ITMO

Objects Detection

Computer Vision

Bag of Words

Bag of Words

Text Classification

- Representation of a document without order: frequencies of words from a dictionary ("bag of words").

Visual Words

- **"Visual word**" is a frequently repeated fragment of the image.

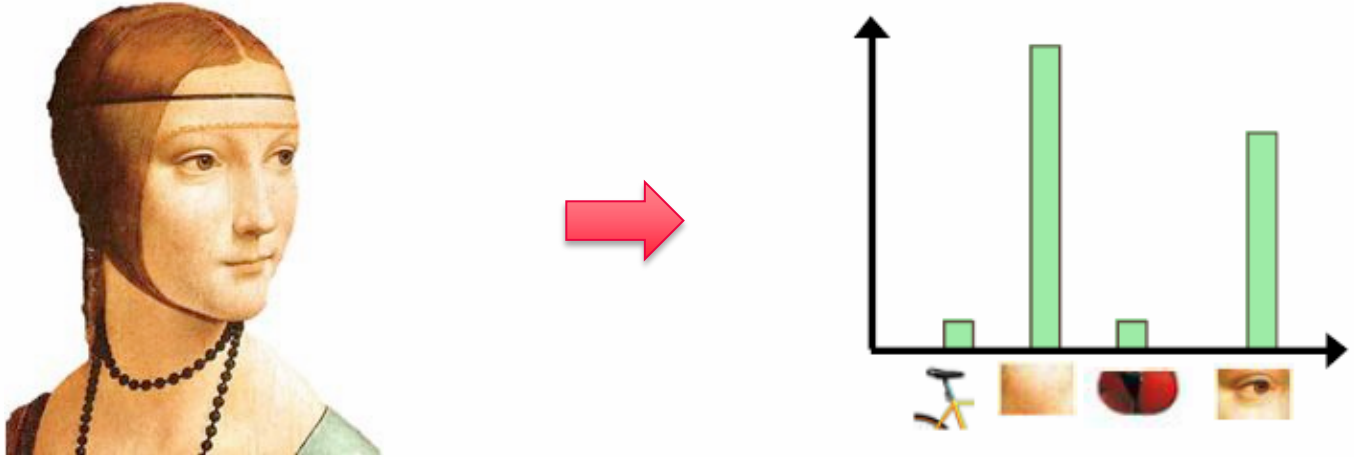
In an image, a visual word can occur:

- only once,
- never,
- many times.

Visual Dictionary

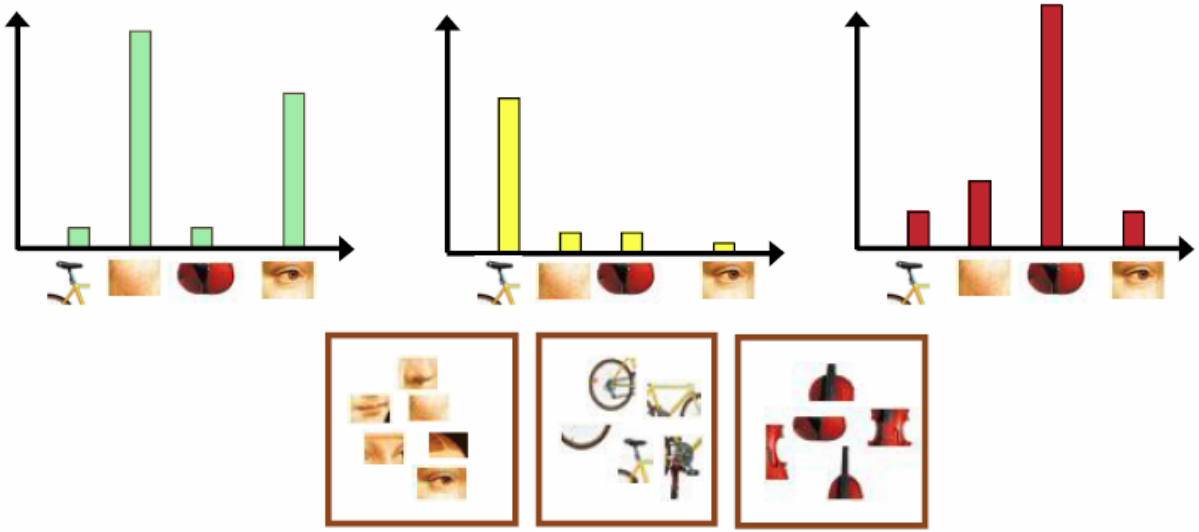
- **A dictionary** is a set of fragments that are often repeated in a collection of images.
- **Compilation of a dictionary:**
  - Make a big list of all the fragments in the entire collection;
  - Divide the entire list into similar groups;
  - All fragments in one group are “instances” of the same word.

Bag of Visual Words



Algorithm

- Fragment extraction;
- Teaching the "visual vocabulary";
- Dictionary fragment quantization;
- Description of the image by the frequencies of "visual words".

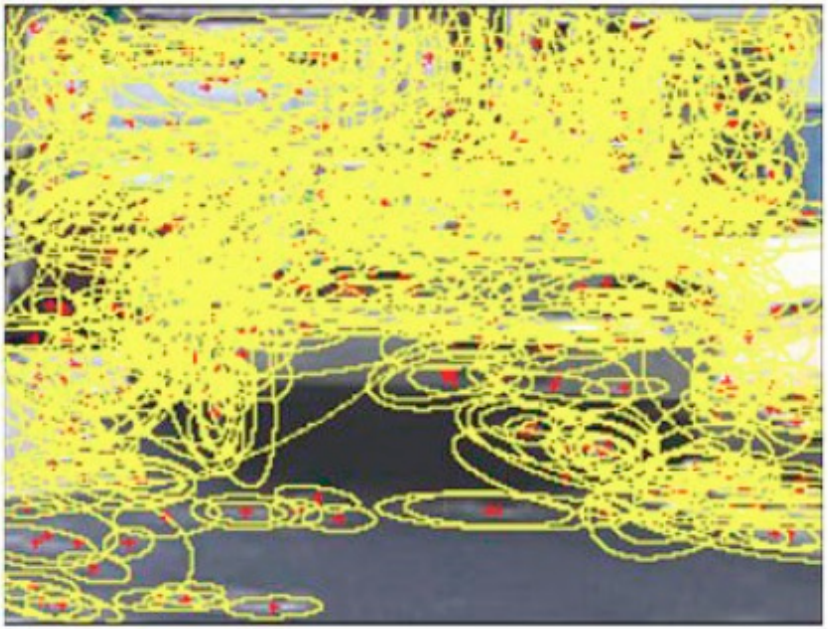


1. Fragment Extraction

- Regular grid (Vogel & Schiele, 2003; Fei-Fei & Perona, 2005)

1. Fragment Extraction

- Regular grid (Vogel & Schiele, 2003; Fei-Fei & Perona, 2005)
- Characteristic points (Csurka et al. 2004; Sivic et al. 2005)

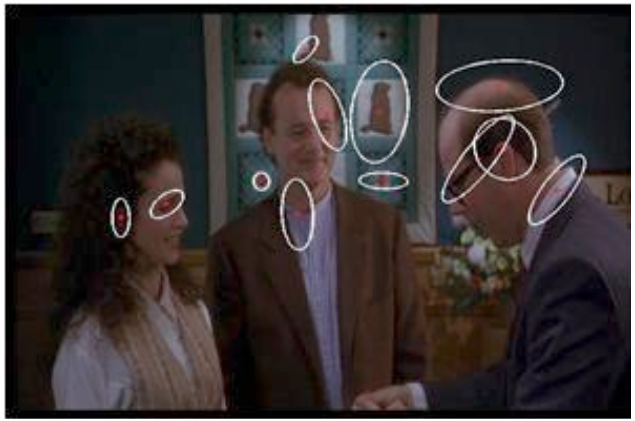


1. Fragment Extraction

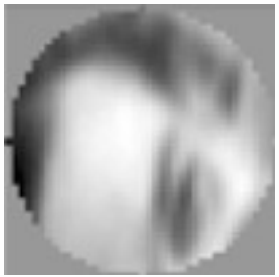
- Regular grid (Vogel & Schiele, 2003; Fei-Fei & Perona, 2005)
- Characteristic points (Csurka et al. 2004; Sivic et al. 2005).
- Other Methods:
  - Random selection (Vidal-Naquet & Ullman, 2002);
  - Segments (Barnard et al. 2003).

