

**INSTITUTO TECNOLÓGICO DE AERONÁUTICA**



**Rafaella Carmo dos Santos**

**USO DE VISÃO COMPUTACIONAL NO  
MAPEAMENTO DE TAREFAS DE HANGARES DE  
MANUTENÇÃO AERONÁUTICA**

Trabalho de Graduação  
2024

**Curso de Engenharia Civil-Aeronáutica**

**Rafaella Carmo dos Santos**

**USO DE VISÃO COMPUTACIONAL NO  
MAPEAMENTO DE TAREFAS DE HANGARES DE  
MANUTENÇÃO AERONÁUTICA**

Orientador

Prof. Dr. Henrique Costa Marques (ITA)

**ENGENHARIA CIVIL-AERONÁUTICA**

**SÃO JOSÉ DOS CAMPOS  
INSTITUTO TECNOLÓGICO DE AERONÁUTICA**

**Dados Internacionais de Catalogação-na-Publicação (CIP)**  
**Divisão de Informação e Documentação**

dos Santos, Rafaella Carmo  
Uso de Visão Computacional no Mapeamento de Tarefas de Hangares de Manutenção  
Aeronáutica / Rafaella Carmo dos Santos.  
São José dos Campos, 2024.  
114f.

Trabalho de Graduação – Curso de Engenharia Civil-Aeronáutica– Instituto Tecnológico de  
Aeronáutica, 2024. Orientador: Prof. Dr. Henrique Costa Marques.

1. Visão Computacional. 2. Hangares de Manutenção. 3. Unity. 4. YOLO. 5. Simulação  
Computacional. I. Instituto Tecnológico de Aeronáutica. II. Título.

**REFERÊNCIA BIBLIOGRÁFICA**

DOS SANTOS, Rafaella Carmo. **Uso de Visão Computacional no Mapeamento de Tarefas de Hangares de Manutenção Aeronáutica**. 2024. 114f. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

**CESSÃO DE DIREITOS**

NOME DA AUTORA: Rafaella Carmo dos Santos

TÍTULO DO TRABALHO: Uso de Visão Computacional no Mapeamento de Tarefas de Hangares de Manutenção Aeronáutica.

TIPO DO TRABALHO/ANO: Trabalho de Conclusão de Curso (Graduação) / 2024

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias deste trabalho de graduação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. A autora reserva outros direitos de publicação e nenhuma parte deste trabalho de graduação pode ser reproduzida sem a autorização da autora.

---

Rafaella Carmo dos Santos  
Rua H8A, Ap. 136  
12.228-460 – São José dos Campos–SP

# USO DE VISÃO COMPUTACIONAL NO MAPEAMENTO DE TAREFAS DE HANGARES DE MANUTENÇÃO AERONÁUTICA

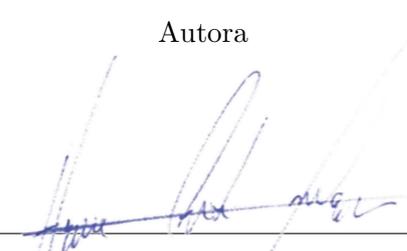
Essa publicação foi aceita como Relatório Final de Trabalho de Graduação



---

Rafaella Carmo dos Santos

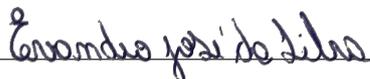
Autora



---

Henrique Costa Marques (ITA)

Orientador



---

Prof. Dr. Professor Evandro José da Silva  
Coordenador do Curso de Engenharia Civil-Aeronáutica

São José dos Campos, 21 de novembro de 2024.

Dedico este trabalho a Deus e a todas as pessoas que, ao longo do tempo, perderam a saúde ou a vida em atividades de manutenção, deixando como legado a importância da segurança, do cuidado e da valorização de cada trabalhador nesse ofício.

# Agradecimentos

Primeiramente agradeço a Deus: em sua infinita misericórdia, plantou sonhos em meu coração que eu acreditava serem impossíveis. E em sua infinita bondade me guiou ao longo da graduação e foi meu alicerce perante os desafios, se mostrando por meio de pessoas que iluminaram e continuam a iluminar os meus dias. “Àquele que é capaz de fazer infinitamente mais do que tudo o que pedimos ou pensamos, segundo o seu poder que atua em nós, a ele seja a glória na igreja em Cristo Jesus, por todas as gerações, pelos séculos dos séculos! Amém!” (Efésios 3:20-21).

Agradeço ao meu professor orientador, Dr. Henrique Costa Marques. Sua orientação, dedicação e confiança foram essenciais para a confecção deste trabalho. Cada orientação, como uma bússola, guiou-me com sabedoria e clareza frente às dúvidas que tive em alguns trechos da pesquisa. Foi, para mim, mais que um orientador, e sim um mentor e uma inspiração, que me trouxe uma vontade de cada vez mais avançar neste trabalho. Por fim, conduziu-me não apenas nos aspectos técnicos, mas em uma formação pessoal e profissional, ao me guiar nos caminhos certos, visando sempre uma pesquisa capaz de melhorar a vida das pessoas.

À Maria da Penha e Marcos Aurélio, por serem os melhores pais do mundo, pelo amor incondicional, pelo zelo e carinho que sempre demonstraram ao longo da minha criação. Tudo o que faço é por vocês.

Ao João Rodrigues (*in memoriam*) estrela que brilha em nosso céu. Vovô, o senhor me ensinou a construir antes da faculdade de engenharia civil e por isso serei eternamente grata.

À Dona Maria, vovó, que me concedeu o privilégio de vê-la dar os seus primeiros passos, três anos após o seu AVC, durante a minha graduação. Seu exemplo de perseverança irá sempre me inspirar a ultrapassar os limites impostos pelo corpo e pela mente.

Ao Thiago Tonaco, meu noivo, por todo o apoio, companheirismo e por demonstrar que é possível cada dia amar e ser amada mais pela mesma pessoa. Você foi parte fundamental da minha formação, a voz dizendo que tudo daria certo no final.

A todos os professores do JIFA, do Colégio Azevedo e do Colégio Militar do Rio de

Janeiro.

A todos os mestres com quem tive o privilégio de assistir aulas no ITA, em especial, pela amizade e pelo brilho nos olhos com que lecionavam, encantando a todos os alunos: Alessandro Vinícius, Cecília de Azevedo, Cláudia Azevedo, Cláudio Jorge, Eduardo Arraut, Eliseu Lucena, Evandro José, Francisco Alex, Giovanna Miceli, Henrique Costa, João Cláudio Bassan, José Schiavon, Lara Kuhl, Mayara Condé, Moacyr Machado, Natália Jodas, Paulo Ivo, Thiago Gomes e Tiara Martini. Devo minha formação profissional aos senhores.

Aos meus melhores amigos de infância, Rafael e Gabrielly.

À Samara Cavalcante e Alessandro Pimentel, por estarem presentes em todos os momentos em que a graduação foi difícil, nas noites sem dormir, nos estudos até o último segundo, nas pedaladas noturnas e, principalmente, pela amizade.

A todos os amigos que fiz no H8, em especial ao Pedro Batista, Mateus Silva, Felipe Marquesini e Fernando Victor, pela amizade que levarei para a vida.

Aos veteranos que me ensinaram, entre almoços e cafeterias, como ser iteana. E por serem a mão estendida prontos a ajudarem quando foi necessário. Em especial ao Thiago Tonaco, Marcos Vieira, Lucas Sousa, Pedro Secchi, Matheus Vidal e Amanda Trópia. “Se consegui ir mais longe, foi por estar apoiada nos ombros de gigantes.”

À Natália, Ingrid e Ana.

A toda a comunidade do H8, que me acolheu e fez destes os melhores, mais desafiantes e mais gratificantes anos da minha vida.

Agradeço ao Instituto Tecnológico de Aeronáutica e à Força Aérea Brasileira, por todas as oportunidades que recebi.

*“A imaginação muitas vezes nos  
leva a mundos que nunca sequer existiram.  
Mas sem ela não vamos a lugar algum.”*

— CARL SAGAN

# Resumo

A disponibilização de novas ferramentas relacionadas à visão computacional e detecção de objetos possibilita uma ampla gama de aplicações capazes de melhorar o mundo contemporâneo. Nesse contexto, o mapeamento de atividades em ambientes industriais e de defesa para a identificação de contexto de manutenção aeronáutica e a diminuição de riscos é objeto de estudo que, antes, seria considerado inviável.

Busca-se, nesse relatório, a integração de ferramentas de visão computacional para o mapeamento de ações e objetos que compõem os hangares de manutenção, com o objetivo de alcançar o reconhecimento computacional das etapas críticas (temporal e probabilisticamente com maior risco) para que, posteriormente, sejam indicadas arquiteturas tecnológicas capazes de suprir a criticidade das operações.

Dessa forma, esse trabalho fez a integração entre *OpenCV*, *YOLOv8* e uma simulação de um hangar de manutenção no *Unity* com subsídios para a implementação em um hangar real, possibilitando uma captura e interpretação nesse ambiente, assim como uma análise de viabilidade do mapeamento de tarefas.

# Abstract

The availability of new tools related to computer vision and object detection enables a wide range of applications capable of improving the contemporary world. In this context, the mapping of activities in industrial and defense environments, and the use of augmented reality as a facilitator of processes (reducing execution time, enabling the identification and reduction of errors) for the identification of contexts in aeronautical maintenance and risk mitigation, is a subject of study that, in the past, would have been considered unfeasible.

This report aims to integrate computer vision tools for mapping actions and objects within maintenance hangars, with the goal of achieving computational recognition of critical stages (both temporally and probabilistically, with higher risk) to subsequently propose technological architectures capable of addressing operational criticality.

Thus, this work integrates *OpenCV*, *YOLOv8*, and a simulation of a maintenance hangar in Unity, providing a basis for implementation in a real hangar. This integration enables data capture and interpretation in such an environment, as well as a feasibility analysis of task mapping.

# Lista de Figuras

FIGURA 2.1 – Exemplo de uma rede neural convolucional e suas diferentes camadas, retirada de (VARGAS <i>et al.</i> , 2023). . . . .	36
FIGURA 2.2 – R-CNN: regiões com características CNN. Imagem retirada de (GIRSHICK <i>et al.</i> , 2014) e traduzida. . . . .	36
FIGURA 2.3 – Linha do tempo das versões do YOLO, retirada de (TERVEN <i>et al.</i> , 2023). . . . .	38
FIGURA 2.4 – Arquitetura do YOLOv8, retirada de (CONTRIBUIDORES, 2023). . .	39
FIGURA 2.5 – Imagem da aeronave C-105, disponibilizada me site da Força Aérea Brasileira (FAB) . . . . .	44
FIGURA 2.6 – Modelo JDL de fusão de dados (versão de 1992), retirada de (STEINBERG <i>et al.</i> , 1999) . . . . .	47
FIGURA 2.7 – Fluxo lógico entre os níveis, retirada de (STEINBERG <i>et al.</i> , 1999) . .	47
FIGURA 2.8 – Proposta da arquitetura do sistema de detecção de quedas, retirado de (OSIGBESAN <i>et al.</i> , 2022) . . . . .	50
FIGURA 3.1 – Tutorial de acionamento da ferramenta <i>Unreal to Unity</i> , versão 1.41. . . . .	55
FIGURA 3.2 – Configurações de exportação da ferramenta <i>Unreal to Unity</i> , versão 1.41. . . . .	56
FIGURA 3.3 – Configuração de deslocamento de agentes no hangar . . . . .	58
FIGURA 3.4 – Configurações de padrões de cores para estudo de detecção de bordas . . . . .	60
FIGURA 3.5 – Passo para anotação de guindaste na plataforma <i>Roboflow</i> . . . . .	63
FIGURA 3.6 – Exibição das imagens anotadas, com as classificações de guindastes e empilhadeiras, na plataforma <i>Roboflow</i> . . . . .	63
FIGURA 3.7 – Exibição de maneira em que é feita a etiquetagem do guindaste na imagem, na plataforma <i>Roboflow</i> . . . . .	64

---

FIGURA 3.8 – Padrão de anotação de imagens na plataforma <i>RoboFlow</i> . . . . .	64
FIGURA 3.9 – Configuração de exportação de modelo .fbx do <i>Character Creator</i> . . . . .	66
FIGURA 3.10 – Configuração de avatar . . . . .	66
FIGURA 3.11 – Exemplificação de <i>Animator Controller</i> implementado com a função “pintar”. . . . .	67
FIGURA 4.1 – Visão do hangar obtida pelo <i>Unreal</i> . . . . .	69
FIGURA 4.2 – Visão do hangar obtida pelo <i>Unreal</i> , imagem capturada durante simulação do hangar . . . . .	69
FIGURA 4.3 – Visão do hangar obtida pelo <i>Unity</i> , configurações do <i>Unity</i> conforme descrito na seção 3.2.1.1. . . . .	70
FIGURA 4.4 – Visão geral do cenário no <i>Unity</i> , configurações conforme descrito na seção 3.2.1.1. . . . .	70
FIGURA 4.5 – Resultado de aplicação de <i>NavMeshSurface</i> na superfície do hangar, em azul. . . . .	71
FIGURA 4.6 – Agente usado como parâmetro na simulação. . . . .	72
FIGURA 4.7 – <i>Unity</i> no modo <i>Play</i> , com cada agente (pessoas em proporção) espalhadas e simulando suas devidas tarefas. . . . .	72
FIGURA 4.8 – Configuração inicial da primeira câmera. . . . .	73
FIGURA 4.9 – Configuração inicial da segunda câmera. . . . .	73
FIGURA 4.10 – Configuração inicial da terceira câmera . . . . .	74
FIGURA 4.11 – Aplicação do algoritmo identificador de bordas inicial, sem tratamento de imagem. . . . .	75
FIGURA 4.12 – Refinamento do algoritmo detector de bordas, mas ainda sem o tratamento da imagem. . . . .	75
FIGURA 4.13 – Uso do algoritmo detector de bordas atualizado e resultado obtido após processamentos de imagem, exibindo-se o resultado mais satisfatório. . . . .	76
FIGURA 4.14 – Foram feitos novos refinamentos, notando-se o aparecimento de novas bordas que são consideradas importantes para o modelo. A imagem exibe o resultado final. . . . .	76
FIGURA 4.15 – Vista de como é feita a exibição da imagem processada. . . . .	77
FIGURA 4.16 – Detecção da aeronave do hangar em vôo. . . . .	78

