

HOGgles

Visualizing Object Detection Features

S.Bak

HOGgles: Visualizing Object Detection Features

[Carl Vondrick](#) [Aditya Khosla](#) [Tomasz Malisiewicz](#) [Antonio Torralba](#)
Massachusetts Institute of Technology

[Oral presentation at ICCV 2013](#)

We introduce algorithms to visualize feature spaces used by object detectors. The tools in this paper allow a human to put on "HOG goggles" and perceive the visual world as a HOG based object detector sees it.

Check out this page for a few of our experiments, and read [our paper](#) for full details. Code is available to make your own visualizations.

Quick Jump:

1. [Code](#)
2. [Overview](#)
3. [Why did my detector fail?](#)
4. [Visualizing Top Detections](#)
5. [What does HOG see?](#)
6. [Eye Glass](#)
7. [Visualizing Learned Models](#)
8. [Recovering Color](#)
9. [Videos](#)
10. [HOGgles](#)

Inverting and Visualizing Features for Object Detection*

Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, Antonio Torralba
Massachusetts Institute of Technology
{vondrick, khosla, tomasz, torralba}@csail.mit.edu

Abstract

We introduce algorithms to visualize feature spaces used by object detectors. The tools in this paper allow a human to put on "HOG goggles" and perceive the visual world as a HOG based object detector sees it. We found that these visualizations allow us to analyze object detection systems in new ways and gain new insight into the detector's failures. For example, when we visualize the features for high scoring false alarms, we discovered that, although they are clearly wrong in many ways, they do look extremely similar to true positives in feature space. This result suggests that many of these false alarms are caused by our choice of feature space, and indicates that creating a better learning algorithm or building better detectors is unlikely to correct these errors. By visualizing feature spaces, we can gain a more intuitive understanding of our detection systems.



Figure 1: An image from PASCAL and a high scoring detection from DPM [10]. Why did the detector fail?



Figure 2: We show the way for the false car detection from Figure 1. On the right, we show our visualization of the HOG features for the same patch. Our visualization reveals that this false alarm actually looks like a car in HOG space.

1. Introduction

Figure 1 shows a high scoring detection from an object detector with HOG features and a linear SVM classifier trained on PASCAL. Despite our field's incredible progress in object recognition over the last decade, why do our detectors still think that lakes look like cars?

Unfortunately, computer vision researchers are often unable to explain the failures of object detection systems. Some researchers blame the features, others the training set, and even more the learning algorithm. Yet, if we wish to build the next generation of object detectors, it seems critical to understand the failures of our current detectors.

In this paper, we introduce tools to explain many of the failures of object detection systems. We present algorithms to visualize the feature spaces of object detectors. Since features are too high dimensional for humans to directly inspect, our visualization algorithms work by projecting features back to natural images. We found that these projections

provide an intuitive and accurate visualization of the feature spaces used by object detectors.

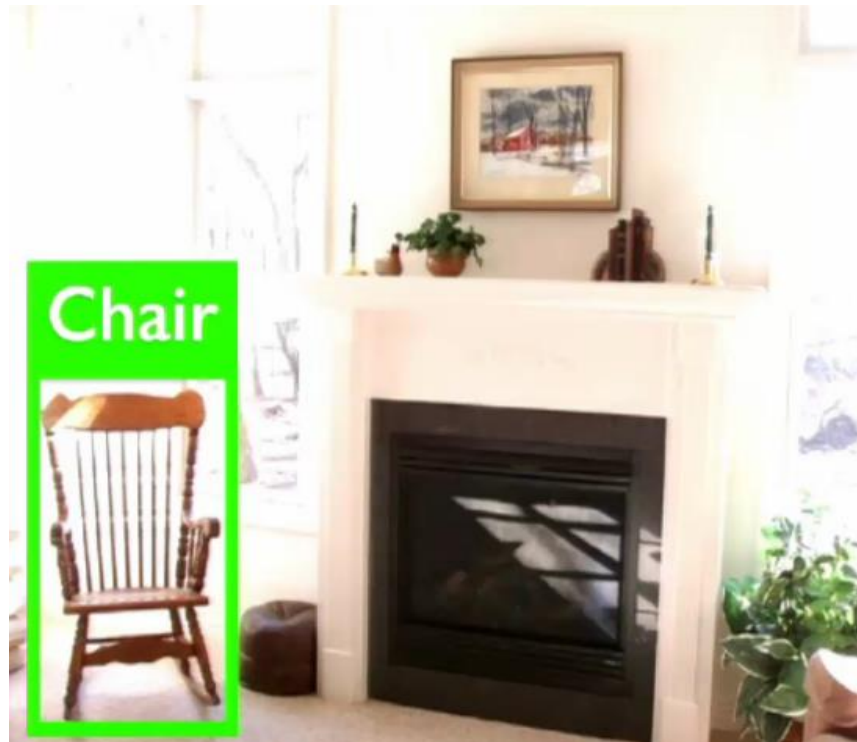
Figure 2 shows the output from our visualization on the features for the false car detection. This visualization reveals that, while there are clearly no cars in the original image, there is a car hiding in the HOG description. HOG features see a slightly different visual world than what we see, and by visualizing this space, we can gain a more intuitive understanding of our object detectors.