1. Fragment Extraction

- Feature vector calculation for each fragment
- SIFT descriptor is usually used



Selected Fragments

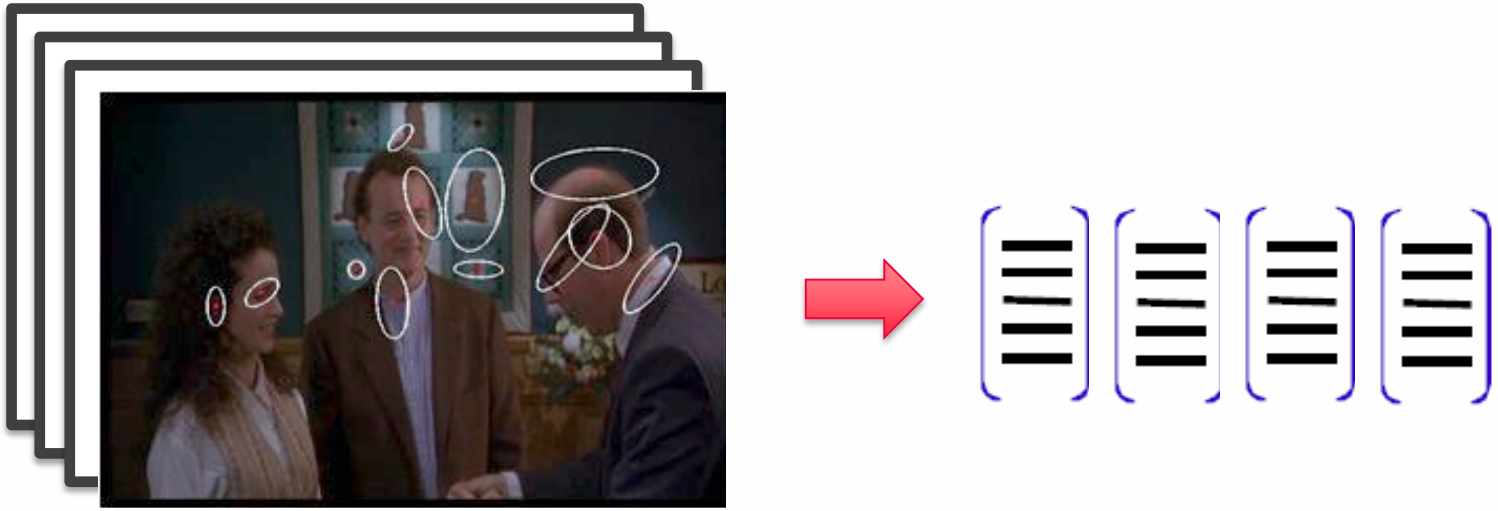


Area normalization

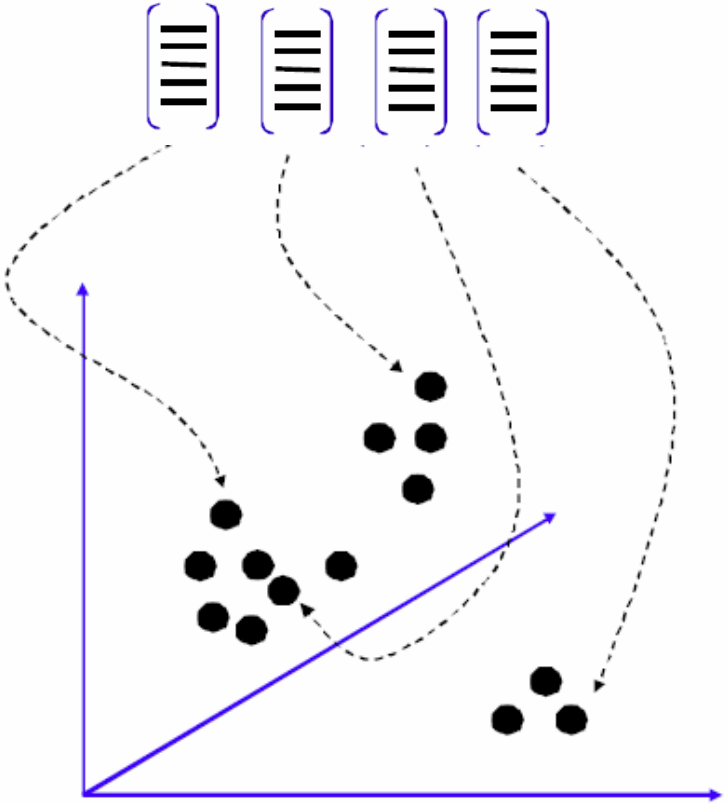
Feature-vector

1. Fragment Extraction

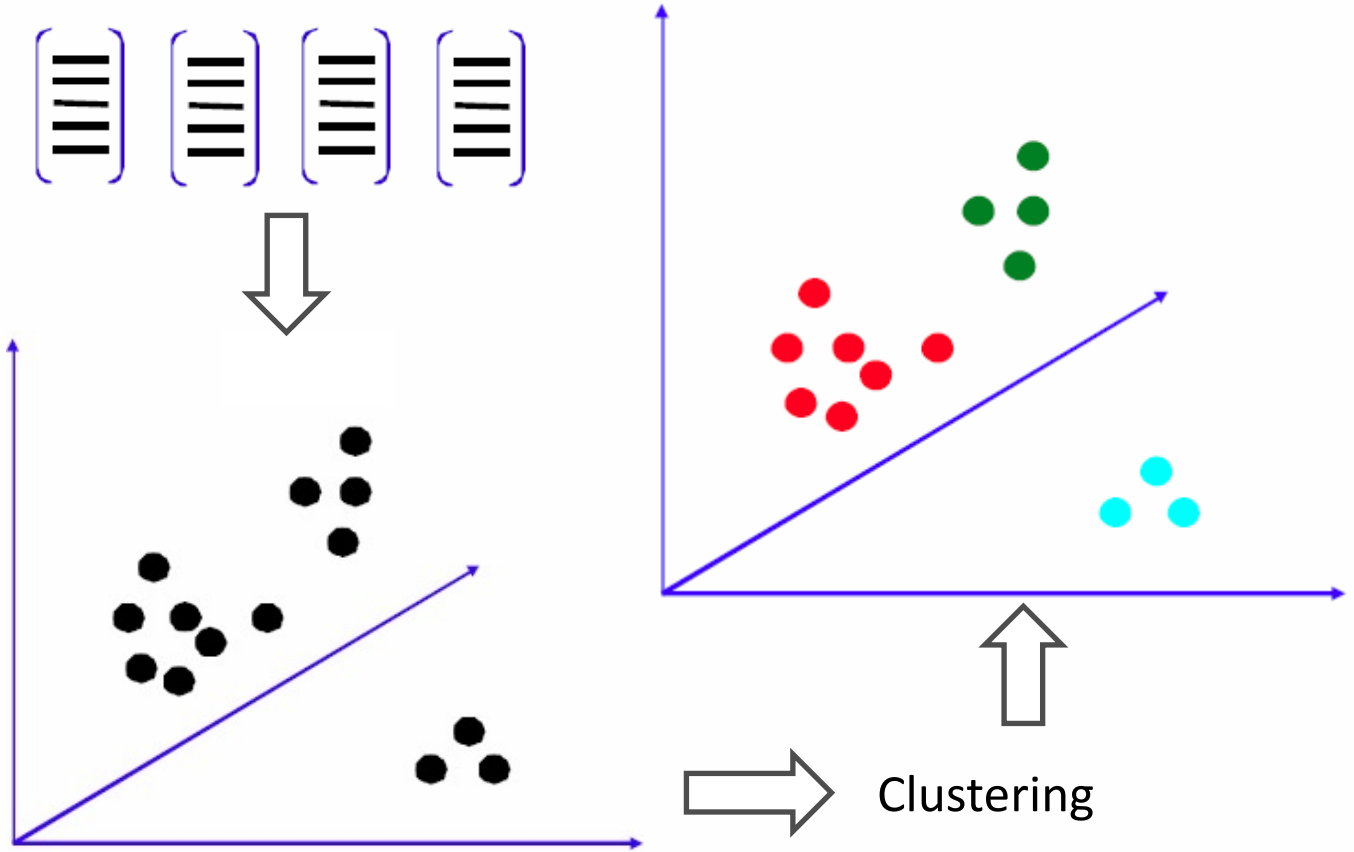
- Unordered list of feature vectors of all fragments of all images in the sample



