

**<< ENERGIA VISUAL EMOCIONAL (E.V.E. VERSÃO 2.0) -
A INTELIGÊNCIA ARTIFICIAL COMO FERRAMENTA DE INCLUSÃO DE
PESSOAS COM DEFICIÊNCIA VISUAL OU CEGUEIRA POR MEIO DO
RECONHECIMENTO FACIAL UTILIZANDO A FALA >>**

Nathalia Assunção das Chagas, João Pedro Gomes de Almeida

Robson Ferreira Lopes (Orientador)

Gema Galgani Rodrigues Bezerra (Coorientadora)

Rodrigo Campos Bortoletto (Coorientador)

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Guarulhos,
Guarulhos/SP.

Resumo

A comunicação entre interlocutores depende de diversos aspectos, como as palavras, o tom de voz, a fisionomia, entre outros que otimizam nosso potencial comunicativo. Por outro lado, a pessoa com deficiência visual ou cegueira pode ter diversas dificuldades na comunicação e na integração social devido à impossibilidade de reconhecer a face e as expressões fisionômicas de seu interlocutor. O projeto Energia Visual Emocional (E.V.E.) visa contribuir para a melhoria da qualidade de vida das pessoas com deficiência visual ou cegueira, consolidando sua autonomia comunicativa. Para isso, o projeto partiu de uma pesquisa bibliográfica e vem utilizando o método de engenharia a partir da prototipação. Iniciou-se pelo desenvolvimento de um protótipo de software com uma biblioteca capaz de identificar as características de uma face por meio de uma imagem. Usando técnicas de Aprendizagem Profunda (*Deep Learning*) é possível reconhecer a face em uma imagem e identificá-la por meio de áudio, possibilitando à pessoa com deficiência visual ou cegueira identificar quantas e quem são as pessoas presentes em uma imagem. Os próximos passos consistem em aprimorar a programação do código, identificando as ações fundamentais dos músculos individuais (*Action Units*) e suas combinações, extraindo as informações contidas e adaptando-as de forma rápida

e eficiente para serem utilizadas na comunicação de pessoas com deficiência visual ou cegueira.

Palavras-chave: Inclusão. Deficiência visual ou cegueira. Inteligência artificial. Reconhecimento de faces.

Esse projeto surgiu de um desafio proposto pelo Prof. Robson Ferreira Lopes na disciplina de Redes de Computadores, quando propôs substituir a avaliação convencional pela submissão de um projeto de pesquisa à Feira de Ciência e Engenharia de Guarulhos - FECEG 2019. De acordo com a proposta, a ferramenta para a solução do problema delineado no projeto deveria ser relacionada à Internet das Coisas (*Internet of Things*). Durante uma conversa informal com um de nossos colegas de turma, que é surdo-cego, encontramos o nosso tema e a questão que levantamos foi a dificuldade das pessoas com deficiência visual para detectar os sentimentos das pessoas. Criamos, então, o projeto Energia da Vibração Emocional, que consistia em transmitir vibrações das expressões da face por meio do Arduino. Em estudos preliminares apresentamos essa ideia como viável e com esse projeto participamos da FECEG e fomos premiados como melhor projeto de 2019, o que nos encaminhou para a FEBRACE. Na continuidade da pesquisa foi possível aprender sobre inteligência artificial e com isso surgiu uma nova ideia, a qual denominamos Energia Visual Emocional (E.V.E Versão 2.0), que é objeto de estudo desde então e cujos fundamentos teóricos serão apresentados a seguir

A ambiguidade é um elemento muito presente na comunicação de pessoas com deficiência visual ou cegueira (pessoas doravante mencionadas neste trabalho como deficientes visuais ou cegas), pois as interações entre interlocutores dependem de diversos aspectos, como as palavras, o tom de voz, a fisionomia, entre outros que otimizam nosso potencial comunicativo, e muitos desses aspectos são captáveis por meio de informações visuais.

