



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO



IFMT *CAMPUS* PONTES E LACERDA (FRONTEIRA OESTE)

RELATÓRIO FINAL

PROJETO: VISÃO COMPUTACIONAL NA SALA DE
AULA: REGISTRO DE FREQUÊNCIA ESCOLAR
USANDO RECONHECIMENTO FACIAL EM IMAGENS

EDITAL IFMT Nº 036/2017

Coordenador do Projeto: Bernardo Janko Gonçalves Biesseck

Aluno(s) Bolsista(s): Leuri Rabelo Zanetti

Pontes e Lacerda-MT, 08 de fevereiro de 2019



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

1. Identificação

- a) Título do Projeto: Visão Computacional na Sala de Aula: Registro de Frequência Escolar Usando Reconhecimento Facial em Imagens
- b) Início e fim do Projeto: 01/07/2017 a 30/12/2018
- c) Nome do Orientador: Bernardo Janko Gonçalves Biesseck
- d) Nome do Aluno Bolsista: Leuri Rabelo Zanetti
- e) Instituição/Campus: Instituto Federal de Mato Grosso *campus* Pontes e Lacerda

2. Resumo do Projeto

Este projeto é parte integrante do desenvolvimento de um aplicativo móvel chamado CapFace para registro automático de frequência escolar (chamada) através de reconhecimento facial em sala de aula. O objetivo geral era desenvolver o servidor CapFace-Server, composto pelo módulo de reconhecimento facial e pelo módulo de lançamento de presenças e faltas no Q-Acadêmico. A primeira estratégia para implementar o CapFace-Server foi utilizar o Servidor FTP ProFTPD [1], dado que ele é um *software* livre e possui um módulo chamado *mod_exec* que permite criar *triggers* para disparar processos automaticamente quando algum evento de interesse acontece. No entanto, devido a algumas limitações do *mod_exec*, optamos por fazer o desenvolvimento do servidor diretamente em Java, utilizando as classes Socket e ServerSocket para fazer a comunicação com os clientes. O módulo de reconhecimento facial foi desenvolvido em linguagem Python utilizando o algoritmo *Viola-Jones* [6] e a CNN *FaceNet* [7]. Já o módulo de lançamento da chamada foi feito em Java através das bibliotecas *Selenium* [4] e *JSON-simple* [5]. Após a conclusão deste projeto o CapFace-Server é capaz de realizar as tarefas para o qual foi desenvolvido. Ele recebe as imagens capturadas pelo professor através do aplicativo CapFace, reconhece as faces dos estudantes, gera uma lista de presença em um arquivo de texto no formato JSON e efetua o lançamento das presenças e faltas no diário da disciplina do respectivo professor.

Palavras-chave: aplicativo, CapFace-Server, frequência escolar, reconhecimento facial.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

3. Metodologia ou Material e Métodos

A primeira etapa deste projeto foi buscar na internet Servidores FTP gratuitos que permitissem a conexão de múltiplos usuários e também a criação de *triggers* para executar o módulo de reconhecimento facial automaticamente quando o professor realizasse o envio de imagens. Depois de analisar algumas possibilidades decidimos utilizar o servidor ProFTPD [1], dado que ele é um software livre e possui um módulo chamado *mod_exec* que permite fazer o que precisávamos. No entanto, devido a algumas limitações do *mod_exec* ao acessar arquivos e executar programas externos, decidimos implementar o CapFace-Server diretamente em linguagem Java. Para isso foram utilizadas as seguintes classes do pacote JDK [8]:

- *Socket* e *ServerSocket*: permitem realizar a comunicação em rede com o aplicativo CapFace;
- *DataInputStream*: permite receber arquivos através de uma conexão em rede;
- *FileOutputStream*: permite escrever (salvar) arquivos em disco.

O módulo de reconhecimento facial foi desenvolvido em linguagem Python através das bibliotecas *face_recognition* [2] e *OpenCV* [3]. Este módulo é responsável por detectar e reconhecer as faces dos estudantes presentes nas imagens capturadas pelo professor em sala de aula. Depois disso ele gerar um arquivo de texto no formato JSON contendo a lista de presença da respectiva aula. A detecção das faces foi feita através do algoritmo Viola-Jones [6] e o reconhecimento através da CNN *FaceNet* [7]. Para cada face a *FaceNet* [7] gera um vetor de características de 128 floats, que é comparado com as faces da base de dados através da distância euclidiana. A face que tiver a menor distância é considerada como reconhecida.

