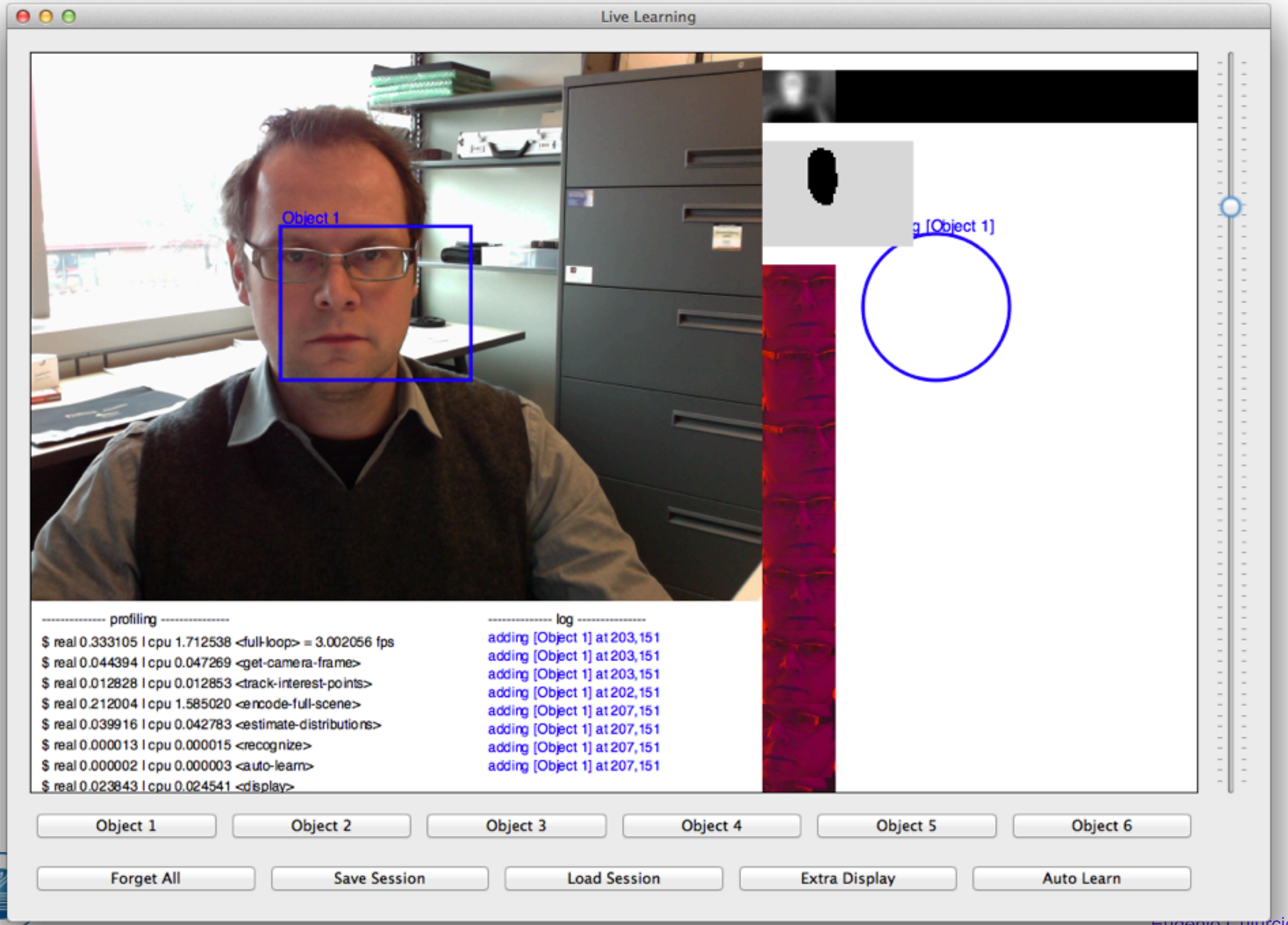
Artificial and robotic vision

Eugenio Culurciello

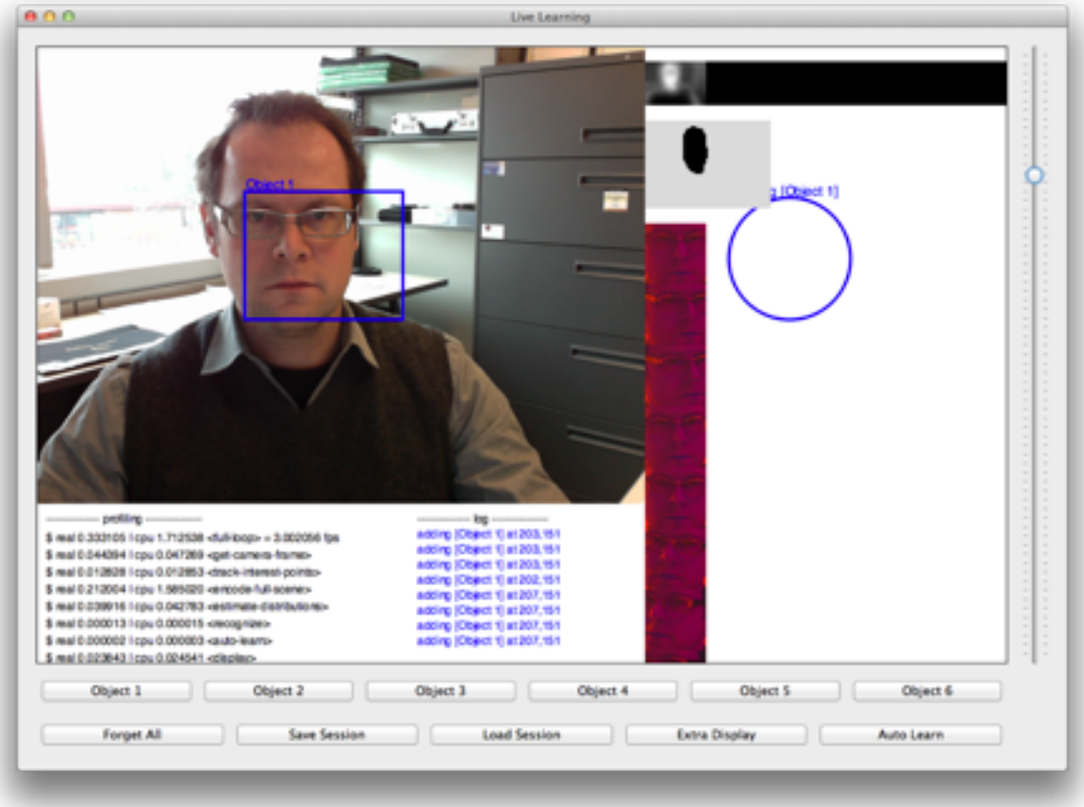
Lecture 10: online learning

e-Lab

online learner



online learner



- one shot learning:
  learn subsequent new views
  enough to train a classifier?
  need to be "general purpose"

- multiple targets option
- target present or not? threshold
- of course: find target in clutter!

- a great application of vision systems!

e-Lab

online learner

https://github.com/e-lab/online-learner

originally developed by
Yann LeCun, Clement Farabet

online learner components

1 - Convnet/CNN

