UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Sidarta Lu Ye Almeida

ANÁLISE DE DESEMPENHO DE UM SISTEMA DE ESTACIONAMENTO INTELIGENTE POR MEIO DE RECONHECIMENTO FACIAL ATRAVÉS DA REUTILIZAÇÃO DE UM SISTEMA DE CÂMERAS DE SEGURANÇA PRÉ-EXISTENTES NO AMBIENTE

Araranguá
2021

Sidarta Lu Ye Almeida

ANÁLISE DE DESEMPENHO DE UM SISTEMA DE ESTACIONAMENTO INTELIGENTE POR MEIO DE RECONHECIMENTO FACIAL ATRAVÉS DA REUTILIZAÇÃO DE UM SISTEMA DE CÂMERAS DE SEGURANÇA PRÉ-EXISTENTES NO AMBIENTE

Trabalho de Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Antonio Carlos Sobieranski, Dr.

Araranguá

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Almeida, Sidarta Lu Ye
Análise de desempenho de um sistema de estacionamento inteligente por meio de reconhecimento facial através da reutilização de um sistema de câmeras de segurança pré existentes no ambiente / Sidarta Lu Ye Almeida ; orientador, Antonio Carlos Sobieranski, 2021.
36 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2021.

Inclui referências.

1. Engenharia de Computação. 2. Visão Computacional. 3. Sistema de Estacionamento Inteligente. 4. Distância Euclidiana. 5. Reconhecimento Facial. I. Sobieranski, Antonio Carlos . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Sidarta Lu Ye Almeida

ANÁLISE DE DESEMPENHO DE UM SISTEMA DE ESTACIONAMENTO INTELIGENTE POR MEIO DE RECONHECIMENTO FACIAL ATRAVÉS DA REUTILIZAÇÃO DE UM SISTEMA DE CÂMERAS DE SEGURANÇA PRÉ-EXISTENTES NO AMBIENTE

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pelo Curso de Engenharia de Computação

Araranguá, 23 de setembro de 2021.

Documento assinado digitalmente
Fabricio de Oliveira Ourique
Data: 04/10/2021 11:52:47-0300
CPF: 916.167.860-00
Verifique as assinaturas em https://v.ufsc.br

_____

Prof. Fabrício de Oliveira Ourique, Dr.
Coordenador do Curso

Banca Examinadora:

Documento assinado digitalmente
Antonio Carlos Sobieranski
Data: 05/10/2021 20:14:44-0300
CPF: 005.305.809-77
Verifique as assinaturas em https://v.ufsc.br

Prof. Antonio Carlos Sobieranski, Dr.
Orientador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Anderson Luiz Fernandes Perez
Data: 02/10/2021 16:19:57-0300
CPF: 020.803.459-58
Verifique as assinaturas em https://v.ufsc.br

Prof. Anderson Luiz Fernandes Perez, Dr.
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Jim Lau
Data: 04/10/2021 14:37:10-0300
CPF: 613.464.702-00
Verifique as assinaturas em https://v.ufsc.br

Prof. Jim Lau, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Fábio Rodrigues De La Rocha, Dr.
Avaliador Suplente
Universidade Federal de Santa Catarina

Análise de Desempenho de um Sistema de Estacionamento Inteligente Por Meio de Reconhecimento Facial Através da Reutilização de um Sistema de Câmeras de Segurança Pré-Existente no Ambiente

Sidarta Lu Ye Almeida * Antonio Carlos Sobieranski †

2021, Setembro

RESUMO

Este trabalho trata sobre o desenvolvimento de uma aplicação voltada para um sistema de estacionamento inteligente de baixo custo por meio de reconhecimento facial utilizando a biblioteca baseada em *dlib Face Recognition*. Esta aplicação foi construída em Python e utiliza um sistema de segurança com câmeras analógicas Intelbras, já existentes no ambiente para aquisição das imagens faciais dos usuários. O sistema de comparação e reconhecimento dos rostos dos usuários é feito utilizando funções existentes na biblioteca *Face Recognition*, através de cálculos de Distância Euclidiana. Como prova de conceito, propõe-se um protótipo funcional, utilizando sensores para representar uma cancela de um estacionamento real. A aplicação obteve uma taxa de acerto em cerca de 96 das 100 amostras existentes no *dataset* quando comparado à similaridade entre dois indivíduos na primeira tentativa e uma taxa de acerto em cerca de 98 de 100 amostras na segunda tentativa durante a simulação em um teste sintético fechado de acurácia.

Palavras-chaves: Sistema de Estacionamento Inteligente. Face Recognition, Reconhecimento Facial. Python. Distância Euclidiana.

* sidarta.almeida@grad.ufsc.br

† a.sobieranski@ufsc.br

Performance Analysis of an Intelligent Parking Lot System Through the use of Facial Recognition Reutilizing a Pre-Existing Security System Camera in the Environment

Sidarta Lu Ye Almeida * Antonio Carlos Sobieranski †

2021, September

ABSTRACT

This work tackles the development of an application aimed at a low-cost parking lot system through the use of face recognition using the dlib-based library Face Recognition. The application was built in Python and uses a security system with preexisting Intelbras analog cameras to acquire facial images from the users. The system of comparison and recognition of the users' faces is made using functions that exists in the Face Recognition library, through Euclidean Distance method calculations. As a proof of concept, a functional prototype is proposed, using sensors to represent a real parking gate. The application obtained a hit rate in about 96 of the 100 samples existing in the dataset when compared to the similarity between two individuals at the first try, at the second try the application obtained a hit rate in about 98 of the 100 samples during the simulation in a closed synthetic accuracy test.

Key-words: Rotating Parking System. Face Recognition, Face Recognition. Python. Euclidean Distance.

* sidarta.almeida@grad.ufsc.br

† a.sobieranski@ufsc.br

1 INTRODUÇÃO

De acordo com Anil e Arun (2007, p.01) “A biometria é a ciência que estabelece a identidade de um indivíduo baseado em seus atributos físicos, químicos ou comportamentais”. Existem diversas aplicações de biometria existentes, entre as mais comuns, tem-se: biometria digital ou de palma, retina e a utilizada neste trabalho, facial. Também há outras formas de biometria menos comuns, como a que identifica o formato da orelha, *ear* recognition e uma menos usual ainda que identifica uma pessoa pelo seu jeito de caminhar, a *gait* recognition.

Com a evolução computacional presenciada nos últimos anos, tanto em hardware quanto em software, muitos problemas computacionais que há alguns anos atrás seriam muito difíceis de resolver e exigiriam alto poder de processamento e, consequentemente de preço, hoje em dia passaram a ser simples. O *machine learning* é a capacidade de um computador poder aprender e construir modelos de aprendizagem com o objetivo de realizar predições em certos domínios por conta própria. Datado de 1980, o *machine learning* passou por diversas evoluções para enfim chegar nas modernas Redes Neurais Convolucionais, conhecidas como CNN’s. Redes Neurais Convolucionais são classes de redes neurais artificiais que obtiveram ótimos resultados no processamento e na análise de imagens digitais. Um dos problemas computacionais que possuíam alto grau de dificuldade anos atrás e que devido ao *machine learning* e às CNN’s hoje em dia é trivial é o reconhecimento de imagens, mais especificamente o reconhecimento de rostos humanos, ou seja, a biometria facial. Dentro do campo da biometria facial e as técnicas empregadas para a sua utilização, encontra-se uma biblioteca extremamente eficaz e bastante precisa, a *Face Recognition*. Construída usando o estado da arte da dlib, outra famosa biblioteca, a *Face Recognition* foi criada utilizando técnicas de deep learning, possui acurácia de 99.38% no benchmark *Labeled Faces in the Wild*, um famoso benchmark público para ferramentas de reconhecimento facial. A *Face Recognition* possui diversas funcionalidades, desde encontrar rostos em fotos ou vídeos, encontrar partes isoladas no rosto de pessoas (olhos, nariz, boca e queixo), classificar pessoas por identidade em fotos e vídeos, entre muitas outras funções.

Utilizar a infraestrutura de um sistema de segurança já existente no ambiente é um dos fatores que torna a aplicação deste trabalho econômica e fácil de implementar, pois dentro dos sistemas de câmeras de segurança, utiliza-se muito dois aparelhos chamados DVR (*Digital Video Recorder*), assim como o NVR (*Network Video Recorder*), que basicamente gerenciam o acesso às câmeras do ambiente. Além de transmitir as imagens para a internet por acesso

remoto. Outra de suas funções principais é a gravação local ou na nuvem dos *frames* captados pelas câmeras. A diferença entre DVR e NVR é que o DVR já é um aparelho mais antigo, porém ainda utilizado por algumas pessoas e empresas, voltado para câmeras analógicas e o NVR já é um aparelho voltado para câmeras digitais e mais moderno. As câmeras IP disponíveis no mercado utilizam o protocolo RTSP (*Real Time Streaming Protocol*) que cria um canal por onde o áudio e vídeo serão transmitidos em tempo real. Esse canal pode ser acessado através de um link disponibilizado pelo fabricante e fica operando ativamente dentro de uma rede local LAN. Há dois métodos para obter o link RTSP, um que funciona para câmeras IP digitais, e outro que funciona para câmeras não IP analógicas e estas devem estar conectadas em um DVR para funcionarem.

Atualmente, a biometria facial está sendo difundida em diversos setores do mercado, pois além de eficiente, o custo para implementação não costuma ser elevado e os resultados obtidos são muito satisfatórios do ponto de vista econômico. Encontramos a biometria facial em sistemas de segurança e alerta veicular, onde o proprietário do veículo é cadastrado, junto com o modelo do seu carro, gerando alertas de e-mail e mensagem contendo fotos do motorista, localização do carro por GPS e o motivo do alerta (Espinosa *et al.*, 2019).

Uma das grandes preocupações de quem trabalha na área de vendas, é a satisfação do consumidor e, portanto, um sistema que reconheça expressões faciais de clientes para identificar a satisfação do consumidor pelos produtos oferecidos é ideal. Com um sistema baseado em *deep learning*, CNN e classificador Haar Cascade, é possível identificar e quantificar níveis de satisfação de clientes ao se deparar com certos produtos (Indira *et al*. 2021).

