



# **Biometria Pessoal utilizando Técnicas para Detecção e Reconhecimento Facial**

Marcelo Costa da Conceição

Orientador: Prof. Dr. Dênis Fernandes

# ORGANIZAÇÃO

- **Introdução**
- **Referencial Teórico**
- **VIOLA-JONES (Detecção de Faces)**
- **EIGENFACES (Reconhecimento de Faces)**
- **Biblioteca OpenCV**
- **Implementação do Sistema**
- **Demonstração do Protótipo**
- **Considerações Finais**

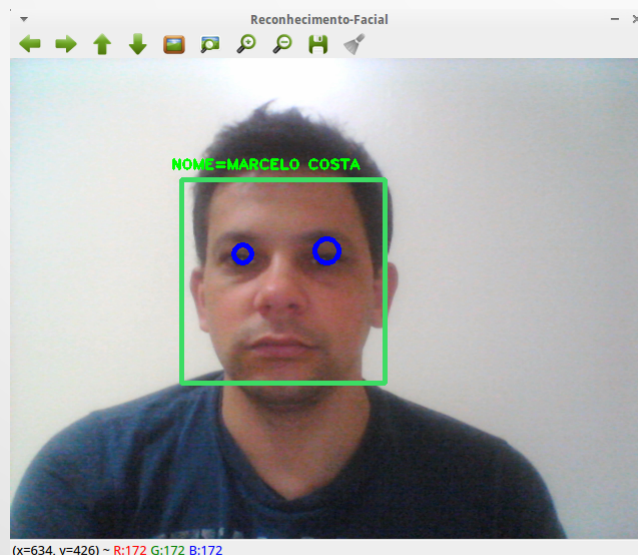
# INTRODUÇÃO

- **O projeto do sistema consiste basicamente nas seguintes etapas:**
- **Deteccção de faces**  
Utiliza o método proposto por Viola e Jones, que é baseado em filtros em cascata, que tem como base o treinamento de classificadores, agregando diversos classificadores fracos, gerando um classificador forte.
- **Extração de características**  
Utiliza a análise de componentes principais, que reduz a dimensionalidade do espaço enquanto preserva o máximo possível de informação.
- **Reconhecimento**  
Foram utilizadas métricas baseada em distância entre projeções matriciais.
- **Implementação**  
Foi utilizada a biblioteca de processamento de imagens OpenCV, em linguagem de programação C/C++.

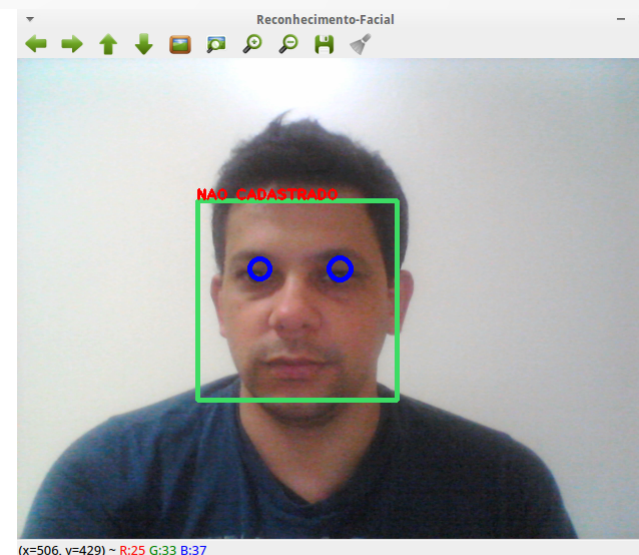
# INTRODUÇÃO

- **Objetivos**

A arquitetura básica do protótipo, consiste em receber como entrada o código correspondente a um dos usuários cadastrados (matrícula, código de barras, etc.), buscar e fazer o treinamento das imagens do respectivo usuário e identificar se a pessoa que está na frente da webcam corresponde ao código do usuário informado. Caso seja o usuário, mostra na tela o seu nome, fala uma mensagem via voz sintetizada, salva uma foto e escreve em um arquivo de relatório, o nome, a data e a hora. Caso não seja, somente mostra na tela a mensagem “NAO CADASTRADO”.



