

A utilização de visão computacional em atividades cotidianas

Eduardo D. Agostinho
 Prof. Dr. Alex Fabiano Bueno
Universidade Federal de Santa Catarina
Departamento de Engenharias
 Blumenau, Brasil

Resumo—Este artigo tem por objetivo comparar soluções desenvolvidas relacionadas a visão computacional, além de apresentar possibilidades de implementações de softwares e tecnologias ligadas ao assunto em situações cotidianas. O passo inicial ocorreu por meio da matéria de projeto integrador na Universidade Federal de Santa Catarina, onde foi criado um sistema de chamada automatizado com reconhecimento facial. O projeto tem por objetivo captar a imagem de alunos que entram em sala de aula verificando a presença e armazenando os dados em um banco de dados que estará associado às informações da instituição. A ideia de visão computacional pode, portanto, ser adaptada e aplicada em diversos locais, inclusive em indústrias e comércios.

Index Terms—Inteligência artificial, Visão computacional, OpenCV, Automação de processos.

I. INTRODUÇÃO

A Nalisando o processo de chamadas feito na maioria das escolas, nota-se que na maioria dos casos ainda apresenta algo ultrapassado dada às tecnologias presentes no mercado. A experiência tradicional de chamada se resume em o professor chamando nome por nome dos alunos até que se conclua quem estava presente em aula ou não. Em alguns locais é possível que o professor tenha o nome dos alunos cadastrados em um portal, mas em vários casos a anotação é feita em um papel. Outra tecnologia que está se tornando mais utilizada é a de cartões, onde alunos possuem uma carteira de identificação e passar por uma catraca, geralmente funcionando por NFC ou RFID. O tema deste artigo surgiu da perda de tempo que ocorre durante a chamada tradicional, sendo que este poderia ser destinado ao ensino. O Banco Mundial avaliou 15,6 mil salas de aula, em que mais da metade delas são no Brasil (classes dos ensinos fundamental e médio em MG, PE e RJ), e calcula-se que, em média, apenas 64% do tempo da classe seja usado para transmissão de conteúdo, 20 pontos percentuais abaixo de padrões internacionais [1]. Logo, analisando estes dados, nota-se uma necessidade de que alguma proposta seja feita para reduzir esse desperdício de tempo. Desta forma, a proposta de inclusão da visão computacional em atividades cotidianas como essa é uma alternativa para solucionar problemas diversos. Uma das linguagens de

programação compatíveis com este tema é Python, sendo que esta possui uma biblioteca chamada OpenCV (onde CV é sigla para Computer Vision, ou seja, Visão computacional) com várias aplicações já pré-desenvolvidas e que podem ser aplicadas nas mais diversas situações. Neste artigo será dado um enfoque ao processo de chamada, como já mencionado, porém toda e qualquer atividade que exija reconhecimento de pessoas, padrões, cores, características, entre outros pode usar essa biblioteca para uma possível solução.

