

Drone Autônomo com Visão Computacional para Identificação de Pessoas em Áreas Restritas

Alisson Garcia da Rosa¹, Igor Yepes¹

¹Instituto Federal Farroupilha (IFFar) - Campus Frederico Westphalen
Caixa Postal 169 – 98.400-000 – Frederico Westphalen – RS

alissongarciadarosa@gmail.com, igor.yepes@iffarroupilha.edu.br

Abstract. *This paper reports the development and analysis of an autonomous drone system designed to detect, recognize, and track individuals in restricted environments using classical Computer Vision methods. The proposed solution integrates the DJI Tello drone with Haar Cascade-based face detection and LBPH facial recognition, both implemented in Python and executed externally. The system was evaluated through a comprehensive set of experiments involving parameter tuning, lighting variation tests, distance-based performance assessment, and measurement of the drone's responsiveness during autonomous movement. The objective of this study is to examine whether these traditional algorithms, when combined with the hardware constraints of an entry-level drone, can effectively support real-time facial identification and tracking.*

Resumo. *Este artigo apresenta o desenvolvimento e a análise de um sistema autônomo baseado em drone, projetado para detectar, reconhecer e acompanhar pessoas em áreas restritas utilizando métodos clássicos de Visão Computacional. A solução integra o drone DJI Tello com o algoritmo Haar Cascade para detecção facial e o método LBPH para reconhecimento, ambos implementados em Python e executados externamente. A avaliação do sistema incluiu testes de calibração de parâmetros, experimentos sob diferentes condições de iluminação, variação de distâncias e análise da resposta de movimentação do drone durante o acompanhamento automático. O objetivo central do estudo é investigar se tais algoritmos tradicionais, aliados às limitações de hardware de um drone de entrada, são capazes de sustentar um processo confiável de identificação e tracking em tempo real.*

1. Introdução

Diversos estudos apontam que o uso de drones em aplicações de vigilância tem crescido devido à sua mobilidade e flexibilidade, porém muitas abordagens atuais baseiam-se em técnicas avançadas de aprendizado profundo, que demandam elevado poder computacional [Singh et al. 2018]

Estudos comparativos recentes reforçam a viabilidade do uso de detectores leves em aplicações de tempo real sobre plataformas com recursos computacionais limitados. Tais estudos indicam que modelos computacionalmente simples, como o LBP (*Local Binary Pattern*) e o Haar Cascade, apresentam menor latência, menor consumo de energia e maior adequação para execução em dispositivos embarcados, ainda que com uma

redução moderada de acurácia quando comparados a abordagens baseadas em aprendizado profundo [Balaji and Priyanka 2025]. De forma complementar, análises experimentais evidenciam que soluções computacionalmente eficientes mostram-se mais viáveis em cenários de baixo custo e aplicações em tempo real, reforçando sua aplicabilidade em sistemas embarcados [Liu et al. 2025].

Nesse sentido, torna-se relevante investigar a viabilidade do uso de técnicas clássicas de Visão Computacional, como Haar Cascade e LBPH, integradas a um drone de entrada. Esse problema afeta diretamente estudantes, pesquisadores e desenvolvedores que buscam soluções acessíveis para sistemas autônomos de vigilância, além de instituições que necessitam avaliar o custo-benefício entre diferentes abordagens tecnológicas.

Assim, a investigação proposta justifica-se por permitir avaliar, de forma experimental e sistemática, os limites e possibilidades dessas técnicas, contribuindo para a compreensão de sua aplicabilidade prática e servindo como base para decisões futuras sobre a adoção de soluções mais robustas.

O objetivo geral deste trabalho é desenvolver e avaliar a viabilidade de um sistema autônomo baseado em drone capaz de detectar, reconhecer e acompanhar pessoas, utilizando técnicas clássicas de Visão Computacional integradas ao drone de baixo custo. Para atingir esse objetivo, o estudo busca implementar e integrar os algoritmos Haar Cascade para detecção facial e LBPH para reconhecimento facial em um ambiente de processamento externo, avaliar o desempenho desses algoritmos sob diferentes condições de distância, iluminação e movimentação, analisar a influência da parametrização dos modelos na taxa de acertos e erros do sistema, e investigar a capacidade de resposta do drone durante o processo de acompanhamento automático de indivíduos. Além disso, o trabalho tem como propósito identificar as limitações impostas tanto pelos algoritmos utilizados quanto pelo hardware do drone, de modo a verificar a viabilidade dessa abordagem em cenários reais de vigilância e controle de acesso, bem como apontar possíveis melhorias e direções para pesquisas futuras.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, discutindo estudos que utilizam drones e técnicas de visão computacional e inteligência artificial aplicadas à vigilância e ao reconhecimento de pessoas. A Seção 3 descreve a metodologia adotada, incluindo as tecnologias utilizadas, o processo de implementação do sistema, o funcionamento integrado dos algoritmos de detecção e reconhecimento facial, bem como as metodologias de teste empregadas para avaliação do desempenho. Na Seção 4 são apresentados e discutidos os resultados obtidos a partir dos experimentos realizados, contemplando a análise dos algoritmos, das condições de iluminação, da distância e do comportamento de movimentação do drone. Por fim, a Seção 5 apresenta as conclusões do trabalho, destacando as principais contribuições, limitações identificadas e possíveis direções para trabalhos futuros.

2. Trabalhos relacionados

Uma proposta semelhante a do projeto em questão é o desenvolvido por [Maan et al. 2020], cujo foco está na detecção de objetos e reconhecimento de ações em tempo real, aplicados principalmente em contextos de segurança e vigilância. O sistema foi implementado em Python, utilizando bibliotecas como OpenCV e YOLO, e é

capaz de detectar até 82 objetos diferentes em um fluxo de vídeo contínuo. O reconhecimento de ações permite monitorar os movimentos dos sujeitos, categorizando sub-ações e associando-as a atividades completas. Os resultados demonstraram alta acurácia na detecção, embora com limitações quanto à taxa de quadros por segundo (FPS), exigindo, portanto, uma GPU robusta para garantir desempenho satisfatório. A principal contribuição do trabalho está na integração de técnicas de visão computacional com aprendizado profundo (deep learning) em aplicações de tempo real.

Em comparação, o presente projeto se diferencia quanto à implementação e ao foco de aplicação das tecnologias. Embora também utilize técnicas de visão computacional em tempo real, o objetivo principal é o reconhecimento e a identificação de pessoas, e não de objetos. Além disso, busca-se a implementação de um algoritmo mais leve, que dispense a necessidade de uma GPU de alto desempenho para obter os resultados desejados. Outro diferencial importante está na abrangência da área de monitoramento: enquanto o trabalho analisado está limitado ao uso de câmeras de segurança fixas, que observam uma única área, este projeto propõe o uso de drones, o que possibilita a cobertura de regiões mais amplas e dinâmicas, ampliando consideravelmente o escopo da aplicação

Outro trabalho relevante para este estudo é o proposto por [Singh et al. 2018], que apresenta um sistema de vigilância por drones em tempo real, voltado à identificação de indivíduos envolvidos em atividades violentas em áreas públicas. Inicialmente, o sistema utiliza a *Feature Pyramid Network* (FPN) para realizar a detecção de pessoas em imagens aéreas. Em seguida, a região da imagem onde a pessoa foi identificada é processada pela *ScatterNet Hybrid Deep Learning* (SHDL), uma rede neural profunda que realiza a estimativa de pose humana. A partir da orientação dos membros corporais estimados, é possível identificar comportamentos violentos utilizando um classificador SVM. O sistema ainda processa as imagens dos drones em nuvem, garantindo desempenho em tempo real. Os resultados obtidos mostraram que o modelo proposto supera técnicas de ponta em termos de identificação de indivíduos violentos com base na pose corporal.

