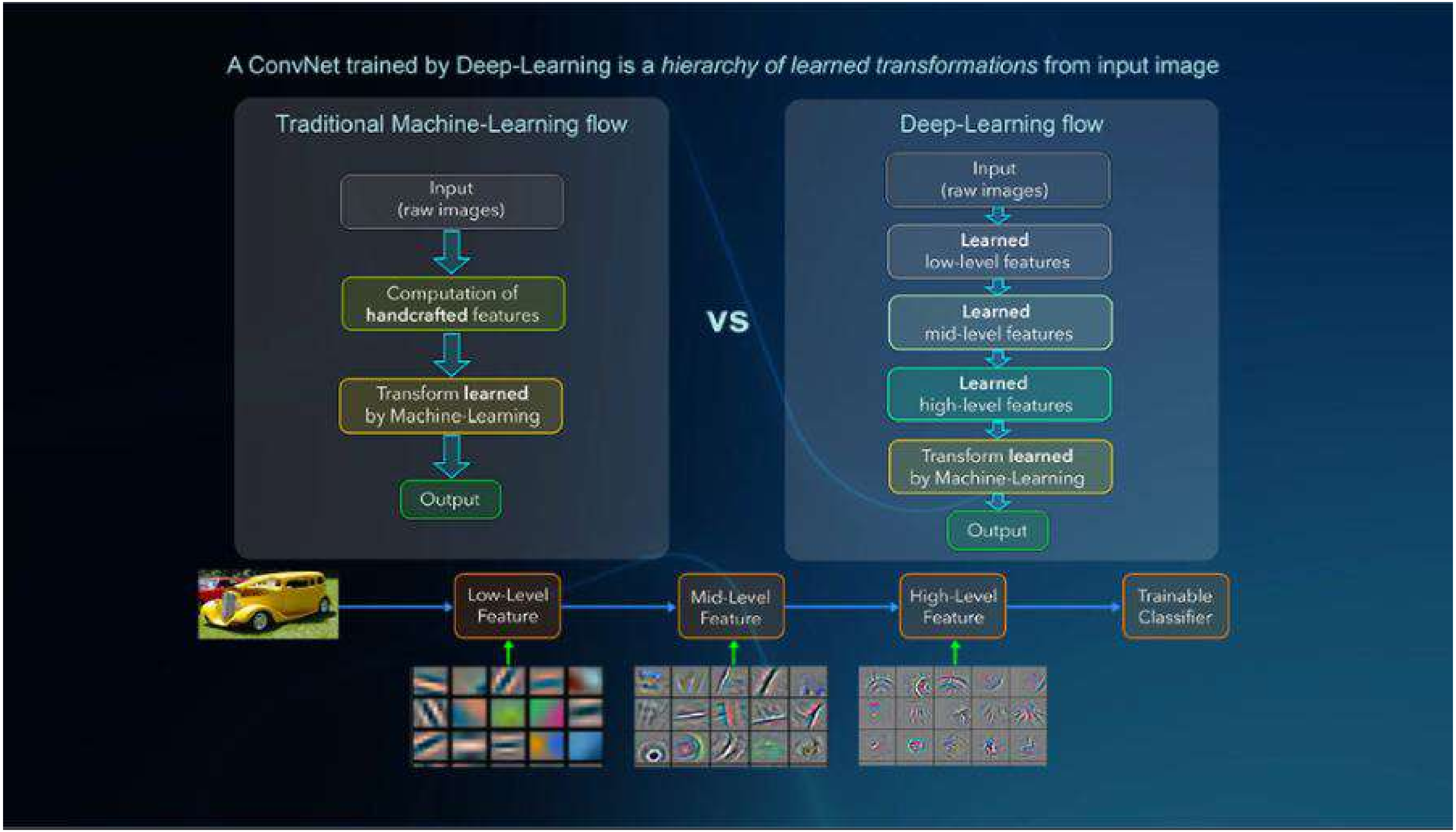
Objects localization

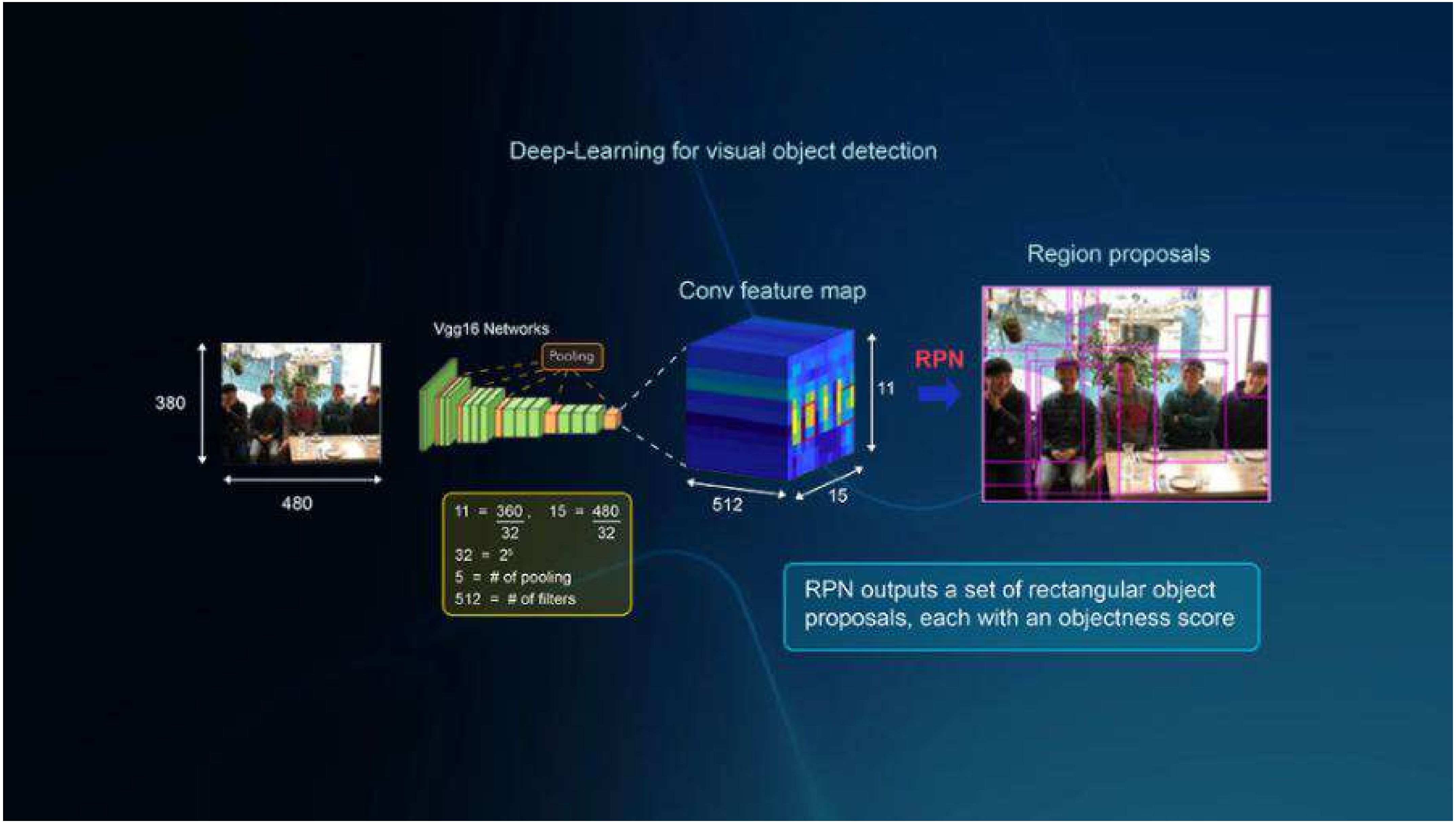
Done by first training a classifier on a large set of keypoint descriptors labelled with "car" for those found within a car bounding-box, and "non-car" for others



