

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA AEROESPACIAL

**MONITORAMENTO INTELIGENTE DO
PERÍMETRO OPERACIONAL TERRESTRE DO
CENTRO DE LANÇAMENTOS DE ALCÂNTARA
UTILIZANDO PROCESSAMENTO DE IMAGENS E
INTELIGÊNCIA ARTIFICIAL**

Juan Pablo do Nascimento Vale

São Luís - MA
2022

Juan Pablo do Nascimento Vale

Monitoramento inteligente do perímetro operacional terrestre do Centro de Lançamentos de Alcântara utilizando processamento de imagens e inteligência artificial

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Aeroespacial (UFMA) como parte dos requisitos para obtenção ao título de Mestre em Engenharia Aeroespacial

Orientador: Prof. Dr. Alex Oliveira Barradas Filho

Coorientador: Prof. Dr. Geraldo Braz Junior

São Luís - MA

2022

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Vale, Juan Pablo do Nascimento.

Monitoramento inteligente do perímetro operacional terrestre do Centro de Lançamentos de Alcântara utilizando processamento de imagens e inteligência artificial / Juan Pablo do Nascimento Vale. - 2022.

67 f.

Coorientador(a): Geraldo Braz Junior.

Orientador(a): Alex Oliveira Barradas Filho.

Dissertação (Mestrado) - Programa de Pós-graduação em Engenharia Aeroespacial, Universidade Federal do Maranhão, São Luís, 2022.

1. Centro de Lançamento de Alcântara. 2. Reconhecimento facial. 3. Sistema de monitoramento. I. Barradas Filho, Alex Oliveira. II. Braz Junior, Geraldo. III. Título.

Juan Pablo do Nascimento Vale

Monitoramento inteligente do perímetro operacional terrestre do Centro de Lançamentos de Alcântara utilizando processamento de imagens e inteligência artificial

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Aeroespacial (UFMA) como parte dos requisitos para obtenção ao título de Mestre em Engenharia Aeroespacial

Aprovada em

BANCA EXAMINADORA

**Prof. Dr. Alex Oliveira Barradas
Filho**
Universidade Federal do Maranhão

Prof. Dr. Geraldo Braz Junior
Universidade Federal do Maranhão

**Prof. Dr. Allan Kardec Dualibe
Barros Filho**
Universidade Federal do Maranhão

Prof. Dr. Daniel Lima Gomes Júnior
Instituto Federal do Maranhão

São Luís - MA
2022

AGRADECIMENTOS

Agradeço a minha família por todas as opções, suporte e incentivo que me deram em todos esses anos.

Agradeço aos professores pela formação dada no mestrado em Engenharia Aeroespacial

Agradeço aos meus orientadores pela paciência e auxílio em todos os momentos desse trabalho.

Agradeço aos amigos e companheiros do DARTi-Lab pela a ajuda e suporte.

Agradeço aos meus colegas de curso pela ajuda em dificuldades, provas e trabalhos.

Agradeço a minha namorada e aos meus amigos, pelos incentivos e bons momentos.

RESUMO

A principal base de lançamentos espaciais do Brasil é o Centro de Lançamentos de Alcântara, que devido a sua extensão territorial, trabalho de natureza sensível, e requisições de segurança, necessita de uma forma automatizada de controle de perímetro. Atualmente novas formas de monitoramento de perímetro surgem baseadas em tecnologias de inteligência artificial, a chamada *Smart Surveillance*. Neste trabalho, propomos um sistema embarcado de monitoramento inteligente baseado em detecção e reconhecimento facial, para controle de regiões críticas do espaço terrestre do Centro de Lançamento de Alcântara. O hardware embarcado é responsável pela captura e transmissão das imagens, enquanto o processamento é feito em um servidor central. Para a detecção facial são utilizados os algoritmos de Viola-Jones e Multi-Task Cascade Convolutional Neural Network (MTCNN), com F1-Score de 61,28% e 87,29% respectivamente na base de dados DroneFaces. No processo de reconhecimento facial utilizou-se o FaceNet para extração de características, e para classificação foram comparados os modelos kNN, SVM e Random Forest. Em testes de qualidade de reconhecimento facial, atingiu-se ROC-AUC de 0,999 utilizando o classificador SVM na base de dados FaceScrub, e utilizando o classificador Random Forest na base de dados DroneFace atingiu-se F1-Score de 56,6%. Em testes de tempo real, o sistema atinge uma média de 5,8 fps utilizando Viola-Jones, Facenet e Random Forest, e uma média de 2,3 fps utilizando MTCNN, Facenet e Random Forest.

Palavras-chaves: Centro de Lançamento de Alcântara; Sistema de monitoramento; Reconhecimento facial

ABSTRACT

The main space launch base in Brazil is the Alcântara Launch Center, which, due to its territorial extension, sensitive work, and security requirements, requires an automated form of perimeter control. Currently, new forms of perimeter monitoring are emerging based on artificial intelligence technologies, the so-called *Smart Surveillance*. In this work, we propose an embedded intelligent monitoring system based on facial detection and recognition, to control critical regions of terrestrial space at the Alcântara Launch Center. Embedded hardware is responsible for capturing and transmitting images, while processing is done on a central server. For facial detection, the Viola-Jones and Multi-Task Cascade Convolutional Neural Network (MTCNN) algorithms are used, with F1-Score of 61.28% and 87.29% respectively in the DroneFaces database. In the facial recognition process, FaceNet was used to extract features, and the kNN, SVM and Random Forest models were compared for classification. In facial recognition quality tests, an ROC-AUC of 0.999 was achieved using the SVM classifier in the FaceScrub database, and using the Random Forest classifier in the DroneFace database, an F1-Score of 56.6% was reached. In real-time tests, the system averages 5.8 fps using Viola-Jones, Facenet and Random Forest, and an average of 2.3 fps using MTCNN, Facenet and Random Forest.

Keywords: Alcântara Launch Center; Smart security; Face recognition

LISTA DE ILUSTRAÇÕES

Figura 1 – Localização do CLA	14
Figura 2 – Comparação entre os Sistemas Especialistas e os modelos baseados em Aprendizagem de Máquina.	18
Figura 3 – Hierarquia das atividades de Aprendizagem de Máquina	19
Figura 4 – Esquema simplificado de Rede Neural Artificial	20
Figura 5 – Áreas correlatas à visão computacional	21
Figura 6 – Formação de uma imagem através de matrizes.	22
Figura 7 – Exemplo de uma operação de convolução sobre uma imagem	22
Figura 8 – A Aprendizagem Profunda é uma sub-campo da Aprendizagem de Máquina, que por sua vez é um sub-campo da Inteligência Artificial	23
Figura 9 – Esquema básico de uma CNN. A rede recebe uma imagem como entrada, e aplica sucessivas camadas convolucionais sobre esta imagem.	24
Figura 10 – Processo de reconhecimento facial	25
Figura 11 – Exemplos de Haar-features.	27
Figura 12 – Arquitetura MTCNN	27
Figura 13 – Descrição das partes de um Raspberry Pi 3B	32
Figura 14 – Esquema básico de uma comunicação via soquetes	33
Figura 15 – Pipeline de vigilância e controle de acesso. Em vermelho as atividades desenvolvidas pelo sistema embarcado, e as atividades executadas pelo servidor em branco	36
Figura 16 – Algoritmo básico do cliente	37
Figura 17 – Algoritmo básico do servidor	37
Figura 18 – Pipeline de reconhecimento facial.	38
Figura 19 – Alinhamento facial	39
Figura 20 – Exemplo de faces após cada etapa de pré-processamento	40
Figura 21 – Processo de extração de características e de classificação. O processo inicia com uma imagem facial alinhada e padronizada, e conclui com uma distribuição de probabilidades para cada uma das classes.	42
Figura 22 – Reconhecimento facial e autorização de acesso	43
Figura 23 – Fluxo de trabalho do processo de <i>dataset augmentation</i>	44
Figura 24 – (a) Exemplo de imagens após o <i>downsampling</i> em comparação com (b) imagens em diferentes distâncias disponíveis na base de dados DroneFace	44
Figura 25 – Diagrama de classes do sistema	48
Figura 26 – Exemplo de imagens disponibilizadas na base de dados FaceScrub	49
Figura 27 – Exemplos de imagens da FaceScrub, destacando variações de posição, idade e aparência dentro de uma mesma classe.	49

Figura 28 – Exemplo de imagens disponibilizadas na base de dados DroneFace . . .	50
Figura 29 – Taxa de Verdadeiros Positivos para detecções na base de dados DroneFace.	52
Figura 30 – Mapa de calor da acurácia de Reconhecimento Facial na base de dados DroneFaces. 1360 imagens de testes, 72 imagens de treino. Todas as imagens foram alinhadas e padronizadas.	54
Figura 31 – Mapa de calor da acurácia de Reconhecimento Facial na base de dados DroneFaces. 1360 imagens de testes, 360 imagens de treino (base de dados ampliada). Todas as imagens foram alinhadas e padronizadas. .	55
Figura 32 – Histograma de frequência probabilidades	57
Figura 33 – Curva Precisão-Recall de desempenho de verificação de classes conhecidas e desconhecidas	57
Figura 34 – Testes de transmissão. (a) Transmissão sem processamentos. Média de 25 <i>fps</i> . (b) Transmissão com HaarCascade e Facenet. Média de 5,8 <i>fps</i> . (c) Transmissão com MTCNN e Facenet. Média de 2,3 <i>fps</i>	58
Figura 35 – Situações em de falha do pipeline de reconhecimento facial.	59

LISTA DE TABELAS

Tabela 1	– Modelos e respectivos hiperparâmetros a serem otimizados	41
Tabela 2	– Resultado geral dos modelos de detecção facial na base de dados DroneFace.	51
Tabela 3	– Resultados preliminares dos classificadores. Condições de treino: 29817 amostras de treino, 12779 amostras de teste, 530 identidades, padronização dos canais da imagem, alinhamento facial.	52
Tabela 4	– Modelos de classificadores com hiper-parâmetros otimizados. Condições de treino: 29817 amostras de treino, 12779 amostras de teste, 530 identidades, padronização dos canais da imagem, alinhamento facial.	53
Tabela 5	– Lista de hiper-parâmetros selecionadas após a etapa de otimização de classificadores sobre a base de dados FaceScrub.	53
Tabela 6	– Reconhecimento facial na base de dados DroneFace. 1360 imagens de testes, 72 imagens de treino. Todas as imagens foram alinhadas e padronizadas.	54
Tabela 7	– Reconhecimento facial na base de dados DroneFace. 1360 imagens de testes, 360 imagens de treino (base de dados ampliada). Todas as imagens foram alinhadas e padronizadas.	55

LISTA DE ABREVIATURAS E SIGLAS

- AEB:** Agência Espacial Brasileira
- CLA:** Centro de Lançamentos de Alcântara
- CNN:** Convolutional Neural Network
- ERT:** Estação Redundante de Telemetria
- FAB:** Força Aérea Brasileira
- FN:** Falso Negativo
- FP:** Falso Positivo
- FPS:** Frames por Segundos
- kNN:** k-Nearest Neighbor
- LBPH:** Local Binary Patterns Histogram
- MAN-SGO:** Manual de Segurança Operacional do CLA
- MTCNN:** Multitask Cascade Convolutional Neural Network
- NUCLA:** Núcleo do Centro de Lançamento de Alcântara
- PEB:** Programa Espacial Brasileiro
- ROC:** Receiver Operating Characteristic
- SSD:** Single Shot Detector
- SVM:** Support Vector Machine
- VN:** Verdadeiro Negativo
- VP:** Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Centro de Lançamento de Alcântara	14
1.2	Segurança dos Lançamentos Espaciais	15
1.3	Inteligência Artificial	17
1.3.1	<i>Os primórdios da inteligência artificial</i>	17
1.3.2	<i>A aprendizagem de máquina</i>	18
1.3.3	<i>Visão Computacional</i>	20
1.3.4	<i>Aprendizagem Profunda</i>	23
1.4	Novas abordagens de monitoramento	24
1.4.1	<i>O processo de reconhecimento facial</i>	25
1.4.1.1	<i>Detecção Facial</i>	26
1.4.1.2	<i>Preprocessamento de imagens</i>	27
1.4.1.3	<i>Extração de características</i>	29
1.4.1.4	<i>Classificação</i>	30
1.4.2	<i>Sistemas embarcados</i>	31
1.5	Estado da Arte	32
2	OBJETIVOS	35
2.1	Objetivo geral	35
2.2	Objetivos específicos	35
2.3	Contribuições esperadas	35
3	METODOLOGIA	36
3.1	Pipeline de reconhecimento facial e autorização de acesso	37
3.1.1	<i>Detecção Facial</i>	37
3.1.2	<i>Alinhamento facial</i>	38
3.1.3	<i>Padronização de canais</i>	39
3.1.4	<i>Extração de características</i>	40
3.1.5	<i>Classificadores</i>	40
3.1.6	<i>Verificação de acesso</i>	42
3.2	Downsampling: Ampliação artificial dos dados de treino	43
3.3	Critérios de avaliação dos modelos	45
3.4	Materiais e desenvolvimento de software	46
3.5	Datasets	48
4	RESULTADOS E DISCUSSÃO	51

4.1	Avaliação dos modelos de detecção facial	51
4.2	Avaliação do sistema de reconhecimento facial	52
4.2.1	<i>Base de dados FaceScrub</i>	52
4.2.2	<i>Base de dados DroneFace</i>	53
4.2.3	<i>Verificação de acesso</i>	56
4.3	Avaliação de desempenho do sistema	57
5	CONCLUSÃO	60
	REFERÊNCIAS	62

1 INTRODUÇÃO

A indústria aeroespacial é um setor de alta tecnologia, que requer altos investimentos monetários e humanos, empregando uma força de trabalho altamente qualificada e bem remunerada (BARTELS, 2011). Os produtos gerados são utilizados e adaptados em todos os outros setores econômicos, têm altíssimo valor agregado e retornos no médio e longo prazo (BARTELS, 2011). A tecnologia espacial também é marcada pela dualidade civil-militar, o que faz da atividade espacial um ponto estratégico para os poucos países que a dominam, desenvolver o poder aeroespacial significa estar em vantagem política, tecnológica, econômica e militar, incentiva o desenvolvimento científico e industrial da nação, e é um atraente produto de exportação (BARTELS, 2011).

Para inserir o Brasil no contexto espacial, foi fundada a Agência Espacial Brasileira (AEB) e o Programa Espacial Brasileiro (PEB), e construído o Centro de Lançamentos de Alcântara (CLA) (FORÇA AÉREA BRASILEIRA, 2014d). O CLA é uma base de lançamento de foguetes da Força Aérea Brasileira, e é uma das principais estruturas físicas do PEB, sendo um ponto presente no Programa Nacional de Atividades Espaciais 2012-2021 (PNAE 2012-2021) (AEB, 2012). Além disso, em seu Relatório de Gestão 2019, a AEB destinou mais de 19 milhões de reais em Termos de Execução Descentralizada para os centros de lançamento, e após a assinatura do acordo de salvaguarda tecnológica a AEB estuda a ampliação dos investimentos no CLA (AEB, 2020).

Um dos grandes problemas enfrentados pelo CLA é o monitoramento e restrição de acesso ao perímetro de lançamento do CLA durante a fase de operações, uma vez que de acordo com o Manual de Segurança Operacional do CLA (MAN-SGO) é de responsabilidade do Centro garantir a segurança de todos aos riscos associados com as operações de lançamento. O MAN-SGO afirma que o Centro estabelecerá Zonas de Segurança para proteger o pessoal e as instalações, e que deve prover treinamento adequado para todo o pessoal executando operações de risco, que possam impactar a segurança da área de lançamento (BRASIL, 2017). Um agravante para a resolução deste problema é a grande área de monitoramento que deve ser coberta pelo sistema.

Diante do exposto sobre a relevância da atividade aeroespacial, da importância do CLA para o Programa Espacial Brasileiro, e dos processos de segurança necessários no CLA, fica clara a necessidade da manutenção da segurança e o monitoramento do perímetro operacional do CLA para manter a base segura de ações danosas e preparada para a identificação prévia e/ou a prevenção de não-conformidades durante as operações de lançamento. Assim, enunciou-se o seguinte problema: Como monitorar uma porção crítica do centro de lançamento e, de forma automatizada, predizer quem é autorizado ou não a estar nesta área?

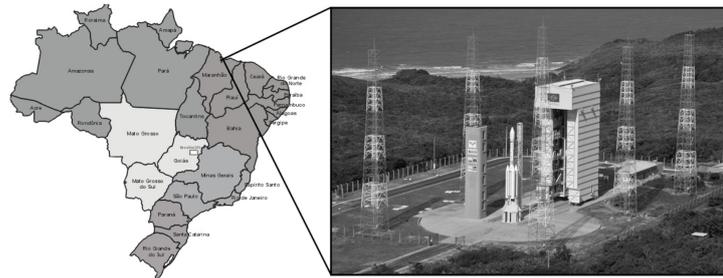
Para efetuar monitoramento de perímetro das porções críticas do espaço terrestre do

Centro de Lançamento Alcântara, propôs-se uma solução baseada em video-monitoramento utilizando técnicas de inteligência artificial para detectar e identificar pessoas. O sistema proposto pode ser aplicado com o uso de câmeras de vigilância em pontos fixos distribuídas ao longo do perímetro, ou câmeras embarcadas em veículos não tripulados capazes de efetuar rondas periódicas em uma determinada área.

1.1 Centro de Lançamento de Alcântara

O Centro de Lançamento de Alcântara (CLA) é a principal base espacial do Brasil, localizada nas intermediações do município de Alcântara - MA, com extensão territorial de $420km^2$ e mais $200km^2$ de área marítima, região considerada de uso militar, de acesso restrito e monitorada pela Força Aérea Brasileira (FAB) (FORÇA AÉREA BRASILEIRA, 2014a). A base foi inaugurada em 1983, com o nome de Núcleo do Centro de Lançamento de Alcântara (NUCLA), com a visão de “Ser reconhecido, em nível nacional e internacional, como um Centro de excelência nas atividades relacionadas com lançamento e rastreo de engenhos aeroespaciais” (FORÇA AÉREA BRASILEIRA, 2014e).

Figura 1 – Localização do CLA



Fonte – Adaptado de Silva e Fisch (2014)

Uma característica importante do CLA é a proximidade com a linha do equador, o que resulta em até 30% de economia de combustível para o lançamento de satélites, se comparado com outras bases de lançamento em latitudes maiores. Ademais, as condições de estabilidade climática, regime de chuvas bem definidos e ventos em limites aceitáveis contribuem para atrair ainda mais a atenção de instituições e empresas internacionais que realizam atividades espaciais (FORÇA AÉREA BRASILEIRA, 2014a).

Além do Centro de Lançamentos de Alcântara, o complexo espacial presente no Maranhão se estende por mais dois sítios: o escritório de São Luís, responsável por questões logísticas e burocráticas relacionadas com o efetivo da base; e a Estação Redundante de Telemetria do Sítio Raposa (ERT), localizada no município de Raposa, responsável por dar apoio técnico-operacional nas atividades do CLA (FORÇA AÉREA BRASILEIRA, 2016).

As operações do CLA envolvem lançamentos de rotina para treinamento constante do efetivo (FORÇA AÉREA BRASILEIRA, 2014b). Atividades de pesquisa e desenvolvimento

de tecnologia espacial, como lançamento do primeiro foguete nacional de combustível líquido VS-30 V13 (FORÇA AÉREA BRASILEIRA, 2014f). E o desenvolvimento do Programa Microgravidade, que realiza lançamentos sub-orbitais com cargas científicas para estudos em microgravidade. Em seu tempo de atividade o CLA já realizou mais de 470 lançamentos nacionais e internacionais, em mais de 90 operações (FORÇA AÉREA BRASILEIRA, 2016).

Cada operação de lançamento do CLA começa pelo transporte dos diversos módulos, estágios, sistemas do veículo e da carga útil a ser transportada. A integração e montagem de todos estes componentes é a fase mais delicada da operação. Após as verificações e testes, a equipe recebe o treinamento e efetua simulações de lançamento, e por fim é iniciado o lançamento real. A etapa final da operação é o pós-voo, em que a equipe efetua a desmontagem, embalagem, transporte e armazenamento dos meios de solo utilizados e resíduos da operação (FORÇA AÉREA BRASILEIRA, 2014c)

1.2 Segurança dos Lançamentos Espaciais

Cada um dos processos e operações desenvolvidas no CLA seguem rígidas exigências que buscam minimizar os riscos para as equipes operacionais, terceiros e as instalações envolvidas. Entre os gerenciamentos de riscos, cita-se as três categorias de segurança: Segurança Pública, Segurança do Complexo e Segurança da Área de Lançamento.

