

FERRAMENTA DE INDEXAÇÃO DE FACES HUMANAS EM VÍDEOS

Aluno: Leonardo Leal Oliveira

Orientador: Dalton Solano dos Reis

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos
- Especificação
- Implementação
- Operacionalidade da implementação
- Resultados e discussões
- Conclusões e sugestões
- Demonstração

Introdução

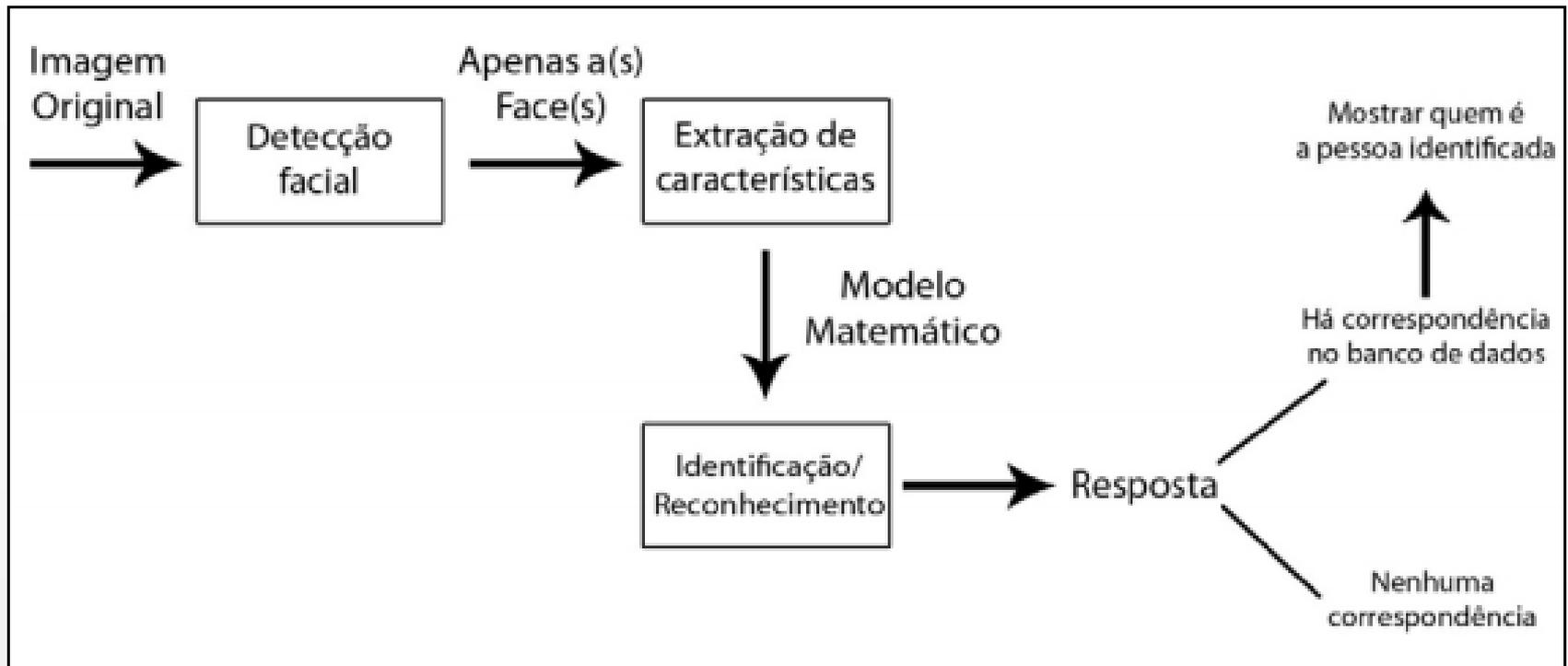
- Motivação
 - Grande quantidade de vídeos
 - Celulares
 - Câmeras de segurança
- Busca em coleções/bibliotecas
 - Indexação
 - Faces humanas

Objetivos

- Implementar uma ferramenta para indexar uma coleção de vídeos com base nas faces humanas.
- **Específicos:**
 - I. disponibilizar uma ferramenta reaproveitável e multiplataforma com a tecnologia dotNet;
 - II. disponibilizar uma coleção com as faces reconhecidas em cada vídeo;
 - III. possibilitar marcar/identificar um rosto durante a reprodução de um vídeo.