Deste modo, a biometria facial não permanece somente em aplicações voltadas para o consumo, mas também podem ser aplicados na área de segurança, mais especificamente na identificação de prisioneiros em celas de presídios. Por meio de um classificador em cascata e redes neurais convolucionais, é possível criar um sistema de reconhecimento multifaces que identifica cada prisioneiro em cada cela independente do lugar e posição que o mesmo se encontra (Diyasa *et al*. 2021).

Com o objetivo de propor uma solução computacional de baixo custo ao fazer uso da infraestrutura já existente de um sistema de câmeras de segurança, que grande parte dos estabelecimentos atuais possuem hoje em dia, o presente trabalho foi proposto. Este trabalho propõe a criação de um protótipo funcional de um sistema de estacionamento inteligente e alternativo que utiliza reconhecimento facial, ao invés dos atuais *tickets* com código de barras.

Foi utilizado um sistema de câmeras de segurança analógicas Intelbras, já presentes no ambiente de teste para obter a imagem do rosto de usuários, um computador servindo como servidor para executar o código e realizar as operações de reconhecimento facial mais pesadas e sensores servindo como cancela, criou-se o atual protótipo deste trabalho. Como *dataset* para os usuários testados no sistema, foi utilizado o conjunto de fotos do projeto Migma da UFSC.

O restante deste artigo está organizado da seguinte forma: Na Seção 2, são apresentados os trabalhos correlatos com sistemas já existentes de estacionamento que utilizam alguma forma de reconhecimento facial. A Seção 3 apresenta a fundamentação teórica necessária para a compreensão das próximas seções. A Seção 4 descreve a metodologia utilizada para o desenvolvimento deste trabalho, como análises realizadas, algoritmo utilizado, o *dataset* utilizado e soluções de otimização dessas imagens. Na Seção 5 são apresentados os resultados experimentais obtidos da simulação, além da calibração e validação do sistema. Por fim na Seção 6 é apresentado a conclusão e discussões acerca do trabalho desenvolvido, além de possíveis trabalhos futuros.

2 TRABALHOS CORRELATOS

Através da revisão da literatura foi possível identificar trabalhos que abordam o tema de Estacionamentos Inteligentes. Grande parte dos trabalhos pesquisados abordam a criação de aplicações para sistemas SPL. Algumas das aplicações encontradas abordam o tema de ocupação de vagas, identificando vagas disponíveis em um estacionamento, com o objetivo de reduzir filas, tempo de espera para encontrar vagas, entre outros, focando na identificação do espaço. Outro tema encontrado são aplicações voltadas para estacionamentos com usuários previamente cadastrados em um banco de dados. Estes são sistemas mais voltados para a segurança e permitem a entrada somente de pessoas autorizadas pelo sistema. Por fim, o último tópico se relaciona com o tema deste trabalho, sistemas de estacionamento para usuários sem cadastro prévio no sistema, com o objetivo agilizar os processos em que o usuário tem interações com o sistema (entrada, saída e pagamento por exemplo).

Destacam-se os seguintes trabalhos no contexto de ocupação de vagas:

- Sarkar *et al.*, (2019): Neste trabalho foi proposto a utilização de um drone para realizar a detecção de espaços vagos em um estacionamento. Inicialmente é feito um mapeamento utilizando coordenadas do drone, então um algoritmo de programação dinâmica é utilizado para determinar a rota mais curta que possa cobrir o maior número possível de espaços vagos do estacionamento no menor tempo. Dependendo da área do

estacionamento a ser investigada, o caminho e o ângulo do *gimball* do drone mudam dinamicamente ao capturar as imagens. Por fim, um sistema de monitoramento de estacionamento baseado em rede neural profunda é utilizado para determinar quantas vagas estão ocupadas e livres no local. Cada imagem de um espaço no estacionamento que foi capturado pelo drone é testada com um modelo pré-treinado baseado em imagens de objetos que são carros ou não. Após isso, um algoritmo de reconhecimento de placas automático é usado para aplicar as regras do estacionamento. Por fim, os resultados são verificados em uma aplicação web conectada a um servidor na nuvem.

- Delibaltov *et al*., (2013): Neste trabalho é proposto a criação de um *framework* para detecção automática de espaços livres em um estacionamento por meio de uma câmera fixada em um poste de luz. É feito um modelo 3D do volume de vagas disponíveis no estacionamento baseado em sua geometria. A ocupação do local é obtida através de um detector de veículos e o volume inferido de cada espaço. O método foi avaliado em três *datasets* diferentes e obteve-se uma acurácia perto dos 80% para uma larga variedade de imagens testadas.

Os trabalhos correlatos a Estacionamentos Inteligentes com foco em usuários cadastrados em um banco de dados visando a segurança são destacados a seguir:

- CHUNG *et al.*, (2015): Este trabalho propõe a aplicação da tecnologia de reconhecimento de imagens em um sistema de gerenciamento de segurança de estacionamento de apartamentos. Utilizando uma combinação de um sistema de reconhecimento facial em conjunto com um sistema de reconhecimento de placas veiculares foi possível aumentar a segurança e o gerenciamento de um estacionamento, além de controle efetivo da identidade dos usuários cadastrados no sistema. Em resultados experimentais obteve-se valores muito bons de acurácia no reconhecimento facial, o menor valor obtido em testes foi de 87% e o maior foi de 100%. O sistema conseguiu provar que houve controle efetivo da informação do usuário, além da redução de custo e risco quando comparado aos guardas humanos tradicionais.

- Mahmood *et al.*, (2015): Este trabalho propõe a criação de um *framework* de computação paralela para detecção e reconhecimento de objetos visando um estacionar seguro do usuário. O *framework* foi dividido em três partes: Na primeira há a detecção do veículo na entrada do estacionamento. Na segunda é feita a detecção do rosto do motorista. Por fim, na última etapa é feita a identificação do rosto do motorista através de uma busca dentro do banco de dados do sistema. É utilizado o algoritmo de aumento adaptativo para detecção do veículo e do rosto do motorista e *Eigenfaces* para o reconhecimento facial. Além disso, provou-se que a escalabilidade do sistema que executa paralelamente o algoritmo de reconhecimento facial do motorista é muito mais rápida do que a sua execução em série.

Por fim a pesquisa que mais se assemelhou ao tema proposto neste artigo e que visa ambientes que não necessitam de cadastro prévio de usuários, voltado para agilidade durante

operações onde cliente interage com o sistema:

- Persada *et al*., (2019): Este trabalho propõe um sistema de segurança em estacionamentos inteligentes baseado em reconhecimento facial e de placas veiculares. Foi utilizado o método SSIM (Índice de Similaridade Estrutural) para os processos de reconhecimento facial. O método SSIM trabalha ao comparar as diferenças de iluminação, contraste e estrutura entre a imagem original e a imagem *host* que está sendo comparada. E para o reconhecimento de placas de carros foi utilizado uma combinação entre a biblioteca *OpenCV* e algumas outras bibliotecas, mais especificamente para detecção de caracteres foi utilizado a biblioteca OpenALPR. Em testes feitos com trinta amostras, obteve-se o maior valor SSIM de 0.83 com a maior acurácia de 76.67%.

Através da análise das aplicações encontradas na literatura, é possível perceber que apesar de ser um tópico não tão explorado no Brasil, os sistemas de Estacionamento Inteligente ainda são um tema bastante atual na comunidade acadêmica. Em todos os trabalhos pesquisados notou-se que um dos objetivos principais era o baixo custo da implementação das aplicações criadas, além da alta acurácia presente em sua maioria, demonstrando que além de eficientes estes sistemas podem ser aplicados sem um custo elevado. Nota-se que apesar de algo simples como um estacionamento, muitas são as possibilidades de inovar e criar aplicações para diferentes setores presentes nele, desde detecção de vagas, sistemas particulares com usuários cadastrados e sistemas para usuários não cadastrados. Todos possuem seus desafios e tecnologias utilizadas próprios e, com o crescimento da utilização de biometria facial em diversos setores no mercado, pode-se esperar que num futuro próximo existirão sistemas cada vez maiores e mais complexos de Estacionamentos Inteligentes tanto no Brasil quanto fora dele.

3 FUNDAMENTAÇÃO TEÓRICA

Atualmente no mercado, existem diversos sistemas de estacionamento tradicionais já muito bem consolidados, sendo o mais comum deles o que gera um *ticket* descartável com códigos de barras contendo as informações relacionadas ao tempo de permanência do usuário dentro do estacionamento. Outro sistema que é adotado em alguns locais é o que utiliza um cartão de plástico ao invés do *ticket*. Estes sistemas, apesar de consolidados no mercado, podem ocasionar problemas aos seus usuários, pois todo o sistema gira em torno dos dados presentes no *ticket* (ou cartão) de estacionamento que o usuário deve manter com ele o tempo inteiro, e apresenta-lo na saída para poder ser liberado. O *ticket* é basicamente um pequeno pedaço de papel, suscetível a perda, com a perda deste *ticket* o usuário poderia vir a enfrentar uma série

de problemas para poder comprovar que é o legítimo dono daquele veículo, o seu tempo de permanência no estacionamento, sua hora de chegada, etc.

Outro fator, ainda que mínimo é o fato deste pequeno *ticket* ser descartado após todo o processo, gerando acumulo desnecessário de lixo e não ser sustentável ao meio ambiente, bem como a tinta necessária para a impressão do código de barras. Existe ainda outro problema relacionado à tinta, caso o usuário fique por um tempo considerável no estabelecimento e coloque o *ticket* em algum local que ele fique constantemente exposto a outros materiais (chaves, celular, carteira, etc.) como uma bolsa por exemplo, poderá ocorrer o desgaste desta tinta, tornando a leitura do *ticket* difícil ou mesmo impossível. Além de problemas de má impressão do *ticket*, portanto a manutenção da impressora para trocas de cartucho deve ser constante.

No sistema que utiliza o cartão, como ele é reaproveitado na saída para depois ser reutilizado por usuários diferentes, não ocorre o problema do desperdício de papel e a utilização de tinta, porém ainda é insustentável ao meio ambiente, pois devem ser produzidos cartões em quantidade suficiente para todos os possíveis usuários do estacionamento e os cartões que vierem a apresentar problemas ou acabarem quebrados serão descartados. Caso o descarte desses cartões não seja feito da maneira correta, eles podem vir a poluir o meio ambiente.

