PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Aprendizagem de Máquina utilizando a API de detecção facial do Azure*

Um sistema de *Home Security*, para implementação de um serviço de segurança de acesso utilizando detecção facial

Link para acesso aos códigos e imagens desenvolvidos: **Clique Aqui**.

Lucas Teixeira Matos Diniz[1]
Mateus Felipe Mendes Viana[2]
Otávio Augusto Martins[3]
Thales Durães de Aguiar Marques[4]

Resumo

O reconhecimento digital de imagem é um grande artifício utilizado em empresas para reconhecer seus funcionários, realizando assim a sua liberação ou bloqueio. O reconhecimento facial é uma tecnologia que pode aprender a identificar traços humanos para diferenciar indivíduos. Esta inteligência artificial é capaz de converter as imagens em fórmulas matemáticas como por exemplo distância entre os olhos, tamanho do cabelo entre outras características e realizar a categorização do indivíduo. Com isso podemos montar uma base de dados, com centenas de rostos e passar um pequeno grupo de funcionários autorizados a entrar em uma determinada empresa, assim a inteligência diferencia cada individuo e autoriza ou não sua entrada. Desta forma é apresentado um código desenvolvido em Python em relação ao objetivo do trabalho proposto.

Palavras-chave: Reconhecimento facial, Inteligência artificial, Categorização, Indivíduo.

*Artigo apresentado ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais como pré-requisito para obtenção de nota na disciplina de Inteligência Artificial.

[1]Aluno do Programa de Graduação em Engenharia da Computação, Brasil– ltmdiniz@sga.pucminas.br.
[2]Aluno do Programa de Graduação em Engenharia da Computação, Brasil– mfmviana@sga.pucminas.br.
[3]Aluno do Programa de Graduação em Engenharia da Computação, Brasil– otavio.martins@sga.pucminas.br.
[4]Aluno do Programa de Graduação em Engenharia da Computação, Brasil– thales.aguiar@sga.pucminas.br.

Abstract

Digital image recognition is a great device used in companies to recognize their employees, thus carrying out their release or blocking. Facial recognition is a technology that can learn to identify human traits to differentiate individuals. This artificial intelligence is capable of converting the images into mathematical formulas such as distance between the eyes, hair size, among other characteristics and to categorize the individual. With that we can set up a database, with hundreds of faces and pass a small group of employees authorized to enter a certain company, so the intelligence differentiates each individual and authorizes or not their entry. In this way, a code developed in Python is presented in relation to the objective of the proposed work.

Keywords: Facial recognition, Artificial intelligence, Categorization, Individual.

1 INTRODUÇÃO

A automação residencial, designada também como domótica, corresponde ao uso de inovações tecnológicas para satisfazer, facilitar as necessidades e, principalmente, dar conforto aos integrantes de determinada habitação. A palavra domótica tem origem na palavra latina "Domus"que significa "casa", unida a palavra "Robótica", que é a automatização e controle de qualquer processo. A área encontra-se em crescente evolução nas últimas décadas, acompanhada pelo avanço da tecnologia e aproximação da mesma com atividades ligadas ao cotidiano. A domótica tem como principal origem a automação industrial, enriquecida com o surgimento dos Controladores Lógicos Programáveis (CLPs) durante a década de 60. Dessa maneira, a domótica permite ao usuário controlar dispositivos eletrônicos de sua residência através de interfaces de controle e recursos como reconhecimento facial, de digitais, por voz, por íris, entre outros (EUZÉBIO; MELLO, 2011).

A biometria é a ciência que estabelece a identidade de um indivíduo baseada em seus atributos físicos, químicos ou comportamentais como explica (SILVA; CINTRA, 2015) . Este tipo de sistema possui inúmeras aplicações em diversas áreas, podemos citar como uma das principais a de segurança, um sistema de gerenciamento de identidade é um dos seus principais e mais eficazes sistemas, cuja funcionalidade é autenticar a identidade de um indivíduo no contexto de uma aplicação.