Fundamentação Teórica

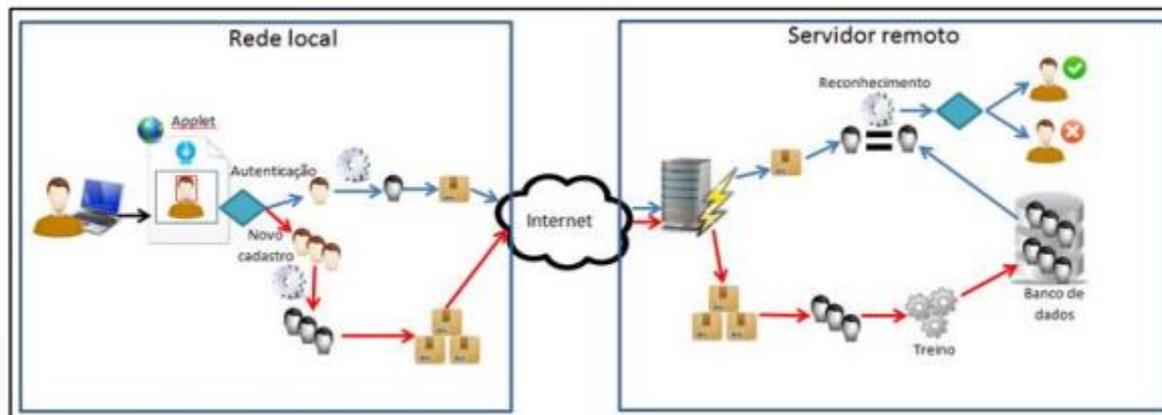
- Reconhecimento facial



Trabalhos Correlatos

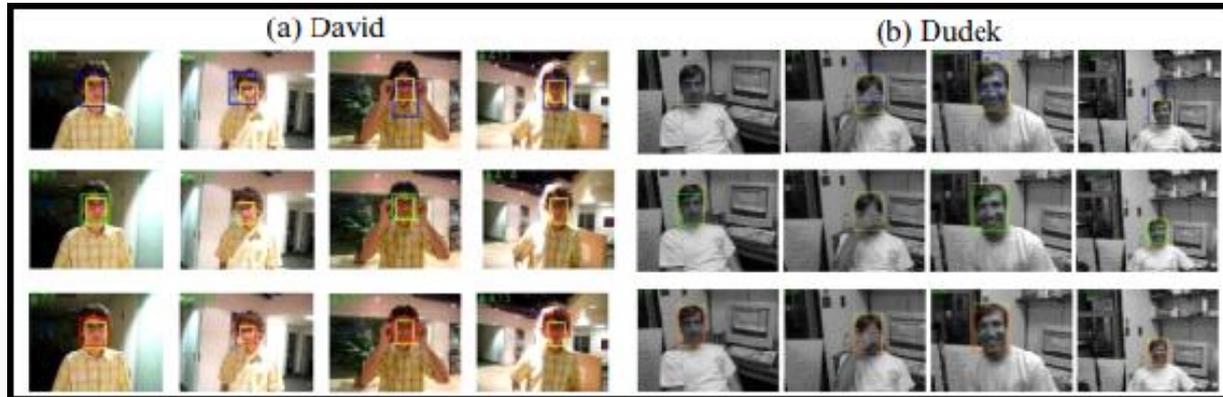
- Koch (2012)
 - Visão computacional para reconhecimento de faces aplicado na identificação e autenticação de usuários na web
- Jiang et al. (2016)
 - A fusion method for robust face tracking
- Video Indexer (2017)
 - Video Indexer

Koch (2012)



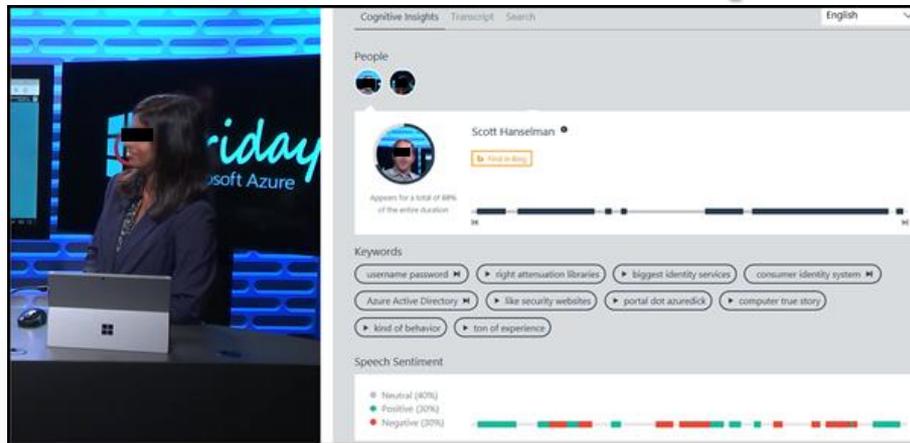
	Trabalhos	Koch (2012)
Características		
ferramenta reaproveitável (API)		Não
processamento distribuído		Sim
efetua identificação		Sim
realiza detecção facial em tempo real		Sim
permite marcar/identificar uma face durante a execução/visualização do vídeo		Não