Para desenvolver o módulo de lançamento de chamada foi necessário realizar um treinamento com o estudante bolsista sobre a Linguagem de Programação Java. Ele teve que aprender a sintaxe básica para criar classes e objetos simples e também algumas classes padrão do JDK [8], como *String* e *ArrayList*. Utilizamos a IDE Eclipse como ferramenta de estudo e desenvolvimento. Depois disso foi necessário baixar e estudar as bibliotecas *Selenium* [4] e *JSON-simple* [5] para realizar o lançamento da frequência escolar de maneira automática no Q-Acadêmico. Com ela é possível criar programas que executam e controlam navegadores como o Google Chrome e o Mozilla Firefox e os fazem acessar páginas web através de cliques virtuais. É uma ferramenta muito poderosa que utilizamos para fazer o nosso servidor acessar o Q-Acadêmico com as credenciais do professor.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

4. Resultados e Discussão

O programa desenvolvido CapFace-Server funciona executando repetidamente os seguintes passos:

1. Repita indefinidamente:
2. Aguarda conexão TCP do aplicativo CapFace na porta 3322;
3. Recebe do aplicativo CapFace os dados: *username*, *filename*, *filesize*;
4. Recebe o arquivo zipado com as imagens capturadas em sala de aula pelo professor;
5. Descompacta o arquivo recebido e gera um novo diretório contendo as imagens;
6. Inicializa o módulo de reconhecimento facial;
7. Inicializa o módulo de lançamento de frequência escolar;
8. Fim_Repita

Após se conectar ao servidor (linha 2) o aplicativo CapFace envia os dados *username*, *filename* e *filesize* (linha 3), utilizados para gerenciar as imagens e realizar o seu armazenamento no diretório adequado. Feito isso, as imagens são enviadas em um arquivo compactado no formato ZIP (linha 4), o qual é descompactado em seguida (linha 5), permitindo a leitura das imagens em formato JPG ou PNG.

A seguir o CapFace-Server inicializa o módulo de reconhecimento facial (linha 6) executando a função *inicializarModuloDeReconhecimentoFacial()* apresentada a seguir:

```
1. public void inicializarModuloDeReconhecimentoFacial() throws IOException {
2.     String pythonCommand = "python3 recognizeFaces.py " + dirRegistroAula + " inicial.json";
3.     Process p = Runtime.getRuntime().exec(pythonCommand);
4.     BufferedReader stdInput = new BufferedReader(new InputStreamReader(p.getErrorStream()));
5.
6.     String s;
7.     while ((s = stdInput.readLine()) != null) { // exibe as saídas impressas pelo módulo de
8.         System.out.println(s);                // de reconhecimento facial
9.     }
10. }
```

4.1 Módulo de Reconhecimento Facial

O módulo de reconhecimento facial têm seu funcionamento baseado em 2 grupos de imagens: as imagens de base e as imagens de teste. As imagens de base são aquelas já capturadas pela secretaria escolar ou setor pedagógico quando o estudante realiza a matrícula na instituição. Elas são individuais e permitem que o sistema extraia as características de cada estudante separadamente. A Figura 1 apresenta algumas imagens de base de estudantes do 2º ano Vespertino do Curso Técnico em Informática Integrado ao Ensino Médio.



Figura 1: Imagens de base de alguns estudantes do 2º ano Vespertino do Curso Técnico em Informática Integrado ao Ensino Médio.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

As imagens de teste são aquelas que o professor captura em sala de aula através do aplicativo CapFace instalado em seu *smartphone*. Elas são utilizadas para verificar quais estudantes estão presentes e depois realizar a chamada. Após o envio das imagens as faces existentes são detectadas através do algoritmo Viola-Jones [6] e recortadas automaticamente para que não seja necessário processar as imagens inteiras na próxima etapa. Em seguida as características de cada face, extraídas com a CNN FaceNet [7], são comparadas com as características das imagens de base para fazer o reconhecimento. A última etapa consiste em gerar um arquivo de texto no formato JSON contendo a quantidade de faltas de cada estudante. Este arquivo é utilizado pelo módulo de lançamento de chamada. A Figura 2 apresenta uma representação do processo de reconhecimento facial e a Figura 3 um exemplo do arquivo JSON gerado para cada aula.