O reconhecimento facial é uma técnica biométrica que consiste em identificar padrões em características faciais como formato da boca, do rosto, distância dos olhos, entre outros. Um humano é capaz de reconhecer uma pessoa familiar mesmo com muitos obstáculos com distância, sombras ou apenas a visão parcial do rosto. Uma máquina, no entanto, precisa realizar inúmeros processos para detectar e reconhecer um conjunto de padrões específicos para rotular uma face como conhecida ou desconhecida. Para isso, exitem métodos capazes de detectar, extrair e classificar as características faciais, fornecendo um reconhecimento automático de pessoas.

Posteriormente em seu artigo (SILVA; CINTRA, 2015) descreve que a tecnologia biométrica oferece vantagens em relação a outros métodos tradicionais de identificação como senhas, documentos e tokens. Entre elas estão o fato de que os traços biométricos não podem ser perdidos ou esquecidos, são difíceis de serem copiados, compartilhados ou distribuídos. Os métodos requerem que a pessoa autenticada esteja presente na hora e lugar da autenticação, evitando que pessoas má intencionadas tenham acesso sem autorização. A autenticação é o ato de estabelecer ou confirmar alguém, ou alguma coisa, como autêntico, isto é, que as alegações feitas por ou sobre a coisa é verdadeira afirma sem seu trabalho (SILVA; CINTRA, 2015).

Assim, a autenticação biométrica é o uso da biometria para reconhecimento, identificação ou verificação, de um ou mais traços biométricos de um indivíduo com o objetivo de autenticar sua identidade. Os traços biométricos são os atributos analisados pelas técnicas de reconhecimento biométrico(SILVA; CINTRA, 2015) corrobora em seu artigo.

1.1 Processos utilizados no reconhecimento facial

Um sistema de processamento de imagens e constituído de diversas etapas, tais como: formação e aquisição da imagem, digitalização, pré-processamento, segmentação, pós-processamento, extração de atributos, classificação e reconhecimento.

• Aquisição de Imagens Digitais: Dois elementos são necessários para a aquisição digital de imagens, sendo que o primeiro é um dispositivo físico sensível ao espectros de luz, o segundo é chamado de digitalizador, que é um dispositivo que converte o sinal analógico produzido na saída do sensor em uma sinal digital.

• Técnicas de Pré-processamento: As técnicas de pré-processamento têm a função de melhorar a qualidade da imagem.

• Segmentação: Segmentar uma imagem significa, de modo simplificado, separar a imagem como um todo nas partes que a constituem e que se diferenciam entre si.

• Pós-processamento: É nesta etapa que os principais defeitos ou imperfeições da segmentação são devida-mente corrigidos.

• Extração de Atributos: A etapa final de um sistema de processamento de imagens é aquela em que se extrai as informações úteis da imagem processada.

• Classificação e Reconhecimento: O objetivo do reconhecimento e realizar, de forma automática, a "identificação" dos objetos segmentados na imagem.

1.2 Aplicações da Inteligência Artificial e Aprendizado

Inicialmente, precisamos entender o que é inteligência artificial, (OSÓRIO; BITTENCOURT, 2000) define inteligência artificial como nada mais é do que uma tentativa de formalizar o eterno sonho da criação de um "cérebro eletrônico" (termomuito usado na ficção científica e mesmo na época inicial do desenvolvimento dos computadores).

Para o nosso trabalho isto é de extrema importância pois o programa hora nenhuma pode fazer este reconhecimento de maneira errada, uma liberação de um funcionário não credenciado pode causar um tremendo problema para a empresa comprometendo a segurança de todos os outros colaboradores desta mesma empresa.

Em seu mesmo artigo (OSÓRIO; BITTENCOURT, 2000) descreve aprendizado de máquina como ferramentas de I.A., em sua evolução, começaram adquirindo conhecimentos que eram explicitados pelos especialistas de uma certa área (o que equivale a programar um sistema para resolver um problema).

Trazendo para nosso trabalho, o aprendizado de máquina é a capacidade do nosso programa em reconhecer uma imagem qualquer fazer o processamento e armazenar esta informações, e posteriormente a liberação ou não do funcionário, o aprendizado entraria na automatização da liberação do acesso do funcionário e o armazenamento dos colaboradores pré-cadastrados.