(a)



(b)

# REFERENCIAL TEÓRICO

- **Processamento de imagem digital**

o processamento de imagem digital será uma etapa anterior a etapa de visão computacional, isso se deve ao fato de que as imagens de onde queremos extrair alguma informação pode, em alguns casos, precisar ser convertidas para um determinado formato ou tamanho e precisam ainda ser filtradas para remover ruídos provenientes do processo de aquisição da imagem.

- **Técnicas de processamento utilizadas:**

- Conversão para tons de cinza
- Equalização, por histograma (Contrast Limited Adaptive Histogram Equalization)
- Filtro bilateral (reduz ruído, por suavização da imagem, mantendo as arestas)
- Conversão da imagem para 8 bits por pixel
- Redução da resolução

# REFERENCIAL TEÓRICO

- **Visão Computacional**

A visão computacional procura emular a visão humana e possui como entrada uma imagem e a saída é uma interpretação da imagem como um todo, ou parcialmente [ROSENFELD; KAK, 1982].

Em um sistema de reconhecimento facial, várias etapas requerem técnicas de visão computacional, entre elas a localização da face e extração de características.

## VIOLA-JONES (Detecção de Faces)

- A utilização de algoritmos para a extração de características de objetos e a geração de classificadores em cascata capazes de reconhecer padrões de objetos em uma imagem tem demonstrado resultados robustos, eficientes e extremamente rápidos para a detecção de objetos.
- Entre os trabalhos que abordam o problema de detecção de objetos, destaca-se a técnica proposta inicialmente por Viola e Jones (2001) por apresentar uma solução genérica quanto à detecção de qualquer tipo de objeto, além de possuir uma taxa muito pequena de ocorrência de falsos positivos.

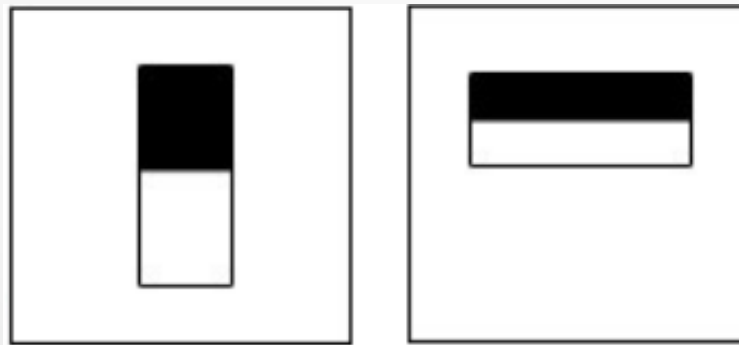
## **VIOLA-JONES (Detecção de Faces)**

- A rotina para detecção de objetos é baseada nas principais características de um objeto previamente extraídas a partir de um algoritmo de aprendizado de máquina. Tais características são responsáveis por diferenciar objetos uns dos outros.
- A maior motivação para o uso de características de um objeto ao invés do uso de seus pixels, é que o tempo de processamento da análise em uma imagem baseada no conjunto de suas principais características é muito menor do que uma análise sobre os seus respectivos pixels, devido a ter uma dimensão maior.



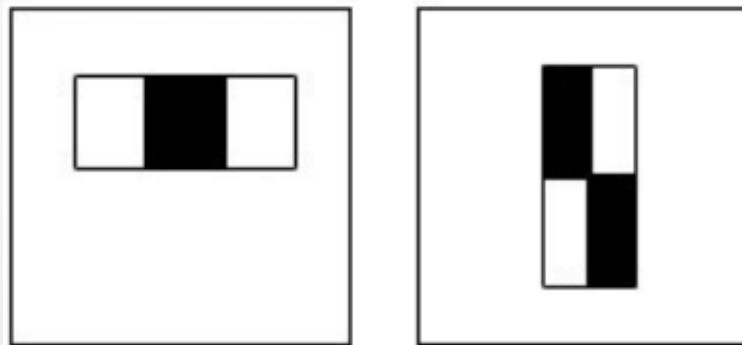
## VIOLA-JONES (Detecção de Faces)

- Viola e Jones (2001) propõem a utilização de quatro formatos de características retangulares:
- Característica de “dois retângulos”, onde o seu valor é a diferença entre a soma dos pixels dentro de duas regiões retangulares adjacentes e de mesmo tamanho;



# VIOLA-JONES (Detecção de Faces)

- Característica de “três retângulos”, cujo valor é a soma em um retângulo central menos a soma de dois retângulos externos;
- Características de “quatro retângulos” que possuem seu valor calculado através da diferença entre pares diagonais de retângulos.