II. CONCEITOS DO TEMA

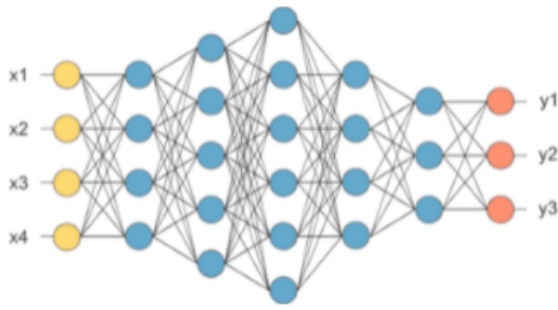
Primeiramente, é importante ter as definições de inteligência artificial bem claras para que se possa compreender o que será apresentado neste artigo. Inteligência artificial é um conceito muito amplo a ser estudado, pois engloba diversas sub-áreas de atuação. No geral, ela refere-se a máquinas e sistemas que simulam a inteligência humana para executar tarefas, além de, em alguns casos, se aprimorar constantemente com base na coleta de dados gerados durante o período de funcionamento. A coleta de dados é fundamental em qualquer algoritmo relacionado ao tema. Dados corretos geram informações corretas e, por consequência, a aplicação torna-se eficiente. Em qualquer tipo de coleta de dados, seja ela antes ou durante a execução do algoritmo, é necessário realizar o treinamento do mesmo para um resultado satisfatório da IA. Explicando um pouco melhor os treinamentos, estes são feitos por meio de informações concretas no caso estudado (de identificação de pessoas para chamada). Ou seja, é necessário treinar o algoritmo com fotos da pessoa e identificá-la. Com base em todas as fotos que o algoritmo mapeou da pessoa, ele encontra padrões para identificação. Após o treinamento, o mesmo está pronto para identificar imagens não classificadas. Na grande maioria dos casos, esta prática de encontrar padrões é feita utilizando redes neurais, que são sistemas de computação que possuem nós interconectados, funcionando de maneira similar ao cérebro humano. Usando algoritmos, elas podem reconhecer padrões escondidos e correlações em dados brutos, agrupá-los e classificá-los. Em sua forma mais geral, uma rede neural é um sistema projetado para modelar a maneira como o cérebro realiza uma tarefa particular, sendo normalmente implementada utilizando-se componentes eletrônicos ou é simulada por propagação em um computador digital. Para alcançarem bom desempenho, as redes neurais empregam

A. D. Eduardo, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, Brasil (e-mail: eddaag@gmail.com).

B. F. Alex, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, Brasil (e-mail: alex.bueno@ufsc.br).

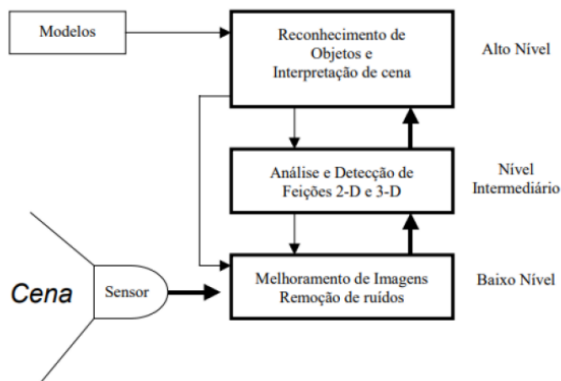
uma interligação maciça de células computacionais simples, denominadas de “neurônios” ou unidades de processamento [2]. O funcionamento de uma rede neural se dá por entrada, camadas de neurônios e saída. A interação entre estes ocorre por recebimento de estímulos, processamento de uma função e retorno de valores. Toda a interação de uma rede neural é matemática e uma representação pode ser vista na figura 1.

Figura 1: Exemplo de rede neural



Com todas as premissas apresentadas, deu-se início ao desenvolvimento proposto. O projeto como um todo procurava além de identificar o aluno, associar o código a um banco de dados para preenchimento e captação de informações, gerar a contagem de alunos durante a aula na sala e enviar a lista de presentes por e-mail para o professor após o término da mesma. Todos estes tópicos serão abordados com menor aprofundamento, visto que o foco do artigo é na aplicação de visão computacional para atividades do dia-a-dia.

Figura 2: Hierarquia de tarefas computacionais



Para realizar o projeto utilizou-se a plataforma PyCharm, desenvolvida pela empresa JetBrains. A IDE fornece análise de código, ferramenta de debugging, autocompletamento de códigos entre outras coisas que facilitam o desenvolvimento, além das biblioteca OpenCV, compatível com visão computacional, Pymysql, compatível com banco de dados e numpy, para trabalhar com os vetores de resultados de treinamento. Essas bibliotecas serão apresentadas na sequência.

III. BIBLIOTECAS

A. OpenCV

OpenCV (Open Source Computer Vision) é uma biblioteca de programação, open source (código aberto) liberado sob a licença BSD (Berkeley Software Distribution) e, portanto, é gratuito para uso acadêmico e comercial, sendo desenvolvido inicialmente pela Intel Corporation. Possui interfaces de C++, C, Python e Java e sendo multiplataforma (Windows, Linux, Mac OS, IOS e Android). OpenCV foi projetado para a eficiência computacional e com um forte foco em aplicações de tempo real. Através do OpenCV é possível a implementação de várias ferramentas de interpretação de imagens, indo desde operações simples como um filtro de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em 3D [3].