---

FIGURA 4.17 – Demonstração de detecção de aeronave, em tempo real, no hangar de manutenção. . . . .	78
FIGURA 4.18 – Detecção de operador (classificado como pessoa) e aeronave em hangar de manutenção. . . . .	79
FIGURA 4.19 – O nível de acurácia da detecção foi variável conforme luminosidade.	80
FIGURA 4.20 – Representação esquemática do funcionamento do algoritmo <i>YOLO</i> no <i>Unity</i> com o <i>OpenCV</i> . . . . .	80
FIGURA 4.21 – Representação do funcionamento do algoritmo <i>YOLO</i> no <i>Unity</i> com o <i>OpenCV</i> . . . . .	82
FIGURA 4.22 – Representação do funcionamento do algoritmo <i>YOLO</i> no <i>Unity</i> com o <i>OpenCV</i> . . . . .	83
FIGURA 4.23 – Representação do funcionamento do algoritmo <i>YOLO</i> no <i>Unity</i> com o <i>OpenCV</i> . . . . .	84
FIGURA 4.24 – Representação do funcionamento do algoritmo <i>YOLO</i> no <i>Unity</i> com o <i>OpenCV</i> . . . . .	85
FIGURA 4.25 – Representação de etiquetagem do cenário descrito no <i>Roboflow</i> . . . . .	87
FIGURA 4.26 – Matriz de confusão do modelo treinado no <i>Google Colab</i> . . . . .	88
FIGURA 4.27 – Perda relacionada à localização das caixas delimitadoras durante o treinamento no <i>Google Colab</i> . . . . .	88
FIGURA 4.28 – Perda de classificação durante o treinamento no <i>Google Colab</i> . . . . .	89
FIGURA 4.29 – Perda Focal Distribuída durante o treinamento no <i>Google Colab</i> . . . . .	89
FIGURA 4.30 – Precisão do modelo durante o treinamento no <i>Google Colab</i> . . . . .	90
FIGURA 4.31 – Capacidade do modelo de identificar corretamente todos os objetos durante treinamento no <i>Google Colab</i> . . . . .	90
FIGURA 4.32 – Perda relacionada à localização das caixas delimitadoras durante a validação do treinamento no <i>Google Colab</i> . . . . .	91
FIGURA 4.33 – Perda de classificação no conjunto de validação durante treinamento no <i>Google Colab</i> . . . . .	91
FIGURA 4.34 – <i>Distributed Focal Loss</i> (DFL) durante validação no <i>Google Colab</i> . . . . .	92
FIGURA 4.35 – A métrica mean Average Precision (mAP) com IoU (Intersection over Union) de 50%. . . . .	92
FIGURA 4.36 – Média das precisões calculadas em vários limiares de IoU, de 50% a 95%. . . . .	93

---

FIGURA 4.37 – Síntese dos gráficos com métricas de treinamento feito no <i>Google Colab</i> . . . . .	93
FIGURA 4.38 – Teste de modelo executado no <i>Google Colab</i> . . . . .	94
FIGURA 4.39 – Resultado de treinamento no <i>Google Colab</i> . . . . .	95
FIGURA 4.40 – Resultado de treinamento no <i>Google Colab</i> . . . . .	96
FIGURA 4.41 – Resultado de treinamento no <i>Google Colab</i> . . . . .	97
FIGURA 4.42 – Validação de detecção no <i>Roboflow</i> . . . . .	98
FIGURA 4.43 – Validação de detecção no <i>Roboflow</i> . . . . .	99
FIGURA 4.44 – Resultado de treinamento no <i>Google Colab</i> , usando o YOLOv8. . . . .	100
FIGURA 4.45 – Resultado de treinamento no <i>Google Colab</i> , usando o YOLOv8. . . . .	100
FIGURA 4.46 – Resultado de treinamento no <i>Google Colab</i> , usando o YOLOv8. . . . .	101
FIGURA 4.47 – Resultado de treinamento no <i>Google Colab</i> , usando o YOLOv8. . . . .	101
FIGURA 4.48 – Resultado de treinamento no <i>Google Colab</i> , usando o YOLOv8. . . . .	102
FIGURA 4.49 – Matriz de confusão do modelo treinado no <i>Roboflow</i> (dados de teste e validação). . . . .	104
FIGURA 4.50 – Análise vetorial do modelo treinado no <i>Roboflow</i> (dados de teste e validação). . . . .	104
FIGURA 4.51 – Gráfico mAP do modelo treinado no <i>Roboflow</i> (dados de teste e validação). . . . .	105
FIGURA 4.52 – Gráficos do modelo treinado no <i>Roboflow</i> (dados de teste e validação). . . . .	105
FIGURA 4.53 – Execução do modelo treinado no <i>Unity</i> . . . . .	106
FIGURA 4.54 – Parâmetros que regem a movimentação do carro de ferramentas. . . . .	106

# Lista de Tabelas

TABELA 2.1 – Comparação entre as principais tecnologias de câmeras de profundidade de (E-CON SYSTEMS, 2024) . . . . .	42
TABELA 2.2 – Comparação entre estudos antecedentes quanto aos assuntos abordados . . . . .	49
TABELA 3.1 – Comparação dos Modelos YOLO por Velocidade e Acurácia . . . . .	61
TABELA 4.1 – Resultados de validação por classe . . . . .	102

# Lista de Abreviaturas e Siglas

CNN	Rede Neural Convolutacional
COCO	<i>Common Objects in Context</i>
DFL	<i>Distributed Focal Loss</i> (Perda Focal Distribuída)
GBR	<i>Green</i> (verde), <i>Blue</i> (azul) e <i>Red</i> (vermelho), nessa ordem.
HD	<i>High Definition</i>
HDRP	<i>High Definition Render Pipeline</i>
mAP	<i>Mean Average Precision</i>
IA	<i>Inteligência Artificial</i>
IoU	<i>Intersection over Union</i>
OpenCV	<i>Open Source Computer Vision</i>
R-CNN	Redes Neurais Convolutacionais Baseadas em Regiões
RGB	<i>Red</i> (vermelho), <i>Green</i> (verde) e <i>Blue</i> (azul), nessa ordem.
SDD	<i>Single Shot Multibox Detection</i>
URP	<i>Universal Render Pipeline</i>
YOLO	<i>You Only Look Once</i>

# Sumário

1	INTRODUÇÃO . . . . .	19
1.1	Contextualização . . . . .	19
1.2	Motivação . . . . .	21
1.3	Problema de pesquisa . . . . .	22
1.4	Objetivo geral do trabalho . . . . .	24
1.4.1	Objetivos específicos do trabalho . . . . .	24
1.5	Relevância da pesquisa . . . . .	24
1.6	Delimitação da pesquisa . . . . .	25
1.7	Organização do trabalho . . . . .	25
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA . . . . .	26
2.1	Fundamentação Teórica . . . . .	26
2.1.1	Hangares de manutenção aeronáutica . . . . .	26
2.1.2	Fatores humanos e a manutenção aeronáutica . . . . .	28
2.1.3	Indústria 4.0 e Operador 4.0 . . . . .	32
2.1.4	Visão computacional . . . . .	34
2.1.5	Unity . . . . .	40
2.1.6	Unreal Engine . . . . .	41
2.1.7	Câmeras de profundidade . . . . .	41
2.1.8	Manuais de manutenção . . . . .	43
2.1.9	Riscos críticos em hangares de manutenção . . . . .	44
2.1.10	<i>Cloud Data Fusion</i> . . . . .	45
2.2	Revisão da literatura . . . . .	48

---

<b>2.3</b>	<b>Considerações finais do capítulo</b> . . . . .	51
<b>3</b>	<b>MATERIAIS E MÉTODOS</b> . . . . .	52
<b>3.1</b>	<b>Materiais</b> . . . . .	52
3.1.1	Requisitos técnicos . . . . .	52
3.1.2	Aparato tecnológico . . . . .	53
<b>3.2</b>	<b>Metodologia</b> . . . . .	54
3.2.1	Simulação do ambiente . . . . .	54
3.2.2	Agentes capazes de se deslocarem na simulação . . . . .	57
3.2.3	Posicionamento estratégico das câmeras . . . . .	58
3.2.4	Uso do <i>OpenCV</i> no <i>Unity</i> . . . . .	59
3.2.5	Integração do <i>Unity</i> com o <i>YOLO</i> e o <i>OpenCV</i> . . . . .	60
3.2.6	Treinamento de imagens para detecção de objetos típicos de hangares de manutenção inicialmente não reconhecidos . . . . .	61
3.2.7	Animação de agentes conforme rotinas pré-estabelecidas . . . . .	65
<b>4</b>	<b>APRESENTAÇÃO DOS RESULTADOS E DISCUSSÃO</b> . . . . .	68
<b>4.1</b>	<b>Ambiente concebido</b> . . . . .	68
4.1.1	Integração de agentes autônomos com o cenário . . . . .	71
<b>4.2</b>	<b>Imageamento</b> . . . . .	73
<b>4.3</b>	<b>Aplicação do <i>OpenCV</i> no <i>Unity</i></b> . . . . .	74
<b>4.4</b>	<b>Integração entre <i>OpenCV</i>, <i>YOLO</i> e <i>Unity</i></b> . . . . .	77
<b>4.5</b>	<b>Treinamento de imagens usando o <i>Roboflow</i> e o <i>Google Colab</i></b> . . . . .	81
<b>4.6</b>	<b>Estudo de caso</b> . . . . .	85
4.6.1	Limitações de Processamento em Tempo Real . . . . .	107
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	108
<b>5.1</b>	<b>Contribuição Acadêmica</b> . . . . .	108
<b>5.2</b>	<b>Aplicação na indústria</b> . . . . .	109
<b>5.3</b>	<b>Praticabilidade Operacional</b> . . . . .	109
<b>5.4</b>	<b>Sugestão de trabalhos futuros</b> . . . . .	110

---

REFERÊNCIAS . . . . . 111

# 1 Introdução

O aumento progressivo da operacionalidade aérea, impulsionado pelas demandas da contemporaneidade, gerou uma necessidade cada vez mais premente de otimização dos processos relacionados ao setor. Esses ambientes, caracterizados por sua complexidade operacional e alto nível de exigência técnica, enfrentam desafios que incluem a organização eficiente dos fluxos de trabalho, a identificação de gargalos operacionais e a incorporação de tecnologias inovadoras para aprimorar a eficiência e a segurança dos processos.

## 1.1 Contextualização

A manutenção de aeronaves, segundo a Associação Internacional de Transporte Aéreo, é classificada como atividade de prioridade secundária durante a fase de planejamento aeroportuário, sendo superada apenas pelas áreas operacionais da aviação (relativas ao pouso e decolagem). Para se definir a escala da operacionalidade das instalações de manutenção, alguns fatores são de especial relevância (ASSOCIATION, 1995):

- Os tipos de serviços existentes nas instalações dos hangares (se serão todos os serviços ou manutenções específicas);
- Quantidade de operadores em atividade e se possuem acesso às peças de reposição;
- Se a manutenção das aeronaves no hangar é feita pelas companhias em toda a frota ou se uma parcela é feita em sua base de origem;
- Quantidade e tipo de aeronaves em operação, sua proporção em relação à baía de manutenção, taxa de utilização e manutenção realizada (A, B, C ou D); e
- Nível de terceirização dos serviços.

Esses ambientes de manutenção são compostos majoritariamente pelas cinco seguintes áreas funcionais (RANTANEN *et al.*, 2003):

1. Área principal do hangar (estacionamento e onde é feita a maioria das manutenções das aeronaves)
2. Área destinada às ferramentas e aos equipamentos
3. Áreas de atividades administrativas e escritórios
4. Local de armazenamento de peças (peças sobressalentes e materiais necessários para a manutenção)
5. Áreas de sistemas contra-incêndio (inclui regiões de alarmes e extintores)
6. Área de infraestrutura (sistemas de ventilação, fornecimento de energia, dentre outros sistemas necessários para suportar as atividades diárias)

O decreto n.º 7.168, de 5 de maio de 2010, define as áreas de manutenção como “espaços e instalações destinados à manutenção de aeronaves, incluindo pátios, hangares, edificações e oficinas, estacionamentos de veículos e vias de acessos adjacentes”.

Ao se pensar no *design* dos hangares de manutenção, uma expressão adquire elevada relevância: *healthy building*, que consiste na conceituação de uma área (construção) capaz suportar a saúde física, psicológica, mental e o bem-estar dos seus habitantes e do contexto ao seu entorno.

As *healthy buildings* possuem elevada atenção aos seguintes aspectos (ASSOCIATION, 1995): qualidade do ar, conforto térmico, conforto visual, conforto acústico, qualidade da água, saúde mental, acesso à alimentação, limpeza e ergonomia. Tais aspectos se tornam fundamentais ao se pensar nas atividades e nível de exigência existente em tais ambientes de manutenção: são tomadas ações em aeronaves para permitir a continuação de suas aeronavegabilidades. Falhas dessas ações, por sua vez, podem causar perdas financeiras, humana e ambientais, podendo determinar o sucesso ou o fracasso de uma operação.

Essas atividades são exercidas pelo mecânico de manutenção aeronáutica, o profissional responsável pelo ato ou supervisão de tarefas de manutenção de aeronaves.

No âmbito da defesa, o especialista em mecânica é o profissional encarregado de realizar manutenção preventiva e corretiva das aeronaves militares. Sua responsabilidade inclui as fases de operação e testes dos seguintes sistemas: motor, hélice, comandos de voo, hidráulico, pneumático, combustível, oxigênio, pressurização, ar condicionado, extinção de incêndio dos motores e afins.

