

## Laboratório de Projetos III

Trabalho Final - The Eye

André Lopes Cançado e Sousa  
Lucas Eduardo Costa Franco  
Paulo Cirino Ribeiro Neto

06 de julho de 2017

### 1 Ideia do Projeto

O projeto inicial consistia no desenvolvimento de um aplicativo para dispositivos mobile que reconhecesse faces capturadas pela câmera, buscasse informações destas em redes sociais e apresentasse essas informações na tela em forma de um menu navegável que acompanhasse o movimento da face. O sistema, portanto, deveria possuir algumas funcionalidades bem desenvolvidas, como:

- Captura de Imagens;
- Detecção de Faces;
- Rastreamento de Faces em movimento;
- Reconhecimento de Faces;
- Apresentação das informações na tela;
- Navegação pelas informações;
- Comunicação cliente/servidor, devido ao fato de que dispositivos mobile apresentam baixa capacidade de processamento;
- Banco de dados para armazenamento das informações dos usuários.

Com o avanço do projeto, modificações foram feitas, e abortou-se a ideia de desenvolvimento para aplicações mobile e houveram problemas com a comunicação cliente/servidor, fazendo com que essa frente de trabalho fosse abortada.

Para conseguir atingir seus objetivos, o sistema projetado utiliza conceitos de *deep learning* e redes neurais para fazer a detecção e classificação das faces reconhecidas. O repositório contendo os arquivos utilizados durante a disciplina está disponível em [1].

### 2 O software

#### 2.1 Premissas de uso

Para que o sistema funcione corretamente, alguns softwares devem ser previamente instalados. Abaixo tem-se uma lista dos softwares necessários para a execução do código e como instalá-los:

- Python 2.7 ou superior;
- OpenCV;
- OpenFace;
- dLib;
- Torch;
- Numpy;
- SciKit Learn;

## 2.2 Detecção/Classificação de faces

O processo projetado para a detecção e classificação das faces reconhecidas pelo sistema se baseia principalmente em, utilizando as bibliotecas de reconhecimento facial do OpenFace, ser capaz de extrair informações de interesse das faces identificadas em tempo real e classificá-las.

## 2.3 Banco de dados

Para a aquisição das informações, ao invés de buscar diretamente em redes sociais, optou-se por criar um banco de dados com todos os alunos da disciplina, onde as informações de cada aluno são armazenadas em arquivos JSON (*JavaScript Object Notation*), que possuem a estrutura mostrada na figura 1. Assim, foi implementado um script para busca das informações nos arquivos JSON, que devem estar na mesma pasta que o código, bem como seguir o formato pré-estabelecido.

```
{
  "personal_informations": {
    "name": "unknown",
    "age": "99",
    "nationality": "Brazilian",
    "marital_status": "single"
  },
  "professional_informations": {
    "role": "none",
    "company": "none",
    "since": "2017",
    "until": "2017"
  },
  "academic_informations": {
    "major": "none",
    "university": "none",
    "year_of_graduation": "2017"
  }
}
```

Figure 1: Exemplo de JSON utilizado no banco de dados

As informações pessoais utilizadas foram o nome, idade, nacionalidade e estado civil do usuário. As informações profissionais utilizadas foram o cargo, companhia em que trabalha e datas de início e término, se houver, do contrato. Da mesma forma, as informações acadêmicas utilizadas foram a graduação, a universidade e o ano de conclusão do curso.

Em paralelo, foram tiradas fotos dos alunos da disciplina para que a rede neural do sistema fosse treinada e capaz de reconhecer os usuários do sistema.

## 2.4 Classificação

Para o treinamento da rede neural utilizada, foram pesquisados rostos de diversos famosos, de forma que o algoritmo seja capaz de reconhecer as faces da câmera. Posteriormente, foram tiradas fotos dos alunos da disciplina, de forma que o algoritmo fosse capaz de reconhecê-los como usuários do sistema. Para que a classificação acompanhasse o movimento das faces, utilizou-se como estratégia considerar uma região muito maior do que a face, de forma que os movimentos, contínuos e bem comportados, estejam sempre enquadrados nessa região de interesse. Para a classificação, utilizou-se a biblioteca OpenFace Lib.

## 3 *Building and Running*

### 3.1 Instalação de Softwares Necessários

Para ser capaz de reproduzir o programa são necessários alguns softwares:

- python 2.7
- opencv 2.4.11
- torch
- dlib
- openface

Além disso são necessárias os seguintes pacotes do python :

- numpy
- pandas
- scipy
- scikit-learn
- scikit-image

E as seguintes dependências do torch:

- dpnn
- nn
- optim
- csvigo
- fblualib
- tds
- torchx
- optnet

Um tutorial complete de como instalar os itens acima pode ser encontrado em <https://cmusatyalab.github.io/openface/setup/> .

## 4 Reproduzindo o Softwares

As imagens com as quais o modelo de classificação foi treinado e as informações pessoais dessas pessoas foram retiradas da repositório por motivos de privacidade.

Porém para retreinar um modelo qualquer simplesmente rode a seguinte linha de código :

```
$./models/classifier.py infer ./models/openface/celeb-classifier.nn4.small2.
```

O resultado disso será um arquivo *.pkl* que é seu modelo treinado.

Um diretório com um arquivo *.json* com as informações pessoais de cada um dos usuários é necessário, esses arquivos devem ter os mesmos nomes das pessoas informadas ao treinamento do classificador.

Por fim resta apenas rodar o arquivo *EYE.py*.

## 5 Considerações Finais

O objetivo da disciplina foi cumprido com sucesso, embora nem todas as propostas para o aplicativo tenham sido cumpridas. Todos os alunos participaram de todas as etapas de um projeto não trivial, desde a concepção da ideia até a entrega do produto, tendo aprimorado os processos de gerência de projetos e engenharia de software. O grupo gostaria de agradecer ao professor Bruno Adorno pelo empenho, forma como desenvolveu a disciplina e auxílio prestado durante o semestre letivo.

## References

- [1] Repositório GitHub utilizado pelos alunos para a disciplina. Disponível em: <https://github.com/PauloCirino/LabProjetosIII>