O Manual de Segurança Operacional do CLA (BRASIL, 2017). é o principal documento que trata do esforço do Centro de Lançamento de minimizar os riscos à segurança. Neste documento, o CLA se responsabiliza pela implementação e disseminação das medidas de segurança à todos aqueles cuja segurança possa ser afetada direta ou indiretamente pelas atividades desenvolvidas no Centro. O Manual também descreve as operações de pré-lançamento e de lançamento conduzidas pela base de Alcântara, bem como atividades para delimitação e controle de acesso às áreas restritas ou de risco, e as possíveis zonas de impacto.

O MAN-SGO trata da organização, princípios, processos e política de segurança e documentações relacionadas. Também é estruturada a hierarquia da organização da segurança, delegando ao Coordenador de Segurança do Centro a responsabilidade geral sobre as ações de segurança, ao Coordenador de Segurança de Superfície a atuação sobre todos os sítios do Centro e complexos de lançamento, e ao Coordenador de Segurança de Voo a responsabilidade pela proteção contra voo errante de veículos lançadores.

Uma das atividades previstas pelo MAN-SGO é a determinação de perímetros de segurança em áreas terrestres, aéreas e marítimas, com o objetivo de atender a Política Nacional de Defesa preservando a integridade física tanto da equipe como da comunidade próxima e, também, do patrimônio material brasileiro. Normalmente, a delimitação de tais áreas restringe o acesso para poucos membros da equipe e a presença de algum elemento

não autorizado no perímetro provoca uma não-conformidade, o que pode proporcionar o cancelamento da operação.

Falando especificamente das atividades de controle de perímetro, o Coordenador de Segurança de Superfície é o responsável pela delimitação das áreas de risco, controle de acesso às áreas restritas ou de risco, como definido no item 4.4.2 do MAN-SGO, e também é responsável pelo controle e desocupação das áreas de impacto, como definido no item 8.10 do MAN-SGO.

4.4.2.1. Compete ao Coordenador da Segurança de Superfície:

[...]

r) definir as áreas de risco, assim como prover orientação para o controle de acesso de pessoas a essas áreas dentro dos limites do sítio de lançamento;
[...]

t) providenciar a evacuação de segurança e orientação para o controle de acesso de pessoal às zonas restritas dentro dos limites do Centro de Lançamento;

u) controlar o acesso às estações nas áreas de risco;

(BRASIL, 2017)

8.10. Desocupação da Área de Impacto

As áreas de impacto são controladas por meio da coordenação de segurança de superfície do Centro. Organizações como a Marinha do Brasil e a Força Aérea Brasileira auxiliarão no controle das áreas de impacto, quando necessário.

(BRASIL, 2017)

Vale ressaltar, contudo, que apesar do controle das áreas de impacto ser responsabilidade da coordenação de segurança de superfície, é de responsabilidade do Coordenador de Segurança de Voo a classificação e delimitação de zonas de impactos, e demais áreas de risco relacionadas a montagem, voo, trajetória, e pouso de artefatos. Para isso são determinadas zonas específicas :

8.4. Filosofias e critérios

[...]

a) zona protegida (ZP): é projetada para a proteção de pessoas, instalações e equipamentos. Dentro desta Zona os riscos são reduzidos a um nível aceitável [...]. A população das áreas ao redor da zona de proteção será informada sobre o perigo de ultrapassar suas demarcações.

b) zona não protegida (ZNP): Sobre esta Zona, o veículo lançador pode evoluir livremente, podendo ser inclusive sujeito a impacto. [...] Notificando aos clientes dos espaços aéreo e marítimo nacional e internacional e, eventualmente, pela evacuação de pessoal presente nesta zona.

c) zona não protegida principal (ZNPP): É uma zona de possível impacto do veículo no caso de alguma anormalidade ocorrer na decolagem. [...] Nesta área, o controle é estabelecido por meio de postos de sentinela, que limitam o acesso de pessoal, e é observado usando-se helicópteros e circuito fechado de televisão (CFTV)

(BRASIL, 2017)

Atualmente, para executar as operações de segurança e monitoramento de todas as fases da missão, o CLA utiliza diversos meios e equipamentos determinados no item 9.2 do MAN-SGO. Os recursos contemplam o Circuito Fechado de Televisão (CFTV), e quando necessário dispõem-se de recursos da Marinha e da Força Aérea. Pro caso específico do controle de perímetro e evacuação das áreas de risco é necessária utilização de helicópteros para monitoramento terrestre, e de embarcações para monitoramento da área marítima, atividade crucial uma vez que a vizinhança do CLA é composta por comunidades pesqueiras e quilombolas (IADANZA; ANDRADE, 2021). Entretanto estes são recursos caros, de operação custosa e que exigem mão de obra qualificada.

9.2. Apoio à Segurança de Superfície

[...]

- a) circuito fechado de televisão (CFTV) para operações de segurança de superfície, serão gravadas imagens através câmeras de vídeo que estarão localizadas em cada instalação do Centro de lançamento para o monitoramento das operações;
- b) aeronaves para zonas de desocupação - utilizadas para controlar e para evitar qualquer presença desavisada na área designada de impacto;
- c) embarcações para zonas de desocupação - utilizadas para controlar e para evitar qualquer presença de embarcações na área marítima designada de impacto;

(BRASIL, 2017)

O texto do MAN-SGO cita a norma técnica ABNT NBR 14882:2002 (ABNT, 2002) como uma das referências base do seu texto, contudo, esta norma foi substituída pela ABNT NBR ISO 14620-2:2013 (ABNT, 2013). De acordo com o texto da versão mais recente, a norma trata sobre a exposição de pessoas (ou grupos populacionais) aos riscos relacionados as operações de centros de lançamentos, em atividades diretamente ou indiretamente relacionada com o lançamento e operação de artefatos aeroespaciais (ABNT, 2013). A norma deixa claro que as operações que oferecem riscos devem ter seu perímetro restringido:

5.5.4 Qualquer operação perigosa executada em um local ou centro de lançamento deve ser projetada para ser executada com o mínimo de pessoal presente dentro da área perigosa correspondente. (ABNT, 2013)

As atividades de monitoramento e restrição de perímetro são uma forma de se prevenir aos riscos relacionados às atividades de lançamento, evitando danos materiais, intelectuais e humanos, bem como se antecipar a problemas, evitar ações danosas, e precedência na tomada de decisões.

1.3 Inteligência Artificial

1.3.1 *Os primórdios da inteligência artificial*

Inteligência Artificial (IA) é um campo da ciência da computação que tem o objetivo de desenvolver máquinas com comportamento inteligente, possibilitando a solução de problemas associados as altas capacidades intelectuais dos humanos (ERTEL, 2017).

A história deste campo data dos anos 1940 com os primeiros modelos baseados em simplificações da fisiologia e funcionalidade dos neurônios (MCCULLOCH; PITTS, 1943), em seu início estes trabalhos estavam relacionados a outros campos como teoria do controle e pesquisa operacional, mas em 1956 o termo “Inteligência Artificial” foi cunhado por John McCarthy na conferência de Dartmouth. Os primeiros usos práticos dos sistemas de inteligência artificial chegaram com os Sistemas Especialistas, programas desenvolvidos para abordar um problema específico através de uma série de regras determinadas por um estudioso do problema (RUSSELL; NORVIG, 2013).

Desde os anos 1990 as aplicações de IA vem se expandindo na indústria, muito devido a fatores como o aumento do poder computacional, a formalização da IA como uma ciência e, por consequência, o uso intensificado do método científico nas pesquisas da área, e a disponibilidade de conjuntos de dados muito grandes, o que permitiu a disseminação de abordagens baseadas em Aprendizagem de Máquina (AM), do inglês, *Machine Learning* (RUSSELL; NORVIG, 2013).

1.3.2 A aprendizagem de máquina

A Aprendizagem de Máquina cresceu como alternativa aos Sistemas Especialistas, pois para problemas muito complexos era difícil definir um conjunto de regras capaz de abarcar todas as situações as quais o sistema estaria exposto. Propôs-se então que ao invés de prover diretamente as instruções de como resolver um problema, seria mais simples desenvolver algoritmos que gerassem estas regras automaticamente baseando-se em diversos exemplos conhecidos (KUBAT, 2017), Figura 2. A Aprendizagem de Máquina é a capacidade de um sistema computacional melhorar seu desempenho através da experiência e da exposição à exemplos (FACELI et al., 2011). Devido a essa diferença de abordagem é dito que os modelos de AM são “treinados” e não “programados” (CHOLLET, 2018).

Figura 2 – Comparação entre os Sistemas Especialistas e os modelos baseados em Aprendizagem de Máquina.

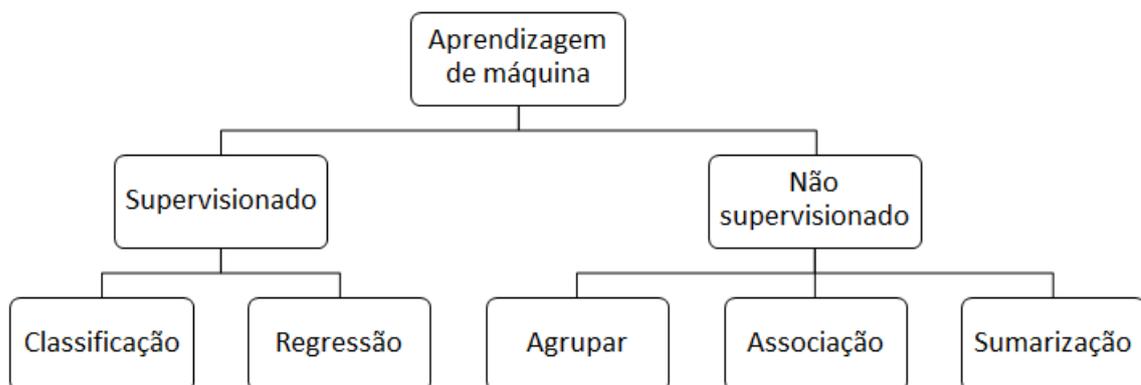


Fonte – Adaptado de (CHOLLET, 2018)

O diferencial dos modelos de Aprendizagem de Máquina é que eles possuem dois momentos distintos de utilização: A primeira fase é o Treino que consiste em fornecer uma quantidade de dados para o modelo, que são utilizados para ajustar os parâmetros internos do modelo; e a segunda fase é a Inferência em que o modelo treinado recebe um novo conjunto de dados como entrada e devolve uma resposta. Se o modelo for adequado, e os dados utilizados no Treino forem capazes de descrever bem o problema, então é esperado que o modelo retorne uma inferência adequada (SKANSI, 2018).

Os modelos de AM podem ser classificados em duas grandes categorias de acordo com o tipo de aprendizagem: Supervisionado e Não Supervisionado, Figura 3. A principal distinção entre as duas categorias é que na forma Supervisionada os dados de treino possuem rótulos que representam a resposta esperada do modelo, já a forma Não Supervisionada se dedica a encontrar padrões desconhecidos nos dados. A diferença nos dados de treino impacta nos tipos de atividades que cada forma desempenha, sendo a Supervisionada primariamente destinada a classificação e regressão, e a Não Supervisionada dedicada a atividades de agrupamento, associação e sumarização (FACELI et al., 2011).

Figura 3 – Hierarquia das atividades de Aprendizagem de Máquina



Fonte – Adaptado de (FACELI et al., 2011)

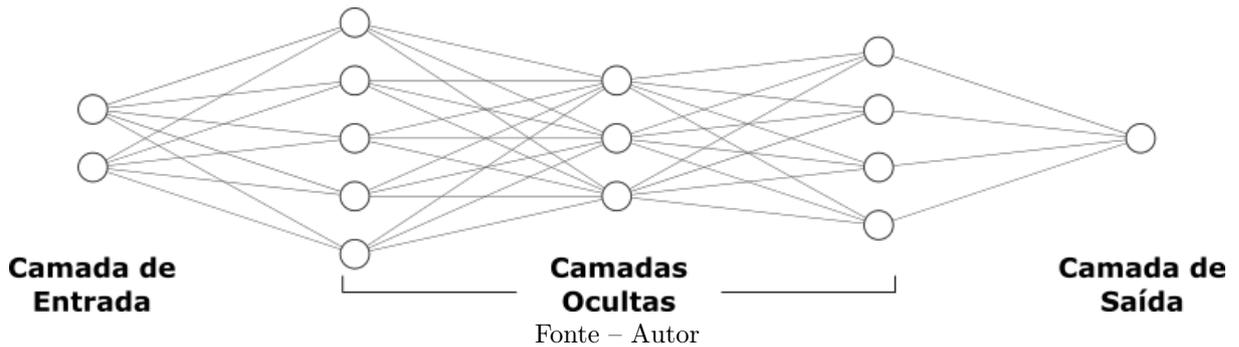
Fica claro que o bom uso de um modelo de aprendizagem de máquina está diretamente ligado à qualidade dos dados a serem analisados. Devido a isso, é necessário cuidado com a seleção e preparação dos dados, uma vez que na maioria das vezes os dados disponíveis não podem ser diretamente aplicados no modelo escolhido e por isso precisam ser pré-processados antes de serem analisados (FACELI et al., 2011). Também é necessário a utilização de técnicas de extração de características, que não apenas selecionam os dados mais importantes dentro do conjunto de dados, mas também aplica operações e transformações sobre os dados para torná-los mais palatáveis aos algoritmos de AM. O desenvolvimento de regras e processos para limpeza de dados e extração de características costumam ser chamado de Engenharia de Características

A atenção à escolha do modelo também é fundamental, para que a complexidade do algoritmo seja adequada à complexidade dos dados. Modelos que sejam muito simples para os dados não conseguem se adequar corretamente e ficam sub-ajustados (do inglês, *underfitting*). Toda via, se o modelo for excessivamente complexo ele será sensível aos ruídos e flutuações aleatórias presentes no conjunto de dados, e tentará ajustar seus parâmetros aos falsos padrões presentes no ruído gerando um sobre-ajuste (do inglês, *overfitting*). Em ambos os casos, a qualidade de inferência do modelo fica comprometida (GÉRON, 2019).

Nos últimos anos, um modelo de aprendizagem de máquina chamado Redes Neurais

Artificiais (do inglês, *Artificial Neural Network - ANN*) vem ganhando espaço. ANNs são um algoritmo de aprendizagem supervisionada, e recebem esse nome pois são baseadas em um modelo simplificado do cérebro humano, onde os neurônios se conectam uns aos outros e enviam informações entre si. De maneira similar, os neurônios artificiais são dispostos em camadas, e se conectam aos neurônios das camadas adjacentes (KUBAT, 2017), Figura 4.

Figura 4 – Esquema simplificado de Rede Neural Artificial



Uma ANN é um modelo composto por unidades simples densamente interconectadas, cada neurônio recebe valores da camada anterior, aplica um peso e um viés sobre a entrada, e retorna um novo valor para a camada seguinte, este peso e viés são os parâmetros a serem treinados. A arquitetura de uma rede neural define quantas camadas terá a rede, bem como a quantidade de neurônios em cada camada e a forma como cada camada se conecta com as outras (FACELI et al., 2011), a grosso modo, a quantidade de camadas e neurônios determina o quão profunda é uma rede.

1.3.3 Visão Computacional

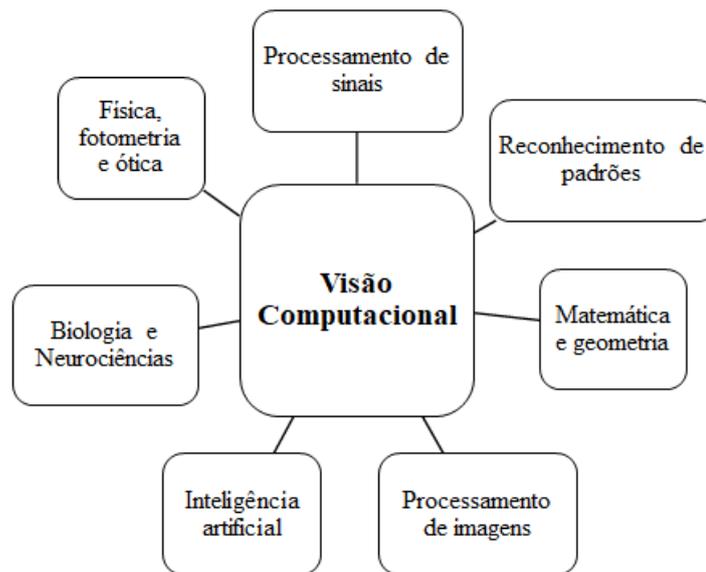
A Visão Computacional (do inglês *Computer Vision - CV*) é a área que tenta ensinar uma máquina a extrair informações úteis a partir de imagens, um objetivo desafiador quando se considera a complexidade dos dados visuais (PRINCE, 2012). A CV quer descrever computacionalmente o mundo tal como visto pelos humanos, e utilizar imagens para reconstruir suas propriedades e extrair informações (SZELISKI, 2011).

A área de estudos da Visão Computacional é muitas vezes confundidas com outros campos correlatos como o Processamento de Imagens, e a Computação Gráfica, porém enquanto estas áreas se preocupam com a modificação ou geração de imagens, a Visão Computacional tem o objetivo de extrair informações do mundo real a partir de imagens (SZELISKI, 2011).

A capacidade de analisar uma imagem diretamente, utilizando apenas uma câmera, torna a CV uma ferramenta poderosa para diversas aplicações como: reconhecimento ótico de caracteres, textos escritos a mão ou impressos; análise de imagens e exames médicos,

auxiliando profissionais no correto diagnóstico; reconstrução de modelos tridimensionais a partir de imagens bidimensionais; aplicações de vigilância, monitoramento de intrusos e análise de tráfego; e aplicações de biometria como reconhecimento facial (SZELISKI, 2011). Para atingir seus objetivos, a visão computacional utiliza ferramentas de diferentes áreas como o processamento de imagem, inteligência artificial, matemática e processamento de sinais, e toma inspirações de outras áreas como a biologia, física, ótica e neurociências (KAKANI et al., 2020).

Figura 5 – Áreas correlatas à visão computacional



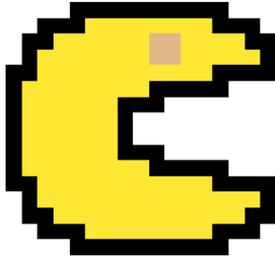
Fonte – Adaptado de (KAKANI et al., 2020)

Do ponto de vista de um computador digital, uma imagem é formada por unidades luminosas conhecidas como pixels. Os pixels são então organizados em matrizes que representam a posição de cada ponto luminoso. Um padrão comum é que cada pixel possa ter uma intensidade que varia entre 0, representando ausência de iluminação, e o 255 que representa iluminação máxima. Utilizando apenas uma única matriz, é possível gerar uma imagem monocromática onde os tons de cinza são formados pelas diferentes intensidades. Porém, é possível gerar imagens coloridas ao representar cada imagem com três matrizes, onde cada matriz representa a intensidade de uma das três cores primárias: vermelho, verde e azul. Este padrão de geração de imagens coloridas é conhecido como RGB, devido ao nome das cores em inglês (*red, green, blue*), e cada matriz de cor é chamada de “canal”(KAPUR, 2017; KAEHLER; BRADSKI, 2017).