Em perspectiva, o presente projeto também utiliza drones e técnicas de visão computacional em tempo real, porém com um foco distinto na identificação e reconhecimento facial de indivíduos, ao invés de analisar gestos e comportamentos violentos. Além disso, enquanto o trabalho de análise realiza o processamento em nuvem, o presente projeto busca uma solução mais leve e eficiente localmente, com menor dependência de infraestrutura externa, que pode tornar o projeto mais caro e menos acessível. Outro aspecto diferenciador é a finalidade da identificação, que aqui está voltada para reconhecimento de identidade em vez de análise de comportamento, levando em consideração que o objetivo proposto é o controle de acesso em áreas restritas.

Outro estudo relevante para esta pesquisa é o trabalho desenvolvido por [Canton et al. 2021], que propõe um sistema de reconhecimento facial embarcado em drones, utilizando técnicas de visão computacional e algoritmos de inteligência artificial. O projeto emprega o drone DJI Tello para captura de vídeo em tempo real, processando as imagens com bibliotecas como OpenCV e Dlib, aplicando métodos como HOG e redes neurais convolucionais (CNN) para detecção e extração de características faciais. O sistema inclui etapas de treinamento dos descritores faciais e utiliza o algoritmo KNN para realizar a identificação dos indivíduos durante o voo. Esse trabalho apresenta relação direta com o presente trabalho, uma vez que ambos exploram a utilização de drones equi-

pados com sistemas de visão computacional para detecção e reconhecimento facial em tempo real. O estudo utiliza o drone DJI Tello como plataforma de captura de vídeo e emprega a biblioteca OpenCV, para o processamento das imagens. Entretanto, enquanto o projeto apresentado faz uso de descritores baseados em HOG combinados com redes neurais convolucionais e o algoritmo KNN para identificação dos indivíduos, o presente trabalho adota técnicas clássicas de visão computacional, como Haar Cascade para detecção facial e LBPH para reconhecimento. Dessa forma, o estudo analisado serve como base comparativa e evidencia diferentes abordagens possíveis para a implementação de sistemas de reconhecimento facial embarcados em drones.

3. Metodologia

O projeto adotou duas abordagens metodológicas: experimental e aplicada. A abordagem experimental se justificou no objetivo de investigar o funcionamento e a precisão de drones voltados à vigilância, por meio de testes práticos. Já a abordagem aplicada se fundamentou na utilização de conhecimentos tecnológicos e científicos para propor uma solução concreta para o problema identificado. A experimentação, ainda que em ambiente controlado, permitiu avaliar a viabilidade e a efetividade da solução proposta, aproximando-a das condições reais de operação.

3.1. Tecnologias utilizadas

Para a construção do projeto, foi realizado um levantamento de requisitos com o objetivo de identificar os componentes de hardware necessários, entre eles um drone com capacidade de voo estável, equipado com uma câmera digital embarcada para captura de imagens, além de possuir compatibilidade com comandos remotos. Paralelamente, foi conduzido o levantamento dos componentes de software, incluindo linguagens de programação e bibliotecas adequadas tanto para o controle do drone quanto para a implementação dos algoritmos de detecção e reconhecimento facial.

Com base nesse levantamento, foram selecionados os recursos necessários para o desenvolvimento do sistema proposto. Para o hardware, a opção escolhida, foi baseada na disponibilidade de equipamento fornecidos pela universidade, nesse caso, o drone Tello, que atenderá aos requisitos definidos por ser equipado com uma câmera digital embarcada, apresentar estabilidade básica durante o voo e permitir o recebimento de comandos remotos [DJI 2018], características essenciais para a execução das tarefas propostas. No que se refere ao software, a linguagem de programação Python foi selecionada em função de sua elevada versatilidade e ampla adoção no meio científico e acadêmico. Python consolidou-se como uma das principais linguagens para computação científica devido à sua sintaxe simples, facilidade de prototipação e, principalmente, à extensa disponibilidade de bibliotecas especializadas [Oliphant 2007]. A partir dessa escolha, foram definidas duas bibliotecas principais: A OpenCV (*Open Source Computer Vision Library*) que é empregada para o tratamento e processamento de imagens digitais, bem como para a implementação de algoritmos de Visão Computacional e Inteligência Artificial. A OpenCV oferece suporte à integração com algoritmos de aprendizado de máquina e aprendizado profundo, permitindo o desenvolvimento de sistemas inteligentes capazes de interpretar informações visuais em tempo real, com elevado desempenho computacional e portabilidade entre diferentes plataformas [Bradski and Kaehler 2008]; e, para o controle e a comunicação com o drone Tello, foi utilizada a biblioteca DJITelloPy, uma

interface de alto nível desenvolvida em Python que abstrai os detalhes de comunicação direta com o dispositivo. Essa biblioteca permite o envio de comandos de voo, como decolagem, pouso, deslocamentos e rotações, além do acesso em tempo real ao fluxo de vídeo capturado pela câmera embarcada no drone [Fuentes Escoté 2018].

Para a detecção facial, foi utilizado o algoritmo Haar Cascade, implementado na biblioteca OpenCV. Esse algoritmo utiliza classificadores em cascata baseados em características Haar, que são padrões visuais simples aplicados para identificar variações de intensidade em regiões específicas de uma imagem [OpenCV 2024]. Uma das vantagens da utilização desse modelo é seu baixo custo computacional, o que o torna adequado para aplicações que exigem processamento em tempo real [Balaji and Priyanka 2025]. Além disso, o modelo já está disponível de forma pré-treinada na biblioteca, o que facilita a implementação sem a necessidade de um processo adicional de treinamento.

Para o reconhecimento facial, foi adotado o algoritmo LBPH (Local Binary Patterns Histograms), disponível na biblioteca OpenCV por meio da classe LBPHFaceRecognizer. Esse método converte a imagem do rosto em representações baseadas em padrões de textura local, comparando cada pixel aos seus vizinhos para gerar códigos binários. Em seguida, histogramas desses padrões são compilados em regiões da face, resultando em uma representação única e robusta para cada indivíduo, adequada a variações de iluminação e expressões faciais [OpenCV 2024, Ahonen et al. 2004]. O LBPH foi escolhido por ser relativamente leve em termos de processamento, podendo ser executado em tempo real mesmo em sistemas com recursos computacionais limitados [Balaji and Priyanka 2025], o que está alinhado com as restrições do ambiente utilizado no projeto. Além disso, assim como o Haar Cascade, o LBPH também é disponibilizado pela OpenCV de forma pré-treinada.

3.2. Implementação

Para desenvolver um sistema de reconhecimento facial robusto, inicialmente foram definidos os processos de detecção e reconhecimento, assim como o processo de controle autônomo do drone, e posteriormente foram conduzidos testes sistemáticos para identificar as melhores configurações de parâmetros, considerando apenas aqueles que exercem maior influência sobre o desempenho geral.