Jiang et al. (2016)



Características	Trabalhos	Jiang et al. (2016)
ferramenta reaproveitável (API)		Não
processamento distribuído		Não
efetua identificação		Não
realiza detecção facial em tempo real		Sim
permite marcar/identificar uma face durante a execução/visualização do vídeo		Não

Video Indexer (2017)



	Trabalhos	Video Indexer (2017)
Características		
ferramenta reaproveitável (API)		Sim
processamento distribuído		Sim
efetua identificação		Sim
realiza detecção facial em tempo real		Não
permite marcar/identificar uma face durante a execução/visualização do vídeo		Não

Requisitos

- **Funcionais:**

- I. receber um vídeo como entrada e retornar uma coleção com as faces reconhecidas;
- II. receber um vídeo e a imagem de uma face como entrada e retornar se a face foi encontrada dentro do vídeo;
- III. ter métodos de busca do status atual da detecção facial durante a execução/processamento do vídeo.

Requisitos

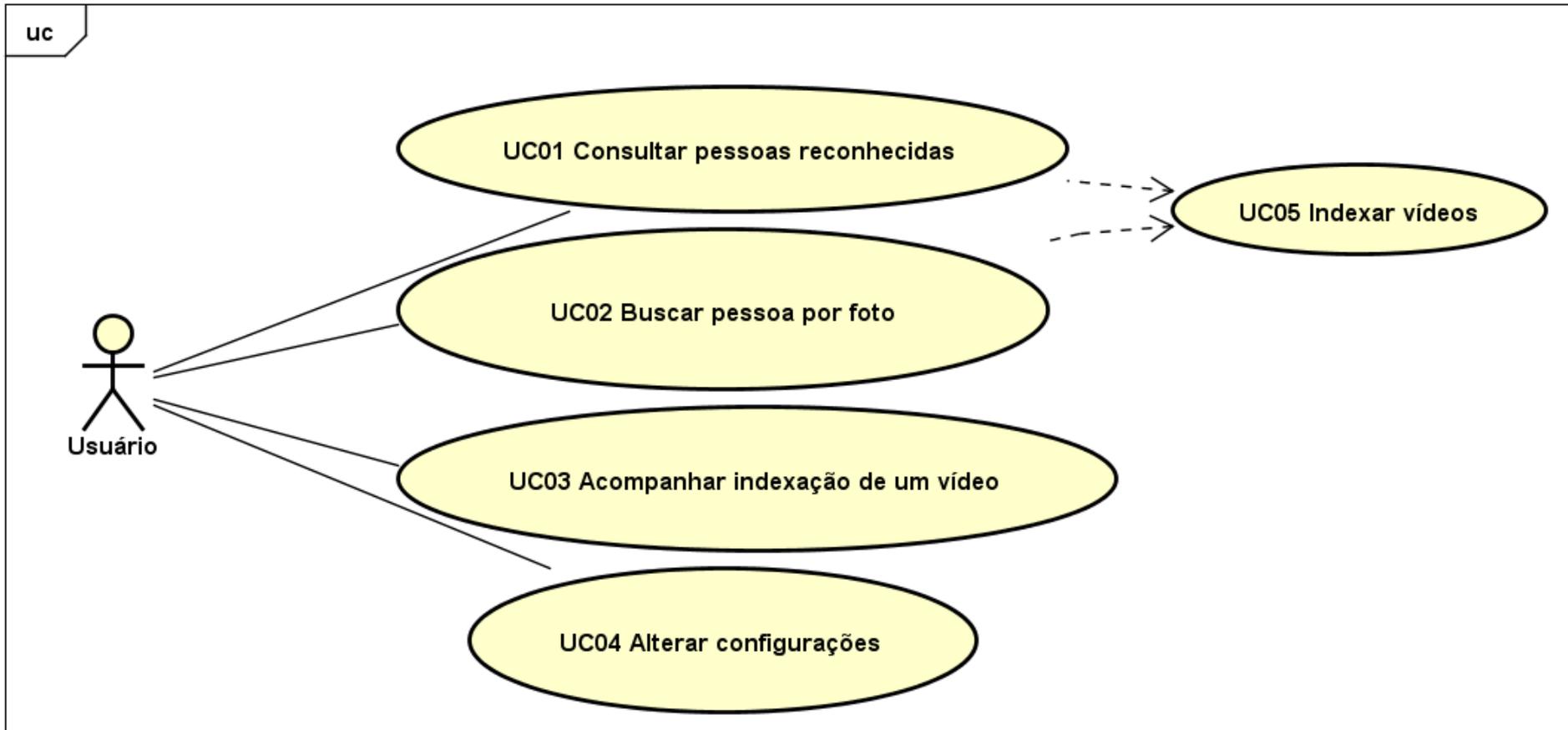
- **Não funcionais:**

- I. ser multiplataforma e reaproveitável para uso em outras aplicações;
- II. ser implementada na linguagem C#, utilizando a IDE Visual Studio 2017 e utilizando a plataforma dotNet;
- III. utilizar algoritmos de pré-processamento e segmentação para a detecção das faces;
- IV. utilizar bibliotecas de processamento de imagens como EmguCV e/ou Viola-Jones.

Especificação

- Diagrama de casos de uso
- Diagrama de atividades
 - Detecção
 - Reconhecimento