Pelas características e natureza do ambiente descrito, esses especialistas estão sujeitos aos acidentes em ambientes de trabalho, tais como: quedas, lesões musculares e esqueléticas, queimaduras e inalação de substâncias químicas, exposição aos ruídos, contusões, lacerações, choques elétricos e colisão com objetos em movimento.

Os acidentes descritos e os incidentes de manutenção relacionam-se com os estresse gerado pelas atividades de manutenção: as tarefas são complexas, estão submetidas a uma pressão de tempo e as condições ambientais por vezes não são favoráveis. Esses fatores se enquadram em erros associados a fatores humanos que são potenciais causadores de consequências econômicas significativas (OLAGANATHAN, 2024): atrasos e cancelamento de voos são alguns exemplos na aviação civil. Na aviação militar, pode-se citar a falha da missão.

Por esses motivos, algumas tendências se tornam presentes na indústria de manutenção aeronáutica: manutenção preventiva, realidade aumentada e virtual, análise de *Big Data*, manufatura aditiva, dentre outros (OLAGANATHAN, 2024). E isso tem se mostrado em análises estatísticas recentes. Conforme a Associação Internacional de Transporte Aéreo, tem-se observado a tendência à digitalização em aeroportos: observa-se uma maioria quantitativa das companhias aéreas investindo, principalmente, em aplicações móveis para serviços do piloto, inteligência de negócio e manutenção de aeronaves. Este último, por sua vez, apresentando um aumento de 86% ao nível de prioridade.

Portanto, o ambiente principal de investigação deste estudo é um hangar de manutenção, mais especificamente a área principal dos hangares e as destinada às ferramentas e aos equipamentos, focalizando as características que descrevem um hangar de manutenção na totalidade e as ações e tarefas executadas pelos técnicos e engenheiros. Centraliza-se a pesquisa na infraestrutura necessária para a implementação das tendências descritas no parágrafo anterior, na definição de *healthy buildings* e nos fatores humanos que interferem as atividades cotidianas.

## 1.2 Motivação

Considerando o contexto previamente exposto, alguns fatores foram preponderantes para a motivação deste trabalho. Tais motivações serão detalhadas nos parágrafos a seguir.

A necessidade latente de implementação nos hangares de manutenção das novas tendências citadas por (OLAGANATHAN, 2024) estabelecem requisitos infraestruturais fundamentais para que a idealização de um hangar 4.0 possa se tornar algo cada vez mais factível.

Dessa forma, pode-se definir uma infraestrutura básica para a execução das tecnologias relacionadas ao aumento da consciência situacional dentro dos hangares (realidade aumentada, realidade virtual, *Digital Twins*, dentre outras):

- Sensores e dispositivos de captura;
- Redes de comunicação de alta velocidade;

- Infraestrutura de dados;
- Segurança da rede e proteção de dados; e
- Dispositivos de visualização (*tablets*, celulares, óculos inteligente ou telas interativas).

Em um trabalho recente sobre a implementação de *Digital Twins* em hangares, o autor Luke Casey et al. destaca que a implementação dessa tecnologia oferece uma prova de avaliação de desempenho, mas que, para uma implementação prática, é essencial ocorrer um aperfeiçoamento tecnológico no contexto desses ambientes de manutenção no que tange ao rastreamento de pessoas, aeronaves e uma vasta classe de objetos FOD (Foreign Object Debris–Objeto estranho que pode causar dano a aeronave situado na zona de um aeródromo) (CASEY et al., 2024).

Segundo o autor, o desenvolvimento de tais técnicas possui o potencial de evitar colisões e possibilitar o planejamento de caminho nesses ambientes. Tais aplicações, alinhadas à melhora da ergonomia de técnicos de manutenção e diminuição de riscos para esses trabalhadores (por meio de um algoritmo tomador de decisão), são as principais motivadoras desse estudo.

A visão computacional é uma subárea da inteligência artificial e essa tecnologia permite uma análise automática de imagens e vídeos, visando extrair informações relevantes para a tomada de decisões baseadas em dados. Em hangares, permite a idealização de um monitoramento e análise das operações em tempo real, identificando riscos potenciais e gargalos, para fornecer informações valiosas para a tomada de decisão. Esse estudo é influenciado por uma necessidade urgente de melhorar a eficiência e a segurança das operações de manutenção, aumentando a confiabilidade das aeronaves e contribuindo para a redução de custos operacionais.

Por fim, este trabalho não pretende apenas contribuir para o avanço tecnológico no campo da aviação, oferecendo a infraestrutura necessária para o desenvolvimento de novas interfaces tecnológicas, mas também promover uma cultura de segurança e eficiência que cumpra as exigências em vigor da aviação moderna.

### 1.3 Problema de pesquisa

Hangares de manutenção são ambientes regidos pela existência de operações complexas que envolvem a execução de múltiplas tarefas simultaneamente. Hoje, a gestão dessas atividades é feita, na maioria das vezes, mediante sistemas tradicionais de monitoramento e métodos manuais, os quais são por vezes obsoletos e ineficientes. O crescente volume de

manutenções de aeronaves, associado ao crescimento da complexidade técnica requerida em tais manutenções (associado ao incremento de rigidez técnica que provém da melhora dos processos de segurança), e a disparidade entre o número de mecânicos atualmente em operação e as necessidades da indústria mostram que a gestão atualmente vigente colabora para o aumento de desafios relacionados à ineficiência operacional.

Esse problema é intensificado pelo aumento das operações aeronáuticas e sua crescente demanda de segurança e eficiência. Falhas em hangares de manutenção possuem consequências negativas para as companhias aéreas e para a área de defesa, que contemplam:

- Atrasos em voo
- Potenciais riscos aos passageiros e à tripulação
- Maior tempo de inatividade da aeronave
- Falha da missão

Além disso, a incapacidade de identificação e mitigação rápida dos riscos operacionais pode levar a falhas críticas decorrentes de erros mecânicos, ocasionando potenciais perdas: econômicas, humanas ou ambientais. Dessa forma, a perda de eficiência ocasionada pelo estresse dos operadores aeronáuticos e o nível de estresse ao qual estão submetidos colabora para a ocorrência de erros durante a manutenção que podem gerar incidentes nos hangares e acidentes durante o voo da aeronave: conforme o painel SIPAER, que trata das ocorrências aeronáuticas na Aviação Civil Brasileira, mais de 10% dos acidentes tem causa atribuída às falhas de manutenção. Nesse contexto, soluções tecnológicas vieram à tona: implementação de realidade aumentada, realidade virtual, dentre outras ferramentas de aumento da consciência situacional. Entretanto, tais soluções falham na implementação, devido à atual lacuna necessária para a captura de tarefas, leitura de cenário e rastreamento de objetos no contexto do hangar.

Para garantir a infraestrutura adequada à implementação das soluções tecnológicas descritas e diminuir o encargo laboral elevado sem, contudo, alterar a quantidade de funcionários no ambiente de trabalho, propõem-se, nesse trabalho, que é possível preencher tais lacunas por meio de uma ferramenta de captura de ações (do mecânico de manutenção e da aeronave) e cenários (hangar na totalidade), criando um algoritmo eficaz para análise de risco e tomada de decisão, com a posterior implementação de um agente autônomo capaz de se deslocar durante as atividades.

Até o presente momento da escrita, não foram propostas soluções capazes de resolver efetivamente o problema descrito. Foram encontradas, apenas, contribuições parciais no contexto de visão computacional em hangares para o mapeamento de atividades humanas,

como em (ROMERAL *et al.*, 2023). Este trabalho, dessa forma, pavimenta o suporte necessário para posteriores tentativas de implementação de soluções digitais nos ambientes de manutenção aeronáutica, fornecendo uma ferramenta treinada de identificação de tarefas e ações, além de possibilitar a implementação de agentes no cenário sem que isso acarrete perdas de eficiência ou aumento de risco.

## 1.4 Objetivo geral do trabalho

Visa-se, com esse trabalho, o preenchimento dos hiatos descritos no problema de pesquisa. Busca-se fazer um estudo sobre configuração de imageamento, rotinas e pontos críticos em um ambiente controlado. Dessa forma, o objetivo principal do trabalho é construir uma abordagem com visão computacional para identificar as atividades em um hangar de manutenção e a interação do operador com o ambiente ao seu redor: aeronaves, outros operadores, ferramentas e maquinários. Posteriormente, pretende-se fazer a implementação de um agente autônomo no ambiente de manutenção.

### 1.4.1 Objetivos específicos do trabalho

Para se obter o objetivo geral do trabalho, foram estabelecidos os seguintes objetivos específicos:

1. Identificar a infraestrutura de imageamento necessária para a captura de ações e tarefas em hangares de manutenção;
2. Determinar as técnicas de visão computacional disponíveis na literatura aplicáveis ao cenário descrito (caso ausente, desenvolver as técnicas necessárias conforme o objetivo geral do trabalho);
3. Captura e classificação dos elementos que compõem um hangar de manutenção e implementação de leitor de tarefas; e
4. Desenvolvimento de agente autônomo virtual capaz de interagir com o cenário.

## 1.5 Relevância da pesquisa

A relevância dessa pesquisa é contribuir significativamente com a indústria aeronáutica, a academia e a defesa nacional. Uma vez que:

- Na indústria e na defesa, o trabalho se destaca como uma ferramenta de apoio à decisão, reduzindo a sobrecarga de trabalho dos técnicos de manutenção e minimizando o risco de erros e acidentes; e
- Na academia, ao fornecer subsídios para a infraestrutura necessária à implementação das principais tendências descritas na literatura no contexto da aviação.

## 1.6 Delimitação da pesquisa

O estudo aqui descrito planeja fornecer os subsídios necessários para uma aplicação em hangares de manutenção aeronáutica, em especial os hangares de defesa, apresentando uma solução generalista que deve, contudo, ser adaptada conforme as condições da aplicação futura que venha a possuir.

## 1.7 Organização do trabalho

O presente trabalho é organizado da seguinte forma: o capítulo 1, finalizado neste subitem, apresentou o contexto de estudo (ambientação física e conceitual), as motivações do trabalho, o problema de pesquisa (qual o problema, por que é um problema, proposta de solução e contribuição acadêmica), os objetivos gerais e específicos e, por fim, a relevância e a delimitação de pesquisa. Pretende-se que fique claro, neste capítulo, que o corrente estudo aborda e visa solucionar uma necessidade latente de aumento da consciência situacional em hangares de manutenção, cuja primazia consiste em fornecer um aparato, pautado na literatura, sendo ou não tecnológico, capaz de resolver os desafios que aqui foram relatados.

O capítulo 2 é dividido em duas partes principais: a primeira descreve a fundamentação teórica necessária para a compreensão do trabalho, embasando, principalmente, a metodologia. Cabe destacar que esse embasamento é feito através da bibliografia presente no contexto acadêmico (no caso de termos já consolidados) ou artigos (ao se tratar de temas recentes). A segunda parte consiste na revisão da literatura, cujo principal objetivo é a análise de como outros autores abordaram problemas semelhantes e o estudo de decisão da metodologia que será aqui empregada.

O capítulo 3 descreve as ferramentas utilizadas e a metodologia do trabalho (arquitetura tecnológica e escolhas feitas) e apresenta uma sequência de ações necessárias para a sua replicação. O capítulo 4 apresenta os resultados obtidos e a discussão, relacionados aos objetivos específicos desta pesquisa. Por fim, o capítulo 5 descreve as conclusões do estudo.

## 2 Fundamentação teórica e revisão da literatura

### 2.1 Fundamentação Teórica

#### 2.1.1 Hangares de manutenção aeronáutica

Hangares de manutenção aeronáutica são locais destinados ao abrigo de aeronaves capazes de fornecer proteção contra as intempéries climáticas durante o período de permanência e manutenção. Seu dimensionamento depende das atividades previstas nas instalações e das aeronaves que serão recebidas em seu interior. Para a sua construção, deve-se atender aos seguintes documentos:

- Normas da ABNT
- Sistema de drenagem da área
- Normas da zona de proteção
- Normas de proteção contra incêndio

Dessa forma, espera-se que tais hangares tenham funcionamento regido pelas regras principais de seu dimensionamento, de forma que todos os requisitos de segurança sejam, a todo momento, cumpridos.

No escopo de hangares de manutenção, quatro grandes denominações se sobressaem: hangares privados (propriedades particulares que atendem aos donos de aeronaves privadas), hangares de aviação geral (acomodam aviões de pequeno porte, usualmente clubes de aviação e aeroportos de baixa demanda), hangares militares (projetados para abrigar as aeronaves de defesa. Se diferenciam dos hangares anteriores pelo seu porte, em geral, maior, e possivelmente existência de instalações direcionadas para épocas de conflito, como local para armazenamento de armamentos. Essa peculiaridade é refletida também nas atividades de manutenção dos operadores, os quais são sempre militares) e, por fim,

hangares destinados às linhas aéreas, que acomodam aviões de grande porte das grandes companhias aéreas.

Hangares de manutenção, de forma geral, são tidos como áreas de apoio para o funcionamento das atividades aeroportuárias. Também podem ser classificadas como áreas logísticas. Sua área engloba, dentre outras, as estruturas a seguir:

- Oficinas
- Pátios e edificações para as aeronaves
- Almoxarifados
- Escritórios

Cabe destacar, também, que no contexto da aviação geral, fazem-se tratamentos ergométricos para a estruturação dos hangares de manutenção. Além disso, em organizações militares, tornou-se comum o uso de hangaretes, específicos para um tipo só de aeronaves, visando proteger todos os sofisticados componentes dos infortúnios do tempo. De forma geral, hangaretes possuem portas grandes para a passagem de aeronaves e capazes de suportar um grande fluxo de operadores e veículos de apoio.

O interior dos grandes hangares de manutenção é um ambiente dinâmico: têm-se configurações variáveis que mudam a todo instante, povoado por pilotos, operadores (técnicos de manutenção), inspetores, técnicos de segurança, funcionários de limpeza, dentre outros. Grandes hangares de manutenção podem chegar a ter mais de 600 funcionários, que se dividem em três turnos de 8 horas por dia (24 horas), 7 dias na semana.