3.2.1. Funcionamento integrado dos algoritmos

O Haar Cascade pode ser empregado diretamente para a detecção facial, enquanto o LBPH é utilizado para o reconhecimento de indivíduos, sendo ambos algoritmos pré-treinados, evidenciando que esses algoritmos podem ser integrados de maneira imediata em sistemas de reconhecimento facial sem a necessidade de desenvolvimento completo dos modelos subjacentes [Beri et al. 2024]. Dessa forma, o principal trabalho envolvido na implementação de ambos não está na construção dos algoritmos, mas sim na correta parametrização de suas variáveis. Esses parâmetros determinam como cada algoritmo se comporta diante de diferentes condições, como iluminação, distância, tamanho do rosto e quantidade de ruído na imagem, e influenciam diretamente a estabilidade, precisão e sensibilidade do sistema como um todo [OpenCV 2024].

A primeira etapa do processo consiste no treinamento do algoritmo LBPH. O

mesmo opera conforme ilustrado na Figura 1. As imagens que compõem o conjunto de treinamento são carregadas e organizadas em diretórios separados por classes, onde cada diretório representa um indivíduo. Cada imagem passa primeiro pelo algoritmo Haar Cascade, responsável pela detecção da região facial e pelo descarte das áreas irrelevantes. A região recortada é então normalizada e enviada ao LBPH para extração das características faciais.

O algoritmo LBPH (Local Binary Patterns Histogram) fundamenta-se na técnica Local Binary Patterns (LBP), a qual realiza a análise da textura local da imagem por meio da comparação da intensidade de um pixel central com a de seus pixels vizinhos. Essas comparações resultam em padrões binários que são posteriormente convertidos em valores decimais, representando a textura da região analisada. Para o reconhecimento facial, a imagem do rosto é segmentada em células definidas pelos parâmetros `grid_x` e `grid_y`, sendo gerado um histograma dos códigos LBP para cada célula. A concatenação desses histogramas forma o vetor de características utilizado para representar o indivíduo no modelo final de reconhecimento facial [Ojala et al. 2002].

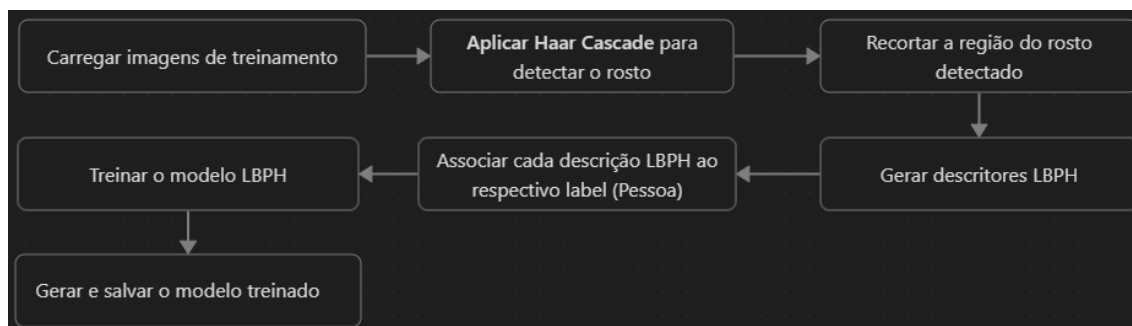


Figura 1. Fluxograma funcionamento do algoritmo LBPH

Após concluído o treinamento, o processo de detecção e reconhecimento facial em tempo real segue o fluxograma da Figura 2. Cada quadro capturado pela câmera é submetido ao Haar Cascade, que identifica possíveis regiões faciais. Em seguida, o LBPH extrai o vetor de características da face detectada e o compara com os vetores armazenados no modelo treinado. O sistema então gera uma previsão de identidade acompanhada por um valor de confiança ou *confidence score*, onde valores menores indicam maior similaridade. Com base nesse valor e nos parâmetros definidos, o rosto é classificado como conhecido ou desconhecido [Beri et al. 2024].

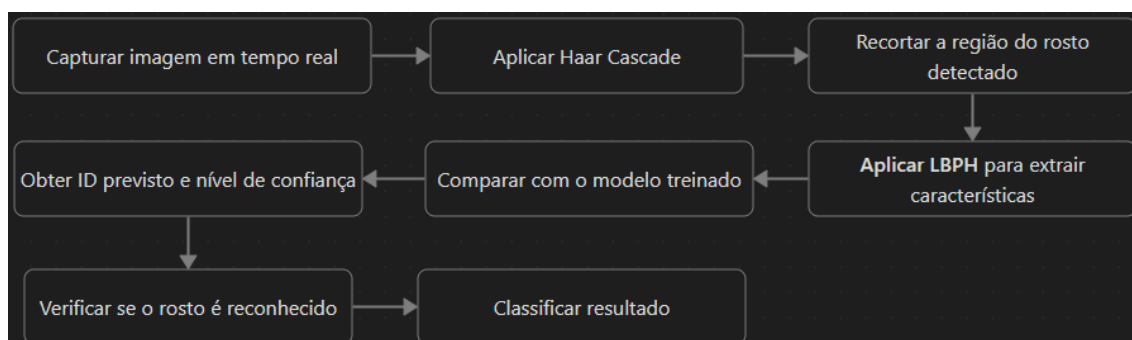


Figura 2. Fluxograma do funcionamento do reconhecimento facial

3.2.2. Movimentação autônoma do drone

A movimentação do drone é controlada com base nas informações extraídas do rosto detectado no quadro de vídeo. Inicialmente, o algoritmo identifica a posição e o tamanho da face utilizando o Haar Cascade. A partir das coordenadas do retângulo de detecção, calcula-se o centro do rosto e a sua área. Esses valores são comparados ao centro do frame, permitindo estimar o erro de alinhamento horizontal e vertical. Esse desvio é convertido em comandos de velocidade para o drone, onde o erro horizontal ajusta a rotação e o erro vertical controla o movimento de subida ou descida, ambos limitados por um valor máximo configurado.

Além disso, o tamanho do rosto é utilizado como uma métrica aproximada de distância: áreas menores indicam que o rosto está longe, fazendo o drone avançar, enquanto áreas maiores sugerem proximidade excessiva, levando o drone a recuar. Caso nenhuma face seja detectada, todas as velocidades permanecem zeradas, mantendo o dispositivo estático.

Ao final de cada ciclo, os valores calculados de avanço/retrocesso, deslocamento lateral, elevação e rotação são enviados ao drone. Assim, cada atualização da imagem recebida resulta em um novo conjunto de ajustes, produzindo um comportamento contínuo e responsivo, no qual o drone se alinha automaticamente ao rosto detectado e mantém uma distância aproximada do alvo.

3.3. Metodologia de testes

Diferentes componentes do fluxo do projeto possuem sensibilidades distintas: o Haar Cascade depende fortemente da escolha adequada de parâmetros como `scaleFactor` e `minNeighbors`; o LBPH é influenciado pela variabilidade luminosa e pela textura do ambiente; e o drone adiciona desafios práticos de movimentação e autonomia. Assim, segmentar o estudo em etapas metodológicas específicas permitiu controlar variáveis e observar comportamentos de forma independente, evitando que efeitos de um componente encubram ou distorçam a análise de outro. Sendo assim, essa sub-sessão irá descrever as metodologias de testes aplicadas para cada etapa do projeto.

3.3.1. Ambiente de Teste

Os experimentos foram realizados em uma sala de 2.8×3.4 metros, com piso e paredes cerâmico, e ausência de incidência de luz solar, garantindo controle total sobre a iluminação. O drone foi posicionado no centro da sala, iniciando sempre a 2,0 m de altura. O sujeito do teste foi posicionado a distâncias fixas de 1 m, 2 m e 3 m, medidas com trena.