Em aprendizado de maquina podemos descrever algumas etapas como:

- Adaptação do comportamento (melhoria)
- Correção dos erros cometidos no passado
- Otimização da performance do sistema (melhoria)
- Interação com o meio, experimentação e descoberta
- Representação do conhecimento adquirido (Memória e compressão dos conhecimentos)

Dentro deste contexto, diversas tentativas de replicar a inteligência humana através de máquinas foram feitas, dando origem ao que é reconhecido atualmente por inteligência artificial (IA). Entende-se que a IA tem como objetivo copiar ou assemelhar-se à inteligência humana. Assim, ela é definida como um conjunto de teorias e técnicas de implementação lógica que permite a uma máquina a capacidade de adquirir conhecimento (LAROUSSE, 1999).

1.3 Plataforma da Azure

Taurion (2011) define a computação na nuvem como um conjunto de recursos como capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na Internet.

Para o nosso trabalho a plataforma Microsoft Azure oferece recurso necessários para o desenvolvimento e conclusão do nosso trabalho, oferecendo uma plataforma totalmente gratuita, devido a parceria com nossa instituição de ensino, com alta disponibilidade.

O Microsoft Azure é o serviço de hospedagem e gerência de dados na nuvem da Microsoft feito especialmente para desenvolvedores ou gerentes de TI de clientes empresariais (ALVES, 2016). Entre suas vantagens está a integração de servidores do data center próprio do usuário com a infraestrutura de nuvem, fornecendo ferramentas de testes, de bug e lançamento de aplicações para as máquinas da rede com poucos cliques.

O Azure fornece suporte a diversos sistemas operacionais, linguagens de programação, ferramentas, bancos de dados e dispositivos. É possível compilar aplicativos com JavaScript, Python, .NET, PHP, Java e Node.js, além de compilar back-ends para dispositivos iOS, Android e Windows e outras tarefas.

O serviço é flexível. Dependendo da demanda necessária do usuário, a capacidade de armazenamento, computação, largura de banda e outros recursos do Azure são rapidamente escalados ou reduzidos verticalmente para acompanhar e o consumidor tem a opção de pagar somente pelo que de fato usar.

1.4 Python

Para (MENEZES, 2010) A linguagem de programação Python é muito interessante como primeira linguagem de programação devido à sua simplicidade e clareza. Embora simples, é também uma linguagem poderosa, podendo ser usada para administrar sistemas e desenvolver grandes projetos.

Python é uma linguagem de programação de altíssimo nível (VHLL - Very HighLevel Language), criada pelo holandês Guido Van Rossum sob o ideal de "Programação de Computadores para todos". Este ideal fez com que o desenvolvimento de Python tivesse sempre em mente a liberdade (gratuita, código aberto), disponibilidade (Python roda em Windows, Linux, Mac, Palm, em celulares, e outra infinidade de sistemas) e principalmente a clareza de sintaxe, que hoje é responsável pela alta produtividade só conseguida com Python (LABAKI, 2013).

É uma linguagem Orientada a Objetos, um paradigma que facilita entre outras coisas o controle sobre a estabilidade dos projetos quando estes começam a tomar grandes proporções. Mas como a Orientação a Objetos ainda é vista como um paradigma para "experts", Python permite que o usuário programe na forma procedural, se desejar.

Optou-se por esta linguagem pois como na citação acima é clara e objetiva, e acabou-se tendo muitos problemas com a linguagem C#.

1.5 Integração entre linguagem Python e plataforma Azure

O processo de criação de aplicações dentro da Aruze se da de uma maneire bem simples, pois existem vários tutorias nas plataformas da Microsoft. O suporte personalizável ao Python no Serviço de Aplicativo do Azure é fornecido como um conjunto de extensões de site do Serviço de Aplicativo e cada uma contém uma versão específica do runtime do Python.

O Microsoft Azure permite a criação de aplicativos Web Python em nuvem, assim pode-se criar aplicativos Web melhores, mais rapidamente, com a plataforma de aplicativos gerenciada otimizada para Python. É possível conectar aplicativos aos dados usando serviços do Azure para bancos de dados relacionais e não relacionais populares.

Além disso, a plataforma é flexível para IA e aprendizado de máquina com Python, podendo criar, treinar, hospedar e implantar modelos de forma rápida e fácil em qualquer ambiente Python, com serviços do Azure para ciência de dados e aprendizado de máquina.