# VIOLA-JONES (Detecção de Faces)

- Com o intuito de otimizar o cálculo de tais características retangulares (Haar), Viola e Jones (2001) propõem a utilização de uma representação intermediária da imagem, chamada de Imagem Integral.
- Nesta forma de representação, cada ponto  $x, y$  da imagem contém o somatório de pixels da origem da imagem até a sua localização.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

# VIOLA-JONES (Detecção de Faces)

Imagem integral é uma representação da imagem original que permite avaliar eficientemente a soma dos valores dos pixels (intensidade dos níveis de cinza) de uma área retangular em uma sub-região da imagem.

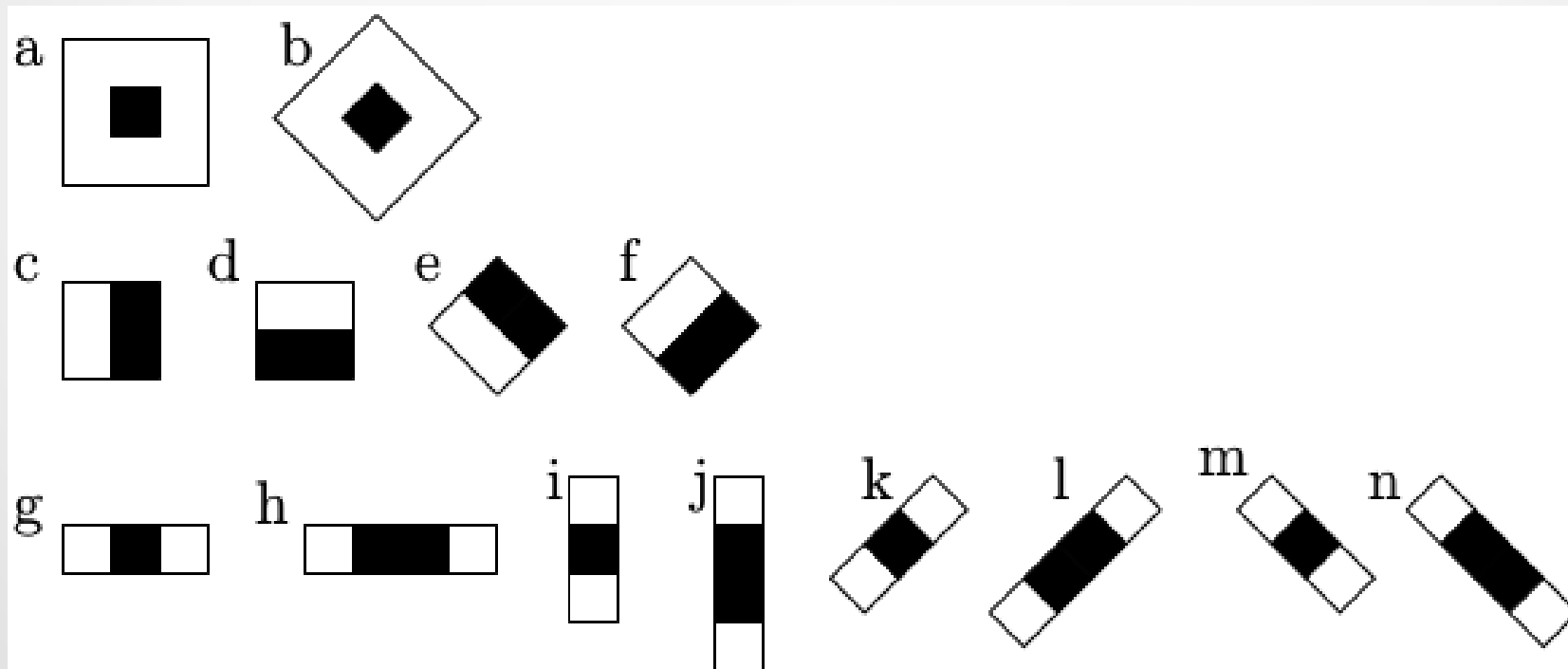
Assim, o valor para cada pixel pode ser obtido facilmente por uma única varredura da imagem, já que pode ser calculado pelos valores de pixels adjacentes:

$$ii(x, y) = i(x, y) + ii(x-1, y) + ii(x, y-1) - ii(x-1, y-1)$$

Isto nos permite calcular a soma de pixels em um dado retângulo de tamanho qualquer usando apenas quatro valores básicos.