Os equipamentos utilizados nos testes foram: Drone DJI Tello com uma câmera embarcada de 720p, 30 fps. Os algoritmos são executados em um notebook externo, sendo ele um Acer Nitro 5 AN515-54, 24 GB RAM. Tecnologias usadas para rodar o projeto foram Python 3.10, OpenCV 4.8.0 e `djitellopy` 2.4. Ambiente com iluminação controlada por luz de led regulável apenas.

A iluminação foi medida por meio da média de intensidade da região facial (0–255 pixels) utilizando a função *mean* da biblioteca *Numpy*, que é nativa do *Python*. Foram

definidas três faixas de iluminação, baseadas nas possibilidades de ajuste de ambiente: Baixa: valores menores do que 20; Média: valores entre 40–120; Alta: valores maiores do que 120;

3.3.2. Protocolo experimental

- O drone foi elevado automaticamente a 2,0 m de altura.
- Para os testes de variações de configurações, a iluminação foi mantida em uma faixa de valores entre 60 e 80 na região do rosto detectado, valores que possibilitaram clara visão do rosto presente.
- Para testes de robustez do algoritmo, a iluminação foi ajustada de acordo com o necessário para cada teste.
- Cada teste consistiu em 200 frames consecutivos, gravados diretamente do stream do drone.
- Apenas uma pessoa permaneceu em cena.
- Para cada combinação de parâmetros do Haar Cascade e do LBPH, o teste foi repetido duas vezes em cada distância.
- Os falsos positivos, falsos negativos e valores de confiança foram registrados e analisados estatisticamente.

3.3.3. Justificativa das escolhas metodológicas

A escolha do conjunto metodológico adotado neste trabalho segue critérios técnicos relacionados à capacidade operacional do drone Tello, às características dos algoritmos utilizados e aos objetivos de avaliação da pesquisa. Os valores de distância (1 m, 2 m e 3 m), quantidade de repetições (2 testes por distância) e o número de amostras analisadas (200 frames por teste) foram definidos com base nos seguintes fatores:

1. Limitação de alcance do HaarCascade

Testes preliminares demonstraram que o HaarCascade utilizando a câmera do Tello, com resolução de 720p, tem dificuldade de reconhecer rostos, mesmo em condições ideais de iluminação, a partir de distâncias maiores do que 3 metros. Assim, 3 m representa o limite máximo no qual ainda é possível realizar detecção e reconhecimento facial com alguma estabilidade. Justificando, assim, sua inclusão como limite superior.

2. Variáveis e Intervalos Testados — Haar Cascade

O Haar Cascade possui dois parâmetros principais que influenciam diretamente o comportamento do detector: `scaleFactor` e `minNeighbors`. Os valores selecionados para variação do parâmetro `scaleFactor` foram 1.1, 1.2 e 1.3, enquanto para o parâmetro `minNeighbors` foram utilizados todos os números inteiros entre 1 e 5. Essa faixa segue as recomendações presentes na documentação do OpenCV, que indica que esses intervalos correspondem à zona de maior influência sobre o comportamento do detector. Dessa forma, evitam-se tanto valores que produziriam alterações pouco significativas quanto configurações que poderiam tornar o algoritmo inviável do ponto de vista computacional [OpenCV 2024].

3. Justificativa das Variáveis e Intervalos Testados — LBPH

Para esta avaliação, optou-se por variar apenas um intervalo restrito de valores nas configurações de `grid_x` e `grid_y`, que neste estudo foram definidos entre 7 e 9 enquanto as outras variáveis presentes se mantiveram nos valores padrões que também são descritas na documentação da *OpenCV*. Essa escolha se baseia em testes prévios, nos quais intervalos mais amplos apresentaram desempenho significativamente inferior. O LBPH funciona dividindo a imagem do rosto em uma grade de regiões (`grid_x` $\times$ `grid_y`) e calculando padrões binários locais em cada uma dessas regiões para gerar o histograma que representa o rosto. Quando o número de recortes é muito alto (valores acima de 9), pequenas variações na iluminação, sombras ou expressão facial podem afetar fortemente cada região, tornando o reconhecimento instável ou impedindo a detecção do rosto. Por outro lado, valores muito baixos para `grid_x` e `grid_y` (abaixo de 7) resultam em regiões grandes e pouco detalhadas, o que reduz a capacidade do algoritmo de distinguir características finas e aumentando o número de falsos positivos. Esses valores podem variar de acordo com a resolução da imagem fornecida. Levando em consideração a câmera do drone que fornece imagens a 720p, esse intervalo de valores se mostra suficiente.

4. Representação de distâncias reais de operação

Em cenários reais de vigilância, a pessoa monitorada costuma estar entre 1 m e 3 m do dispositivo, especialmente considerando ambientes internos. As distâncias escolhidas refletem situações práticas de uso e fornecem uma avaliação progressiva da degradação do desempenho conforme o alvo se afasta.

5. Consistência estatística mínima

Cada teste utiliza 200 frames, pois essa quantidade é suficiente para suavizar a mudanças instantâneas de detecção e representar o comportamento real do algoritmo sem exigir amostras excessivamente grandes. As 2 repetições por condição são necessárias para reduzir o impacto de variações aleatórias, resultando em média e desvio padrão confiáveis.

6. Controle de variáveis externas

A escolha de utilizar apenas uma pessoa no ambiente visa garantir que qualquer detecção adicional seja necessariamente um falso positivo, permitindo mensurar esse tipo de erro de forma objetiva.

3.3.4. Metodologia de testes para ajuste do Haar Cascade (Detecção facial)

Já que o Haar Cascade é utilizado também na fase de treinamento do LBPH, é essencial que, antes de iniciar esse treinamento, fossem identificadas as melhores configurações para o detector.

Sendo assim, usando o ambiente de teste já apresentado, iniciou-se a etapa de calibração dos parâmetros responsáveis pela detecção facial. Os testes avaliaram as combinações de `scaleFactor` (1.1, 1.2, 1.3) e `minNeighbors` (1,2,3,4,5) nas distâncias de 1 m, 2 m e 3 m. Para as demais variáveis de configuração do algoritmo, os valores foram mantidos nos padrões que são descritos na documentação da *OpenCV*. Para cada combinação foram realizadas 2 repetições independentes, registrando-se 200 frames por repetição.

Espera-se que exista ao menos uma combinação dos parâmetros `scaleFactor` e `minNeighbors` que apresente desempenho superior às demais nas métricas de detecção facial, especialmente em termos menor ocorrência de falsos positivos. Assim, por meio das baterias de testes realizadas nas diferentes distâncias, acredita-se que seja possível identificar uma configuração do Haar Cascade que otimiza o equilíbrio entre sensibilidade e precisão, resultando em uma detecção mais eficiente nas condições experimentais propostas.

3.3.5. Metodologia de teste para o ajustes do LBPH (Reconhecimento facial)

Para efetuar os testes referentes ao LBPH, assim como no HaarCascade, apenas os parâmetros de maior influencia foram alterados, no caso o `grid_x` e `grid_y`. Os demais parâmetros permaneceram com seus valores padrões que são descritos na documentação da OpenCv.

Para o trianmento do algoritmo foram usados dois *datasets*, para duas pessoas diferentes. Cada um deles era composto por 40 imagens em distâncias diferentes e com leves variações no angulo do rosto.