A representação de imagens como matrizes implica que, do ponto de vista matemático, as operações de processamento de imagem e visão computacional são transformações lineares e não lineares, que podem ser aplicadas sobre um pixel individualmente, sobre um conjunto de pixels, ou até sobre toda a imagem e seus canais. Tais operações permitem alterações nos valores dos pixels, modificações nas cores, ajustes na luminosidade e

Figura 6 – Formação de uma imagem através de matrizes.

a) Imagem RGB 16x16 pixels



b) Três matrizes 16x16. Cada elemento de uma matriz representa a intensidade de uma cor primária em um pixel

																255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	255	255	255
																255	255	0	0	0	51	51	51	51	51	51	51	51	51	0	0	0	255
255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255															
255	255	0	0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255															
255	0	0	255	255	255	255	255	255	255	222	222	255	255	255	255	0	0	255															
255	0	255	255	255	255	255	255	255	255	222	222	255	255	255	255	255	0	255															
0	0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	0	0	255															
0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	255	255	255	255															
0	255	255	255	255	255	255	255	0	0	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
0	255	255	255	255	255	255	255	0	255	255	255	255	255	255	255	255	255	255															
255	0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255															
255	0	0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255															
255	255	0	0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255															
255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255															

Fonte – Autor

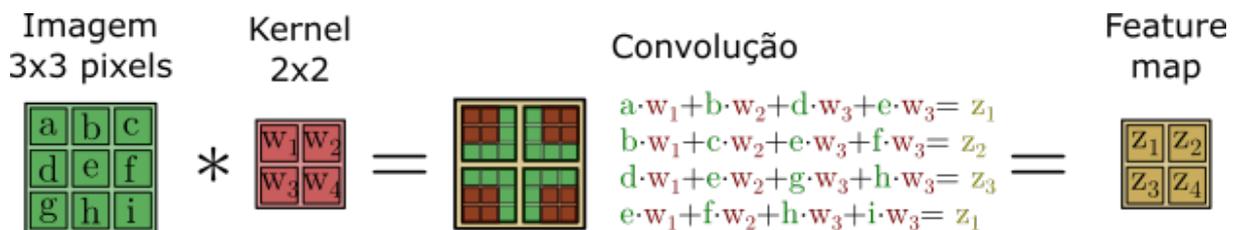
saturação da imagem, redimensionamento, rotação, recorte da imagem e aplicação de filtros diversos. Quando trata-se especificamente da filtragem de imagens, é possível analisar os pixels no domínio espacial, atuando diretamente sobre os pixels da imagem, ou no domínio da frequência, aplicando-se transformada de Fourier sobre a matriz de imagem para convertê-la em um somatório de senos e cossenos (BACKES; SÁ JÚNIOR, 2016)

Uma operação que vem ganhando atenção é a convolução, que é a base de boa parte dos processos de filtragem de imagem (BACKES; SÁ JÚNIOR, 2016) e tem se tornado importante em modelos de redes neurais (LECUN et al., 1990; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Uma convolução é uma operação linear aplicada sobre a região em que duas funções ou matrizes se sobrepõem, de um ponto de vista geométrico, uma matriz de entrada I é sobreposta por uma matriz K que desliza sobre a matriz de entrada efetuando a soma dos produtos elemento-a-elemento na região sobreposta, resultado em uma matriz final S . A expressão algébrica para uma convolução S em um domínio discreto é dado pela Equação 1.1 (GOODFELLOW; BENGIO; COURVILLE, 2016):

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n) \tag{1.1}$$

Para aplicações de filtragem e tratamento de imagens a matriz I representa a imagem, a matriz K representa o kernel da convolução.

Figura 7 – Exemplo de uma operação de convolução sobre uma imagem

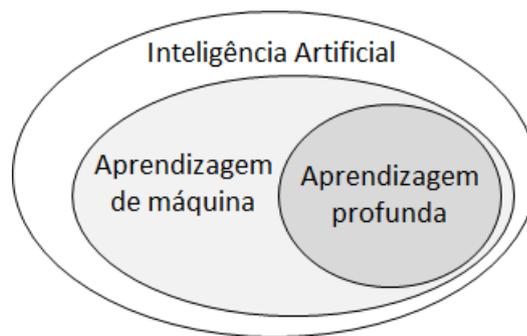


Fonte – Autor

1.3.4 Aprendizagem Profunda

De maneira simples, a Aprendizagem Profunda (do inglês, *Deep Learning*) pode ser dita como um ramo da Aprendizagem de Máquina que lida com redes neurais muito profundas, possibilitando a aplicação de redes neurais em atividades mais complexas, porém, em contrapartida trazendo maiores dificuldades relacionadas ao custo computacional, a dimensionalidade dos modelos e a necessidade de grandes quantidades de dados para treino (ERTEL, 2017). As técnicas de Aprendizagem Profunda têm obtido bons resultados em aplicações diversas como: classificação de imagens, processamento de linguagem natural, algoritmos de busca e geração procedural de conteúdo, e outros, que têm sido aplicados no desenvolvimento de sistemas de recomendação, assistentes virtuais, carros autônomos, e inteligência artificial para jogos. (CHOLLET, 2018).

Figura 8 – A Aprendizagem Profunda é uma sub-campo da Aprendizagem de Máquina, que por sua vez é um sub-campo da Inteligência Artificial



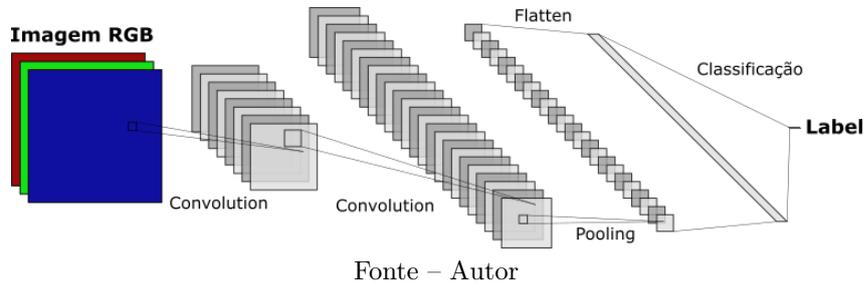
Fonte – Adaptado de (CHOLLET, 2018)

Redes Neurais Profundas (do inglês, *Deep Neural Network - DNN*) são modelos com dezenas de camadas, onde cada camada pode ter centenas ou até milhares de neurônios, e milhões de parâmetros a serem treinados. O tamanho desses modelos torna necessário o uso de quantidades massivas de dados, que devido ao tamanho e quantidade podem ser difíceis de rotular e processar; a grande quantidade de parâmetros traz um considerável risco de *overfitting*; e o custo computacional associado torna os processos de treinamento e inferência muito mais lentos (GÉRON, 2019).

Apesar das dificuldades, a utilização de *deep learning* tem tornado o processo de resolução de problemas mais simples, uma vez que automatiza e aprimora quase completamente o processo de extração de características, facilitando o processo de engenharia de características, isso é chamado de aprendizagem de características (do inglês, *feature learning*). Isto possibilita simplificar o fluxo de trabalho de um sistema, substituindo uma sequência complexa de operações de extração de características, por uma única rede neural profunda (CHOLLET, 2018).

Uma das responsáveis pelo sucesso da Aprendizagem Profunda são as operações de convolução, que possibilitaram o surgimento de uma nova ferramenta: as Redes Neurais

Figura 9 – Esquema básico de uma CNN. A rede recebe uma imagem como entrada, e aplica sucessivas camadas convolucionais sobre esta imagem.



Convolucionais (do inglês, *Convolutional Neural Network - CNN*). Um dos primeiros usos bem sucedidos das CNNs foi na classificação de dígitos escritos à mão (LECUN et al., 1990), e se tornaram populares após serem utilizadas para vencer a competição ImageNet em 2012 (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Em uma rede convolucional, os parâmetros a serem treinados são os kernels de cada convolução, que podem ser tratados como os “neurônios” desta camada. Devido a forma que o processo de convolução atua deslizando sobre uma matriz, é possível extrair informações relacionadas à organização espacial dos valores da matriz, o que as tornam especialmente úteis em aplicações de análise de imagens. Devido a isso, as CNNs são largamente usadas em visão computacional, sendo empregadas para detecção de objetos, classificação de imagens, reconhecimento óptico de caracteres, detecção e extração de plano de fundo, descrição de cena, e muitas outras aplicações (ALOYSIUS; GEETHA, 2017)

1.4 Novas abordagens de monitoramento

As novas abordagens de monitoramento de perímetro se utilizam largamente de tecnologias como sensores, inteligência artificial e redes de computadores. A chamada Segurança Inteligente (*Smart Surveillance*) permite a interconexão e descentralização dos sistemas de vigilância, integração de informações, e o uso de equipamentos e tecnologias inteligentes que possibilitam a automatização de processos e tomadas de decisão (HAMPAPUR et al., 2003). Um bom exemplo são os modelos de autenticação biométrica, que podem ser baseados em rosto, impressões palmares, impressões digitais, íris e voz, que são usados para aumentar a segurança, personalizando e facilitando o acesso a áreas restritas (ARSLAN et al., 2016).

No caso específico do reconhecimento facial, técnicas de Machine Learning podem ser aplicadas para detectar e identificar faces, possibilitando uma forma não intrusiva de autenticação, de baixo custo, e capaz de operar com descrição (ORTIZ et al., 2018). Contudo, aplicar detecção e reconhecimento facial em atividades de monitoramento e segurança é uma tarefa desafiadora devido as variações de iluminação, plano de fundo e expressões faciais (LEI et al., 2009), que juntamente com a baixa resolução das imagens de

monitoramento tornam o reconhecimento facial para aplicações de vigilância um desafio a ser estudado (CHENG; ZHU; GONG, 2018).

Se combinarmos os avanços da inteligência artificial aos modernos sistemas embarcados, é possível desenvolver equipamentos de vigilância autônoma. Uma ideia sendo explorada é o uso de drones de vídeo monitoramento (MISHRA et al., 2020; SAHA; KUMAR; SAHU, 2018; BOLD; SOSORBARAM; LEE, 2016), podendo ser usados para vigiar grandes áreas, seguir alvos, e buscar pontos de interesse.

1.4.1 O processo de reconhecimento facial

A problemática do reconhecimento facial pode ser enunciada da seguinte maneira: *Possuindo um banco de dados contendo faces e suas identidades, como é possível determinar a identidade de uma nova face?* (JAFRI; ARABNIA, 2009). Os primeiros modelos de reconhecimento facial datam dos anos 1960 Bledsoe (1964), mas somente a partir de 1991 os primeiros métodos conseguiram atingir resultados praticáveis em ambientes pouco controlados (TURK; PENTLAND, 1991), e atualmente, já existem modelos com acurácia comparável à capacidade humana (WANG; DENG, 2018).

Diversas técnicas de visão computacional e aprendizagem de máquina foram aplicadas no desenvolvimento de novos algoritmos de reconhecimento facial, mas de maneira geral, o processo de reconhecimento facial pode ser resumido em quatro etapas: Detecção Facial, Pré-processamento, Extração de características e Classificação, Figura 10.

Figura 10 – Processo de reconhecimento facial



Fonte – Autor

A primeira etapa é a Detecção de Facial, que determina a existência e localização de faces em uma imagens, a segunda é o Pre-processamento de Imagens que envolve diversos tratamentos e transformações para realçar detalhes específicos da imagem, a terceira etapa é a extração de características, que transforma uma imagem em uma representação vetorial, e por ultimo têm-se a etapa de classificação, que utiliza aprendizagem de máquina para determinar a identidade da face(KUMAR; KAUR; KUMAR, 2019; DHARAVATH; TALUKDAR; LASKAR, 2014; KUMAR; BHATIA, 2014; JAFRI; ARABNIA, 2009).

1.4.1.1 Detecção Facial

O problema de detecção facial pode ser enunciado da seguinte forma: *Dada uma imagem, detecte e localize um número desconhecido, se houverem, de faces.* Aqui vasculha-se a imagem em busca de rostos, para determinar sua localização, e por vezes determinar pontos chaves úteis como a posição dos olhos, nariz e boca. O desafio de detectar faces é essencialmente um problema de classificação e segmentação, além de envolver a detecção de características faciais em ambientes não controlados (HJELMÅS; LOW, 2001).

As dificuldades do processo de detecção facial podem ser associadas a variações de escala, localização, orientação, posicionamento, expressão facial, oclusões. Estão principalmente relacionadas a complexidade do plano de fundo, faces semiocultas, distância, condições de luminosidade, resolução da imagem, tom de pele, e o excesso ou ausência de faces na imagem (KUMAR; KAUR; KUMAR, 2019).

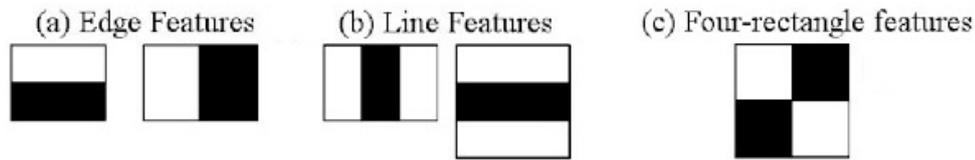
Este processo é o primeiro passo de diversas atividades de análise facial, como: alinhamento, modelagem, reconhecimento, leitura de expressões, e muito outros (KUMAR; KAUR; KUMAR, 2019). Os primeiros algoritmos de detecção facial datam dos anos 1960, porém a maioria destes métodos não proviam bons resultados em aplicações práticas, fora de ambientes controlados, devido as diversas dificuldades do processo (ZAFEIRIOU; ZHANG; ZHANG, 2015).

Apesar dessas dificuldades, desde o início dos anos 2000 uma gama de algoritmos tem sido capazes de solucionar essa problemática no mundo real, em ambiente não controlado. O trabalho seminal de Viola e Jones (2001) deu o pontapé inicial em toda uma sorte de algoritmos baseados em *boosting* e descritores de imagem, entre eles: Haar-features, LBP, HoG e SIFT (ZAFEIRIOU; ZHANG; ZHANG, 2015). Estes descritores são costumeiramente denominados como *handcraft features* (“características feitos à mão”, em tradução livre) pois utilizam regras determinadas por um programador para extrair as características de uma determinada imagem, e o resultado é usado com classificadores simples ou técnicas de aprendizagem rasa.

Para analisar toda uma imagem, se utiliza o conceito de “janela deslizante”, onde uma região de tamanho definido é deslizada sobre a imagens, e cada conjunto de pixels é classificado como “face” ou “não-face”. Para considerar a possibilidade da janela ser pequena ou grande demais para as faces existentes, a imagem pode ser redimensionada múltiplas vezes (MUKHERJEE et al., 2017)

O mais bem sucedido método desta categoria é o Algoritmo de Viola-Jones (VIOLA; JONES, 2001), também conhecido como Haar Cascade. O método utiliza uma sucessão de descritores chamados Haar-features que são deslizados sobre a imagem para identificar mudanças de luminosidade, cantos e bordas, que são então unificados pelo método Adaboost (JONES; VIOLA, 2003). É um algoritmo robusto e veloz, comumente utilizado para detecção facial em tempo real em sistemas de baixo poder computacional (DANG; SHARMA, 2017).

Figura 11 – Exemplos de Haar-features.



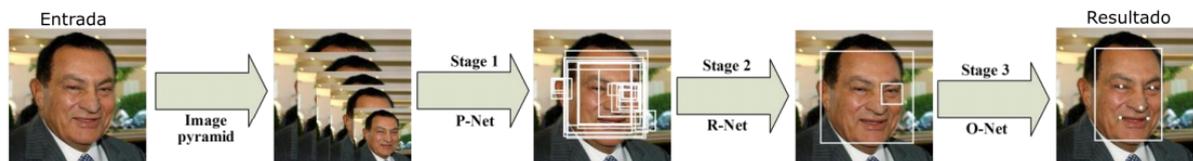
Fonte – Adaptado de (BRADSKI, 2000).

Um novo salto na qualidade da detecção facial veio com os métodos baseados em redes neurais convolucionais (MINAEE et al., 2021), formando um novo grupo de técnicas baseada no uso de redes neurais profundas (MUKHERJEE et al., 2017). O sucesso das redes convolucionais para detecção facial originou diversos métodos derivados, com quatro tipos principais de arquiteturas: Cascatas de CNN; CNN baseada em região (R-CNN); modelos *Single Shot Detector* (SSD); e modelos baseado em Rede Piramidal de Características (do inglês, *Feature Pyramid Network - FPN*) (MINAEE et al., 2021)

Ao contrário das abordagens “feitas à mão”, as CNNs desenvolvem descritores automaticamente partindo das imagens de treino, sem fazer qualquer pré-suposição sobre padrões ou características da imagem a ser analisada (ZAFEIRIOU; ZHANG; ZHANG, 2015). É fato que o número de camadas é proporcional à qualidade das previsões da rede, mas em compensação, há um grande aumento no custo computacional (MUKHERJEE et al., 2017).

O algoritmo *Multitask Cascaded Convolutional Neural Networks* (MTCNN), é um modelo baseado em cascatas de CNN e que utiliza uma estrutura sequencial de três redes neurais convolucionais capazes de prever a localização de uma face e seus pontos-chaves em uma imagem (ZHANG et al., 2016). O modelo MTCNN atinge altas taxas de precisão e recall, entretanto, pode ser computacionalmente custoso.

Figura 12 – Arquitetura MTCNN



Fonte – Adaptado de (ZHANG et al., 2016)

1.4.1.2 Preprocessamento de imagens

Para a maioria das aplicações de visão computacional, um dos primeiros passos aplicados é o Preprocessamento de Imagens, para converter as imagens em uma forma mais palatável para os processos seguintes. Na prática, o preprocessamento de imagens é uma coleção de técnicas que visam realçar ou remover certas características da imagem. Entre

as técnicas mais populares estão redimensionamento, rotacionar, correção de luminosidade, balanceamento de cores, redução de ruído, etc. (SZELISKI, 2011).

Estas técnicas têm alto impacto na qualidade final dos modelos, uma vez que aplicar imagens brutas aos modelos de classificação não traz bons resultados (PAL; SUDEEP, 2016). Além disso, efetuar pré-processamentos antes da extração de características ajuda a solucionar dificuldades relacionadas à qualidade das imagens (DHARAVATH; TALUKDAR; LASKAR, 2014). A escolha das técnicas de pré-processamento variam de acordo com os dados disponíveis e com os métodos de extração de características e classificação selecionados.

O alinhamento facial é um dos grandes desafios do pré-processamento de faces. Esse desafio consiste em duas etapas, primeiro devem ser encontrados os pontos-chaves de um rosto e em seguida aplicam-se transformações geométricas sobre a imagem tendo os pontos-chave como guias. Em ambiente controlado, o alinhamento facial já é um problema resolvido, porém em condições reais esta é uma tarefa longe de ser solucionada por conta dos múltiplos fatores que afetam as imagens faciais, principalmente devido às variações de posição, mudanças de expressão facial, possíveis oclusões e alterações nas condições de iluminação (JIN; TAN, 2017).

Uma importante etapa de pré-processamento é o redimensionamento de imagem. A maioria dos modelos de extração de características é preparado para lidar com apenas um único tamanho de imagem. Embora as câmeras possam ter uma resolução fixa, as faces, por outro lado, podem possuir diferentes tamanhos e razões de aspecto. O processo de redimensionamento é a forma de uniformizar o tamanho das imagens que entram no extrator de características. Uma vez que imagens são matrizes de dados, o processo de redimensionar deve remover dados (quando reduzindo) ou adicionar dados (quando ampliando), através do uso de métodos de interpolação, como por exemplo, as interpolações linear, bilinear, ou por relação de área (BRADSKI, 2000).

Outro método de pré-processamento é a normalização, uma técnica que possibilita encontrar uma nova faixa de valores que seja mais agradável ao extrator de características ou ao classificador (PATRO; SAHU, 2015). De maneira geral, as matrizes de imagem são representadas por números inteiros no intervalo $[0, 255]$, porém isso pode ser modificado para um intervalo desejado. As técnicas de normalização mais comuns são: Normalização Min-Max que trunca os pixels entre um valor mínimo e máximo predeterminado; Z-Score que busca fazer a distribuição de pixel ter desvio padrão próximo de 1 e média próxima de zero; e Escalonamento Decimal que dispõe os pixels numa faixa fixa entre -1 e 1 .

Os métodos de pré-processamento como redimensionamento, normalização, filtragem e tratamento de imagem são essenciais em um pipeline de classificação de imagens, pois tornam a imagem acessível aos processos de extração de características (DHARAVATH; TALUKDAR; LASKAR, 2014).

1.4.1.3 Extração de características

Um processo de extração de características consiste em transcrever as principais características de uma imagem em uma forma matemática que possa ser utilizada em processos de aprendizagem de máquina. O principal objetivo da extração de características é obter e representar informações relevantes em um espaço dimensional reduzido (KUMAR; BHATIA, 2014). Extrair características visuais eficientes é um passo crucial em diversas atividades de processamento de imagem (TIAN et al., 2013), e é uma etapa vital do reconhecimento facial, uma vez que o uso de imagens “cruas” reduz consideravelmente a performance do sistema (SRISUK et al., 2018).