Apesar de mínimos, os problemas existem para estes tipos de modelos de estacionamento. Utilizar biometria facial para automatizar todo o processo tornaria a necessidade de carregar *tickets* ou cartões desnecessária, o que já eliminaria todos os problemas citados acima, pois somente com a leitura facial do rosto do usuário todas as informações relacionadas ao seu tempo de permanência, data e hora de chegada estariam salvas dentro de um servidor, livrando o usuário de qualquer problema.

3.1 BIOMETRIA

3.1.1 Conceito de biometria, aplicações e importância

A biometria é amplamente utilizada em diversos tipos de aplicação, sendo a mais comum entre eles, e o objetivo deste trabalho, o controle de acesso. Na biometria, existem diversos tipos de técnicas diferentes, como reconhecimento pela digital ou mão, retina, voz, etc., mas todas possuem problemas que podem dificultar o reconhecimento de um usuário específico, seja pela epiderme danificada no reconhecimento pela digital, pela sensibilidade do dispositivo que captura a retina, pelo som ambiente que pode causar interferência no

reconhecimento pela voz, todas possuem algum problema que torna dificultosa a identificação de determinado usuário (Jain, 2007). Porém o reconhecimento facial traz um conjunto de informações que torna fácil a identificação de indivíduos e são mais difíceis de serem falsificados.

Caso atenda à alguns critérios básicos citados abaixo, qualquer característica fisiológica ou comportamental humana pode ser usada como característica biométrica (Clarke, 1994):

- Universalidade: Todos que serão autenticados deverão possuir essa característica;
- Unicidade: Essa característica deve ser única para cada indivíduo. Se duas ou mais pessoas possuírem a mesma característica deve ser nula ou desprezível;
- Permanência: A característica deve ser imutável, não pode se alterar com o tempo;
- Coleta: Deve ser possível mensurar essa característica por meio de um dispositivo;
- Aceitação: O indivíduo deve se sentir confortável na coleta dessa característica.

Com o passar dos anos, diversas tecnologias biométricas foram desenvolvidas. As tecnologias biométricas que existem atualmente são classificadas em dois grupos: características fisiológicas ou estáticas e características comportamentais ou dinâmicas. O primeiro grupo concentra traços fisiológicos, originários do DNA do indivíduo, com pouca variação ao longo do tempo, já foram citadas anteriormente: aparência facial, padrão da íris, geometria das mãos e impressões digitais. Outras características menos comuns também podem ser incluídas, como: impressão da palma, formato das orelhas, padrão vascular da retina, odor do corpo e o padrão da arcada dentária. Já o segundo grupo concentra características aprendidas ou desenvolvidas ao longo da vida e do uso constante e podem ser alteradas por vontade do indivíduo. As mais conhecidas são o padrão de voz e a assinatura. Outras menos comuns são: dinâmica de digitação, modo de andar (*gait recognition*), movimento labial, som da assinatura, vídeo da assinatura e imagens mentais (Costa *et al.*, 2006).

A biometria está intrinsecamente relacionada com segurança, mais especificamente na área de autenticação. Ela por sua vez é de suma importância, pois ao solicitar acesso a algum recurso, é verificado se o indivíduo que está solicitando este acesso possui autorização com base na sua identidade associada, após isso é concedida ou negada esta autorização. Ao solicitar o acesso a certo recurso, uma credencial precisa ser fornecida para autorizar seu acesso ou não, esta credencial é uma evidência fornecida por uma entidade, neste caso a biometria (Miller, 1994). Esses dados obtidos pela biometria são os traços únicos que cada ser humano possui,

eles são medidos e computados como um identificador biométrico único, sendo difícil forjar, roubar, compartilhar ou alterar (Costa *et al.*, 2006). Utilizando as informações obtidas com a biometria, é possível desenvolver aplicações com um nível de segurança maior e menos violável se comparado aos outros meios de segurança, como uma simples senha.

3.1.2 Biometria Facial

Parafraseando Silva e Cintra, (2015, p.01) "o reconhecimento facial é uma técnica que consiste em identificar padrões em características faciais, como formato da boca, do rosto, distância dos olhos, entre outros". O reconhecimento facial dispensa equipamentos de biometria especializados, sendo necessário apenas uma câmera para a identificação do usuário e autenticação do mesmo e um banco de dados, para armazenar as informações do rosto daquele usuário e seus dados agregados. Também é possível distinguir o usuário caso ele esteja usando acessórios (óculos, lentes de contato, brincos, piercings, etc.)

O conjunto de informações que se é obtido por meio do reconhecimento facial, é captado através de cálculos matemáticos que definem pontos e distâncias entre partes específicas do rosto humano (*landmarks*) que diferencia um indivíduo de outro. Através destas *landmarks*, é gerada uma espécie de máscara, diferente para cada indivíduo, independente dos acessórios utilizados (chapéu, óculos, brincos, piercings, etc.) e com essa máscara é feito o reconhecimento facial de um indivíduo específico. É possível inclusive reconhecer um indivíduo mesmo que ele tenha mudanças na sua aparência após o reconhecimento facial inicial (barba, mudança no corte de cabelo, entre outros) (Costa, 2006).

3.1.3 Métodos de Biometria Facial

Técnicas utilizando *deep learning* são cada vez mais populares dentro do campo de processamento de imagens, porém utilizar tais técnicas requer grande quantidade de dados para um melhor desempenho, além da maioria exigir *hardware* mais avançado e alto valor monetário. Porém técnicas que utilizam métodos clássicos de *machine learning* podem ter desempenho satisfatório com pequenas quantidades de dados, além de poder ser utilizadas em equipamentos menos custosos (Ahsan *et al.*, 2021).

Método Baseados em Características Locais (Nunes, 2017):

Local Binary Pattern Histogram: O LBP ou LBPH é uma ferramenta que trabalha como um descritor de texturas. O operador aloca um rótulo para cada valor de pixel que uma imagem possui, criando uma vizinhança de pixels com 3x3 de área ao redor do pixel inicial, o resultado obtido é um número binário (SÁNCHEZ, 2010).

Método Baseados em *Templates* (Lopes, 2005):

Cascade Classifier: Conhecido como classificador em cascata, consiste em uma lista de estágios onde cada nível cuida da análise de um conjunto de atributos diferentes, assim como avaliam se estes atributos representam ou não um objeto de interesse (Sander., 2014). Cada um desses estágios é composto por um ou mais classificadores fracos (*weak classifiers)* que são treinados geralmente por algoritmos baseados em métodos de *boosting*, até que atinjam uma taxa máxima de falsos positivos para definir a acurácia do classificador.

Métodos Baseados em imagem (Nunes 2017):

Eigenfaces: Técnica criada por Sirovich e Kirby para representar padrões encontrados em imagens de rostos através do método de Análise de Componentes Principais (PCA). É um conjunto de autovetores de uma matriz de covariância que é formada por imagens de rostos. Utiliza o método de distância Euclidiana para calcular a distância do *eigenvector* entre as *eigenfaces*. Quanto menor for essa distância, maior é a taxa de identificação do indivíduo, porém quanto maior for a distância, menor é a chance de os dois indivíduos serem a mesma pessoa (Ahsan *et al.*, 2021).

Redes Neurais Convolucionais: Conhecida também como CNN, é uma variação das redes de *Perceptrons* de Múltiplas Camadas, inspirada no processo biológico de processamento de dados visuais. Sendo amplamente utilizada em aplicações de classificação, detecção e reconhecimento de imagens e vídeos, a CNN pode aplicar filtros em dados visuais, mantendo a simetria entre os pixels próximos de sua vizinhança ao longo de todo o processamento da rede (Vargas *et al.*, 2016). As CNNs se baseiam tipicamente em três camadas: convolução, *pooling* e densa (YAMASHITA *et al.*, 2018).

Redes Neurais Artificiais: Conhecidas também como RNA, são modelos paramétricos não-lineares. Sendo uma técnica muito utilizada atualmente, por ser uma ferramenta capaz de lidar com os problemas complexos, que envolvem grandes massas de dados que devem ser modelados e analisados (Kovács, 2002). É capaz de detectar implicitamente qualquer relação

não-linear entre a variável resposta e as variáveis explicativas. Podem ser aplicadas em problemas de regressão, classificação e compactação de dados (Santos *et al.*, 2005).

3.2 APRENDIZADO DE MÁQUINA

O termo aprendizado de máquina se refere à detecção automatizada de padrões de dados significativos. Nas últimas décadas tornou-se uma ferramenta comum em quase qualquer tarefa que exija a extração de dados isolados de um grande número de amostras. Atualmente estamos cercados pela tecnologia de aprendizado de máquina, desde ferramentas de buscas que nos trazem os melhores resultados possíveis, baseados nas nossas pesquisas, até câmeras digitais que detectam rostos por meio de aplicações já instaladas. Além de ser utilizado em muitas aplicações nas áreas de bioinformática, medicina e astronomia (Shalev-Shwartz *et al.*, 2014).

Existem dois tipos principais de aprendizado de máquina: o aprendizado supervisionado e o não supervisionado. A diferença entre os dois está na forma como seu treinamento é realizado. O aprendizado supervisionado conta com uma variável de saída, conhecido como rótulo, que direciona a IA ao auxiliar quais dados estariam corretos ou incorretos dentro de um *dataset* durante a etapa de treinamento (Hastie *et al.*, 2009). Já o aprendizado não supervisionado não conta com este rótulo no seu treinamento e aprende por si próprio como diferenciar dados corretos e incorretos, por meio de associações de padrões semelhantes dentro de diversos dados diferentes em um grande *dataset.*

3.3 BIBLIOTECAS

3.3.1 Biblioteca dlib

Dlib é uma biblioteca *open source* voltada inicialmente para engenheiros e pesquisadores. Ela provê uma infinidade de soluções para o desenvolvimento de *machine learning* em diversas linguagens, o seu diferencial foi ser criada na linguagem C++. A dlib possui um *toolkit* completo de álgebra linear com suporte BLAS. Além disso possui implementações de algoritmo para inferência em redes *Bayesian* e métodos baseados em *kernel* para classificação, regressão, *clustering*, detecções anormais e diversas outras funcionalidades (King, 2009).