# VIOLA-JONES (Detecção de Faces)

A fim de aumentar a qualidade na a detecção de faces, foram adicionadas ao algoritmo, características rotacionais que permitiram tolerância à inclinação [LIENHART; MAYDT, 2002].



# VIOLA-JONES (Detecção de Faces)

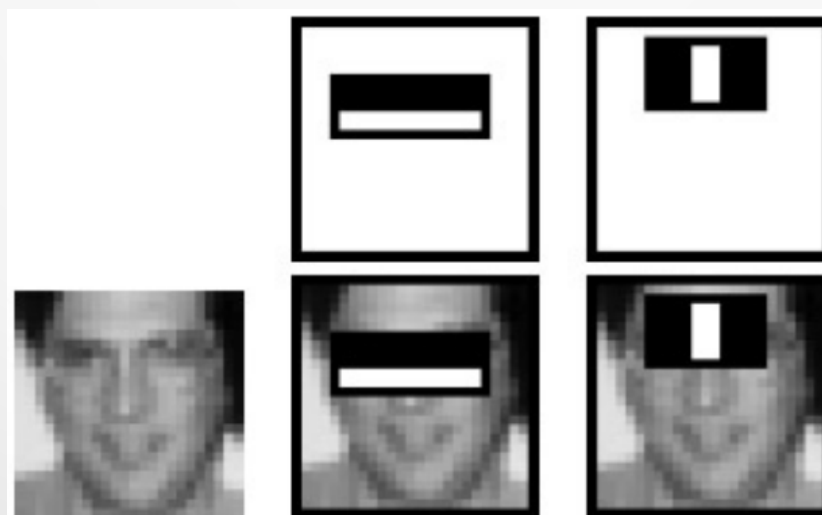
- **Classificação**

O objetivo principal a ser alcançado por uma função de classificação é o de ser capaz de classificar um dado objeto a partir do conjunto de suas principais características.

- O algoritmo escolhido por Viola e Jones (2001) é uma variante do algoritmo de aprendizado AdaBoost (Freund e Schapire, 1995).

## VIOLA-JONES (Detecção de Faces)

Um classificador construído sobre um bom conjunto de características passa a avaliar regiões da imagem de forma correta e precisa. No exemplo da figura abaixo, o classificador percorre a imagem procurando regiões com os mesmos padrões que as características do objeto desejado, no exemplo da figura em questão este objeto é uma face humana.



## VIOLA-JONES (Detecção de Faces)

- Para que o algoritmo de aprendizado AdaBoost, seja executado, são necessários dois grupos de imagens do objeto de interesse. **Imagens positivas**, e um segundo grupo de imagens que não contenha o objeto, chamado de **imagens negativas**.
- O AdaBoost é utilizado tanto para selecionar as características quanto para treinar o classificador, sendo capaz de encontrar um pequeno número de características altamente relevantes no objeto comparando as imagens nos conjuntos positivos e negativos.



# VIOLA-JONES (Detecção de Faces)

- **Pseudocódigo de Treinamento pelo algoritmo AdaBoost**

- Dadas as imagens exemplo  $(x_1, y_1), \dots, (x_n, y_n)$  de 24x24 pixels:
  - $y_i = 0, 1$  para imagens negativas e positivas respectivamente.
- Inicializar pesos  $w_i = 1/2m, 1/2l$  para  $y_i = 0, 1$  respectivamente.
  - $m$ : número de amostras negativas;
  - $l$ : número de amostras positivas.
- De  $t = 1$  até  $T$  repetir:
  - Normalizar os pesos:  $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$
  - Para cada característica  $j$ , treinar um classificador  $h_j$  que seja restrito a utilizar apenas uma característica. O erro é calculado pela equação:
$$w_t, E_j = \sum_i w_i |h_j(x_i) - y_i|$$
  - Escolher o classificador  $h_t$ , com o menor erro  $E_t$