Diversos métodos já foram usados como extratores de características para criar representações robustas de imagens faciais. Os métodos de extração de características clássicos são desenvolvidos por *handcraft features*, cujo os processos de análise de imagem são determinados pelo programador e por isso podem ser treinados em base de dados pequenas (ou até nenhuma base de dados) (SRISUK et al., 2018). Os principais métodos dessa categoria são baseados em características visuais comuns como formato, cor e textura (TIAN et al., 2013). Alguns exemplos bem sucedidos de métodos clássicos de extração de características são o *Local Binary Patterns*, um descritor de textura que considera a relação entre um pixel e seus vizinhos (AHONEN; HADID; PIETIKÄINEN, 2004), e o *Eigenfaces* um método baseado em análise dos componentes principais para reduzir a dimensionalidade de uma imagem (TURK; PENTLAND, 1991)

Uma segunda categoria de métodos de extração de características são os baseados em aprendizagem profunda e redes neurais convolucionais em processos de *feature learning*. Nestes modelos o programador define a topologia de uma rede neural convolucional, e submete esta rede a um treinamento em uma extensa base de dados (SRISUK et al., 2018). O modelo DeepFace desenvolvido pelo Facebook foi um dos primeiros a estreitar a comparação entre a capacidade humana e a capacidade computacional de reconhecimento facial (TAIGMAN et al., 2014). Outro modelo foi o VGG Face Descriptor que atingiu alta acurácia mesmo sem utilizar maiores pré-processamentos sobre as imagens (PARKHI; VEDALDI; ZISSERMAN, 2015). Uma série de redes chamadas de DeepID foram desenvolvidas, com o diferencial de que a rede é supervisionada não apenas na última camada, mas em vários outros passos intermediários também (SUN; WANG; TANG, 2015). Um outro modelo baseado em CNN é o FaceNet que foi desenvolvido pela Google utilizando um tipo especial de função de custo chamado *triplet loss* que possibilitou uma maior distinção entre as classes (SCHROFF; KALENICHENKO; PHILBIN, 2015).

O FaceNet é um modelo baseado em *deep learning* capaz de atingir níveis de reconhecimento facial comparáveis à capacidade humana. O processo de treino do FaceNet é focado em aprender como mapear uma imagem de face em um vetor euclidiano compacto, de forma que a distância entre dois vetores é diretamente proporcional à similaridade das faces (SCHROFF; KALENICHENKO; PHILBIN, 2015), dessa maneira, o vetor compacto

pode ser utilizado em uma grande variedade de algoritmos de classificação. O vetor compacto retornado pelo FaceNet podem ser aproveitado para atividades de classificação ou agrupamento. Utilizando o FaceNet como um extrator de características juntamente com um algoritmo de classificação é possível efetuar reconhecimento facial.

1.4.1.4 Classificação

Um classificador é uma função que recebe uma amostra desconhecida e atribui a ela uma classe mais provável (FACELI et al., 2011). No caso do reconhecimento facial, a amostra desconhecida recebida é um vetor de características, e a classe a ser determinada é a identidade a ser descoberta. Diversas técnicas de classificação estão disponíveis para uso, e algumas das mais utilizadas são: *k-Nearest Neighbor* (kNN), *Support Vector Machines* (SVM), *Decision Trees* e *Random Forest*

O *k-Nearest Neighbor* é um método de aprendizagem baseado em distância. Durante o seu processo de treino, o kNN simplesmente armazena todas as amostras de treino, e ao inserir uma amostra desconhecida para inferência, o algoritmo observa a distância entre a amostra desconhecida e todas as amostras de treino, o resultado da classificação é determinado pela classe dos k mais próximos (FIX; HODGES, 1989).

A versão mais comum do algoritmo se vale simplesmente da contagem dos vizinhos próximos e considera a classe mais comum como a resposta final, e utiliza a distância euclidiana para o cálculo. Porém, versões mais elaboradas do método podem considerar esquemas diferenciados de votação, utilizar outros tipos de distância que favoreçam algum atributo de interesse, considerar o a distância como um fator de ponderação aumentando ou diminuindo o poder de voto de acordo com a distância, e outras variações (KUBAT, 2017).

Em suma, o kNN é um algoritmo simples, rapidamente treinável, e a despeito de sua simplicidade, pode ter boa performance e ser usado para comparar outros métodos mais complexos (FACELI et al., 2011). Este foi o primeiro modelo de classificação a ser utilizado em conjunto do extrator de características FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015). Em contrapartida, o processo de treino não cria, de fato, nenhum modelo de previsão, e para cada nova inferência é necessário calcular a distância entre o novo ponto e todos os membros de treino, o que pode ser computacionalmente custoso (FACELI et al., 2011)

O *Support Vector Machines* (SVM) é um algoritmo baseado em aprendizagem estatístico, que tenta traçar um fronteira de decisão entre as classes de um problema. Em uma distribuição de dados, o algoritmo cria um hiperplano que separa uma classe abaixo deste hiperplano, e a outra classe acima do hiperplano. Quando uma amostra desconhecida é inserida, sua posição em relação ao hiperplano define a classe a ser prevista (HEARST et al., 1998).

A formulação mais simples deste método utiliza um kernel linear e lida com problemas linearmente separáveis, porém foram desenvolvidas variações para abarcar dados mais complexos. Estas abordagens se baseiam na ideia aplicar transformações no espaço de características de forma a encontrar um novo espaço em que os dados sejam linearmente separáveis. Os kernels mais comuns são o Polinomial, o Sigmoidal e o de Função de Base Radial (RBF) (FACELI et al., 2011).

A ideia original das SVMs é adequada a problemas de classificação binária, mas uma estratégia comum para abarcar problemas de múltiplas classes é a Um Contra Todos em que é gerado um classificador binário para cada classe, e durante a inferência consideramos o resultado do melhor classificador. Outra opção é a abordagem Um Contra Um em que é treinado um classificador binário para cada par de classes, e consideramos o resultado da classe que venceu o maior número de classificadores (GÉRON, 2019).

O modelo SVM é reconhecidamente versátil, robusto, e apresenta bons resultados tanto em dados lineares quanto em não-lineares, e com o kernel adequado é capaz de se ajustar bem a conjuntos de dados complexos (GÉRON, 2019). Como desvantagem as SVMs podem perder desempenho em classificações multiclasse, e além disso podem ser muito sensíveis aos valores dos parâmetros (FACELI et al., 2011).

Uma *Decision Tree* (DT) é um modelo de aprendizagem de máquina baseada em procura, uma vez que o modelo é formulado para buscar a resposta em um espaço de possíveis soluções. A estratégia adotada é dividir o problema complexo em uma sucessão de problemas simples (FACELI et al., 2011). Dessa forma, o modelo de Decision Tree (SAFAVIAN; LANDGREBE, 1991) é um algoritmo capaz de se moldar a uma grande variedade de dados complexos. Devido sua estrutura, este modelo requisita mais tempo de treino do que os anteriores, porém possui capacidade de fazer inferências rápidas (GÉRON, 2019).

Por fim, o Random Forest (BREIMAN, 2001), é um *ensemble* de diversas Decision Trees, o que resulta em uma considerável melhoria na qualidade das previsões do modelo, em contrapartida, o tempo de treinamento e o tempo de previsão podem sofrer grande aumento quando comparado ao modelo de Decision Tree (GÉRON, 2019).

1.4.2 *Sistemas embarcados*

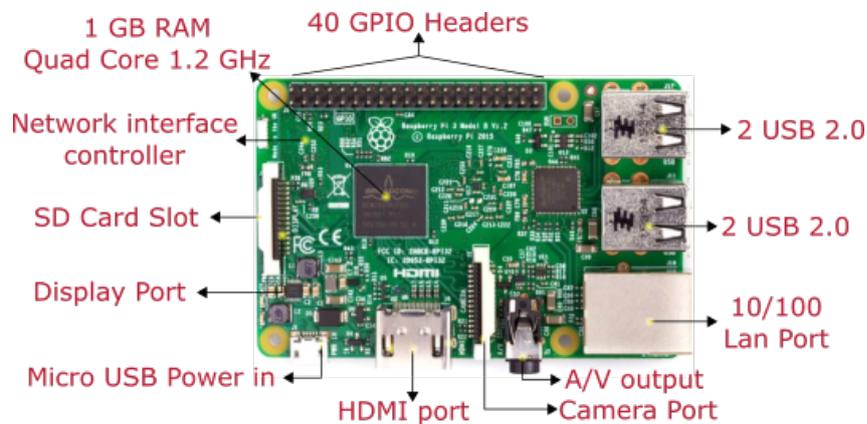
Os processos de vigilância inteligente atuais utilizam fortemente a tecnologia dos sistemas embarcados como forma de estender sua presença física. Estes sistemas são complexos computacionais encapsulados, de funcionalidade específica, que atendem às restrições de custo, robustez e portabilidade do projeto (OLIVEIRA; ANDRADE, 2006).

Atualmente, uma das principais opções de sistemas embarcados do mercado é o Raspberry Pi. Um Sistema em um Chip (do inglês, *System-on-a-Chip* - *SoC*), originalmente desenvolvido como um computador de custo acessível para ser usado em escolas de baixa renda. Mas a versatilidade do equipamento rapidamente o tornou uma plataforma popular

para prototipagem e desenvolvimento de produtos na indústria e na academia (UPTON; HALFACREE, 2014).

Uma das grandes vantagens do Raspberry Pi é a interconexão do sistema operacional Linux com pinos GPIO, possibilitando controlar diferentes tipos de sensores, câmeras, motores, e outros componentes eletrônicos através de *scripts* e linguagens de alto nível (POLIANYTSIA; STARKOVA; HERASYMENKO, 2016), além de se mostrar com uma plataforma eficiente e barata capaz de adquirir, processar e transmitir vídeos em tempo real (COCORULLO et al., 2015).

Figura 13 – Descrição das partes de um Raspberry Pi 3B



Fonte – Adaptado de (Raspberry Pi Trading LTD, 2020)

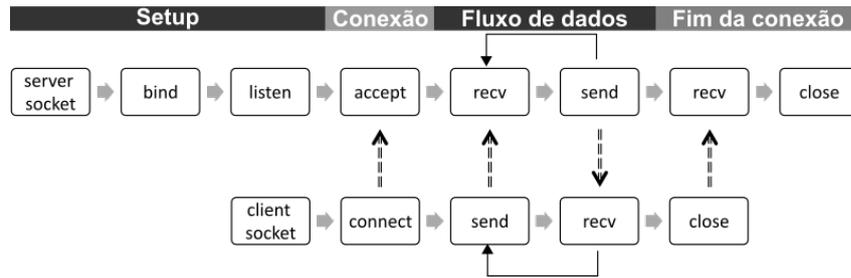
Devido suas qualidades o Raspberry Pi aparece como solução de hardware para diversos autores que desenvolveram sistemas embarcados ou autônomos usados em atividades de vigilância e segurança (AMIN; AHMAD; ALI, 2016; NGUYEN et al., 2015; SAHA; KUMAR; SAHU, 2018; COCORULLO et al., 2015; GUPTA et al., 2016)

Para o pleno funcionamento do sistema de segurança inteligente, é indispensável que o módulo de captura de imagens esteja em constante comunicação com o servidor central. O Raspberry Pi consta com conexão Wi-Fi, permitindo conexão por soquetes e protocolo TCP/IP. Soquetes são estruturas que atuam como fim de curso para envios e recebimentos de mensagens ao longo de uma rede, e atualmente são utilizados em quase todos os sistemas compatíveis com POSIX, incluindo Windows, MacOS e Linux (RHODES et al., 2014). Um esquema simplificada de comunicação por soquetes pode ser visto na Figura 14.

1.5 Estado da Arte

Diversos artigos sobre sistemas de vigilância baseados em IoT e *Deep Learning* têm sido publicados, por exemplo, Kaliappan et al. (2019) desenvolveram um sistema de autenticação biométrica para acesso a um campus universitário. Xu (2020) compara e discute diferentes técnicas de detecção e reconhecimento facial, propõem um sistema de segurança inteligente baseado em video vigilância e concluem que a combinação de FaceNet e MTCNN atinge

Figura 14 – Esquema básico de uma comunicação via soquetes



Fonte – Autor

as melhores taxas de precisão. Nguyen et al. (2015) descrevem um sistema de vigilância baseado em Raspberry Pi, que reduz a necessidade de armazenamento de dados pois a captura de imagens é vinculada a detecção de movimentos e a transmissão das imagens é feita através da internet. O trabalho de Cocorullo et al. (2015) apresentou um sistema embarcado baseado em Raspberry Pi que utiliza técnicas de subtração de background para detecção de movimento. Zhai e Cheng (2011) desenvolveram um sistema de *smart home* capaz de supervisionar o funcionamento de eletrodomésticos, e monitorar em tempo real o status da segurança residencial.

Alguns trabalhos se focaram na resolução das dificuldades técnicas do processo. O artigo de Wang et al. (2017) concluiu que uma forma de suplantar as dificuldades geradas pelas variações de posição, iluminação e expressão facial, é treinar uma rede neural convolucional em uma base de dados que contemple vídeos de vigilância reais. Há também discussões sobre a descentralização do esforço computacional, Amin, Ahmad e Ali (2016) propõem um sistema de vigilância com reconhecimento facial descentralizado, em que o computador embarcado realiza a etapa de detecção e extração facial, e o servidor realiza a etapa de reconhecimento, minimizando a largura de banda necessária.

Outra ideia sendo explorada é o uso de drones em tarefas de monitoramento por vídeo, já tendo sido propostos sistemas autônomos para auxiliar atividades de busca e resgate, utilizando técnicas de detecção de pessoas e detecção de ações (MISHRA et al., 2020). Há também sistemas capazes de rastrear objetos em movimento e reconhecer faces utilizando Haar Cascade e Eigenfaces (BOLD; SOSORBARAM; LEE, 2016), ou com o algoritmo Local Binary Patterns (SAHA; KUMAR; SAHU, 2018).

Aplicar técnicas de reconhecimento facial nas condições de trabalho de um drone não é uma tarefa trivial e é tópico de pesquisas recentes (DEEB; ROY; EDOH, 2020), e estimula a publicação de base de dados especializadas nesta problemática (HSU; CHEN, 2017; KALRA et al., 2019). Tais pesquisas concluíram que imagens capturadas em ângulos muito fechados ou em grandes distâncias podem dificultar a detecção e reconhecimento facial (HSU; CHEN, 2015), mas que a inserção de imagens de treino que representem essas condições podem beneficiar o modelo final (DEEB; ROY; EDOH, 2020).

A comunidade científica disponibiliza diferentes datasets de imagens capturadas por

drones ou condições similares. Hsu e Chen (2017) disponibilizaram o DroneFace, um dataset especializado no teste de desempenho de algoritmos de reconhecimento facial em drones, com capturas de imagens em diferentes distâncias e ângulos. Kalra et al. (2019) lançaram o DroneSURF, dataset focado em vigilância e detecção de faces, contendo 200 vídeos em condições adversas de captura, e com faces marcadas de 58 identidades.

Hsu e Chen (2015) estudaram as condições de trabalho de um drone de vigilância, concluindo que a captura de imagens em ângulos muito fechados, ou em grandes distâncias podem dificultar a detecção facial. Deeb, Roy e Edoh (2020) comparam o desempenho de diferentes arquiteturas de CNNs em imagens capturadas por drones em condições desfavoráveis, e seus resultados sugerem que inserir imagens de treino em diferentes distâncias e ângulos de captura podem beneficiar o modelo final.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver tecnologia para monitorar regiões críticas do perímetro operacional terrestre do Centro de Lançamentos de Alcântara para manter a base segura de possíveis ações danosas e preparada para a identificação prévia e/ou a prevenção de não-conformidades durante as operações de lançamento de veículos aeroespaciais.

2.2 Objetivos específicos

1. Desenvolver um sistema integrado de software que realize identificação de pessoal.
2. Elaborar estrutura mínima de controle e tráfego de dados, entre servidor e sistema embarcado.
3. Realizar testes de desempenho dos modelos de detecção e reconhecimento facial.

2.3 Contribuições esperadas

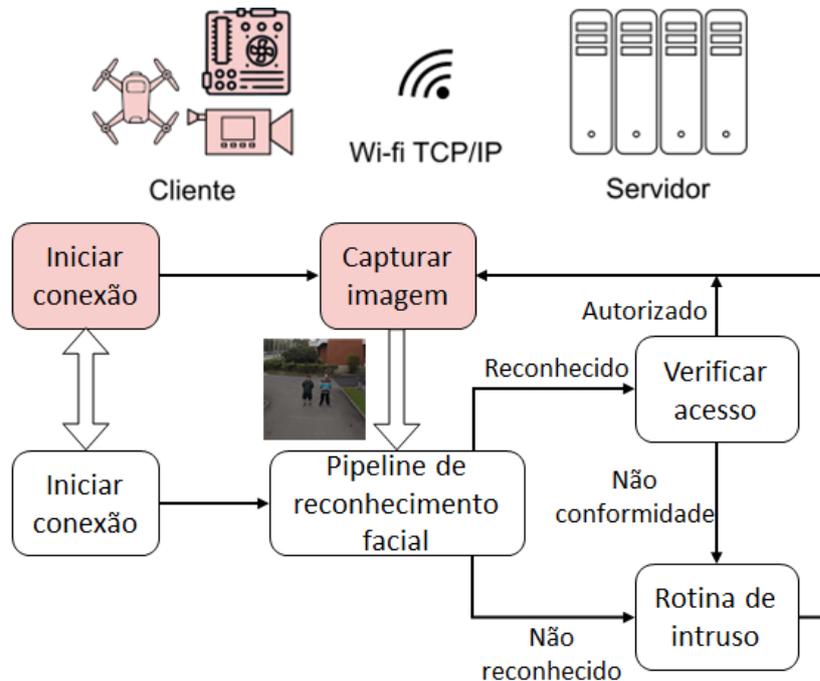
Este trabalho irá contribuir com o desenvolvimento da problemática de monitoramento de perímetro, incrementando a questão com uma aplicação inovadora em um centro espacial e desenvolvendo uma nova ferramenta.

Além disso, este trabalho poderá ser utilizado como referência para o desenvolvimento de futuros sistemas de monitoramento inteligentes e como exemplo de aplicações de inteligência artificial em resolução de problemas de monitoramento de perímetro.

3 METODOLOGIA

Propôs-se um sistema capaz de identificar pessoal autorizado por meio de reconhecimento facial, e ser embarcado em veículos não-tripulados. Um cliente composto por um computador de bordo, uma câmera e opcionalmente um veículo não tripulado, é responsável pela captura e transmissão de imagens. O Servidor é responsável pela recepção de imagens, reconhecimento facial e controle de acesso. A rede de modelo cliente-servidor mantém a conexão Wi-fi por soquetes TCP/IP. A Figura 15 mostra o fluxograma de atividades a serem desenvolvidas pelo servidor e cliente.

Figura 15 – Pipeline de vigilância e controle de acesso. Em vermelho as atividades desenvolvidas pelo sistema embarcado, e as atividades executadas pelo servidor em branco



Fonte – Autor

O sistema que foi proposto pode ser utilizados para atividades de vigilância diversas, com foco no controle de perímetro e reconhecimento de pessoal autorizado. As “rotinas de intruso” mencionadas na Figura 15 representa possíveis respostas a uma situação irregular: desde de notificar uma equipe responsável, disparar alarmes, acionar destacamento de segurança, trancar portas, etc.

O fluxo de trabalho do sistema é explicitado no fluxogramas das Figuras 16 e 17. As imagens serão capturadas por um Raspberry Pi 3B equipado com um módulo Pi Camera rev 1.3. Após a captura, as imagens são codificadas e decodificadas através do módulo HDF5 e transmitidas por conexão wi-fi, através de soquetes e protocolo TCP/IP. Para o

gerenciamento da transmissão por soquetes foi usada o pacote ImageZMQ (BASS, 2018), que estabelece e gerencia conexões entre um servidor e múltiplos clientes.