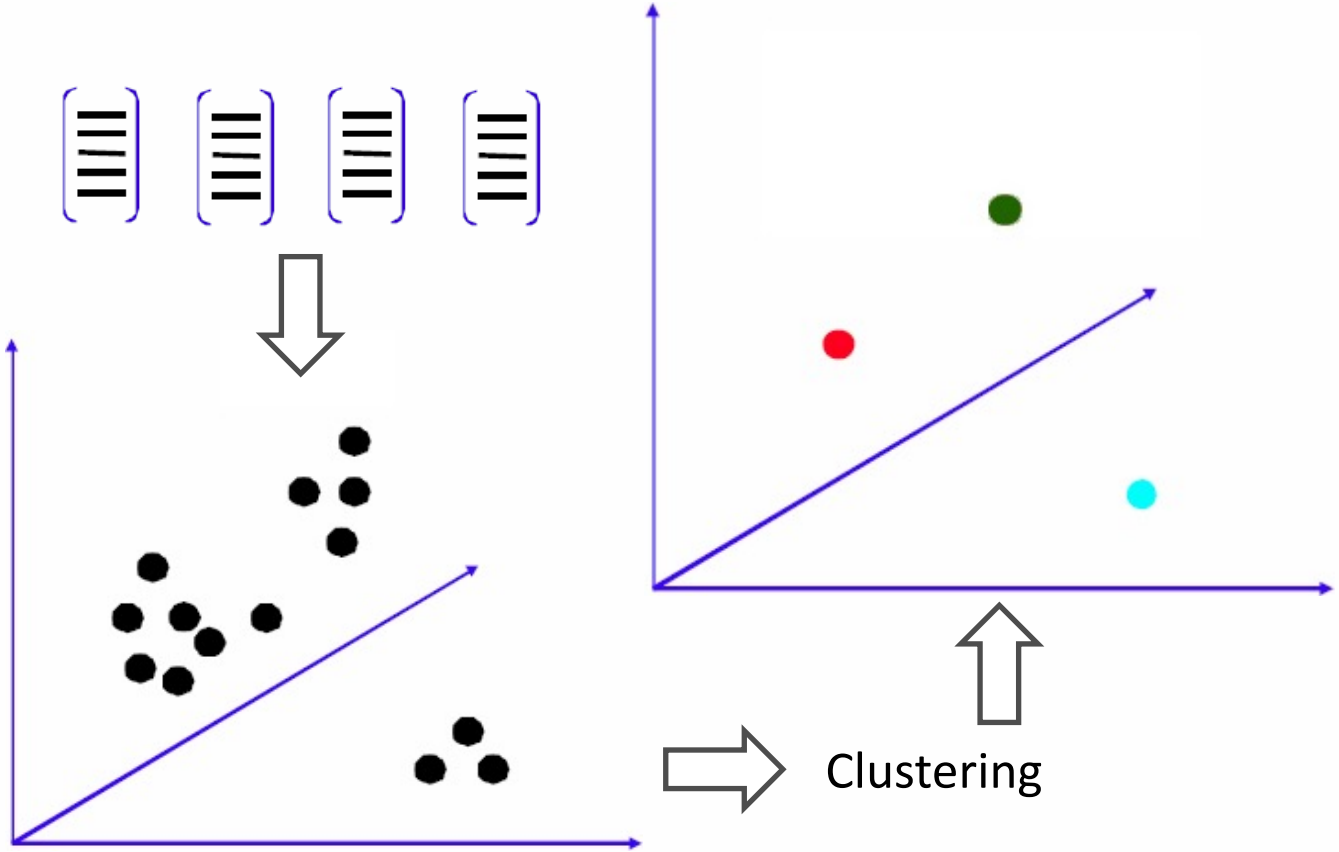
2. Vocabulary Training



2. Vocabulary Training

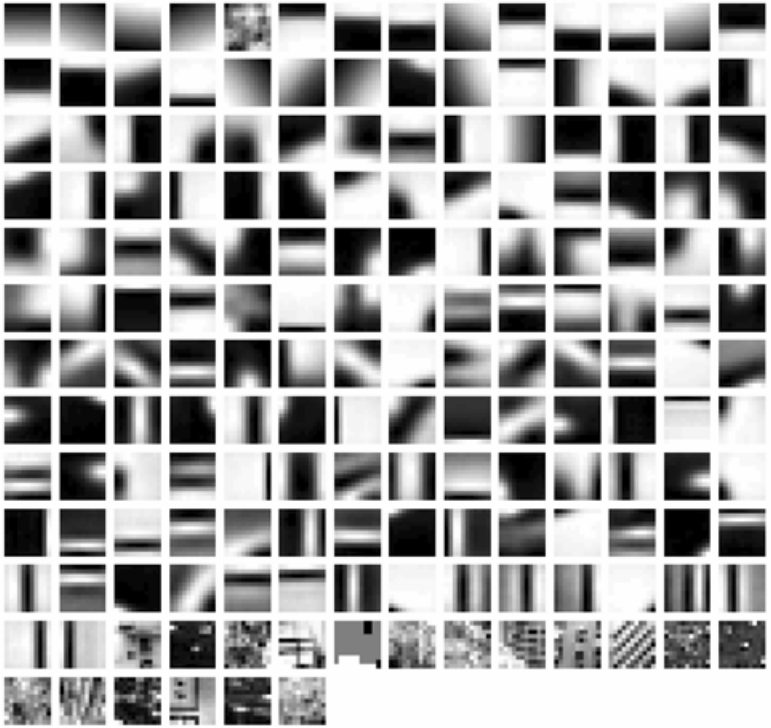


2. Vocabulary Training



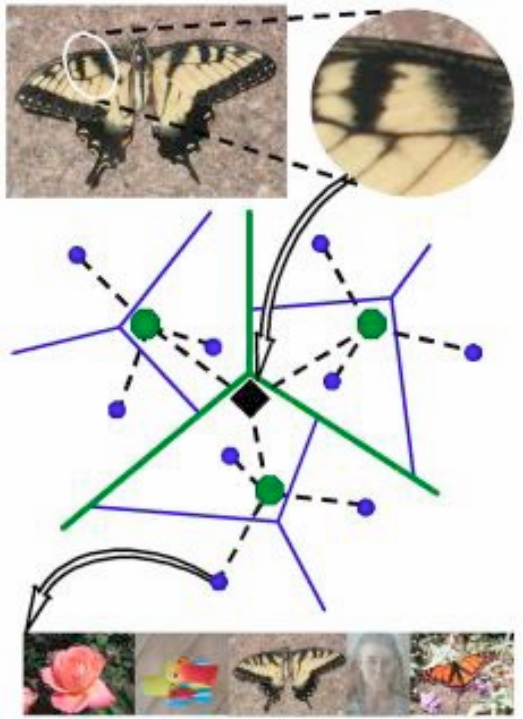
2. Vocabulary Training

- Visual Dictionary Example



2. Vocabulary Training

- **Dictionary size selection:**
  - **Small:** words cannot describe all the features.
  - **Large:** overfitting.
- **Computational complexity:**
  - Dictionary Trees;
  - Approximate Methods;
  - Hashing.



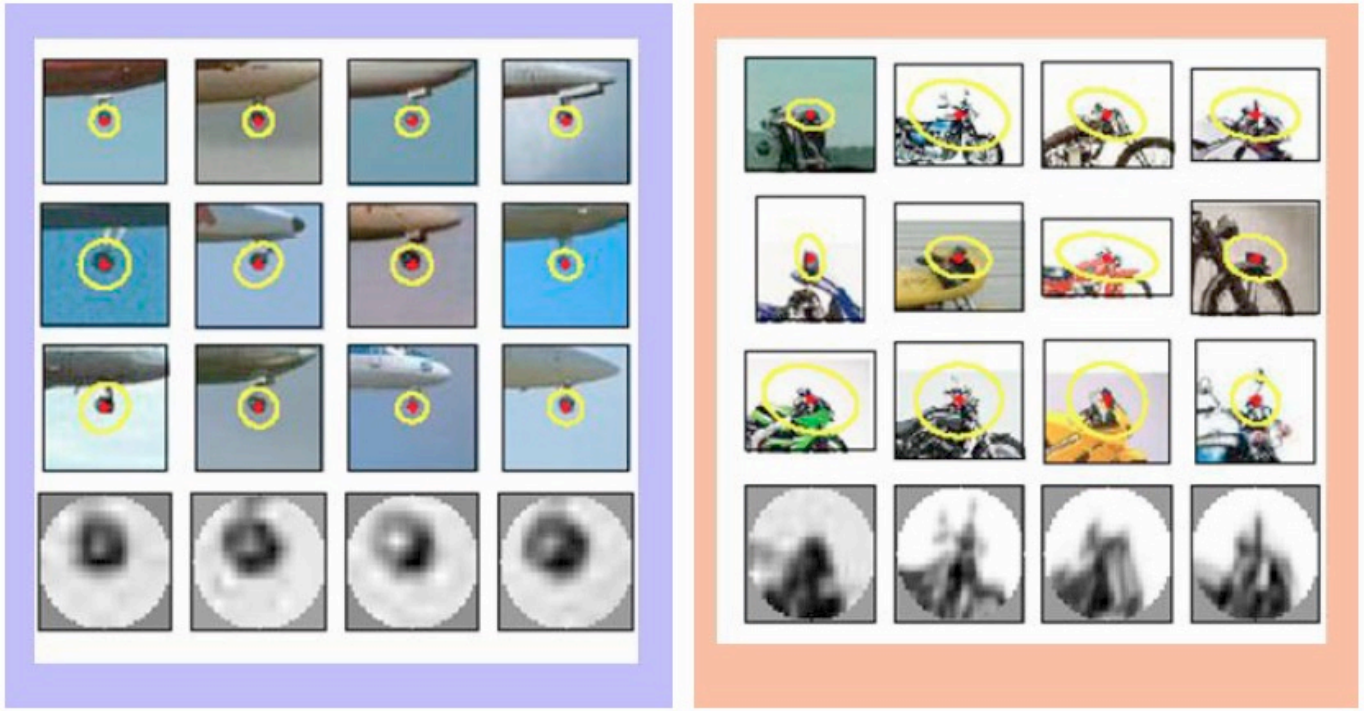
3. Quantization

ITMO

- For each fragment of the image, we find the nearest word in the descriptor in the dictionary.
- In other words, we quantize the image – each fragment is assigned a number from 1 to N (N is the size of the dictionary).
- If the training sample is sufficiently representative, then the "universal" dictionary.
- "Code book" = "Visual dictionary".
- "Code vector" = "Visual word".

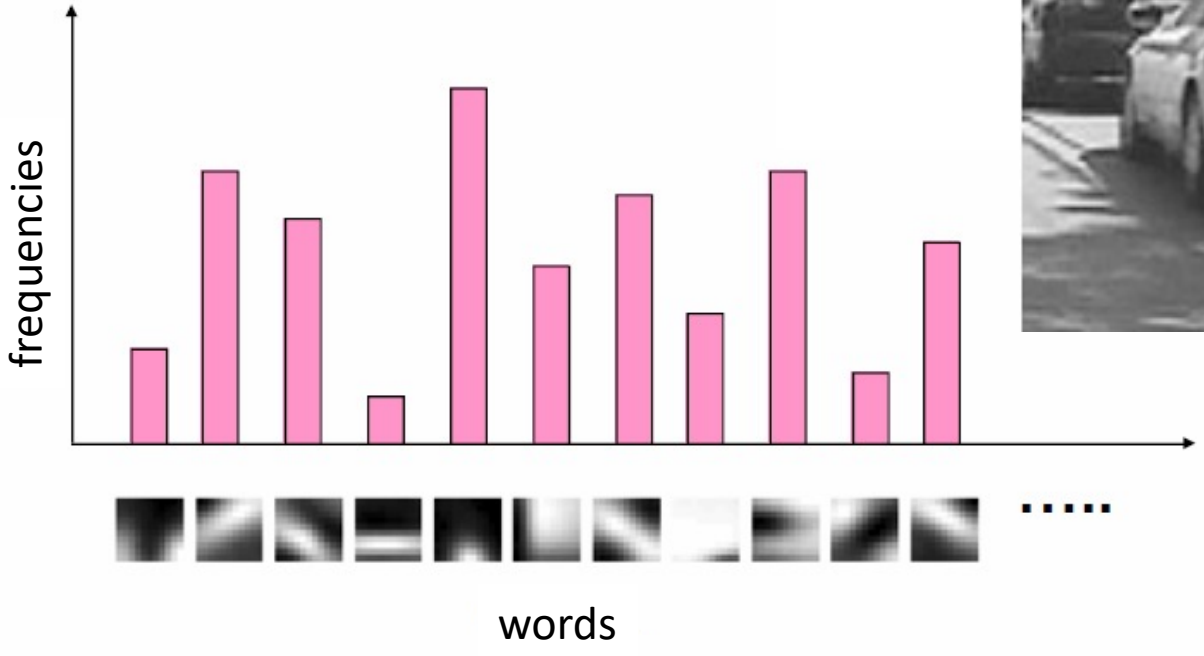
3. Quantization

- Example of visual words



3. Quantization

- Example of a bag of words



Advantages

- A very effective image recognition tool.
- The sparse (by characteristic points) and dense (PHOW) versions are often used together.
- Implementation – Library VLFeat: https://www.vlfeat.org/

Objects Detection

Recognition Tasks

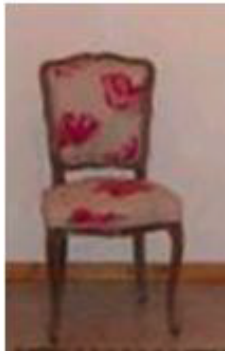
ITMO

- **Image classification**
  - Does the image contain an object (airplane)?