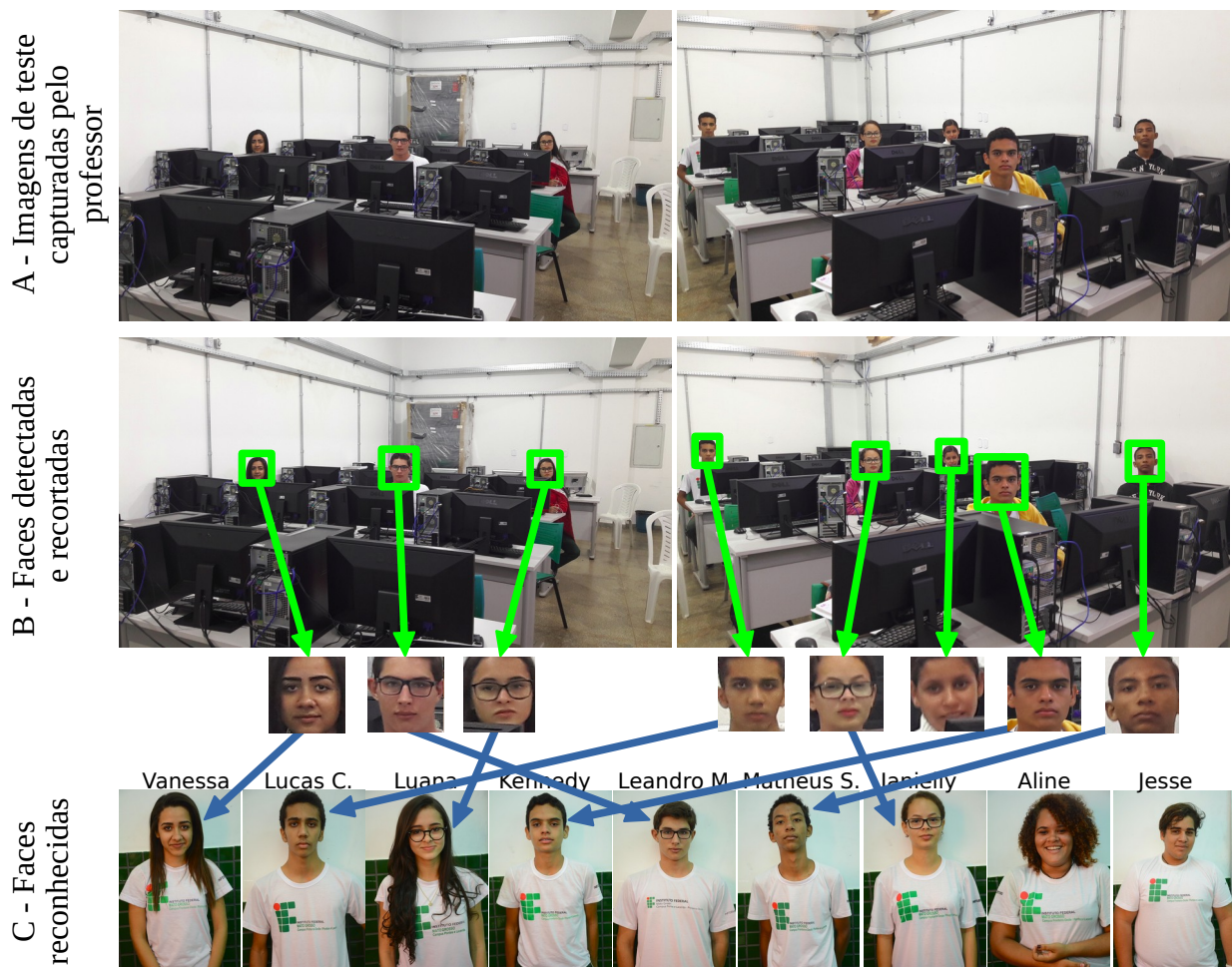


Figura 2: processo de identificação dos estudantes executado pelo módulo de reconhecimento facial. A) Imagens de teste capturadas pelo professor em um laboratório de informática durante a aula; B) Faces detectadas e recortadas automaticamente; C) Faces reconhecidas através da comparação com os rostos das imagens de base.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

```
{
  "Disciplina": "CASTERD.086 - Administração de Redes (72H/85H)",
  "Codigo": "158611",
  "Professor": "Professor Ficticio",
  "Turma": "2Ano_Informatica_Vespertino",
  "data da aula": "06/04/2018",
  "Horario de inicio": "07:00",
  "Horario de fim": "07:50",
  "quantidade de aulas": "1",
  "Pergunta": "Bimestral",
  "Bimestre": "1",
  "Conteudo": "configuração de endereços ipv4",
  "imgFiles": [
    "20180606_153246.jpg",
    "20180606_153249.jpg"
  ],
  "lista de alunos": [
    "Ana Vitoria",
    "Leandro Marcolin",
    "Luana",
    "Vanessa",
  ],
  "Faltas": [
    "1",
    "0",
    "0",
    "0"
  ]
}
```

Figura 3: Arquivo de texto no formato JSON gerado pelo módulo de reconhecimento facial. Neste arquivo são anotadas as faltas de todos os estudantes da turma.