Figura 16: Algoritmo básico do cliente

```

início
  Conecte ao servidor
  Envie mensagem inicial ao servidor
  repita
    Capture imagem da câmera
    Envie imagem para o servidor
    Esperar solicitação de imagem.
  até desconectar do servidor;

```

Figura 17: Algoritmo básico do servidor

```

início
  Iniciar servidor
  repita
    Estabeleça conexão com cliente
    para cada cliente faça
      Receber imagem.
      Efetuar reconhecimento facial
      se Pessoa não autorizada então
        Iniciar rotina de intruso.
        Solicitar nova imagem ao cliente
    até Desativar servidor;

```

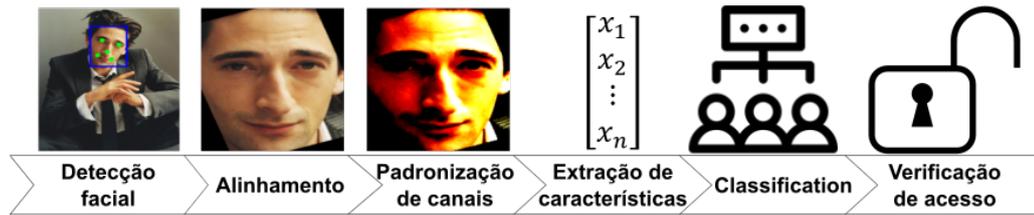
3.1 Pipeline de reconhecimento facial e autorização de acesso

Durante o funcionamento do sistema, Figura 12, as imagens foram adquiridas pelo Cliente e transmitidas para o servidor. Após a recepção, a imagem foi entregue ao processo de detecção facial, responsável por vasculhar a imagem em busca das faces e seus pontos-chave (olhos, nariz e boca). As faces detectadas foram então recortadas, alinhadas e padronizadas durante o pré-processamento da imagem. Cada face teve suas características extraídas e o vetor resultante entregue ao algoritmo de classificação, que em seguida tiveram suas credenciais de acesso verificadas.

3.1.1 Detecção Facial

Para o desenvolvimento do projeto foram selecionados dois algoritmos diferentes de detecção facial: MTCNN e Viola-Jones. Os modelos não foram utilizados juntos, mas como alternativas intercambiáveis onde o modelo MTCNN se destacou por sua precisão e o modelo de Viola-Jones por sua velocidade.

Figura 18 – Pipeline de reconhecimento facial.



Fonte – Autor

Para cada algoritmo foi desenvolvida uma classe, e para possibilitar a intercambialidade, usou-se tipagem *ducktyping*¹, assim, apesar de internamente os algoritmos serem distintos e apresentarem comportamentos muito diferentes, os métodos e atributos de cada classe possuem os mesmos identificadores, entradas e saídas. Dessa forma, do ponto de vista das demais classes do sistema de vigilância, não há diferença na utilização das classes de detecção facial.

O principal processo de detecção facial foi o modelo de aprendizagem profunda MTCNN, capaz de retorna a localização das faces encontradas, e seus pontos-chave. A detecção dos pontos-chave, especialmente dos olhos, foi essencial para a etapa de alinhamento facial. No desenvolvimento do código utilizou-se um modelo MTCNN de código aberto (CENTENO, 2018), e partindo desta implementação desenvolveu-se uma classe herdeira que modifica o método de detecção para tornar a saída compatível com os outros *frameworks* utilizados no sistema de vigilância. O modelo MTCNN propriamente dito foi tratado como *blackbox*² e a classe desenvolvida focou-se na adaptação das entradas e saídas do modelo.

A implementação do algoritmo de Viola-Jones utilizou como base a versão disponibilizada na biblioteca OpenCV (BRADSKI, 2000). Afim de detectar as faces e seus ponto-chave, a classe foi desenvolvida para executar o algoritmo de Viola-Jones duas vezes seguidas, primeiro vasculhando a imagem em busca de rostos, e em seguida vasculhando cada face encontrada em buscas de olhos. O resultado deste processo foi a localização das faces e dos olhos.

3.1.2 Alinhamento facial

O alinhamento facial é um importante passo intermediário para atividades de análise facial, sendo utilizado para aumentar a robustez dos processos de reconhecimento facial diminuindo o impacto gerado por variações de posição (JIN; TAN, 2017). Na literatura já foram propostos diversos modelos de alinhamento, alguns baseados em geometria espacial rebuscada, e outros baseados em modelos de aprendizagem profunda (BULAT;

¹ Um estilo de programação que determina um objeto através de chamadas aos seus métodos e atributos (Python Software Foundation, 2021)

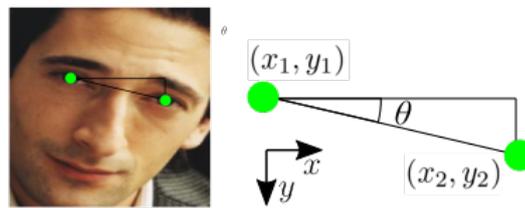
² Modelo que pode ser analisado em termos de suas entradas e saídas, sem considerar seu funcionamento interno.

TZIMIROPOULOS, 2017; JIN; TAN, 2017; KOWALSKI; NARUNIEC; TRZCINSKI, 2017; ZHU et al., 2016)

Optou-se, entretanto, por uma abordagem simplificada de geometria plana, utilizando a posição (x, y) já conhecida dos olhos, e calculando o ângulo θ entre eles pela Equação 3.1, as imagens eram então rotacionadas $-\theta$ radianos ao redor de seu centro de forma a manter os olhos em uma mesma linha horizontal.

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (3.1)$$

Figura 19 – Alinhamento facial



Fonte – Autor

A simplicidade do método de alinhamento favoreceu a velocidade de processamento, porém apresentou algumas limitações: ① o centro de rotação da imagem é próximo, mas não igual, ao centro da imagem; ② não foi consideradas nenhuma forma de distorção lateral; e ③ quando não foi possível encontrar ambos os olhos, o método não pôde ser aplicado. Após o alinhamento, cada imagem foi redimensionada por interpolação bilinear para 160×160 pixels. A Figura 20 mostra alguns exemplos de imagens alinhadas.

3.1.3 Padronização de canais

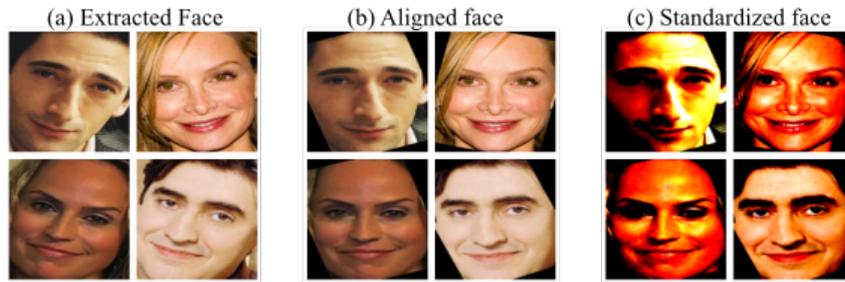
Antes de entregar uma imagem à rede neural de extração de características, aplicar processos de normalização na imagem podem aumentar significativamente a qualidade da classificação (PATRO; SAHU, 2015), fazendo com que as imagens tenham uma distribuição de pixels padronizadas.

Aplicou-se o processo de padronização de pixels, Equação 3.2, também chamado z-score, normalizando os pixels de forma a obter a média zero e o desvio padrão igual a um. Além de normalizar a distribuição dos pixels, este processo ajuda a diminuir o impacto que as diferenças de iluminação exercem sobre a qualidade do reconhecimento facial (PATRO; SAHU, 2015). A Figura 20 mostra alguns exemplos de imagens padronizadas.

$$Img_{std} = \frac{Img - \overline{Img}}{\sigma(Img)}, \quad (3.2)$$

onde, Img_{std} é a imagem padronizada, \overline{Img} é a média dos pixels, e $\sigma(Img)$ é o desvio padrão dos pixels da imagem.

Figura 20 – Exemplo de faces após cada etapa de préprocessamento



Fonte – Adaptado de Ng e Winkler (2014)

3.1.4 Extração de características

Após ser alinhada, redimensionada e padronizada, as faces detectadas puderam ser entregues ao método de extração de características. Após uma revisão de literatura, selecionou-se o modelo de aprendizagem profunda FaceNet, que apresentou taxa de acurácia de 99,63% na base de dados Labeled Faces in the Wild e de 95,12% na base de dados YouTube Faces DB (SCHROFF; KALENICHENKO; PHILBIN, 2015), além disso, este método de extração de características é capaz de ser utilizado com uma grande variedade de modelos de classificação. Selecionou-se uma implementação de código aberto do modelo FaceNet (TANIAI, 2018), esta implementação foi disponibilizada pré-treinada na base de dados MS-Celeb-1M, com 10 milhões de faces e 100 mil indivíduos.

Durante sua atuação, foram entregues ao FaceNet imagens RGB com os canais padronizados, e resolução 160×160 pixels, e como resultado o FaceNet retornou para cada imagem um vetor de 128 dimensões que encapsulou as características de uma determinada face. Neste espaço vetorial de 128 dimensões, a distância entre dois vetores é diretamente proporcional à similaridade das faces que representam.

3.1.5 Classificadores

O processo de classificação consiste em utilizar um modelo de aprendizagem de máquina para categorizar as faces de acordo com suas identidades. Foram avaliadas quatro possibilidades de classificadores, e cada um deles foi submetido a um processo de otimização de hiperparâmetros através do método GridSearch. A Tabela 1 mostra cada um dos modelos considerados e os hiperparâmetros selecionados para otimização. Em todos os casos, as faixas de valores dos hiperparâmetros foram decididos com base em exemplos presentes na documentação da API Scikit-Learn (PEDREGOSA et al., 2011).

Para o modelo kNN foram avaliados quatro hiperparâmetros: *Number of neighbors* e *Weights* estão diretamente relacionados com a precisão da classificação, e os *Algorithm* e *Leafsize* estão associados a velocidade de classificação. No modelo SVM, os hiperparâmetros *C*, *Gamma* e *Kernel* afetam diretamente a acurácia do modelo, os valores do hiperparâmetro *Degree* só afetam o modelo quando o kernel é polinomial, e o formato da função de decisão

Tabela 1 – Modelos e respectivos hiperparâmetros a serem otimizados

Modelo	Hiperparâmetros
kNN	n_neighbors: [3,5,7], weights: ['uniform', 'distance'], algorithm: ['ball_tree', 'kd_tree', 'brute'], leaf_size: [10,20,30,40,50]
SVM	C: [1, 10, 100, 1000], kernel: ['linear', 'rbf', 'sigmoid', 'poly'], degree: [3,5,7], gamma: [0.001, 0.0001, 'auto', 'scale'], class_weight:['balanced', None], decision_function_shape: 'ovo'
Tree	criterion: ['gini', 'entropy'], max_features: ['auto', 'sqrt', 'log2'], splitter: ['best', 'random'], class_weight: [None, 'balanced']
Forest	n_estimators: [100, 300, 500], max_features: ['auto', 'sqrt', 'log2'], criterion: ['gini', 'entropy'], ccp_alpha: [0.0, .5, 1.]

Fonte – Autor

é definido para “one-vs-one”, uma metodologia de treino rápida e sem impacto significativo na precisão.

Para o modelo Decision Tree avaliou-se os hiperparâmetro *Criterion* função que mede a qualidade de uma ramificação, o *Splitter* que define a estratégia de ramificação em cada nó, e o *Max Feature* que determina a quantidades de características que devem ser observadas em cada ramo.

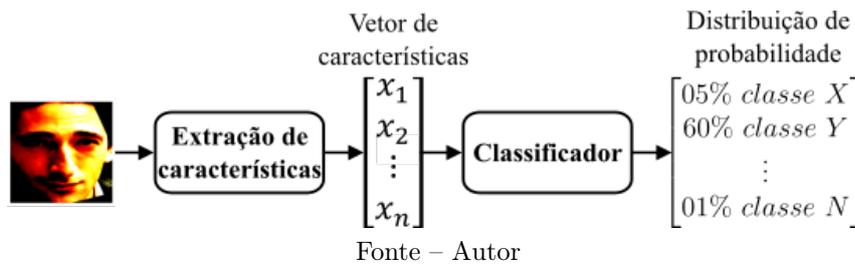
Por fim, para o modelo Random Forests o *Number of Estimator* é o hiperparâmetro mais importante, pois afeta diretamente a acurácia do modelo, além de requisitar maior poder computacional. O parâmetro *Cost Complexity* determina um limiar de complexidade para as sub arvores do modelo.

Do ponto de vista dos classificadores cada uma das identidades presentes na base de dados foi considerada como uma classe, e cada classe foi representada por um número de identificação distinto. Durante a etapa de treino cada um dos classificadores teve seus hiperparâmetros otimizados na base de dados FaceScrub, pelo método Grid Search, testando todas as combinações de hiperparâmetros possíveis e buscando a que apresentava maior acurácia. Em cada iteração da busca utilizou-se validação cruzada k-Fold com três pastas. Posteriormente cada classificador foi testado na base de dados DroneFaces, na qual avaliou-se o desempenho de cada modelo nas condições de trabalho de um drone.

Na etapa de inferência, e durante o funcionamento do sistema de vigilância, o resultado do processo de classificação é um vetor de distribuição de probabilidade, que determina a

porcentagem de chance de uma determinada imagem pertencer a uma determinada classe. Para o processo de otimização dos classificadores, considerou-se que a classe com maior probabilidade era a previsão do classificador.

Figura 21 – Processo de extração de características e de classificação. O processo inicia com uma imagem facial alinhada e padronizada, e conclui com um distribuição de probabilidades para cada uma das classes.



3.1.6 Verificação de acesso

A etapa final do sistema de monitoramento é a verificação de acesso, que cruza o resultado de um classificador com uma base de dados que contém as definições de nível de acesso de cada uma das classes. Na Figura 22 apresenta-se o processo de verificação de acesso.

O resultado de um classificador é um vetor de probabilidades que indica a chance de uma amostra pertencer a cada classe. Uma forma de gerenciar isso é definir que a resposta do classificador é a classe com maior probabilidade. Porém isso não é compatível com a possibilidade de uma face desconhecida ser exposta ao sistema. Espera-se, contudo, que uma face conhecida tenha um vetor de probabilidade com um valor de destaque perante as demais classes, e que uma face desconhecida tenha valores homogeneamente dividido entre todas as classes conhecidas.

Com base nesta hipótese, foi determinando um limite mínimo de probabilidade de classificação, exigindo que ao menos uma das classes atingisse um valor mínimo de probabilidade para que a face fosse reconhecida. Se nenhuma das classes atingiu esta probabilidade mínima, então a face foi dita como desconhecida. Se mais de uma classe atingiu este valor mínimo, então considerou-se que a classe com maior probabilidade era a correta. Este limite de probabilidade mínimo foi chamado de *threshold de classificação*.

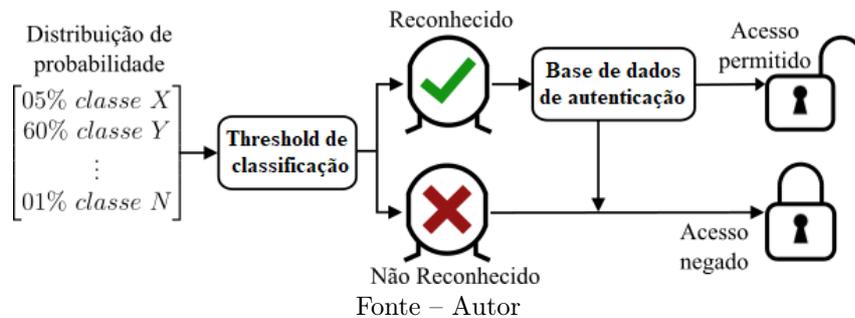
A determinação do valor exato do *threshold* de classificação foi feita empiricamente por meio de testes na base de dados FaceScrub. As classes da base de dados foram separadas em dois grupos: os Conhecidos que foram as classes usadas para treinar um classificador, e os Desconhecidos que foram as classes que não foram treinadas. Em seguida, o classificador efetuou inferências em imagens que pertenciam aos dois grupos, Conhecidos e Desconhecidos. O resultado desta inferência foi plotado em um gráfico mostrando as

curvas de Precisão-Recall contra a variação do threshold, e o ponto de otimização destas duas métricas foi considerado como o threshold ideal.

Após passar pelo *threshold* as faces reconhecidas foram cruzadas com o banco de dados de autenticação. Se a face reconhecida existisse no banco de dados, suas credenciais de acesso foram conferidas e determinou-se se o acesso era permitido ou negado. Nas demais situações em que a face não foi reconhecida, ou no caso de uma identidade reconhecida não existir na base de dados, então o acesso do indivíduo em questão é automaticamente determinado negado.

Para a realização dos testes, foi criada uma base de dados contendo apenas três entradas: um atributo inteiro que corresponde ao número de identificação da classe, um atributo *string* com identificação nominal do indivíduo, e um atributo booleano que determina se este é um indivíduo de "acesso permitido" ou de "acesso negado".

Figura 22 – Reconhecimento facial e autorização de acesso



3.2 Downsampling: Ampliação artificial dos dados de treino

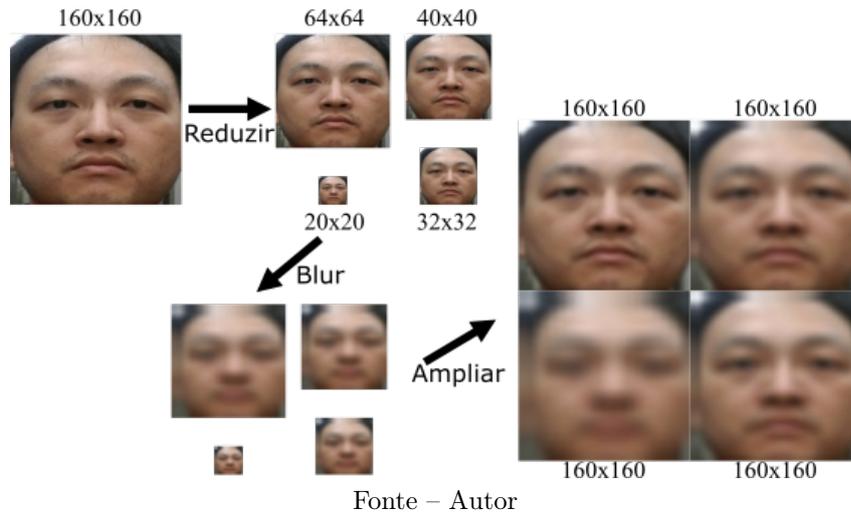
Estudos que aplicaram reconhecimento facial em condições de trabalho de drone notaram uma queda significativa na qualidade da identificação em situações de grandes distâncias ou de ângulos muito fechados (HSU; CHEN, 2015). Uma possível solução para este problema é inserir imagens de treino que correspondam a estas condições de trabalho (XU, 2020), mas nem sempre é possível adquirir imagens de treino que correspondam a estas condições. Nestes casos, é possível utilizar técnicas de *dataset augmentation*, que consistem ampliar artificialmente a base de dados aplicando modificações nos dados disponíveis.

A base de dados DroneFaces, que foi utilizada para testar a capacidade do sistema em trabalhar nas condições de um drone, não consta com imagens de treino que representem as reais condições de trabalho. Dispunha-se de poucas faces de treino, de alta resolução, e próximas do alvo, as faces de inferência, por outro lado, possuíam uma grande variedade de resoluções, e em diferentes distâncias, alturas e ângulos em relação ao alvo, 28. Para sanar essa questão, aplicou-se princípios de *dataset augmentation* sobre os dados de treino para

que representassem melhor as imagens de inferência (SHORTEN; KHOSHGOFTAAR, 2019).

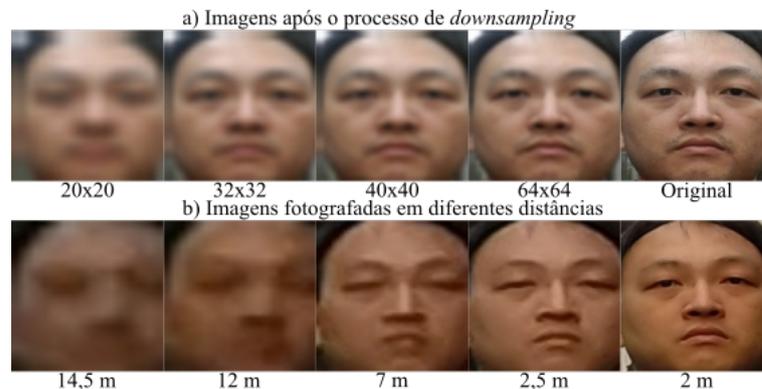
Para cada imagem de treino, foram geradas quatro cópias com resolução reduzida para 20×20 , 32×32 , 40×40 e 64×64 . Em seguida aplicou-se um leve blur gaussiano sobre cada uma das novas imagens e ampliou-se cada uma das imagens para a resolução padrão de 160×160 , Figura 23.