- **Objects detection**
  - Where is the object in the image (airplane)?

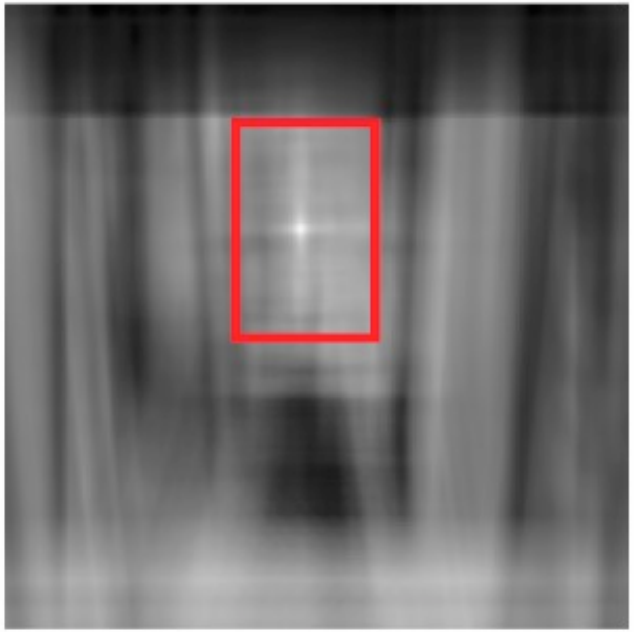
- **Semantic segmentation**
  - Which pixels belong to the object (airplane)?

The Complexity of the Object Detection Task



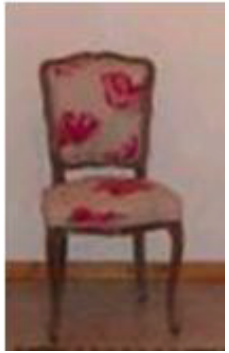
Chair

Detection a chair in an image

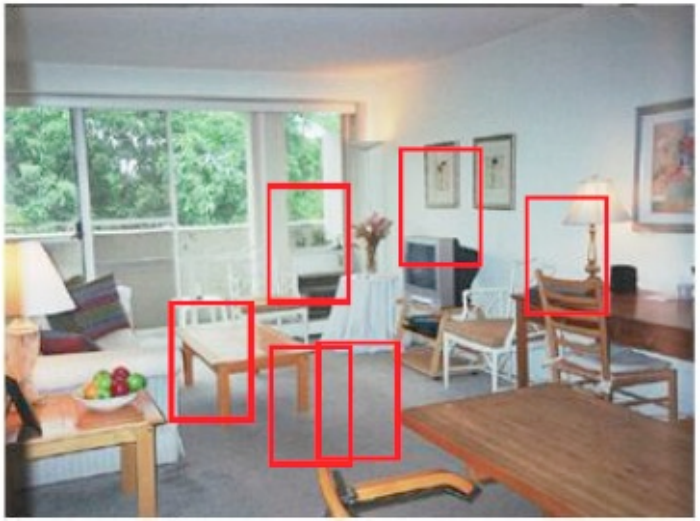


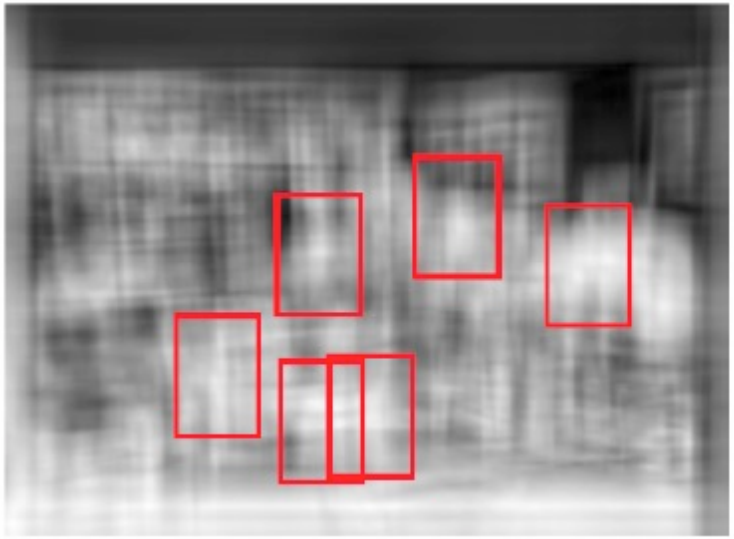
Correlation function output

The Complexity of the Object Detection Task



Chair





Simple pattern matching does not solve the problem

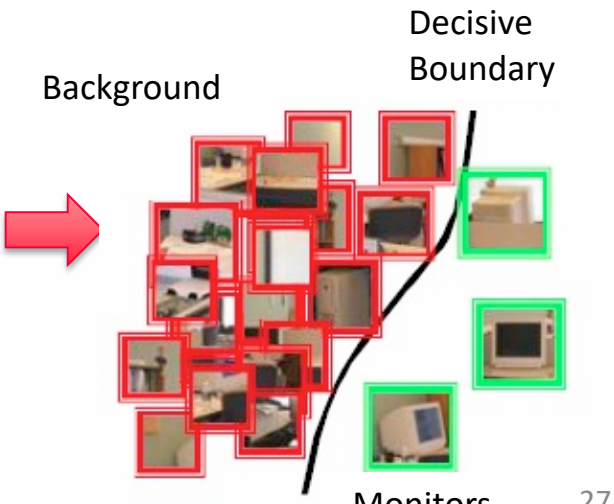
Complex background, different angles,
intra-class mutability of objects

Sliding Window

- Divide an image into multiple overlapping windows.
- The task of object detection is the task of classifying the “object / non-object” window.
- To train the classifier, we collect samples of windows with and without objects.

Find all monitors

Bag of fragments



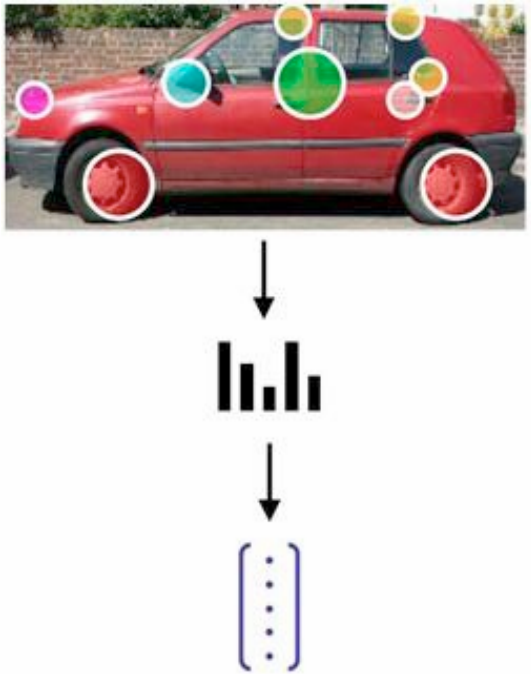
Sliding Window Disadvantages

İTMO

- Frame ratio
- Shift resolution
- Partial overlap
- Multiple responses

Bag of Words

- Let's apply the "bag of words" – a good approach for classifying images.
- Let's apply the SIFT method:
  - Find local features;
  - Calculate the SIFT descriptor.
- Perform clustering with *k*-means and design a dictionary.
- Quantize the features and design a descriptor – a bag of words.
- Train and apply the SVM classifier.

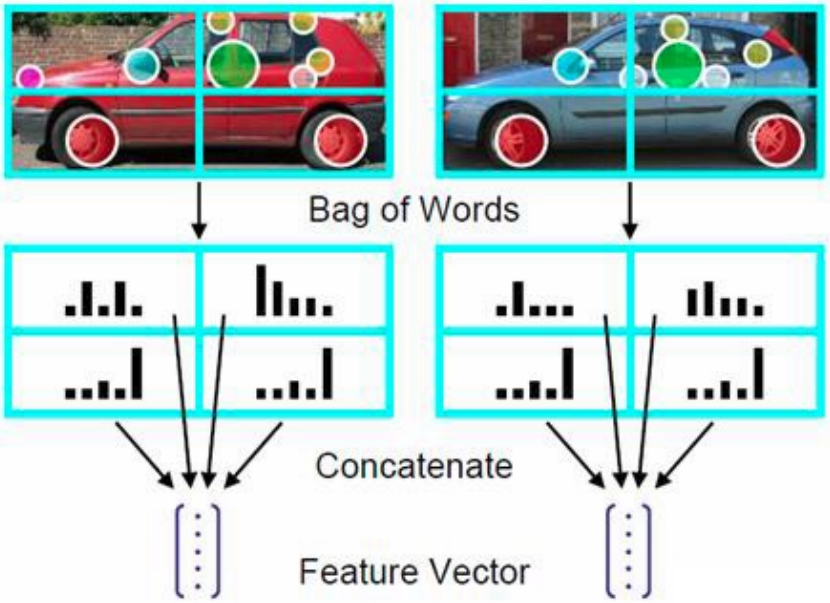


Properties of the BoW Approach

- No explicit modeling of spatial information:
  - «+» invariance to position and orientation in the image;
  - «–» worse distinguishing ability.