4.2 Módulo de Lançamento de Chamada

O módulo de lançamento de chamada é inicializado pelo CapFace-Server após o encerramento do módulo de reconhecimento facial. Ele carrega os dados do arquivo JSON apresentado na Figura 3 e acessa o Q-Acadêmico automaticamente através dos navegadores Google Chrome ou Mozilla Firefox para lançar a chamada da respectiva aula.

A página do Q-Acadêmico foi construída com as linguagens HTML e JavaScript, assim é possível localizar através do código-fonte as *tags* dos links correspondentes a cada disciplina e também dos campos de inserção dos dados da aula (data da aula, hora de início, hora de fim, faltas dos estudantes, conteúdo, etc.). A Figura 4 apresenta uma visualização da página do Q-Acadêmico com as *tags* HTML de cada componente.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

```
<input type="text" name="DT_AULA_MINISTRADA"
size="10" id="idDT_AULA_MINISTRADA" maxlength=""
onkeypress="maskData(window.event.keyCode, this);"
class="data maskData" tabindex="1" style="width: 75px;">
```

```
<input name="HORARIO_INI" type="text" class="numero
maskHora" tabindex="2" size="5" value="" maxlength="5"
autocomplete="off">
```

```
<input name="HORARIO_FIN" type="text"
class="numero maskHora" tabindex="2"
size="5" value="" maxlength="5"
autocomplete="off">
```

```
<input name="N_AULAS" type="text"
class="numero" tabindex="2" size="2"
value="" maxlength="2"
onfocusout="ValidaNumero(this)">
```

```
<input name="N_FALTAS_111612"
type="text" class="numero maskNumero
faltas" tabindex="3" size="2" value=""
maxlength="2"
onblur="ValidaNumero(this);ValidaMaximo(t
his)">
```

```
<input type="submit" tabindex="25"
class="botao" value="Inserir">
```

```
<textarea name="CONTEUDO" cols="50"
rows="4" style="height:200;width:100%;"
tabindex="26"></textarea>
```

Mês e ano:		Inserir		7/2018							8/2018						
Dia:		Total		26	27	2	3	16	17	20	23	24					
Horário de Início:				07:00	08:30	07:00	09:50	07:00	09:50	07:50	07:00	09:50					
Horário Final:				08:39	10:39	08:39	10:39	08:39	10:39	09:59	08:39	10:39					
Aulas Dadas/Previstas:		32/32		2	1	2	1	2	1	2	2	1					
Matricula	Nome	Apropriar	Inserir														
2016107243010237	Adryan Lucas Soares de Barros	6															
2015107243010428	Allan Kevin Tawan Freitas Inacio																
2016107243010199	Antonio Krauze Feltosa	1															
2016107243010334	Rafael Soldeira Santos	1															
Alterar/Excluir:		Apropriar	Inserir														
Dia:		Total		26	27	2	3	16	17	20	23	24					
Mês e ano:				7/2018							8/2018						

Conteúdo ministrado:

Últimos conteúdos digitados:

Clique em <<<< para colar o texto. | pesquisar...

Clique aqui para carregar o conteúdo das últimas aulas ministradas

Mostrar 10 | Mostrar 20 | Mostrar Todos

Figura 4: Visualização do código-fonte HTML dos elementos que compõem a página do Q-Acadêmico. Cada elemento visualizado pelo usuário possui uma tag (comando) correspondente que permite

5. Conclusões

O desenvolvimento deste projeto foi de grande importância para a validação do aplicativo CapFace, cujo objetivo é facilitar o trabalho do professor ao realizar a chamada. Os usuários normalmente não sabem, mas por trás dos aplicativos que utilizamos no *smartphone* existem servidores ligados 24 horas por dia, 7 dias por semana, que recebem e processam os dados inseridos. Através deste projeto pudemos testar várias possibilidades de implementação do CapFace-Server e agora podemos continuar o seu desenvolvimento para melhorar a ferramenta proposta.

6. Anexos

Nenhum anexo.