Figura 23 – Fluxo de trabalho do processo de *dataset augmentation*.



Esse processo gerou imagens com diferentes níveis de qualidade, tornando-as semelhantes às faces capturadas em grandes distâncias em relação ao alvo. Por consistir em um processo de redução da qualidade da imagem, chamou-se esse processo de *downsampling*, um processo que pôde ser categorizado como uma injeção de ruído nas imagens de treino. Na Figura 24 é possível comparar as imagens geradas pelo *downsampling* com imagens obtidas diretamente da base de dados DroneFace.

Figura 24 – (a) Exemplo de imagens após o *downsampling* em comparação com (b) imagens em diferentes distâncias disponíveis na base de dados DroneFace



3.3 Critérios de avaliação dos modelos

Para calcular o desempenho do sistemas, em cada teste foi contabilizada a quantidade de Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN). E como indicadores de desempenho foram utilizadas as métricas de classificação: acurácia, precisão, recall e F1-score (GÉRON, 2019).

- Acurácia: Taxa de previsões corretas.

$$acuracia = \frac{VP}{Total\ de\ Amostras} \quad (3.3)$$

- Precisão: É a taxa de previsões positivas corretas.

$$precisao = \frac{VP}{VP + FP} \quad (3.4)$$

- Recall: É a taxa de verdadeiros positivos.

$$recall = \frac{VP}{VP + FN} \quad (3.5)$$

- F1-Score: É a média harmônica entre a precisão e o recall. Assim, só se tem uma boa F1-score quando se têm bons valores de precisão e recall.

$$F_1 = 2 \times \frac{precisao \times recall}{precisao + recall} \quad (3.6)$$

Para a acurácia foi feito o cálculo do índice de confiança por proporção binomial com índice de confiança estatística $p < 0,05$ pela fórmula de Wilson:

$$IC = 1,96 \cdot \sqrt{\frac{acuracia * (1 - acuracia)}{n_{amostras}}} \quad (3.7)$$

Avaliou-se também a relação entre a precisão e o recall através da Curva Precisão-Recall, que mostrou a variação do recall e da precisão em relação a um determinado threshold de classificação. Um outro método gráfico de avaliação utilizado foi a Curva ROC (do inglês, *Receiver Operating Characteristic*) que plota a taxa de verdadeiros positivos em relação a taxa de falsos positivos. Estas duas curvas são mais facilmente visualizadas em classificações binárias, para classificação multi-classe é possível utilizar a Área Abaixo da Curva ROC (do inglês, *ROC Area Under the Curve - ROC AUC*).

A avaliação do sistema se deu na seguinte ordem: Primeiro, foram avaliados os sistemas de detecção facial MTCNN e Viola-Jones. Posteriormente, os modelos de classificação foram otimizados e testados. A etapa de verificação de acesso foi analisado em seguida, e através desta avaliação obteu-se um valor otimizado para o threshold de classificação. E por último, o desempenho do sistema como um todo foi testado.

Os dois modelos de detecção facial, MTCNN e Viola-Jones, foram avaliados sobre a base de dados DroneFaces. Foi possível avaliar o desempenho de cada um dos detectores em função da altura e distância em relação a face alvo através de um mapa de calor. Também aplicou-se as métricas de classificação para avaliar o desempenho geral dos modelos.

Os quatro modelos de classificação (kNN, SVM, Decision Tree e Random Forest) foram otimizados sobre a base de dados FaceScrub com objetivo de otimizar a acurácia de classificação, e testado na mesma base de dados para avaliar as demais métricas de classificação, além do tempo de treino e de teste da ROC AUC.

Em seguida os classificadores otimizados foram avaliados na base de dados DroneFaces, considerando que todas as faces foram corretamente detectadas sem nenhum falso positivo, o que possibilitou a análise do desempenho de cada classificador de acordo com a distância e altura em relação a face alvo através de um mapa de calor, e também foram analisadas as métricas de classificação. Para fins de comparação, também foi considerado um *Classificador Dummy*, um modelo que não efetua nenhum cálculo e apenas responde todas as inferências com a classe mais frequente.

O sistema de verificação de acesso foi analisado na base de dados FaceScrub, dividindo as classes desta base em dois grupos: Conhecidos e Desconhecidos, assim a verificação de acesso foi considerada como um classificador binário com as classes. Essa consideração possibilitou a utilização da Curva Precisão-Recall para encontrar um ponto de otimização para o threshold de classificação, e também possibilitou o uso da Curva ROC para analisar visualmente o desempenho do sistema.

Para avaliar o desempenho do sistema como um todo, foi avaliada a velocidade de quantos frames por segundo (FPS) são processados em tempo real, enquanto analise imagens com três voluntários. Esta análise foi útil para determinar a qualidade geral do sistema, sendo indicativo para o custo computacional dos algoritmos implementados.

3.4 Materiais e desenvolvimento de software

O projeto é desenvolvido em Python, linguagem bem adotada na comunidade científica, e que conta com diversas ferramentas de aprendizagem de máquina. Os seguintes pacotes foram utilizados no desenvolvimento:

Python 3.7	Tensor Flow	Scikit-Learn	Scikit-Image
OpenCV	Keras	Numpy	ImageZMQ

O desenvolvimento seguiu o paradigma de Programação Orientada a Objetos e tipagem *ducktyping*, com cada módulo desenvolvido como uma classe distinta, que se relacionam através dos métodos presentes em cada objeto. Dessa forma, foi possível tratar cada Cliente como um objeto que contém métodos de captura e transmissão de imagem a serem acessados pelo Servidor. Os requisitos necessários para a inicializar o Cliente são tratados

por um método construtor, e o desligamento do cliente é tratado por um método destrutor que aciona o coletor de lixo de linguagem.

Cada etapa do fluxo de reconhecimento facial foi implementado através de uma classe. No caso específico da detecção facial foram implementadas duas classes, uma para a MTCNN e outra para o algoritmo de Viola-Jones. Para permitir a intercambialidade, os métodos públicos de ambas as classes foram desenvolvidos com os mesmos identificadores, recebendo os mesmos parâmetros, e retornando os mesmos tipos e estruturas de dados. O classe MTCNN foi implementada com base no modelo pré-treinado disponibilizado por Centeno (2018), e classe Viola-Jones foi implementada com base no algoritmo de Viola-Jones distribuído pela biblioteca OpenCV (BRADSKI, 2000). Ambas as classes se comunicam com as demais partes do código através da função membro pública 'detect', que recebe uma imagem BGR e retorna o boundingbox e pontos chaves de cada face.

As etapas de pré-processamento de imagem, por sua vez, foram programadas em funções não-membro independentes. Essa abordagem permite que estas funções sejam facilmente integradas dentro de cada uma das classes do sistema. Foram desenvolvidas funções específicas para conversões entre as diferentes escalas de cores utilizadas (RGB, BGR e escala de cinza), uma função específica para extrair as faces da imagem, uma função para o alinhamento das faces, e uma outra para aplicar a padronização z-score.

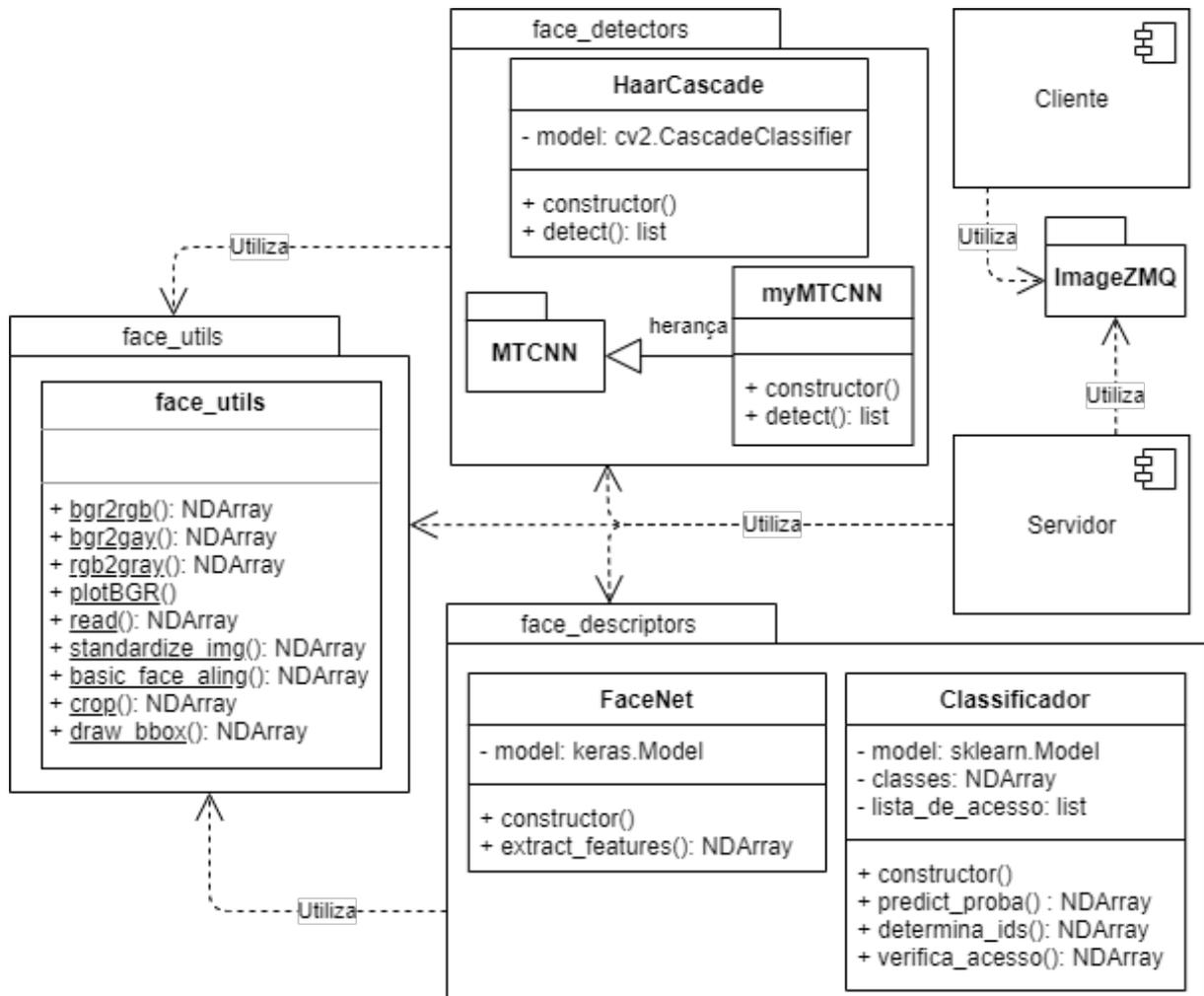
Foi desenvolvida uma classe para encapsular o modelo de extração de características, utilizando uma versão pré-treinada de código aberto do modelo FaceNet (TANIAI, 2018; SANDBERG, 2017). No método construtor da classe o modelo pré-treinado é inicializado através da API Keras. Essa classe se conecta com as demais através do método 'extract_features', que recebe uma imagem BGR, uma lista com os boundingboxes, e um dicionário que contém a localização dos ponto-chave de cada rosto, o método então acessa as funções de recorte, alinhamento e padronização, aplica a extração de característica, e retorna um vetor de 128 dimensões para cada rosto.

No desenvolvimento do sistema de comunicação foi utilizada a biblioteca ImageZMQ (BASS, 2018), uma implementação de ZeroMQ e TCP/IP desenvolvida para controlar a troca de imagens entre um servidor e um cliente, e gerenciar a fila de mensagens.

A Figura 25 mostra o diagrama UML do sistema. O Cliente e o Servidor são tratados como componentes do sistema, e ambos utilizam o pacote ImageZMQ (BASS, 2018) para realizar a comunicação. O pacote *face_utils* foi desenvolvido contendo diversas funções estáticas com utilidades para o processamento de imagens faciais. O pacote *face_detectors* contém as classe *HaarCascade* e a classe *myMTCNN* que herda suas características do pacote *MTCNN* desenvolvido por Centeno (2018). A pacote *face_descriptor* contém a classe de extração de características *FaceNet*, e a classe *Classificador* que têm como principal atributo um modelo da API Scikit-Learn.

O sistema embarcado foi desenvolvido utilizando um Raspberry Pi modelo 3B e um módulo Pi Camera rev. 1.3 de 5 MP 1080p30. O sistema operacional utilizado é o

Figura 25 – Diagrama de classes do sistema



Fonte – Autor

Raspberry Pi OS Kernel 5.10. Para treinar os modelos de aprendizagem de máquina será utilizada uma instância do Google Colaboratory. Para fins de testes, o servidor é executado em um Lenovo Ideapad L340, Windows 10, processador Core i5-9300H, 8GB de memória RAM.

3.5 Datasets

Para avaliar o desempenho do sistema em detectar e reconhecer faces, foram selecionados dois bancos de dados de imagens de acesso público: FaceScrub e DroneFace.

O FaceScrub é uma bases de dados públicas contendo imagens faciais, Figura 26. As faces foram extraídas de imagens públicas encontradas na internet, passando por um processo automático de detecção de faces, e posteriormente por um processo de conferência manual. Atualmente, o banco de dados consta com 42596 imagens de 530 identidades de personalidades pública, distribuídas pela licença *Creative Commons*. (NG; WINKLER, 2014)

Figura 26 – Exemplo de imagens disponibilizadas na base de dados FaceScrub



Fonte – Adaptado de Ng e Winkler (2014)

Uma das vantagens de utilizar a FaceScrub é grande diversidade de imagens, contendo pessoas de diferentes gêneros, etnias e idades. Há também uma grande diversidade interna a uma mesma identidade, contendo faces de pessoas em diferentes fases da vida, desde de faces infantis à faces adultas de uma mesma pessoa, e em diferentes estilos de moda, como os exemplos da Figura 27

Figura 27 – Exemplos de imagens da FaceScrub, destacando variações de posição, idade e aparência dentro de uma mesma classe.



Fonte – Adaptado de Ng e Winkler (2014)

O outro conjunto de dados que foi utilizado é o DroneFace, dataset disponibilizado por Hsu e Chen (2017), que se destaca por reproduzir as condições de trabalho de um drone, oferecendo imagens capturadas de diferentes distâncias e alturas em relação a face alvo. O banco de dados contém 620 imagens, de 11 indivíduos, com resolução 3680×2760 que simulam as condições de trabalho de um drone, como mostrado na Figura 28(a). Para cada indivíduo são fornecidas 6 ou 7 imagens de treino com resolução média de 900×800 pixels, como as mostradas na Figura 28(b).

As fotos simuladas foram tiradas em alturas que variam entre $1,5m$ e $5m$ e distâncias variadas entre $2m$ e $17m$, totalizando 124 imagens para cada indivíduo. Devido a estas

mudanças, a resolução das faces extraídas varia entre 23×31 a 384×384 pixels, Figura 28(c).

Figura 28 – Exemplo de imagens disponibilizadas na base de dados DroneFace



Fonte – Adaptado de Hsu e Chen (2017)

4 RESULTADOS E DISCUSSÃO

4.1 Avaliação dos modelos de detecção facial

Para avaliar a capacidade de detecção facial em um cenário adversos, utilizou-se a base de dados DroneFace. Os resultados podem ser vistos na Figura 29. É perceptível a grande influência que a variação de altura e distância sobre a qualidade das previsões de ambos os modelos.

Pela análise dos mapas de calor podemos dizer que a região ideal de trabalho é de até $3m$ de altura e até $8,5m$ de distancia para o modelo de Viola-Jones, e para o modelo de MTCNN de até $5m$ de alturas e até $11,5m$ de distância. Nota-se uma certa superioridade do modelo MTCNN, e um desempenho satisfatório do algoritmo de Viola-Jones.

Uma segunda análise é feita na Tabela 2 que apresenta o resultado geral dos modelos sobre as métricas de avaliação. Enquanto o desempenho dos modelos parece razoavelmente equivalentes na Figura 29, uma grande discrepância entre os dois métodos pode ser vista na Tabela 2, que mostra que apesar de uma quantidade similar de Verdadeiros Positivos, o método de Viola-Jones possui valores muito maiores de Falsos Positivos e Falsos Negativos. Isso se traduz no desempenho superior da MTCNN quando se trata das métricas de Precisão, Recall e F1-Score.

Algo interessante a se notar é que no modelo MTCNN a maioria dos Falsos Positivos se concentram em imagens de 4 ou 5 metros de altura e de mais de 10 metros de distância, enquanto no modelo de Viola-Jones a quantidade de falsos positivos é uniformemente distribuída. Este efeito compromete o uso do algoritmo de Viola-Jones mesmo em situações em que a câmera e o alvo estariam à pequenas distâncias.

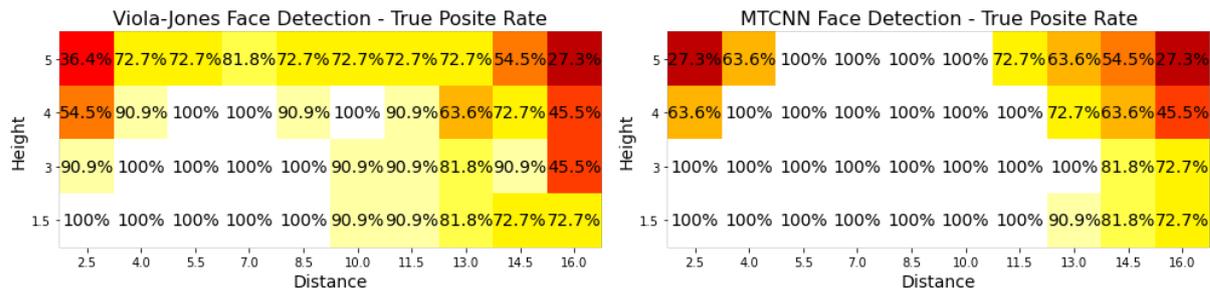
Os resultados obtidos são compatíveis com os observados na literatura. O desempenho do algoritmo de Viola-Jones já havia sido estudado em outros trabalhos com resultados semelhantes (HSU; CHEN, 2015; DINESH et al., 2018; DARYANAVARD; HARIFI, 2018). Já a MTCNN mostrou resultado promissor, com desempenho superior ao dos algoritmos proprietários estudados por Hsu e Chen (2017), e desempenho similar ao do modelo RetinaFace (LUO; CHEN; MA, 2020), com alta precisão e de detecção até cerca de $10m$ em relação a face alvo, e com vertiginosa queda de qualidade em distância maiores.

Tabela 2 – Resultado geral dos modelos de detecção facial na base de dados DroneFace.

Modelo	VP	FP	FN	Precisão	Recall	F1-score
MTCNN	1140	108	224	91,34%	83,57%	87,29%
Viola-Jones	1076	1072	288	50,09%	78,89%	61,28%

Fonte – Autor

Figura 29 – Taxa de Verdadeiros Positivos para detecções na base de dados DroneFace.



4.2 Avaliação do sistema de reconhecimento facial

4.2.1 Base de dados FaceScrub

Análises preliminares dos quatro algoritmos de classificação foram feitas utilizando os hiperparâmetros padronizados da biblioteca, onde avaliou-se acurácia, o tempo de treino e o tempo de teste para cada classificador, os resultados estão expostos na Tabela 3. Comparando os resultados obtidos pelos quatro classificadores, notou-se que o modelo Decision Tree tem acurácia muito inferior aos demais classificadores, e por isso foi abandonado. Ainda nesta avaliação preliminar, a tabela sugere que o modelo de SVM apresenta o valor mais alto de acurácia, e que o modelo de Random Forest apresenta maior velocidade de previsão. Para a análise seguinte, esses dois modelos foram considerados os mais promissores.

Tabela 3 – Resultados preliminares dos classificadores. Condições de treino: 29817 amostras de treino, 12779 amostras de teste, 530 identidades, padronização dos canais da imagem, alinhamento facial.

Modelo	Acurácia	Tempo de treino (segundos)	Tempo de testes (segundos)
kNN	98,8%	0,480	81,03
SVM	99,0%	101,5	297,8
Tree	56,2%	155,3	0,055
Forest	97,5%	960,70	6,33

Fonte – Autor

O principal objetivo do uso da base de dados FaceScrub foi otimizar os classificadores selecionados em uma base de dados extensa. Para cada classificador, a otimização foi efetuada de acordo com a Tabela 1, considerando todas as combinações possíveis de hiper-parâmetros, através de busca exaustiva e validação cruzada k-Fold com três pastas, buscando a combinação que resultasse em maior acurácia.

