

Face Detection

Image Processing — scc0251

`www.icmc.usp.br/~moacir` — `moacir@icmc.usp.br`

ICMC/USP — São Carlos, SP, Brazil

2011

Agenda

- 1 Detectando faces
- 2 Detector de faces Viola-Jones
 - Imagem Integral
 - Atributos retangulares
 - Boosting
 - Viola-Jones Boosting e Classificação em Cascata

Agenda

1 Detectando faces

2 Detector de faces Viola-Jones

- Imagem Integral
- Atributos retangulares
- Boosting
- Viola-Jones Boosting e Classificação em Cascata

Porque detectar faces?



Análise do problema

- Definição de limites: apenas faces frontais, perfis, de cabeça para baixo?
 - Na maioria dos casos apenas faces frontais ou pequena mudança de pose são tratadas.
 - A entrada é uma imagem em níveis de cinza
 - Tarefa fácil?



Humanos: máquinas de reconhecer faces



Detector de faces Viola-Jones

- Método:
 - Percorre a imagem usando uma janela.
 - Realiza a detecção utilizando detectores em várias escalas.
 - Utiliza imagem integral e extrai atributos retangulares de forma rápida.
 - Classifica utilizando um algoritmo de *boosting*.

Imagem Integral

- Imagem Integral: cada pixel é igual a soma de todos os pixels acima e à esquerda.

1	1	1
1	1	1
1	1	1

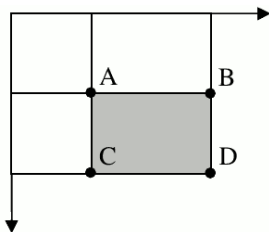
Input image

1	2	3
2	4	6
3	6	9

Integral image

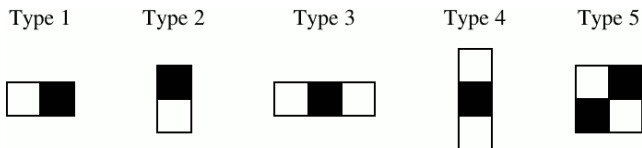
Imagem Integral

- Essa abordagem permite calcular a soma dos pixels dentro de um retângulo usando apenas quatro valores: A , B , C , D .
- A , B , C , D são os pixels na imagem integral relativos aos cantos do retângulo na imagem de entrada.
- Soma = $D - (B + C) + A$.
- A precisou ser somado pois tanto B quanto C incluem A .



Atributos retangulares

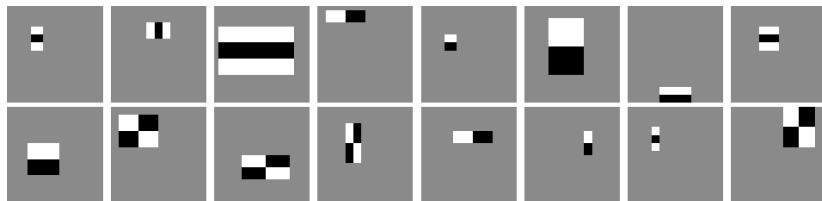
- A soma dos pixels dentro de retângulos de tamanho arbitrário pode ser calculada em tempo constante.
- O detector de faces Viola-Jones analisa uma dada janela usando atributos consistindo de dois ou mais retângulos, de diversos tipos:



- Cada atributo resulta num valor numérico, calculado subtraindo a soma do(s) retângulo(s) branco(s) da soma dos retângulo(s) preto(s).

Atributos retangulares

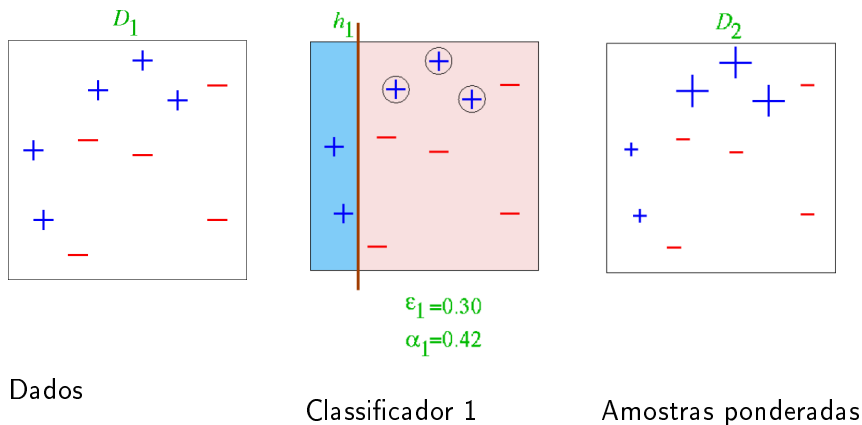
- Em cada janela de uma imagem (tamanho base é de 24×24 pixels), cada um dos atributos é calculado em diferentes escalas e deslocamentos.
- Aproximadamente 160 mil atributos são possíveis considerando todas as combinações possíveis de deslocamento e escala.