Pires (2018), em artigo no qual discute estatísticas sobre interpretação da linguagem corporal atribuídas a Albert Mehrabian – que publicou em 1971 um livro intitulado *Silent Messages* (Mensagens silenciosas) – explica que o senso comum se apropriou de forma equivocada dos seus achados e a isso Pires denomina o mito de Mehrabian. Segundo Pires, “o raciocínio do professor Mehrabian é de que **as palavras, a voz e a linguagem corporal devem ser consistentes** umas com as outras durante a comunicação” (grifo do autor) e, no caso de haver inconsistências, “as expressões faciais têm impacto mais relevante” na comunicação dos sentimentos do que a forma

como se fala ou do que a própria mensagem. Assim, a impossibilidade de captar tais informações acaba restringindo a comunicação dos deficientes visuais, devido ao fato de não poderem interpretar a face de seu interlocutor.

Por isso destaca-se a importância do olhar para a acessibilidade, analisando dados no Brasil e no mundo: de acordo com a OMS (Organização Mundial da Saúde), existem 39 milhões de pessoas cegas no mundo todo e, só no Brasil, de acordo com o IBGE (Instituto Brasileiro de Geografia e Estatística), 6,5 milhões de habitantes apresentam deficiência visual, dentre os quais quase 700 mil são cegos.

Em artigo publicado no site da Fundação Dorina, Freitas (2018) comenta a afirmação da psicóloga Rita Lobo, segundo a qual a perda de visão é um grande gatilho para ansiedade, perda de vontade de viver e depressão, sendo semelhante à dor de perder um ente querido. Por isso, remover esse obstáculo na comunicação é um grande passo para reduzir as lacunas de uma vida sem visão, incluindo os deficientes visuais na sociedade e diminuindo os casos de isolamento e depressão. Sendo assim, nossa pesquisa procura responder: como possibilitar a uma pessoa cega identificar outras pessoas em uma imagem e saber quem está nela? É possível construir uma ferramenta que reconheça pessoas numa imagem e as identifique por meio de áudio? É possível dar autonomia aos deficientes visuais para saberem que pessoas estão em uma fotografia?

O objetivo é melhorar a autonomia das pessoas com deficiência visual usando inteligência artificial para identificar pessoas em uma imagem e reconhecê-las por meio de áudio, melhorando a sua qualidade de vida e facilitando o seu convívio com pessoas videntes. O projeto aplica elementos do *Deep Learning* com a ciência social aplicada.

Para isso, o projeto partiu de uma pesquisa bibliográfica e vem utilizando o método de engenharia a partir da prototipação.

O protótipo desenvolvido conta com o reconhecimento facial por meio de uma biblioteca denominada *face_recognition*, que usa conceitos de rede neural para identificar quantas e quem são as pessoas presentes em uma imagem e a biblioteca *Espeak*, que emite um som com o nome das pessoas encontradas na imagem.

Os resultados preliminares indicam que é bem possível desenvolver uma ferramenta capaz de auxiliar as pessoas com deficiência visual a terem uma maior autonomia nesse sentido.

Nas considerações finais são apresentadas as possibilidades de melhoria no projeto, bem como os próximos passos para sua continuação.

1. METODOLOGIA

a. Deep Learning

Deep Learning ou aprendizagem profunda é uma tecnologia que emprega algoritmos para realizar o processamento de dados, imitando o desempenho de raciocínio realizado pelo cérebro humano. Como hiperônimo de aprendizagem profunda, encontramos a Inteligência Artificial (IA), ramo da ciência da computação que propõe modelos de dispositivos que simulem a capacidade de raciocínio humano. O aprendizado de máquina baseia-se na ideia de que os computadores podem aprender

com dados a partir do reconhecimento de padrões e realizar tarefas específicas, produzindo decisões e resultados confiáveis, com a menor intervenção humana possível. No projeto escolhemos a biblioteca *Face Recognition*, que aprende sobre pontos da face por meio de imagens, o que permite identificar se há faces em uma imagem e registrar os pontos identificados em variáveis que podem ser usadas para traduzir pontos faciais em outras posições.