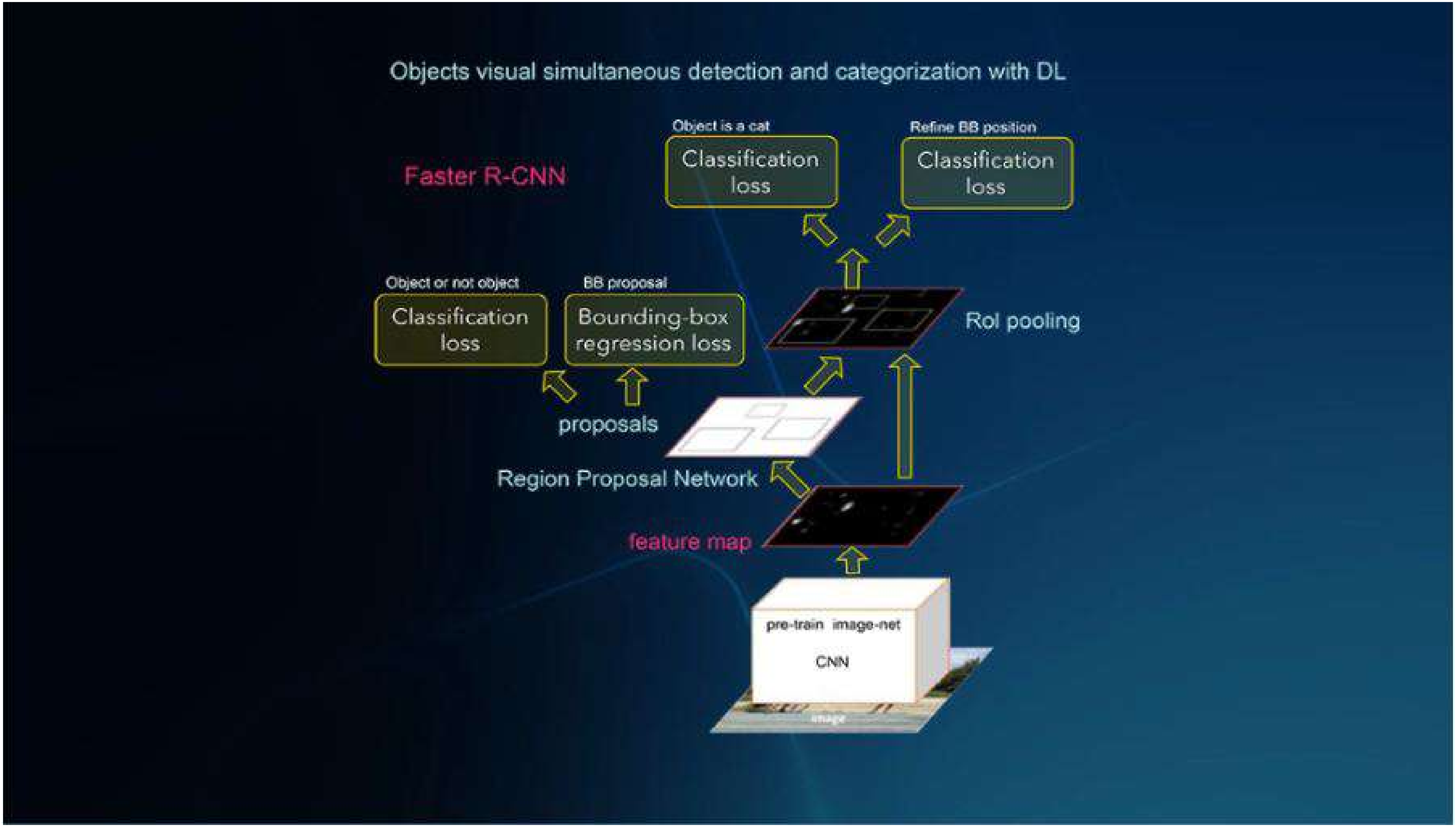
Result of research conducted by center for Robotics of MINES ParisTech


28.2
25.8
16.4
11.7
7.3
6.7
3.57
152 layers
19 layers
22 layers
shallow
8 layers
8 layers
ILSRVC'10
ILSRVC'11
ILSRVC'12
AlexNet
ILSRVC'13
ILSRVC'14
VGG
ILSRVC'14
GoogleNet
ILSRVC'15
ResNet
ImageNet Classification top-5 error (%)