Dentro das ferramentas de *machine learning*, mais especificamente as voltadas para o reconhecimento facial, dois métodos de detecção facial se destacam:

- HOG + Linear SVM face detector: A função *get_frontal_face_detector* ao ser chamada, retorna o modelo pré-treinado de detecção facial HOG + Linear SVM que está incluso na biblioteca dlib. Por utilizar o método *Histogram of Oriented Gradients* (HOG) combinado com o *Linear Support Vector Machine (*SVM) esse modelo pré-treinado é capaz de alcançar resultados excelentes no treinamento de classificadores de objetos altamente precisos, neste caso, detectores humanos. Além da alta precisão, não é necessário a utilização de uma placa de vídeo dedicada, pois este método atua somente com o processador (CPU), tornando-o um método de baixo custo, preciso e de rápido processamento. Porém um dos defeitos deste método é o de não ser capaz de detectar rostos em posições muito diferentes da frontal, ou seja, há problemas com a rotação do rosto.

- Max-Margin (MMOD) CNN face detector: Enquanto o método anterior tem dificuldades com rotações do rosto, o método *Max-Margin CNN* atua com facilidade nesse quesito. Por ser um método construído em redes neurais, possui uma detecção facial muito mais robusta, para a utilização deste método a função *cnn_face_detection_model_v1* precisa ser chamada. Esse método chama o modelo pré-treinado *dlib_face_recognition_resnet_modelv1.dat.bz2* que é uma rede ResNet de 29 camadas. Essa rede foi treinada em um *dataset* com cerca de 3 milhões de rostos, este *dataset* é a junção de dois grandes *datasets: Face Scrub Dataset* e *VGG Dataset*, além de diversas outras imagens adquiridas através da internet. O modelo resultante após o treinamento possui erro médio de 0.993833, com desvio padrão de 0.00272732 no *benchmark Labeled Faces in the Wild.*

3.3.2 Biblioteca Face Recognition

A biblioteca *Face Recognition* foi construída utilizando o estado da arte da biblioteca dlib e possui acurácia de 99.38% no *benchmark Labeled Faces in the Wild*. Foi criada para facilitar a manipulação de elementos de biometria facial já existentes na biblioteca dlib e pode ser utilizada para reconhecer e manipular rostos humanos na linguagem Python ou por linha de comando, é conhecida como a biblioteca de reconhecimento facial mais simples do mundo. Possui diversos recursos que vão além da biblioteca dlib e outros já existentes nela, porém acessíveis de forma versátil. Com esta biblioteca é possível encontrar rostos em fotos e vídeos, encontrar e manipular características faciais como olhos, nariz, boca e queixo de fotos e vídeos. Também é possível identificar pessoas em fotos e vídeos ao atribuir um nome para uma lista de características ou *encodings* de um indivíduo específico ao utilizar funções em conjunto com a biblioteca *OpenCV* (Geitgey, 2018). A aplicação desenvolvida neste trabalho utiliza algumas funções desta biblioteca, são elas:

face_recognition.face_encodings(image),

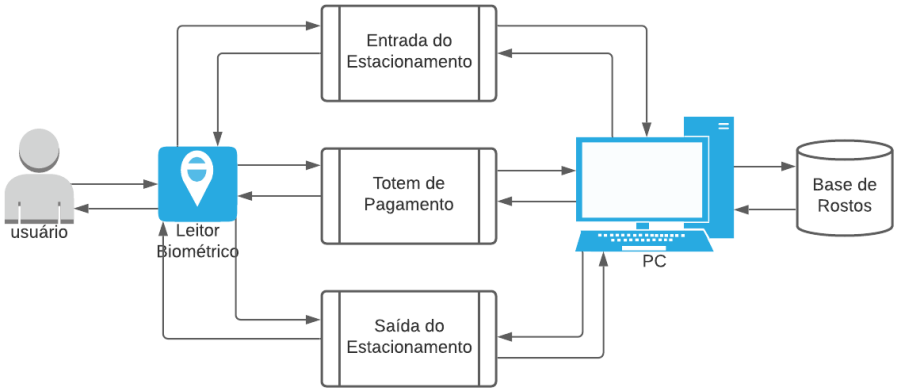
face_recognition.face_locations(image) e *face_recognition.face_distance(face_1, face_2).*

4 METODOLOGIA

O processo principal da aplicação proposta neste trabalho está ilustrado na Figura 1, que reutiliza o sistema de segurança já presente no ambiente. Nesta figura é possível observar que há uma troca de interações entre o usuário, o leitor biométrico (câmera de segurança) e os processos de entrada do estacionamento, totem de pagamento de saída do estacionamento.

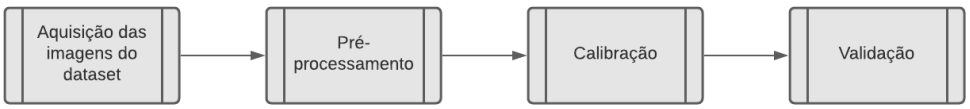
Já a Figura 2 ilustra o processo de teste com o *dataset* utilizado pela aplicação, pode-se observar que primeiro tem-se a etapa de obtenção das imagens do *dataset*, seguido da etapa de pré-processamento, após isso é feita a calibração do sistema com algumas fotos tiradas dentro do ambiente de teste e, para finalizar tem-se a etapa com a validação do modelo proposto. As etapas de aquisição das imagens e pré-processamento serão discutidas na Seção 5, porém as etapas de calibração e validação serão discutidas na Seção 6.

Figura 1 - Fluxo da principal entidade metodológica do trabalho



Fonte: Elaborado pelo Autor

Figura 2 - Diagrama da aquisição e operações utilizando o *dataset* de teste



Fonte: Elaborado pelo Autor

4.1 AMBIENTE

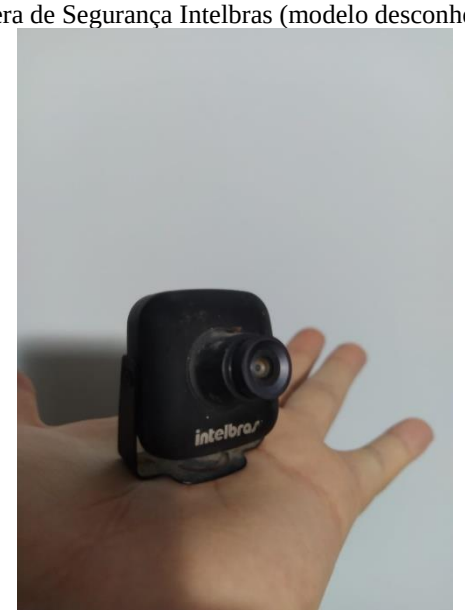
Para o desenvolvimento desta aplicação, foi utilizado somente um ambiente para testes. Durante estes testes, foram utilizados uma câmera analógica Intelbras em conjunto com

a *webcam* do computador para compilar a aplicação. A câmera analógica estava conectada diretamente à um DVR, de onde as imagens eram obtidas pela aplicação.

O ambiente de desenvolvimento, onde foram executadas as etapas de aquisição do *dataset*, acesso às câmeras de segurança, captura das imagens e classificação das imagens foi composto por um notebook Acer Nitro 5 com processador Intel®Core™ i7-7700HQ que utiliza frequência base de 2.80 GHz e 4 núcleos de processamento, com 16 GB de memória RAM de 2400 MHz. Como não há processo de treinamento do modelo por conta da aplicação utilizar funções da biblioteca Face Recognition, o ambiente utilizado não precisa possuir hardware avançado, porém quanto melhor a CPU mais rápido serão os processos. Pelo fato do modelo utilizado na aplicação ter sido o HOG, não é necessário o uso de placa de vídeo, porém a placa presente no ambiente é uma NVIDIA GeForce GTX 1050 com 4 GB de memória GDDR5 de 7000 MHz e 640 núcleos CUDA.

O sistema de segurança utilizado no ambiente e que foi reutilizado pela aplicação conta com um Gravador de vídeo digital (DVR) modelo VD 3004 com capacidade para quatro câmeras analógicas. Uma dessas quatro câmeras analógicas foi utilizada pela aplicação, porém todas foram testadas e são possíveis de serem utilizadas. A câmera utilizada para testes exibida na Figura 3 é de um modelo mais antigo que não se encontra no mercado, por conta das condições da câmera, não é possível identificar o modelo exato, mas é da linha de câmeras CFTV Intelbras e não possui infravermelho.

Figura 3 - Câmera de Segurança Intelbras (modelo desconhecido)



Fonte: Elaborado pelo Autor

4.2 DATASET

Por conta da condição de pandemia e isolamento social que o Brasil viveu na testagem da aplicação em 2021, foi necessário utilizar um *dataset* com um conjunto de usuários que simulariam clientes dentro do ambiente do trabalho. Seriam necessárias fotos de perfil de indivíduos em condições de luz semelhantes e mesmo enquadramento da foto para aumentar a acurácia na classificação das imagens. Portanto, como *dataset* foram utilizadas fotos únicas de 100 (cem) indivíduos diferentes escolhidos de forma aleatória do projeto da UFSC MIGMA. Desses 100 indivíduos, 32.075% são mulheres e 67.925% são homens, 5.66% são afrodescendentes e os 94.34% restantes englobam Euro-Americanos e outras etnias. As 100 amostras do *dataset* MIGMA possuem resolução de 1920x1080 *pixels* e, antes do pré-processamento, chegam a atingir 114 MB de dados. As amostras obtidas por meio da câmera de segurança possuem tamanho médio variando entre 20 e 30 KB.

4.3 PRÉ-PROCESSAMENTO

Como as 100 amostras do *dataset* contam com imagens em alta resolução e considerável tamanho, além de serem fotos sem o tratamento inicial de recorte da Região de Interesse (ROI). Foi necessário realizar um pré-processamento deste conjunto de imagens com o objetivo de reduzir a quantidade inicial de dados desta imagem, tornando-as mais leves e diminuindo o tempo de processamento das etapas subsequentes da aplicação, assim como eliminando as informações de fundo presentes nas fotos para evitar processamento desnecessário ou até mesmo reconhecimentos de padrão errôneos por parte da aplicação. A identificação da ROI destas imagens do *dataset* foi executada por uma aplicação à parte feita somente para este intuito, pois dentro da aplicação original deste trabalho, já existe um pré-processamento inicial das imagens que será discutido a seguir.