Figure 3 invents new top detections on PASCAL for a few categories. Can you guess which are false alarms? Take a minute to study the figures since the next sentence might ruin the surprise. Although every visualization looks like a true positive, all of these detections are actually false alarms. Consequently, we can conclude that, even with a better learning algorithm or more data, these false alarms will likely persist. In other words, the failures are in feature. The principal contribution of this paper is the presentation

*This is a pre-print of our conference paper. We made a publicly available version of this paper on 5/15/13. [Click here for more info.](#)

[Download Paper](#)

Read about it in the [MIT news!](#)
Download [slides](#) or [watch](#)



Chair



Car



Aeroplane

Aeroplane

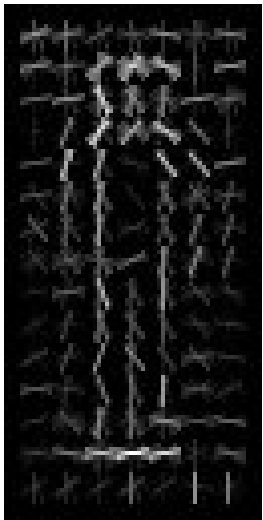


Person

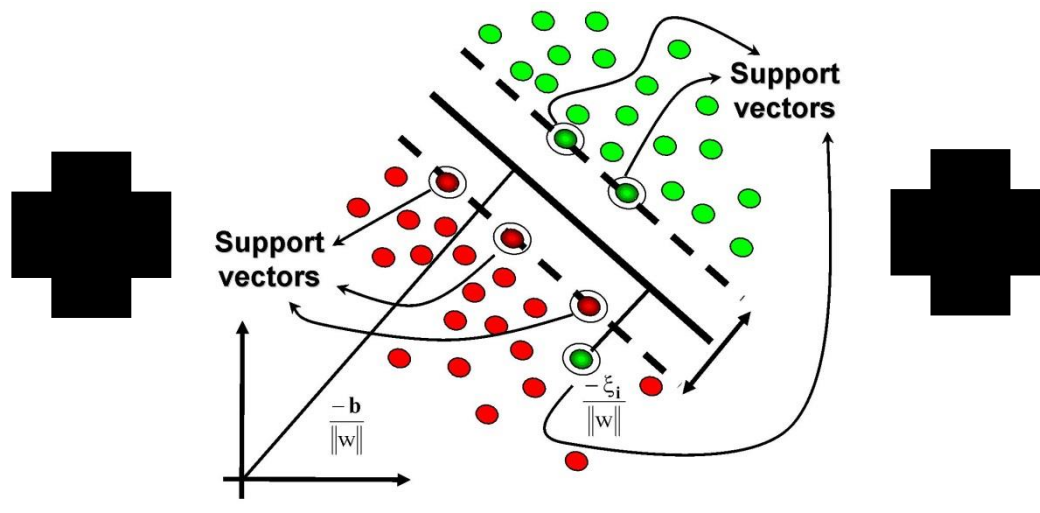


Common detectors...

Feature (e.g. HOG)



Classifier (e.g. SVM)



Training data



DETECTOR



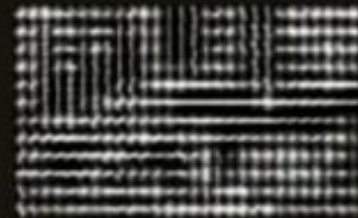
“Some researchers blame **the features**, others the **training set**, and even more the **learning algorithm**.”

What information does HOG have?

Image



HOG



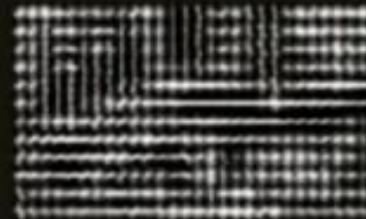
Nearest Neighbors

What information does HOG have?

