

Reconhecimento de Padrões

(Detecção de Faces)

Sprint1

- ✓ **Introdução: O que é o Reconhecimento de Padrões**
 - ✓ Definições
 - ✓ Tipos:
 - ✓ Linguístico/Simbólico
 - ✓ Estatístico
 - ✓ Aplicações no Mundo Atual
- ✓ **Extração de Características**
 - ✓ Etapas de um Sistema de Reconhecimento de Padrões
 - ✓ Descritores de Objetos e Vetores de Características
 - ✓ Sistemas de Cores (RGB)
 - ✓ Histograma
 - ✓ Equalização de Histogramas (Contraste)
 - ✓ Suavização
 - ✓ Distribuição Normal ou Gaussiana
 - ✓ Binarização
 - ✓ Segmentação e Detecção de Bordas
 - ✓ Algoritmo Canny
 - ✓ Identificando e Contando Objetos
 - ✓ Optical Character Recognition
 - ✓ Sumarização

Sprint2

- **Redução de Dimensionalidade**
 - ✓ Definições
 - ✓ Maldição da Dimensionalidade
 - Detecção de Faces em Imagens
 - ✓ Haarcascades
 - ✓ Wavelets de Haar
 - ✓ Imagem Integral
 - ✓ Algoritmo Adaboost
 - Classificadores em Cascata
 - Detecção de Faces em Vídeos
 - Seleção de Características
 - Análise de Componentes Principais

Sprint3

- **Modelos de Descoberta**
 - **Clusterização**
 - K-means
 - **Regras de Associação**
 - Apriori
- Integração de Conceitos
 - PCA + Kmeans
 - Kmeans + Apriori

Breve Discussão Sobre Wavelets de Haar e Adaboost

Wavelet de Haar

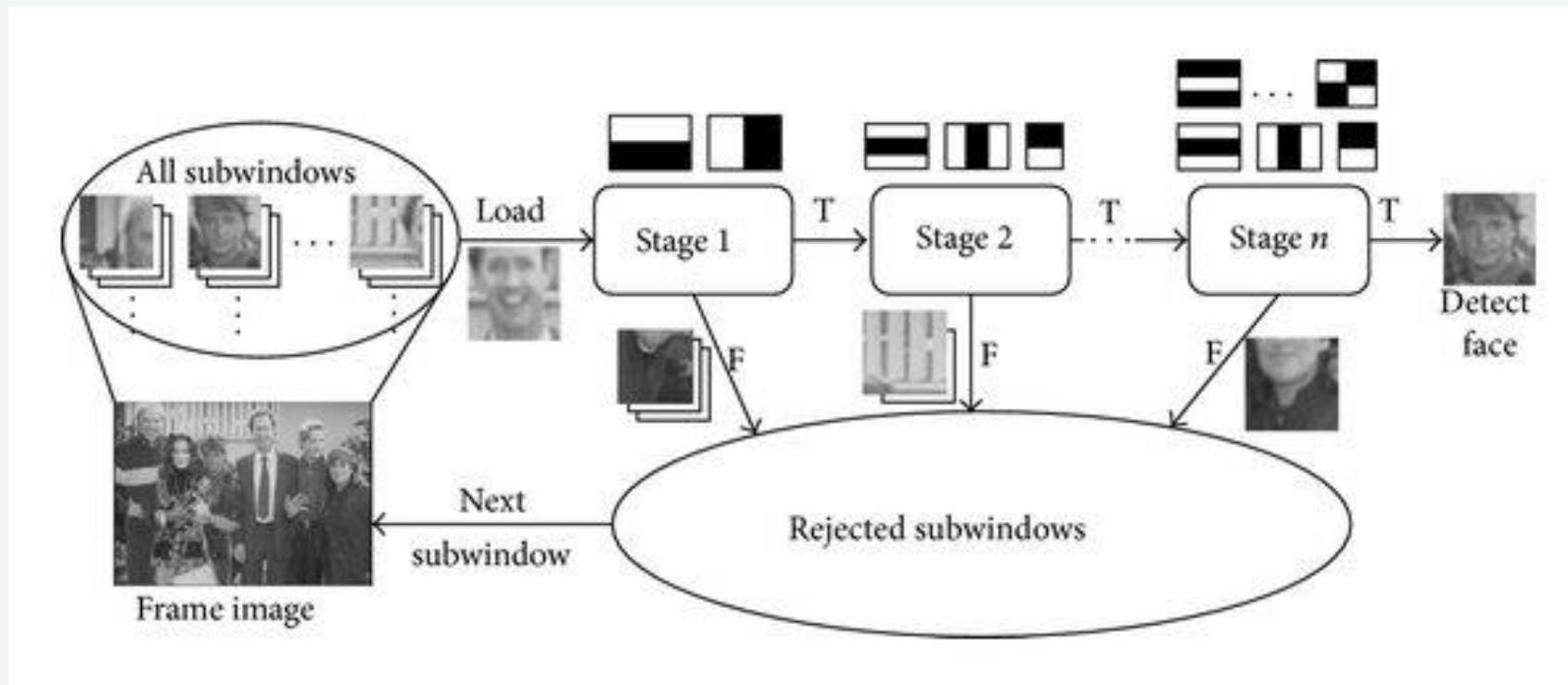
- Wavelet é uma função capaz de descrever outra função, de maneira que esta segunda possa ser analisada em diferentes escalas de frequência e tempo.

