Object detection with classifiers

Roman Juránek

POV, 8.10.2019

Overview

- Detection with **boosted classifiers**
- Classifier **cascade**
- **WaldBoost**
- Feature Channels and **ACF detector**
- **Demos - Python (OpenCV, Waldboost), Matlab (ACF)**

Motivation

Fast detection - Boosted detectors are easy to evaluate and use very simple image features. Detection can be parallelized. Detection speed can be in orders of milliseconds on HD image.

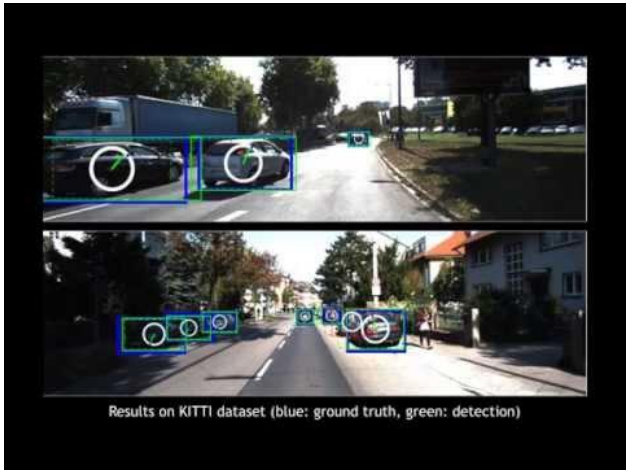
Embedded devices - Easy to implement on different platforms e.g. in FPGA.

Detection of small objects - e.g. face detection from 20x20 px.

Applications

Detection of rigid objects - Frontal faces, License plates, Traffic signs, Cars (front, rear), Pedestrians, Markers

Objects detection (and possibly tracking) done in real-time precedes non real-time tasks e.g. character recognition, facial landmark detection, other object analysis



What happens during detection

Image pre-processing - Image pyramid, filtering, feature map calculation

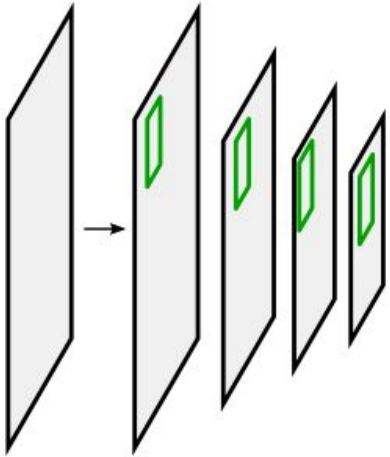
Detection process - Execution of classifier on all positions of feature map

Post-processing - Non-maxima suppression

Image pyramid

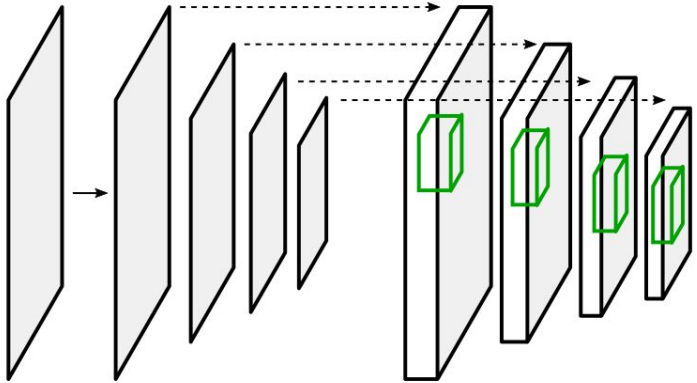
Basic scaling

- Classifier cascade
- WaldBoost
- Usually grayscale image



Feature channels

- More flexible but costly
- ACF detector
- Channels - LUV, HOG, Convolutions
- User-defined



The *Feature channel* scheme can be used in Cascade and WaldBoost classifiers as well

What happens during detection

Image pre-processing - Image pyramid, filtering, feature map calculation

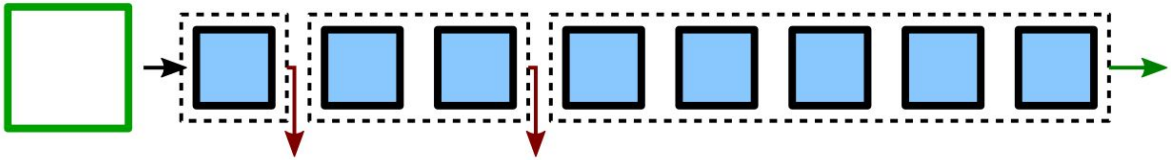
Detection process - Execution of classifier on all positions of feature map

Post-processing - Non-maxima suppression

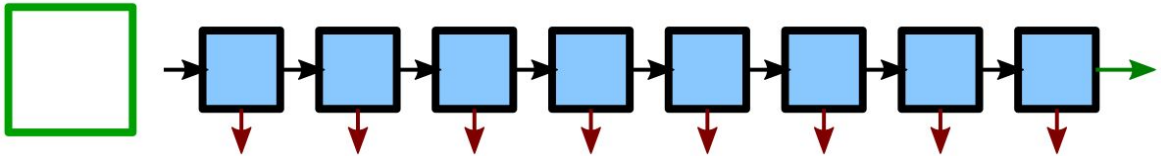
What is the (boosted) classifier?