2 OBJETIVO GERAL

Esse trabalho tem como objetivo exercitar todo o ciclo de vida de aprendizado de máquina, desde a coleta de dados, indução de modelo, validação do modelo e distribuição em um serviço na nuvem. O objetivo geral é a implementação de um Sistema Distribuído versando sobre o tema geral *Home Security* utilizando a API de detecção facial do Azure. Implementando um serviço de segurança de acesso utilizando detecção facial.

2.1 Objetivo Especifico

Treinamento do *Machine Learning Studio* no programa de treinamento Microsoft AcademIA, disponível no *website* da Microsoft (https://www.microsoft.com/pt-br/academia). Cursando os seguintes módulos propostos:

- *Webinar* preparatório - Nível Básico
- Treinamento digital interativo *Microsoft Learn* - Nível intermediário - Em português (parte 1 a 6)
- Publicar um experimento de aprendizado de máquina com o Microsoft Azure *Machine Learning Studio*
- Processar e classificar imagens com o Serviços de Visão Cognitiva do Azure.

3 DESENVOLVIMENTO

Usando o reconhecimento de face a API de detecção facial do Azure, foi feito um sistema que reconhece pessoas autorizadas e não autorizadas. O código foi desenvolvido na linguagem de programação Python e o código possui algumas diretrizes.

- Controlar o acesso de pessoas ao ambiente através do reconhecimento de face.
- Capturar imagens a partir de uma câmera de qualquer tipo, que será utilizada para liberar ou negar o acesso a esta pessoa.
- Utilizar algoritmos baseados em aprendizagem de máquina para autorização de acesso.
- Consumir um web service na nuvem, cuja função será controlar as regras de acesso, conforme definido no algoritmo

Foi usado a função disponibilizada no Azure chamada *person group* para criar um conjunto de pessoas autorizadas e adicionada imagens das pessoas. O conjunto de pessoas foi nomeado com "autorizados". Então, foi adicionado mais fotos dos indivíduos para possuir uma base de teste maior. O modelo então foi treinado para que consiga reconhecer essas pessoas com imagens diferentes.

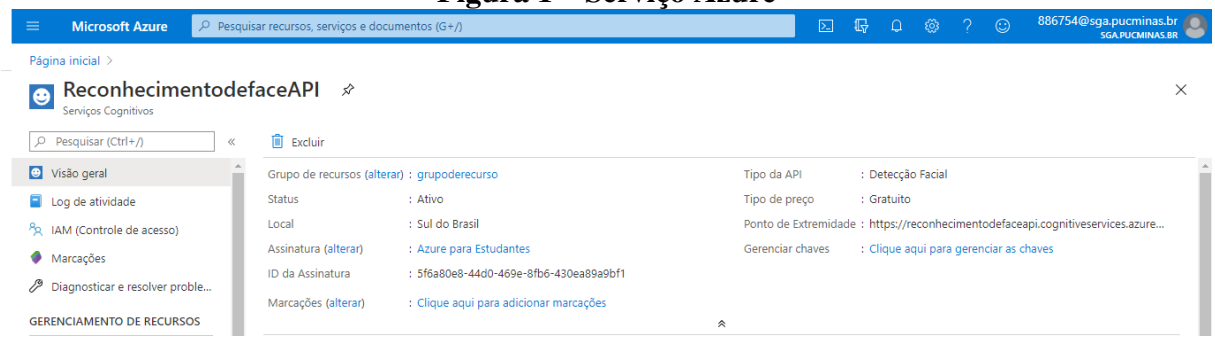
Se o sistema, a partir de uma imagem reconhecer a pessoa pertencente ao person group criado, então essa pessoa estará autorizada no sistema de segurança. Caso não consiga reconhecer, essa pessoa será discriminada, e não estará autorizada.

Os códigos desenvolvidos e imagens estão disponíveis no GitHub. O Link para acesso se encontra disponível no hiperlink a seguir: **Clique Aqui**.

3.1 Explicando o funcionamento do Sistema desenvolvido

Para iniciar, foi criado um serviço na azure para utilização do recurso de reconhecimento facial. O serviço criado foi na zona "Sul do Brasil". Ela pode ser observada na Figura 1.