Para os propósitos dessa pesquisa, focar-se-à nos hangares de médio/grande porte, caracterizados por uma maior quantidade de funcionários, maior quantidade de inspeções, diversificação de aeronaves, extensa área construída para manutenção, presença de veículos motorizados passando a todo instante e diversificação de ferramentas.

Existem 4 principais tipos de manutenções de aeronaves:

- A manutenção de linha, mais simples, que acontece entre o pouso e a próxima decolagem. É mais rápida, devido ao pouco tempo disponível para avaliações, e visa verificar a existência de irregularidades que possam ter ocorrido durante o vôo. Caso alguma irregularidade seja observada, são acionados os mecânicos para o conserto de tais problemas.
- Manutenção de hangar (ou manutenção diária), é feita quando a aeronave fica por um período mais extenso (superior a 6 horas) no hangar. Visa detectar alguma anomalia que não seria observada em uma inspeção mais rápida e consertar esses

defeitos. São feitas no período noturno, em que são realizados checks previstos em manual (check A separado pela noite).

- Manutenção de check: está relacionada à quantidade de horas de voo e ao tipo de aeronave. É uma revisão completa e pode ser separada em A (realizada uma vez por mês, em período usualmente noturno), B (realizada em frotas mais antigas, com avaliações mais pesadas, podendo até ter duração de 48 horas) ou C (revisão completa da aeronave, ocorre após cerca de 12 ou 24 meses em aeronaves de companhias aéreas).
- O overhaul consiste na desmontagem completa da aeronave, sendo uma perícia completa que acontece, em geral, a cada 4 ou 6 anos. Nessa manutenção, a aeronave fica inoperante em um processo que pode durar um período superior a 2 meses.

No cenário descrito, somado ao fato de que os objetos de manutenção possuem turbinas, hélices rotativas e químicos perigosos e que a manutenção é feita com ferramentas capazes de ferir, torna-se natural pensar que a coexistência de várias operações simultâneas pode gerar um ambiente propício para a ocasionalidade de incidentes e acidentes que podem gerar riscos à vida, perda financeira, aumento da carga de trabalho na manutenção e, até, determinar o sucesso ou o fracasso de uma organização.

### **2.1.2 Fatores humanos e a manutenção aeronáutica**

De acordo com (HOBBS, 2008), fatores humanos referem-se ao estudo das capacidades e limitações humanas e suas interações com sistemas, equipamentos e procedimentos. No contexto da manutenção aeronáutica, aborda questões como ergonomia, comunicação, gestão de riscos e treinamento, visando minimizar erros, aumentar a segurança e otimizar a eficiência operacional. Muitas vezes, atribui-se ao fator humano a causa de incidentes, criando uma impressão de serem as pessoas as causadoras únicas e diretas. Entretanto, a maioria dos erros no ambiente, decisões e ações é resultado da maneira como se organiza o ambiente de trabalho, o design das tarefas, dos equipamentos, medidas de controle, influência dos líderes na cultura organizacional, dentre outros (OCIMF, 2024).

Conforme escrito por (STUSTER, 2006), os termos “fatores humanos” e ergonomia se associam à psicologia da engenharia, ao estudar o desempenho humano na operação de sistemas. Dessa forma, psicólogos de fatores humanos e engenheiros que atuam nessa área se preocupam com qualquer variável capaz de afetar o desempenho do operador de sistemas (seja este um hardware, um software ou pessoas). Dessa forma, atuam tanto no estudo quanto na aplicação de design ergonômico para equipamentos e procedimentos operacionais, além da seleção científica e do treinamento de operadores (STUSTER, 2006).

No escopo da manutenção aeronáutica, os fatores humanos desempenham um importante papel para a segurança e economia das companhias aéreas. Além disso, cerca de 18% dos acidentes e incidentes são relacionados aos fatores de manutenção. Na indústria, é comum que reparos e inspeções de aeronaves tenham duração superior a 12 horas (OLAGANATHAN, 2024). Dessa forma, técnicos precisam revezar os turnos, levando ao ato de mais de uma pessoa estar envolvida em uma determinada inspeção. Quando isso ocorre, são usadas instruções por escrito entre os técnicos de manutenção. Entretanto, instruções vagas geram maior ocorrência de erros. Além disso, problemas relacionados à fadiga, distúrbios gastrointestinais, estresses físicos e mentais, além de uma comunicação e interação ineficientes entre a equipe de manutenção, podem prejudicar o desempenho dos mecânicos e tornar a operação mais suscetível a erros (OLAGANATHAN, 2024).

Além dos fatores humanos citados, vários outros exercem significância crucial, tais como: a pressão e a temperatura dentro dos hangares de manutenção, ruído, espaço confinado e um ambiente com uma alta pressão demandante de eficiência temporal. Citam-se, também, problemas relacionados à falha de treinamento (lacuna técnica), falta de ferramentas específicas (na aviação geral e militar) e design para as operações no hangar. Esses fatores aumentam consideravelmente a probabilidade de ocorrência de erros, na medida em que interferem na capacidade de tomada de decisão da equipe de manutenção (OLAGANATHAN, 2024).

Em 1993, o pesquisador Gordon Dupont, da *Transport Canada*, fez um estudo aprofundado dos relatórios de 2000 acidentes e incidentes relacionados à manutenção que haviam sido classificados anteriormente como “erro humano”. O resultado desse estudo foi a descoberta de 12 condições gerais que causam uma predisposição aos erros de manutenção, que passaram a ser denominadas *Dirty Dozen*, ou seja, as doze causas de erros humanos, as quais são as seguintes (MELLEMA, 2018):

1. Fadiga
2. Falta de recursos
3. Distrações
4. Falta de Comunicação
5. Estresse
6. Falta de assertividade
7. Complacência
8. Normas

9. Falta de conscientização
10. Falta de conhecimento
11. Falta de trabalho em equipe
12. Pressão

O Chartered Institute of Ergonomics & Human Factors (CIEHF) (traduzindo para o português, Instituto de Ergonomia e Fatores Humanos) fez a categorização das áreas de engenharia e fatores humanos, das quais para melhor compreensão desse trabalho, e da ferramenta nele desenvolvida, serão aqui abordadas e descritas (conforme o CIEHF):

— No campo da psicologia:

- Atenção: as teorias relacionadas à forma como as pessoas prestam atenção e processam informações, e o conhecimento sobre limitações comuns.
- Comportamento e atitudes: as teorias relacionadas às influências e processos que afetam atitudes e comportamentos.
- Segurança comportamental: as atitudes e comportamentos relacionados à segurança, juntamente com as teorias e princípios envolvidos na criação de comportamentos seguros.
- Gestão de mudanças: os fatores e métodos envolvidos na gestão de mudanças nas organizações.
- Comunicação: As relações e comportamentos associados à comunicação pessoa-a-pessoa ou em grupo, tanto a nível individual quanto organizacional.
- Tomada de decisão: os processos cognitivos e vieses envolvidos na seleção de um curso de ação ou opinião.
- Aprendizagem: como os indivíduos adquirem novos conhecimentos, habilidades e atitudes, ou modificam os já existentes, por meio de experiência, estudo ou treinamento.
- Memória: os processos cognitivos envolvidos na aquisição, armazenamento e recuperação de informações a curto e longo prazo.
- Percepção: os mecanismos pelos quais as pessoas sentem, processam e interpretam informações através de seus sentidos.
- Psicofísica: A relação entre estímulos físicos e a resposta sensorial/perceptual de um indivíduo.

- Consciência situacional: como um indivíduo e/ou grupo percebe uma situação física/cognitiva em tempo real, como essa percepção muda, como influencia a tomada de decisão, e como pode ser medida, modelada e avaliada.
- Carga de trabalho: Como um indivíduo e/ou grupo é afetado pela carga de trabalho física, ou mental, especialmente sobrecargas, e o conhecimento de técnicas e restrições relacionadas à sua medição.

— No campo de pessoas e sistemas:

- Interação humano-computador: o projeto, a avaliação e a implementação de sistemas de computação interativos para uso humano.
- Integração de fatores humanos: os métodos/processos para a integração de fatores humanos na engenharia de sistemas durante o desenvolvimento de projeto.
- Sistemas homem-máquina: o projeto, a avaliação e a implementação de sistemas de máquinas interativos para uso humano.
- Design de trabalho: os fatores relacionados aos empregos e ao trabalho e sua relação com requisitos organizacionais, sociais e pessoais.
- Tarefas manuais: a natureza das tarefas manuais, os riscos envolvidos (por exemplo, fadiga, distúrbios musculoesqueléticos e lesões) e como esses riscos podem ser evitados ou mitigados conforme os regulamentos de manuseio manual.
- Design de produto: os métodos envolvidos no projeto, desenvolvimento, teste e uso de produtos.
- Cultura de segurança: os valores, atitudes, percepções e comportamentos exibidos por uma organização em relação à segurança.
- Sistemas sociotécnicos: as interações entre sistemas sociais e tecnológicos e seus efeitos na biologia humana, psicologia e desempenho em tarefas.
- Engenharia de sistemas: os métodos e processos no projeto e gestão de sistemas humanos e de engenharia complexos.

— No campo do ambiente de trabalho:

- Ambientes anormais: as normas, propriedades e efeitos de ambientes anormais e extremos na biologia, psicologia e desempenho em tarefas humanas.

- Ambiente auditivo: as normas, propriedades e efeitos do ambiente auditivo, incluindo ruído, reverberação e sons, na biologia, psicologia e desempenho em tarefas humanas.
- Ambiente mecânico: as normas, propriedades e efeitos do ambiente mecânico, incluindo vibração, choque, oscilação, forças G altas/baixas e variáveis, na biologia, psicologia e desempenho em tarefas humanas.
- Ambiente térmico: as normas, propriedades e efeitos do ambiente térmico, incluindo temperatura, umidade e movimentação do ar, na biologia, psicologia e desempenho em tarefas humanas, e como aplicar esse conhecimento.
- Ambiente visual: as normas, propriedades e efeitos do ambiente visual, incluindo nível e fluxo de luz, brilho, estroboscópios e cintilação, na biologia, psicologia e desempenho em tarefas humanas.
- Projeto e avaliação do local de trabalho: o projeto e a avaliação do espaço físico de trabalho.

— No campo dos métodos e ferramentas:

- Análise de tarefas: os métodos usados para representar tarefas de forma estruturada e descrever as atividades físicas e mentais dessas tarefas.

Os campos e subcampos descritos, retirados do CIEHF, abordam questões cruciais para o desenvolvimento desse trabalho e as principais motivações para a elaboração da ferramenta proposta.

### 2.1.3 Indústria 4.0 e Operador 4.0

O operador 4.0 consiste no trabalhador atuante na indústria 4.0, também conhecida como indústria da manufatura inteligente. Essa denominação é tida como a quarta revolução industrial, tendo sido precedida pelas seguintes denominações industriais (Jeevan S. Devagiri *et al.*, 2022):

- Indústria 3.0, pautada na automatização máxima possível dos processos;
- Indústria 2.0, focada na produção em massa;
- Indústria 1.0, que foi resultado de uma crescente mecanização.

De forma geral, a indústria 4.0 é pautada em tecnologias que objetivam acelerar o trabalho remanescente que exigem experiência e habilidades humanas, além de combinar uma robusta infraestrutura automatizada para fazer sistemas robóticos autônomos e capazes de tomar decisões baseados em dados (Jeevan S. Devagiri *et al.*, 2022).

Dessa forma, conforme se obtiveram progressos significativos que possibilitaram a classificação do que hoje se considera a indústria 4.0, foi observada uma consequente transição do operador para conseguir atuar nessa indústria. O operador 4.0 passa a ser entendido, então, como um operador que é mais esperto e habilidoso do que os que o precederam, podendo performar não somente em trabalhos cooperativos com robôs como também usando avançadas tecnologias que fazem a interface humano-máquina e automação adaptativa para se alcançar um sistema de trabalho simbiótico entre humano e automação (ROMERO *et al.*, 2016). Por isso, também é chamado de operador cognitivo.

Esse operador, quando inserido em seu local de trabalho, recebe atribuições e ferramentas distintas, podendo-se classificá-lo conforme a sua tarefa e como a executa. A seguinte categorização do operador 4.0 pode ser definida (ROMERO *et al.*, 2016):

- *Operador de Superforça*, movido por exoesqueletos que trabalham em cooperação com o operador, possibilitando o movimento dos membros, para aumentar a força e a resistência (SYLLA *et al.*, 2014).
- *Operador aumentado*, guarnecido com uma tecnologia de realidade aumentada capaz de sobrepor, no campo de visão do operador, informações digitais de som e mídia que enriquecem o ambiente real da fábrica, aumentando consideravelmente a consciência situacional do operador (Jeevan S. Devagiri *et al.*, 2022).
- *Operador virtual*, trabalha com sistemas de realidade virtual, sendo capaz de, por meio da criação de gêmeos digitais, replicar *design*, montagem ou ambiente de manufatura, permitindo a simulação com feedback imediato e risco reduzido de processos que poderiam ser, na prática, perigosos ou propensos a erros. Dessa forma, pode-se fazer a calibração dos equipamentos sem que, para isso, corra-se perigo (GKAGIANNIS; KOMNINOS, 2023).
- *Operador de bem-estar amplificado*, munido com dispositivos capazes de fazer a mensuração de dados relacionados à saúde, tais como níveis de estresse e frequência cardíaca. Dessa forma, pode-se fazer o emprego de bio informação em operadores que exercem funções delicadas como uma forma de melhorar o fluxo de trabalho, indicando, por exemplo, se o operador enfrentará uma situação de ambiente adversa (exposição a químicos acima do normal, calor ou frio extremo, dentre outros) para poder se preparar ou fazer as atividades em condições melhores (SUN *et al.*, 2020).