Para definir um intervalo válido de teste, considerou-se também o valor de confiança retornado pelo LBPH na previsão de um rosto. Nesse algoritmo, quanto menor o valor de confiança, maior a certeza do reconhecimento. Entretanto, valores de confiança entre 30 e 60 foram adotados como referência, pois indicam que o treinamento contém detalhes suficientes para o reconhecimento sem ser excessivamente sensível a pequenas variações. Valores abaixo de 30 podem indicar que o modelo foi pouco detalhado, resultando em maior risco de falsos positivos, enquanto valores maiores do que 60 sugerem que a confiança do algoritmo é mais baixa, tornando-o mais suscetível a pequenas alterações no rosto, iluminação ou sombras, podendo reduzir a precisão do reconhecimento.

Para a execução dos testes, o algoritmo foi treinado utilizando cada um dos conjuntos de combinações possíveis. Para cada configuração, realizou-se um novo treinamento sempre com os mesmos datasets, e posteriormente os resultados foram avaliados usando a câmera do drone em tempo de execução. Assim como no procedimento adotado com o HaarCascade, foram realizadas duas baterias de testes de 200 frames para cada configuração, nas distâncias de 1 m, 2 m e 3 m.

Da mesma forma, espera-se que exista ao menos uma combinação dos parâmetros `grid_x` e `grid_y` capaz de melhorar significativamente o desempenho do LBPH em termos de taxa de reconhecimento facial e redução de falsos negativos, constando um valor de confiança médio entre 30 e 60. Acredita-se que grades de granularidade intermediária ofereçam o melhor equilíbrio entre detalhamento da textura facial e estabilidade do modelo, permitindo identificar, por meio das baterias de testes, a configuração mais adequada para operar com qualidade dentro das condições impostas pelo hardware disponível.

3.3.6. Metodologia dos testes de iluminação

Tantos os teste efetuados para o HaarCascade quanto para o LBPH foram feitos em cenários ideais de iluminação, ou seja, com pouco ou nenhuma variação de iluminação,

sendo assim, para validar a robustez e o comportamento dos algoritmos, foi efetuada uma bateria de testes seguindo os moldes das já efetuadas anteriormente. Essa escolha foi baseada no fato de que tanto o Haar Cascade quanto o LBPH são algoritmos baseados em intensidade de pixels, sendo assim, altamente sensíveis a alterações de luz. Mudanças na iluminação modificam diretamente o contraste e os padrões locais utilizados para detecção e reconhecimento, podendo gerar distorções nos descritores e comprometer o desempenho do sistema. Além disso, a iluminação é a variável mais difícil de controlar em cenários reais, apresentando grandes variações entre ambientes e ao longo do tempo, o que a torna o principal fator de risco para perda de desempenho.

Com as configurações definidas nos testes anteriores, o algoritmo foi avaliado sob diferentes condições de luminosidade, com o objetivo de medir sua robustez. A classificação desses níveis de luz foi estabelecida com base na média de intensidade dos pixels da imagem na área do rosto, ignorando os valores gerais de iluminação, variando de 0 a 255 pontos, onde valores próximos de 0 representam baixa luminosidade e valores próximos de 255 indicam alta luminosidade.

Levando em consideração essa classificação de luminosidade, os testes foram realizados em uma faixa que varia de 10 a 120. O valor mais baixo corresponde ao ponto em que ainda é possível identificar um rosto presente no ambiente, enquanto o valor mais alto decorre da limitação de luminosidade que pôde ser aplicada ao rosto durante os testes, já que, mesmo com iluminação direta, não foi possível atingir valores constantes superiores a 120.

Espera-se que condições de iluminação insuficiente ou excessiva reduzam a qualidade da imagem capturada pelo drone, prejudicando a definição dos traços faciais e limitando a distância máxima em que o rosto pode ser detectado com consistência. Assim, pressupõe-se que exista uma faixa de iluminação ideal que permita ao algoritmo manter desempenho estável, enquanto desvios significativos dessa faixa diminuem a precisão e o alcance da detecção.

3.3.7. Metodologia de testes para a movimentação do drone

Parte importante de grande relevância desse projeto é o processo de acompanhamento (*tracking*) dos indivíduos detectados. Para isso, foi definida uma métrica específica relacionada à velocidade de resposta do sistema, mensurada a partir do tempo que o drone leva para centralizar o rosto de uma pessoa após a sua detecção inicial no quadro.

A centralização é um aspecto fundamental de sistemas de tracking, pois indica a capacidade do equipamento de reagir rapidamente às mudanças na posição do alvo. Um sistema lento, mesmo que reconheça corretamente uma face, pode se mostrar ineficiente para tarefas de acompanhamento autônomo.

Para realizar essa medição, o código foi instrumentado para registrar automaticamente o instante em que o primeiro rosto é identificado no frame. A partir dessa detecção inicial, um cronômetro interno é acionado. Em seguida, o drone executa os ajustes necessários para reduzir a diferença entre o centro do rosto e o centro da imagem capturada. O processo é considerado concluído quando o rosto permanece dentro de uma região pré-definida, neste caso, quando o ponto central do rosto se alinha ao ponto central da imagem

com uma margem de erro de 20 pixels. Assim que essa condição de centralização é atendida, o cronômetro é interrompido e o tempo obtido é registrado como uma amostra de desempenho.

Para cada teste realizado, havia apenas um participante presente no ambiente, posicionado a 1 metro de distância do drone, com um deslocamento lateral de cerca de 40 cm em relação ao seu eixo frontal. Nessa configuração, embora o drone tivesse o rosto dentro de seu campo de visão periférico, era necessário executar um giro aproximado de 45° seguido de um movimento de aproximação de cerca de 400 cm para que o rosto fosse completamente centralizado e considerado validado como corretamente alinhado.

Foram realizadas cinco baterias de testes, cada uma composta por três execuções. Ao final do experimento, os resultados analisados correspondem à média do tempo necessário para realizar o mesmo movimento em cada bateria.

Não existe um tempo “correto” ou universal para que um drone execute a movimentação necessária para centralizar um rosto, pois esse tempo varia de acordo com fatores como o modelo do drone, sua aceleração segura, as condições do ambiente e os critérios adotados para definir quando o rosto está realmente centralizado. Cada combinação desses elementos altera diretamente a estabilidade do voo e a qualidade do rastreamento, tornando inviável definir um único valor ideal aplicável a todos os cenários. Sendo assim, espera-se que o drone seja capaz de realizar correções simples de posicionamento, como girar para centralizar o rosto ou ajustar sua distância até o alvo, dentro de um intervalo de tempo suficientemente curto para permitir o uso prático do sistema de detecção e acompanhamento. Assim, pressupõe-se que a velocidade de movimentação e a capacidade de resposta do drone sejam adequadas para executar pequenos ajustes sem comprometer a estabilidade do rastreamento ou a continuidade da identificação facial.

4. Resultados e discussões

Esta seção apresenta os resultados obtidos com a implementação geral do projeto e com os testes realizados a partir das metodologias descritas na seção anterior.

O projeto foi implementado com sucesso, permitindo a integração entre o drone e os algoritmos de detecção e reconhecimento facial. Com isso, o sistema é capaz de realizar o tracking básico de rostos e identificar as pessoas presentes no ambiente.

