Fast Object Detection and Localization

Oliver Williams

omcw2@cam.ac.uk
This lecture

• This lecture is about

  **Object detection and localization**
  task: find a pre-defined object in an image

• This lecture is **not** about

  **Object recognition**
  task: identify an object as belonging to one of $n$ classes
Localization or Tracking?

[Romdhani et al. ICCV 2001]
[Viola and Jones CVPR 2001]
Localization and Tracking
Outline

• Object detection
  – Discriminative classifiers
  – Boosting
  – Fast feature detectors
  – Cascade of classifiers

• Object tracking
  – Tracking with a discriminative classifier
  – Learning to track: the Displacement expert

• Detection + tracking
  – The best of both worlds?
Discriminative methods

Object detection and recognition is formulated as a classification problem. The image is partitioned into a set of overlapping windows … and a decision made at each window about whether it contains the target object.

Where are the screens?

Bag of image patches

In some feature space

Decison boundary

Background

Computer screen
Discriminative vs. generative

- Generative model
  *(The artist)*

- Discriminative model
  *(The lousy painter)*

- Classification function

\[
label = F_{Zebra}(Data)
\]
Formulation

- Formulation: binary classification

Features \( x = x_1, x_2, x_3, \ldots, x_N \)

Labels \( y = -1, +1, -1, -1, \ldots \)

Training data: each image patch is labeled as containing the object or background

Test data

Classification function

\[ \hat{y} = F(x) \quad \text{Where} \quad F(x) \quad \text{belongs to some family of functions} \]

Minimize misclassification error

(Not that simple: we need some guarantees that there will be generalization)
Discriminative methods

**Nearest neighbor**

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005

10^6 examples

**Support Vector Machines and Kernels**

Guyon, Vapnik
Heisele, Serre, Poggio, 2001

**Neural networks**

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998

**Conditional Random Fields**

McCallum, Freitag, Pereira 2000
Kumar, Hebert 2003
Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998

- Provides efficient algorithm for sparse visual feature selection
  - Tieu & Viola, 2000
  - Viola & Jones, 2003

- Easy to implement
Boosting

- Defines a classifier using an additive model:

\[ F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots \]
Defining a classifier using an additive model:

\[ F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots \]

- We need to define a family of weak classifiers

\[ f_k(x) \] from a family of weak classifiers
Boosting

- Sequential procedure:

Each data point has a class label:
\[ y_t = \begin{cases} 
+1 (& \circ) \\
-1 (& \bigcirc) 
\end{cases} \]

and a weight:
\[ w_t = 1 \]
Toy example

Weak learners from the family of lines

Each data point has
a class label:
$y_t = \begin{cases} 
+1 & (\circ) \\
-1 & (\bigcirc) 
\end{cases}$

and a weight:
$w_t = 1$

$h \Rightarrow \text{p(error)} = 0.5$ it is at chance
Toy example

Each data point has a class label:
\[ y_t = \begin{cases} 
+1 & (\bullet) \\
-1 & (\bigcirc) 
\end{cases} \]

and a weight:
\[ w_t = 1 \]

This one seems to be the best

This is a ‘**weak classifier**’: It performs slightly better than chance.
We set a new problem for which the previous weak classifier performs at chance again.
Toy example

We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[ y_t = \begin{cases} +1 & (\circ) \\ -1 & (\bigcirc) \end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp\{-y_t H_t\} \]
Toy example

We set a new problem for which the previous weak classifier performs at chance again:

Each data point has a class label:

\[ y_t = \begin{cases} 
+1 \ (\circ) \\
-1 \ (\bigcirc) 
\end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp \{-y_t H_t\} \]
Toy example

We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[ y_t = \begin{cases} +1 & (\circ) \\ -1 & (\circ) \end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp\{-y_t H_t\} \]
The strong (non-linear) classifier is built as the combination of all the weak (linear) classifiers.
From images to features: Weak detectors

We will now define a family of visual features that can be used as weak classifiers ("weak detectors")

\[ h_i(I, x, y) \]

Takes image as input and the output is binary response. The output is a weak detector.
Weak detectors

Haar filters and integral image
Viola and Jones, ICCV 2001

\[ J(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \]

The average intensity in the block is computed with four
sums independently of the block size.
We want the complexity of the 3 features classifier with the performance of the 100 features classifier:

We increase precision using the cascade
Cascade of classifiers
Fleuret and Geman 2001, Viola and Jones 2001
8192 search windows
Cascade of classifiers

Fleuret and Geman 2001, Viola and Jones 2001

188 search windows
Cascade of classifiers
Fleuret and Geman 2001, Viola and Jones 2001
48 search windows
Cascade of classifiers
Fleuret and Geman 2001, Viola and Jones 2001
“Head in the coffee beans problem”

Can you find the head in this image?
Outline

• Object detection
  – Discriminative classifiers
  – Boosting
  – Fast feature detectors
  – Cascade of classifiers

• Object tracking
  – Tracking with a discriminative classifier
  – Learning to track: the Displacement expert

• Detection + tracking
  – The best of both worlds?
Support Vector Tracking [Avidan 2001]

Use discriminative classifier to create a tracker

\[ f_{svm}(I) = f_{svm}(I(u_t)) \approx f_{svm}(I_0 + \nabla I \delta u) \]

\[ \delta u = \arg \max_{\delta u} f_{svm}(I_0 + \nabla I \delta u) \]
The Displacement Expert

- **Advantage:** No model; no optimization
- **Drawback:** Must learn from training data
Creating a Training Set

- Select a few “seed” stills
- Simulate translation, scaling and rotation
  - → labelled training set
Displacement expert

Frame $t$

$u_t$

Frame $t+1$

$u_{t+1}$

$I_t$

$I_{t+1}$

$I_{t+1}^*$
Displacement Expert

- At each frame...

\[ I^*_t+1 \]

Extract subimage

\[ u_{t+1} \]

New state estimate

\[ I_{t+1} \]
Face tracking
Car tracking

- Algorithm not specific to any class of objects
“Hand mouse”
Outline

• Object detection
  – Discriminative classifiers
  – Boosting
  – Fast feature detectors
  – Cascade of classifiers

• Object tracking
  – Tracking with a discriminative classifier
  – Learning to track: the Displacement expert

• Detection + tracking
  – The best of both worlds?
Initialization & Recovery

- Algorithm trains from “seed” subimages
  - Provided by localisation algorithm

Localisation algorithm also used for
  - Initialization (frame 0)
  - Validation
  - Re-initialization
Results: Computational Efficiency
Automatic cameraman