2 - Tracker

- run convnet on selected area >> features
- run tracker
- run convnet, if result is as tracker, ok
  otherwise use result as next sample

this works fairly well...

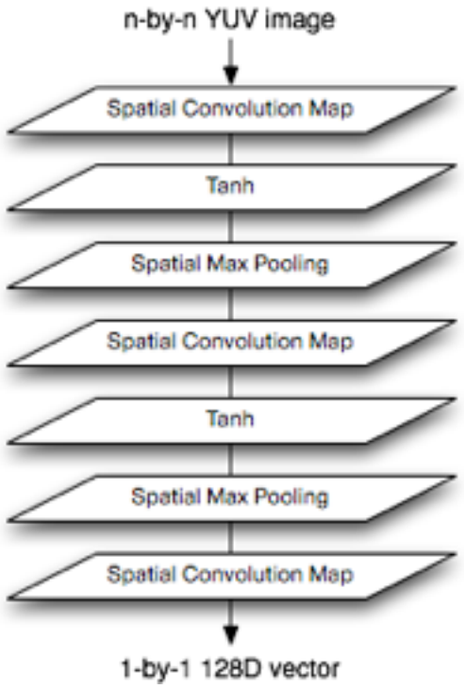
e-Lab

online learner algorithm

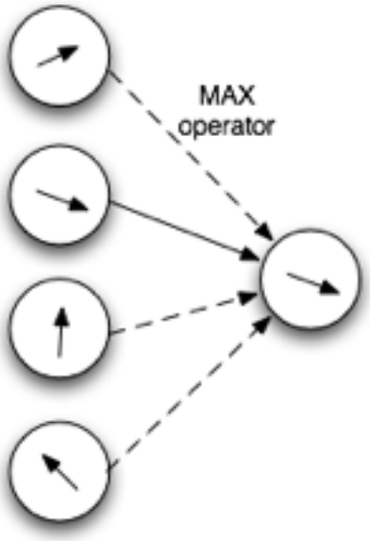
- **Tracker**
  - *GoodFeaturesToTrack* from OpenCV

- **Detector** (if tracker fails)
  - *Convolutional Neural network (CNN)* as a classifier pre-trained with *Barcelona dataset*

- Vector matching using SAD $\Sigma||x - d_k||$
- Choose max. prob among positive prototypes
- *Auto-learning* catches new prototype if $p(x) < 0.80$