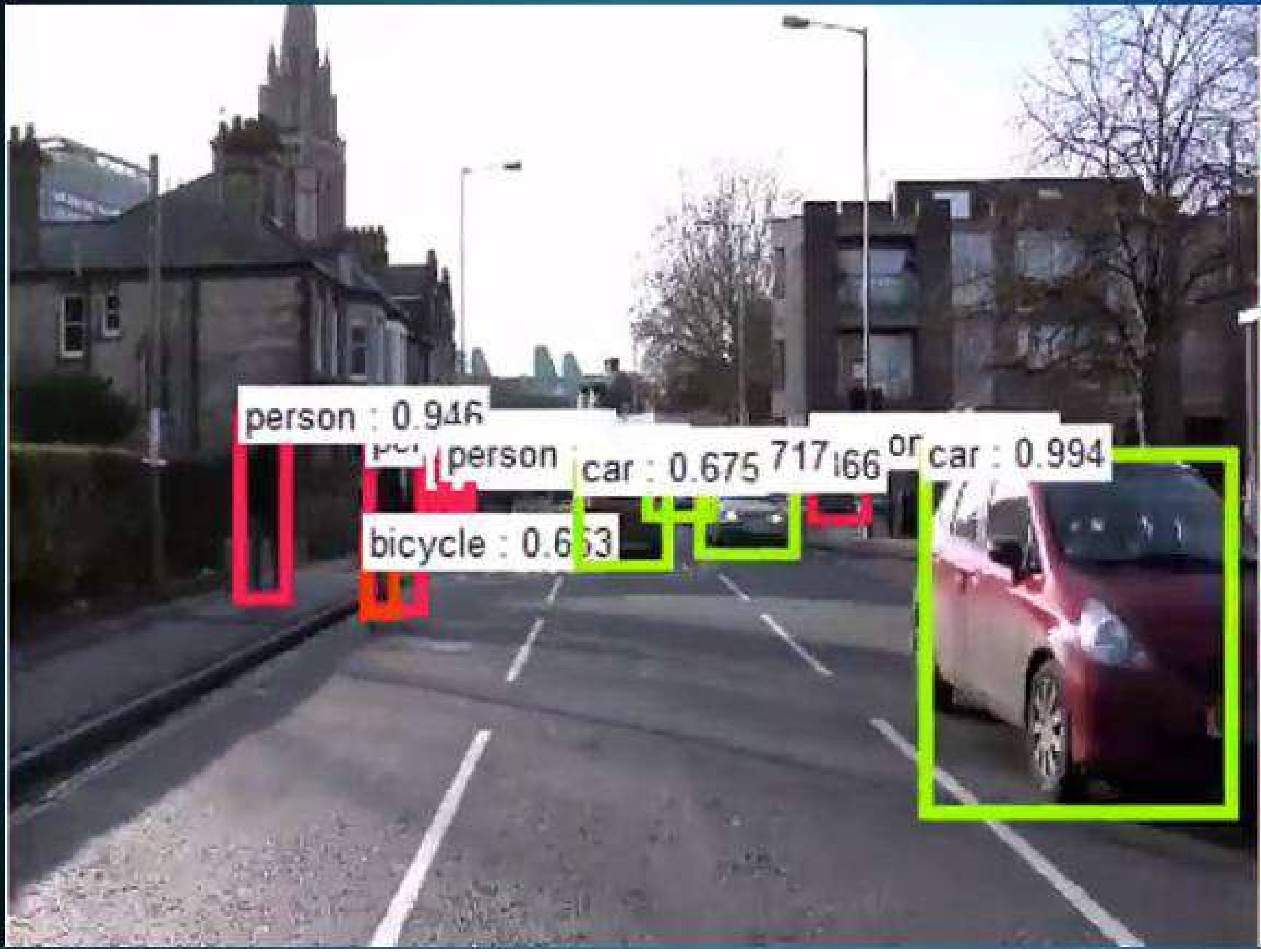

A ConvNet trained by Deep-Learning is a *hierarchy of learned transformations* from input image
Traditional Machine-Learning flow
Input (raw images)
Computation of **handcrafted** features
Transform **learned** by Machine-Learning
Output
VS
Deep-Learning flow
Input (raw images)
Learned low-level features
Learned mid-level features
Learned high-level features
Transform **learned** by Machine-Learning
Output
Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier



Deep-Learning for visual object detection
Region proposals
Conv feature map
Vgg16 Networks
Pooling
380
480
RPN
11
512
15
11 = 360/32, 15 = 480/32
32 = 2⁵
5 = # of pooling
512 = # of filters
RPN outputs a set of rectangular object proposals, each with an objectness score

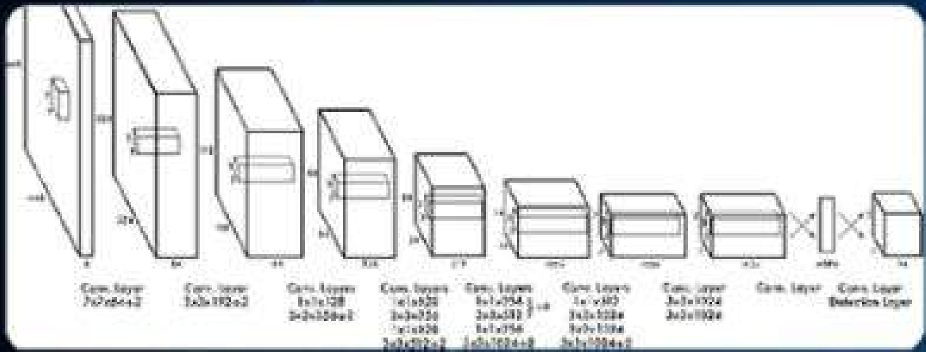

Objects visual simultaneous detection and categorization with DL
Faster R-CNN
Object is a cat
Classification loss
Refine BB position
Classification loss
Object or not object
Classification loss
BB proposal
Bounding-box regression loss
RoI pooling
proposals
Region Proposal Network
feature map
pre-train image-net
CNN

Example video of objects visual simultaneous detection and categorization with R-CNN