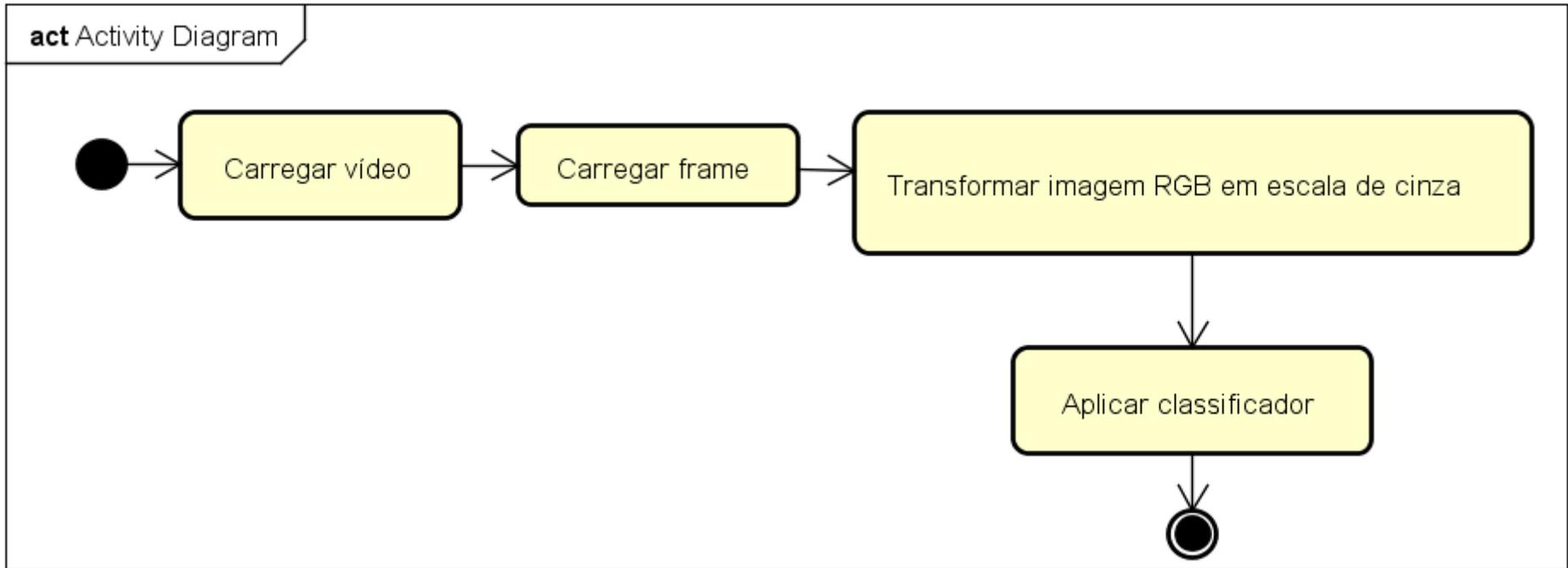
Diagrama de casos de uso



powered by Astah

Diagrama de atividades

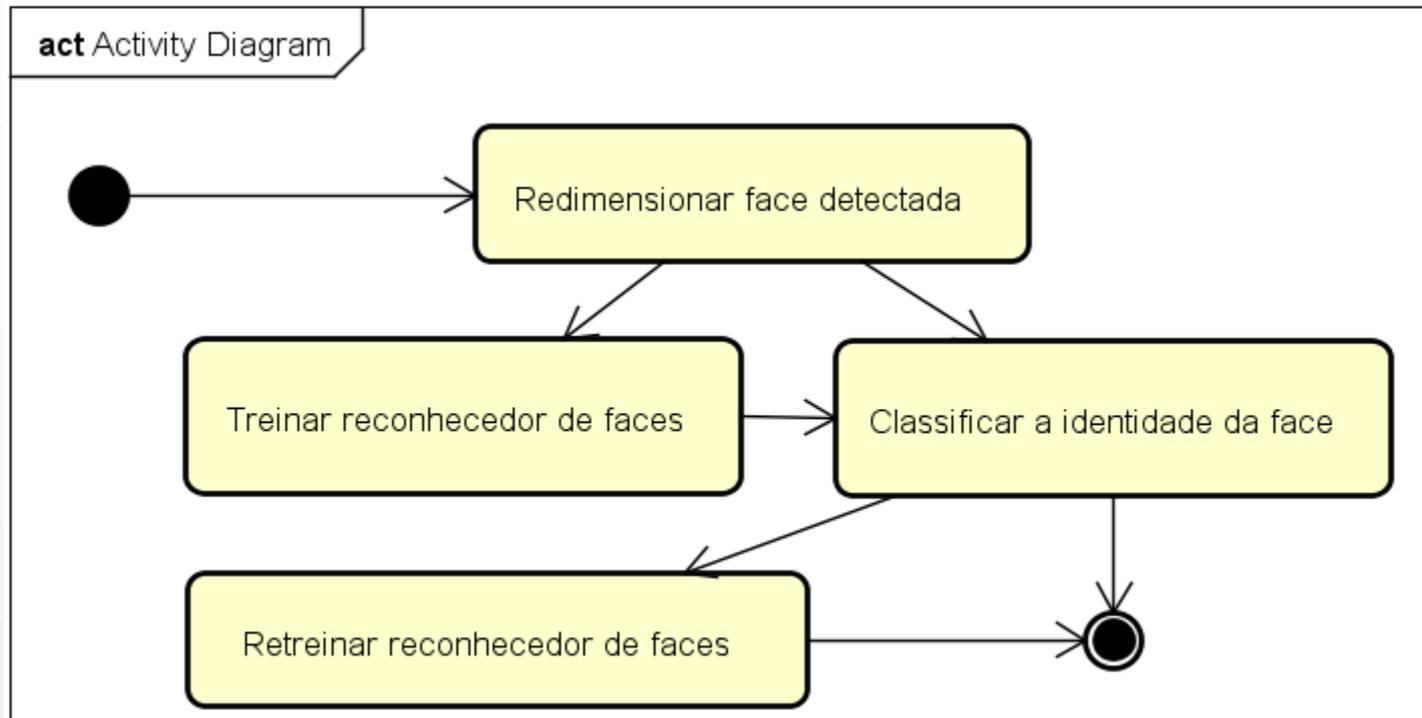
- Detecção Facial



powered by Astah

Diagrama de atividades

- Reconhecimento Facial



Implementação

- Detecção facial
 - Carregamento do vídeo e primeiro *frame*
 - Escala de cinza e classificação
- Reconhecimento facial
 - Normalização da face
 - Treinamento do reconhecedor
 - Classificação das faces
 - Desenho do retângulo e nome da pessoa

Detecção Facial

- Carregamento do vídeo e primeiro *frame*

```
Capture = new Capture(Video.FilePath);
```

```
CurrentFrame = new Image<Bgr, Byte>(Capture.QueryFrame().Bitmap);
```

Detecção Facial

- Escala de cinza e classificação

```
Image<Gray, Byte> currentGrayFrame = CurrentFrame.Convert<Gray, Byte>();  
  
Rectangle[] facesDetected =  
CascadeClassifier.DetectMultiScale(currentGrayFrame);
```

Reconhecimento Facial

- Normalização da face

```
Image<Gray, Byte> faceRegion = currentGrayFrame.Copy(face);  
  
faceRegion = faceRegion.Resize(DataFile.config.FaceRectangleSize,  
DataFile.config.FaceRectangleSize, Emgu.CV.CvEnum.Inter.Cubic);
```

Reconhecimento Facial

- Treinamento do reconhecedor

```
if (TrainImages.Count == 0)
{
    idPerson = TrainLabels.Count + 1;

    TrainImages.Add(faceRegion);
    TrainLabels.Add(idPerson);
    FaceRecognizer.Train(TrainImages.ToArray(), TrainLabels.ToArray());

    SaveFace(CurrentFrame.Copy(originalFace), originalFace, idPerson);
}
```