The structure of ConvNet



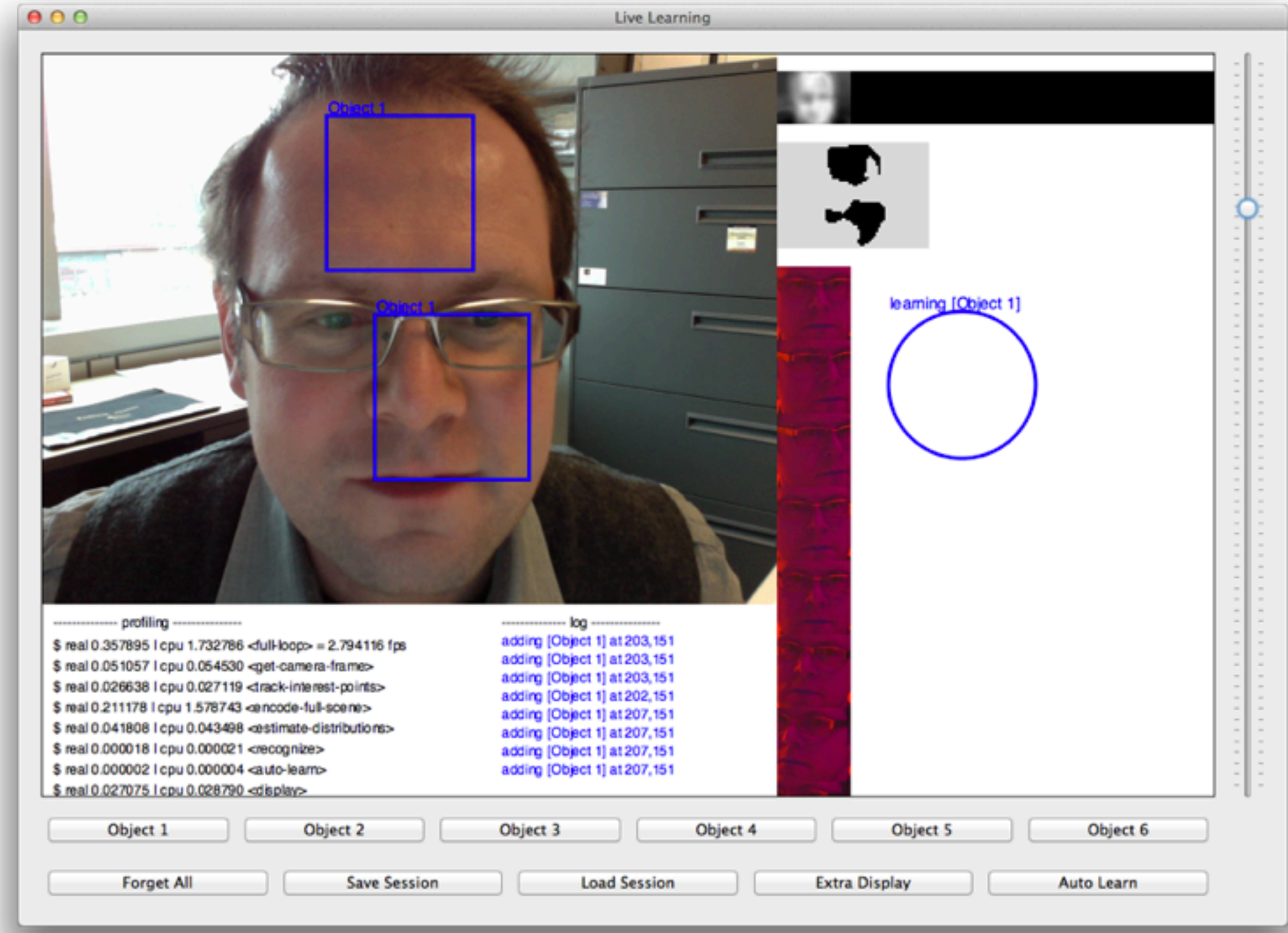
Choose maximum probability

* http://yann.lecun.com/exdb/publis/pdf/farabet-pami-13.pdf

online learner

issues:
no multiscale

but CNN* used is rather good with size invariance and also with rotations!



e-Lab

* http://yann.lecun.com/exdb/publis/pdf/farabet-pami-13.pdf

online learner algorithm

```
AutoLearn:

if features of view of object x < threshold
then
    learn new features from tracked view of object
    add features to template list
```