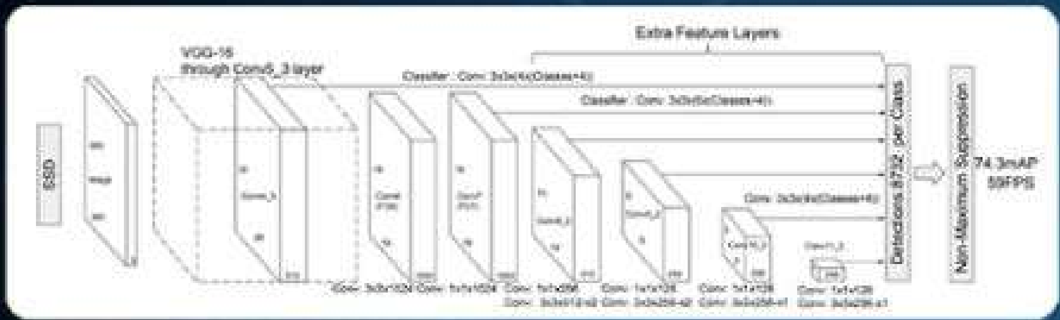


Real-time visual objects detection with DL: latest models

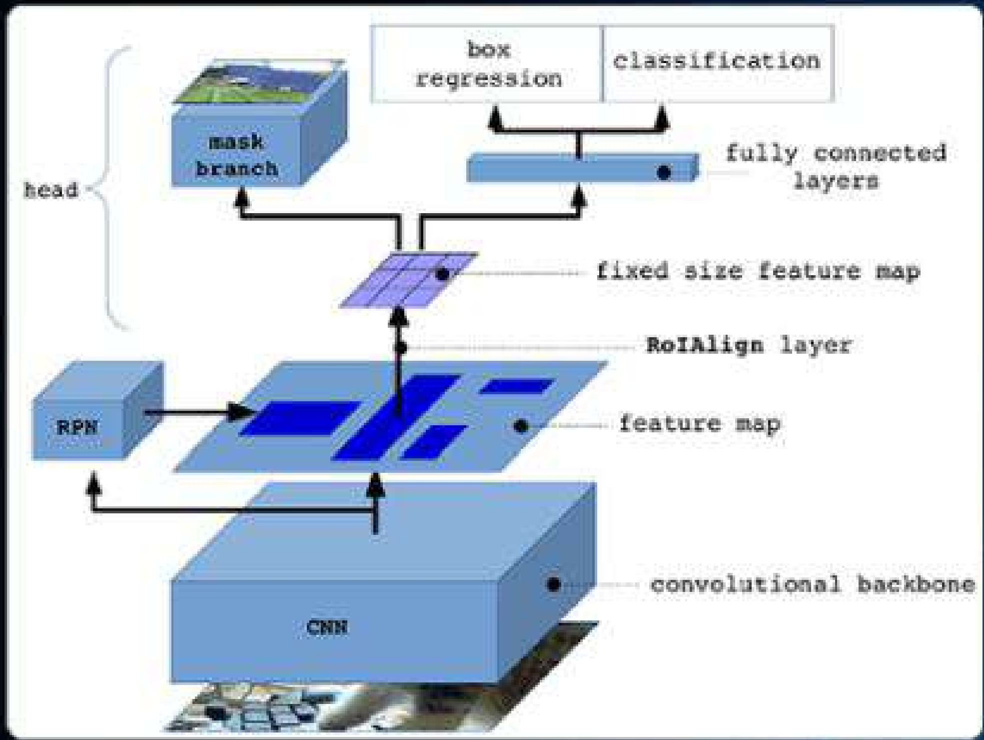
Yolo architecture

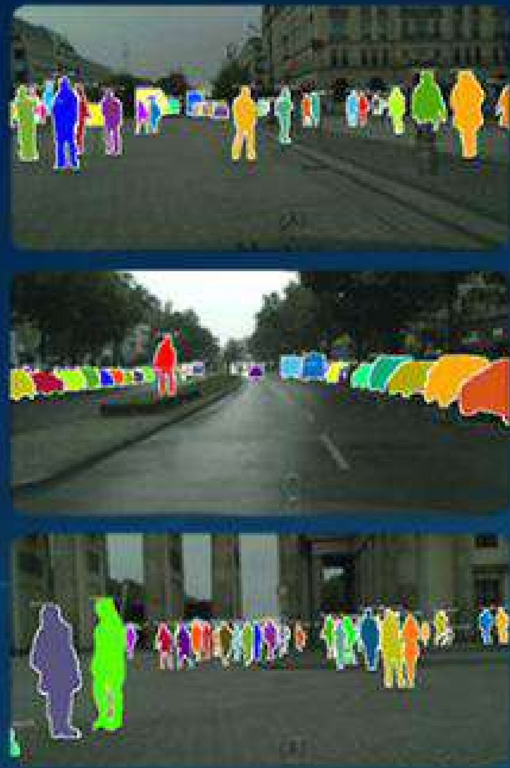


SSD architecture



Mask R-CNN

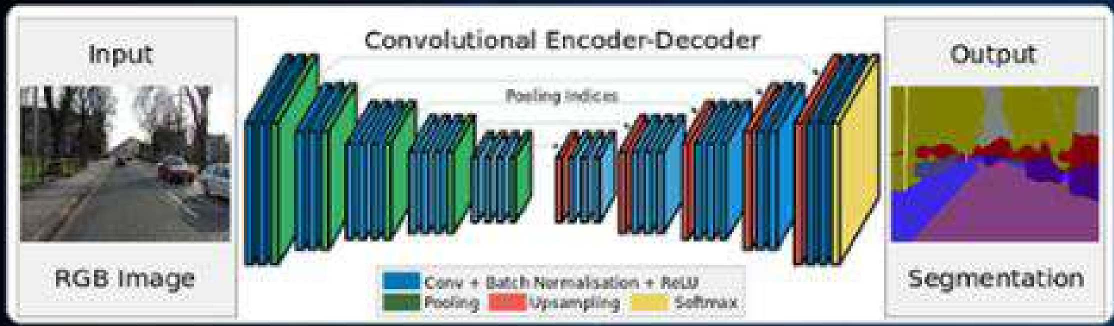




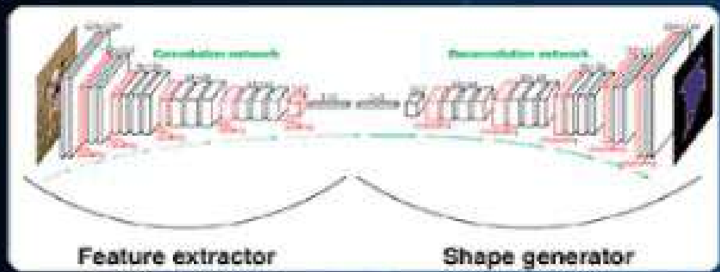
Mask R-CNN architecture extract detailed contours and shape of objects instead of just bounding-boxes

- Image Segmentation: Identify groups of contiguous pixels that "go together"

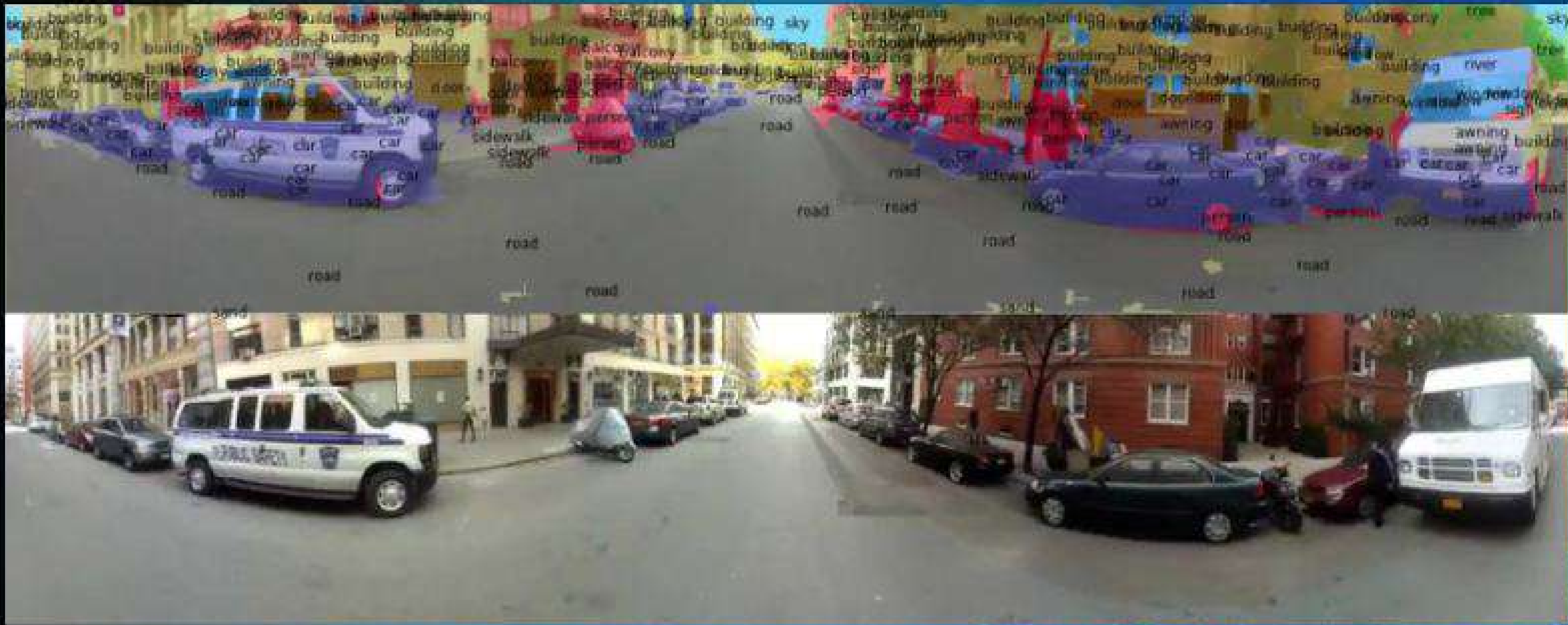
- Semantic Segmentation: Identify groups of contiguous pixels that belong to the same physical object



Convolutional Encoder-Decoder approach



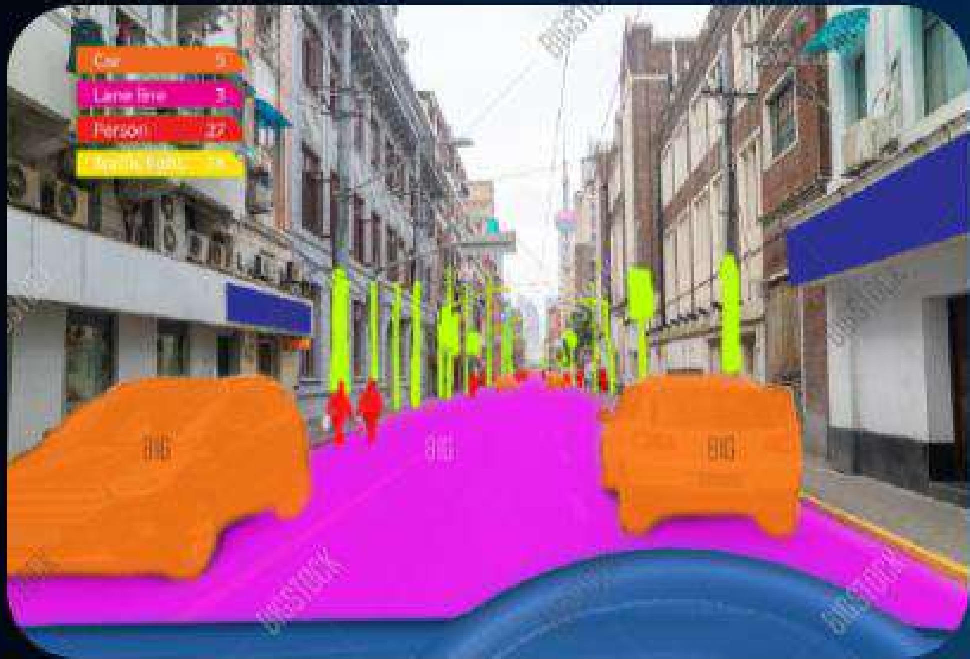
Several other presently competing Deep-Learning models/algorithms for semantic segmentation: SegNet (2015), U-Net (2015), RefineNet (2016), Adversarial Network (2016)