- *Operador inteligente* é o operador que interage com uma inteligência artificial desenvolvida para poder auxiliar o trabalhador na interface com máquinas, computadores e sistemas de informação, além de administrar compromissos de tempo, também envolvendo a execução de tarefas ou serviços por meio de interações que se assemelham às humanas (MYERS *et al.*, 2007).
- *Operador colaborativo* são robôs colaborativos atuando em conjunto aos outros operadores da indústria, executando uma variabilidade de tarefas que não são ergonômicas e são repetitivas (NILSSON; JACOBSSON, 2017).
- *Operador social*, é o operador que possui conexões sociais e móveis com os recursos da indústria 4.0. Essas conexões podem ser as relações sociais existentes na equipe ou entre os operadores e os dispositivos inteligentes, de modo a possibilitar o suporte para tomada de decisões (WUEST *et al.*, 2012).
- *Operador analítico*, que usa a análise de Big Data como uma ferramenta importante para descobrir informações importantes e prever eventos relevantes (WUEST *et al.*, 2016).

Cabe ressaltar, portanto, que na indústria 4.0 não se tem a substituição dos trabalhadores por máquinas, pelo contrário: o operador 4.0 passa a ser o cerne das operações, devendo a tecnologia fornecer subsídios para exercer a sua função da forma mais eficiente e segura possível.

Nesse contexto, os hangares 4.0 surgem como hangares de manutenção de aeronaves nos quais os processos são tidos como equivalentes aos da indústria 4.0 e os técnicos são tidos, dessa forma, como possíveis operadores cognitivos.

#### 2.1.4 Visão computacional

Para se entender as técnicas de visão computacional mais recentes, deve-se primeiro compreender o processo de imageamento que permite que os dados de imagem possam ser trabalhados. A visão computacional é uma tecnologia que permite às máquinas capturar, interpretar e interagir com imagens de forma inteligente, alinhada a objetivos específicos. Com a crescente demanda por automação e precisão em diversas áreas, o aprimoramento e a aplicação contínua dessas técnicas torna-se cada vez mais necessário no mundo moderno.

Primeiramente, ressaltam-se duas representações possíveis para uma imagem: a computacional e a matemática. Duas importantes definições que podem ser abordadas são (SZELISKI, 2011):

a) Uma matriz discreta com valores discretizados (essa é a representação padronizada em imagens na computação). Tem-se a discretização da imagem em imagens menores

(normalmente com 8-bits) que recebem o nome de *pixel* e ocupam as posições da matriz. A cada um desses *pixels* serão atribuídos valores de 0 a 255.

b) Matriz discreta de valores “reais/contínuos”: tem-se a discretização da imagem em imagens menores (normalmente com 8-bits) que recebem o nome de *pixel* e ocupam as posições da matriz. A cada um desses *pixels* serão atribuídos valores de contínuos (usualmente, valores entre 0 e 1).

Além dessas representações, existem também operações possíveis que são resultado do uso do caso contínuo (por exemplo, derivadas e transformações lineares) para a atribuição dos valores do *pixel*. Por meio do recebimento desses dados, a visão computacional visa extrair informações da imagem como texturas, objetos presentes, contexto e posições. Constituem áreas da visão computacional, por exemplo, (SZELISKI, 2011):

- Segmentação de imagem
- Rastreamento de movimento
- Reconhecimento de objetos
- Reconstrução 3D

No presente momento, o estado da arte quando se trata de detecção de objetos é a YOLO (You Only Look Once). Essas redes podem fazer a classificação e a detecção dos objetos (caso YOLOv8, também, a segmentação). Industrialmente, essa arquitetura se torna aplicável a inspeções visuais, possibilitando a diminuição de erros em produtos e agilidade na linha de produção (GONZAGA *et al.*, 2023).

Ainda sobre a detecção de objetos e tarefas, alguns dos algoritmos existentes, além do YOLO, ganham destaque: CNN (Redes Neurais Convolucionais, ou, no inglês, *Convolutional Neural Network*), R-CNN, Fast R-CNN, Faster R-CNN e SDD (Detector de Única Passagem, ou, no inglês, *Single Shot Detector*) (KRISHNA *et al.*, 2023).

As Redes Neurais Convolucionais (CNN) são um algoritmo de aprendizagem profunda capaz de processar uma imagem inicial, atribuindo importância aos diversos aspectos e objetos, diferenciando-os. Comparativamente aos outros algoritmos de classificação, exige menos pré-processamento. Isso porque, enquanto os métodos primitivos necessitam de uma criação manual de filtros, na CNN esses filtros podem ser aprendidos com treinamento suficiente. Pode ser representada de forma análoga pelos neurônios do cérebro humano.

O uso dessas redes permite capturar as dependências espaciais e temporais existentes em uma imagem por meio de aplicação de filtros relevantes. Com isso, ajusta-se melhor os dados da imagem, devido à redução do número de parâmetros e à reutilização de pesos. Dessa forma, a rede compreende melhor a complexidade da imagem (PHAND *et al.*, 2023).

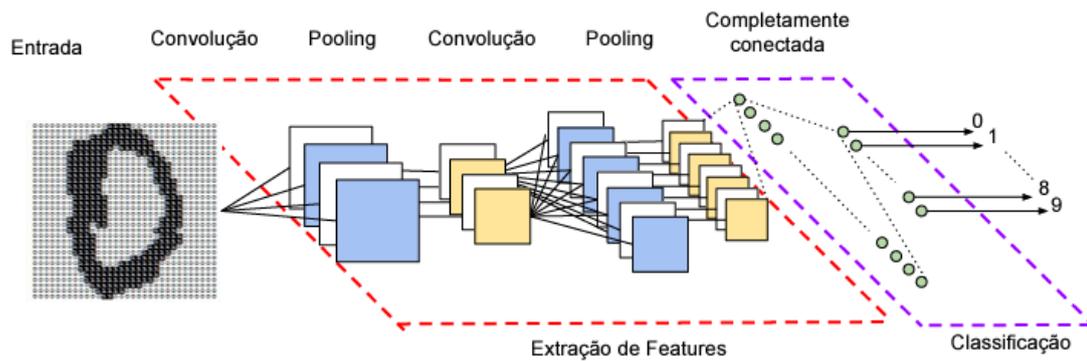


FIGURA 2.1 – Exemplo de uma rede neural convolucional e suas diferentes camadas, retirada de (VARGAS *et al.*, 2023).

O R-CNN (*Region-Based Convolutional Neural Networks*) consiste em um algoritmo de detecção que usa busca seletiva para gerar 2000 regiões por meio da imagem inicial (propostas de região). O sistema de detecção de objetos é formado por três componentes: o primeiro cria propostas de região independentemente de categoria, o segundo consiste em uma grande CNN que extrai um vetor de características e, por fim, o terceiro é composto por uma coleção de Máquinas de Vetores de Suporte (OLORUNSHOLA *et al.*, 2023).

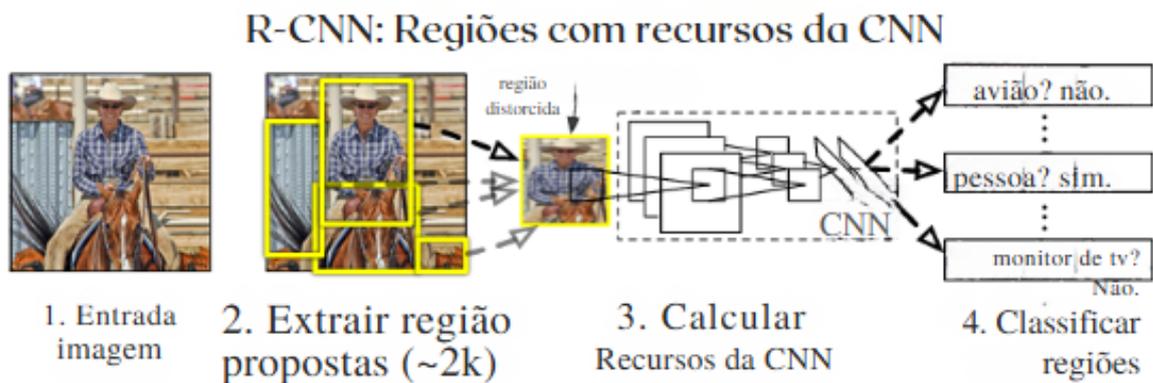


FIGURA 2.2 – R-CNN: regiões com características CNN. Imagem retirada de (GIRSHICK *et al.*, 2014) e traduzida.

O Fast R-CNN é uma melhoria do algoritmo R-CNN. Nele, a imagem é inserida para criar, ao invés do plano de 2000 regiões, um mapa de características convolucionais. Esse algoritmo utiliza diversas inovações para aumentar a precisão da detecção, enquanto aumenta a velocidade de treinamento e de teste do modelo (OLORUNSHOLA *et al.*, 2023).

O Faster R-CNN é resultado da introdução das redes de proposta de região (RPNs). Devido ao compartilhamento de propriedades convolucionais da rede de detecção downs-

tream, a fase de proposta de região é isenta de custos. Nesse algoritmo, um sistema unificado de detecção de objetos baseado em Deep Learning consegue operar em velocidade de quadros em um tempo quase real. De forma geral, tem-se o aumento da precisão de detecção de objetos por meio da RPN aprendida, melhorando a qualidade da proposta de região. Enquanto o R-CNN e o Fast R-CNN usam busca seletiva, demandada para encontrar propostas de região, o Faster R-CNN usa RPN, o qual é mais rápido (OLORUNSHOLA *et al.*, 2023).

No SDD (*Single Shot Multibox Detection*), tem-se a introdução de técnicas de detecção multi-referência e multirresolução, aumentando-se de maneira significativa a precisão de detecção de estágio único (principalmente para objetos pequenos). Em termos amplos, a principal diferença entre o SDD e os métodos anteriormente explicados é que o SDD faz a detecção de objetos em 5 escalas diferentes, enquanto os outros métodos fazem a detecção em suas camadas anteriores (OLORUNSHOLA *et al.*, 2023).

OpenCV consiste em uma biblioteca de programação de código aberto (Open Source Computer Vision). É utilizada em diferentes tipos de análise de imagens e vídeos. Funciona por meio do processamento de imagens em tempo real, possuindo mais de 300 algoritmos de visão computacional. Possui o seguinte algoritmo (BRADSKI, 2000):

1. Captura de imagem
2. Detecção
3. Pré-processamento (filtro)
4. Inspeção e análise
5. Resultado

O *Barracuda* é uma biblioteca desenvolvida pela Unity que permite executar modelos de aprendizado profundo diretamente no Unity, sem a necessidade de software adicional. Ela é especialmente útil para executar modelos de machine learning em dispositivos móveis e outras plataformas onde o desempenho é crucial.

Por fim, na arquitetura YOLO, a imagem é dividida em grades de mesmo tamanho, individuais, com caixas delimitadoras únicas. Para cada uma das caixas delimitadoras, serão criadas umas probabilidades de classe e um número de deslocamentos. Essas caixas delimitadoras são usadas com probabilidades de classe em número acima de um número limite para que o objeto na imagem seja encontrado. De maneira inovadora, o YOLO separa espacialmente as caixas delimitadoras e probabilidades de classe relacionadas, fazendo a detecção de objetos em frames, de maneira análoga a um problema de regressão.

Posteriormente, foram lançadas séries (v2,v3 e assim por diante até o momento de condução da pesquisa (v8) com melhorias em relação às versões de YOLO anteriores. De forma geral, o YOLO demonstra eficiência e agilidade para reconhecimento e localização (OLORUNSHOLA *et al.*, 2023).

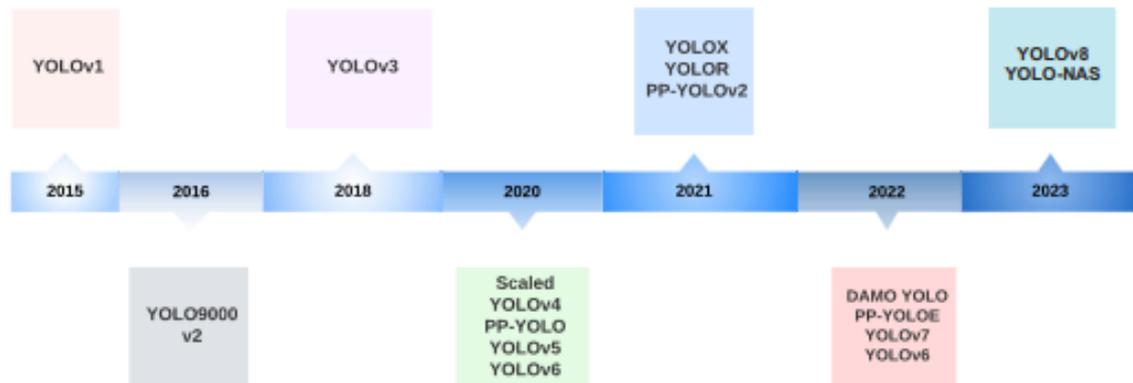


FIGURA 2.3 – Linha do tempo das versões do YOLO, retirada de (TERVEN *et al.*, 2023).

O YOLOv8 possui a arquitetura dada pela figura 2.4.