Figura 1 – Serviço Azure



Fonte: Autores

Inicialmente foi realizado um código, que captura a foto do individuo. Com auxilio da biblioteca “opencv”, demonstrada na Figura 2. Para usar essa biblioteca primeiro foi necessário instalar a biblioteca no sistema operacional.

Figura 2 – Biblioteca OpenCV

Fonte: Autores

Depois foi feito feito o código, descrito na Figura 3. O programa “tirar foto” abre o dispositivo de vídeo padrão do sistema operacional, no nosso de notebooks é uma webcam. Se pressionando a tecla ‘s’ captura uma foto e salva ela no diretório raiz da do programa.

Figura 3 – Código responsável por tirar foto

Fonte: Autores

Para comunicar com a API da azure e uso de sistema de reconhecimento facial, foi necessária a instalação no Sistema Operacional da biblioteca “azure_cognitiveservices_vision face”, visualizada na Figura 4.

Figura 4 – Biblioteca Azure

Fonte: Autores

O código “FaceAPIConfig”, descrito na Figura 5, armazena as configurações de URL e chave de subscrição para comunicação com a API do Azure.

Figura 5 – Comunicação API Azure

FaceAPIConfig.py

```
subscription_key = "af5864b3cb4e455abc050aa2ab83c7ff"
face_api_url = 'https://brazilsouth.api.cognitive.microsoft.com/face/v1.0/detect?'
facegroup_api_url = 'https://brazilsouth.api.cognitive.microsoft.com/face/v1.0'

def config():
    print("Call Config")
    return subscription_key, face_api_url
```

Fonte: Autores

O programa “AzureFaceAPI_Basic”, descrito na Figura 6, faz uma análise da imagem tirada pela câmera. Através de uma comunicação com a API da azure, é retornada em console das informações da pessoa reconhecida na imagem. As informações são salvas em um arquivo de texto na pasta raiz do programa, com o título “análise da imagem”.

Figura 6 – Código para reconhecimento da imagem

AzureFaceAPI_Basic.py

```
import requests
from PIL import Image
import os
import FaceAPIConfig as cnfg

image_path = os.path.join('imagens/fotocamera.png')
image_data = open(image_path, "rb")

subscription_key, face_api_url = cnfg.config();

headers = {'Content-Type': 'application/octet-stream',
           'Ocp-Apim-Subscription-Key': subscription_key}

params = {
    'returnFaceId': 'true',
    'returnFaceLandmarks': 'true',
    'returnFaceAttributes': 'age,gender,headPose,smile,facialHair,glasses,emotion'
}


response = requests.post(face_api_url, params=params, headers=headers, data=image_data)
response.raise_for_status()
faces = response.json()
print(faces)
arquivo = open("analise da imagem.txt","w")

arquivo.write(str(faces))
arquivo.close()
```

Fonte: Autores

A análise da imagem mostrada no console é mostrada na Figura 7.

