**Programa de Reconhecimento Facial Escolar (Recog)**

Pedro José Berbert Bruno, pedrojbbruno@gmail.com¹

Mateus Freitas, mateusfreitas.silveira@gmail.com2

Igor Martins Silverio, igor.martins.silverio@gmail.com3

André Luís dos Reis Gomes de Carvalho, andre@cotuca.unicamp.br4

1Colegio Ténico de Campinas - COTUCA, R. Jorge Fiqueiredo Correa, 735- Campinas– SP

2Colegio Ténico de Campinas - COTUCA, R. Jorge Fiqueiredo Correa, 735- Campinas– SP

3Colegio Ténico de Campinas - COTUCA, R. Jorge Fiqueiredo Correa, 735- Campinas– SP

4Colegio Ténico de Campinas - COTUCA, R. Jorge Fiqueiredo Correa, 735- Campinas– SP

Resumo: *O programa de reconhecimento facial escolar (Recog) foi criado com o intuito de testar a capacidade de aplicação do OpenCV e do Algoritmo de Rosetta em um sistema de médio porte. Visa também facilitar a entrada e controle de pessoas no ambiente escolar. O sistema consiste de três aplicações conectadas a um banco de dados Microsoft SQL Server, desenvolvidos em Java e Arduino.*

***Palavras-chave:*** *Reconhecimento facial, Opencv, Acessibilidade, SQL Server, Java****.***

1. **INTRODUÇÃO**

Sistemas de segurança ao redor do mundo estão avançando diariamente, e novas maneiras de incrementá-la têm surgido diariamente, descartando as anteriores, agora obsoletas. Uma técnica muito utilizada é a biometria, em que se identifica a digital do individuo para lhe garantir a entrada ao estabelecimento ou ao sistema em questão. Porem, poucos destes fazem uso do reconhecimento facial em suas funcionalidades, ainda que o mesmo esteja numa curva ascendente de uso.

Devido esse fato, buscamos desenvolver uma aplicação na qual, não apenas detectamos o rosto, como também o reconhecemos, liberando ou não o acesso da pessoa ao estabelecimento. Realizamos o projeto em Java utilizando a biblioteca open source OpenCV 3.0[1] – para a detecção facial -; um algoritmo de diferenciação RGB, denominado algoritmo de Rosetta[2], para o reconhecimento; e o Microsoft SQL Server para armazenamento das imagens e dados.

1. **FUNDAMENTAÇÃO TEÓRICA**

Neste capitulo serão apresentados os fundamentos e ferramentas utilizadas para o desenvolvimento do projeto. Dentre elas estão:

* 1. **OpenCV**

A biblioteca OpenCV é encontrada disponível nas linguagens C / C++, Python e Java, e é através dela que capturamos a imagem vinda da câmera e encontramos os rostos. Eis algumas funções e classes únicas do OpenCV:

* Classe VideoCapture – classe voltada para a captura de imagems.
* faceDetection.detectMultiScale – detecta rostos na imagem fornecida
* Classe MatOfRect – classes com uma matriz de imagems usada para o faceDetection
  1. **Banco de Dados SQL Server**

SQL Server é Sistema Gerenciador de Banco de dados (SGBD) feito pela Microsoft. Este é um dos mais utilizados e, através dele, é possível criar tabelas, as quais são capazes de armazenar e gerenciar dados.

Na Figura 2 é apresentado o *Managerment Studio* ,que é a ferramenta de gerenciamento e visualização de dados armazenados na Base de dados.

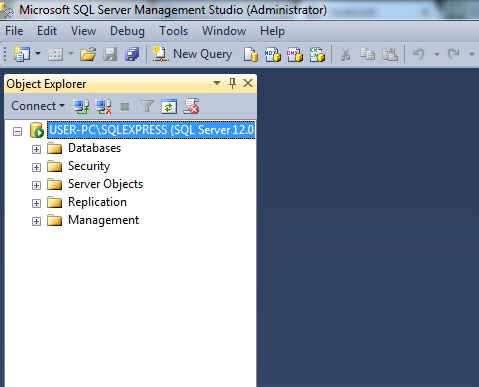


Figura 2 - Managerment Studio

* 1. **– Linguagem de Programação Java**

A linguagem de programação Java foi criada com o intuito de ser uma linguagem orientada a objetos, por James Gosling e sua equipe, com o nome original de Oak (Carvalho). Com o surgimento da internet, ocorreu a disseminação do Java, na qual ele ganhou mais e mais adeptos, atingindo a marca de três milhões de desenvolvedores em 2004.

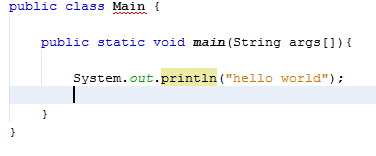


Figura 3 – Hello World

A figura 3 mostra um exemplo do clássico “hello word”, com sua devida main. A linguagem de programação Java é completamente voltada para a orientação a objetos.

1. **RESULTADOS**

Tendo como suporte as bases fundamentadas anteriormente, desenvolvemos uma aplicação para o cadastro e reconhecimento de pessoas no sistema.