A identificação da ROI foi executada por meio do método Histogram of Oriented Gradients (HOG), implementado dentro da própria biblioteca *Face Recognition.* Após o recorte da ROI, as 100 amostras que antes possuíam 114 MB de tamanho total diminuíram para cerca de 21.8 MB de dados totais, uma redução de 80.88% no tamanho total.

A aplicação original, como já mencionado, possui em sua primeira etapa um pré-processamento onde é exibido a ROI em tempo real através de um monitor por onde o usuário

interage com o sistema. A identificação da ROI na aplicação original utiliza o mesmo método HOG, implementado dentro da própria biblioteca *Face Recognition*, o método HOG foi escolhido por sua eficácia e apesar de possuir menos acurácia para ângulos de rosto que não sejam frontais, o método é mais rápido em CPUs e, portanto, seu uso em uma aplicação que utiliza um sistema de câmeras com imagens ao vivo foi a melhor escolha. Além de que dentro do ambiente desta aplicação, o usuário do sistema deverá interagir com as câmeras de forma frontal, portanto outros ângulos que não sejam frontais não serão considerados. Outro método testado foi a CNN, porém ele exige que se tenha placa de vídeo (GPU) e CUDA por ser um modelo *deep learning* com aceleração gráfica e exigiria mais do hardware. Este método se provou, ineficaz, pois não conseguiu compilar de forma eficaz dentro da aplicação desenvolvida. Descobriu-se que para a utilização do método CNN para aplicações em tempo real é necessário possuir uma placa de vídeo (GPU) dedicada de nível muito superior à existente no ambiente utilizado para ser capaz de atingir a eficiência que se obtém atualmente com o método HOG.

4.4 A APLICAÇÃO

A aplicação é formada por três algoritmos que estarão atuando simultaneamente. Todos eles compartilham funções em comum que são herdadas do algoritmo principal, porém cada um deles difere do outro por possuir objetivos finais específicos, além de funções individuais próprias.

4.4.1 ParkingCamera – Função principal de aquisição das imagens

A primeira etapa de todos os três algoritmos compartilha uma função herdada que é comum para todas as classes filhas, a de capturar o rosto do indivíduo e salvá-lo temporariamente. Antes da obtenção da imagem da câmera, é feito uma conversão de BGR (*Blue Green Red)* que é o padrão utilizado pela biblioteca *OpenCV* para RGB (*Red Green Blue)* que é utilizado pela biblioteca *Face Recognition.* Após isso, utilizando a biblioteca *Face Recognition* e o método HOG, obtêm-se a ROI do rosto do usuário que irá aparecer em tempo real no monitor.

Para usuários que nunca utilizaram o sistema antes, em busca de evitar falsos positivos como fotos tiradas por descuido do mesmo ao ficar de frente para a câmera enquanto lê as instruções do sistema, um conjunto de regras que devem ser seguidas foram implementadas:

Com a ajuda da biblioteca *OpenCV* foi criado uma área que está centralizada no meio da filmagem da câmera que ocupa um espaço de 40% de altura e 30% de comprimento da região total oferecida pela câmera. Portanto, a ordem a ser seguida é a de que o sistema somente irá reconhecer que o usuário está pronto para ter a sua fotografia tirada caso o mesmo permaneça com a ROI do rosto completamente dentro desta área. Após cumprir essa demanda, o sistema entenderá que o usuário deseja de fato adentrar no estabelecimento. Então através da biblioteca *OpenCV* e utilizando a última ROI que estava presente na área especificada da filmagem, será salvo um recorte desta ROI em uma variável temporária dentro da aplicação.

A imagem a seguir demonstra esse processo: na primeira amostra da Figura 4, a ROI do rosto do usuário está fora da área desejada, portanto o sistema entende que o usuário ainda não está pronto para tirar a foto. Já na segunda amostra, a ROI do rosto do usuário está dentro da área desejada e o recorte da mesma é feito e salvo no disco, ao mesmo tempo que a cancela simulada foi aberta (área retangular na cor amarela para indicar cancela aberta).

Figura 4 - Aquisição da Imagem

Fonte: Elaborado pelo Autor

4.4.2 GateCamera – Função principal de extração das características

Nesta segunda etapa do processo que é compartilhada apenas pelas classes filha *TotemCamera* e *ExitCamera*, os eventos próprios dela ocorrem após os passos de aquisição da imagem que foram descritos na Seção 4.4.1 acontecerem e a ROI do usuário ser salva em uma variável temporária. Após estes eventos, ocorrerá então uma busca pelo usuário que teve a foto tirada. Através de uma indicação de entrada do estacionamento situada no nome de todas as

fotos de todos os usuários que entraram no estacionamento é possível saber quem ainda não pagou pelo estacionamento. Com isso, a busca irá ocorrer somente para usuários com essa indicação de não pagante, essa busca é realizada utilizando Regex da biblioteca *re* combinadas com regras de caminho do sistema da biblioteca *os*.

A busca mencionada acima ocorre por conta da extração de uma lista de 128 *face encodings* que é obtida ao chamar a função *face_recognition.face_encodings(faces)*, essa lista pode ser visualizada na Figura 5.

Figura 5 - Extração de Características

Fonte: Elaborado pelo Autor

Com essa lista de 128 características únicas da foto deste usuário específico, é feito uma comparação utilizando a lista deste usuário com a lista de 128 características de cada foto de outro usuário presente dentro do banco de rostos que foi identificado como não pagante.

4.4.3 GateCamera – Função principal de Distância Euclidiana

Nesta terceira e última etapa do processo que é compartilhada apenas pelas classes filha *TotemCamera* e *ExitCamera*, após os passos descritos na Seção 4.4.2 de busca e extração da lista de 128 características, é realizado a comparação entre as listas através do cálculo de Distância Euclidiana comparando a distância das 128 características da foto do usuário atual com os valores das distâncias das 128 características da foto de cada um dos outros usuários não pagantes do banco de rostos. O cálculo da Distância Euclidiana é feito chamando a função:

face_recognition.face_distance(face_1, face_2),
onde *face_1* é a lista de características da foto do usuário atual e *face 2* é a lista de características de cada foto de outro usuário dentro do banco de rostos.

Em seguida, terminado a comparação da Distância Euclidiana com a lista de *encodings* de todos os outros usuários, é escolhido o menor valor de distância dentre todos os valores obtidos durante a comparação. Através dos cálculos de *threshold* obtidos na etapa de calibração que será explicada na sessão 5, o sistema entenderá que este menor valor de distância obtido, caso seja menor que 0.40, corresponde de fato à foto inicial que este usuário tirou na entrada do estacionamento, com isso, é confirmado que ambas as fotos tratam do mesmo usuário. Caso o menor valor de distância encontrado durante as comparações seja superior a 0.40, o sistema entenderá que o usuário atual não possui uma foto de entrada no estacionamento e informará que o usuário não foi encontrado.

4.4.4 Entrada do Estacionamento

4.4.4.1 EntranceCamera - Classe filha

Os eventos ocorridos que são citados na Seção 4.4.1 são referentes ao algoritmo mãe e suas funções são herdadas por esta classe. A função que é específica desta classe será descrita na sequência. Este algoritmo trata da entrada do usuário no estabelecimento e a sua passagem pela cancela de entrada.

4.4.4.2 Aquisição da Imagem

Após o recorte da ROI do rosto do usuário no último frame da filmagem, durante toda a operação descrita na Seção 4.4.1, a aplicação de entrada irá acessar a variável temporária onde foi salvo o recorte da ROI e, através da função *inwrite* da biblioteca *OpenCV,* irá salvar no disco rígido do sistema em uma pasta especifica este recorte da ROI do usuário. Este recorte do rosto do usuário será salvo com o horário e a data atual do sistema, através da biblioteca *Date Time* da própria linguagem de programação *Python,* além de uma indicação de que esta foto se trata de um usuário da entrada do estacionamento, informando ao sistema que até o momento não houve o pagamento da tarifa para este usuário específico. Após a foto do usuário ser salva no disco, a cancela será aberta indicando que o usuário pode adentrar no estabelecimento.

4.4.5 Totem de Pagamento

4.4.5.1 TotemCamera – Classe filha

Os eventos ocorridos que são citados na Seção 4.4.1 e 4.4.3 são referentes ao algoritmo mãe e suas funções são herdadas por esta classe. A função que é específica desta classe será descrita logo a seguir. Este algoritmo trata do processo de pagamento que o usuário realiza em um totem de pagamentos presente dentro do estabelecimento.

4.4.5.2 Confirmação do pagamento

A busca realizada pelo sistema nesta etapa ocorrerá somente entre os indivíduos que possuírem a indicação de usuário não pagante, para evitar realizar uma busca desnecessária com usuários que já pagaram pelo estacionamento e estão dentro dos 15 minutos de tolerância. A indicação é extraída com a ajuda da biblioteca *re*. Após confirmação de que é o mesmo usuário em ambas as fotos (valor de distância menor que 0,40), tanto no totem de pagamento quanto na entrada do estacionamento, obtidas através do menor valor de Distância Euclidiana, é extraída a hora de entrada do estacionamento da foto obtida na entrada do estacionamento e é feito o cálculo de quanto deve ser pago pelas horas permanecidas com base em uma subtração do horário atual com o horário de entrada. Após realizado o pagamento, a foto inicial do usuário terá o horário alterado para o horário em que foi realizado o pagamento e será adicionado ao nome da foto uma indicação que diz ao sistema que este usuário específico realizou o pagamento e a partir de agora é um usuário pagante. As operações de alteração do horário e acréscimo da indicação são realizadas através da função *os.rename* da biblioteca *os* em conjunto com a biblioteca *Date Time*. Ao terminar de pagar, o usuário receberá uma mensagem na tela do totem avisando que ele terá cerca de 15 minutos para sair do estacionamento antes de perder a garantia.

Para valores de Distância Euclidiana maiores que 0,40 o sistema entenderá que o usuário atual que está tentando pagar não possui uma foto semelhante tirada na entrada do estacionamento e indicará ao mesmo que o usuário não foi encontrado e, em caso de engano, solicitará ajuda do estabelecimento.