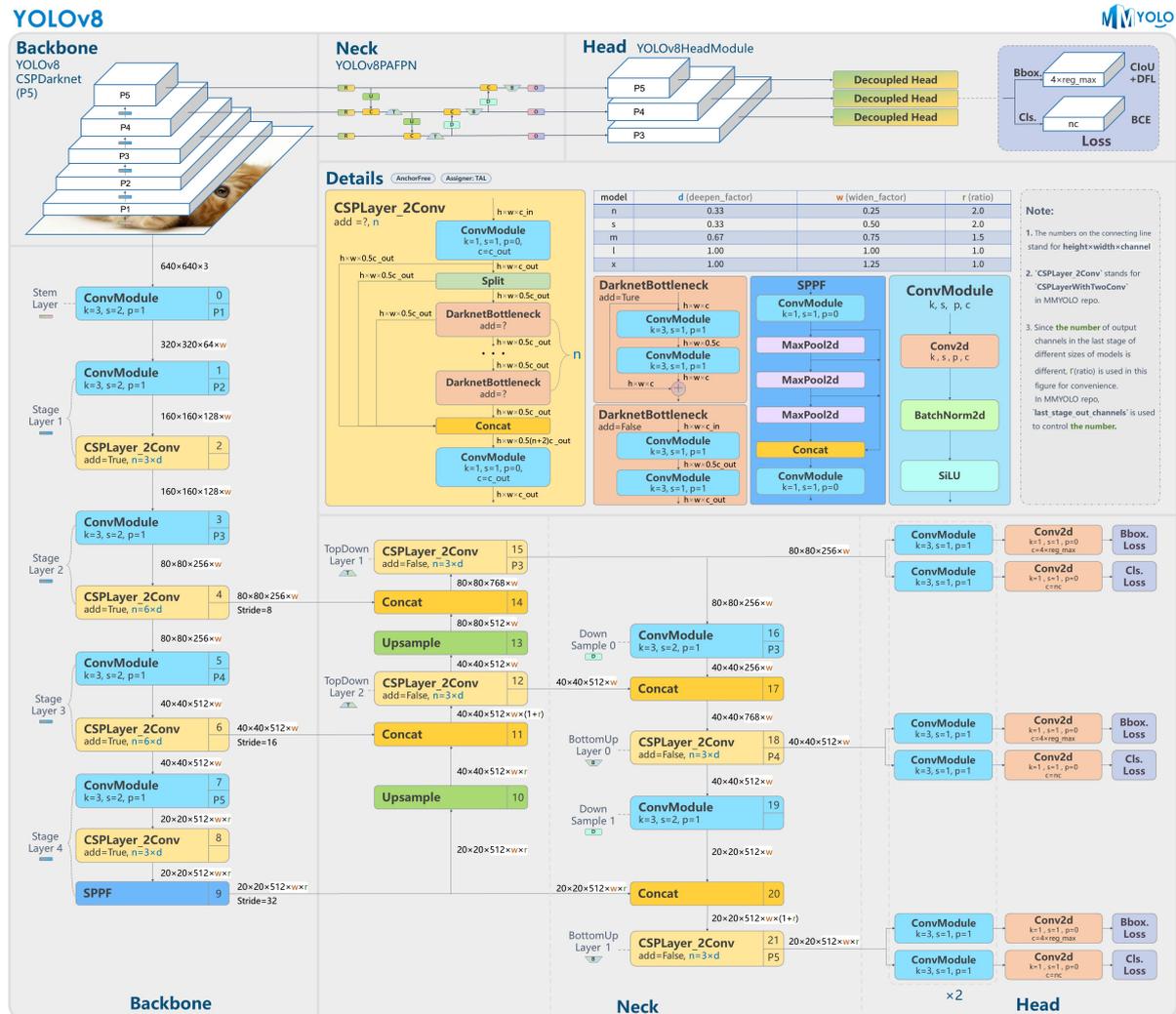


FIGURA 2.4 – Arquitetura do YOLOv8, retirada de (CONTRIBUIDORES, 2023).

Conforme (TERVEN *et al.*, 2023), as versões principais de YOLO são explicadas, cronologicamente, da seguinte forma: o YOLOv1, pioneiro ao integrar a detecção de objetos em um único passo, usa uma arquitetura que divide a imagem em uma grade fixa e prevê caixas delimitadoras e classes para cada célula. Se caracteriza por ser rápido, mas apresentando limitações para a precisão da localização e da detecção de objetos. O YOLOv2 obteve consideráveis melhorias de precisão ao adicionar normalização em lote, resolução mais alta durante o treinamento, âncoras para prever caixas delimitadoras e uma camada de passagem que conecta características de baixa resolução com alta resolução.

O YOLOv3 fez a introdução de uma nova arquitetura de backbone chamada Darknet-53, melhorou a previsão multi-escala e utilizou classificadores binários independentes para cada classe, permitindo a sobreposição de rótulos. O YOLOv4 fez o aumento de mosaicos, regularizações como DropBlock e técnicas avançadas como SAT (treinamento adversarial) e focou em melhorias como conexões parciais entre estágios (CSPNet). O Yolov5 foi desenvolvido pela Ultralytics e caracteriza-se por ser otimizado em PyTorch e ter introduzido

melhorias como ancoragem automática e aumento de dados avançado, tornando-o fácil de treinar e implantar.

O YOLOv6 fez a melhora da eficiência com novos blocos backbone, estratégias de distorção de canal e perdas de regressão aprimoradas, mantendo alta precisão e velocidade. Já o YOLOv7 fez a proposta do E-ELAN para agregar recursos de diferentes camadas e técnicas de escalonamento para otimizar a arquitetura com foco em eficiência e desempenho. Por fim, o YOLOv8 faz a utilização de um modelo sem âncoras e uma cabeça desacoplada para melhorar a precisão, especialmente em objetos pequenos. Oferece suporte a múltiplas tarefas, como detecção, segmentação e estimativa de pose, conforme citado anteriormente.

### 2.1.5 Unity

A *Engine Unity* é um motor para desenvolvimento 3D em tempo real que permite criar diversos sites, aplicações e jogos. É um motor de renderização e física, possuindo uma interface gráfica (*Unity Editor*), que permite uma prototipagem rápida e o desenvolvimento de jogos e ambientes simulados. Historicamente, o foco de criação do *Unity* foi o desenvolvimento de um motor de propósito geral, capaz de suportar diversas plataformas, tipos de aplicações e níveis de experiência dos desenvolvedores. Os desenvolvedores podem usar somente a interface disponibilizada pelo *Unity* ou trabalhar com a criação de *scripts* com a linguagem de programação C#. Um projeto no *Unity* possui os seguintes componentes (JULIANI *et al.*, 2018):

- *Assets*: Arquivos que compõem o projeto
- *Cenas*: um tipo específico de *Asset* que definem o ambiente e a hierarquia de um projeto
- *Game Objects*: correspondem aos objetos reais (sejam físicos ou puramente lógicos) no ambiente
- Outros componentes integrados, como câmeras, malhas, corpos rígidos, renderizadores, dentre outros.

Devido às suas características, essa *engine* permite a criação de gêmeos digitais, simulações de ambientes e tarefas, além da imposição de gatilhos capazes de reagir aos elementos do ambiente, tornando-a uma ótima alternativa para a condução dessa pesquisa.

### 2.1.6 Unreal Engine

De maneira similar ao *Unity*, o *Unreal Engine* constitui um motor de jogo capaz de criar aplicações em realidade virtual e realidade aumentada.

Em ambientes industriais como a aviação (mais especificamente, manutenção aeronáutica), a simulação de situações complexas permite a antecipação de problemas (tais como a colisão entre mecânicos, aeronaves e veículos) e a otimização do fluxo de trabalho.

Nesse escopo, a utilização de um motor gráfico como o *Unreal Engine* possibilita a criação de representações visuais realistas essenciais para que se possam ser desenvolvidas as simulações citadas no parágrafo anterior. Uma vez que, além de permitir o *design* tridimensional do espaço projetado, implementa as leis físicas e permite a replicação dos processos operacionais em hangares de manutenção.

No trabalho desenvolvido por (QIU *et al.*, 2017), o autor se refere ao *Unreal* como uma das melhores escolhas para o desenvolvimento de trabalhos de realidade virtual, além de ser altamente aberto para a comunidade. Cita-se, também, que o *Unreal* reduz significativamente o custo para o acesso de gráficos computacionais de alta qualidade.

Adicionalmente, o *Unreal* pode ser usado com bibliotecas de visão computacional (por exemplo, o OpenCV e o TensorFlow). Dessa forma, torna-se uma possível ferramenta metodológica para a integração de algoritmos de inteligência artificial e de *machine learning* para a detecção de objetos e análise situacional do hangar.

### 2.1.7 Câmeras de profundidade

Enquanto as câmeras convencionais capturam somente informações de luz e cor em um ambiente, as câmeras de profundidade conseguem registrar informações tridimensionais. De forma que se pode fazer a definição do sensoriamento de profundidade como a capacidade de se medir o distanciamento entre dois objetos ou entre um dispositivo e um objeto.

Ao se usar visão computacional em um ambiente de manutenção, a complexidade das tarefas e das operações tornam imprescindível o uso dos dispositivos aqui relatados para uma leitura eficiente do cenário (ou seja, para que se possa, de fato, mensurar cada aspecto compositor importante do hangar, posicionando-os geograficamente em um espaço 3D, além de conseguir interpretar as informações recebidas para que, a partir delas, possam ser tomadas decisões).

As câmeras de profundidade podem funcionar por tecnologias distintas, das quais se podem citar:

- Estereoscopia, que simula o funcionamento dos olhos humanos
- Time of Flight, em que se estima a distância entre os objetos por meio do cálculo do tempo em que a luz emitida por sensor leva para retornar
- Direct Time of Flight, em que o cenário é escaneado através de pulsos de laser infravermelho
- LiDAR
- Indirect Time of Flight

TABELA 2.1 – Comparação entre as principais tecnologias de câmeras de profundidade de (E-CON SYSTEMS, 2024)

	<b>VISÃO ESTÉREO</b>	<b>LUZ ESTRUTURADA</b>	<b>dToF</b>	<b>iToF</b>
<b>Princípio</b>	Compara disparidades de imagens estéreo de dois sensores 2D	Detecta distorções de padrões iluminados pela superfície 3D	Mede o tempo da luz refletida de um objeto até o sensor	Mede a mudança de fase de pulsos de luz modulada
<b>Complexidade do Software</b>	Alta	Média	Baixa	Média
<b>Custo do Material</b>	Baixo	Alto	Baixo	Médio
<b>Precisão de Profundidade (“z”)</b>	cm	µm cm	mm cm	mm cm
<b>Faixa de Profundidade</b>	Limitada	Escalável	Escalável	Escalável
<b>Baixa Luminosidade</b>	Fraca	Boa	Boa	Boa
<b>Ambiente Externo</b>	Bom	Fraco	Razoável	Razoável
<b>Tempo de Resposta</b>	Médio	Lento	Rápido	Muito Rápido
<b>Compacidade</b>	Baixa	Alta	Alta	Média
<b>Consumo de Energia</b>	Baixo	Médio	Médio	Escalável-Médio

Os algoritmos de visão computacional, quando integrados às câmeras de profundidade, podem ter desempenho aumentado e melhora do reconhecimento de objetos 3D, além de favorecer o rastreamento preciso da movimentação de elementos.

### **2.1.8 Manuais de manutenção**

As manutenções aeronáuticas são feitas por meio de manuais de manutenção, que descrevem os tipos de manutenção que devem ser feitas em uma cronologia pré-definida e as tarefas que deverão ser executadas. Para essa pesquisa, foram utilizados manuais de manutenção usados pela Força Aérea Brasileira e disponibilizados única e exclusivamente para os fins dessa pesquisa.

Os manuais utilizados foram:

- Manual de Mecânico C-105A Amazonas 1<sup>o</sup>/9<sup>o</sup> GAV
- Manual disponibilizado pelo fabricante da aeronave C-105
- Aviation Maintenance Technician Handbook — General

O Manual do Mecânico C-105A Amazonas 1<sup>o</sup>/9<sup>o</sup> GAV tem como finalidade a definição da filosofia e os procedimentos que devem ser observados na operação de manutenção das aeronaves C-105A, de modo a aumentar a segurança, o grau de padronização e a eficiência dos mecânicos do 1<sup>o</sup>/9<sup>o</sup> Grupo de Aviação. Dessa forma, esse manual foi consultado como uma complementação do manual disponibilizado pelo fabricante, de modo a abordar, de maneira mais operacional (no contexto da Força Aérea), as tarefas de manutenção que deverão ser mapeadas. A imagem 2.5 mostra o avião C-105.



FIGURA 2.5 – Imagem da aeronave C-105, disponibilizada no site da Força Aérea Brasileira (FAB)

Salienta-se que uma prática obrigatória nos manuais de manutenção (de uma forma geral) é o uso de check-lists para a garantia de que todas as tarefas sejam feitas da forma correta.

### **2.1.9 Riscos críticos em hangares de manutenção**

Hangares de manutenção são ambientes caracterizados pela natureza complexa das atividades que ocorrem em seus interiores e pela existência de perigos relacionados à presença de equipamentos pesados, produtos químicos, atividades potencialmente perigosas e trânsito entre operadores e servidores de uma forma geral, aeronaves e carros de transporte e ferramentas. Nesse contexto, para garantir a operacionalidade das companhias aéreas, é crucial a garantia da segurança ocupacional e eficiente das atividades mantenedoras. Dessa forma, torna-se fundamental o estudo dos riscos críticos dentro desses ambientes, que serão detalhados nos parágrafos seguintes.

Cabe destacar, também, a natureza dos equipamentos presentes nos hangares, os quais são: guindastes, ferramentas elétricas e maquinário pesado, ferramentas elétricas e plataformas elevatórias.

Podemos classificar os riscos em hangares da seguinte forma:

1. Riscos ergonômicos: relacionados à postura inadequada, movimentos repetitivos e manuseio de cargas pesadas.
2. Riscos elétricos: estão associados aos choques elétricos e curtos-circuitos.
3. Riscos químicos: compõem os riscos químicos os perigos decorrentes do manuseio de substâncias tóxicas e da exposição à poeira gerada (no lixamento de peças), fumos e gases e vapores tóxicos como o monóxido de carbono e outros decorrentes da queima de combustíveis.
4. Riscos de acidentes: os riscos de acidentes são ocasionados por: quedas (de altura ou no mesmo nível (escorregões e tropeços ocasionados por derramamentos, obstáculos ou superfícies irregulares), impactos e colisões, projeções de partículas durante operações de perfurações, cortes ou lixamentos e, por fim, esmagamentos entre partes móveis de máquinas ou entre equipamentos e estruturas.
5. Riscos físicos: podem ser compostos por exposição excessiva a ruídos, os ocasionados por vibrações, altas temperaturas e iluminação inadequada.
6. Riscos Psicossociais: gerados por fadigas, violências e assédios e estresses ocupacionais.
7. Riscos mecânicos: que ocorrem pelo uso de ferramentas inadequadas ou falhas de equipamentos.