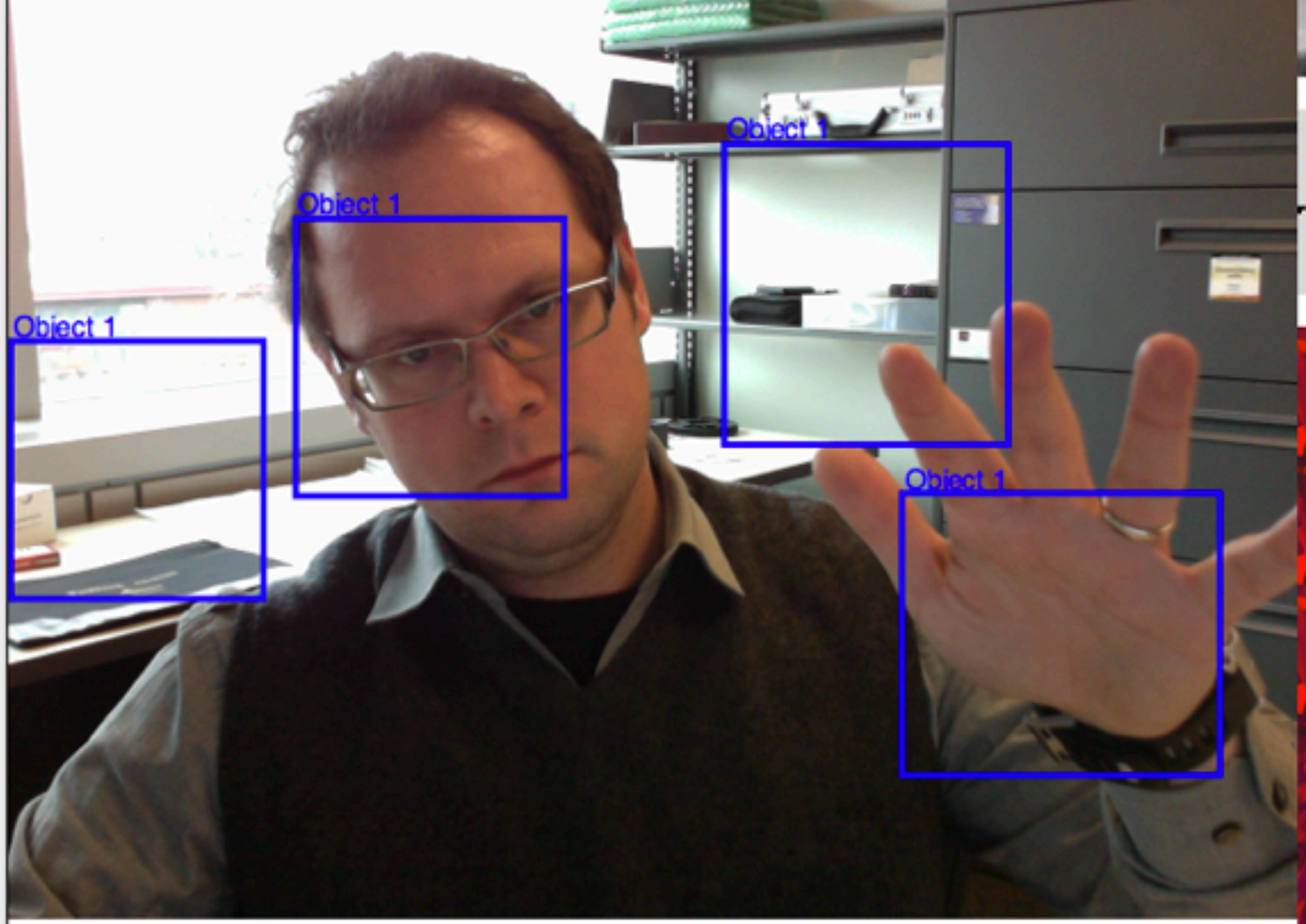
this causes drift of features. Why?

- some features of background are learned
- these features might be “strong” and drive recognition
- they will then become “the object”~~~!!!

e-Lab

online lear

issues:
drifting of features
when:
AutoLearn selected



---------- profiling ----------
$ real 0.422462 I cpu 1.881988 <full-loop> = 2.367077 fps
$ real 0.044548 I cpu 0.046548 <get-camera-frame>
$ real 0.056001 I cpu 0.060200 <track-interest-points>
$ real 0.231760 I cpu 1.677826 <encode-full-scene>
$ real 0.060451 I cpu 0.064645 <estimate-distributions>
$ real 0.000053 I cpu 0.000055 <recognize>
$ real 0.000002 I cpu 0.000004 <auto-learn>
$ real 0.029557 I cpu 0.032663 <display>
$ real 0.010725 I cpu 0.046328 <learn-new-view>

---------- log ----------
auto-learning off...
auto-learning [Object 1]
auto-learning [Object 1]
auto-learning [Object 1]
auto-learning [Object 1]
auto-learning is on !
adding [Object 1] at 217,152
adding [Object 1] at 217,152

e-Lab

online learner

issues:
drifting of features
when: AutoLearn
selected

features drifted as
some background
is learned

e-Lab

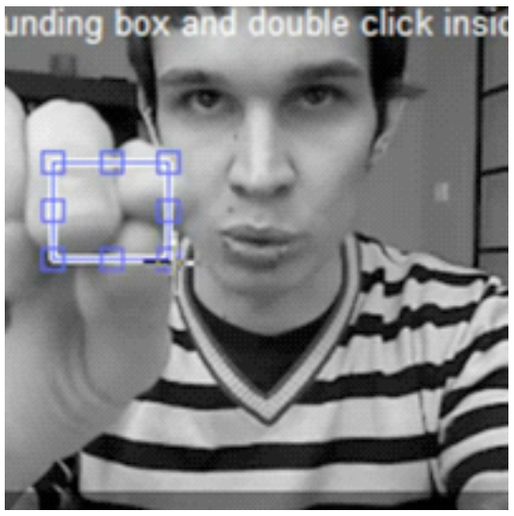
online learner: tester

TLD dataset by

Dr. Zdenek Kalal

TLDVISION

Learning cameras.



http://info.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html

Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *Pattern Analysis and Machine Intelligence*, 2011.

e-Lab

possible solutions

1- segment object
2- remove tracker: use CNN to track object!
3- add RBF layer
4- improve CNN/convnet/classifier