Reconhecimento Facial

- Classificação das faces

```
FaceRecognizer.PredictionResult predictionResult =  
FaceRecognizer.Predict(faceRegion);  
  
if (predictionResult.Label == 0)  
{  
    idPerson = TrainLabels.Count + 1;  
  
    TrainImages.Add(faceRegion);  
    TrainLabels.Add(idPerson);  
    FaceRecognizer.Train(TrainImages.ToArray(), TrainLabels.ToArray());  
  
    SaveFace(CurrentFrame.Copy(originalFace), originalFace, idPerson);  
}  
else  
    idPerson = predictionResult.Label;
```

Reconhecimento Facial

- Desenho do retângulo e nome da pessoa

```
foreach (Face face in frame.Faces)
{
    CurrentFrame.Draw(face.Position, new Bgr(DataFile.config.RectangleColor),
DataFile.config.RectangleThickness);

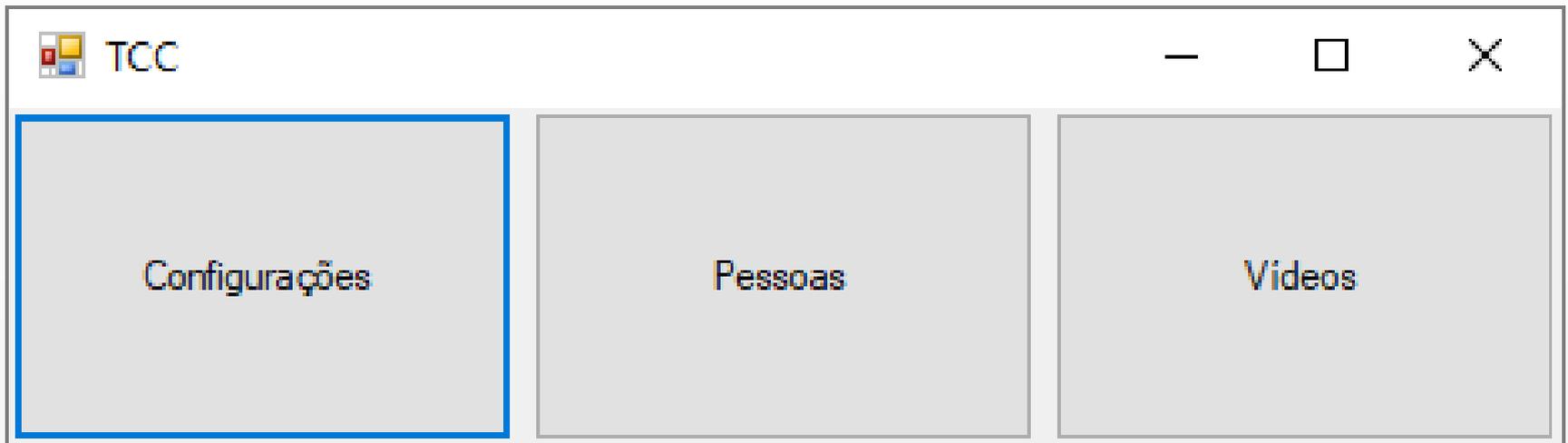
    Graphics g = Graphics.FromImage(CurrentFrame.Bitmap);

    g.DrawString(DataFile.GetPerson(face.IdPerson).Name, new
Font(DataFile.config.FontName, DataFile.config.FontSize), new
SolidBrush(DataFile.config.ColorFont), new Point(face.Position.Location.X,
face.Position.Location.Y));
}
```

Operacionalidade da Implementação

- Telas
 - I. Principal
 - II. Configurações
 - III. Pessoas
 - IV. Vídeo
 - V. Vídeos

Tela Principal



Tela de Configurações

Configurações

Interface

Nome Fonte:

Tamanho Fonte:

Cor Fonte:

Espessura Retângulo:

Cor Retângulo:

Detecção

Haar Cascade:

Tamanho Retângulo Face:

Escala Redimensionamento:

Arquivos

Videos:

Faces:

Extensão Vídeos:

Extensão Faces:

Reconhecimento

Threshold:

Radius:

Neighbors:

GridX:

GridY:

Tela de Pessoas

Windows application window titled "Pessoas".