Treinamento usando boosting

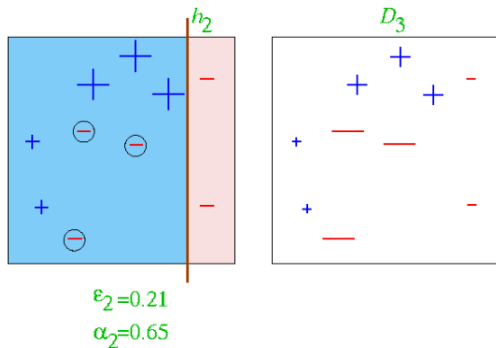
- Seria impeditivo utilizar todos os atributos. Dentre todos, apenas uma pequena parte geram valores consistentes quando posicionados sobre a face.
- Dessa forma, uma variante do algoritmo AdaBoost é usado para selecionar os atributos e treinar classificadores que utilizam esses atributos.
- AdaBoost (adaptive boosting) é um algoritmo capaz de construir um classificador forte a partir de uma combinação ponderada de diversos classificadores fracos.
 - para cada classificador treinado, computa um erro e um valor α que mede a “importância” do classificador.
 - no método de detecção de faces, cada atributo é considerado em um classificador fraco em potencial.

Treinamento usando boosting: exemplo



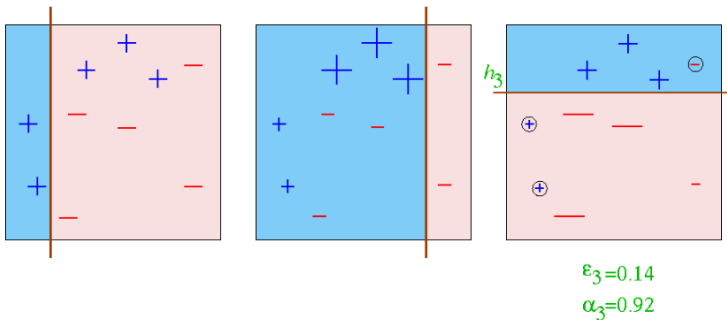
Exemplo obtido do material do prof. Padhraic Smyth.

Treinamento usando boosting: exemplo



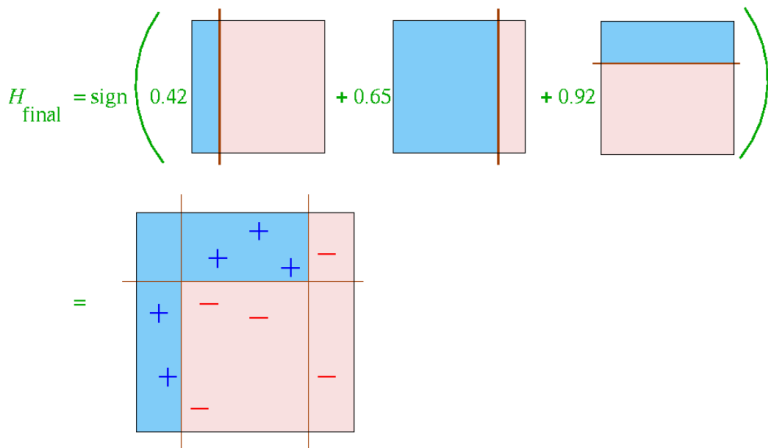
Exemplo obtido do material do prof. Padhraic Smyth.

Treinamento usando boosting: exemplo



Exemplo obtido do material do prof. Padhraic Smyth.