7. Referências Bibliográficas

- [1] ProFTPD Project. ProFTPD - Highly configurable GPL-licensed FTP server software, 2017. Disponível em: <http://www.proftpd.org/>
- [2] Geitgey, A. Face Recognition - The world's simplest facial recognition api for Python and the command line, 2018. Disponível em: https://github.com/ageitgey/face_recognition



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

- [3] OpenCV team. OpenCV - Open Source Computer Vision Library, 2018. *Disponível em:* <https://opencv.org/>
- [4] Badle, S.; Bakken, J.; Barantsev, A.; Beans, E. Selenium Browser Automation, 2018. *Disponível em:* <https://www.seleniumhq.org/download/>
- [5] Google Company. JSON-simple, 2012. *Disponível em:* <https://code.google.com/archive/p/json-simple/downloads>
- [6] Viola, P.; Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features. In CVPR, 2001. *Disponível em:* <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>
- [7] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015. *Disponível em:* <https://arxiv.org/pdf/1503.03832.pdf>
- [8] Oracle Company. JDK – Java Development Kit, 2018. *Disponível em:* <https://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>

8. Dificuldades Encontradas

As principais dificuldades encontradas são referentes à infraestrutura da instituição. Não temos ainda laboratórios específicos para projetos de pesquisa, nosso trabalho vem sendo realizado de maneira compartilhada com outros professores no Laboratório de Hardware. Este fato não é um impeditivo mas é uma das dificuldades que temos.

9. Parecer do Bolsista quanto ao desenvolvimento do Projeto

No início o professor Bernardo realizou uma reunião para explicar a organização do projeto e sua finalidade, na qual foram divididas as atividades para cada integrante.

A minha primeira tarefa foi encontrar o melhor servidor FTP para nossa as nossas necessidades. Para isso, foram feitos testes com os programas *vsftpd*, *wu-ftp* e *proftpd*. O *proftpd* se sobressaiu em relação aos outros devido à disponibilidade de vários módulos para diferentes funções. O módulo utilizado foi o *mod_exec*, que permite configurar scripts com *triggers* quando algum evento acontece. Com o *mod_exec* ativado o servidor FTP tem várias funcionalidades disponíveis, na qual utilizamos o *OnCommand* e o *OnConnect* com a finalidade de criar pastas e mover arquivos quando uma conexão é realizada com o servidor. Para isso acontecer primeiro criamos um *script* em linguagem Shell Script e depois configuramos a sua execução no arquivo *proftpd.conf* dentro do servidor. Após essa



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO
PRÓ-REITORIA DE PESQUISA E INOVAÇÃO
DIRETORIA DE PESQUISA E INOVAÇÃO

configuração foi estudado como criar usuários dentro do servidor, de modo que eles não tivessem acesso ao diretório raiz (“/”).

Após compreender o funcionamento do servidor *proftpd* iniciei os estudos da linguagem HTML para entender os elementos que formam a página do Q-Acadêmico. Foram feitas várias atividades para sobre este assunto para habituar à estrutura da linguagem. Também foi feita uma demonstração de como requisitar o código HTML de uma página através do terminal de comandos do sistema operacional Ubuntu.

Em seguida foi feita uma apresentação da linguagem Java e da IDE Eclipse para implementação e execução de programas. Para facilitar o aprendizado foram desenvolvidos códigos simples com operações matemáticas. Com o básico da linguagem Java dominado comecei a utilizar a biblioteca Selenium, que é responsável pela automatização das tarefas que serão executadas no *browser*. Quando as duas tarefas foram concluídas (configuração do servidor *proftpd* e implementação do módulo de lançamento de chamada) fizemos a junção dos serviços.

Esse projeto de pesquisa foi de suma importância na minha evolução acadêmica, na qual me permitiu participar de evento onde eu tive contatos que podem me dar mais oportunidade de aprofundar na minha área acadêmica. Além de que, por se tratar de um projeto de pesquisa eu tive que ter a muita iniciativa para tomar as decisões mais convenientes segundo meu ponto de vista. Vale lembrar também que tive muito aprendizado agregado a minha pessoa, na qual poderá me ajudar futuramente, já que o projeto é no ramo da informática.

Uma dificuldade encontra foi que a nossa ideia é praticamente inovadora, ou seja, não tem muito conteúdo na internet na qual eu possa usar como base/exemplo para aprender ou solucionar algum problema encontrado no projeto, sendo necessário descobrir e juntar novas ferramentas para que atendesse a nossa necessidade. Também vale ressaltar que não contamos com uma infraestrutura para projetos de pesquisa, muitas das vezes tive de levar o meu computador pessoal para que o projeto pudesse dar andamento.

Pontes e Lacerda-MT, ____ de _____ de 2019.

Bernardo Janko Gonçalves Biesseck
Professor de Informática do campus Pontes e Lacerda
Coordenador do Projeto

Leuri Rabelo Zanetti
Estudante do Curso Superior de Tec. em Redes
Bolsista do Projeto