b. Rede Neural Convolucional

Convolutional Neural Network (CNN), cuja tradução literal é “Rede Neural Convolucional”, é um algoritmo do *Deep Learning* que funciona de modo semelhante aos neurônios de um cérebro humano. Enquanto a abordagem tradicional de computação utiliza uma série de blocos lógicos para executar uma determinada tarefa, Redes Neurais utilizam redes compostas por nós (os quais atuam como neurônios) e arestas (que por sua vez atuam como sinapses) para processar dados. As entradas então percorrem o sistema e uma série de saídas é gerada. Essas saídas são então comparadas com dados conhecidos.

A execução usa camadas de neurônios matemáticos para processar dados, compreender a fala humana e reconhecer objetos visualmente. Dessa tecnologia resulta uma série de camadas de processamento: o computador realiza etapas para tentar descobrir determinada informação. Apresentando a imagem de um objeto para um computador – um *input* (dados fornecidos) –, ele tentaria entendê-lo dividindo a análise desse objeto em diversas camadas de processamento chamadas de *Hidden Layers*. Usando várias camadas de processamento de dados não lineares, obtém-se uma representação complexa e abstrata dos dados de forma hierárquica, sem a intervenção humana no processo.

c. Python e a programação do protótipo

Segundo a python.org, Python é uma poderosa linguagem, fácil de aprender, pois possui estruturas de dados de alto nível, o que a torna mais eficiente com uma abordagem simples e efetiva de programação orientada a objetos.

d. Materiais e Métodos

Para programar o protótipo, foi necessária uma pesquisa que resultou na utilização de uma programação em python em conjunto com a biblioteca *Face Recognition* (escrita em linguagem python), que identifica características do rosto e é uma das ferramentas fundamentais usadas para o reconhecimento facial. Para viabilizar o uso da biblioteca, foram instalados os pacotes python3-numpy, python3-scipy, python3-matplotlib ipython3, python3-pandas, python3-sympy e python3-nose em uma máquina virtual com sistema operacional Linux e, depois, com o pip (ferramenta utilizada para baixar pacotes do repositório Python) fizemos a instalação da *face_recognition*. Além da biblioteca, foram utilizados módulos como dlib, que é um

moderno conjunto de algoritmos escritos em C++ para aprendizado de máquina, PIL.image e espeak (sintetizador de voz compacto de software de código aberto para inglês e outros idiomas).

Vários testes foram realizados até chegar ao protótipo usado neste projeto, em cujo repositório podem ser encontrados todos os códigos utilizados (<https://www.github.com/flrobson77/febrace>).

Na versão beta (Anexo 01), em primeira instância, são importadas a biblioteca *face_recognition* e as bibliotecas adicionais. Em seguida, são carregadas as imagens pré-registradas no banco de dados, como mostram os códigos que vão da linha 08 até a 11, o que permite ao programa comparar as imagens recebidas (linha 14 até a 17) com as que ele já conhece (Anexo 02).

Utilizando o algoritmo da biblioteca, o programa utiliza uma rede neural para aprender sobre as imagens, armazenando os valores da rede em uma variável do tipo array (variável que mantém uma série de dados, geralmente do mesmo tipo e tamanho), sendo criado um outro array para armazenar o nome estabelecido para cada imagem e apresentar ao programa.