Como forma de diminuir a probabilidade dos riscos destacados, existem regulamentações de segurança que devem ser aplicadas em hangares. Cabe mencionar, no entanto, que os erros humanos, causados por fatores como fadigas, estresse e falta de treinamento, aumentam consideravelmente os riscos críticos em hangares.

Entretanto, como o escopo desse trabalho não prevê a mensuração do nível de atenção dos mecânicos e operadores, serão abordadas as formas de mitigação de risco em um viés puramente objetivo, usando para isso a visão computacional.

### **2.1.10 *Cloud Data Fusion***

O termo “*data fusion*” é usado para se referir à fusão de dados e informações. Sinteticamente, a fusão de dados usa a sobreposição de informação para a definição de relacionamentos. Segundo Zheng Liu etc. al., para a garantia da criação de gêmeos digitais, tem-se um ecossistema formado por sensores e medições, internet das coisas industrial, modelagem, simulação e aprendizado de máquina. Computacionalmente, a tecnologia chave para

o seu impulsionamento é a fusão de dados e de informações, capaz de facilitar o fluxo de informações, desde os dados sensoriais brutos até entendimentos e insights de alto nível (LIU *et al.*, 2018). Paralelamente, para a criação de simulações de hangares de manutenção e a predição de eventuais riscos, necessita-se de uma ferramenta capaz de integrar os dados obtidos por meio de câmeras e eventuais sensores para a leitura e interpretação do cenário em estudo.

#### 2.1.10.1 Google Cloud Data Fusion

É uma plataforma oferecida pela Google Cloud de integração de dados que possibilita a criação e o gerenciamento de dados de múltiplas fontes na nuvem, fornecendo uma interface gráfica intuitiva para projetar, desenvolver e implantar fluxos de trabalho, sendo altamente escalável.

#### 2.1.10.2 Modelo JDL Data Fusion

O modelo JDL de fusão de dados distingue diferentes “níveis” de fusão, visando prover uma categorização útil capaz de representar diferentes tipos de problemas, que geralmente serão resolvidos por técnicas diferentes, porém mantendo um grau de consistência com o uso técnico convencional (STEINBERG *et al.*, 1999). Os níveis desse modelo são descritos por (STEINBERG *et al.*, 1999):

Nível 0: é o pré-processamento. É a estimativa e a previsão de estados de sinal e objeto observados por meio da associação e da caracterização dos dados ao nível de *pixel* e sinal.

Nível 1: é o reconhecimento e a classificação do objeto detectado. É a estimativa e a previsão dos estados das entidades com base na associação de observações para o acompanhamento, incluindo a estimativa de estados contínuos (por exemplo, dados cinemáticos, a identificação e a classificação de alvos).

Nível 2: é a avaliação situacional. Faz a estimativa e a previsão das relações entre as entidades, como as relações de forças, contexto físico, influência perceptual etc.

Nível 3: consiste na avaliação de impacto. Nessa classificação, é feita a estimativa e a previsão dos efeitos em situações planejadas ou previstas pelos participantes.

Nível 4: faz-se o refinamento do processo. Aqui, fazem-se a aquisição e o processamento para apoiar os objetivos da missão.

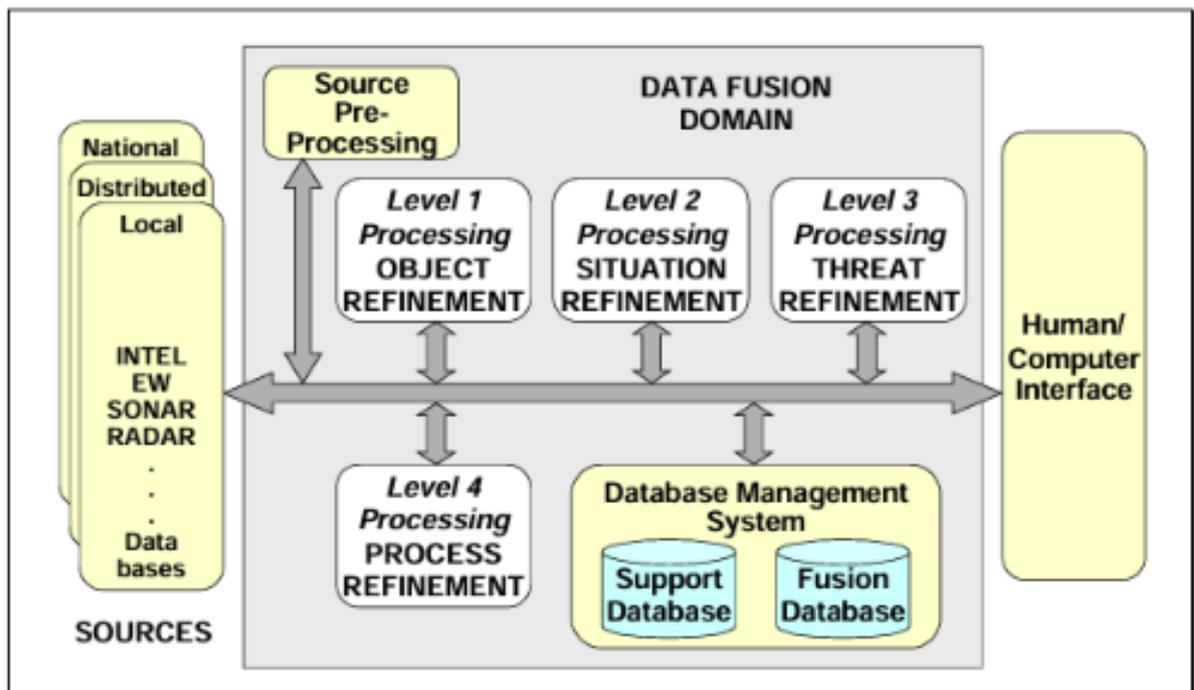


FIGURA 2.6 – Modelo JDL de fusão de dados (versão de 1992), retirada de (STEINBERG *et al.*, 1999)

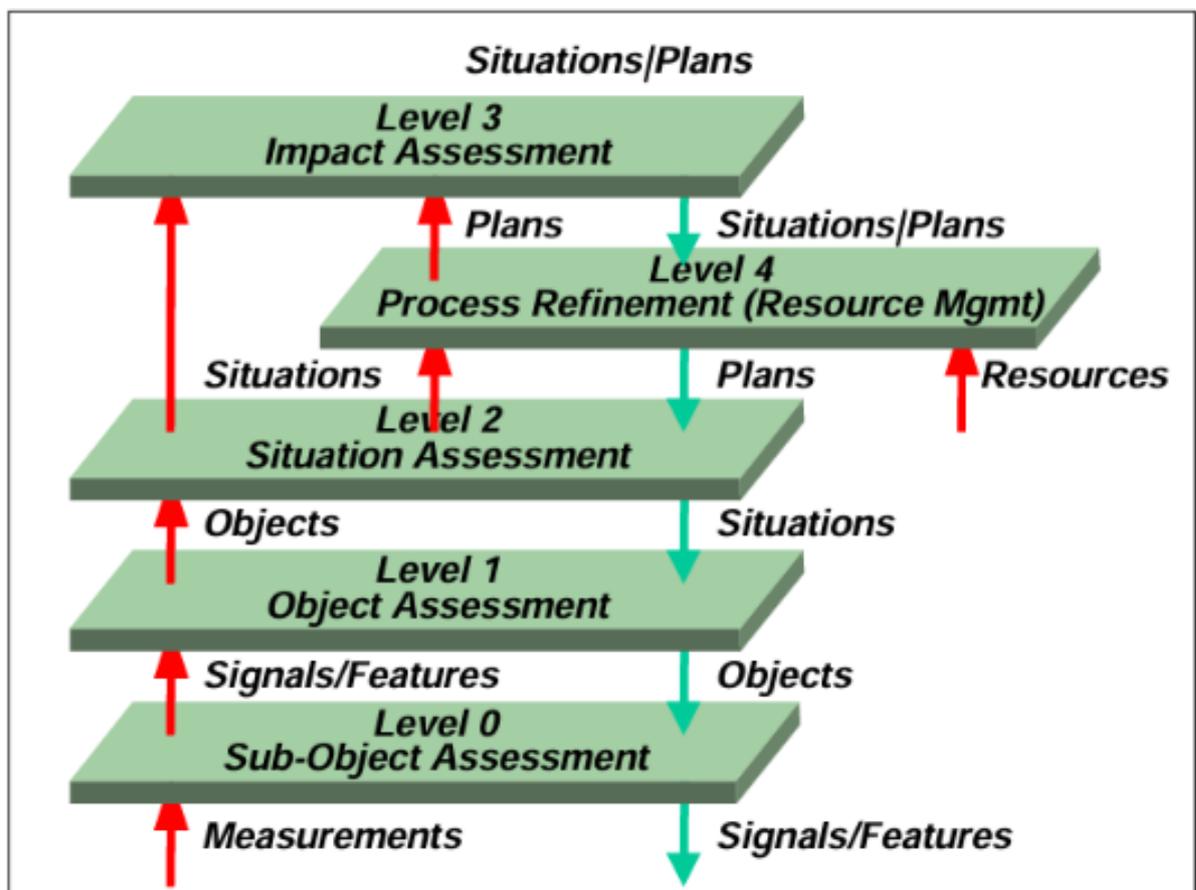


FIGURA 2.7 – Fluxo lógico entre os níveis, retirada de (STEINBERG *et al.*, 1999)

## 2.2 Revisão da literatura

Para a mensuração do impacto desta pesquisa, foi feita uma leitura e seleção criteriosa de publicações acadêmicas que envolvem os tópicos abordados dados a seguir:

- (1) — Manutenção
- (2) — Visão Computacional
- (3) — Mapeamento de tarefas
- (4) — Fatores humanos
- (5) — Indústria 4.0
- (6) — Inteligência Artificial
- (7) — Contexto Aeronáutico

Posteriormente, para um direcionamento ainda maior para a escrita desta seção, foram atribuídos valores iguais a 1 em cada um dos tópicos abordados, de forma que o valor preliminar de cada artigo para essa pesquisa (em uma análise quantitativa que não considera qualitativamente o impacto desses artigos na metodologia abordada) é dado pelo somatório final que compõe cada um dos tópicos. Para a revisão da literatura, selecionaram-se artigos cujo valor foi maior ou igual a 3.

Os artigos principais usados como referência bibliográfica foram dispostos na tabela 2.2.

TABELA 2.2 – Comparação entre estudos antecedentes quanto aos assuntos abordados

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Gonzalez e Safabakhsh (1982)			x	x	x		
Reason (1997)	x	x			x		
Sá e Zachmann (1999)	x	x					x
Golnab e Asadpour (2007)	x				x		x
M. Ward et al. (2010)	x			x	x		x
Crescenzo et al (2011)		x	x	x	x	x	
Orteu et al (2015)	x			x			x
Frigo et al (2016)	x	x			x	x	x
Damla Aslan et al (2019)			x		x	x	
Xu et al (2019)	x		x	x			
Safi et al (2019)	x	x		x	x	x	x
Miranda et al (2019)	x	x			x		x
Bouarfa et al (2020)	x	x			x		x
Aiello (2020)		x			x	x	
Jeevan S. Devagiri et al (2022)	x	x	x				
Osigbesan et al (2022)	x	x		x			x
Pardomoan et al (2022)	x	x				x	
Malta et al (2023)		x			x	x	
Romeral et al (2023)	x	x		x	x	x	x
Ruipeng Zhao e Yin'an Tian (2023)			x	x	x		x
Casey et al (2024)					x	x	x

Defendem-se que mundos virtuais realistas são úteis para pesquisas (principalmente de visão computacional)(QIU *et al.*, 2017), podendo serem usados para muitas tarefas que o mundo real não consegue realizar (por exemplo, testar algoritmos de visão computacional em situações perigosas, realizar treinamentos caros de robótica e gerar inúmeras imagens com parâmetros difíceis de serem obtidos). Segundo o autor, a criação de mundos virtuais realistas em uma *game engine* permite a solução de diversos problemas apenas fazendo a modificação do cenário, possibilitando a criação de mundos virtuais que possam atender às exigências dos pesquisadores. Nesse trabalho, o autor utiliza o *UnrealCV*, ferramenta de visão computacional com integração no Unreal, que será testada como metodologia ao longo deste trabalho de graduação. A defesa do autor é de que tal ferramenta possui uma alta qualidade e que é uma das melhores escolhas para a criação de um ambiente virtual.

No trabalho de (OSIGBESAN *et al.*, 2022), são usadas câmeras e visão computacional para a detecção de quedas em hangares de manutenção, utilizando uma fusão não intrusiva de gravações de vídeo e análise de movimento com visão computacional para a detecção

de quedas. A proposta do autor é o uso de câmeras com lente grande angular instaladas na posição diagonal no teto do hangar para captar os dados visuais, utilizando o modelo de detecção de quedas que permite treinamento em lote e compreensão da distribuição temporal dos dados de entrada. Cabe destacar que o autor sugere um mínimo de 3 câmeras para lidar razoavelmente com oclusões (porém ressalta que esse número pode ser aumentado dependendo das configurações do hangar).