possible solutions

1- segment object

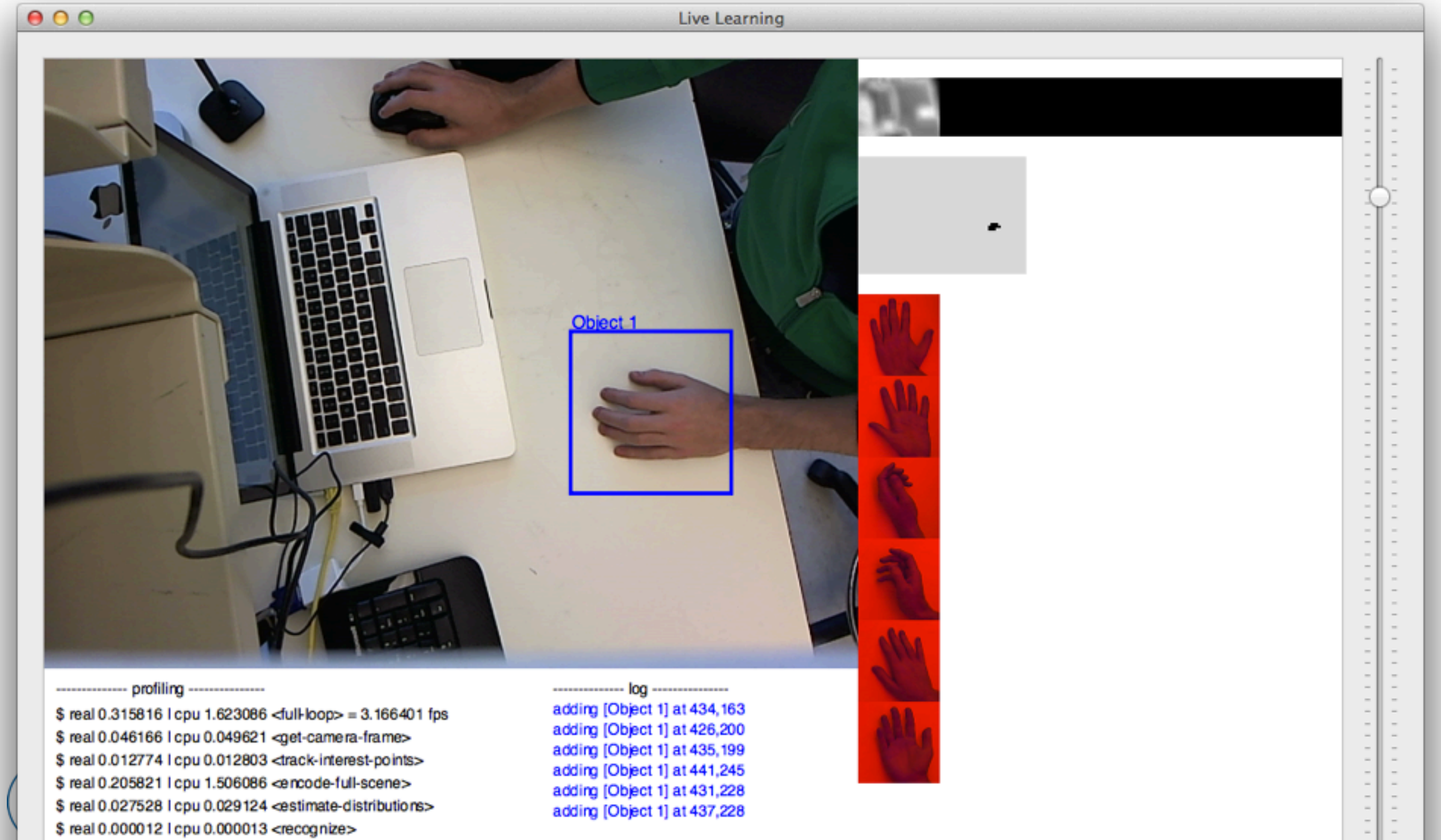
2- remove tracker: use CNN to track object!