Top-left panel: "Nome:" "Face:"

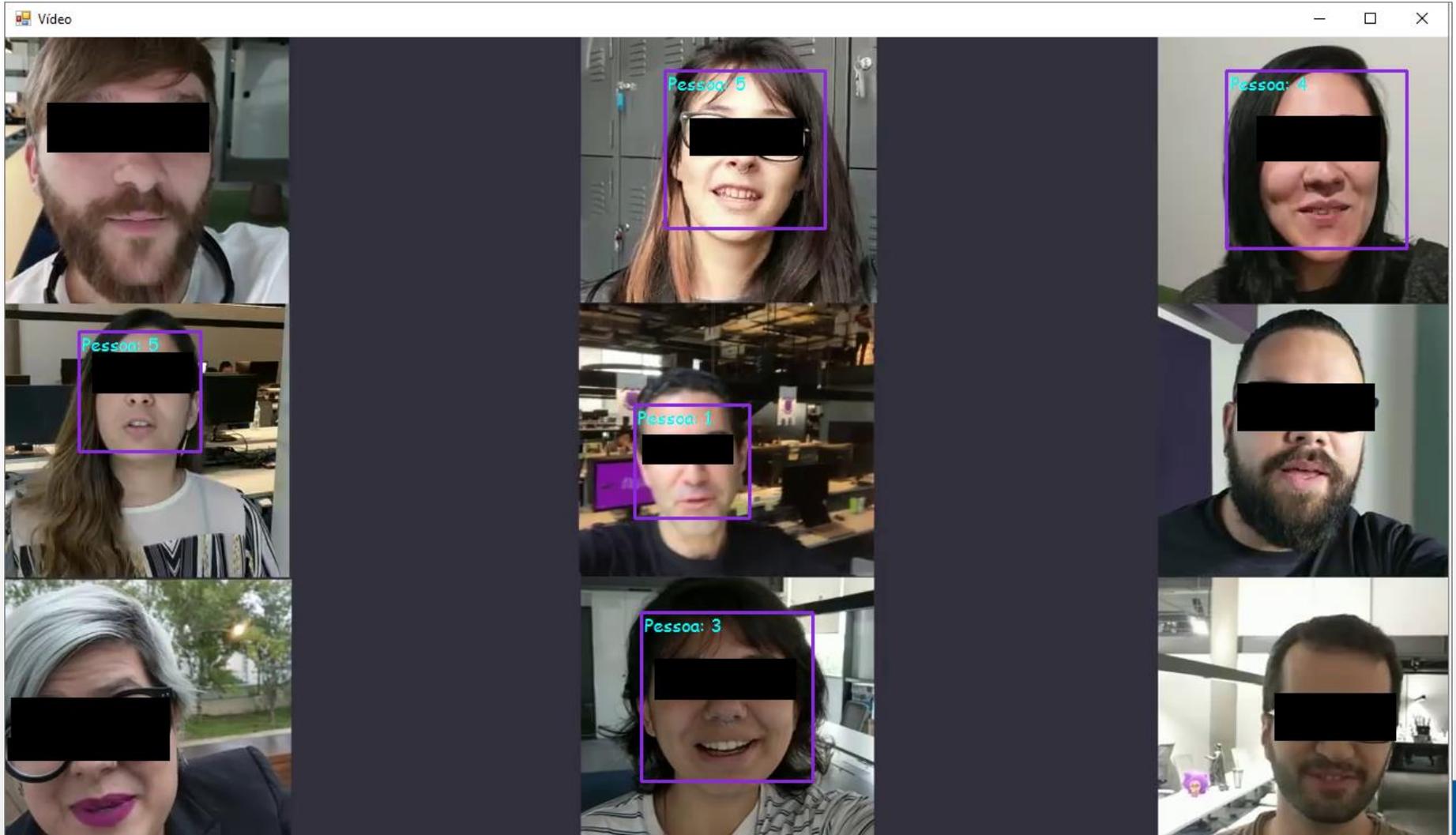
Buttons: "Aplicar", "Limpar"

Top-right panel: Large image of a man's face with a black redaction bar over the eyes.