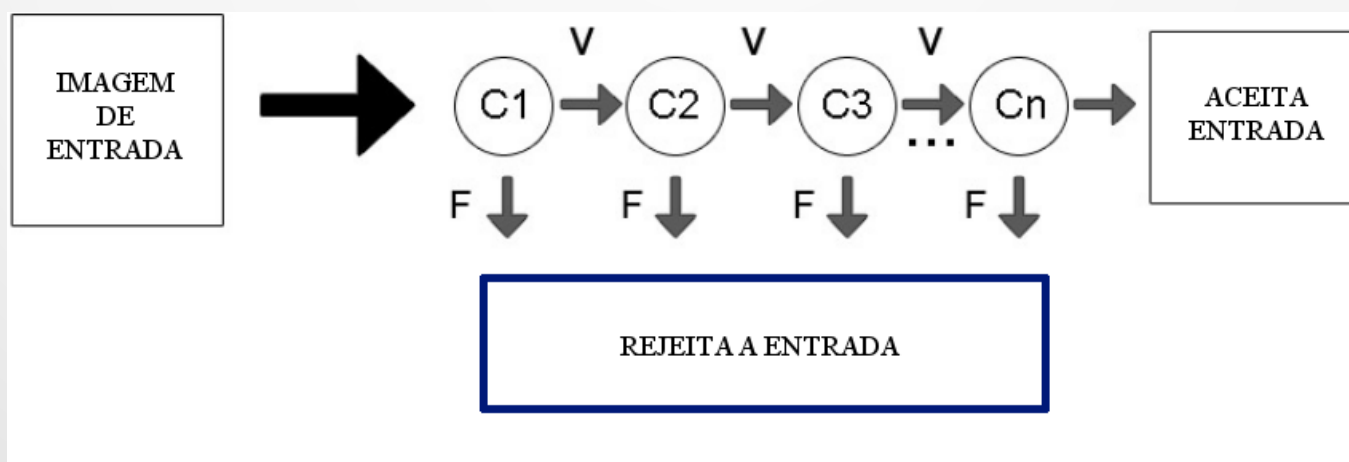
# VIOLA-JONES (Detecção de Faces)

- Atualizar os pesos:  $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$   $\beta_t = \frac{E_t}{1-E_t}$ 
  - Se a amostra  $x_i$  for classificada corretamente,  $e_i=0$ .
  - $e_i=1$ , caso contrário
- Ao término, o classificador forte é: 
$$h(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{caso contrário} \end{cases}$$
  - onde  $\alpha_t = \log\left(\frac{1}{\beta_t}\right)$

# VIOLA-JONES (Detecção de Faces)

- **Cascata de classificadores**

A idéia de usar uma cascata (árvore) de classificadores com  $N$  estágios tem como objetivo otimizar o reconhecimento de objetos. Estágios dentro de uma cascata são criados através da combinação de funções de classificação previamente montadas pelo uso do algoritmo de aprendizado AdaBoost.



# VIOLA-JONES (Detecção de Faces)

- O principal objetivo da cascata de classificadores é fazer com que seus estágios iniciais descartem um grande número de regiões que contém o objeto desejado, e estágios mais avançados sejam cada vez mais precisos para evitar um falso positivo na região sendo analisada. Caso uma área da imagem passe pelo último estágio da cascata, então esta área contém o objeto desejado.
- O processo para geração da cascata é guiado por um conjunto de metas de detecção e de desempenho. O número de estágios na cascata devem ser o suficiente para garantir uma taxa elevada de detecção de objetos, uma baixa ocorrência de falsos positivos e a minimização do tempo de processamento durante o teste de uma região da imagem.

# VIOLA-JONES (Detecção de Faces)

- **Busca pelo objeto de interesse**

É necessário que a imagem seja percorrida diversas vezes em vários tamanhos, fazendo com que cada vez seja analisada uma região de tamanho diferente, o que é chamado de janela de busca multiescala.

- O tamanho mínimo desta janela de busca deve ser maior ou igual ao tamanho com o qual a cascata foi treinada, ou seja, se a cascata foi treinada com imagens positivas de 24 x 24 pixels, o tamanho mínimo da janela de busca deve seguir tal resolução.