Os resultados detalhados do desempenho de cada classificador estão disponíveis na Tabela 4, e os hiper-parâmetros selecionados para cada modelo estão na Tabela 5. Após o

processo de otimização, todos os classificadores apresentaram melhoria na acurácia, com o SVM ainda se destacando com a maior acurácia e também com melhor precisão, recall e F1. O modelo Random Forest apresentou a maior velocidade de previsão, e com o valor de ROC AUC igual ao do modelo SVM, o que sugere que o modelo Random Forest como uma boa opção para classificação em tempo real.

Tabela 4 – Modelos de classificadores com hiper-parâmetros otimizados. Condições de treino: 29817 amostras de treino, 12779 amostras de teste, 530 identidades, padronização dos canais da imagem, alinhamento facial.

Modelo	Acurácia	ROC AUC	Precisão	Recall	F1	Tempo (segundos)
kNN	99,1 ± 0,16%	99,6%	99,1%	99,1%	99,1%	Treino: 0,162 Teste: 14,44
SVM	99,2 ± 0,15%	99,9%	99,3%	99,2%	99,2%	Treino: 126,4 Teste: 34,61
Forest	99,0 ± 0,17%	99,9%	99,1%	99,0%	99,0%	Treino: 943,0 Teste: 10,61

Fonte – Autor

Tabela 5 – Lista de hiper-parâmetros selecionadas após a etapa de otimização de classificadores sobre a base de dados FaceScrub.

Modelo	Hiperparâmetros
kNN	n_neighbors: 3, weights: 'distance', algorithm: 'ball_tree', leaf_size: 10
SVM	C: 10, kernel: 'rbf', gamma: 0.0001, class_weight: 'balanced', decision_function_shape: 'ovo'
Forest	n_estimators: 300, max_features: 'log2', criterion: 'entropy', ccp_alpha: 0.0

Fonte – Autor

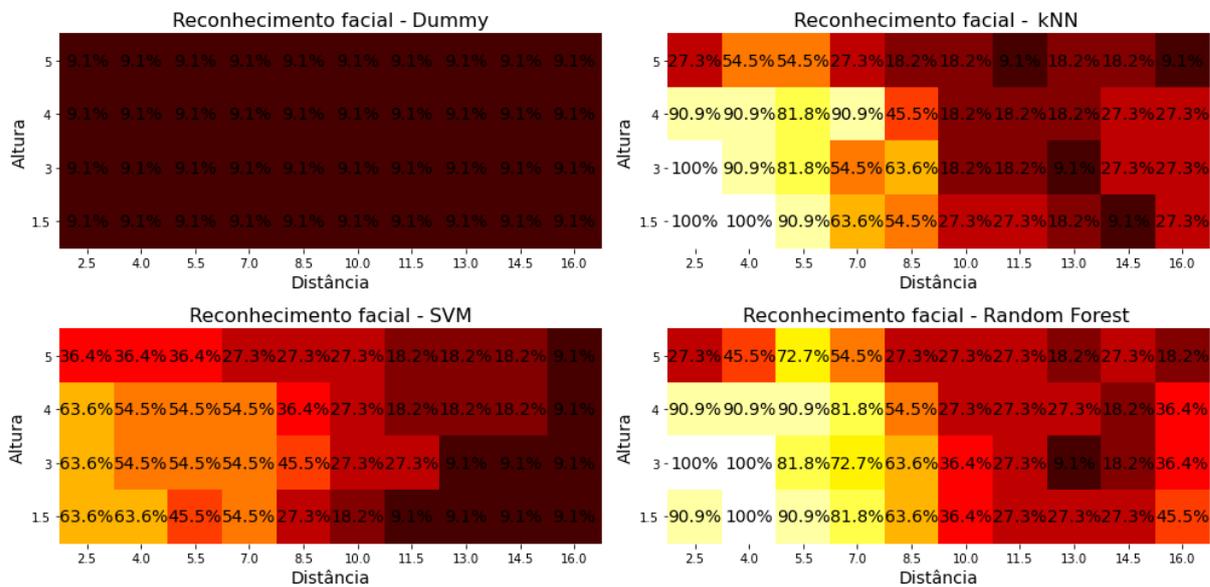
4.2.2 Base de dados DroneFace

Após encontrar os hiper-parâmetros otimizados, os classificadores foram submetidos a uma nova bateria de testes sobre a base de dados DroneFace para determinar a capacidade do classificadores de reconhecer faces nas condições de trabalho de um drone. Para efetuar

os testes considerou-se que todas as 1364 faces foram devidamente detectadas, para utilizar todas as faces da base no processo de avaliação.

Na primeira bateria de testes os classificadores foram treinados com as 72 imagens fornecidas para treino pela base de dados, os resultados gerais podem ser vistos na Tabela 6 e os resultados detalhados por distância e altura estão na Figura 30. Para fins de comparação, também foi utilizado um classificador *Dummy*, que sempre retorna a classe mais frequente independentemente da entrada inserida no classificador.

Figura 30 – Mapa de calor da acurácia de Reconhecimento Facial na base de dados DroneFaces. 1360 imagens de testes, 72 imagens de treino. Todas as imagens foram alinhadas e padronizadas.



Fonte – Autor

Tabela 6 – Reconhecimento facial na base de dados DroneFace. 1360 imagens de testes, 72 imagens de treino. Todas as imagens foram alinhadas e padronizadas.

Modelo	Acurácia	Precisão	Recall	F1-score
Dummy	9,1%	0,8%	9,1%	1,5%
kNN	43,8 ± 2,64%	58,4%	43,8%	43,8%
SVM	30,6 ± 2,45%	40,8%	30,6%	28,0%
Random Forest	46,3 ± 2,65%	60,6%	46,3%	46,6%

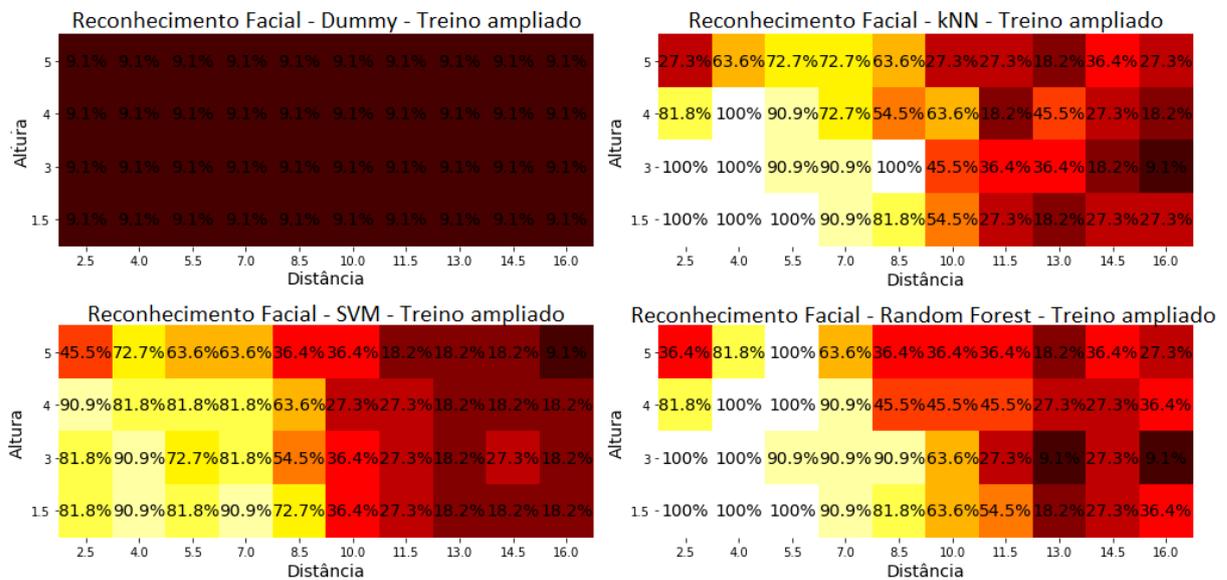
Fonte – Autor

Todos os classificadores (kNN, SVM e Random Forest) se mostraram muito melhores do que o valor base apresentado pelo classificador Dummy. O destaque negativo é o classificador SVM que, contrariando os resultados do teste anterior, apresentou resultados inferiores ao dos classificadores kNN e Random Forest.

As condições de treino da primeira bateria de testes são as mesmas utilizadas nos trabalhos de Hsu e Chen (2015), Hsu e Chen (2017), que obtiveram resultados melhores do que os apresentados na Figura 30 utilizando modelos proprietários.

A segunda bateria de testes utilizou o processo de downsampling para ampliar artificialmente a base de dados de treino para 360 amostras. O resultado dos testes utilizando a base de dados ampliada pode ser visto na Figura 31 e na Tabela 7

Figura 31 – Mapa de calor da acurácia de Reconhecimento Facial na base de dados DroneFaces. 1360 imagens de testes, 360 imagens de treino (base de dados ampliada). Todas as imagens foram alinhadas e padronizadas.



Fonte – Autor

Tabela 7 – Reconhecimento facial na base de dados DroneFace. 1360 imagens de testes, 360 imagens de treino (base de dados ampliada). Todas as imagens foram alinhadas e padronizadas.

Modelo	Acurácia	Precisão	Recall	F1-score
Dummy	9,1%	0,8%	9,1%	1,5%
kNN	56,5 ± 2, 63%	60,9%	56,5%	56,6%
SVM	46,3 ± 2, 65%	71,8%	46,3%	45,8%
Random Forest	56,2 ± 2, 64%	64,3%	56,2%	56,8%

Os resultados dos testes utilizando a base de dados ampliada foram significativamente melhores do que o processo de treino tradicional, com uma clara melhoria nos mapas de calor, e valores mais altos de Acurácia, Precisão, Recall e F1 para todos os classificadores. Os resultados obtidos pelo classificador Random Forest é comparável aos resultados obtidos por soluções proprietárias estudadas por Hsu e Chen (2015).

Os resultados reforçam a necessidade de produzir uma base de dados com *drone-like images* que representem no conjunto de treino as condições reais de trabalho. Dada a grande

melhoria observada nos resultados, e considerando que o processo de downsampling só replica o efeito da distância, podemos presumir que métodos mais sofisticados de ampliação de dataset podem gerar resultados ainda melhores. Isso é reforçado por trabalhos atuais que mostram que técnicas de ampliação de dataset podem melhorar a qualidade do reconhecimento facial, principalmente quando usando abordagens generativas básicas (ZHUCHKOV, 2021), em que o processo de downsampling se enquadra. Trabalhos futuros podem utilizar modelagem facial 3D e técnicas de filtragem mais elaboradas para ampliar a quantidade de poses e variedade de tamanho das faces presentes no conjunto de treino.

Quanto aos classificadores, notou-se que o SVM teve o pior desempenho neste conjunto de dados, o que contrasta com os resultados obtidos na base de dados anterior. Os modelos kNN e Random Forest parecem razoavelmente similares, o que indica uma adaptabilidade maior destes modelos. Vale lembrar contudo que o Random Forest tem maiores velocidades de previsão.

4.2.3 *Verificação de acesso*

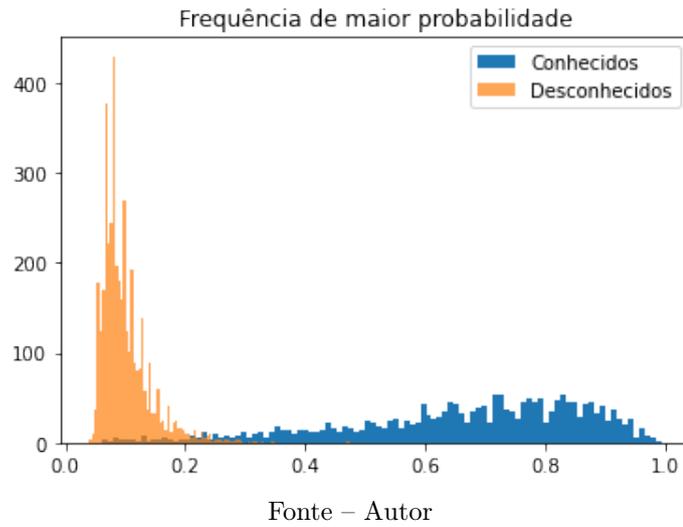
O resultado da classificação foi um vetor de probabilidade que indicava a chance que uma face analisada tem de corresponder a uma das classes conhecidas pelo classificador. Nos testes anteriores, foi considerado que a classe com maior probabilidade seria a resposta do classificador, o que não é viável quando existe a possibilidade de uma face desconhecida ser exposta ao classificador. O sistema de monitoramento foi preparado para trabalhar faces desconhecidas através da determinação de um limite mínimo de probabilidade que precisa ser atingido para efetivar o reconhecimento, o *threshold de classificação*

Para determinar o *threshold* as classes da base de dados FaceScrub foram aleatoriamente divididas entre Conhecidos e Desconhecidos. Metade das faces do grupo Conhecido foi utilizada para treinar um classificador Random Forest. Em seguida, uma quantidade igual de faces Conhecidas e Desconhecidas foram utilizadas para inferência, resultando em vetores de probabilidades. De cada um destes vetores foi extraído o maior valor de probabilidade e plotou-se o histograma da Figura 32:

O histograma mostra a maior probabilidade de cada previsão e sua frequência. Note que a maior parte das classes desconhecidas se concentra na ponta esquerda do gráfico, enquanto a maior parte das classes conhecidas se distribuem ao longo do lado direito do histograma. Essa distribuição confirma a hipótese de que as a distribuição de probabilidade das classes conhecidas possuem um valor de destaque. Pode-se ver no histograma que a maior parte das classes desconhecidas mantêm o valor de probabilidade abaixo de 20%, e poucas classes conhecidas se encontram abaixo deste valor.

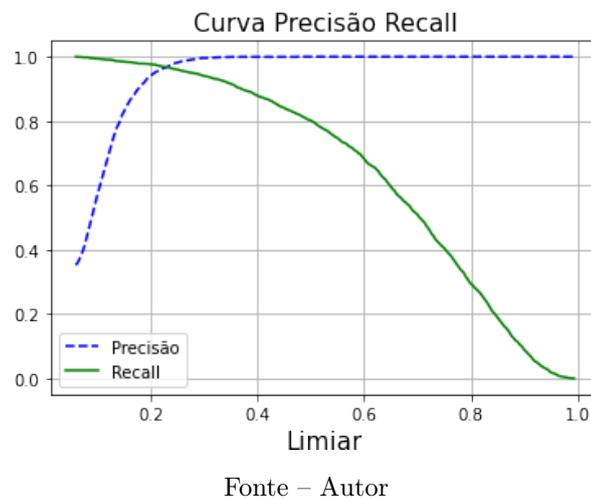
Em seguida foi plotada a curva Precisão-Recall versus o Threshold, Figura 33, que mostra o crescimento da precisão e queda do recall proporcionais ao aumento do threshold. De acordo com Géron (2019), o ponto de cruzamento entre as curvas de precisão e de recall costuma ser um bom ponto de escolha para o threshold, pois otimiza o valor da F1-Score.

Figura 32 – Histograma de frequência probabilidades



Esse ponto corresponde ao threshold de 25,3%, que é o mínimo necessário para que uma classe seja considerada reconhecida com uma precisão de 98,8% e recall de 95,9%.

Figura 33 – Curva Precisão-Recall de desempenho de verificação de classes conhecidas e desconhecidas



4.3 Avaliação de desempenho do sistema

Para descrever o desempenho geral do sistema, avaliou-se a quantidade de frames por segundos (FPS) processados ao longo de todos os processos envolvidos. O processamento foi realizado apenas na CPU, sem qualquer auxílio de placa de vídeo. O classificador utilizado foi o Random Forest treinado com faces de dois indivíduos, um com “acesso autorizado” e outro com “acesso negado”.

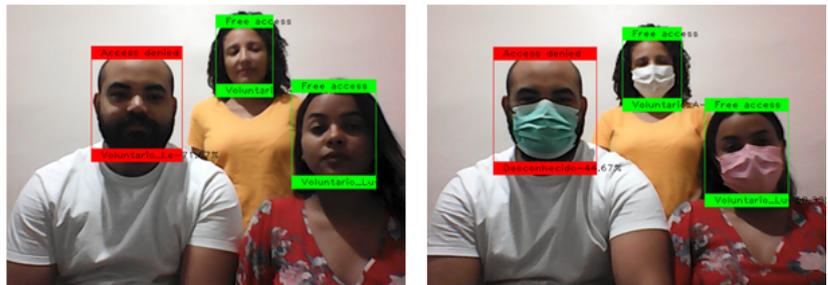
Para fins de teste, cada uma das transmissões foi executada pelo período de 1 minuto, utilizando três voluntários em ambiente fechado. A Figura 34 mostra alguns dos resultados obtidos durante os testes, as faces detectadas estão marcadas de verde quando a identidade é reconhecida e em vermelho quando a identidade não é reconhecida.

Figura 34 – Testes de transmissão. (a) Transmissão sem processamentos. Média de 25 *fps*. (b) Transmissão com HaarCascade e Facenet. Média de 5,8 *fps*. (c) Transmissão com MTCNN e Facenet. Média de 2,3 *fps*

a) Sem detecção facial
Sem reconhecimento facial
Transmissão Wi-fi TCP/IP
Máximo: 29 *fps*
Média: 25 *fps*
Mínima: 18 *fps*



b) Viola-Jones
FaceNet + Random Forest
Transmissão Wi-fi TCP/IP
Máximo: 10,1 *fps*
Média: 5,8 *fps*
Mínima: 3,4 *fps*



c) MTCNN
FaceNet + Random Forest
Transmissão Wi-fi TCP/IP
Máximo: 2,6 *fps*
Média: 2,3 *fps*
Mínima: 2,1 *fps*



Fonte – Autor

A primeira bateria de testes, Figura 34a, consistiu apenas na captura e transmissão sem nenhum processamento sobre as imagens. Nessas condições a velocidade média foi próxima aos 25 *fps*, com mínima de 18 *fps* e picos de 29 *fps*.

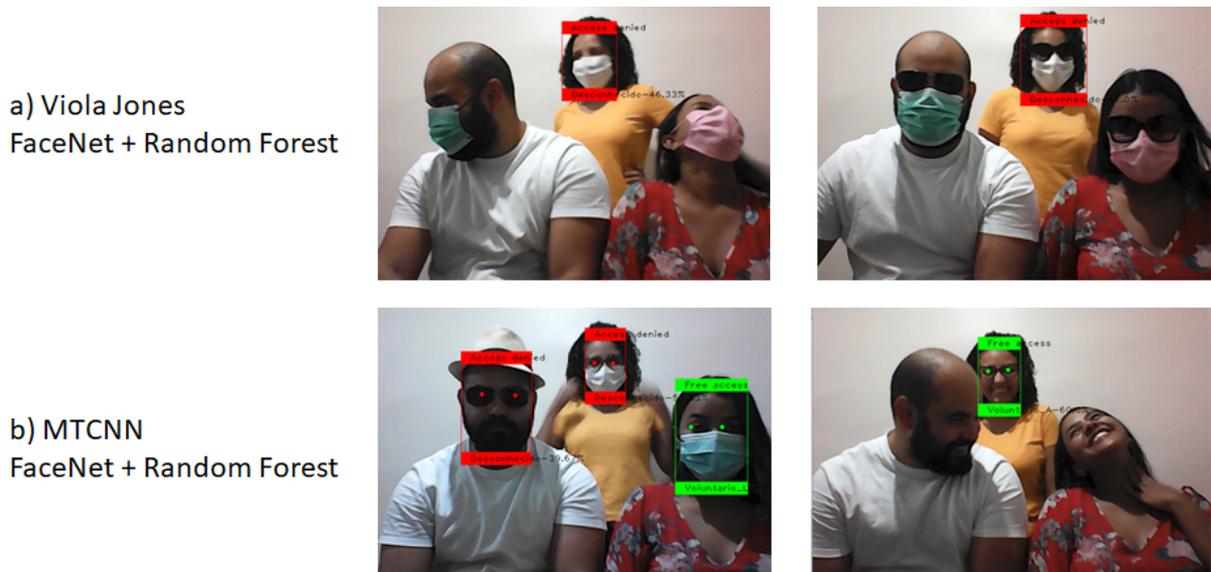
A segunda bateria de testes, Figura 34b, realizou a captura, transmissão das imagens, e executou o processo de reconhecimento facial utilizando o algoritmo de Viola-Jones como detector facial. Nessas condições, atingiu-se uma média de 5,8 *fps*, mínima de 3,4 *fps* e máxima de 10,1 *fps*. A quantidade de FPS é instável e varia conforme a quantidade de faces encontradas, atingido picos de velocidade em momentos que nenhuma face é encontrada. A queda de velocidade deve estar relacionada à detecção de ponto-chave, uma vez que para cada face encontrada é necessário um novo uso do algoritmo de Viola-Jones

para encontrar os olhos.