4.4.6 Saída do Estacionamento

4.4.6.1 ExitCamera – Classe filha

Os eventos ocorridos que são citados na Seção 4.4.1 e 4.4.3 são referentes ao algoritmo principal e suas funções são herdadas por esta classe. A função que é específica desta classe será descrita na sequência. Este algoritmo trata da saída do usuário do estabelecimento e a sua passagem pela cancela de saída.

4.4.6.2 Checagem do horário

A busca realizada pelo sistema nesta etapa ocorrerá somente entre os indivíduos que possuírem a indicação de usuário pagante, para evitar realizar uma busca desnecessária com usuários que ainda não pagaram. A indicação é extraída com a ajuda da biblioteca *re*. Após confirmação de que é o mesmo usuário em ambas as fotos obtidas através do cálculo de Distância Euclidiana (valor de distância menor que 0,40), é extraída a hora de entrada do estacionamento da foto obtida durante os passos da Seção 4.4.4 (Entrada do Estacionamento) e é feito uma checagem do tempo decorrido desde o pagamento realizado no totem de pagamento até a saída do usuário na cancela de saída, este tempo decorrido deverá estar dentro dos 15 minutos de tolerância e é feito com a ajuda da biblioteca *Date Time* em conjunto com a extração dos valores através da biblioteca *re*.

Caso a foto do usuário não seja encontrada entre as fotos de todos os outros usuários com a indicação de pagante no banco de rostos, o sistema entenderá que este usuário atual ainda não realizou o pagamento e exibirá uma mensagem indicando que o usuário retorne ao estabelecimento para fazer o pagamento corretamente ou solicitar ajuda do estabelecimento em caso de engano.

Os cálculos realizados para o sistema reconhecer que o usuário está ou não dentro da tolerância de 15 minutos são feitos com a ajuda da biblioteca *re* para extração dos valores de horas presentes na foto e *Date Time* para obtenção do horário atual. Caso esteja dentro do limite de 15 minutos, a cancela irá abrir e o usuário poderá ir embora. Caso tenha ultrapassado o limite de 15 minutos, a cancela não abrirá e a foto do usuário será movida novamente para o banco de rostos e receberá uma indicação de que este usuário não pagou, devendo este novamente pagar pela diferença para então poder ir embora. Na situação onde o usuário ultrapassa os 15 minutos de tolerância, aparecerá no monitor que ficará próximo à cancela uma mensagem que dirá que

o usuário ultrapassou o tempo de tolerância e que ele deverá retornar e pagar novamente a diferença.

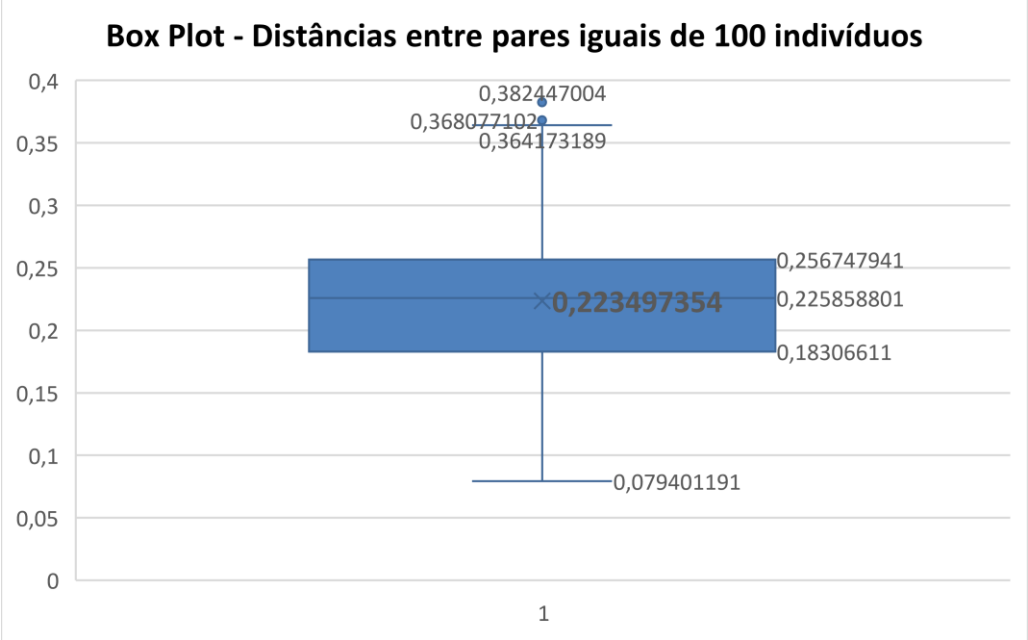
5 CALIBRAÇÃO, VALIDAMENTO E RESULTADOS EXPERIMENTAIS

5.1 CALIBRAÇÃO DA APLICAÇÃO NO AMBIENTE

Através da análise dos valores obtidos com o cálculo da Distância Euclidiana foi possível definir um *threshold* ou um limiar para a aplicação não escolher pessoas semelhantes, mas diferentes enquanto busca pelo par da foto do usuário atual no banco de rostos. Como a aplicação é dependente da iluminação do ambiente e do posicionamento das câmeras para pegar um bom ângulo frontal das pessoas, é necessário fazer pequenos ajustes de calibração com algumas amostras de usuários e com isso definir o limiar.

Foram feitos dois testes utilizando os valores de distância entre os 100 indivíduos do *dataset.* No primeiro teste os 100 indivíduos foram separados em 100 pastas diferentes, onde cada uma das pastas possuía um par de fotos de um mesmo indivíduo, foi realizado então o cálculo da Distância Euclidiana em cada uma das 100 pastas entre cada par de fotos iguais dos 100 indivíduos, com isso foi possível obter 100 amostras de distâncias de indivíduos que são iguais entre si. Por fim, foi gerado um gráfico *Box Plot*, conhecido também como diagrama de caixa onde esses 100 valores de distância foram acrescentados, formando o gráfico ilustrado na Figura 6.

Figura 6 - Box Plot com as distâncias de 100 pares de indivíduos iguais entre si



Fonte: Elaborado pelo Autor

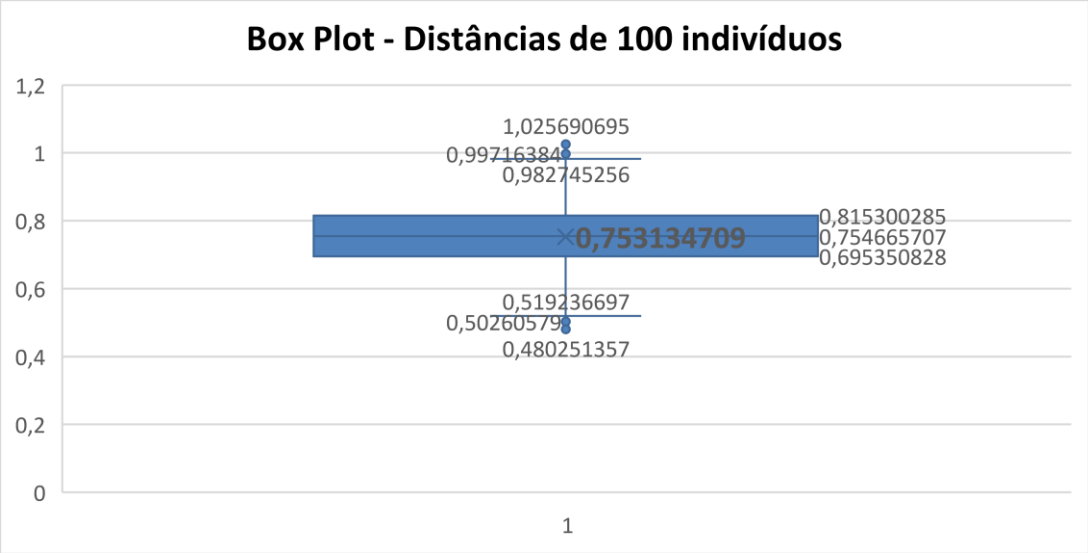
Com o gráfico da Figura 6, referente ao *box plot* das distâncias entre os 100 indivíduos que possuíam pares iguais, obteve-se uma mínima de 0.079401191 e uma máxima de 0.364173189, além disso percebe-se também dois valores de pontos fora da curva, ou *outliers*, sendo o maior deles no valor de 0.382447004. Com todos esses valores em mãos, é possível fazer uma análise do *threshold* entre fotos de usuários iguais. Com os valores do mínimo e do máximo, percebe-se que caso o sistema compare duas fotos de dois indivíduos potencialmente diferentes, e que a distância entre os valores de suas listas de *enconding* esteja entre 0 e 0.364173189, o sistema poderia com alto grau de certeza afirmar que os dois potenciais indivíduos na verdade são a mesma pessoa. Porém, não é possível desconsiderar o maior valor que existe nos *outliers*, pois esse valor por maior que seja, foi uma distância entre dois indivíduos iguais. Portanto, pode-se considerar que valores de distância que se encontram entre 0 e 0.38244 são de fato referentes à um mesmo indivíduo.

Um valor de distância 0 é muito improvável de acontecer, porém idealmente se existir uma distância 0 entre dois indivíduos, eles seriam a mesma pessoa indubitavelmente.

Como foi descoberto a mínima das distâncias entre indivíduos que são a mesma pessoa através do primeiro teste, no segundo teste também foi necessário obter essa distância mínima, porém para indivíduos que não eram a mesma pessoa. O teste foi feito colocando 100 fotos de indivíduos diferentes entre si em uma única pasta e após isso foi feito o cálculo da Distância

Euclidiana entre cada um deles. O segundo teste levou muito mais tempo quando comparado ao primeiro, pois para cada indivíduo dentro da pasta, haveria 99 valores de distância entre eles, após algumas horas de compilação foi possível obter 4950 amostras de distâncias entre cada um dos 100 indivíduos de teste. Assim como no primeiro teste, estas 4950 amostras de distâncias foram utilizadas para gerar um gráfico *Box Plot*, representado o gráfico da Figura 7.

Figura 7 - Box Plot com distâncias de 100 indivíduos diferentes



Fonte: Elaborado pelo Autor

A partir do gráfico apresentado na Figura 7 referente ao *Box Plot* das distâncias entre os 100 indivíduos diferentes, obteve-se uma mínima de 0.519236697 e uma máxima de 0.982745256, além disso percebe-se também dois valores de pontos fora da curva, ou *outliers*, sendo o menor deles no valor de 0.480251357. A partir do gráfico também é possível perceber que o valor da mediana é 0.753134709.