* 1. **Cadastro de usuários**

O cadastro e manutenção de usuários são realizados através do Programa Cadastro, no qual o administrador fotografa e visualiza as fotos do indivíduo, faz pesquisas e gestão de dados - como as fotos ou nome do individuo. Para o cadastro e gestão são necessários somente a conexão com SQL Server e uma câmera conectada ao computador.

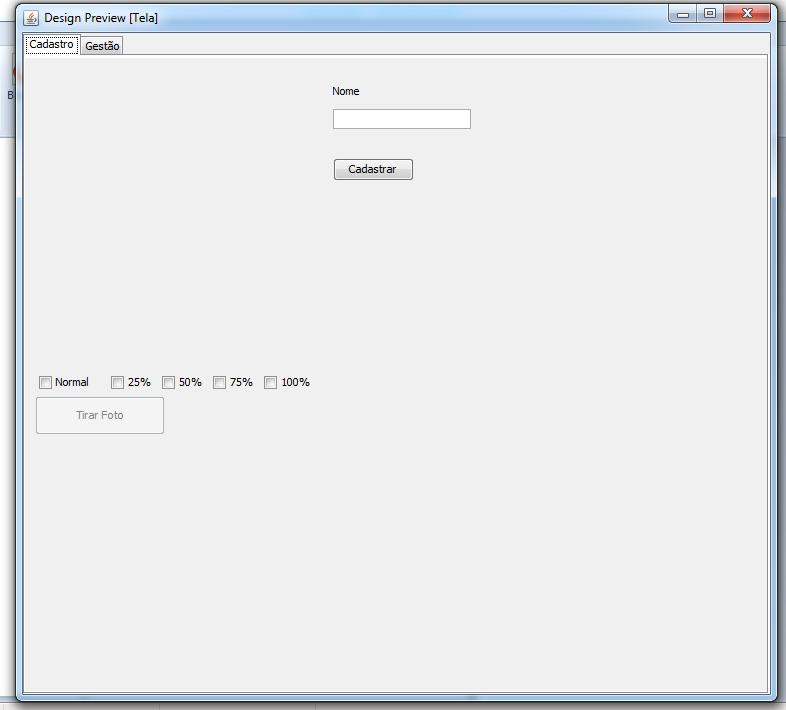


Figura 4 – Tela de cadastro

* 1. **Reconhecimento facial e liberação de catraca.**

Com a câmera devidamente posicionada, o individuo deverá permanecer no campo de visão da mesma, a certa distancia, para disparar o sistema de reconhecimento e ter sua entrada liberada, caso esteja cadastrado. Nos testes, a precisão foi de 85% de acertos.

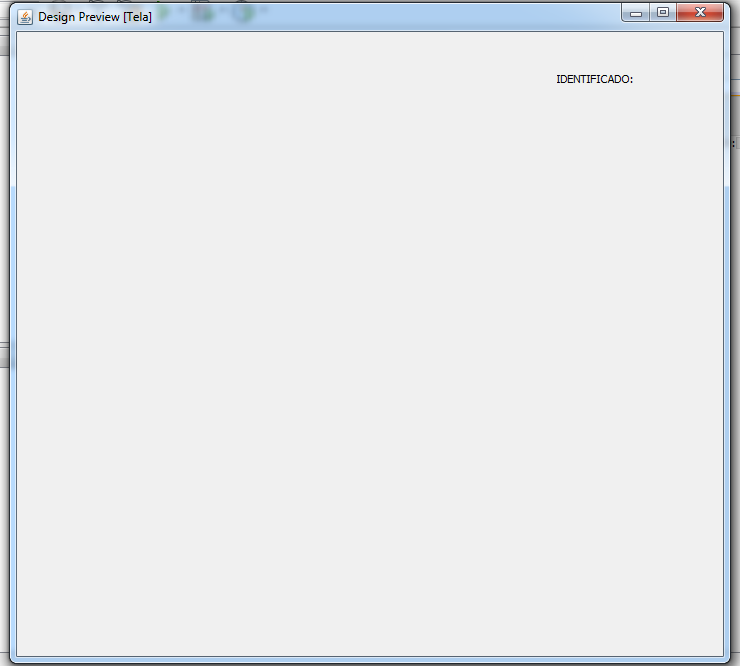


Figura 5 – Tela de reconhecimento

1. **AGRADECIMENTOS**

O desenvolvimento da aplicação foi possível graças ao time de desenvolvimento do OpenCV e aos criadores do Rosetta Code, no qual pudemos encontrar exemplos de código para detecção e diferenciação de imagens. Também agradecemos ao professor André Carvalho pela ajuda e orientação em relação ao projeto.

1. **BIBLIOGRAFIA**

[1] OpenCV, Disponível em: <http://opencv.org/>. Acesso em: Agosto 2015.

[2] Percentage difference between images. Rosetta Code. Disponível em: <http://rosettacode.org/wiki/Percentage\_difference\_between\_images>. Acesso em: Agosto 2015.