A terceira bateria de testes, Figura 34c, utilizou o modelo MTCNN para a detecção facial. Atingiu-se uma média de $2,3\text{fps}$, com mínima de $2,1\text{fps}$ e máxima de $2,6\text{fps}$. Ao contrário do anterior, a quantidade de frames se mantém estável e próxima da média independentemente da quantidade de faces detectadas.

A Figura 35 mostra algumas situações que ofereceram dificuldade aos detectores ou ao classificador. O uso de máscaras apresenta dificuldades diversas, podendo impedir o processo de detecção facial, e dificultado o correto reconhecimento facial, levando a situações em que uma face é incorretamente classificada como “desconhecida”. O uso de outros tipos de acessórios como chapéus e óculos não tiveram impacto notável sobre a detecção ou reconhecimento facial dos indivíduos, porém, variações de posição e postura facial se mostram como um importante obstáculo.

Figura 35 – Situações em de falha do pipeline de reconhecimento facial.



Os testes não foram realizados em campo no Centro de Lançamentos de Alcântara devido as dificuldades resultantes da pandemia de covid-19. Trabalhos futuros devem explorar testes de campo a medida que os riscos relacionados a pandemia forem mitigados.

5 CONCLUSÃO

Estudou-se o uso de detecção e reconhecimento facial para fins de vigilância inteligente aplicada ao Centro de Lançamentos de Alcântara. Os resultados são promissores, e indicam a viabilidade do projeto.

Observou-se que as técnicas de detecção facial MTCNN e Viola-Jones podem ser aplicadas com desempenho satisfatório, porém futuros estudos devem cobrir o uso de hardwares mais potentes. Em geral, o uso de acessórios não oferece limitações ao processo de detecção facial, contudo o uso de máscara ofereceu dificuldades ao algoritmo de Viola-Jones, acredita-se que o uso de faces mascaradas na produção de uma nova Haar Cascade possa diminuir este problema.

Nos testes de detecção facial com a base DroneFaces definiu-se um *sweetspot* de até 3m de altura e até 8,5m de distância para o modelo de Viola-Jones, e de alturas de até 5m e distancias de até 11,5m para o modelo de MTCNN. Deve-se levar em consideração que o modelo de Viola-Jones apresentou uma taxa de Falsos Positivos quase dez vezes maior que o modelo MTCNN.

Quanto aos modelos de reconhecimento facial, os estudos na base de dados FaceScrub indicaram o modelo SVM como o de melhor desempenho, e o modelo Random Forest como o de maior velocidade. Entretanto, os testes posteriores na base de dados DroneFaces mostram que o SVM tem baixa capacidade de reconhecimento facial em condições de drone, e que o modelo Random Forest tem melhor desempenho nessas condições. Diante disto, definiu-se o modelo de Random Forest como o mais adequado para uso em tempo real nessas condições de trabalho.

É preciso ressaltar que o uso de *downsampling* para ampliar a base de dados artificialmente apresentou resultados promissores, e que indicaram que adicionar imagens em condições de drone no conjunto de treino pode trazer significativas melhorias aos modelos de classificação.

A análise da curva Precisão-Recall sugeriu que um threshold de 25,3% seja aplicado às probabilidades de classificação, assim, uma identidade reconhecida com uma certeza menor que este threshold será negada, impedindo o acesso de identidades desconhecidas e aumentando a confiabilidade do sistema

Trabalhos futuros podem ampliar as funcionalidades do sistema, com o uso de GPS, câmera infravermelha, detectores de movimento, e sensores de ambiente. Também devem ser realizados testes de campo no CLA e testes de campo com o sistema embarcado montado em drones, uma vez que os testes apresentados foram realizados apenas em base de dados, novos testes realizados em campo podem trazer novas perspectivas sobre o uso de drones de vigilância. Outros trabalhos podem se concentrar na otimização de rotas de vigilância e o uso de enxames de drones para cobertura otimizada de área. Uma outra área

a ser pesquisada é a estrutura e cobertura de rede, que deve ser projetada para garantir robustez da conexão e a segurança e privacidade da rede. E por fim, deve-se implementar as possíveis rotinas de contenção de intrusos.

REFERÊNCIAS

- AGÊNCIA ESPACIAL BRASILEIRA. *Programa Nacional de Atividades Espaciais (PNAE) 2012 - 2021*. Brasília, 2012. Ministério da Ciência, Tecnologia e Inovação, Agência Espacial Brasileira.
- AGÊNCIA ESPACIAL BRASILEIRA. *Relatório de Gestão 2019*. Brasília, 2020. Ministério da Ciência, Tecnologia, Inovações e Comunicações. Agência Espacial Brasileira.
- AHONEN, T.; HADID, A.; PIETIKÄINEN, M. Face recognition with local binary patterns. In: SPRINGER. *European conference on computer vision*. [S.l.], 2004. p. 469–481.
- ALOYSIUS, N.; GEETHA, M. A review on deep convolutional neural networks. In: IEEE. *2017 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.], 2017. p. 0588–0592.
- AMIN, A. H. M.; AHMAD, N. M.; ALI, A. M. M. Decentralized face recognition scheme for distributed video surveillance in iot-cloud infrastructure. In: IEEE. *2016 IEEE Region 10 Symposium (TENSYMP)*. [S.l.], 2016. p. 119–124.
- ARSLAN, B. et al. Security perspective of biometric recognition and machine learning techniques. In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [S.l.: s.n.], 2016. p. 492–497.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 14882: Sistemas espaciais - Operações de centro de lançamento - Requisitos de Segurança*. Rio de Janeiro, 2002.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO 14620-2: Sistemas espaciais - Requisitos de segurança Parte 2: Operações de centro de lançamento*. Rio de Janeiro, 2013.
- BACKES, A. R.; SÁ JÚNIOR, J. J. d. M. *Introdução à visão computacional usando matlab*. [S.l.]: Alta Books, 2016. ISBN 978-85-508-0023-3.
- BARTELS, W. A atividade espacial e o poder de uma nação. *Desafios Do Programa Espacial Brasileiro*, p. 59–79, 2011.
- BASS, J. *ImageZMQ*. [S.l.]: GitHub, 2018. <<https://github.com/jeffbass/imagezmq>>.
- BLEDSON, W. W. The model method in facial recognition. *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, v. 15, n. 47, p. 2, 1964.
- BOLD, S.; SOSORBARAM, B.; LEE, S. R. Implementation of autonomous unmanned aerial vehicle with moving-object detection and face recognition. In: *Information Science and Applications (ICISA) 2016*. [S.l.]: Springer, 2016. p. 361–370.
- BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

- BRASIL. Ministério da Defesa. Comando da Aeronáutica. Departamento de Ciência e Tecnologia Espacial. Centro de Lançamento de Alcântara. *Manual de Segurança Operacional*. Alcântara - MA, 2017. 41 p. Disponível em: <<https://www2.fab.mil.br/cla/phocadownload/MAN-SGO-001-Manual-de-Seguranca-Operacional-CLA.pdf>>.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- BULAT, A.; TZIMIROPOULOS, G. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In: *International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 14.
- CENTENO, I. de P. MTCNN. *GitHub Repos*, 2018.
- CHENG, Z.; ZHU, X.; GONG, S. Surveillance face recognition challenge. *arXiv preprint arXiv:1804.09691*, 2018.
- CHOLLET, F. *Deep learning with Python*. 1. ed. [S.l.]: Manning Publications Co, 2018. ISBN 978-1-61729-443-3.
- COCORULLO, G. et al. Embedded surveillance system using background subtraction and raspberry pi. In: IEEE. *2015 AEIT International Annual Conference (AEIT)*. [S.l.], 2015. p. 1–5.
- DANG, K.; SHARMA, S. Review and comparison of face detection algorithms. In: *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*. [S.l.: s.n.], 2017. p. 629–633.
- DARYANAVARD, H.; HARIFI, A. Implementing face detection system on uav using raspberry pi platform. In: *Electrical Engineering (ICEE), Iranian Conference on*. [S.l.: s.n.], 2018. p. 1720–1723.
- DEEB, A.; ROY, K.; EDOH, K. D. Drone-based face recognition using deep learning. In: SPRINGER. *International Conference on Advanced Machine Learning Technologies and Applications*. [S.l.], 2020. p. 197–206.
- DHARAVATH, K.; TALUKDAR, F. A.; LASKAR, R. H. Improving face recognition rate with image preprocessing. *Indian Journal of Science and Technology*, Citeseer, v. 7, n. 8, p. 1170–1175, 2014.
- DINESH, M. et al. Development of an autonomous drone for surveillance application. *Proc. Int. Res. J. Eng. Technol. IRJET*, v. 5, p. 331–333, 2018.
- ERTEL, W. *Introduction to Artificial Intelligence*. 2. ed. [S.l.]: Springer, 2017. (Undergraduate Topics in Computer Science). ISBN 978-3-319-58487-4.
- FACELI, K. et al. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011. ISBN 978-85-216-1880-5. Disponível em: <<http://site.ebrary.com/id/10707120>>.
- FIX, E.; HODGES, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, JSTOR, v. 57, n. 3, p. 238–247, 1989.

FORÇA AÉREA BRASILEIRA. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/localizacao>>.

FORÇA AÉREA BRASILEIRA. *11 Foguete de Treinamento Intermediário é lançado com sucesso em Alcântara*. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/2014-12-11-17-51-57/137-11-foguete-de-treinamento-intermediario-e-lancado-com-sucesso-em-alcantara>>.

FORÇA AÉREA BRASILEIRA. *Campanhas - CLA*. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/2014-11-27-11-49-38>>.

FORÇA AÉREA BRASILEIRA. *História - CLA*. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/historias>>.

FORÇA AÉREA BRASILEIRA. *Implantação - CLA*. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/implantacao2>>.

FORÇA AÉREA BRASILEIRA. *Primeiro foguete nacional com combustível líquido é lançado com sucesso em Alcântara*. 2014. Disponível em: <<https://www2.fab.mil.br/cla/index.php/2014-12-11-17-51-57/224-primeiro-foguete-nacional-com-combustivel-liquido-e-lancado-com-sucesso-em-alcantara>>.

FORÇA AÉREA BRASILEIRA. *Cartilha CLA*. 2016. Disponível em: <https://www2.fab.mil.br/cla/images/conteudo/cartilha_cla_2016.pdf>.

GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: O'Reilly Media, 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GUPTA, I. et al. Face detection and recognition using raspberry pi. In: *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. Pune, India: IEEE, 2016. p. 83–86. ISBN 978-1-5090-3745-2. Disponível em: <<http://ieeexplore.ieee.org/document/8009092/>>.

HAMPAPUR, A. et al. Smart surveillance: applications, technologies and implications. In: IEEE. *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*. [S.l.], 2003. v. 2, p. 1133–1138.

HEARST, M. A. et al. Support vector machines. *IEEE Intelligent Systems and their applications*, IEEE, v. 13, n. 4, p. 18–28, 1998.

HJELMÅS, E.; LOW, B. K. Face detection: A survey. *Computer vision and image understanding*, Elsevier, v. 83, n. 3, p. 236–274, 2001.

HSU, H.-J.; CHEN, K.-T. Face recognition on drones: Issues and limitations. In: *Proceedings of the first workshop on micro aerial vehicle networks, systems, and applications for civilian use*. [S.l.: s.n.], 2015. p. 39–44.

HSU, H.-J.; CHEN, K.-T. Droneface: an open dataset for drone research. In: *Proceedings of the 8th ACM on multimedia systems conference*. [S.l.: s.n.], 2017. p. 187–192.

- IADANZA, E. d. E. S.; ANDRADE, M. P. d. Quilombolas de mamuna e canelatiua, alcântara-ma. Universidade de Brasília, 2021.
- JAFRI, R.; ARABNIA, H. R. A survey of face recognition techniques. *journal of information processing systems*, Korea Information Processing Society, v. 5, n. 2, p. 41–68, 2009.
- JIN, X.; TAN, X. Face alignment in-the-wild: A survey. *Computer Vision and Image Understanding*, Elsevier, v. 162, p. 1–22, 2017.
- JONES, M.; VIOLA, P. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, v. 3, n. 14, p. 2, 2003.
- KAEHLER, A.; BRADSKI, G. R. *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. First edition, second release. [S.l.]: O’Reilly Media, 2017. ISBN 978-1-4919-3799-0.
- KAKANI, V. et al. A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, v. 2, p. 100033, 2020. ISSN 2666-1543. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2666154320300144>>.
- KALIAPPAN, J. et al. Surveillance camera using face recognition for automatic attendance feeder and energy conservation in classroom. In: *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. [S.l.: s.n.], 2019. p. 1–5.
- KALRA, I. et al. Dronesurf: Benchmark dataset for drone-based face recognition. In: IEEE. *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. [S.l.], 2019. p. 1–7.
- KAPUR, S. *Computer vision with Python 3: image classification, object detection, video processing, and more*. 1. ed. [S.l.]: Packt, 2017. ISBN 978-1-78829-976-3.
- KOWALSKI, M.; NARUNIEC, J.; TRZCINSKI, T. Deep alignment network: A convolutional neural network for robust face alignment. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2017. p. 10.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105.
- KUBAT, M. *An Introduction to Machine Learning*. 2. ed. [S.l.]: Springer International Publishing, 2017. ISBN 978-3-319-63913-0.
- KUMAR, A.; KAUR, A.; KUMAR, M. Face detection techniques: a review. *Artificial Intelligence Review*, v. 52, n. 2, p. 927–948, Aug 2019. ISSN 1573-7462.
- KUMAR, G.; BHATIA, P. K. A detailed review of feature extraction in image processing systems. In: *2014 Fourth International Conference on Advanced Computing Communication Technologies*. [S.l.: s.n.], 2014. p. 5–12.

- LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1990. p. 396–404.
- LEI, Z. et al. Real-time face detection and recognition for video surveillance applications. In: *2009 WRI World Congress on Computer Science and Information Engineering*. [S.l.: s.n.], 2009. v. 5, p. 168–172.
- LUO, Y.; CHEN, S.; MA, X.-G. Drone lams: A drone-based face detection dataset with large angles and many scenarios. *arXiv preprint arXiv:2011.07689*, 2020.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MINAEE, S. et al. Going deeper into face detection: A survey. *arXiv preprint arXiv:2103.14983*, 2021.
- MISHRA, B. et al. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, Elsevier, 2020. Disponível em: <<https://doi.org/10.1016/j.comcom.2020.03.012>>.
- MUKHERJEE, S. et al. Convolutional neural network based face detection. In: *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*. [S.l.: s.n.], 2017. p. 1–5.
- NG, H.-W.; WINKLER, S. A data-driven approach to cleaning large face datasets. In: IEEE. *2014 IEEE international conference on image processing (ICIP)*. [S.l.], 2014. p. 343–347.
- NGUYEN, H.-Q. et al. Low cost real-time system monitoring using raspberry pi. In: IEEE. *2015 Seventh International Conference on Ubiquitous and Future Networks*. [S.l.], 2015. p. 857–859.
- OLIVEIRA, A. S. d.; ANDRADE, F. S. d. *Sistemas embarcados: hardware e firmware na prática*. [S.l.]: Editora Érica Ltda, 2006.
- ORTIZ, N. et al. Survey of biometric pattern recognition via machine learning techniques. *Contemporary Engineering Sciences*, v. 11, n. 34, p. 1677–1694, 2018.
- PAL, K. K.; SUDEEP, K. S. Preprocessing for image classification by convolutional neural networks. In: *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. [S.l.: s.n.], 2016. p. 1778–1781.
- PARKHI, O. M.; VEDALDI, A.; ZISSERMAN, A. Deep face recognition. British Machine Vision Association, 2015.
- PATRO, S.; SAHU, K. K. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, JMLR. org, v. 12, p. 2825–2830, 2011.
- POLIANYTSIA, A.; STARKOVA, O.; HERASYMENKO, K. Survey of hardware iot platforms. In: *2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S T)*. [S.l.: s.n.], 2016. p. 152–153.

PRINCE, S. J. D. *Computer vision: models, learning, and inference*. 1. ed. [S.l.]: Cambridge University Press, 2012. ISBN 978-1-107-01179-3.

Python Software Foundation. *Glossary - Python documentation*. Python Software Foundation, 2021. Disponível em: <<https://docs.python.org/3/glossary.html>>.

Raspberry Pi Trading LTD. *Raspberry Pi 4 Model B Product Brief*. [S.l.], 2020. Disponível em: <<https://static.raspberrypi.org/files/product-briefs/200521+Raspberry+Pi+4+Product+Brief.pdf>>.

RHODES, B. et al. *Foundations of Python network programming*. 3. ed. New York: Springer, 2014. ISBN 978-1-4302-5854-4.

RUSSELL, S. J.; NORVIG, P. *Inteligência artificial*. 3. ed. [S.l.]: Elsevier, 2013. ISBN 978-85-352-3701-6.

SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 21, n. 3, p. 660–674, 1991.

SAHA, A.; KUMAR, A.; SAHU, A. K. Face recognition drone. In: IEEE. *2018 3rd International Conference for Convergence in Technology (I2CT)*. [S.l.], 2018. p. 1–5.

SANDBERG, D. Facenet: Face recognition using tensorflow. *GitHub Repos*, 2017.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun 2015. Disponível em: <<http://dx.doi.org/10.1109/CVPR.2015.7298682>>.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, Springer, v. 6, n. 1, p. 1–48, 2019.

SILVA, A. F.; FISCH, G. Avaliação do modelo wrf para a previsão do perfil do vento no centro de lançamento de alcântara. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 29, n. 2, p. 259–270, 2014.

SKANSI, S. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Cham: Springer, 2018. ISBN 978-3-319-73004-2.

SRISUK, S. et al. Handcraft and learned feature extraction techniques for robust face recognition : A review. In: *2018 International Electrical Engineering Congress (iEECON)*. [S.l.: s.n.], 2018. p. 1–4.

SUN, Y.; WANG, X.; TANG, X. Deeply learned face representations are sparse, selective, and robust. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 2892–2900.

SZELISKI, R. *Computer vision: algorithms and applications*. 1. ed. [S.l.]: Springer, 2011. (Texts in computer science). ISBN 978-1-84882-934-3.

TAIGMAN, Y. et al. Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 1701–1708.

- TANIAI, H. Keras facenet. *GitHub Repository*, 2018.
- TIAN, D. ping et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, Citeseer, v. 8, n. 4, p. 385–396, 2013.
- TURK, M.; PENTLAND, A. Eigenfaces for recognition. *Journal of cognitive neuroscience*, MIT Press, v. 3, n. 1, p. 71–86, 1991.
- UPTON, E.; HALFACREE, G. *Raspberry Pi user guide*. [S.l.]: John Wiley & Sons, 2014.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. [S.l.], 2001. v. 1, p. I–I.
- WANG, M.; DENG, W. Deep face recognition: A survey. *CoRR*, abs/1804.06655, 2018. Disponível em: <<http://arxiv.org/abs/1804.06655>>.
- WANG, Y. et al. Face recognition in real-world surveillance videos with deep learning method. In: *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. [S.l.: s.n.], 2017. p. 239–243.
- XU, J. A deep learning approach to building an intelligent video surveillance system. *Multimedia Tools and Applications*, Springer, p. 1–21, 2020.
- ZAFEIRIOU, S.; ZHANG, C.; ZHANG, Z. A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, v. 138, p. 1–24, Sep 2015. ISSN 1077-3142.
- ZHAI, Y.; CHENG, X. Design of smart home remote monitoring system based on embedded system. In: *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering*. [S.l.: s.n.], 2011. v. 2, p. 41–44.
- ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, IEEE, v. 23, n. 10, p. 1499–1503, 2016.
- ZHU, X. et al. Face alignment across large poses: A 3d solution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 11.
- ZHUCHKOV, A. Analyzing the effectiveness of image augmentations for face recognition from limited data. *arXiv preprint arXiv:2105.08796*, 2021.