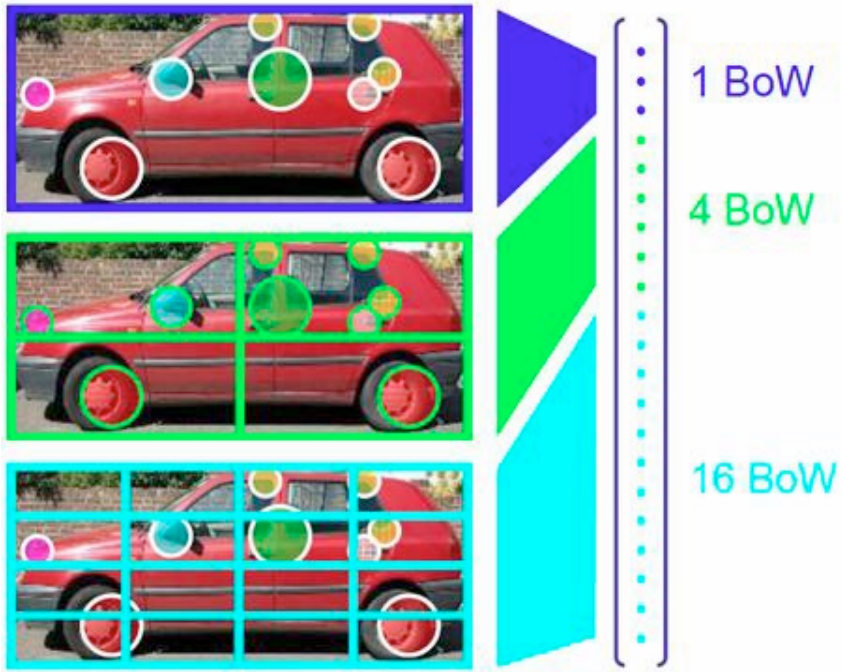
Bag of Words

- Divide the window into regions.
- Keeping a fixed length vector.



Bag of Words

- You can use the "pyramid" of signs.

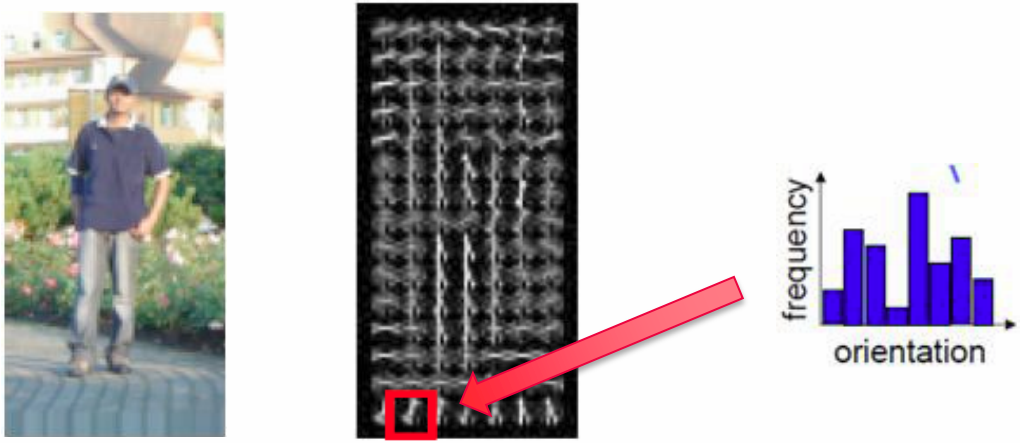


Dense Visual Words

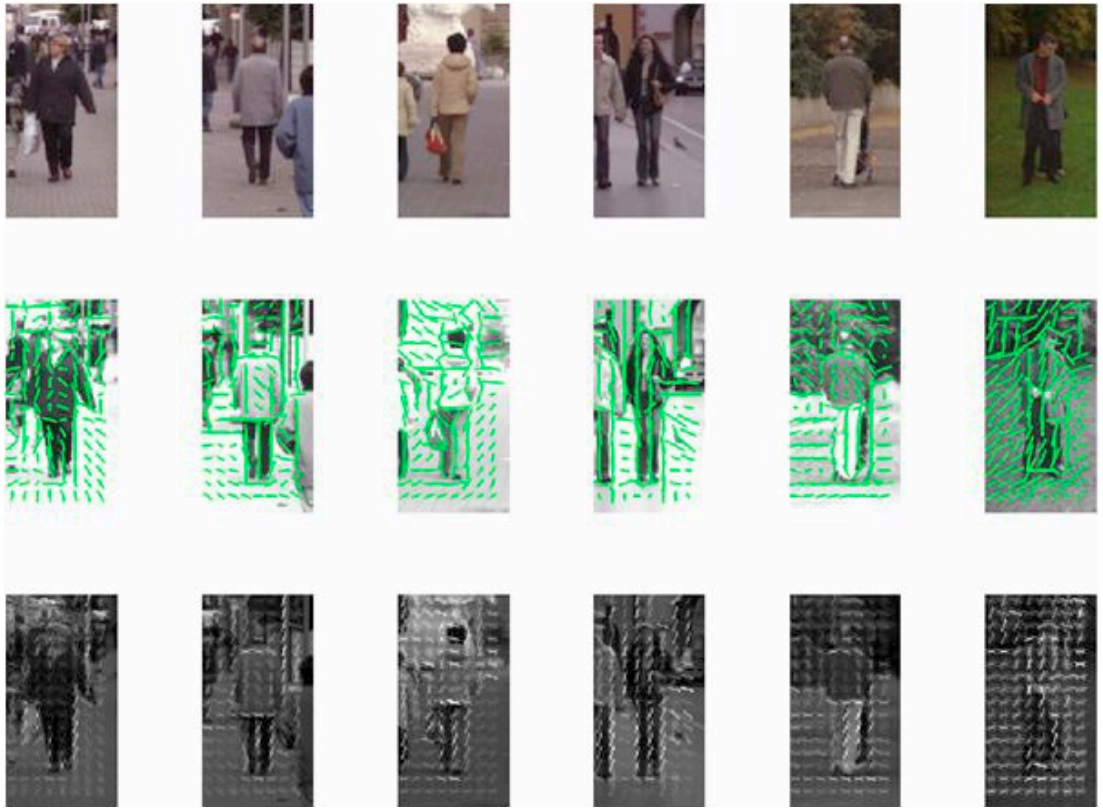
- Extracting individual fragments is good for high invariance, but not relevant for a sliding window.
- Can extract fragments tightly, with overlap:
  - More details, but less invariance;
  - Pyramidal histogram of visual words (PHOW).

Histogram of Oriented Gradients (HOG)

- Let's remove the intermediate stage of fragment quantization and calculate a large SIFT for the entire window.
- The idea was originally proposed for pedestrians.
- Rectangular window 64 x 128 pixels divided into cells 8 x 8 pixels.
- In each cell, we calculate the histogram of gradient orientations (8 bins).



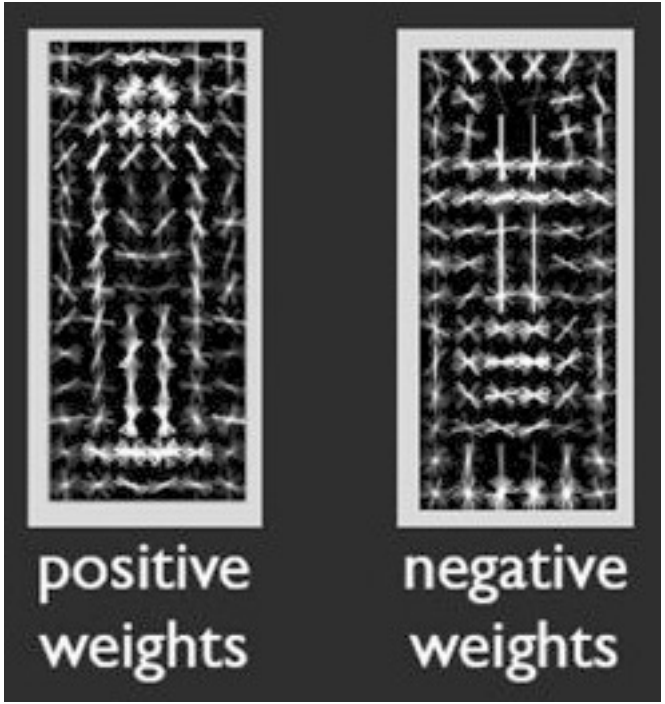
HOG: Example



Support Vector Machine (SVM)

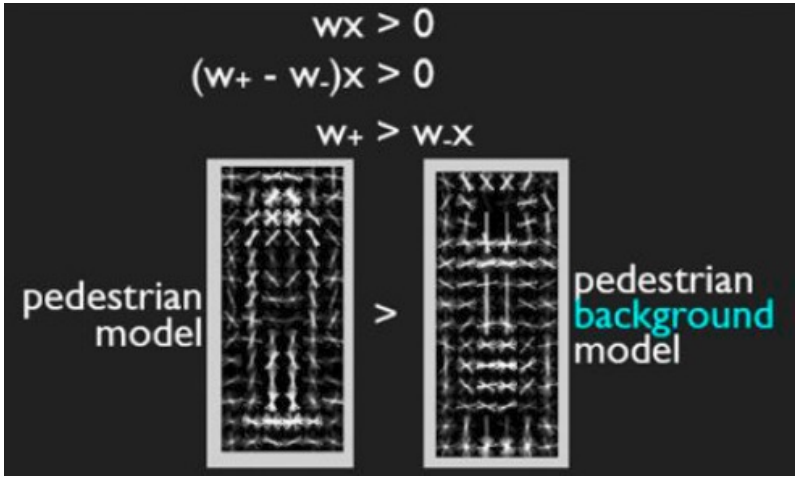
ITMO

- Each support vector is one "difficult" example, a particular pedestrian or background.