Image



HOG



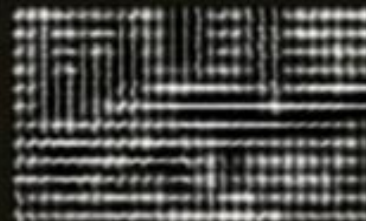
Nearest Neighbors

What information does HOG have?

Image



HOG



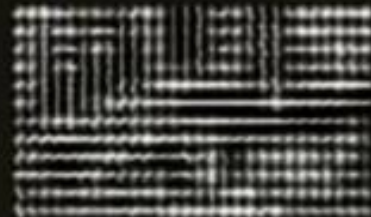
Nearest Neighbors

What information does HOG have?

Image

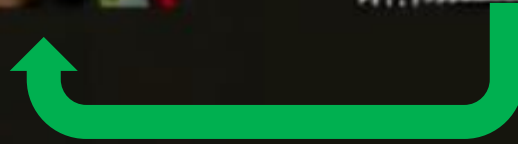
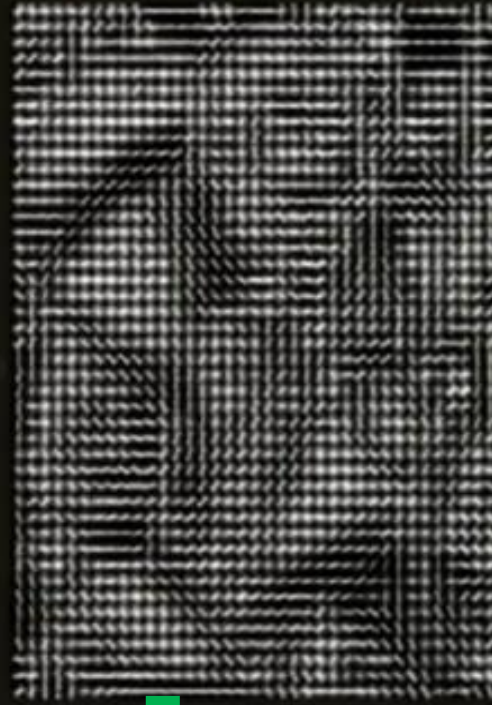


HOG



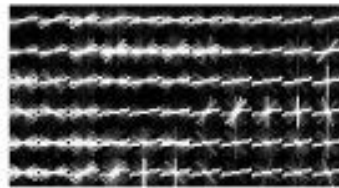
Nearest Neighbors

What information is lost?

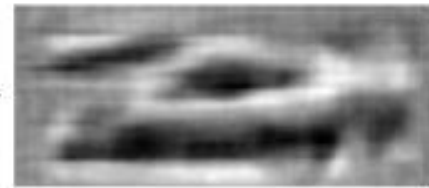




Car Detection



HOG Features

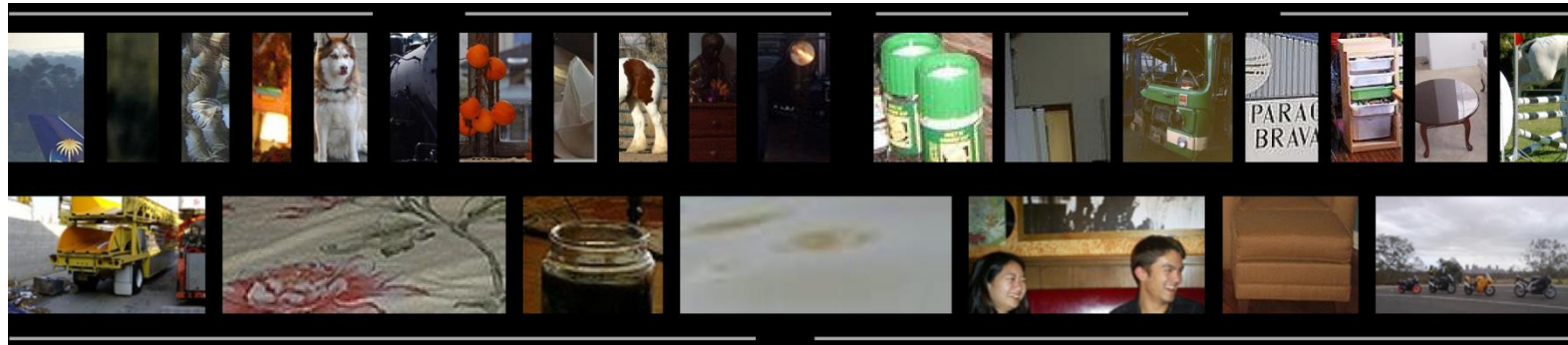
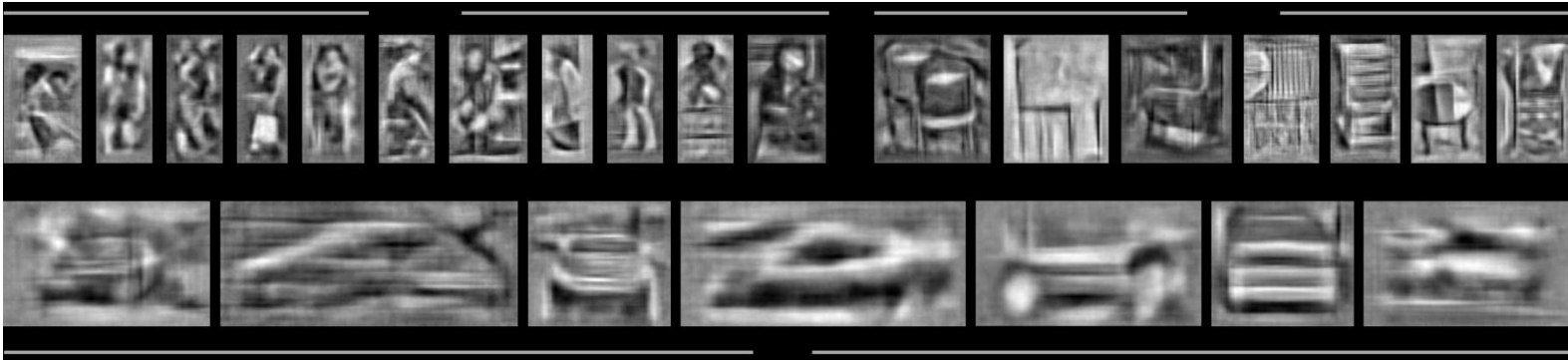