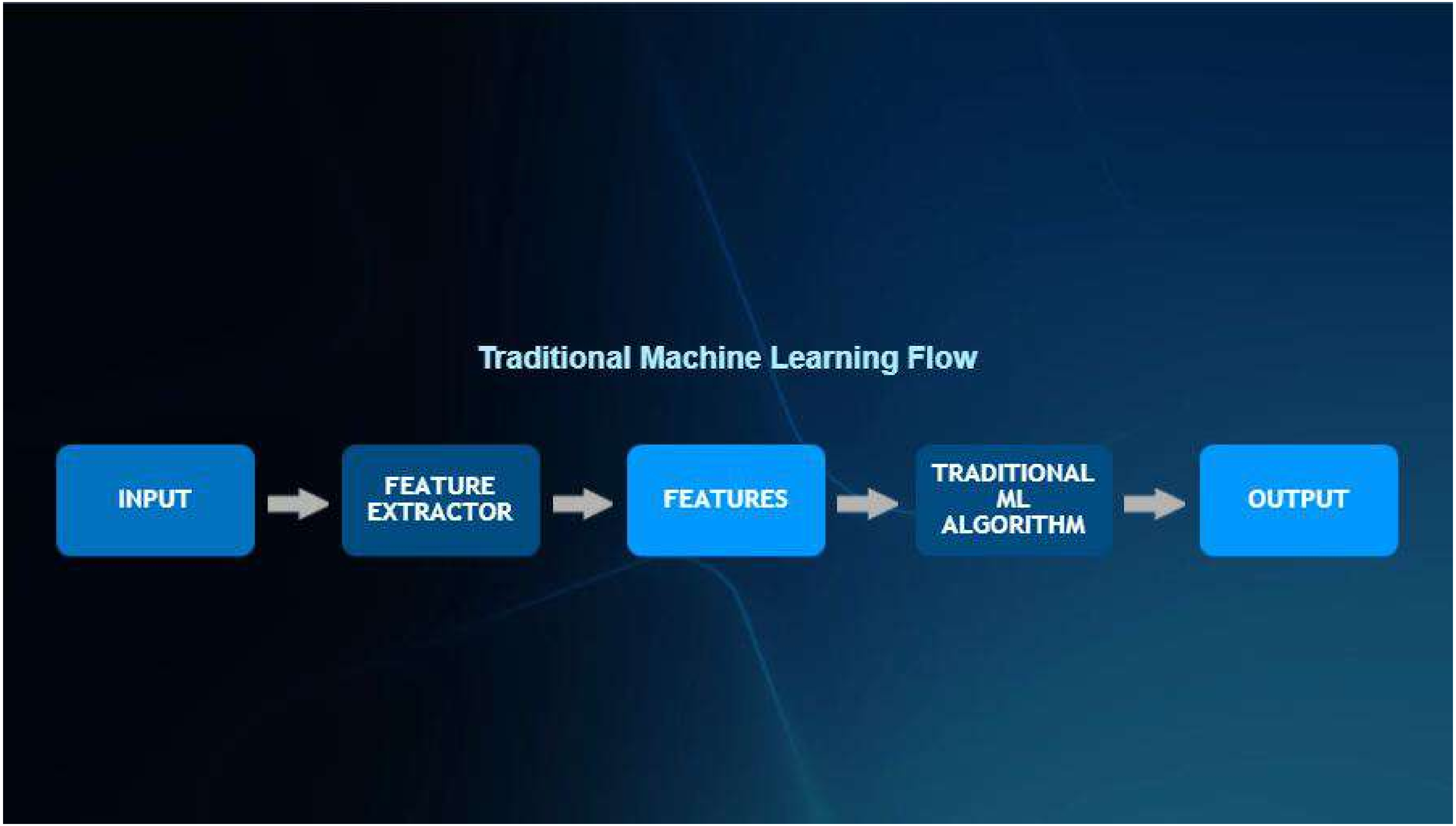
C. Farabet, C. Couprie, L. Najman & Yann LeCun: Learning Hierarchical Features for Scene Labeling, IEEE Trans. PAMI, Aug. 2013

- DETECTION consists in finding WHERE in the image interesting objects are located

- Visual objects detection can be done using various types of approaches:
  - Template matching
  - Shape cues
  - Color cues
  - Window scanning with classifier
  - Keypoints matching
  - Region proposal applied to highest level features of Deep Convolutional Network

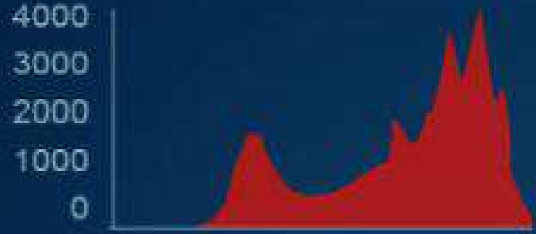


- Robust visual recognition requires independence with regards to:
  - Image size
  - Centering small offsets
  - Rotations (at least small ones)
  - Luminosity & contrast


Traditional Machine Learning Flow
INPUT
FEATURE EXTRACTOR
FEATURES
TRADITIONAL ML ALGORITHM
OUTPUT

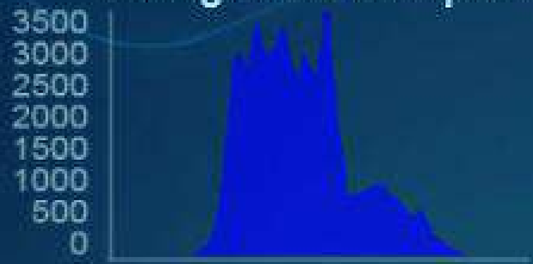
- Color or luminance histograms:
  - Need to be carefully normalized to remain invariant under luminosity and contrast variation
  - Often not sufficiently discriminative to recognize objects

Histogram of red plane



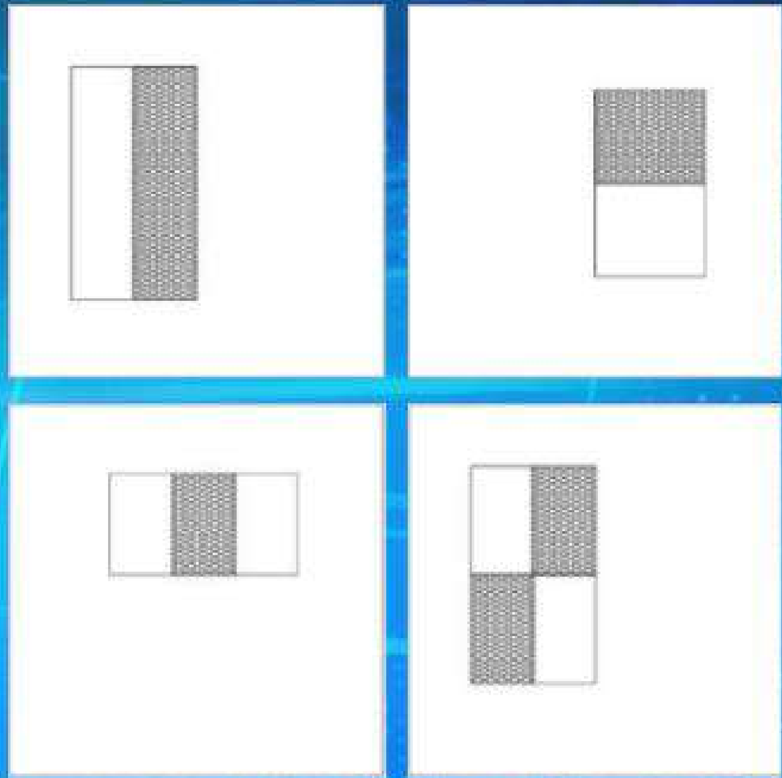
Histogram of green plane

Histogram of blue plane



4 rectangular feature types:

- two-rectangles feature types (horizontal/vertical)
- three-rectangles feature type
- four-rectangles feature type

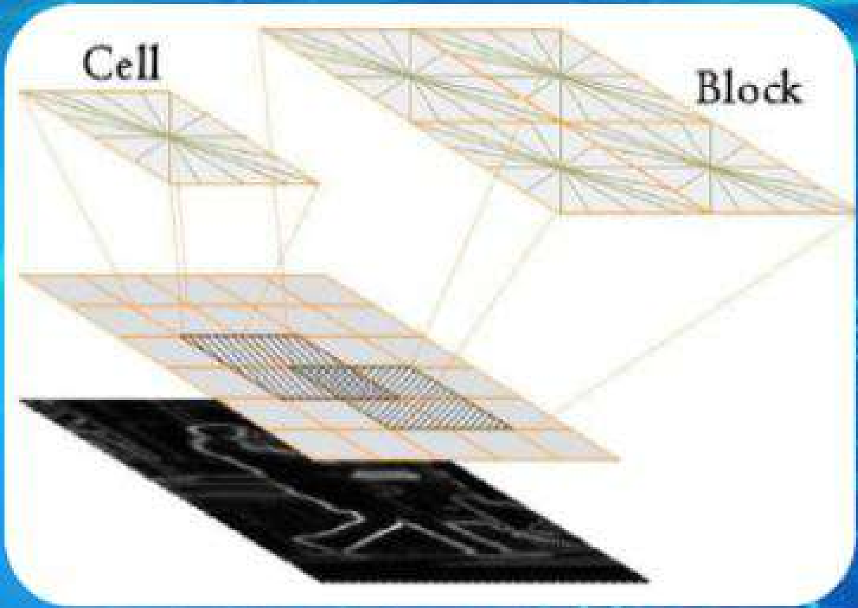


Feature output:

- Σ (pixels in grey rectangles) - Σ (pixels in white rectangles)

Histogram of Orientations of Gradient:

- Computation of vertical and horizontal gradients with 1D derivative mask [-1 0 1] and $[-1\ 0\ 1]^T$
- Accumulation (weighted by gradient magnitude) of gradient orientations in cell bins
- Normalization within overlapping blocks