Treinamento usando boosting: exemplo

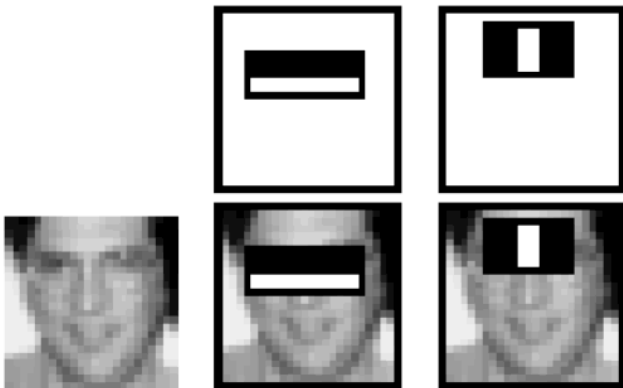


Exemplo obtido do material do prof. Padhraic Smyth.

Treinamento usando boosting

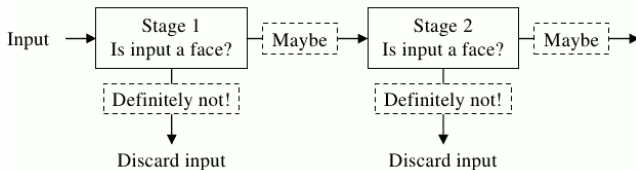
- Cada classificador é um “perceptron” de um único atributo, dentre os $K = 160000$ atributos existentes.
- A cada estágio do boosting:
 - 1 a partir dos pesos obtidos do estágio anterior (no estágio 1 os pesos são uniformes),
 - 2 treinar K classificadores,
 - 3 selecionar o melhor classificador nessa etapa,
 - 4 combinar com os outros classificadores previamente selecionados,
 - 5 recalculer os pesos,
 - 6 repetir até obter T classificadores/atributos.

Treinamento usando boosting



Classificação em cascata

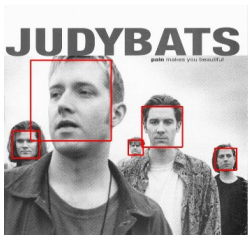
- Assumindo que o algoritmo irá percorrer a imagem várias vezes (utilizando diversas escalas) e que em grande parte das vezes analisará objetos negativos (não-faces),
- Ao invés de tentar encontrar faces diretamente, o problema foi modificado para **descartar não-faces** utilizando uma cascata.
- A idéia é construir classificadores rápidos para descartar as respostas negativas.
 - cada nó é treinado com os falsos positivos do anterior



Implementação

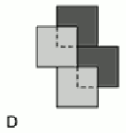
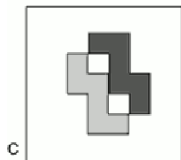
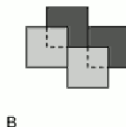
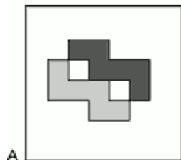
- Janelas de 24×24 pixels
- Escala dos detectores: fator de 1.25
- Deslocamento dos detectores: incremento de 1 pixel.
- Uma face real pode resultar em múltiplas detecções: pós-processar para combinar decisões sobrepostas.
- Treinamento pode durar 5 ou mais dias (ex. com 5 mil faces e 10 mil não-faces, 38 nós na cascata e 6060 atributos)

Alguns resultados



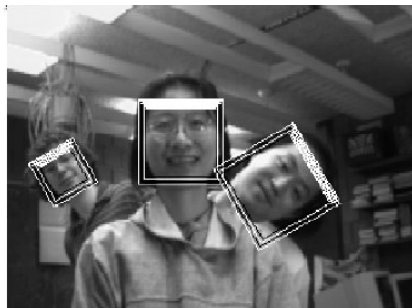
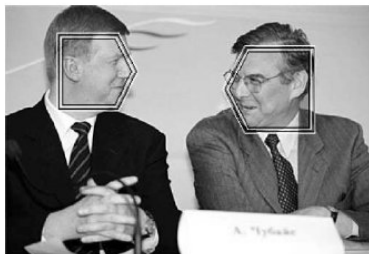
Pose e rotação

- Em 2003 os mesmos autores propuseram o uso de filtros diagonais para detectar faces em rotação e em perfil.



Pose e rotação

- A classificação é feita em 2 estágios para encontrar faces rotacionadas
- Uma árvore de decisão é utilizada para classificar a pose e encontrar faces rotacionadas.



References

- Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. IEEE Proc. Conf. Computer Vision and Pattern Recognition, 2001.
- Viola, P. and Jones, M. Fast Multi-view Face Detection. IEEE Proc. Conf. Computer Vision and Pattern Recognition, 2003.
- Viola, P. and Jones, M. Robust Real-Time Face Detection. Int. Journal Computer Vision, v57, n.2, pp 137–154, 2004.