Our Visualization

Person, chair, and car

Can you guess which are **false** alarms?

High scoring detections from the deformable parts model (DPM) for person, chair, and car.



ALL ARE FALSE ALARMS: Consequently, even with a better learning algorithm or more data, these false alarms will likely persist. In other words, **the features are to blame.**

Inverting HOG descriptor

Let $x \in \mathbb{R}^D$ be an image

$y = \phi(x)$ be the corresponding HOG feature descriptor.

$$\phi^{-1}(y) = ?$$


HOG is highly sensitive to noise and the equation has frequent local minima ☹

The paper presents 4 algorithms to invert HOG

- 3 base lines
- Paired dictionary learning

Paired dictionary learning

first K eigenvectors of $\Sigma_{XX} \in \mathbb{R}^{D \times D}$

Let $x \in \mathbb{R}^D$ be an image and $y \in \mathbb{R}^d$ be its HOG descriptor. 

Suppose we write x and y in terms of bases $U \in \mathbb{R}^{D \times K}$ and $V \in \mathbb{R}^{d \times K}$ respectively, but with shared coefficients $\alpha \in \mathbb{R}^K$:

$$x = U\alpha \quad \text{and} \quad y = V\alpha \quad (5)$$

*Paired dictionaries require finding appropriate bases U and V such that above equation holds. We solve a paired dictionary learning problem, inspired by recent super resolution **sparse** coding work.*

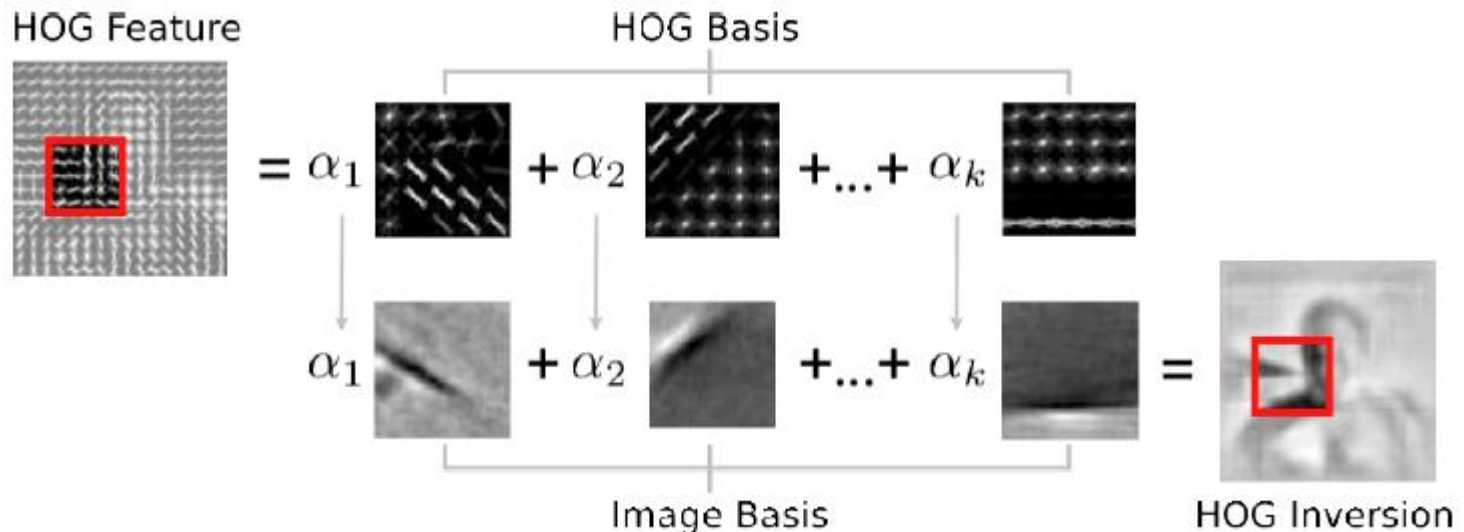
$$\begin{aligned} \operatorname{argmin}_{U, V, \alpha} \sum_{i=1}^N (\|x_i - U\alpha_i\|_2^2 + \|\phi(x_i) - V\alpha_i\|_2^2) \\ \text{s.t.} \quad \|\alpha_i\|_1 \leq \lambda \quad \forall i, \quad \|U\|_2^2 \leq \gamma_1, \quad \|V\|_2^2 \leq \gamma_2 \end{aligned} \quad (7)$$