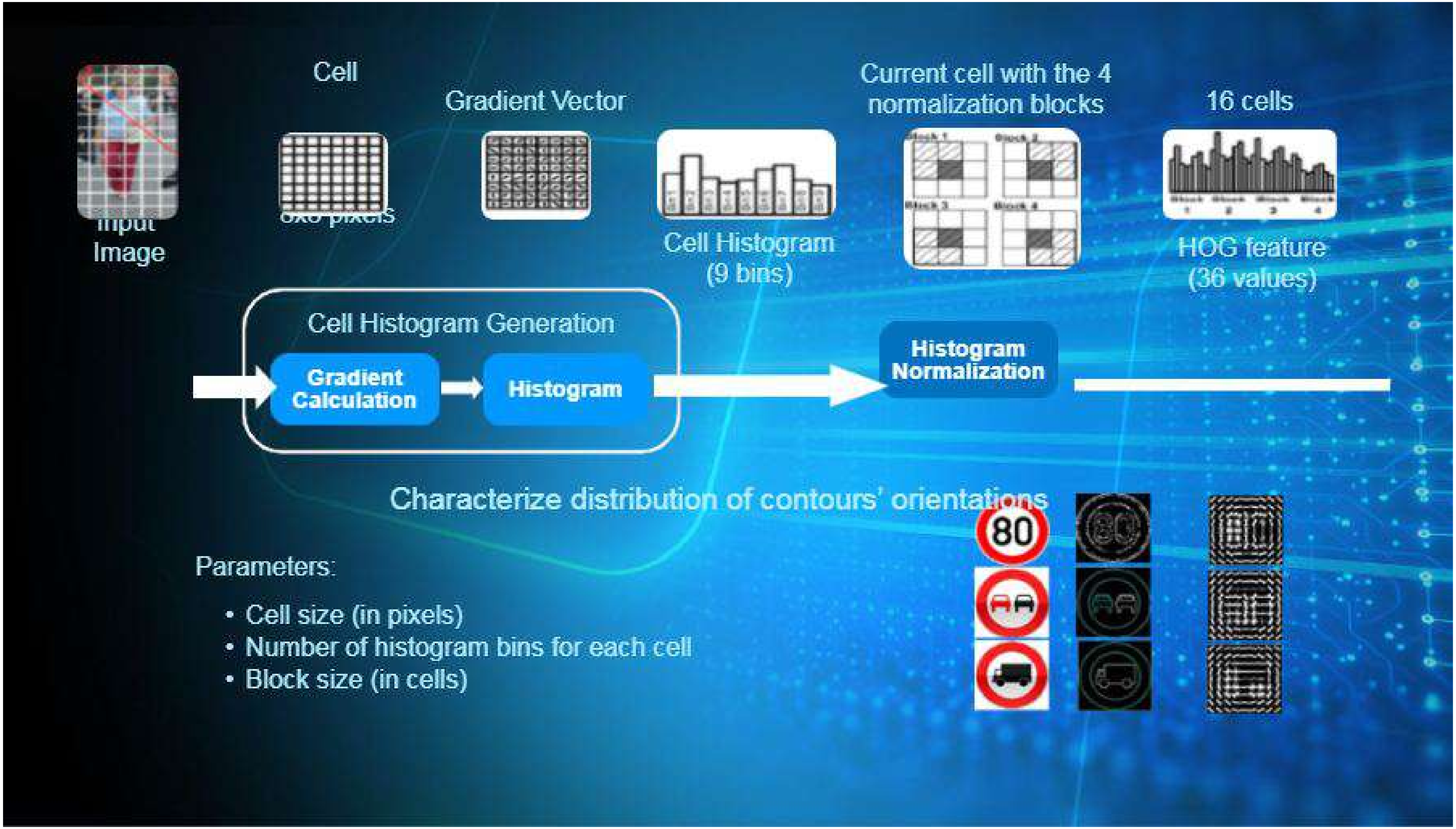
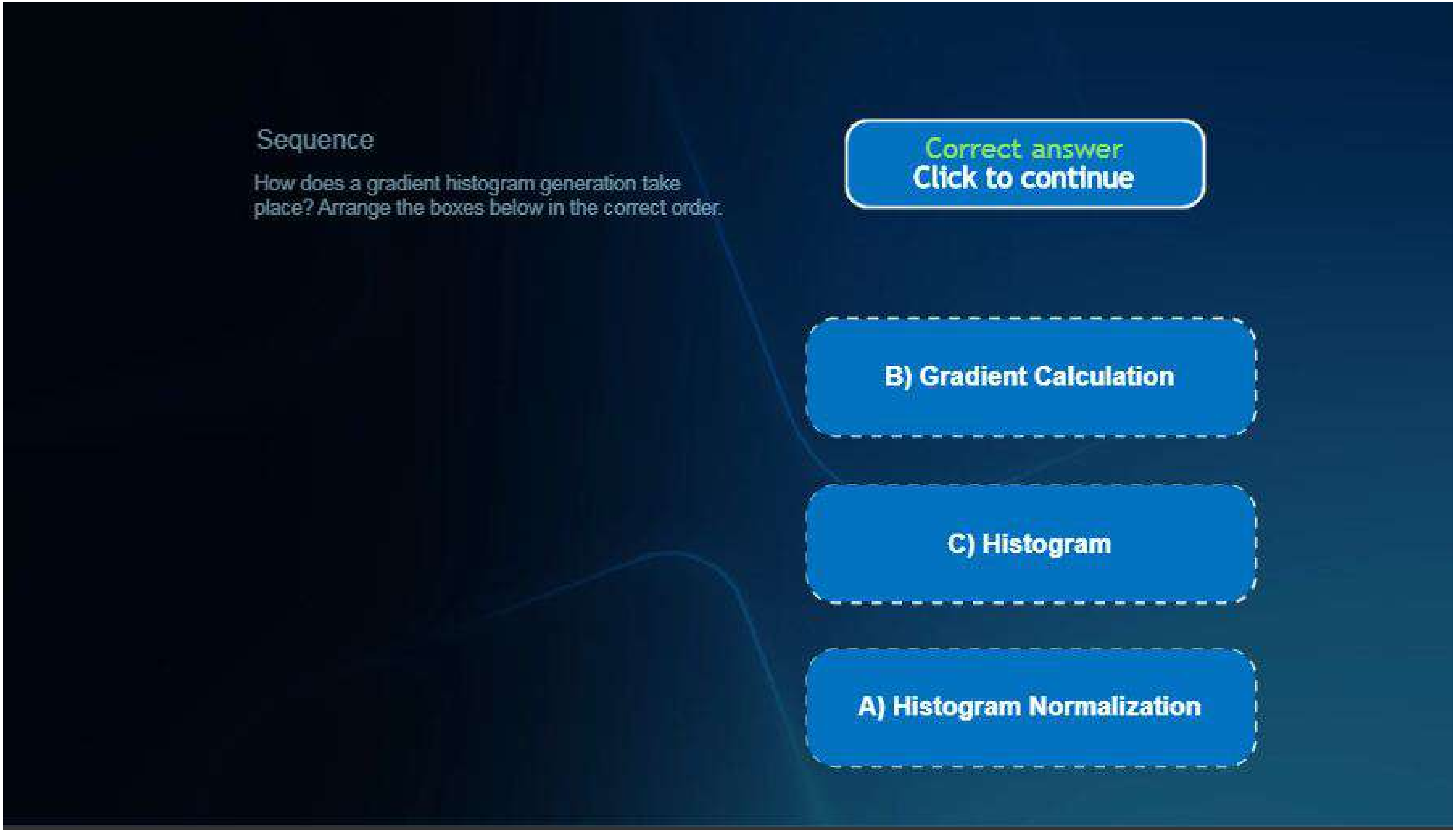
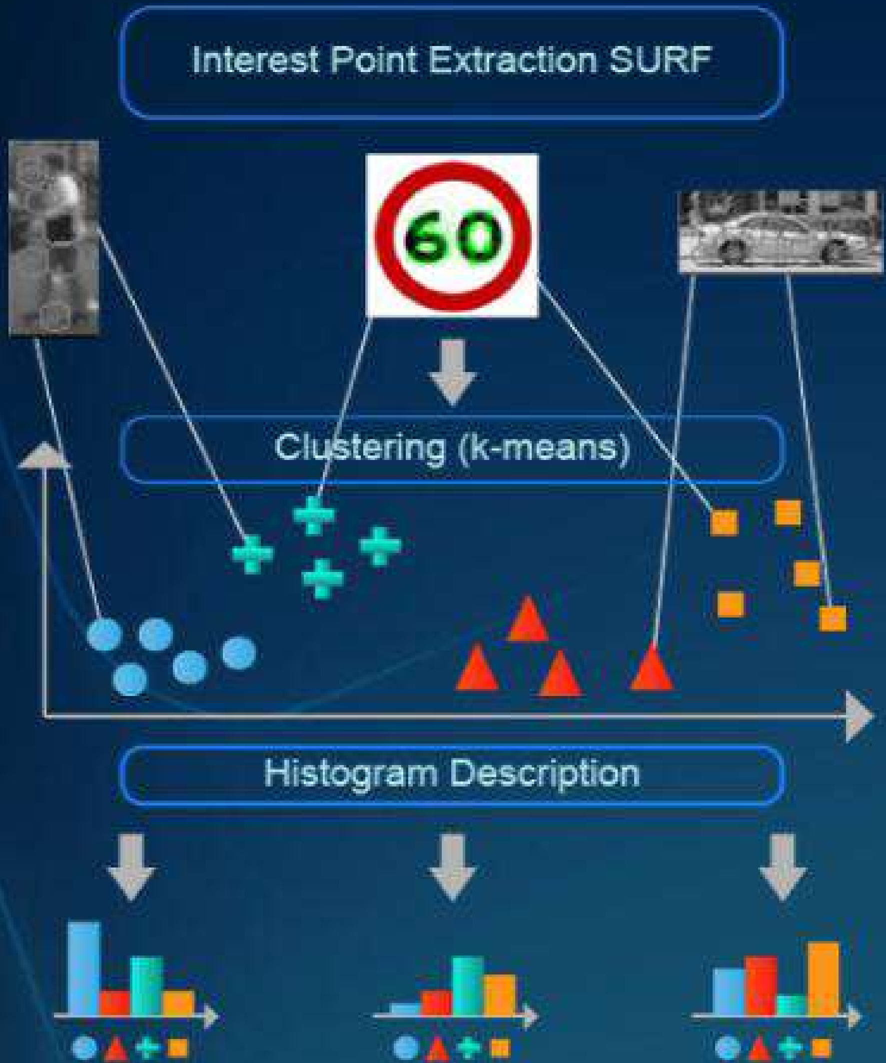
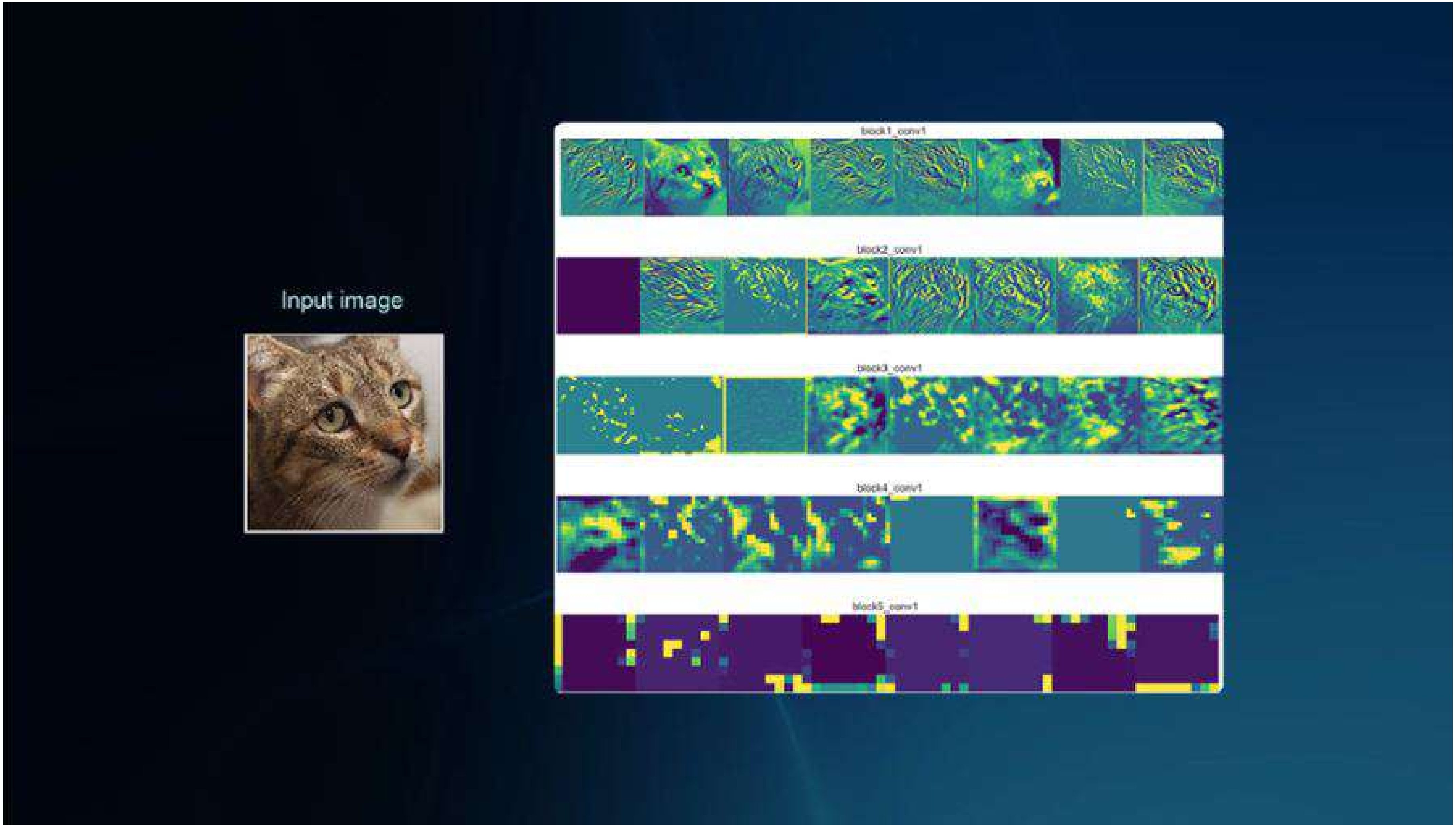