O comportamento do drone ocorre em duas etapas principais. Primeiro, ele ajusta automaticamente sua altura inicial para 2 metros, garantindo uma boa visibilidade das pessoas à frente e reduzindo o risco de colisões. Em seguida, ao detectar um rosto, o tracking é ativado e o drone passa a se movimentar de forma a seguir a pessoa identificada. No entanto, o algoritmo de movimentação apresenta limitações quando há mais de um rosto no quadro, já que não existe um critério robusto para decidir qual deles deve ser seguido.

Para o reconhecimento facial, o sistema adota a regra de confirmar uma pessoa após 15 frames consecutivos de reconhecimento positivo. Quando essa confirmação ocorre, o sistema retorna se aquela pessoa está autorizada ou não a permanecer no local, com base em uma lista pré-definida de nomes permitidos ou proibidos.

Nas subseções seguintes, são apresentados e detalhados os resultados dos testes realizados. Primeiramente, são analisados os dados relacionados aos algoritmos de detecção

e reconhecimento facial. Em seguida, são discutidos os resultados referentes ao comportamento do drone e ao desempenho do sistema de movimentação.

4.1. Resultados dos testes do HaarCascade (Detecção facial)

Com os testes realizados em torno do HaarCascade, foi possível obter uma série de resultados que permitiram a construção de tabelas contendo a média de falsos positivos para cada configuração e distância. A partir dessas médias, calculou-se também o desvio-padrão de cada linha, refletindo a variação entre os testes a 1 m, 2 m e 3 m para cada combinação de parâmetros. Esse desvio-padrão foi fundamental para avaliar a estabilidade de cada configuração: valores mais altos indicam grande variação de desempenho conforme a distância muda, enquanto valores mais baixos sugerem um comportamento mais uniforme e previsível.

Os resultados, apresentados na Tabela 1, mostraram que algumas configurações apresentaram valores reduzidos de falsos positivos e baixa variabilidade entre as distâncias, demonstrando bom desempenho geral. Por outro lado, algumas combinações exibiram quantidades elevadas de falsos positivos ou desvios-padrão altos, evidenciando instabilidade na detecção ou forte sensibilidade à mudança de distância. No conjunto, esses resultados permitiram identificar de forma mais clara quais configurações são mais adequadas e confiáveis para uso no sistema.

scaleFactor	minNeighbors	1m	2m	3m	Desvio padrão
1.1	5	4.0	0	0	2.3
1.1	4	17.0	4.0	6.0	7
1.1	3	95.0	13.0	29.0	43.4
1.1	2	120.0	36.0	76.0	42.0
1.1	1	252.0	157.0	250.0	54.2
1.2	5	1.0	0.0	0.0	0.5
1.2	4	7.0	0.0	0.0	4.0
1.2	3	30.0	5.0	0.0	16.0
1.2	2	77.0	19.0	3.0	38.9
1.2	1	141.0	126.0	77.0	33.4
1.3	5	0.0	1.0	3.0	1.5
1.3	4	1	1	2	0.5
1.3	3	0	3	39	21.7
1.3	2	9	24	99	48.2
1.3	1	67	63	200	70.4

Tabela 1. Resultados de falsos positivos para combinações de scaleFactor e minNeighbors.

Levando em consideração os dados apresentados, é possível identificar um conjunto de configurações que se destaca por apresentar desempenho superior em comparação às demais. Entre as combinações analisadas, tanto scaleFactor = 1.2 / minNeighbors = 5 quanto scaleFactor = 1.3 / minNeighbors = 4 exibiram desvios-padrão bastante baixos, indicando comportamento estável nas diferentes distâncias testadas.

No entanto, a primeira configuração apresentou um número menor de falsos positivos, o que a torna mais precisa e consistente. Dessa forma, a combinação 1.2 / 5 se

consolida como a melhor opção entre todas as avaliadas no conjunto de experimentos realizados.

Entretanto, ao analisar os resultados, nota-se uma variação significativa na quantidade de falsos positivos entre as diferentes combinações de parâmetros. Essa oscilação não é inesperada: ela está diretamente ligada ao funcionamento interno do algoritmo Haar Cascade e à sensibilidade de seus parâmetros de detecção.

O Haar Cascade utiliza uma estratégia de varredura em múltiplas escalas, analisando a imagem diversas vezes em tamanhos distintos para identificar padrões compatíveis com o modelo treinado (neste caso, rostos humanos). Esse processo depende intensamente do ajuste fino dos parâmetros de operação. Como o algoritmo toma decisões por meio de uma cascata de classificadores fracos, que individualmente possuem baixa precisão, qualquer alteração nesses parâmetros afeta profundamente o comportamento das janelas de detecção e, consequentemente, a estabilidade do resultado final.

Além disso, variações de iluminação, como demonstrado posteriormente nos testes de robustez do sistema completo, impactam negativamente o desempenho do Haar Cascade. O método é altamente sensível a mudanças de brilho e contraste, o que reduz a consistência das detecções e aumenta a ocorrência de falsos positivos ou perdas de detecção. Esses fatores evidenciam que, apesar das vantagens relacionadas ao baixo custo computacional, o Haar Cascade apresenta limitações importantes em ambientes reais, nos quais é difícil controlar a iluminação do cenário e a luz incidente sobre o rosto das pessoas.

4.2. Resultado dos testes do LBPH

Com os testes efetuados, foi possível construir duas tabelas que auxiliam na compreensão do comportamento dos parâmetros ao longo das execuções. A tabela 2 apresenta o número médio de falsos negativos obtido, considerando as duas baterias de testes realizadas para cada configuração e distância. De forma semelhante, a tabela 3 apresenta o número médio de confiança para cada distância e configuração.

Observa-se na tabela 2 que, em geral, os falsos negativos aumentam significativamente conforme a distância do sujeito à câmera aumenta. Por exemplo, para a configuração $\text{grid}_x = 7$ e $\text{grid}_y = 9$, o número de falsos negativos passa de 0 a 97, do ponto de 1m para 3m. Configurações com grids maiores, especialmente $\text{grid}_x = 8$ ou 9, tendem a apresentar maior variabilidade no número de falsos negativos, conforme evidenciado pelo desvio padrão mais elevado. Isso indica que, embora o algoritmo consiga reconhecer rostos em distâncias curtas com alta precisão, o desempenho degrada-se rapidamente com o aumento da distância. Alguns grids apresentam falsos negativos constantes em todas as distâncias, como o caso de $\text{grid}_x = 7$ e $\text{grid}_y = 7$, indicando estabilidade para rostos próximos à câmera.

O valor médio de confiança, como é apresentado na tabela 3, por sua vez, tende a aumentar com a distância até certo ponto, mas também apresenta grande variabilidade dependendo da configuração do grid. Por exemplo, para $\text{grid}_x = 8$ e $\text{grid}_y = 9$, a confiança média varia de 58,52 a 110,83. Grids menores tendem a apresentar menor desvio padrão, sugerindo que a confiança é mais consistente, enquanto grids maiores apresentam altos desvios padrão, indicando que o algoritmo pode gerar resultados pouco previsíveis em diferentes distâncias.

grid_x	grid_y	1m	2m	3m	Desvio Padrão
7	7	0	0	0	0,00
7	8	0	47	34	20,78
7	9	0	99	97	57,14
8	7	0	0	15	6,12
8	8	0	170	152	79,01
8	9	0	200	200	94,28
9	7	1	200	200	114,61
9	8	8	200	200	110,21
9	9	53	200	200	82,64

Tabela 2. Número de falsos negativos por distância

grid_x	grid_y	1m	2m	3m	Desvio Padrão
7	7	37.20	48.90	51.20	5,99
7	8	44.51	69.70	73.22	12,99
7	9	55.25	80.69	79.85	12,32
8	7	52.90	65.79	72.40	8,28
8	8	59.60	91.20	82.30	14,08
8	9	58.52	104.36	110.83	22,38
9	7	69.30	96.10	111.60	18,60
9	8	66.80	94.40	106.20	17,02
9	9	79.30	112.50	137.69	23,60

Tabela 3. Valor médio de confiança por distância

Os valores de falsos positivos não foram incluídos nas tabelas, pois em todos os testes realizados nenhum falso positivo foi registrado. Dessa forma, tais valores foram omitidos por não serem relevantes para a análise e para as conclusões apresentadas.