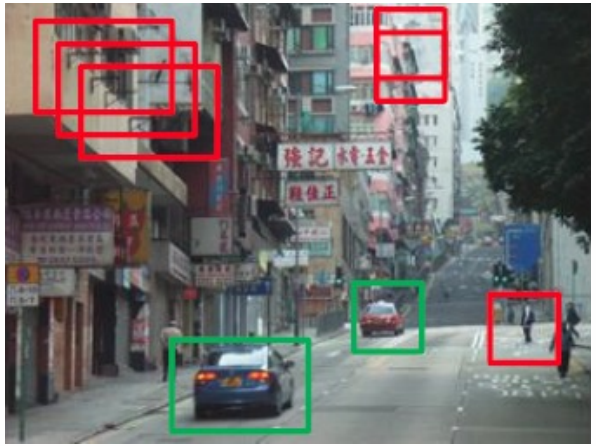
SVM as Image Filters

- We take the scalar product of the support vector and the current window.
- Like linear filter and pattern matching.
- SVM is a set of filters (templates) with weights for the object and for the background.



Detector Training

- The selection of objects is an asymmetric task: there are much fewer objects than "non-objects".
- The class "not an object" is a very complex – you need a lot of different data for training.
- For SVM, the same amount of both background and object is desirable.



Example

- We want to build an "upper body and head" detector.
- We use the HOG + linear SVM scheme.
- Data:
  - 33 movie clips from the Hollywood2 dataset.
  - 1122 frames with marked objects.
- 1-3 people are marked on each frame, 1607 people in total, this is not enough.

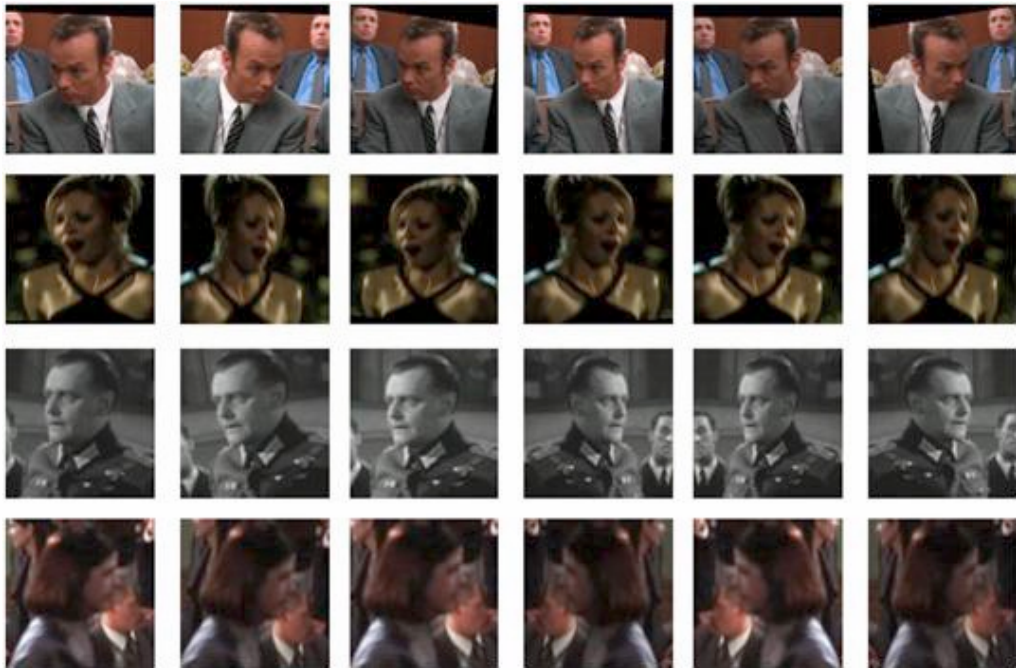
Positive Windows

- What people marked:

Frames show similar position and orientation

Distorted Examples

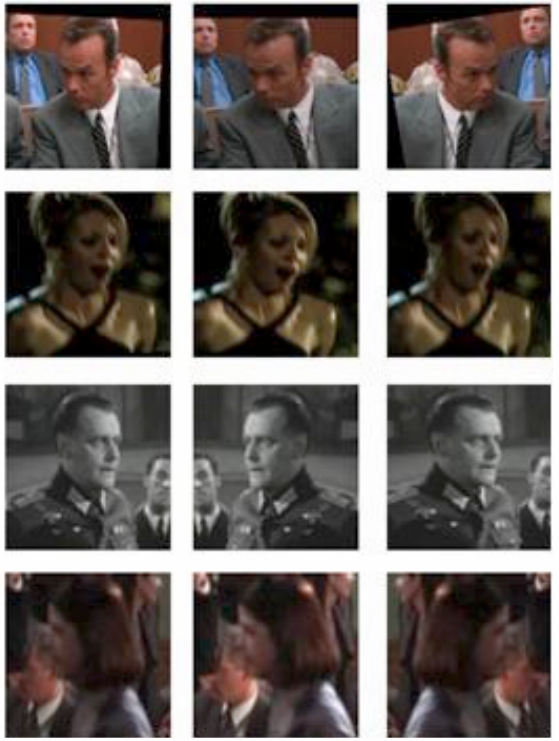
- Let's add data, slightly changing the original:



Small shifts, reflections, rotations and scaling

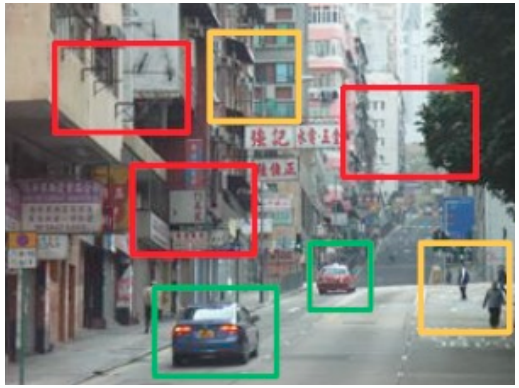
Distorted Examples (Data Augmentation)

- Out of 1607 reference examples, ~32000 jittered examples were obtained.
- From the original 1122 frames, one can collect much more than 32000 negative examples.
- According to SVM, "difficult examples" are needed for the background.



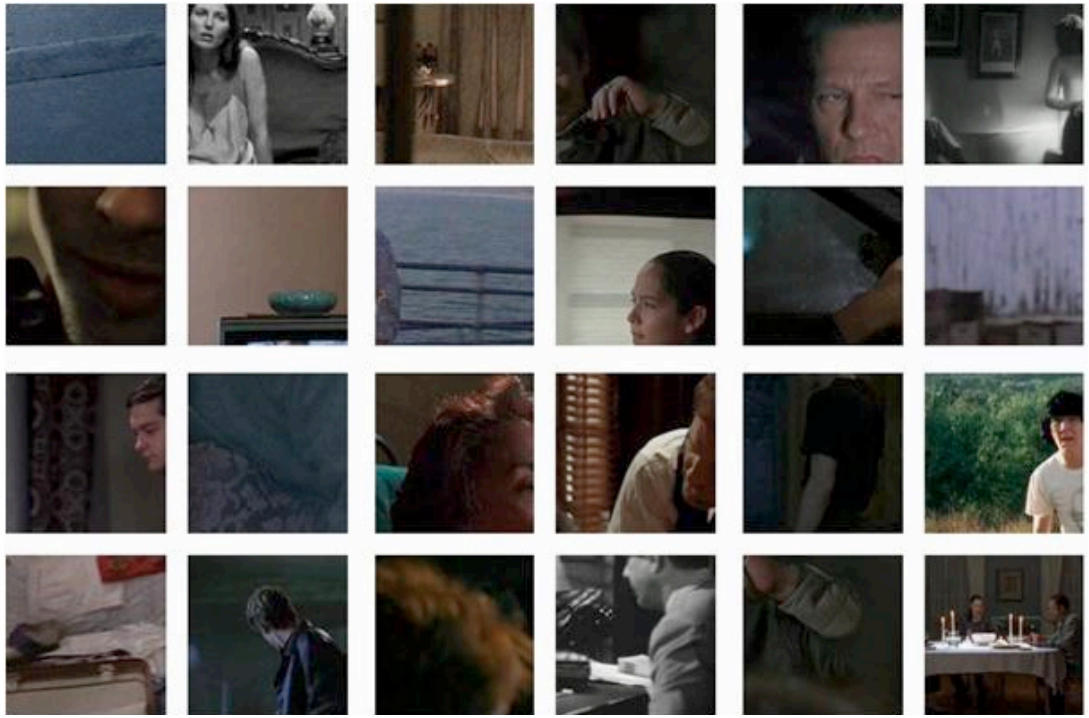
Bootstrapping

- **Algorithm:**
  - Choose negative examples randomly.
  - Train the classifier.
  - Apply to data.
  - Adding False Detections to the sample.
  - Repeat again.
- **Idea:**
  - False detections for the first detector are hard negatives.
  - The background sample will be small, but complex and representative.