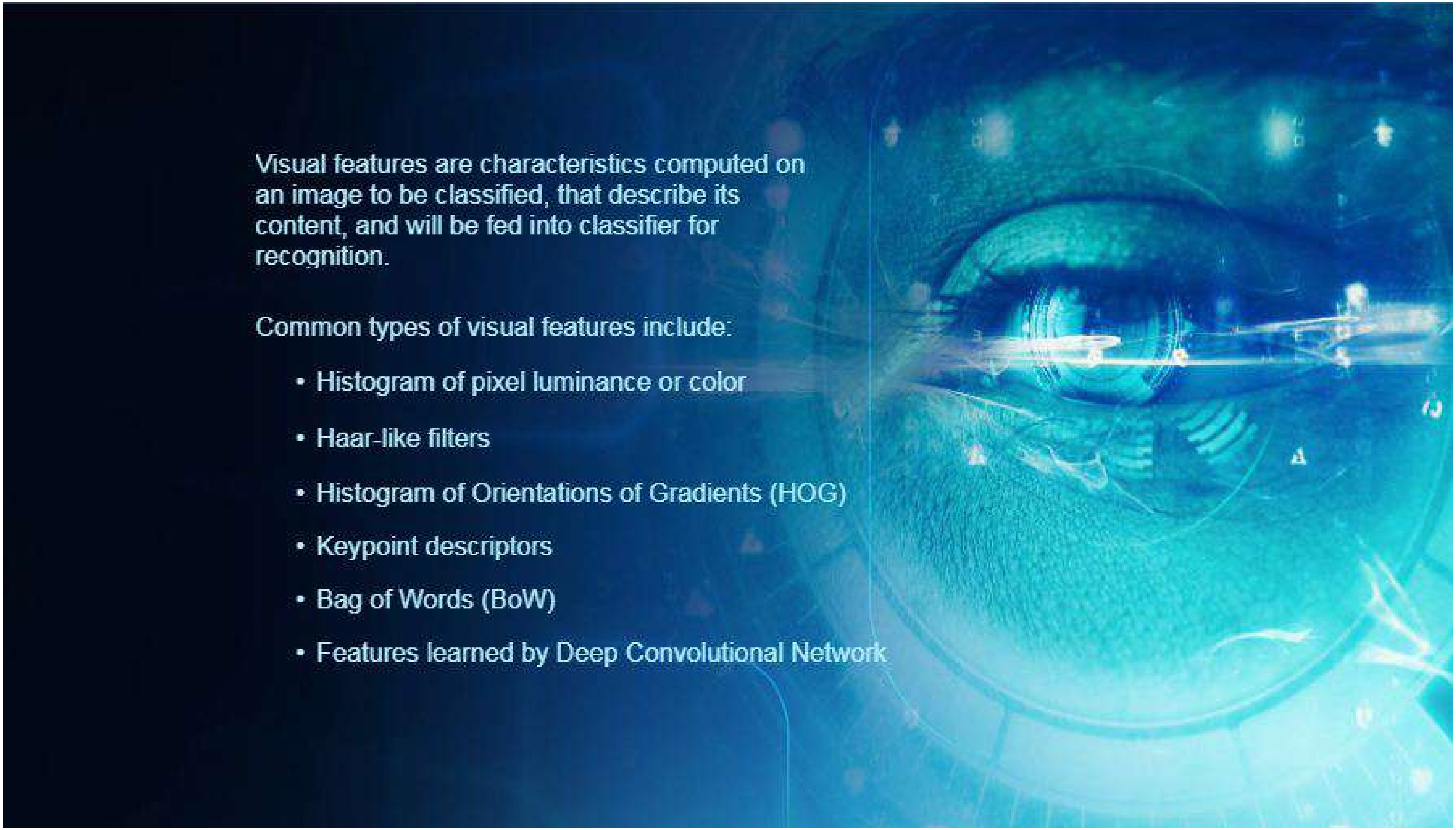
Image
Cell
Gradient Vector
Cell Histogram
(9 bins)
Current cell with the 4
normalization blocks
16 cells
HOG feature
(36 values)
Cell Histogram Generation
Gradient
Calculation
Histogram
Histogram
Normalization
Characterize distribution of contours' orientations
80
Parameters:
• Cell size (in pixels)
• Number of histogram bins for each cell
• Block size (in cells)


Sequence
How does a gradient histogram generation take place? Arrange the boxes below in the correct order.
Correct answer
Click to continue
B) Gradient Calculation
C) Histogram
A) Histogram Normalization

- Adapted to images using keypoints descriptors as a representation of image content:
  - Descriptor vectors are quantized (usually by K-means partitioning) into a codebook of « visual words »
  - A (sub-)image is represented by an histogram of codebook occurences




Input image
block1_conv1
block2_conv1
block3_conv1
block4_conv1
block5_conv1


Visual features are characteristics computed on an image to be classified, that describe its content, and will be fed into classifier for recognition.
Common types of visual features include:
• Histogram of pixel luminance or color
• Haar-like filters
• Histogram of Orientations of Gradients (HOG)
• Keypoint descriptors
• Bag of Words (BoW)
• Features learned by Deep Convolutional Network