# VIOLA-JONES (Detecção de Faces)

- A janela de busca deve ser deslocada um valor  $\Delta$  de pixels na horizontal enquanto houver janelas a serem analisadas no eixo x da imagem, e do mesmo valor  $\Delta$  de pixels na vertical, enquanto houver tamanho suficiente para novas janelas no eixo y.
- A partir de leitura de diversos materiais para estudo, foi constatado que  $\Delta$  igual a 1,1 é um bom valor para o deslocamento horizontal/vertical da janela de busca.

Ex:

```
face_cascade.detectMultiScale(dstClashe, faces, 1.1, 4, 0|CV_HAAR_SCALE_IMAGE, Size(24, 24));
```

# VIOLA-JONES (Detecção de Faces)

- A imagem também deve ser percorrida em várias escalas. Portanto, da mesma forma como se tem um valor  $\Delta$  para o deslocamento da janela de busca, também é necessário existir um fator  $\delta$  de escalonamento da janela, ou seja, um  $\delta$  que aumente o tamanho da janela como um todo.
- Desta forma, cada vez que a janela de busca terminar de percorrer toda a imagem na horizontal e na vertical, a imagem deve ser percorrida novamente com um novo tamanho para a janela de busca.
- Baseado em seus próprios experimentos, Viola e Jones (2001) sugerem 1,25 como sendo o valor ideal para  $\delta$ .

# VIOLA-JONES (Detecção de Faces)

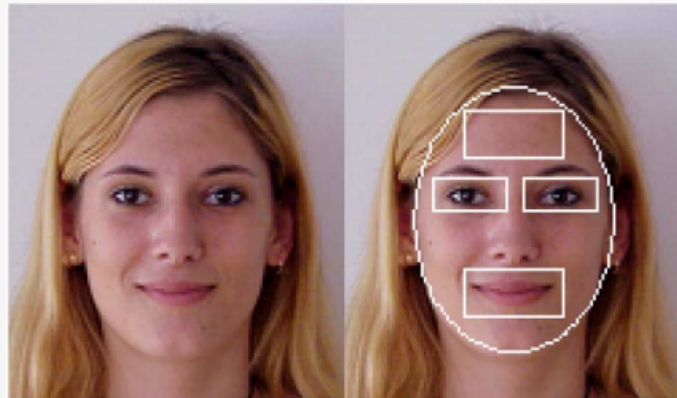
- **Pseudocódigo**

- Tamanho inicial da janela de busca:  $J(w_j, h_j)$
- Tamanho da imagem original:  $I(w, h)$
- Fator de escala  $\delta$
- Fator de deslocamento da Janela  $\Delta$ .
- Enquanto  $J(w_j, h_j) * \delta < I(w, h)$  faça
  - Varrer a imagem com a sub-janela, realizando deslocamentos de  $\Delta$  pixels
    - Verificar se a área da janela de busca na imagem contém ou não o objeto procurado.
    - Caso positivo, armazenar as informações da localização atual da janela de busca.
  - Marcar na imagem as localizações de objetos encontradas.



# VIOLA-JONES (Detecção de Faces)

O algoritmo de Viola-Jones, se bem treinado, tem uma boa precisão, e consegue detectar a maioria das faces que são captadas frontalmente, além de apresentar um baixo número de falsos positivos.



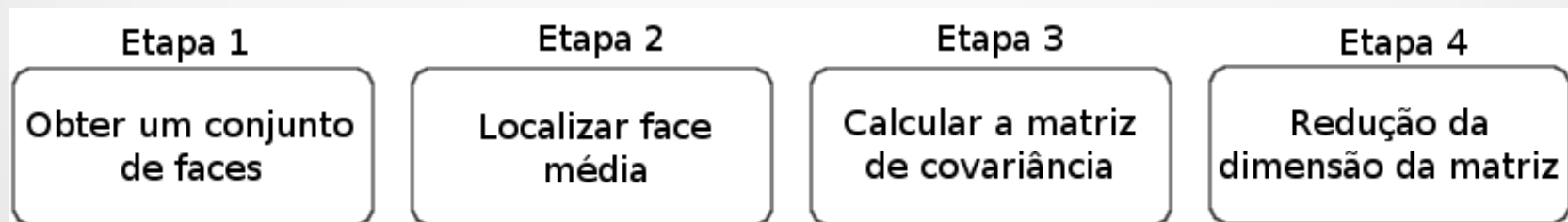
# EIGENFACES (Reconhecimento de Faces)