Adaboost

- Criado por Yoav Freund e Robert Schapire em 1995;
- Nem todas as características são úteis;
- O Adaboost é utilizado para selecionar as melhores características de cada janela;
- Visa melhorar a precisão do classificador através de uma combinação linear de vários classificadores fracos;
- Redução drástica de features de 180.000 para apenas 6.000;

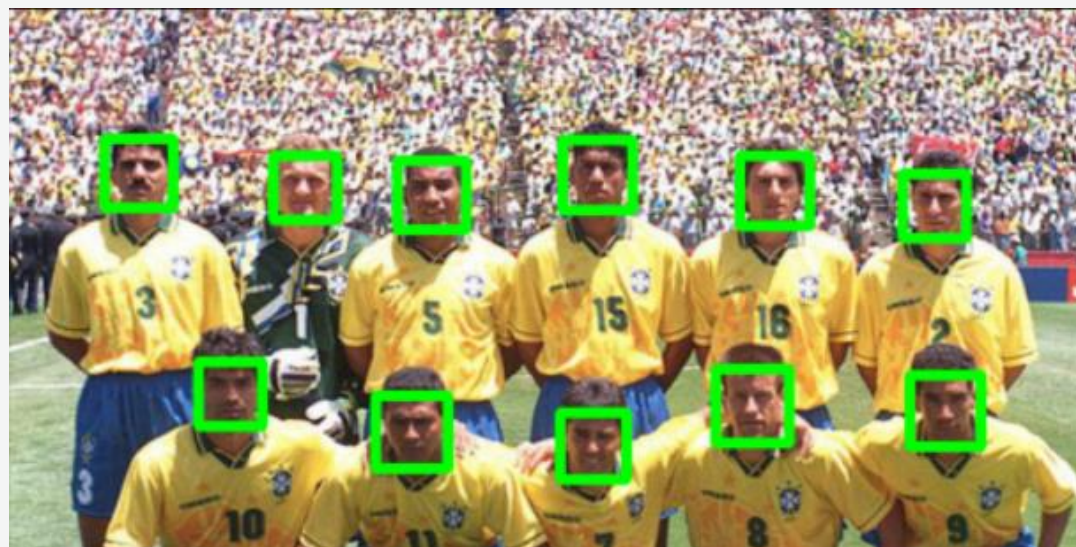
Classificadores em Cascata

- Basicamente, um classificador em cascata consiste em estágios, onde cada estágio é baseado em um classificador forte;
- Elimina a necessidade de aplicar todos os recursos de uma só vez;
- Agrupa os recursos em sub-janelas e o classificador em cada estágio determina se aquela sub-janela possui um face ou não;
- Caso não seja, a subjanela é descartada;
- Caso seja, a subjanela prossegue para o próximo estágio;



Identificação Facial em Imagens Utilizando o Haarcascades

- Carregar o modelo haarcascade a ser implementado. O repositório contendo os modelos mais populares e já treinados estão disponíveis em:
 - <https://github.com/opencv/opencv/tree/master/data/haarcascades>
- Utilizar a função de detecção: **detectMultiScale**, utilizando os seguintes parâmetros obrigatórios:
 - **Imagem**;
 - **ScaleFactor** – em fotos com muitos rostos, alguns destes podem estar mais próximos da câmera do que outros. Este fator compensa este distanciamento;
 - **minNeighbors** – este parâmetro especifica o número “vizinhos” que um retângulo DEVE ter para ser chamado de rosto;
 - Esta função retorna as coordenadas do retângulo (x,y,w,h) referente ao rosto encontrado.



Limitações do Haar Cascade

- Nos exemplos demonstrados o Haar Cascade funcionou de maneira conveniente, porém existem várias limitações:
 - Alta taxa de falsos-positivos;
 - Menos preciso do que as técnicas baseadas em deep-learning;
 - Ajuste manual de parâmetros;
 - Dificuldade em se treinar o haarcascade para um objeto personalizado.

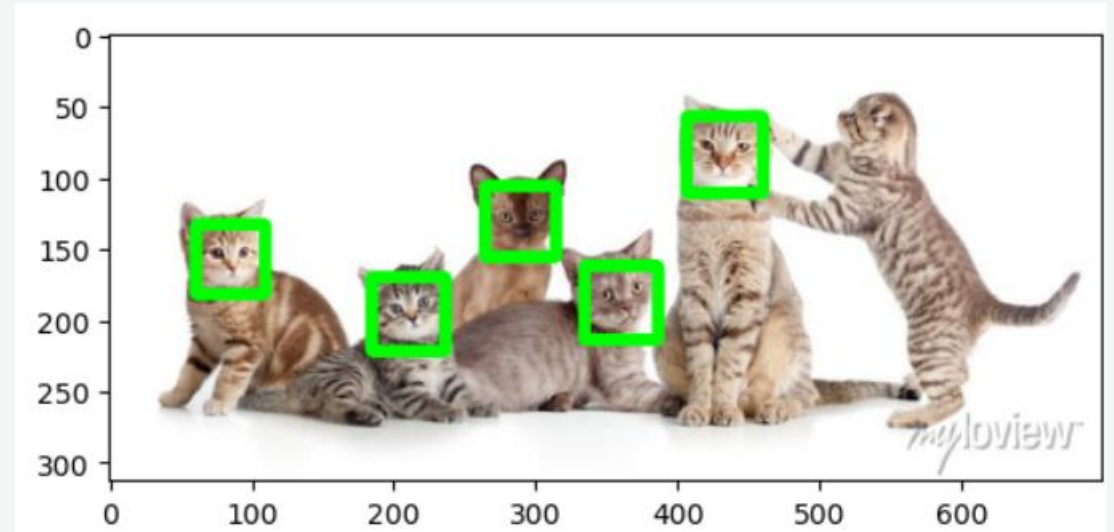
OpenLab

- Na imagem a seguir, reproduza o mesmo resultado, conforme demonstrado:

Imagem Original



Identificação de Rostos



- Lembre-se de anexar o código para que a entrega seja validada.