Automatic design of cascaded classifiers
or: A fast way to make a boosted classifier faster

1

A cascade is useful when:
1: Most input belong to "negative" class.
2: Many can be discarded easily.
Then, a small computation time and good recognition rate can be reached.

2 **The Problem**

Choices when building a cascade:
1: Weak classifier familly.
2: Number of cascade levels.
3: Size/performance of each leve
4: Etc ...

3

We propose:
Specify just 1) weak classifier familly,
2) expected ratio of positives and
3) relative cost of type I & II errors.
All other parameters will be set automatically.

4

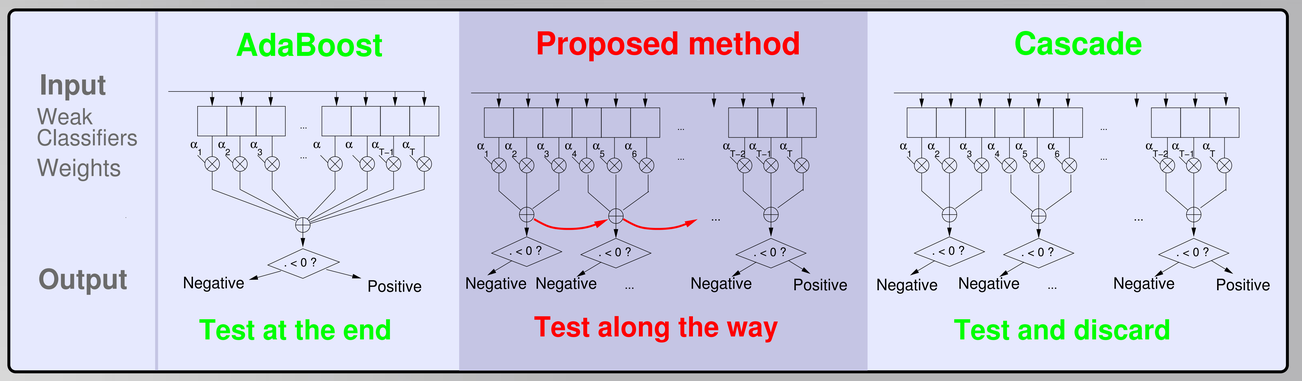
Method:
Build boosted classifier with desired performance.
Slice it into a near–optimal cascade.

5

Advantages of slicing a boosted classifier:
1: A single classifier is trained: a single
2: No design choices concerning size or performance of intermediate levels.

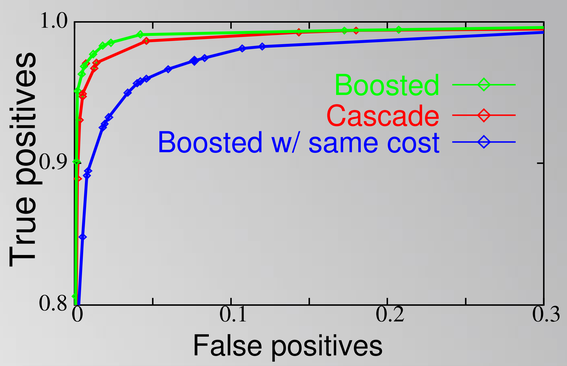
Idea: schedule tests along the computation of weak classifiers, so that comp. cost is minimized, while preserving output on a given dataset.

Computation models of Adaboost, the proposed method and cascade.

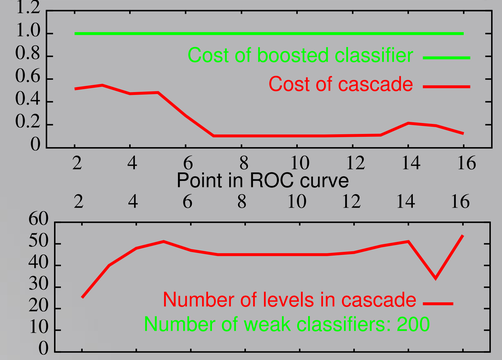


Experiments: face detection w/ "Haar" linear classifiers.

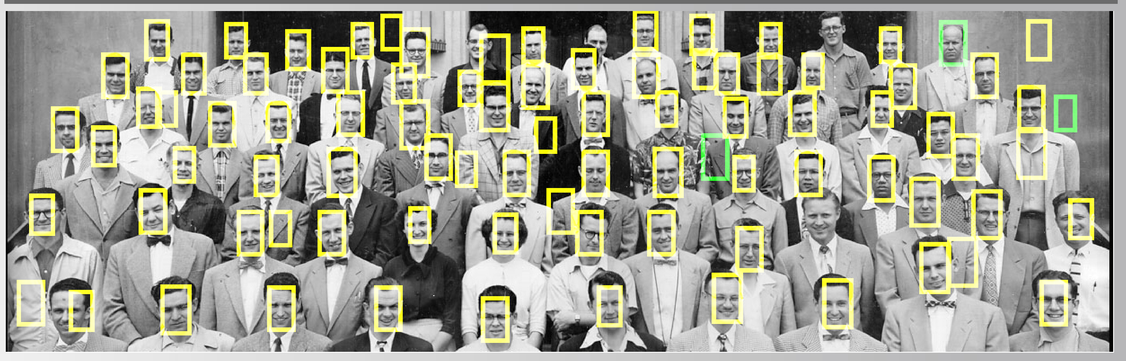
Cascade has nearly same ROC



Gain of time of cascade



Examples of detected faces. Yellow: Cascade and boosted. Green: Boosted only.



UK Center for Visualization & Virtual Environments

Etienne Grossmann etienne@cs.uky.edu

This work has been supported by the Kentucky Office of New Economy