Figura 7 – Saída do Programa

```
Call Config
[{'faceId': '732b9adc-af5b-4063-b1f2-8c5564e9d623', 'faceRectangle': {'top': 244, 'left': 252, 'width': 120, 'height': 120}, 'faceLandmarks':
{'pupilLeft': {'x': 286.2, 'y': 278.4}, 'pupilRight': {'x': 339.7, 'y': 273.2}, 'noseTip': {'x': 304.2, 'y': 307.1}, 'mouthLeft': {'x': 294.5,
'y': 336.4}, 'mouthRight': {'x': 333.7, 'y': 332.6}, 'eyebrowLeftOuter': {'x': 266.9, 'y': 269.6}, 'eyebrowLeftInner': {'x': 292.1, 'y':
267.3}, 'eyeLeftOuter': {'x': 277.5, 'y': 280.1}, 'eyeLeftTop': {'x': 284.6, 'y': 275.9}, 'eyeLeftBottom': {'x': 285.9, 'y': 281.5},
'eyeLeftInner': {'x': 292.5, 'y': 278.2}, 'eyebrowRightInner': {'x': 315.8, 'y': 265.5}, 'eyebrowRightOuter': {'x': 358.6, 'y': 266.0},
'eyeRightInner': {'x': 331.4, 'y': 274.4}, 'eyeRightTop': {'x': 339.1, 'y': 270.8}, 'eyeRightBottom': {'x': 339.9, 'y': 276.9},
'eyeRightOuter': {'x': 348.1, 'y': 273.3}, 'noseRootLeft': {'x': 299.8, 'y': 279.1}, 'noseRootRight': {'x': 313.9, 'y': 277.8},
'noseLeftAlarTop': {'x': 295.3, 'y': 297.0}, 'noseRightAlarTop': {'x': 319.1, 'y': 295.6}, 'noseLeftAlarOutTip': {'x': 293.0, 'y': 309.2},
'noseRightAlarOutTip': {'x': 326.1, 'y': 306.6}, 'upperLipTop': {'x': 309.6, 'y': 328.7}, 'upperLipBottom': {'x': 310.0, 'y': 332.8},
'underLipTop': {'x': 309.7, 'y': 335.0}, 'underLipBottom': {'x': 310.4, 'y': 342.2}}, 'faceAttributes': {'smile': 0.0, 'headPose': {'pitch':
-4.6, 'roll': -2.6, 'yaw': -14.1}, 'gender': 'male', 'age': 31.0, 'facialHair': {'moustache': 0.6, 'beard': 0.6, 'sideburns': 0.6}, 'glasses':
'NoGlasses', 'emotion': {'anger': 0.005, 'contempt': 0.0, 'disgust': 0.0, 'fear': 0.0, 'happiness': 0.0, 'neutral': 0.991, 'sadness': 0.003,
'surprise': 0.0}}}]
[Finished in 1.2s]
```

Fonte: Autores

Para utilizar a função de *"Person Group"*, primeiro foi necessária a instalação da biblioteca no sitema operacional "Cognitive_face", exibido na Figura 8.

Figura 8 – Biblioteca *Person Group*

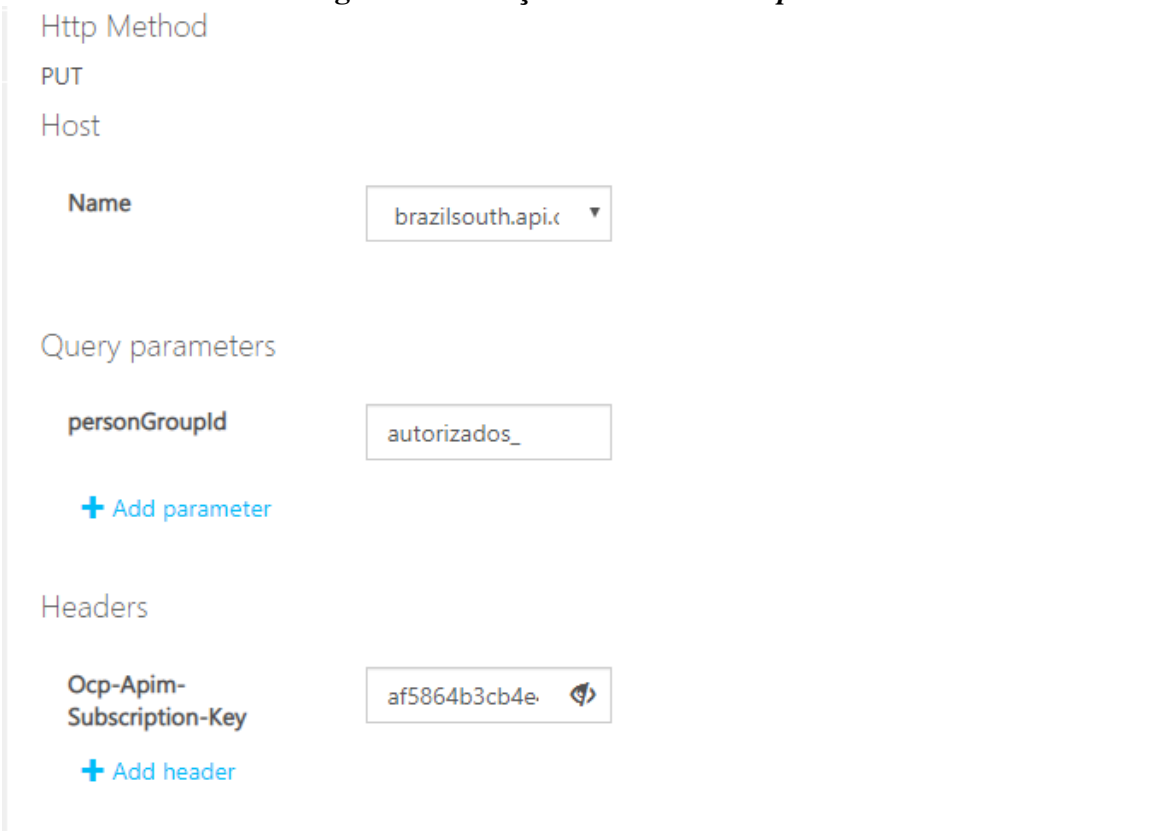
```
C:\Users\lucas>pip install cognitive_face
Collecting cognitive_face
  Downloading https://files.pythonhosted.org/packages/59/16/d332bf7de96d651e9f83dfe15c2b7e2bf47beea73b72f9aa1d35884e3fcb/cognit
ive_face-1.5.0-py3-none-any.whl
Requirement already satisfied: requests in c:\users\lucas\appdata\local\programs\python\python38-32\lib\site-packages (from cog
nitive_face) (2.23.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\lucas\appdata\local\programs\python\python38
-32\lib\site-packages (from requests->cognitive_face) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lucas\appdata\local\programs\python\python38-32\lib\site-packages
 (from requests->cognitive_face) (2020.4.5.1)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\lucas\appdata\local\programs\python\python38-32\lib\site-packages
(from requests->cognitive_face) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\lucas\appdata\local\programs\python\python38-32\lib\site-packages (from
 requests->cognitive_face) (2.9)
Installing collected packages: cognitive-face
Successfully installed cognitive-face-1.5.0
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\lucas>
```