Random Background Fragments

- Background selection elements for the first iteration:

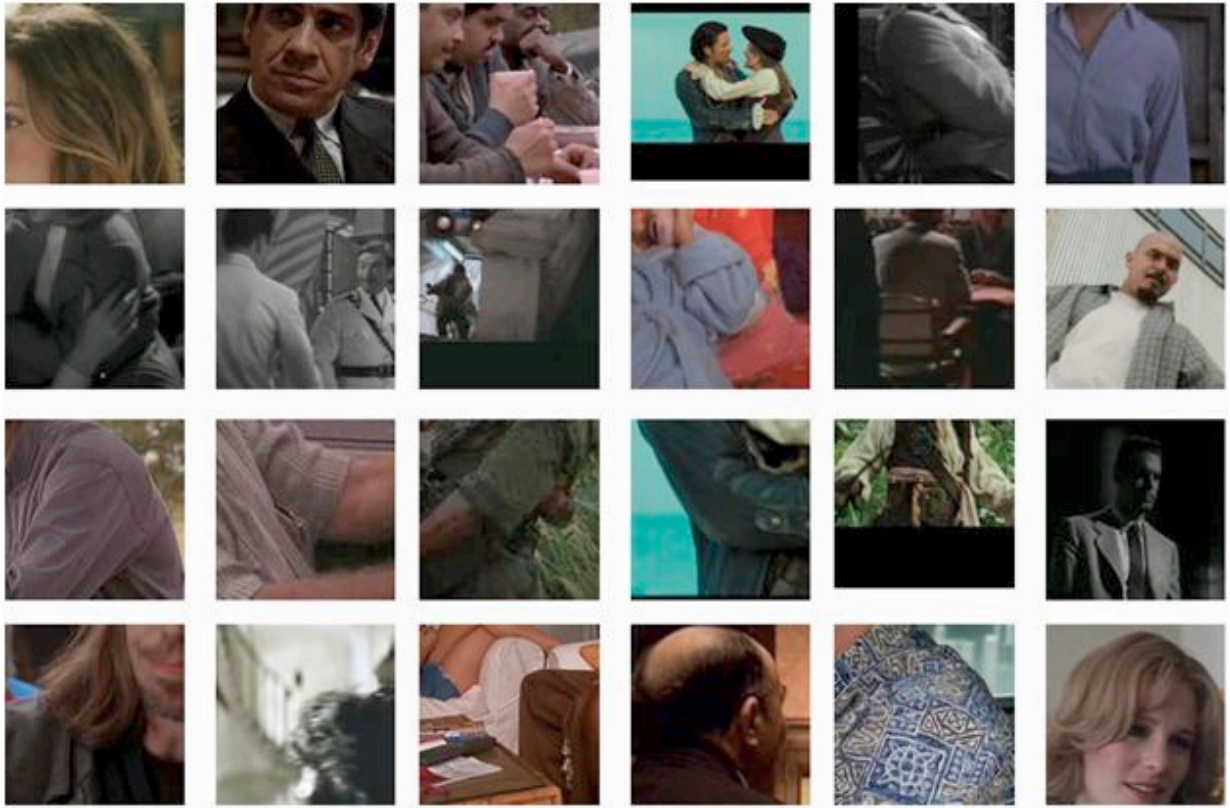


First stage

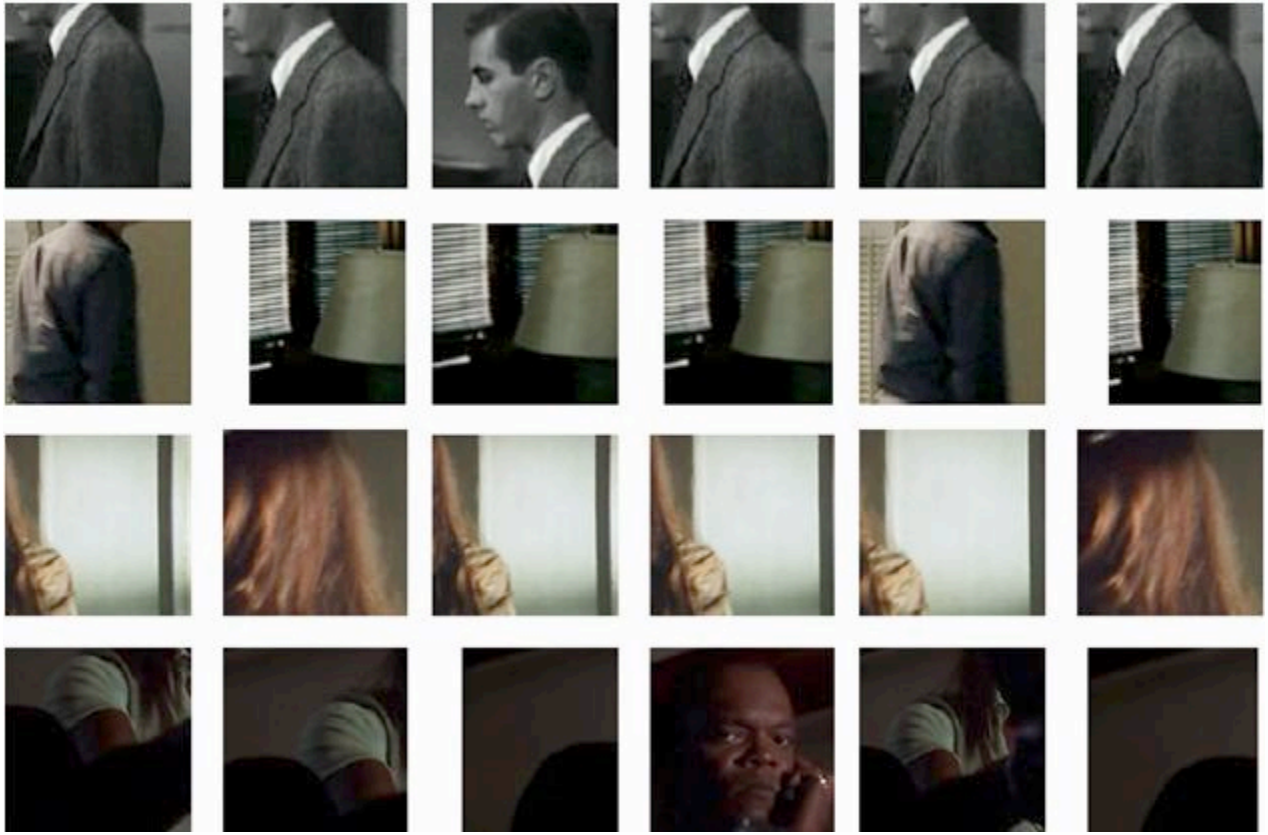
- We are looking for false positives with a high rating.
- We use them as difficult negative examples.