Para iniciar a execução do reconhecimento um arquivo é carregado (Anexo 01, linha 34) para teste de reconhecimento. Nas linhas 37 e 38 os métodos *face_locations* e *face_encoding* identificam respectivamente as faces presentes na imagem e a matriz de dados gerada pela rede neural do *face_recognition*. Na linha 40 a biblioteca PIL gera uma nova imagem de modo a não alterar a imagem original e, na linha 42, é criado o objeto retângulo que será impresso no novo arquivo com as faces identificadas. Nas linhas 46 a 49 é executado um laço for que identifica a quantidade de faces existentes na imagem e, em seguida, nas linhas 51 a 66, temos um outro laço for que compara as faces apresentadas na imagem de teste (Anexo 3) com as imagens que o software conhece (Anexo 2). As faces identificadas recebem um retângulo e, abaixo, um label com o nome da pessoa. Caso o programa não reconheça a face, será registrada a palavra “Desconhecido” no label da face, como indicado na linha 54 do código. No final, na linha 69, temos a principal função do sistema, que é a reprodução em áudio dos nomes das pessoas encontradas e a indicação das pessoas desconhecidas.

Para uma apresentação mais clara do que foi feito, utilizamos um comando que exibe o resultado do reconhecimento (Anexo 1, linha 73), sendo possível observar como o programa identifica cada face. Após os testes, a imagem é armazenada no diretório do programa (Anexo 1, linha 74).

2. RESULTADOS E DISCUSSÃO

Nesse momento o protótipo está cumprindo o objetivo proposto. O programa conseguiu identificar a quantidade de pessoas na foto e exibi-las separadamente, identificando cada uma delas por meio de áudio.

Durante nossos testes com outras imagens, percebemos que se as faces estiverem de perfil, podem não ser reconhecidas pelo algoritmo da *face_recognition*.

O protótipo mostrou-se eficiente quanto às técnicas de programação e uso da biblioteca, mas cabe salientar que essa ainda é a primeira fase do projeto que, nas fases seguintes, realizará novos testes para que o programa consiga identificar a face em diferentes posições, além de testes de reconhecimento da face em vídeo, o que permitirá testar o programa com pessoas deficientes visuais ou cegas, validando-o devidamente.

3. CONCLUSÕES

Entendemos que com a evolução do projeto E.V.E. será possível alcançar uma solução viável para atender às expectativas formuladas no objetivo, contribuindo para melhorar a socialização dos deficientes visuais e pessoas com cegueira.

No decorrer do desenvolvimento desse primeiro protótipo, foi possível aprender conceitos como *machine learning* e redes neurais, que são conhecimentos avançados na área de inteligência artificial. Aprendemos, ainda, a programar usando a linguagem python.

Para o futuro, nosso projeto deve avançar para o reconhecimento de faces em vídeo e câmeras. O código precisa evoluir em sua usabilidade para facilitar a inclusão de novas imagens que o software precisa aprender.

Para tanto, é preciso aprimorar a programação do código, o qual deverá permitir a identificação das ações fundamentais dos músculos individuais (*Action Units*) e suas combinações, extraindo as informações contidas e adaptando-as de forma rápida e eficiente para tornar possível o reconhecimento de imagens, por exemplo, usando máscaras.

Além disso, o projeto pretende, ainda, trazer futuramente o software para o dispositivo móvel, em formato de aplicativo, por meio do qual o deficiente visual poderá contar com uma descrição de imagens que potencializará o desenvolvimento e consolidação de sua autonomia comunicativa.

4. REFERÊNCIAS

FREITAS, Fernando. Setembro amarelo: precisamos falar sobre suicídio. 2018. Disponível em: <<https://www.fundacaodorina.org.br/blog/suicidio-setembro-amarelo/>>. Acesso em 02 jun. 2020.

GEITGEY, Adam. Face Recognition documentation. 2017. Disponível em: <<https://face-recognition.readthedocs.io/en/latest/index.html>>. Acesso em: 07 set. 2020.

PIRES, Sergio Fernandes Senna. O Mito de Mehrabian. **Instituto Brasileiro de Linguagem Corporal**. 2018. Disponível em: <<https://ibralc.com.br/o-mito-de-mehrabian/>>. Acesso em: 01 jun. 2020.

PYTHON Software Foundation. **Documentação Python 3.8.2** Disponível em: <<https://docs.python.org/pt-br/3/index.html>>. Acesso em: 01 jun. 2020.