Paired dictionary learning

The key observation is that inversion can be obtained by first projecting the HOG features y onto the HOG basis V , then projecting α into the natural image basis U :

$$\phi_D^{-1}(y) = U\alpha^*$$

$$\text{where } \alpha^* = \underset{\alpha \in \mathbb{R}^K}{\operatorname{argmin}} \|V\alpha - y\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_1 \leq \lambda \quad (6)$$



Paired dictionary learning

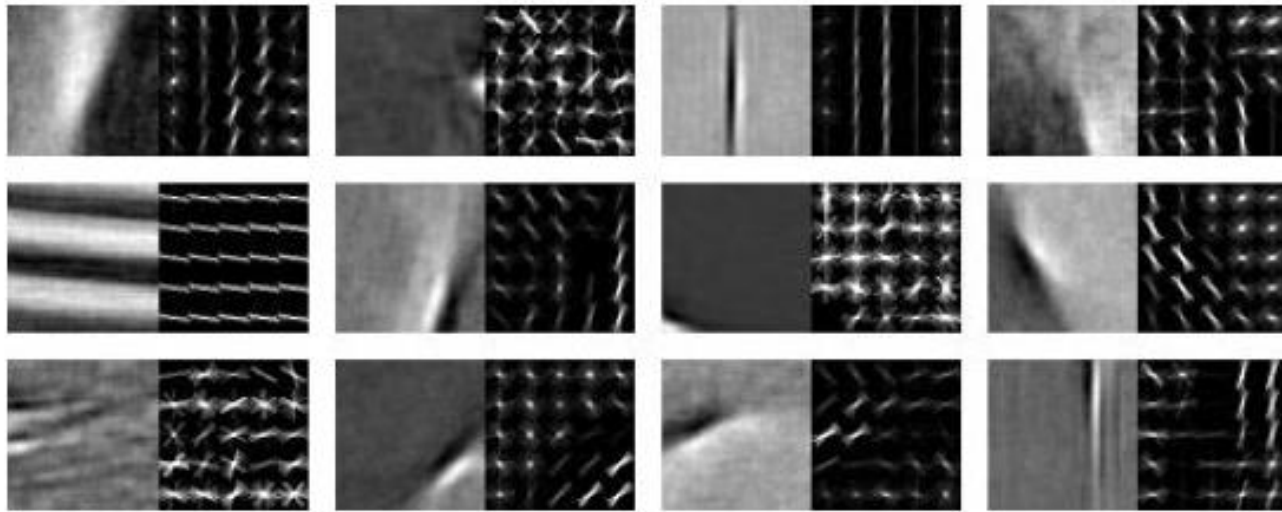
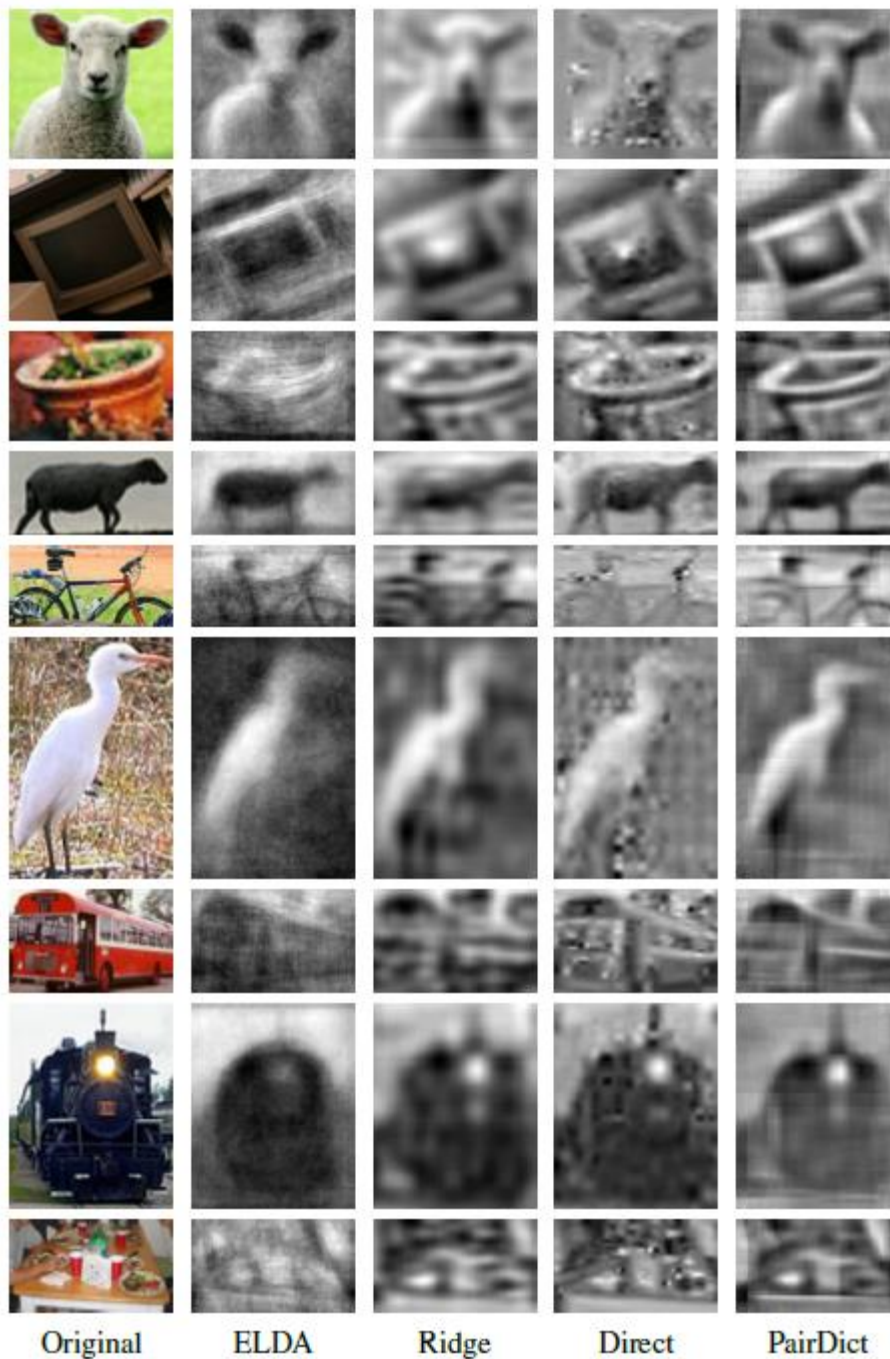


Figure 7: Some pairs of dictionaries for U and V . The left of every pair is the gray scale dictionary element and the right is the positive components elements in the HOG dictionary. Notice the correlation between dictionaries.