- Geralmente as técnicas para reconhecimento facial, necessitam de várias imagens e, a partir delas, aprendem ou codificam somente o que é necessário para realizar a detecção da característica de interesse. Exemplos de tal abordagem são as Redes Neurais, Eigenfaces e Modelos Ocultos de Markov [LOPES, 2009].
- Um dos métodos mais consagrados é o de Autofaces (Eigenfaces), derivado da Transformada de Karhunen-Loève [MELLO, 2003], também conhecida por Análise de Componentes Principais (PCA).

# EIGENFACES (Reconhecimento de Faces)

As Eigenfaces buscam identificar um pequeno número de características que são relevantes para diferenciar uma face de outras faces.

Essas características podem ser analisadas apenas com a variação dos valores assumidos pelos pixels, em um conjunto de imagens de faces [DINIZ; et al., 2013].



# EIGENFACES (Reconhecimento de Faces)

- **Conjunto de Faces**

Para esta etapa é necessário que as imagens tenham as mesmas características, como tamanho (altura e largura da imagem), condições de iluminação e elementos faciais facilmente localizados.

Esse conjunto normalmente é formado por imagens em escala de cinza e equalizadas, e denominamos como conjunto de treinamento



# EIGENFACES (Reconhecimento de Faces)

- **Localização da face média**

Para eliminar informação redundante na face, isto é, uma imagem de face poderia ser representada com poucas componentes principais devido à sua redundância, é feito uma média aritmética do somatório dos vetores das faces

$$\bar{\Gamma} = \frac{1}{N} \sum_i \Gamma_i$$

# EIGENFACES (Reconhecimento de Faces)

- **Cálculo da matriz de covariância**

Após a subtração da imagem média de cada imagem, gera-se uma matriz  $A$  e assim podemos calcular a matriz de covariância, com a transposição de  $A$ .

$$C = A * A^T$$

# EIGENFACES (Reconhecimento de Faces)

- **Redução da dimensionalidade da matriz**

Por fim, na quarta etapa, são aplicadas a redução de dimensionalidade na matriz  $A$ , através do método de Análise de Componentes Principais (PCA), que necessita da matriz de covariância  $C$ , criando em uma nova matriz  $P$  com os autovalores  $u$  e autovetores  $v$  mais significativos da matriz  $C$ .

$$P = v * \Phi$$

# OpenCV

- OpenCV (Open Source Computer Vision Library). Originalmente, desenvolvida pela Intel, em 2000, é uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional
- Windows, Linux, Mac, Android, IOS
- C/C++, Python, Visual Basic, Ruby, Matlab



# OpenCV - Funcionalidades

- Manipulação de dados de imagens
- E/S de imagem e vídeo
- Manipulação de matrizes e vetores
- Rotinas de álgebra linear
- Estruturas de dados dinâmicas
- Processamento de imagens básico