3- add RBF layer

4- improve CNN/convnet/classifier

I- segment object

- create a ‘segmented’ hand against lab table



I - segment object

- segmentation must be done at match, not only at prototyping

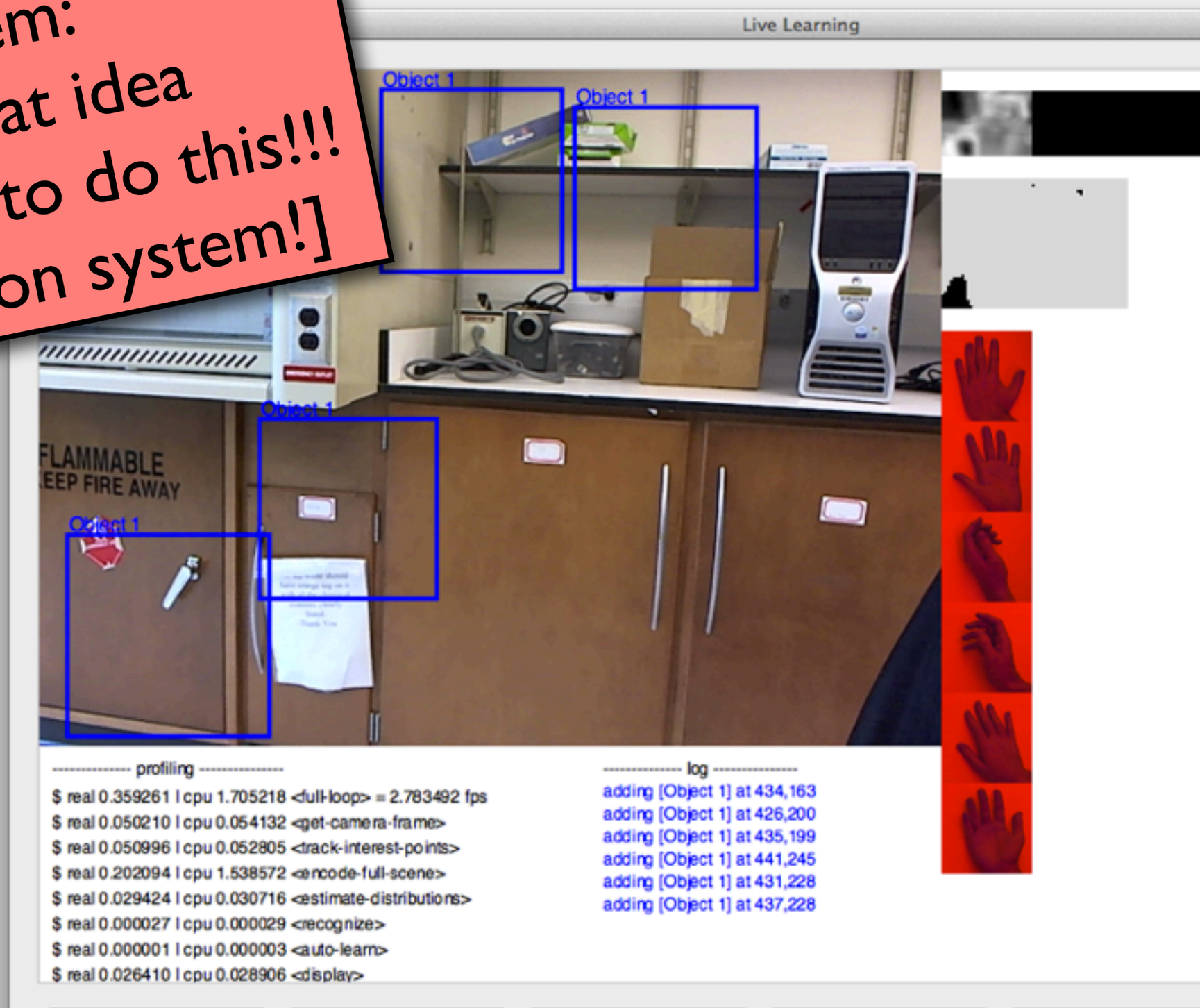
false negative

e-Lab

I - segment object

false positives at
default threshold

problem:
not a great idea
classifier has to do this!!!
[unified vision system!]



e-Lab

possible solutions

1- segment object

2- remove tracker: use CNN to track object!

3- add RBF layer

4- improve CNN/convnet/classifier

e-Lab

2- remove tracker: use CNN to track object!

- Convolutional Neural Network (CNN) tracker without patch update (use only first patch given by user) is fairly good for most of dataset except for the David dataset (very dark at first frame).
- (*potential for development) For the panda dataset (lots of noises), CNN tracker without patch update is the best way to track the object for a long time.
- With patch update, the tracker causes the problem in patch updating (shifting problem)

Example of shifting patch problem

2- remove tracker: use CNN to track object!

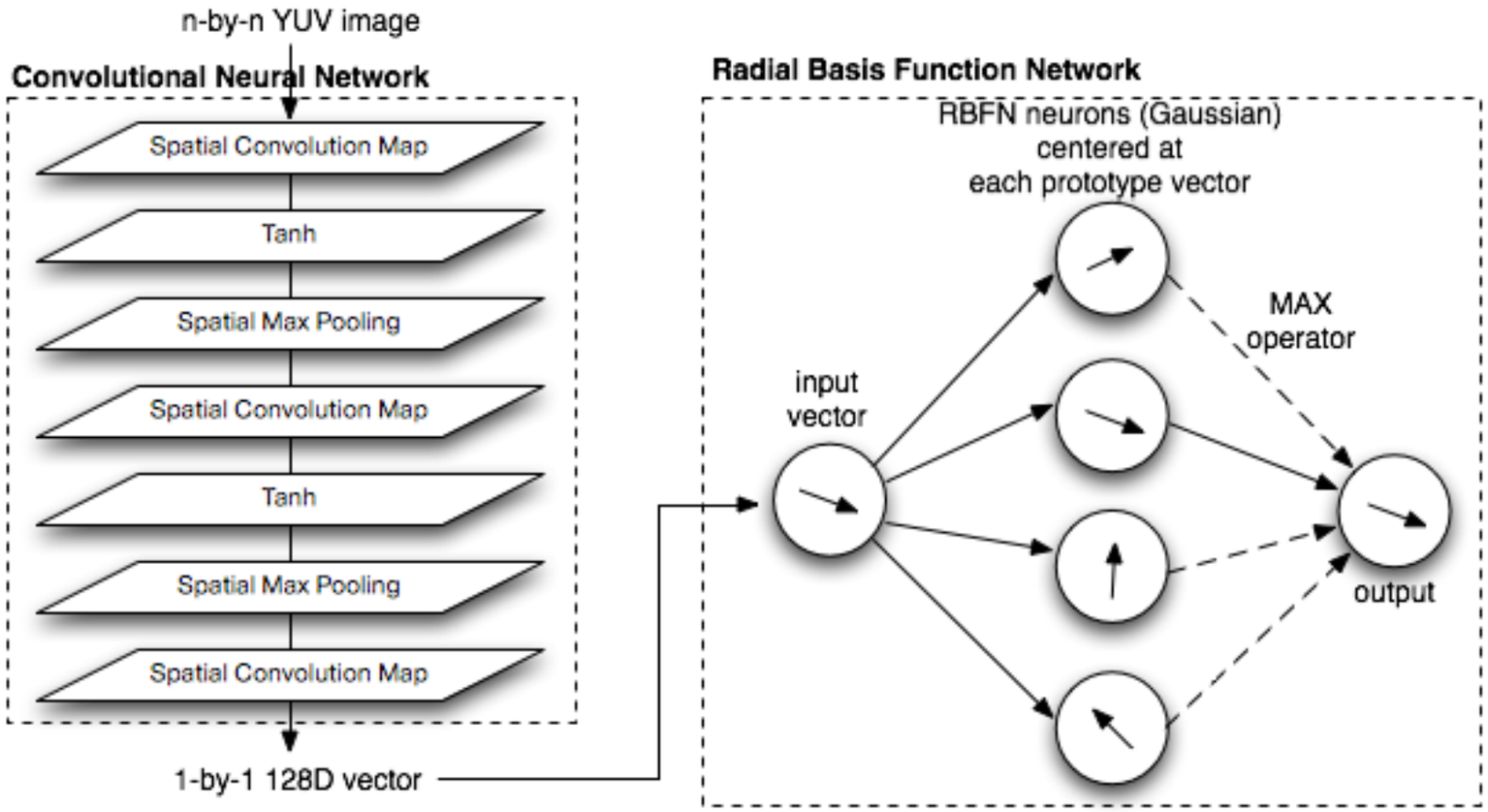
problem:
CNN classifier not good enough to track