Levando em consideração os resultados apresentados em ambas as tabelas, foi possível identificar, como esperado, uma configuração que melhor atende à necessidade de reconhecer rostos, especialmente em relação ao menor número de falsos negativos. Os valores grid_x = 7 e grid_y = 7 foram a configuração que apresentou o melhor desempenho, tanto por não registrar nenhum falso negativo quanto por apresentar um valor médio de confiança dentro do intervalo esperado, entre 30 e 60. Além disso, o desvio padrão, em ambos os casos, foi o mais baixo em comparação com as demais configurações.

Entretanto, assim como observado nos testes anteriores com o HaarCascade, o LBPH, embora tenha apresentado bons resultados nos testes realizados, também sofre com o problema de que pequenas alterações nos parâmetros resultam em variações significativas tanto nos valores quanto na dispersão de falsos positivos em diferentes distâncias. Esse comportamento ocorre devido à forma como o LBPH realiza o reconhecimento facial, pois, assim como o HaarCascade, ele também se baseia em padrões visuais que podem ser facilmente confundidos em ambientes muito complexos.

4.3. Resultado dos testes de iluminação com o algoritmo

Os resultados obtidos na bateria de testes demonstram que, como esperado, o algoritmo apresenta limitações significativas quando exposto a condições extremas de iluminação

ou quando o rosto se encontra a grandes distâncias da câmera. Como apresentado na Tabela 4, o reconhecimento facial atinge até 3 metros, somente quando a iluminação incidente no rosto está em um nível moderado e adequado, entre 40 e 120 pontos de luz. Em ambientes com baixa luminosidade, especialmente abaixo de 30 pontos, a distância máxima de detecção reduz-se drasticamente para cerca de 1 metro. Essas restrições estão diretamente relacionadas ao funcionamento tanto do LBPH quanto do HaarCascade, já que ambos dependem de padrões visuais que são fortemente influenciados pela distância e pelas condições de iluminação.

Intervalo de Iluminação	Distância Máxima	Confiança Média
10 – 30	1 m	45.7
40 – 60	3 m	39.8
60 – 80	3 m	40.7
80 – 100	3 m	37.6
100 – 120	3 m	40.5

Tabela 4. Confiança média por intervalo de iluminação e distância máxima

Os valores de confiança média indicam que, embora haja uma redução na distância máxima de reconhecimento quando a iluminação incidente no rosto é mais baixa, o nível de confiança permanece estável e dentro de limites aceitáveis.

Embora os testes considerem especificamente a luminosidade incidente no rosto, eles também evidenciaram que a presença de sombras influencia de forma decisiva o processo de reconhecimento, muitas vezes impedindo que o algoritmo identifique corretamente a face. Dessa forma, mesmo em ambientes com iluminação geral elevada, diferenças acentuadas de luz entre o fundo e o rosto podem gerar sombras que comprometem a detecção. Nesses cenários, o valor médio de luminosidade do ambiente pode ser alto, mas a iluminação direta sobre o rosto permanece baixa, dificultando o reconhecimento. Esse efeito pode ser observado na imagem X: apesar de o ambiente apresentar um nível elevado de luminosidade (99,7), conforme indicado no canto superior direito, o rosto está parcialmente sombreado, o que impede sua identificação clara pelo algoritmo.

Isso também ocorre quando há sombras parciais sobre o rosto, pois elas interferem diretamente nos padrões visuais que o HaarCascade utiliza para identificar uma face. Assim, como pode ser observado na imagem Y, mesmo com o rosto parcialmente visível, o algoritmo não foi capaz de reconhecê-lo.

Levando em consideração os resultados, a distância máxima de reconhecimento pode ser interpretada sob duas perspectivas. Para o caso em que o drone precisa apenas identificar o rosto de uma pessoa já localizada, os 3 metros são suficientes para cumprir esse objetivo. Entretanto, em um cenário prático de busca, onde o drone precisa varrer uma área em busca de possíveis pessoas, um alcance de apenas 3 metros se mostra limitado e pouco eficiente.

Por mais que tanto o HaarCascade quanto o LBPH atendam a alguns requisitos, como baixo custo computacional, fácil implantação e capacidade de executar tarefas básicas de reconhecimento, se levarmos em conta a análise feita em relação a ambas as tecnologias até aqui, é possível concluir que esses métodos apresentam limitações severas quando aplicados a cenários dinâmicos e desafiadores.

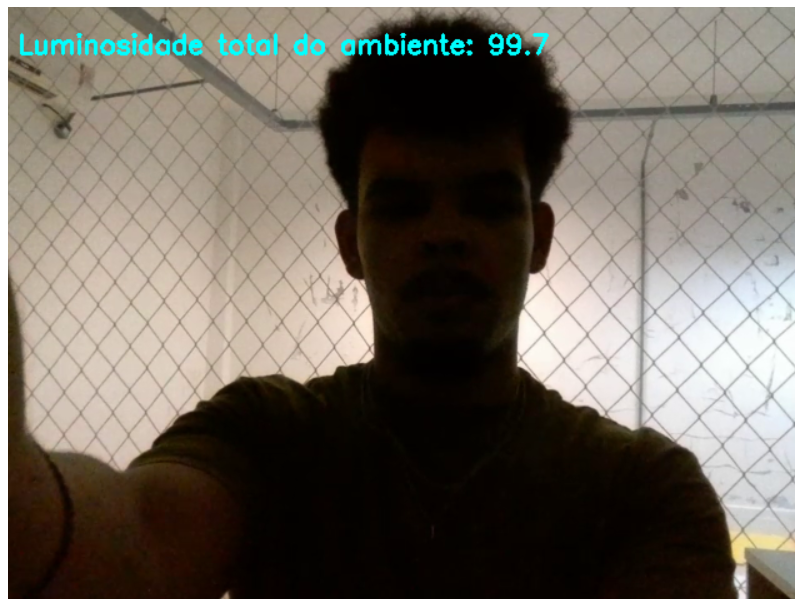


Figura 3. Efeito de sombras na qualidade do reconhecimento



Figura 4. Falha de detecção devido a sombras parciais

Em termos práticos, isso significa que, embora sejam algoritmos historicamente importantes, leves e simples de usar, eles não foram projetados para operar em ambientes dinâmicos. Seu desempenho cai em situações com alterações de iluminação, variação de distância, movimento constante ou presença de ruído visual, exatamente o conjunto de fatores mais comuns em sistemas de segurança dinâmicos e móveis, como é o caso desse projeto.