Com todos esses valores em mãos, é possível fazer uma análise do *threshold* para fotos de usuários que não são a mesma pessoa. Com os valores da máxima e da mínima, percebe-se que caso o sistema compare duas fotos de dois indivíduos potencialmente diferentes, e que a distância entre suas listas de *enconding* esteja entre 1 e 0.519236697, o sistema poderia com alto grau de certeza afirmar que os dois potenciais indivíduos são de fato pessoas diferentes.

Porém, não é possível excluir o menor valor que existe nos *outliers*, pois esse valor por menor que seja, foi uma distância entre dois indivíduos diferentes. Portanto, pode-se considerar que valores de distância que se encontram entre 1 e 0.48025 são de fato referentes à

dois indivíduos diferentes. Um valor de distância 1 é difícil de acontecer, porém idealmente se existir uma distância 1 entre dois indivíduos, eles seriam pessoas diferentes indubitavelmente.

Para finalizar, a Tabela 1 mostra para quais valores de distância mínima o sistema foi configurado. Um arredondamento foi feito para garantir uma margem de erro segura tanto para valores de distância mínima entre dois usuários iguais quanto para dois usuários diferentes entre si. Onde x é a distância do usuário atual da aplicação:

Tabela 1 - Limiares do reconhecimento facial da aplicação

Para dois indivíduos que são diferentes entre si	**Para dois indivíduos que são a mesma pessoa**
$x > 0.40$	$0 \leq x \leq 0.40$

Fonte: Elaborado pelo Autor

5.2 VALIDAÇÃO E RESULTADOS EXPERIMENTAIS DA APLICAÇÃO

O objetivo deste trabalho foi analisar a viabilidade da implementação da aplicação criada em um estabelecimento real, para isso foi feito a seguinte simulação para visualizar o desempenho (em segundos) que a aplicação apresenta com quantidades variadas de usuários presentes no banco de rostos que se encontra na Tabela 2.

Tabela 2 - Simulação de entrada e saída de clientes do estabelecimento em um dia útil

Hora Atual	**Nº Total de pessoas no estabelecimento**	**Desempenho Totem de Pagamento (s)**	**Desempenho Cancela de Saída (s)**	**Total (s)**
8:00 AM	3	2,56	2,29	4,85
8:30 AM	6	4,89	4,38	9,27
9:00 AM	9	6,5	6,64	13,14
9:30 AM	12	7,29	8,21	15,5
10:00 AM	15	9,48	8,65	18,13
10:30 AM	18	11,33	11,85	23,18
11:00 AM	21	14,02	12,66	26,68
11:30 AM	24	13,9	12,8	26,7
12:00 AM	27	16,33	16,45	32,78

12:30 PM	30	17,38	16,53	33,91
1:00 PM	40	21,52	22,87	44,39
1:30 PM	33	18,03	19,01	37,04
2:00 PM	36	21,59	21,55	43,14
2:30 PM	39	21,21	21,1	42,31
3:00 PM	42	23,42	24,3	47,72
3:30 PM	45	25,32	25,3	50,62
4:00 PM	48	25,94	25,8	51,74
4:30 PM	50	27,27	27,01	54,28
5:00 PM	55	29,07	29,1	58,17
5:30 PM	60	31,62	31,8	63,42
6:00 PM	65	34,57	35,6	70,17
6:30 PM	70	37,19	36,95	74,14
7:00 PM	75	40,18	39,12	79,3
7:30 PM	80	42,43	43,02	85,45
8:00 PM	85	45,14	44,2	89,34
8:30 PM	100	54,05	51,81	105,86
9:00 PM	69	37,04	37,05	74,09
9:30 PM	39	21,97	21,87	43,84
10:00 PM	19	11,58	12,05	23,63
10:30 PM	1	2,78	2,26	5,04

Fonte: Elaborado pelo Autor

Com os valores detalhados nesta tabela, percebe-se que, para uma quantidade maior de usuários dentro do estacionamento do estabelecimento ao mesmo tempo, há um aumento no tempo de processamento em dois dos três algoritmos da aplicação. Esse aumento se deve pelo fato de que para cada usuário que for realizar o pagamento através do totem de pagamento ou sair pela cancela de saída (considerando que dentro do banco de rostos a quantidade de indivíduos existentes seja muito grande) será feito um cálculo da Distância Euclidiana comparando o usuário atual com os que já estão dentro do banco de rostos, logo, quanto maior for a quantidade de usuários maior será a quantidade de cálculos de distância que serão feitos para cada indivíduo.

Ainda, de acordo com a tabela, caso o estabelecimento tenha um dia cheio e o estacionamento tenha 100% das vagas ocupadas (neste cenário, às 08:30 PM), a demora para o pagamento ser processado ou a cancela ser levantada na saída varia de 51.81 segundos até 54.05 segundos. Os tempos de processamento também variam para uma mesma quantidade de indivíduos, isso se deve ao fato da Distância Euclidiana variar entre cada comparação, podendo ser mais rápida ou lenta dependendo de cada indivíduo comparado, pois para 100 indivíduos diferentes, existirão 100 valores de distância mais altos ou mais baixos e os indivíduos que saíram e entraram novamente no estabelecimento podem não ter sido os mesmos, pois foram escolhidos arbitrariamente.

Uma solução para diminuir o tempo de processamento que o sistema leva para calcular a Distância Euclidiana seria diminuir o tamanho das fotos, como teste foi utilizado o *dataset* do projeto MIGMA e suas fotos mesmo após o pré-processamento ainda são muito mais pesadas quando comparadas às fotos obtidas pela câmera de segurança. Uma foto do *dataset* varia entre 100 a 400 KB, enquanto que uma foto obtida pela câmera varia de 20 a 30 KB.

Outra solução, porém, que iria contra um dos fundamentos deste trabalho, seria a obtenção de um ambiente mais robusto e moderno tanto em *hardware* do servidor central que executa a aplicação, quanto em sistema de câmeras que poderia agilizar esses cálculos e inclusive aumentar a precisão da biometria facial ao utilizar o modelo de redes neurais CNN, ao invés do modelo atual HOG.

Outro teste realizado foi de acurácia, para analisar a porcentagem de acertos que a aplicação criada obteria através da biblioteca *Face Recognition*. Por conta da falta de usuários para poder interagir com a aplicação pessoalmente, foi decidido criar uma simulação que reproduzisse essa interação.

Foram impressas 100 fotos (diferentes das existentes dentro do *dataset)* das 100 amostras que já estavam presentes no *dataset* utilizado e, como o objetivo deste teste era somente a acurácia da aplicação e não o tempo de processamento, foi criada uma simulação sintética onde todos estes usuários utilizariam o totem de pagamento um a um, porém após obterem a indicação de usuários pagantes, eles voltariam a possuir a indicação de usuários não pagantes.

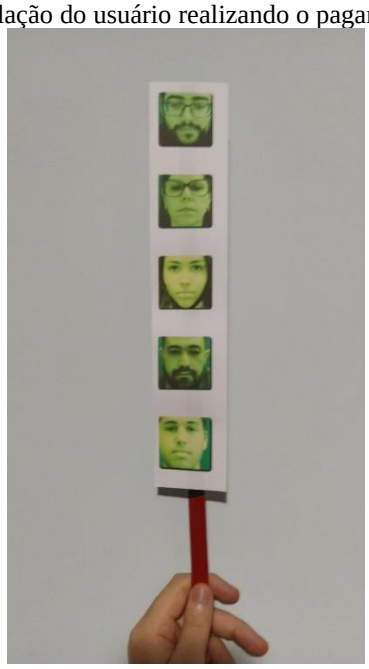
Dessa forma sempre que houvesse a comparação da distância euclidiana da lista de características da foto do usuário interagindo com o totem com a distância das amostras

presentes no banco de rostos, haveria sempre um cálculo da distância entre todos os outros 100 indivíduos não pagantes presentes dentro do banco de rostos.

Resumidamente cada um dos 100 usuários teria sua similaridade testada dentro do banco de rostos com outros 99 outros usuários também presentes, com o objetivo de verificar se a aplicação é capaz de encontrar um usuário específico em uma situação onde o banco de rostos se encontra na capacidade máxima de lotação do ambiente criado.

Como teste, cada uma das fotos adquiridas através do *dataset* MIGMA (Matias *et al.*, 2021) foi presa em um longo bastão e uma a uma as fotos foram exibidas na frente da câmera do totem de pagamento como se fossem usuários que de fato estariam tentando pagar pelas horas permanecidas no estacionamento. A Figura 8 ilustra o resultado final.

Figura 8 – Simulação do usuário realizando o pagamento



Fonte: Fotos do *dataset* MIGMA

Um ponto a ser destacado é que por conta de problemas técnicos, a qualidade de impressão das fotos acabou ficando debilitada, como é possível observar na Figura 9.

Figura 9 - Má qualidade da impressão

Fonte: Fotos do *dataset* MIGMA

Porém, apesar da má qualidade da impressão, a aplicação se saiu muito bem ao obter a menor Distância Euclidiana entre a foto impressa que estava sendo apresentada para a câmera e cada uma das fotos em alta qualidade presentes no banco de rostos. Relembrando a composição étnica dos indivíduos presentes no *dataset* utilizado: 100 indivíduos onde 32.075% são mulheres e 67.925% são homens, 5.66% são afrodescendentes e os 94.34% restantes englobam Euro-Americanos e outras etnias. A aplicação se mostrou correta durante a primeira tentativa de comparação ao afirmar que dois indivíduos eram a mesma pessoa em 96 das 100 amostras testadas, 2 das 4 amostras foram identificados como a mesma pessoa na segunda tentativa de comparação, a terceiro das quatro amostras foi identificado como a mesma pessoa depois de 3 tentativas e a última das quatro amostras não conseguiu ser identificado mesmo após diversas tentativas.