Original

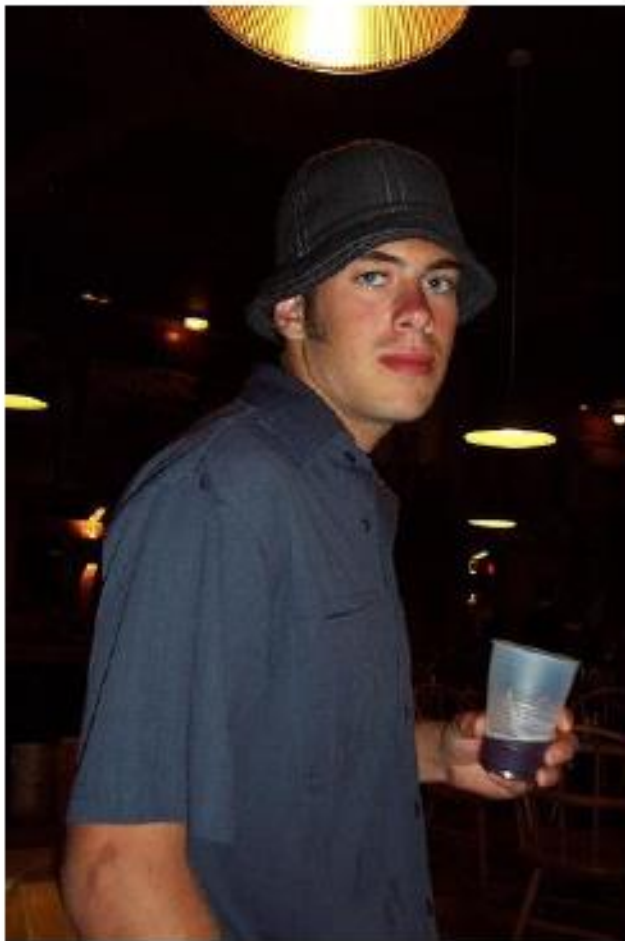
ELDA

Ridge

Direct

PairDICT

What object detectors see?



(a) Human Vision



(b) HOG Vision

Estimating color images



Limitations

- Not optimal



PairDict (seconds)



Greedy (days)