Dessa forma, a partir dos dados coletados e da análise experimental, fica evidente que HaarCascade e LBPH podem ser suficientes para aplicações estáticas, ambientes controlados ou prototipagem inicial, mas não representam uma solução robusta e confiável para sistemas de reconhecimento facial em tempo real em drones. Esses resultados jus-

tificam a necessidade de investigar técnicas mais modernas e resilientes, como métodos baseados em deep learning, detectores treinados com maior variedade de cenários e classificadores menos sensíveis a mudanças de iluminação e distância.

4.4. Resultados dos testes de movimentação do drone

O objetivo principal deste teste foi verificar se a movimentação do drone seria capaz de responder, em tempo hábil, aos comandos necessários para manter o rosto de uma pessoa centralizado na imagem. A partir da metodologia apresentada anteriormente, foi possível efetuar algumas análises de tempo médio que o drone leva para reconhecer um rosto e realizar os ajustes de posição necessários para centralizá-lo adequadamente na imagem.

Ao analisar os resultados obtidos na Tabela 5, observa-se que o tempo necessário para que o drone consiga centralizar o rosto é consideravelmente elevado, levando em consideração que o movimento que o drone efetua é relativamente simples, composto por uma rotação de 45° e um movimento de aproximação de 400 cm.

Bateria de teste	Tempo médio para centralização
1	9.683 s
2	11.45 s
3	10.465 s
5	12.678 s

Tabela 5. Tempo de centralização por teste

Esses resultados demonstram que o comportamento do drone, na implementação atual do projeto, não foi satisfatório. Mesmo considerando a menor média registrada, 9,683 segundos, o tempo de resposta ainda é excessivamente alto. Em cenários reais, uma movimentação tão lenta torna-se inviável, pois facilita que a pessoa monitorada se desloque e saia do campo de visão do drone, levando à perda do tracking e comprometendo o objetivo do sistema.

Esse comportamento pode ser explicado por algumas particularidades do funcionamento do drone. Uma das principais dificuldades no trabalho com esse dispositivo é a comunicação de comandos para ele; além de utilizar comunicação via UDP, que apesar de rápida apresentou perdas de pacotes constantes quando o drone recebia muitos comandos por segundo, existe um problema ainda mais relevante: a forma como o drone trata os comandos recebidos. De maneira geral, o drone recebe um comando de movimento, retorna apenas um status informando se o comando foi aceito ou não e, posteriormente, não informa se a execução do movimento foi concluída. Ou seja, não é possível identificar, a nível de código, quando um movimento terminou para que outro seja iniciado, o que se torna um grande impeditivo para obter um controle mais preciso e robusto do dispositivo. Essas limitações são coerentes com a proposta original do Tello, que foi projetado principalmente para fins educacionais e recreativos, e não para aplicações profissionais que exigem respostas contínuas, controle refinado e comunicação confiável para operações complexas.

5. Conclusão e trabalhos futuros

Ao concluir o projeto e avaliar os resultados, percebe-se que, embora a implementação tenha sido bem-sucedida, ainda existem limitações significativas relacionadas ao desem-

penho dos algoritmos utilizados para detecção e reconhecimento facial. Tecnologias baseadas em padrões visuais, como intensidade de pixels, mostram-se insuficientes para um cenário real, onde não é possível controlar a iluminação de forma tão rigorosa, evitar sombras ou garantir que as pessoas se movimentem lentamente o bastante para que o drone consiga realizar o tracking facial de maneira adequada.

A partir desse contexto, é possível concluir que o drone Tello, assim como os algoritmos pré-treinados utilizados neste trabalho, são adequados para projetos experimentais e protótipos, apresentando resultados preliminares satisfatórios. No entanto, ambos se mostram inadequados para aplicações reais, nas quais movimentação precisa e rápida, bem como alta confiabilidade no reconhecimento facial, são requisitos fundamentais.

Este estudo demonstra que é possível utilizar visão computacional integrada a drones para compor um sistema de segurança autônomo. No entanto, tecnologias mais avançadas tendem a oferecer resultados superiores, especialmente no que diz respeito à acurácia na detecção.

Com base nas limitações identificadas ao longo do trabalho, é possível mapear alguns direcionamentos para pesquisas futuras que visem uma implementação mais robusta. Entre eles, a adoção de um drone com maior capacidade de movimento e recursos específicos para vigilância, como os modelos da própria DJI. Um exemplo é o Matrice 4E Enterprise, projetado para aplicações de monitoramento e equipado com funcionalidades avançadas, como detecção inteligente, medição via telémetro a laser, lanternas para melhor visibilidade do ambiente e modos de voo inteligente mais completos, capazes de mapear áreas sobrevoadas e gerar mapas de cobertura de vigilância [DJI Enterprise 2025].

Além disso, estudos anteriores reforçam que o uso de técnicas mais avançadas de inteligência artificial pode elevar significativamente a precisão de sistemas embarcados em drones. [Canton et al. 2021] demonstra nos resultados do seu projeto que abordagens baseadas em redes neurais profundas, como CNNs, oferecem resultados muito superiores aos métodos tradicionais ao realizar detecção e reconhecimento facial em cenários reais, justamente por serem capazes de extrair características mais robustas e invariantes, mesmo em condições desfavoráveis de iluminação ou movimento. Nesse sentido, uma alternativa tecnológica altamente recomendada para futuras implementações é a utilização de modelos modernos de reconhecimento facial, como FaceNet ou ArcFace, por apresentarem acurácia elevada, baixa taxa de falsos positivos e excelente desempenho em ambientes dinâmicos. A integração desse tipo de tecnologia ao processamento a bordo de drones mais avançados permitiria um sistema autônomo consideravelmente mais confiável, eficiente e adequado para aplicações reais de monitoramento.

Referências

- Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. *European Conference on Computer Vision*.
- Balaji, S. and Priyanka, C. P. (2025). A comparative study of lightweight face detection models for real-time mobile applications. *International Journal of Engineering Development and Research (IJEDR)*, 13(2):135–154.
- Beri, N., Srivastava, V., and Malik, N. (2024). Face recognition attendance management system using lbph and haar cascade. *Journal of Trends in Computer Science and Smart*

Technology, 6(3):257–273.

- Bradski, G. and Kaehler, A. (2008). Learning opencv. *Dr. Dobb's Journal of Software Tools*.
- Canton, G. A. D., Padilha, R. J., and Yepes, I. (2021). Uso de drones na identificação facial baseado em machine learning e visão computacional. Trabalho de conclusão de curso, Instituto Federal Farroupilha, Frederico Westphalen, RS.
- DJI (2018). *Tello User Manual*. DJI.
- DJI Enterprise (2025). Dji matrice 4 series — the age of intelligent flight.
- Fuentes Escoté, D. (2018). Djitelloy: Python interface for the dji tello drone. GitHub repository. Disponível em: <https://github.com/damiafuentes/DJITelloPy>. Acesso em: Jun. 2025.
- Liu, H., Zhang, Q., and Sun, Y. (2025). Comparative study of lightweight object detection models for real-time applications on edge devices. *Sensors*, 25(3):1123–1140.
- Maan, S., Tyagi, A., Kumar, S., and Kumar, A. (2020). Security and surveillance using computer vision. *International Journal of Computer Applications*, 176(27):30–35.
- Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20.
- OpenCV (2024). *OpenCV Documentation*. Open Source Computer Vision Library. Acesso em: Jun. 2025.
- Singh, A., Patil, D., and Omkar, S. N. (2018). Eye in the sky: Real-time drone surveillance system (dss) for violent individuals identification using scatternet hybrid deep learning network. arXiv preprint. arXiv:1806.00746v1.