Após os testes, algumas conclusões puderam ser feitas:

A amostra que não pôde ser identificada no banco de rostos não importando o número de tentativas de comparação pertencia ao sexo feminino e englobava a etnia Euro-Americana, se considerar somente a taxa de acertos, a aplicação obteve 99 acertos dos 100 totais, portanto uma possível conclusão para este caso seria uma combinação da pele branca da usuária com a má-qualidade da impressão da foto, pois ao avaliar a foto de forma aproximada, nota-se que

algumas partes do rosto perderam profundidade dando impressão de brilho excessivo e talvez por isso a identificação não pôde ser realizada pela biblioteca, infelizmente não foi possível obter uma foto impressa em qualidade superior à utilizada em teste.

A terceira amostra, que foi identificada após 3 tentativas de comparação coincidentemente possuí um modelo de óculos muito parecido com outro indivíduo (que não faz parte do *dataset*) que chegou a testar a aplicação previamente e, curiosamente os mesmos erros aconteceram com ambos e a única similaridade entre eles é o formato dos óculos. Levando a crer que óculos com certos formatos podem vir a atrapalhar a identificação, não houve nenhum problema semelhante em nenhuma outra amostra que utilizava óculos dentro do *dataset* ou fora dele.

Outra característica que pôde ser vista durante os testes foi que o enquadro do rosto por parte do usuário, na área especificada pela aplicação, deve ser bem posicionado, pois em duas vezes que não houve bom enquadramento do rosto na área, os indivíduos não puderam encontrar seu par associado dentro do banco de rostos, porém após melhor enquadramento, eles foram encontrados em todas as outras tentativas.
Outras observações gerais podem ser vistas na Tabela 3.

Tabela 3 - Observação gerais do teste de acurácia

Características das fotos	**Indivíduo encontrou o seu par associado**	**Indivíduo não encontrou o seu par associado**
Inclinações de até 45°	X	
Fotos borradas ou distorcidas		X
Rosto muito iluminado		X
Rosto com caretas	X	
Mudanças no penteado	X	
Óculos de grau	X	
Indivíduo muito distante		X
Pequenas poluições na área de posicionamento (parte do cenário por trás do indivíduo)	X	

Fonte: Elaborado pelo Autor

6 CONCLUSÃO E TRABALHOS FUTUROS

Os avanços na tecnologia, tanto em termos de *hardware* quanto das técnicas baseadas em redes neurais têm proporcionado progresso em diversas áreas onde análise de dados ou inteligência artificial são essenciais. As soluções que já existem também passam por essa metamorfose, pois novas técnicas e tecnologias são aplicadas a elas, possibilitando que novos patamares sejam explorados. Neste trabalho foi apresentado uma aplicação de um sistema para automatizar estacionamentos inteligentes por meio de reconhecimento facial através de soluções simples, porém robustas e com alto nível de acurácia.

Em termos de acurácia, a aplicação apresentou resultados excelentes. Durante a etapa de teste de acurácia houve somente uma única comparação entre dois indivíduos que não foi realizada, ou seja, dentro do ambiente de testes fechado, com o enquadramento certo, boa luminosidade e em uma simulação sintética, realizado com o *dataset* contendo os 100 indivíduos já citados há uma acurácia de 99% ao calcular o menor valor de Distância Euclidiana entre dois indivíduos idênticos (considerando valores totais). Na simulação realizada, o tempo de processamento também foi bastante satisfatório, pois uma situação onde as vagas ocupadas no estacionamento alcancem a capacidade total dificilmente ocorreria todos os dias. Portanto, o sistema em situações normais funcionaria de forma eficaz com variações de tempo entre 21,52 e 22,87 segundos para até 40 indivíduos diferentes no estacionamento do estabelecimento.

O projeto construído pode ser utilizado em inúmeras situações diferentes, não se limitando a estacionamentos. Podendo ser implementado em qualquer local que necessite de controle de fluxo de pessoas, como hospitais e condomínios por exemplo, podendo inclusive vir a substituir sistemas de controle de ponto em empresas. É possível também aperfeiçoar o sistema atual, não somente com *hardware* mais avançado, podendo também incluir um banco de dados na nuvem e tornar o gerenciamento do estacionamento remoto. Em um ambiente de educação, poderia ser utilizado para automatizar o sistema de presença dos alunos em cada sala, realizando um cadastro das aulas que serão realizadas em todos os períodos do semestre e os alunos que frequentarão estas aulas, poupando os professores de realizar tal tarefa, aumentando a duração da aula e a produtividade. Outra ideia seria a integração de um sistema de cadastro para clientes do estabelecimento com a aplicação feita, com isso o sistema poderia oferecer ofertas personalizadas para qualquer cliente com base nas compras realizadas por ele. Respeitando todas as leis de segurança da nova lei da LGPD (Lei Geral de Proteção aos Dados).

Resumindo, a aplicação criada neste trabalho foi capaz de alcançar altíssimos níveis de precisão, o custo para a implementação também é muito baixo quando comparado aos sistemas tradicionais de *tickets* por código de barras, pois além da economia ao adquirir os equipamentos, há também a economia em tinta e papel, contribuindo para um meio ambiente sustentável. Seu desempenho em velocidade de processamento acabou pecando para situações adversas, porém com as melhorias citadas seria possível tentar corrigir esse fator.

O código da aplicação na integra pode ser encontrada no GitHub, no seguinte endereço: https://github.com/sid1689/parking_lot_gate_final

REFERÊNCIAS

AHSAN, Md Manjurul et al. Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions. **Technologies**, v. 9, n. 2, p. 31, 2021.

CANAL, Felipe Zago et al. Reconhecimento de Expressão Faciais Baseado em RedesNeurais Convolucionais para Aplicação no Sistema TutorInteligente MAZK. 2021.

CHUNG, Yi-Nung et al. Applying Real Time Image Recognition Technology to Apartment Parking Lot Security System. In: **Proceedings of the Annual Conference of Biomedical Fuzzy Systems Association 28**. Biomedical Fuzzy Systems Association, 2015. p. 273-276.

CLARKE, Roger. Human identification in information systems: Management challenges and public policy issues. **Information Technology & People**, 1994.

COSTA, Luciano R.; OBELHEIRO, Rafael R.; FRAGA, Joni S. Introdução á biometria. **Livro-texto de Minicursos-VI SBSeg**, 2006.

DELIBALTOV, Diana et al. Parking lot occupancy determination from lamp-post camera images. In: **16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)**. IEEE, 2013. p. 2387-2392.

DIYASA, G. S. M. et al. Multi-face Recognition for the Detection of Prisoners in Jail using a Modified Cascade Classifier and CNN. In: **Journal of Physics: Conference Series**. IOP Publishing, 2021. p. 012005.

ESPINOSA, Paul et al. Vehicle Security and Alert System, Based on Facial Recognition and GPS Location. In: **2019 International Conference on Information Systems and Computer Science (INCISCOS)**. IEEE, 2019. p. 222-229.

GEITGEY, Adam. **face_recognition Github Library**. 2018. Disponível em: https://github.com/ageitgey/face_recognition. Acesso em: 05 set. 2021.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. Unsupervised learning. In: **The elements of statistical learning**. Springer, New York, NY, 2009. p. 485-585.

INDIRA, DNVSLS; SUMALATHA, L.; MARKAPUDI, Babu Rao. Multi Facial Expression Recognition (MFER) for Identifying Customer Satisfaction on Products using Deep CNN and Haar Cascade Classifier. In: **IOP Conference Series: Materials Science and Engineering**. IOP Publishing, 2021. p. 012033.

JAIN, Anil K.; FLYNN, Patrick; ROSS, Arun A. (Ed.). Handbook of biometrics. Springer Science & Business Media, 2007.

KING, Davis E. Dlib-ml: A machine learning toolkit. **The Journal of Machine Learning Research**, v. 10, p. 1755-1758, 2009.

KOVÁCS, Zsolt László. **Redes neurais artificiais**. Editora Livraria da Fisica, 2002.

LOPES, Eduardo Costa; BINS FILHO, José Carlos; NO, RELATÓRIO TÉCNICO. Detecção de faces e características faciais. **Porto Alegre**: PUCRS, 2005.

MAHMOOD, Zahid et al. A parallel framework for object detection and recognition for secure vehicle parking. In: **2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems**. IEEE, 2015. p. 892-895.

MATIAS, Jhennifer Cristine et al. MIGMA: The Facial Emotion Image Dataset for Human Expression Recognition. In: **CIARP25 Porto 25th Iberoamerican Congress on Pattern Recognition**, p. 88-98, 2021.

MILLER, Benjamin. Vital signs of identity [biometrics]. **IEEE spectrum**, v. 31, n. 2, p. 22-30, 1994.

NUNES, Fernanda Todesco et al. Técnicas de biometria baseadas em padrões faciais e sua utilização na segurança pública. 2017.

PERSADA, Reivind P. et al. Automatic face and VLP's recognition for smart parking system. **Telkomnika (Telecommunication Computing Electronics and Control)**, v. 17, n. 4, p. 1698-1705, 2019.

SÁNCHEZ LÓPEZ, Laura. Local Binary Patterns applied to Face Detection and Recognition. 2010.

SANTOS, Alcione Miranda dos et al. Usando redes neurais artificiais e regressão logística na predição da hepatite A. **Revista Brasileira de Epidemiologia**, v. 8, p. 117-126, 2005.

SARKAR, Sayani; TOTARO, Michael W.; ELGAZZAR, Khalid. Intelligent drone-based surveillance: application to parking lot monitoring and detection. In: **Unmanned Systems Technology XXI**. International Society for Optics and Photonics, 2019. p. 1102104.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. **Understanding machine learning: From theory to algorithms**. Cambridge university press, 2014.

SILVA, Alex Lima; CINTRA, Marcos Evandro. Reconhecimento de padrões faciais: Um estudo. **Encontro Nacional de Inteligência Artificial e Computacional**, p. 224-231, 2015.

SOARES, Matheus André et al. Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de Supermercado. 2021.

SOO, Sander. Object detection using Haar-cascade Classifier. **Institute of Computer Science, University of Tartu**, v. 2, n. 3, p. 1-12, 2014.

VARGAS, Ana Caroline Gomes; PAES, Aline; VASCONCELOS, Cristina Nader. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: **Proceedings of the xxix conference on graphics, patterns and images**. sn, 2016.

YAMASHITA, Rikiya et al. Convolutional neural networks: an overview and application in radiology. **Insights into imaging**, v. 9, n. 4, p. 611-629, 2018.