Original

- Template size dependency



40 × 40



20 × 20

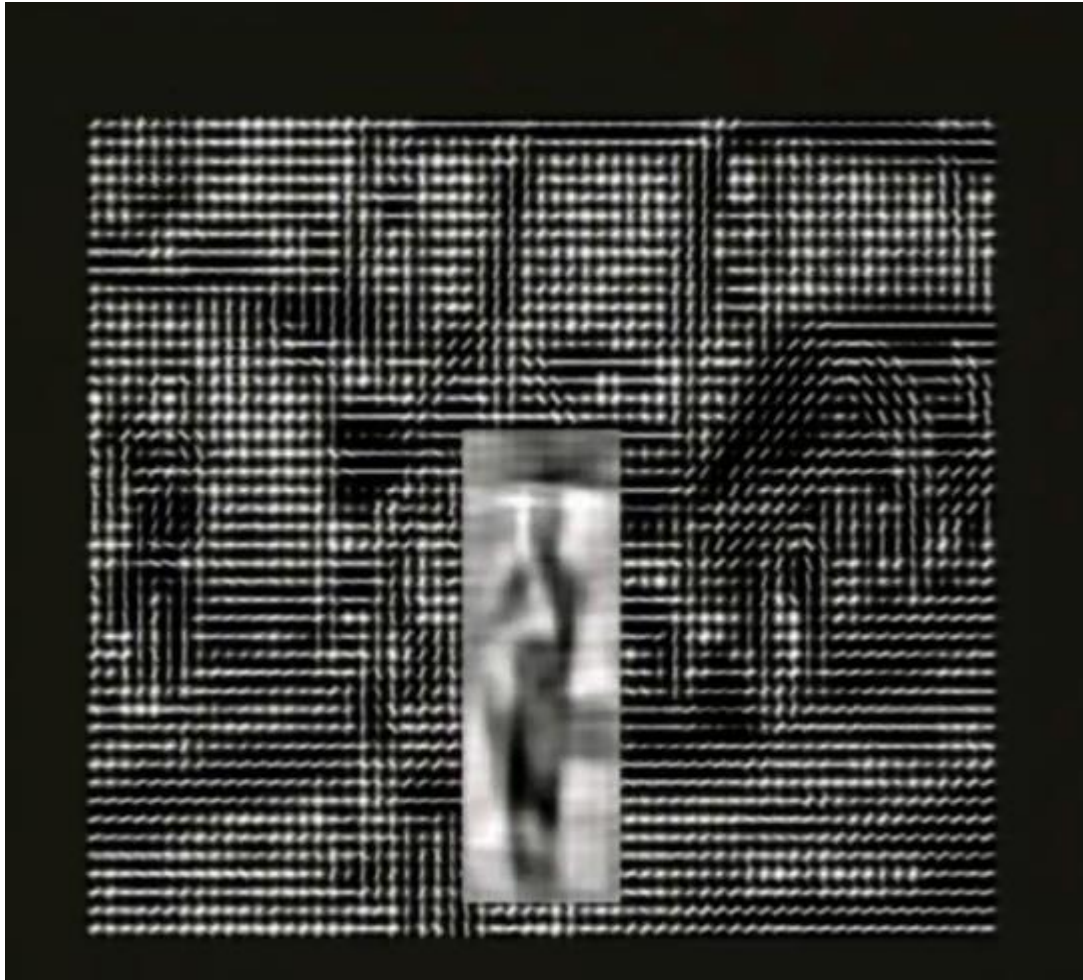


10 × 10

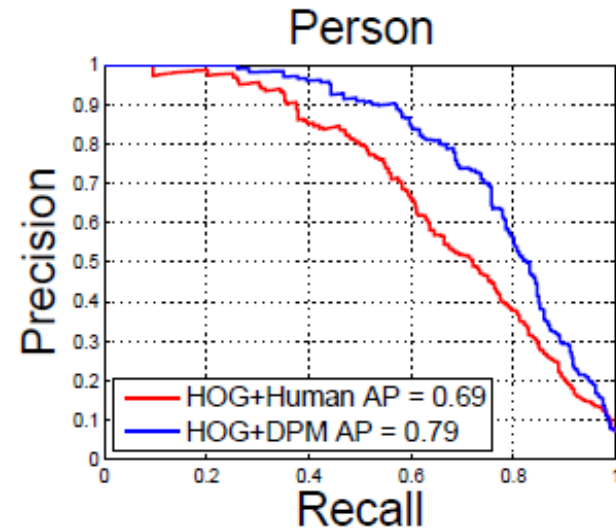
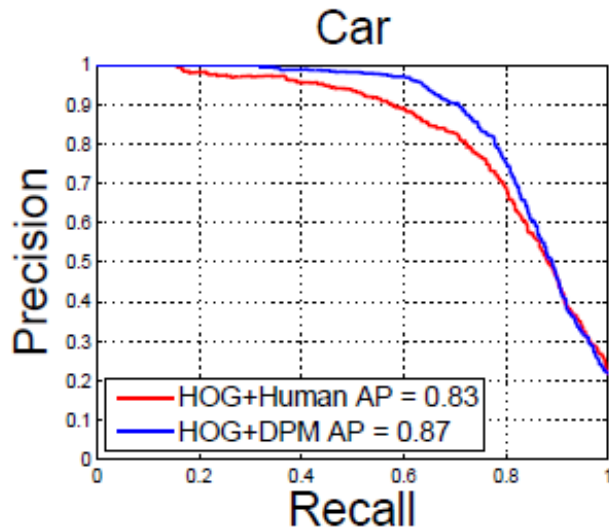
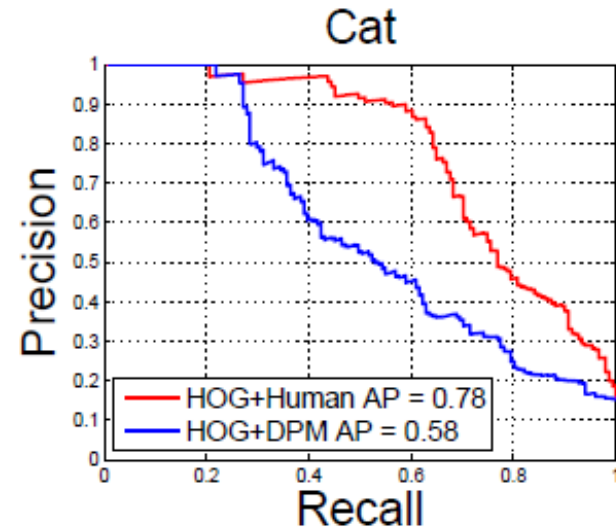