needs improvements
how?

possible solutions

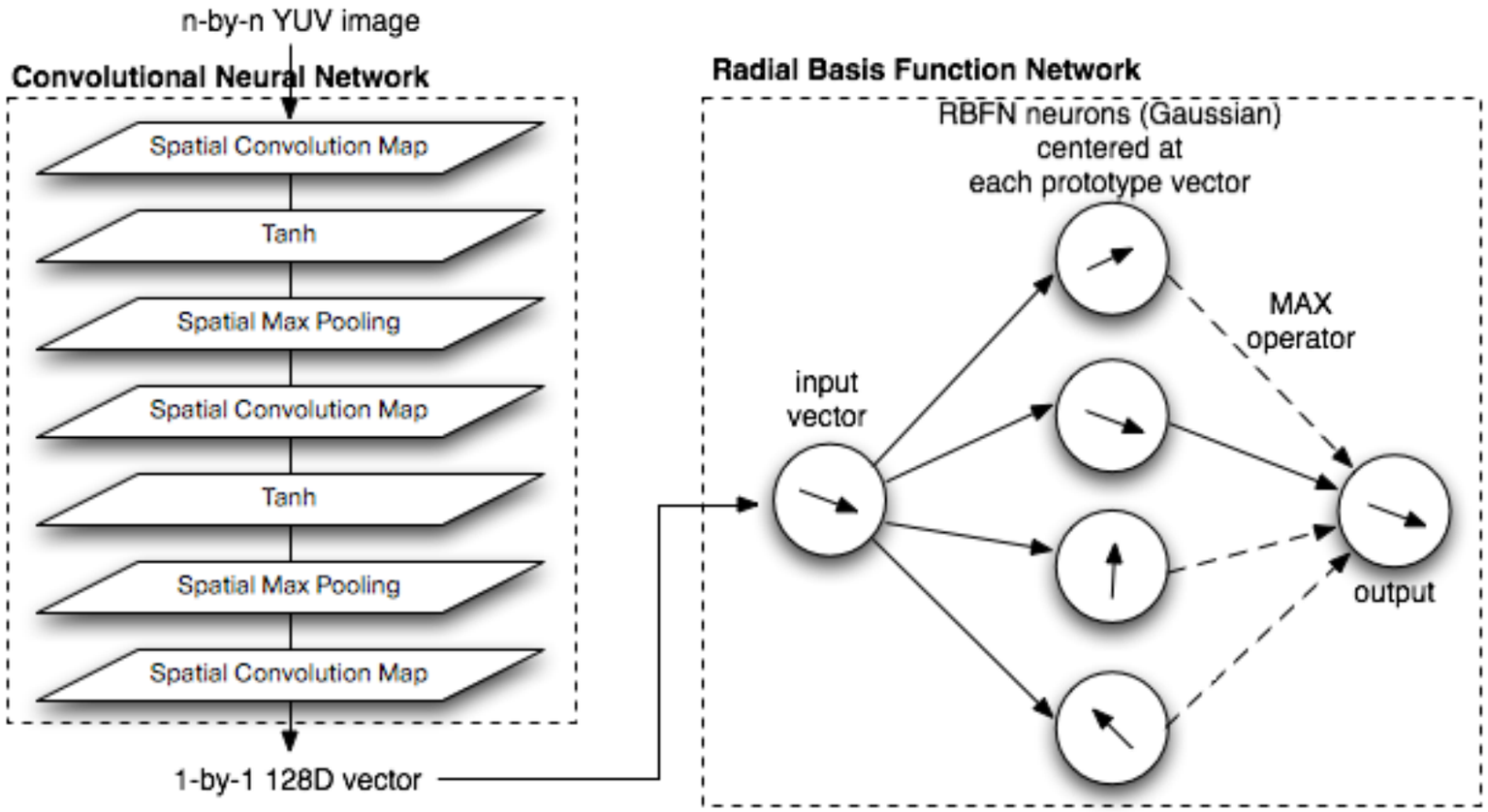
1- segment object
2- remove tracker: use CNN to track object!
3- add RBF layer
4- improve CNN/convnet/classifier