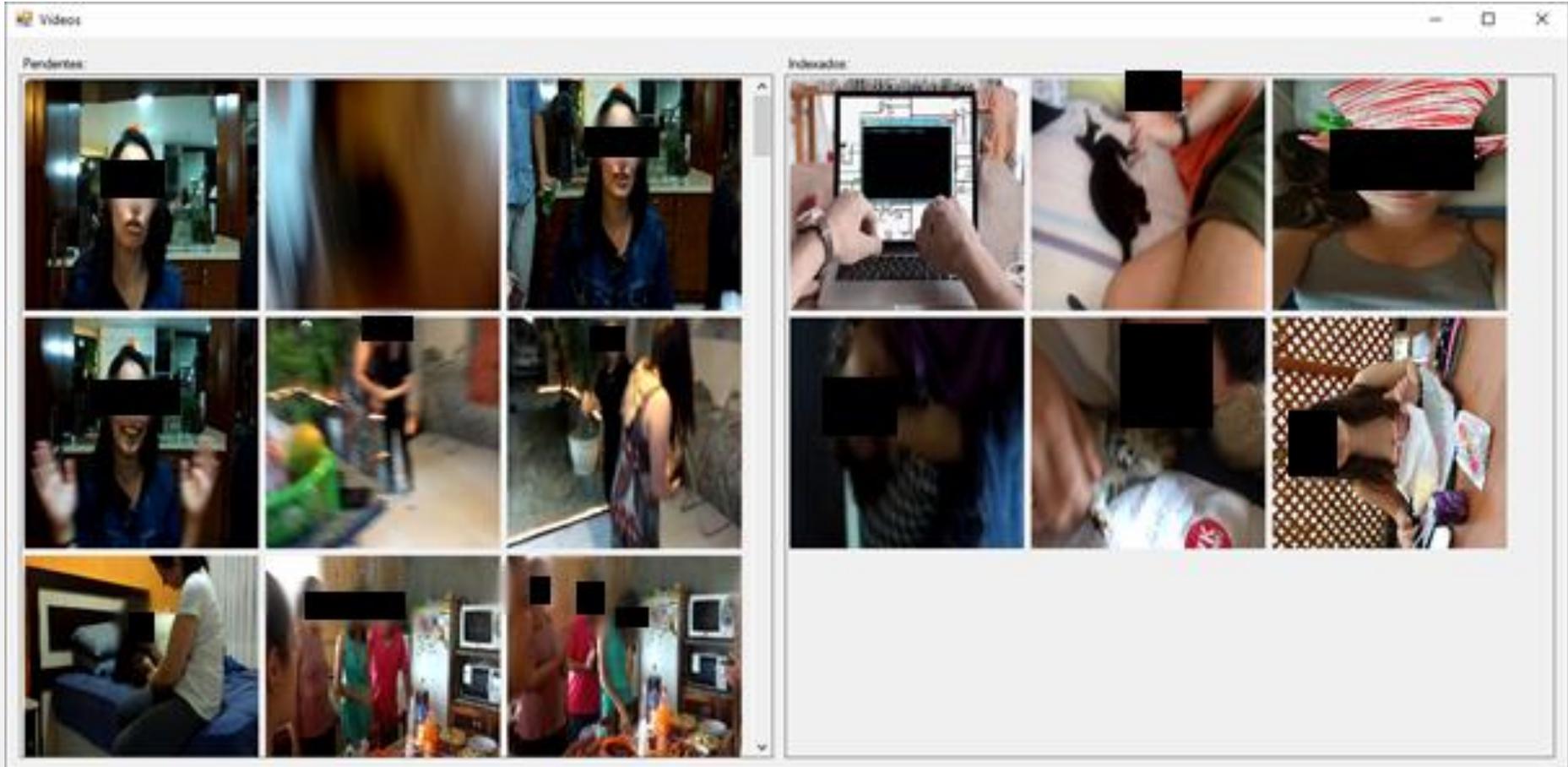
Bottom-left panel: Grid of 6 smaller face images with black redaction bars over the eyes.

Bottom-right panel: "Pessoa: 8"
"Clones:" 
"Videos:" 

Tela de Vídeo



Tela de Vídeos



Resultados e Discussões

- Experiência do usuário (5)
 - Vídeo fornecido
 - Modo livre
- Experiência do desenvolvedor (2)
 - Biblioteca
- Análise de memória e performance
 - Tempo
 - Resolução

Comparação com Correlatos

Trabalhos	Koch (2012)	Jiang et al. (2016)	Video Indexer (2017)	Trabalho desenvolvido (2018)
Características				
ferramenta reaproveitável (API)	Não	Não	Sim	Sim
processamento distribuído	Sim	Não	Sim	Não
efetua identificação	Sim	Não	Sim	Sim
realiza detecção facial em tempo real	Sim	Sim	Não	Sim
permite marcar/identificar uma face durante a execução/visualização do vídeo	Não	Não	Não	Não

Conclusões e Sugestões

- Conclusões
 - Objetivos
 - Usuário marcar durante visualização
 - Taxa de acerto do algoritmo
 - Biblioteca
- Sugestões
 - Retreinamento da detecção (XML OpenCV)
 - Algoritmo mais assertivo
 - Detecção pela GPU (CUDA)

Demonstração

FERRAMENTA DE INDEXAÇÃO DE FACES HUMANAS EM VÍDEOS

Aluno: Leonardo Leal Oliveira

Orientador: Dalton Solano dos Reis