Fonte: Autores

Para criar um *person group*, primeiro entramos no domínio que está salvo o serviço de reconhecimento facial do azure disponível em: <https://brazilsouth.dev.cognitive.microsoft.com>. Assim conseguimos criar um *person group*, evidenciado na Figura 9, para segregar os autorizados dos não autorizados.

Figura 9 – Criação do *Person Group*



Fonte: Autores

A figura 10, mostra o código desenvolvido que autoriza o usuário identificado pela câmera. Inicialmente é criado um *person group* chamado "autorizados_"que adiciona um usuário com o nome de "Lucas Diniz"para o grupo. Com uma biblioteca de fotos já pre-estabelecida, é adicionada ao grupo. Esse grupo então é treinado. Usando a função *"Detect"*, o programa usando a foto obtida pela câmera de reconhecimento, tenta detectar o usuário no *person group*. Logo após, com a função *"Identify"*, tenta identificar qual o nome da pessoa e o grau de confiabilidade da comparação das fotos. Se o resultado obtido estiver dentro do alcance a pessoa é autorizada no sistema. Caso não esteja dentro do alcance, ela não estará autorizada.

Figura 10 – Criação do *Person Group*

```
import http.client, urllib.request, urllib.parse, urllib.error, base64
import cognitive_face as cf
import FaceAPIConfig as cnfg
PERSON_Group_ID = 'autorizados_'
cf.BaseUrl.set (cnfg.facegroup_api_url)
cf.Key.set (cnfg.subscription_key)
#https://brazilsouth.api.cognitive.microsoft.com/face/v1.0/persongroups/autorizados_
subscription_key, face_api_url = cnfg.config();
name = "Lucas Diniz"
user_data = 'autorizado'
recognitionModel = 'recognition_02'
response = cf.person.create(PERSON_Group_ID,name,user_data)
person_id = response['personId']
#adicionando as imagens no grupo "autorizados_"
cf.person.add_face('imagens/imagem_PESONGROUP.jpeg', PERSON_Group_ID, person_id)
#mostrando no console as pessoas que estão adicionadas no grupo dos autorizadas
print (cf.person.lists(PERSON_Group_ID))
#sempre que adicionamos uma face nova é necessario treinar o modelo para melhorar a confiabilidade dos resultados
cf.person_group.train(PERSON_Group_ID,)
#resposta do retorno do treinamento do person group
response = cf.person_group.get_status(PERSON_Group_ID)
status = response['status']
#Verificando com outra foto se o inidividuo existe no grupo dos autorizados
response = cf.face.detect('imagens/fotocamera_AUTORIZADO.png',PERSON_Group_ID,recognitionModel)
face_ids = [d['faceId'] for d in response]
print (face_ids)
identified_faces = cf.face.identify( face_ids,PERSON_Group_ID,recognitionModel)
print (identified_faces)
# Identify faces
results = face_client.face.identify(face_ids, PERSON_GROUP_ID)
print('Identifying faces in {}'.format(cf.path.basename(image.name)))
if not results:
    print('ALERTA! Pessoa nao esta no grupo! nao foi autorizada')
for person in results:
    print('Pessoa para a id{} é identificada no grupo {} com um grau de confianca de {}.'.format(person.face_id, cf.path.basename(image.name), person.candidate
```