B. Pymysql

A biblioteca Pymysql é uma interface cliente MySQL escrita em Python puro, baseada na PEP 249 (Python Enhancement Proposals), que permite realizar conexão e interação com bancos de dados MySQL e MariaDB.

C. Numpy

O NumPy é uma biblioteca para a linguagem Python com funções para se trabalhar com computação numérica. Seu principal objeto é o vetor n-dimensional, ou ndarray. Um vetor n-dimensional também é conhecido pelo nome tensor. A principal característica do ndarray é que ele deve ser homogêneo, ou seja, diferentemente do objeto lista, todos os seus elementos devem ser do mesmo tipo.

IV. ALGORITMOS ÚTEIS NA IMPLEMENTAÇÃO DE RECONHECIMENTO FACIAL

Como dito anteriormente, os treinamentos do algoritmo são geralmente feitos utilizando redes neurais. As redes neurais possuem diversas arquiteturas, como por exemplo Redes Multilayer Perceptrons, Redes Neurais Convolucionais, Redes Neurais Recorrentes, Redes de Hopfield, etc. Cada uma com sua característica, sua utilidade mais específica para determinado tipo de aplicação e seus valores. Neste caso, a rede mais adequada para reconhecimento facial é a convolucional. Porém, mesmo com a rede definida, é necessário definir a quantidade de camadas, as proporções, os valores dos parâmetros e todas as etapas necessárias para criar uma rede neural em código. Isso se torna um trabalho árduo, pois além de ser necessário definir camada por camada de uma rede neural, o usuário também deve comparar resultados com outras redes neurais criadas, ou seja, a depender dos dados, torna-se um método de tentativa e erro até chegar no melhor resultado. Outra opção é a utilização de uma rede neural pronta, porém ainda será necessário entender sua construção para melhorar os resultados. Uma opção simples e útil, portanto, é a utilização de classificadores nativos do openCV. O trabalho deles é semelhante a uma rede neural, porém estes não necessitam serem escritos em códigos por estarem já inclusos na biblioteca, ou seja, eles são chamados por função e já treinam

o algoritmo com base nas informações dadas. Existem alguns tipos de classificadores nativos do openCV, como por exemplo o Eigenfaces, o Fisherfaces e o LBPH.

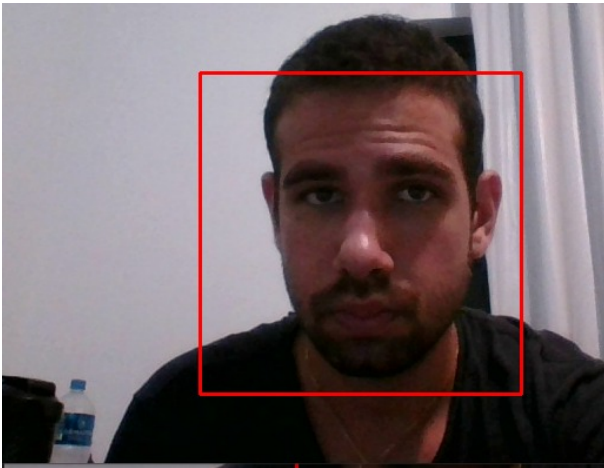
A. Classificadores da biblioteca OpenCv

Eigenfaces é um algoritmo baseado na técnica de PCA (Principal Component Analysis, do português Análise de Componentes Principais). Representa o conjunto de autovetores de uma matriz de covariância formada por imagens de faces. Fisherfaces é um algoritmo baseado na técnica de LDA (Linear Discriminant Analysis, do português Análise discriminante linear). Procura reduzir a dimensionalidade (eliminar redundância) dos dados extraídos de um conjunto. LBPH (Local Binary Patterns Histograms) trata de um algoritmo baseado no conceito de números binários (0 e 1, ou, para o caso, pixel claro ou escuro). Procura transformar a imagem em um padrão binário e analisar grupos de pixels.