- Analyzes image **patch** of fixed size 
- Is composed from **weak classifiers** that looks in the patch 
- Rejects (background) or accepts (object) the patch

Cascade
Haar cascade
LBP Cascade



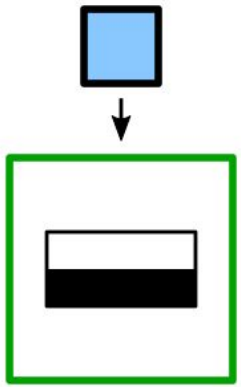
Soft cascade
WaldBoost
ACF detector



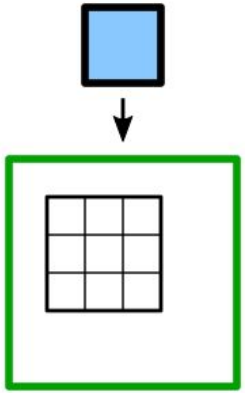
Whats inside a weak classifier?

A weak classifier looks into the image patch, evaluates a feature and produces a response based on the feature value,

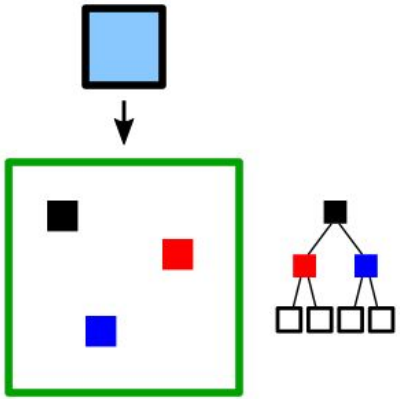
Haar features
Continuous, discretized to **N** bins
Haar Cascade, WaldBoost



LBP - **256** responses
LRD - **17** responses
LBP Cascade, WaldBoost



Decision trees
2^D responses (D is depth)
ACF



Feature **parameters** (shape, position) and **response values** are trained by the training algorithm - usually AdaBoost.

What happens during detection

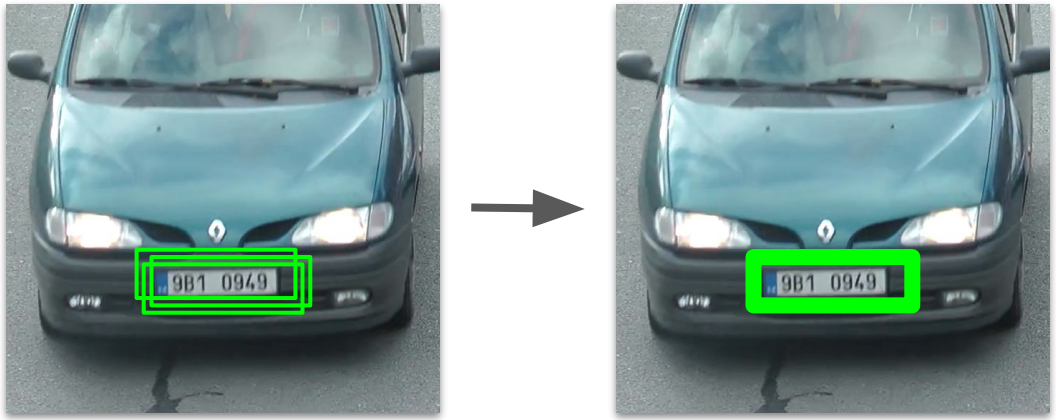
Image pre-processing - Image pyramid, filtering, feature map calculation

Detection process - Execution of classifier on all positions of feature map

Post-processing - Non-maxima suppression

Non-maxima suppression (NMS)

Each object can be “seen” in multiple image patches on different scales. NMS removes nearby detections and generates a single bounding box for each object.



Find overlapping bounding boxes and merge them. Take the one with the **max response**; or **average** the group locations and sizes; … there’s no standard way.

Detection algorithms

Cascades (OpenCV)

- More or less original Viola and Jones detector
- Grayscale pyramid, usually no filtering
- Haar or LBP features
- OpenCV provides tools for training new models

WaldBoost

- Grayscale pyramid
- Soft cascade with optimal thresholds after each weak classifiers
- Haar, LBP, LRD, …

ACF detector

- Feature channels (LUV+gradient+histogram of gradients)
- Constant Soft cascade
- Decision tree-based weak classifiers
- Piotr's Image and Video Toolbox for MATLAB
- Supports extensions like LDCF and custom channels

Demo

- License plate detection
- HaarCascade in Python
- WaldBoost in Python
- ACF detector in MATLAB (you'll need an external toolbox)

↓ DOWNLOAD

https://git.fit.vutbr.cz/ijuranek/pov_demo/archive/v2019.zip

Summary

Boosting-based detectors are good for detection of well-defined rigid objects

Can be implemented to operate extremely fast

Cascade classifiers, Soft cascade

Thank you