Fonte: Autores

A imagem 11 mostra o resultado da autentificação quando comparada a foto obtida pela câmera de reconhecimento, com as imagens pertencentes ao ID do *person group*

Figura 11 – Resposta obtida pelo *Person Group*

```
Call Config
[{'personId': '8bf625e7-96e7-41ae-a9eb-ac350d35d9fa', 'persistedFaceIds': ['83821555-6116-4692-8869-d5473d4cdf33'], 'name': 'Lucas Diniz', 'userData': 'autorizado'}, {'personId':
'bf15d338-4c63-4753-9648-6adb55683ecf', 'persistedFaceIds': ['0f54bcac-4787-4f1f-bc53-c13b576f902f'], 'name': 'Lucas Diniz', 'userData': 'autorizado'}]
['59752094-c08c-4cb5-a939-8fcc78d22aed']
Pessoa para a id bf15d338-4c63-4753-9648-6adb55683ecf é identificada no grupo Autorizados_ com um grau de confianca de 0.83.
```

Fonte: Autores

4 CONCLUSÃO

Neste trabalho abordamos o assunto aprendizagem de máquina utilizando a API de detecção facial do Azure e concluímos que o programa desenvolvido realmente funcionou e que o se este mesmo algoritmo fosse implementado em uma grande empresa para a liberação de seus funcionários obteria exito em todas as suas verificações. Cumprimos todos os objetivos que tínhamos proposto uma vez que depois do termino do trabalho realizamos teste de validação de resultados descrito no capitulo do desenvolvimento. Este trabalho foi muito importante para o nosso conhecimento, compreensão e aprofundamento deste tema, pois visto que permitiu-nos compreender melhor todos os processos de reconhecimento digital de imagem e aprendizado de máquina, além de ter-nos permitido aperfeiçoar competências de organização e trabalho em grupo.

REFERÊNCIAS

ALVES, PAULO. Azure é o serviço da microsoft para desenvolvedores. **Tech Tudo**, 2016.

EUZÉBIO, M V M; MELLO, E R. Droidlar- automação residencial através do celular android. **Sistemas de Telecomunicações, Instituto Federal de Santa Catarina.São José, SC**, 2011.

LABAKI, Josue. **Introdução a Python**. 2013. Disponível em: <http://www.dcc.ufrj.br/~fabiom/python/pythonbasico.pdf>.

LAROUSSE. Grande enciclopédia larousse cultural. **Editora Nova Cultural**, 1999.

MENEZES, Nilo Ney Coutinho. Introdução a programação com python. **São Paulo: Novatec**, 2010.

OSÓRIO, Fernando S; BITTENCOURT, João R. Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens. In: **I Workshop de inteligência artificial**. [S.l.: s.n.], 2000.

SILVA, Alex Lima; CINTRA, Marcos Evandro. Reconhecimento de padrões faciais: Um estudo. In: **Encontro Nacional de Inteligência Artificial e Computacional, 2015, Proceedings ENIAC**. [S.l.: s.n.], 2015. p. 224–231.

TAURION. **Integração de aplicações e serviços utilizando computação na nuvem com a plataforma Microsoft Windows Azure**. 2011. Tese (Doutorado) — Universidade Regional de Blumenau - Sistemas Distribuídos.