Difficult negative example

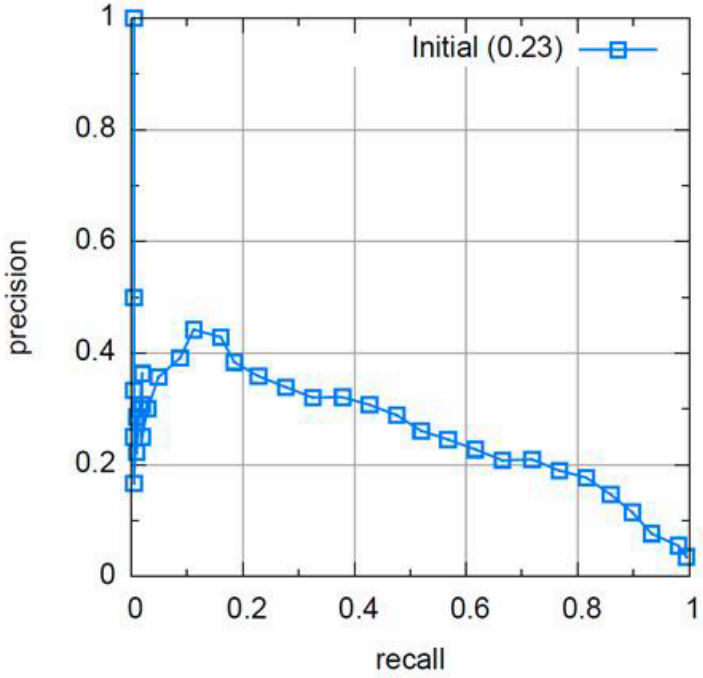
Difficult Negative Examples

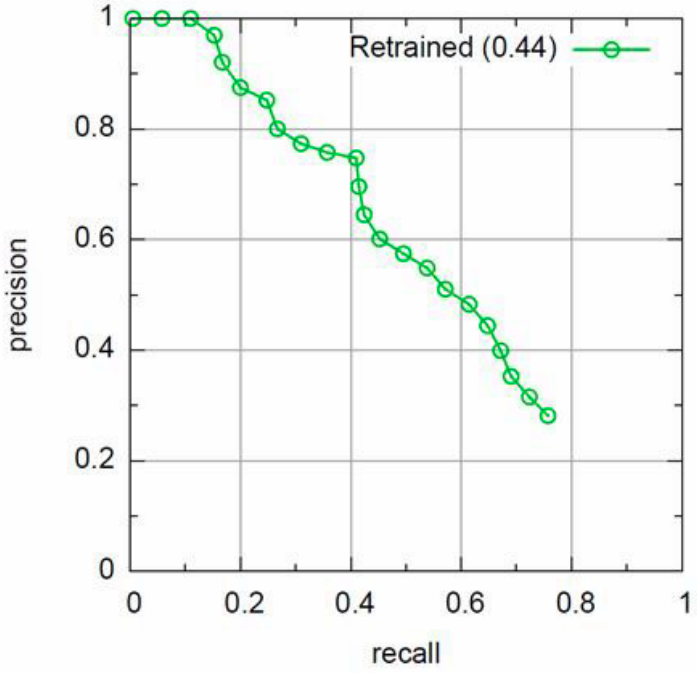


Difficult Negative Examples



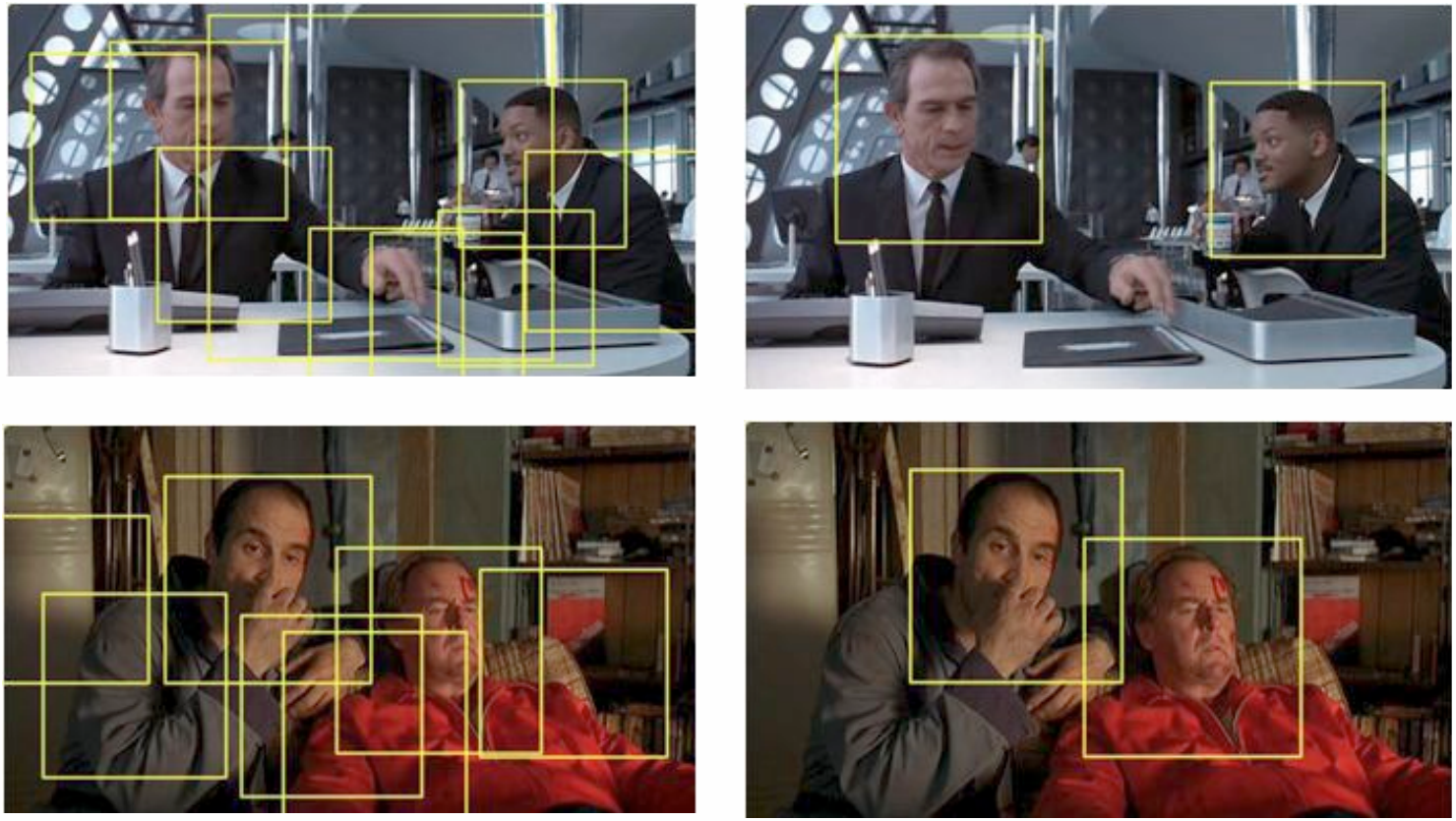
Quality Comparison



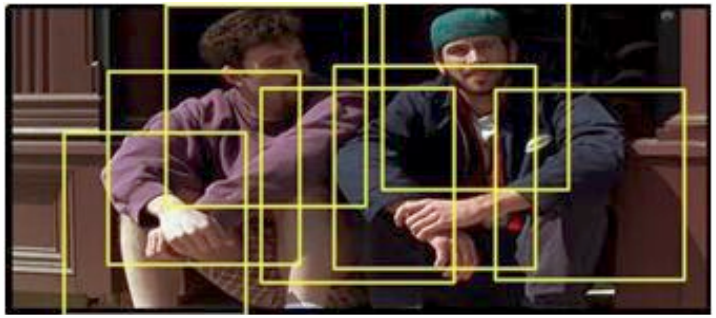


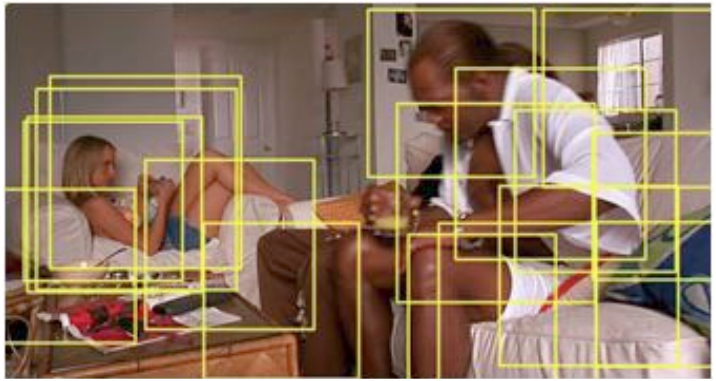
After retraining

Quality Comparison

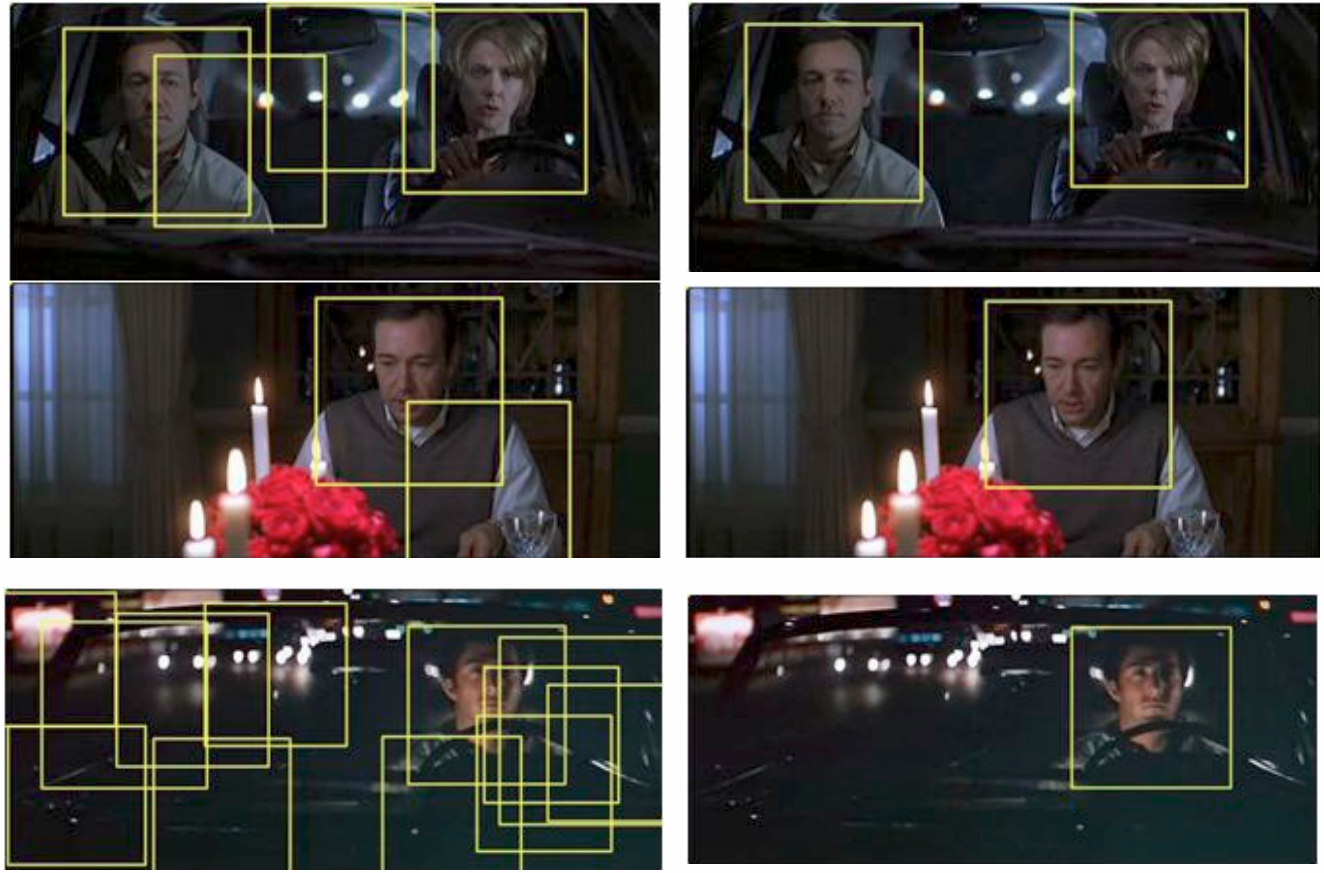


Quality Comparison





Quality Comparison



Final Algorithm for Objects Detection

ITMO

1. We use a sliding window.
2. Calculate feature vector based on HOG:
   - Split the window into cells;
   - In each cell, we calculate the histogram of the orientation of the gradients.
3. We train linear SVM.
4. For learning:
   - We multiply the reference examples of objects;
   - Using the bootstrapping scheme to select background examples:
     - At the first stage, we take random windows for the background;
     - At the next stages, we select false positives of the detector as “difficult” examples.

Requirements for Objects Detection Algorithm

- For a 1MP image, you need to view about 1M windows (for example, for the case of faces).
- One image usually has 0-10 faces.
- To avoid false positives, the type II error should be below 10^{-6}.
- Need to quickly discard false windows.

Test on Lectures 9-10

CS3 Test

AT3 Test

THANK YOU FOR YOUR TIME!

ITMO more than a UNIVERSITY

s.shavetov@itmo.ru