B. Testes realizados

Para avaliar os classificadores, foram realizados testes. Estes foram feitos em conjunto com o HC (Haar Cascade), um detector de rostos. Portanto, o HC reconhecia que havia um rosto na webcam e tirava uma foto, o usuário informava quem era a pessoa da imagem e cada um dos algoritmos era treinado com base nas mesmas fotos para posterior reconhecimento na câmera. As imagens de treinamento foram convertidas para cinza.

Figura 3: visão da câmera durante captura



Na prática, o treinamento realiza leitura das fotos captadas utilizando o reconhecedor do OpenCV. Deste modo ele utiliza o reconhecedor para percorrer todas as fotos e encontrar padrões em cada pessoa informada. O reconhecedor então gera um arquivo .yml a ser lido pela biblioteca do OpenCV. Um ponto importante é que todo o treinamento é feito chamando uma função, sendo necessário poucas linhas de código, diferentemente de uma aplicação com rede neural. Isso pode ser visto na figura 4.

De acordo com os testes realizados com a mesma base de dados, o algoritmo LBPH foi o que se destacou no processo de

Figura 4: Código de treinamento do algoritmo

```
import cv2
import os
import numpy as np

lbph = cv2.face.LBPHFaceRecognizer_create()

def getImagemComId():
    caminhos = [os.path.join('fotos', f) for f in os.listdir('fotos')]
    faces = []
    ids = []
    for caminhoImagem in caminhos:
        imagemFace = cv2.cvtColor(cv2.imread(caminhoImagem), cv2.COLOR_BGR2GRAY)
        id = int(os.path.splitext(caminhoImagem)[-1].split('.')[1])
        ids.append(id)
        faces.append(imagemFace)

    return np.array(ids), faces

ids, faces = getImagemComId()

print('treinando')
|
lbph.train(faces, ids)
lbph.write('classificadorlbph.yml')

print('treinamento finalizado')
```

identificação. Vale lembrar que o algoritmo necessita identificar a pessoa que está na câmera como alguém do seu banco, ou seja, se o treinamento foi realizado com as pessoas A e B, e uma pessoa C aparecer na frente da câmera, o algoritmo identificará C como A ou B. Para arrumar este problema do algoritmo é utilizado o conceito de índice de confiança, onde quanto menor o valor, maior a probabilidade da identificação estar correta. Em valores práticos obtidos no LBPH, valores abaixo de 35 são satisfatórios para confirmar que a pessoa está sendo corretamente classificada. Portanto, se apenas A e B foram cadastrados e a câmera mostra a pessoa C, podemos solicitar que o algoritmo defina quem é aquela pessoa apenas se ele identificar que o índice de confiança na predição dele é menor que 35.

V. OUTRAS FUNCIONALIDADES DO ALGORITMO DESENVOLVIDO

Por mais que o foco do artigo seja a visão computacional em si, outras funcionalidades foram introduzidas no sistema de chamada automatizada criado. Ao total, foram utilizadas 9 bibliotecas, responsáveis pela integração da solução. A linguagem python permite o uso da visão computacional com os mais diversos tipos de automatização, facilitando tarefas simples cotidianas. O processo é o seguinte:

- Aula cadastrada informando professor e matéria;
- Alunos cadastrados no banco, realizada a captura de imagens e treinamento do algoritmo;
- Professor seleciona a matéria e começa a chamada;
- Aluno para em frente da câmera e recebe mensagem confirmando sua presença, o sistema armazena horário de entrada do aluno;
- Uma câmera secundária fica na sala e a cada período de tempo conta quantas pessoas estão presentes.
- Professor encerra chamada quando desejar, é gerado um pdf com informações de alunos presentes e enviado por e-mail.

Figura 5: Reconhecimento facial do projeto de chamada



VI. OUTRAS APLICAÇÕES PARA VISÃO COMPUTACIONAL

Atualmente é possível ver aplicações de visão computacional nos mais diversos meios:

- Militar: O uso de tecnologia está maior no meio militar, envolvendo cada vez menos risco aos seres humanos. A detecção de mísseis teleguiados por exemplo é uma aplicação de visão computacional.
- Serviços públicos: Buscando aumentar as seguranças das ruas, câmeras espalhadas por uma determinada região com identificação de movimentação suspeita ou cuidado do tráfego de veículos são exemplos.
- Segurança: Sistemas de reconhecimento facial estão se espalhando nos mais diversos aplicativos, como em verificações de identidade por exemplo, que pede para a pessoa realizar movimentos específicos com a cabeça para comprovar que é realmente ela.
- Marketing - identificação e categorização para que ações de marketing digital possam ser realizadas com base em pessoas específicas.