3- add RBF layer



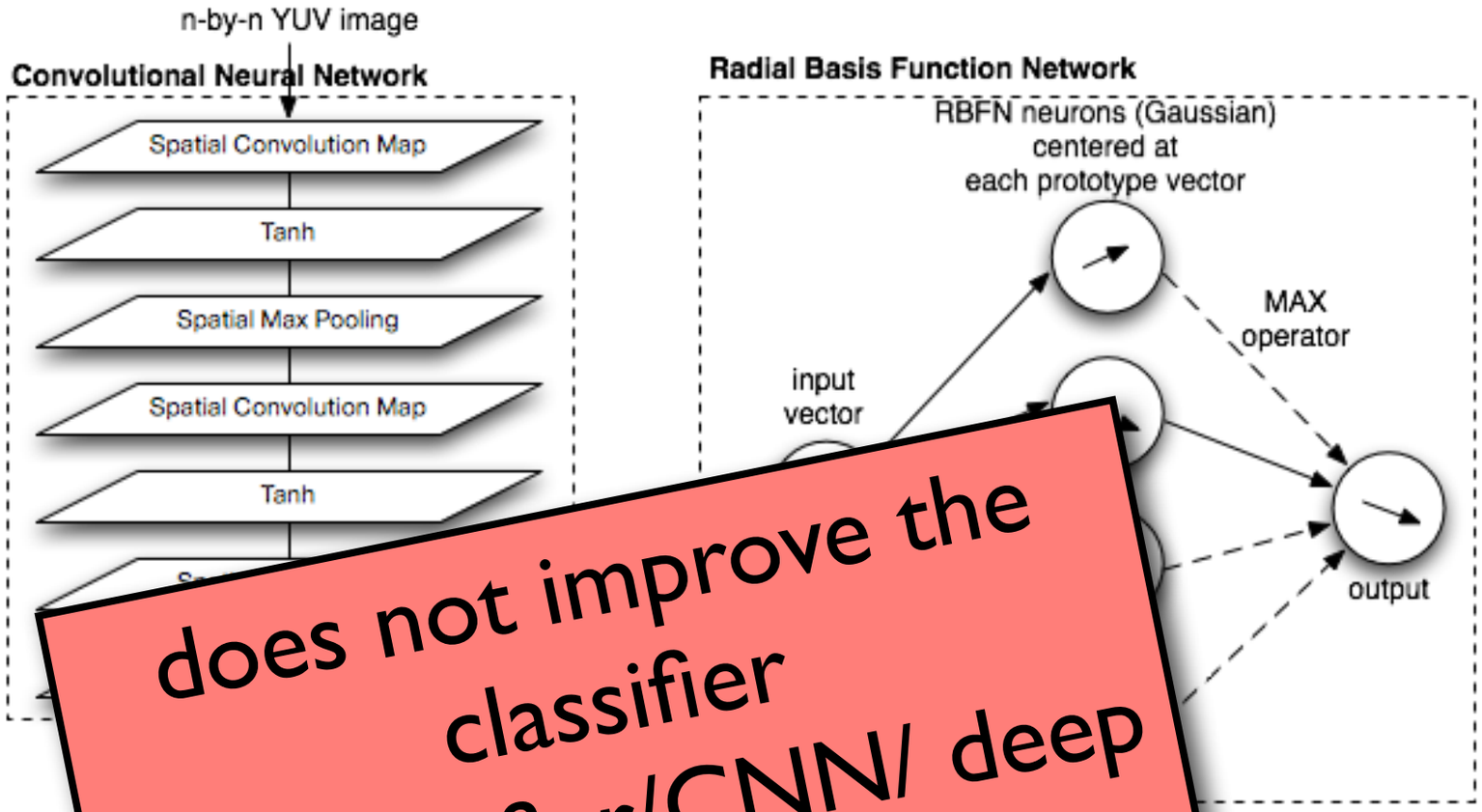
- *Radial Basis Function Network (RBFN)* is added to the *Convolutional Neural Network (CNN)*.
- CNN in real-time is not possible with few samples. Train RBFN layer instead.
- The number of neurons increases or decreases by growing or pruning process.
- Growing / Pruning criteria should be determined before use.

3- add RBF layer



- parameters: spread of gaussian
- use only positive prototypes (inspired by our vision)
- adds one centroid for each object view: growing computation
- prune prototypes: if not seen for a while remove

3- add RBF layer



does not improve the classifier

the classifier/CNN/ deep net is the LIMITING FACTOR!

possible solutions

1- segment object
2- remove tracker: use CNN to track object!
3- add RBF layer
4- improve CNN/convnet/classifier

4- improve CNN/convnet/classifier

we still do not know how to do this
TOUGH PROBLEM!

Conclusions:

- online learning: a VERY interesting problem
- complicated by one-shot learning
- need a classifier that can segment, track
- but also is fast for real-time application

- A GREAT PROBLEM for DEEP NETS!!!!

e-Lab