# OpenCV - Funcionalidades

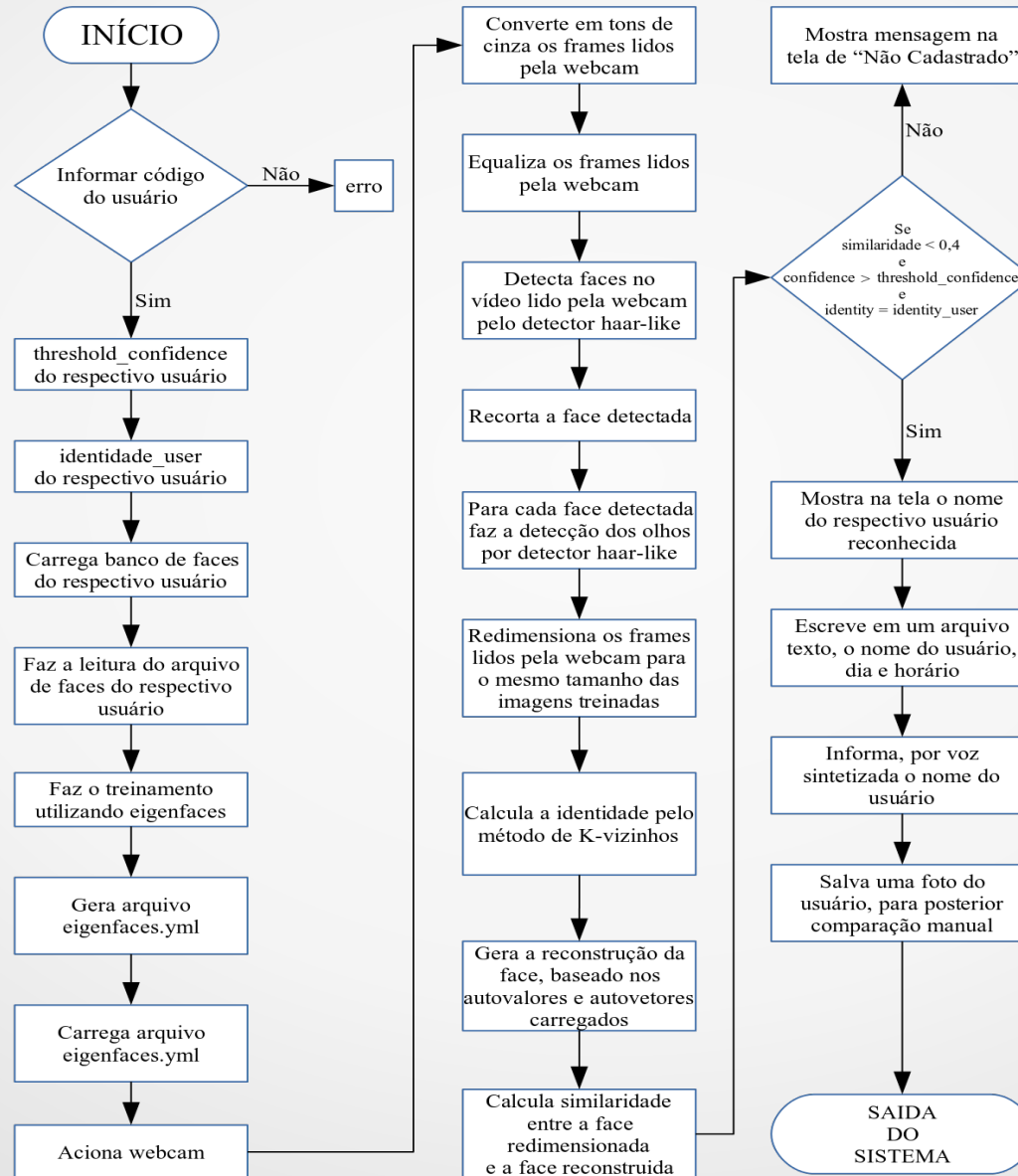
- Análise estrutural
- Calibragem de câmera
- Análise de movimento (tracking)
- Reconhecimento de objetos
- GUI Básica
- Rotulagem de imagem

# Implementação do Sistema

- Criação do conjunto de faces que serão treinadas;
- Treinar as faces e criar um modelo;
- Detecção de faces no vídeo;
- Pré-processamento da imagem de entrada;
- Reconhecimento e identificação da face.

# Implementação do Sistema

## Fluxograma Funcional



# Considerações Finais

- O sistema desenvolvido é constituído em módulos de detecção facial, extração de características e de reconhecimento.
- O módulo de detecção facial foi baseado no algoritmo de Viola-Jones e apresentou resultado excelente, detectando com eficiência 92% das faces.
- O módulo de extração de característica, baseado na análise de componentes principais (PCA), mostrou-se eficiente e rápido ao calcular o subespaço de características (autovalores e autovetores).
- Na parte de reconhecimento, a identificação correta atingiu bons níveis de eficiência em faces plenamente frontais e em ambiente controlado.

# Considerações Finais

- A abordagem proposta não fez nenhum tipo de teste de detecção de vida. De posse de uma foto de boa qualidade de um legítimo um “impostor” poderia ter o acesso a determinado recurso indevidamente liberado.
- A taxa de falsos-positivos, em torno de 20% ainda é uma limitação para o uso em aplicações de segurança no mundo real.

# Considerações Finais

- Implementar técnicas de detecção de vida (Liveness Detection).
- Aumentar a robustez do sistema em relação à pose do usuário
- Implementar outras métricas para o reconhecimento de faces