ANEXOS

Anexo 01 - Código fonte da versão beta do programa E.V.E.

```
01 #Bibliotecas adicionais
02 import face_recognition
03 import os
04 from subprocess import call
05 from PIL import Image, ImageDraw
06
07 #Carregando banco de imagens
08 image_joel = face_recognition.load_image_file('./images/joel.jpg')
09 image_robson = face_recognition.load_image_file('./images/robson.jpg')
10 image_rodrigo = face_recognition.load_image_file('./images/rodrigo.jpg')
11 image_tarcisio = face_recognition.load_image_file('./images/tarcisio.jpg')
12
13 #Identificando as imagens do banco que o programa conhece
14 joel_face_encoding = face_recognition.face_encodings(image_joel)[0]
15 robson_face_encoding = face_recognition.face_encodings(image_robson)[0]
16 rodrigo_face_encoding = face_recognition.face_encodings(image_rodrigo)[0]
17 tarcisio_face_encoding = face_recognition.face_encodings(image_tarcisio)[0]
18
19 know_face_encodings = [
20     joel_face_encoding,
21     robson_face_encoding,
22     rodrigo_face_encoding,
23     tarcisio_face_encoding
24 ]
25
26 know_face_names = [
27     "Joel",
28     "Robson",
29     "Rodrigo",
30     "Tarcisio"
31 ]
32
33 #Carregando Imagem de Teste
34 test_image = face_recognition.load_image_file('./images/robrodtar.jpg')
35
36 #Encontrando faces em imagem de teste
37 face_locations = face_recognition.face_locations(test_image)
38 face_encodings = face_recognition.face_encodings(test_image, face_locations)
39 #Gerando nova imagem
40 pil_image = Image.fromarray(test_image)
41
42 #Criar instancia ImageDraw
43 draw = ImageDraw.Draw(pil_image)
44
45 #Coordenadas dos rostos encontrados
46 for face_location in face_locations:
```

```

47 #Mostra a posição de cada face
48 top, right, bottom, left = face_location
49 print("Posição da face localizada Topo:{}, Esquerda:{}, Fundo:{}, Direita:{} ".format(top, left,
bottom, right))
50
51 #Estrutura de repetição identificando as faces conhecidas
52 for(top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
53 matches = face_recognition.compare_faces(know_face_encodings, face_encoding)
54
55 name = "Desconhecido"
56
57 if True in matches:
58     first_match_index = matches.index(True)
59     name = know_face_names[first_match_index]
60 #Desenha um retângulo em torno da face
61 draw.rectangle(((left, top), (right, bottom)), outline=(0,255,255))
62
63 #Coloca uma etiqueta com nome em baixo da face
64 text_width, text_height = draw.textsize(name)
65 draw.rectangle(((left, bottom - text_height - 20), (right, bottom)), fill=(0,0,255),
outline=(0,0,255))
66 draw.text((left + 6, bottom - text_height - 6), name, fill=(255,255,255,255))
67
68 #Fala o nome encontrado, os nao identificados serao nominados com desconhecido
69 call(["espeak",name])
70
71 del draw
72 #Exiba a imagem com as faces reconhecidas
73 pil_image.show()
74 pil_image.save('./images/nvimage/identify005.jpg')

```

Fonte: elaborado pelos autores.

Anexo 02 - Arquivos de imagem usados como base para comparação pelo programa do projeto

E.V.E.

			
Prof. Joel	Prof. Robson	Prof. Rodrigo	Prof Tarcísio

Foto: Acervo pessoal cedido para o projeto pelo orientador

Anexo 3 - Imagem com três rostos de professores do IFSP
Câmpus Guarulhos usada como teste no projeto



Foto: Acervo pessoal cedido para o projeto pelo orientador

Anexo 4 - Imagem com os três rostos identificados pelo
programa EVE versão beta.



Foto: Acervo pessoal do orientador cedido para o projeto.