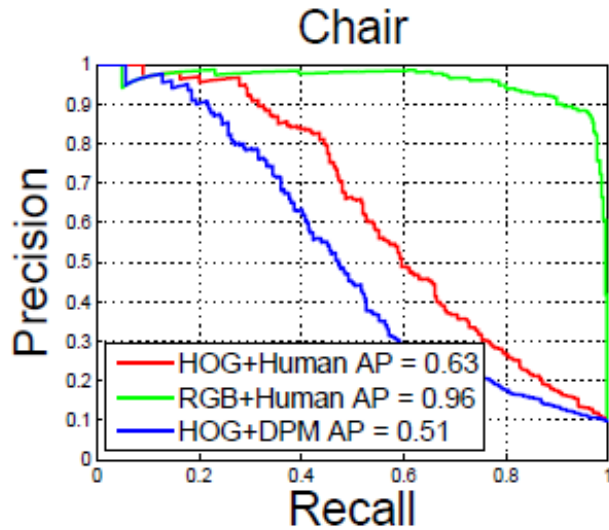


5 × 5

HOG+Human vs HOG+DPM



HOG+Human vs HOG+DPM



Message to Go

- tool for visualizing object detection features
- choice of feature matters
- DPM is close to the performance limit of HOG

THE END



Thank you for your attention