A indústria também é uma grande beneficiada da visão computacional, como em questões de controle de qualidade, monitoramentos em geral e inclusive sensoreamento (por exemplo, carros autônomos). Todos esses pontos levantados possuem algo em comum: a ideia de que a tecnologia está muito longe de ser desenvolvida em atividades cotidianas e limita-se a locais com grande investimento. Porém, como já mostrado anteriormente, é possível a criação de um algoritmo eficiente com poucas linhas de código e conhecimento básico de ferramentas. Existem possibilidades diversas de projetos em várias tarefas que facilitam os processos, deixando estes cada vez mais padronizados.

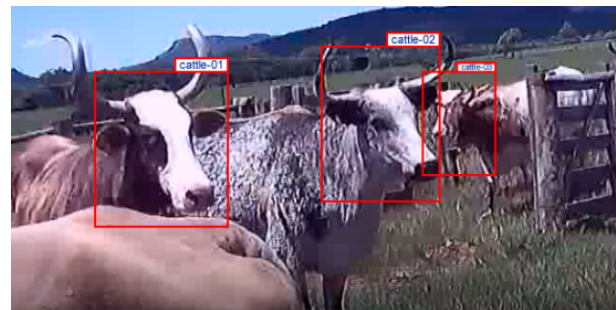
A. Reconhecimento facial em casa

Algoritmos de visão computacional podem ser criados para auxiliar humanos em suas vidas cotidianas. Dentre as opções, estão desde o reconhecimento para segurança (destrancar portas apenas por identificação, por exemplo) até identificador de fadiga em atividades como trabalhar no computador.

B. Identificação de animais

A visão computacional tem uma ampla utilidade com animais. É possível, por exemplo, auxiliar a pecuária, identificando animais que estão próprios para o abate. Outras aplicações possíveis são automatizações do processo de cuidado dos pets, como identificação do nível de água ou comida e reposição dos mesmos.

Figura 6: Reconhecimento de animais



C. Contador de pessoas

Todos os locais com fluxo de pessoas como shows, eventos ou até supermercados podem contar com visão computacional, tanto nas entradas quanto nos ambientes. Ela auxilia o acesso à informação de maneira rápida e simples.

D. Estacionamentos e detecção veicular

O acesso à informação advinda dos carros é um ramo com muito potencial a ser explorado no campo de visão computacional. Informações como cor, placa e modelo podem ser utilizadas. Um exemplo possível é a criação de um sistema em um estacionamento, que identifica automaticamente a entrada e saída do carro pela placa, realizando a cobrança de maneira automática.

E. Aplicativos

É possível a criação de aplicativos que utilizem os conceitos de visão computacional, como por exemplo borrar rosto automaticamente ou retornar a silhueta em fotos.

VII. CONCLUSÃO

Do projeto de chamada automatizada concluiu-se que algoritmos de visão computacional podem ser implementados de maneira simples para pessoas com conhecimento básico no tema. Existem diversas situações que fazem parte do cotidiano e que a aplicação dessa tecnologia seria eficiente. Ela pode ser utilizada para segurança, comodidade, acesso a informação ou simplesmente para automatizar um processo.

Figura 7: Identificação de veículos



REFERÊNCIAS

- [1] P. R7, *Desperdício de tempo em sala de aula 'drena' investimentos em educação*, Rede Record, 2021. [Online]. Available: <https://noticias.r7.com/educacao/desperdicio-de-tempo-em-sala-de-aula-drena-investimentos-em-educacao-19112014>
- [2] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2007.
- [3] T. D. Bissi, "Reconhecimento facial com os algoritmos eigenfaces e fisherfaces," *Universidade Federal de Uberlândia*, p. 41, 2018.