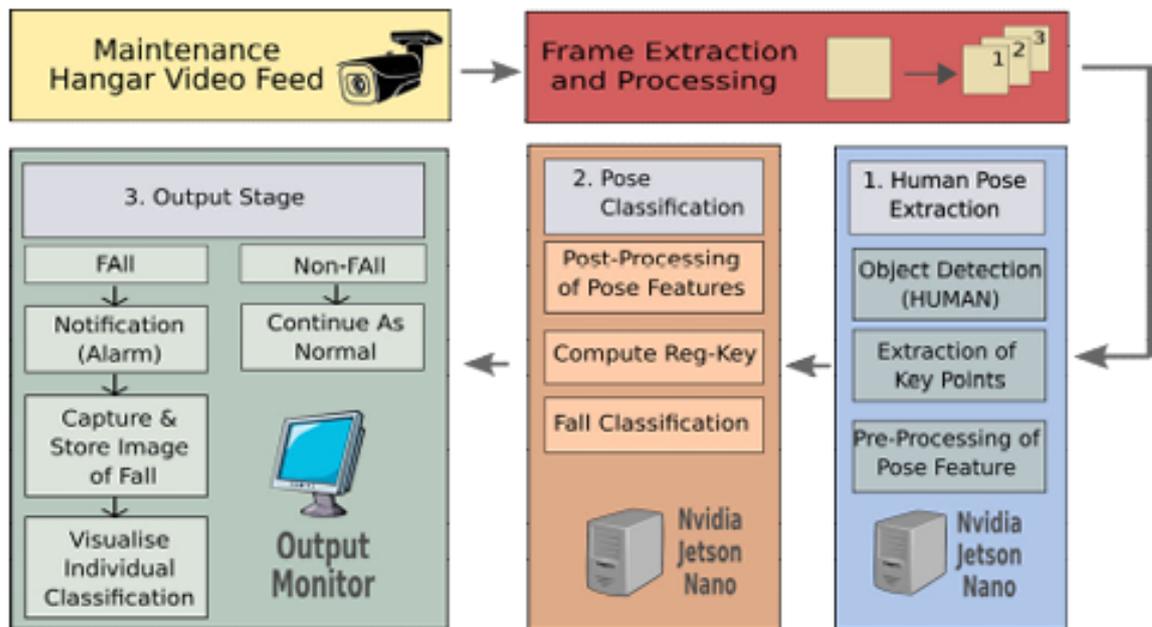


FIGURA 2.8 – Proposta da arquitetura do sistema de detecção de quedas, retirado de (OSIGBESAN *et al.*, 2022)

Dessa forma, a proposta desse trabalho é utilizar metodologia similar à utilizada por (OSIGBESAN *et al.*, 2022), em um contexto de hangar de manutenção aeronáutica, a detecção de objetos, aeronaves, técnicos e carros de ferramenta, para o cenário descrito poder ser capturado, identificado e interpretado computacionalmente, de forma que o trabalho, por si só, constitua uma ferramenta gerencial capaz de fornecer *insights* do hangar para os respectivos responsáveis por cada informação.

(ROMERAL *et al.*, 2023), em seu trabalho, analisa tendências no uso de visão computacional, como detecção de objetos e segmentação, apresentando exemplos práticos que demonstram como a visão computacional pode ser adaptada para inspeções em hangares, aumentando a precisão e reduzindo tempos de manutenção. Ainda no escopo da inspeção visual, (XU *et al.*, 2019) abrangeu técnicas de visão computacional aplicadas a construção e manutenção, com foco em detecção de falhas e monitoramento estrutural. Resultados sugerem que abordagens similares podem ser eficazes em aeronáutica, especialmente para inspeções em tempo real. Similarmente, (ZHAO; TIAN, 2023) destacou, embora o foco do trabalho seja a construção civil, como técnicas de processamento de imagem e visão

computacional podem ser adaptadas para inspeções em aeronaves. Resultados mostraram maior confiabilidade em detecções automatizadas.

No que tange às principais tecnologias observadas no contexto de hangares de manutenção, destaca-se que o uso de realidade aumentada aumenta a precisão em tarefas complexas e reduz custos associados à manutenção (SAFI *et al.*, 2019). Foram obtidos resultados, melhorias no aprendizado e na retenção de conhecimento técnico por parte dos operadores. Em (Sá; ZACHMANN, 1999), são apresentadas tendências que mostraram a redução de erros e aumento da eficiência em até 40% ao serem utilizadas aplicações de realidade virtual para a verificação de processos de montagem de manutenção. O emprego de realidade virtual foi destacado como ferramenta para simular cenários perigosos sem risco para operadores. Cabe destacar que, para o uso das tecnologias descritas, o emprego de técnicas de visão computacional se torna de fundamental implementação.

### **2.3 Considerações finais do capítulo**

Ao longo deste capítulo, foram abordados os assuntos necessários para a compreensão deste trabalho, visando a ampla compreensão do leitor à metodologia aqui explorada.

Foi feito um enfoque no que tange às principais tecnologias referentes ao anteparo tecnológico aqui explorados, com as respectivas definições, e buscou-se uma conceituação básica para que alguém não familiarizado com hangares de manutenção aeronáutica compreenda o ambiente descrito.

Por fim, ressalta-se que o capítulo que aqui se encerra objetivou a conceituação dos subsídios técnicos para a resolução da problemática do problema de pesquisa definido na introdução, cuja resolução será abordada no capítulo seguinte.

## 3 Materiais e Métodos

O presente trabalho explora a aplicação de técnicas de visão computacional para a captura de ações e tarefas em um hangar de manutenção aeronáutica. Para a concretização do trabalho, faz-se um aprofundamento e a concomitante inspeção das ferramentas computacionais mais recentes no que tange ao aprendizado de máquina (principalmente *deep learning*). Além disso, o escopo do projeto é respaldado em linhas de pesquisa das seguintes frentes:

- Engenharia de fatores humanos;
- Engenharia logística; e
- Transporte aéreo.

Adicionalmente, salienta-se que a investigação rigorosa da literatura acadêmica possibilita a criação de uma fundamentação teórica que viabiliza a redução de testes de soluções tecnológicas no âmbito do trabalho aqui desenvolvido, por meio da filtragem das metodologias mais adequadas, de forma em que sejam abordadas estratégias mais assertivas.

### 3.1 Materiais

#### 3.1.1 Requisitos técnicos

Para a execução do projeto, foram mapeados os seguintes requisitos técnicos:

- Deve-se ter uma ferramenta de modelagem tridimensional capaz de reproduzir uma simulação de um hangar de manutenção aeronáutico.
- Deve-se conseguir capturar as imagens dessa simulação
- A simulação deverá conter os seguintes elementos, necessariamente: aeronave, pessoas (operadores), carros, guindastes, escadas, plataformas elevatórias e sinalizadores.

- Os elementos mínimos presentes na simulação deverão estar em movimento.
- Deve-se conseguir processar as imagens da simulação, utilizando algum algoritmo de visão computacional, capaz de classificar os objetos de interesse na imagem.
- A classificação de objetos deverá ser feita em tempo real, ou seja, em um parâmetro de tempo rígido (na ordem dos milissegundos).
- Deve-se fazer a implementação de um agente autônomo.
- Como *output*, devem ser fornecidas as imagens das câmeras com reconhecimento de objetos, pessoas (operadores), tarefas, carros, guindastes, escadas, plataformas elevatórias e sinalizadores e reconhecimento de agente autônomo presente na simulação.

### 3.1.2 Aparato tecnológico

Como forma de cumprir os requisitos levantados na seção 3.1, além de atingir aos objetivos propostos na seção 1.4.1, serão utilizadas as seguintes ferramentas:

- *Unity* e *Unreal*, como ferramentas usadas para a simulação, paralelamente e não complementar entre si, para serem exploradas as simulação e detecções em diferentes parâmetros, explorando as características técnicas que diferenciam ambas as plataformas
- Dentre os prováveis cenários de simulações utilizados, optou-se pelo cenário “*Aircraft maintenance hangar*”, desenvolvida por “Roy Sousa” e disponibilizada para venda na *Epic Games Store*, por cumprir a todos os requisitos levantados. Após a compra, todos os direitos foram obtidos e nenhuma licença da simulação é aplicável ao projeto. O valor de obtenção foi de R\$302.78, convertido para reais.
- Como forma de unificar a simulação trabalhada no *Unity* e no *Unreal*, para serem feitas comparações sólidas entre ambos, foi utilizado o pacote “*Exporter for Unreal to Unity 2024*”, adquirido na loja “*Unity Assets Store*” por \$49.99. Esse pacote pode transferir os objetos e texturas do Unreal para o Unity com possibilidade de variação de parâmetros.
- Pacote *OpenCV for Unity*, disponível na *Unity Assets Store* por \$95, que possibilita a utilização do *openCV* no Unity.
- Algoritmo de detecção de objetos em tempo real YoloV8.

- O pacote *NPC AI Engine — Dialog, actions, voice and lipsync — Convai*, que ao longo deste relatório será simplificado como *Convai*, para a programação de elementos representativos dos operadores de aeronaves.
- O pacote *YOLOv8WithOpenCVForUnityExample*, disponível gratuitamente em <https://github.com/EnoxSoftware/YOLOv8WithOpenCVForUnityExample/blob/master/README.md>
- O conjunto de dados COCO (Objetos Comuns em Contexto) é um dataset em grande escala destinado a tarefas de detecção, segmentação e legendagem de objetos em imagens.
- A plataforma *Roboflow*, usada para a anotação de elementos de imagens obtidas por meio de vídeo no hangar de simulação trabalhado.
- A plataforma *Google Colab*, usada para o treinamento de elementos do hangar de manutenção considerados importantes para o trabalho e que não estavam disponíveis no COCO.
- Os programas *IClone* e *Character Creator* para a importação no *Unity* de animações pré-estabelecidas.

## 3.2 Metodologia

### 3.2.1 Simulação do ambiente

Para a simulação do hangar aeronáutico, foram cogitadas duas abordagens distintas, que serão descritas nos parágrafos seguintes.

A primeira possibilidade de prototipagem da simulação consiste na utilização de cenários pré-criados com viés de uso em jogos de entretenimento ou *design* arquitetônico. A fonte para a obtenção deste cenário poderia ser diversa, desde lojas oficiais do *Unity* e do *Unreal*, até lojas disponíveis para modelistas gráficos venderem os seus produtos. Convém mencionar que o cenário adquirido deve estar em formato adequado para poder ser usado nos programas mencionados. Posteriormente, são feitas as animações necessárias no *Unity* ou no *Unreal* para uma aproximação de uma simulação real.

A segunda abordagem idealizada para a execução deste trabalho, no que tange à simulação do ambiente, foi pensada com o propósito de atender às demandas existentes caso a primeira abordagem não fosse viável. Consiste no dimensionamento de um hangar aeronáutico para as aeronaves dos manuais aeronáuticos obtidos e a sua posterior prototipagem, usando-se, para isso, o *software SketchUp*, empregado para a criação de mode-

lagens 3D de estruturas arquitetônicas, design de interiores e projetos de engenharia. Por meio dessa ferramenta, pode-se fazer a modelagem do hangar aeronáutica, que poderá ser exportado posteriormente como um modelo de extensões .FBX ou .OBJ, que poderá ser importado no *Unity* ou no *Unreal* (entretanto, cabe destacar que, até esse ponto, tem-se, somente, um hangar aeronáutico contendo as paredes, pisos, teto, escritórios e mobília interna). Os passos seguintes consistem em se buscar os pacotes necessários contendo modelos de aeronaves, carros de ferramentas e outros maquinários, além de modelos tri-dimensionais das pessoas (operadores do hangar). Posteriormente, trabalham-se com as animações dos elementos para que o hangar se aproxime cada vez mais da simulação de um ambiente real.

### 3.2.1.1 Exportação de cenário desenvolvido no *Unreal* para o *Unity*

Constitui uma necessidade desse projeto de pesquisa que a simulação trabalhada em ambos os programas (*Unity* e *Unreal*) sejam as mesmas. Em um viés logístico, entende-se que a construção da simulação de um ponto inicial nulo, nos dois programas, é inviável considerando o escopo de tempo desse projeto. Dessa forma, o uso de ferramentas capazes de dinamizar esse processo se torna imprescindível. Para esse fim, foi considerado o pacote “*Exporter for Unreal to Unity 2024*”, capaz de converter os projetos do *Unreal Engine* para o *Unity*, adquirido na *Unity Assets Store*, sendo um *plugin* do *Unreal* que dele exporta níveis completos (ou seja, converte em lotes malhas estáticas com LODs, decalques, sondas de reflexão, luzes, malhas *spline*, terrenos, formas de colisão, materiais, texturas, posicionamento de atores e malhas esqueléticas com animação).

Essa ferramenta possibilita a conversão do cenário variando as configurações mostradas na figura 3.2.

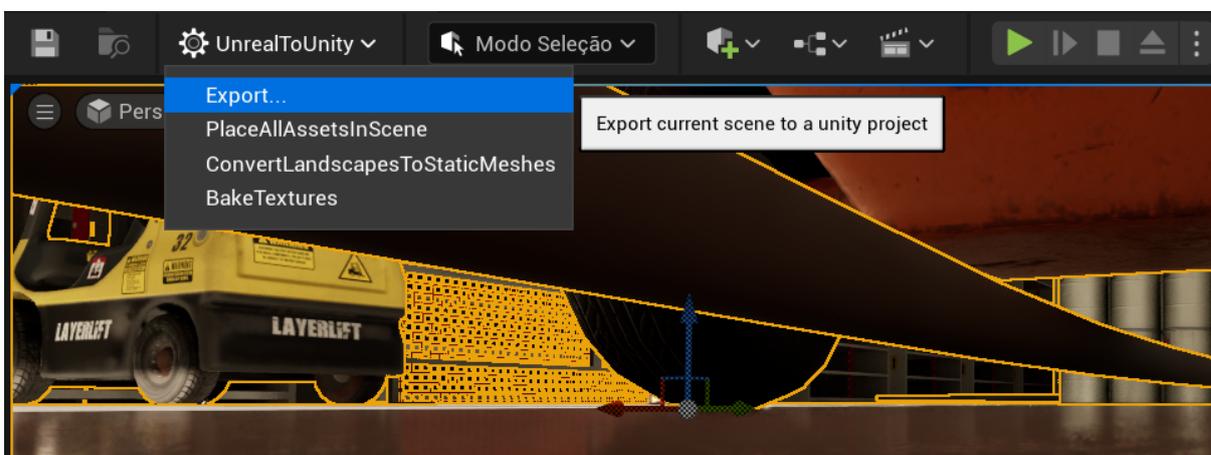
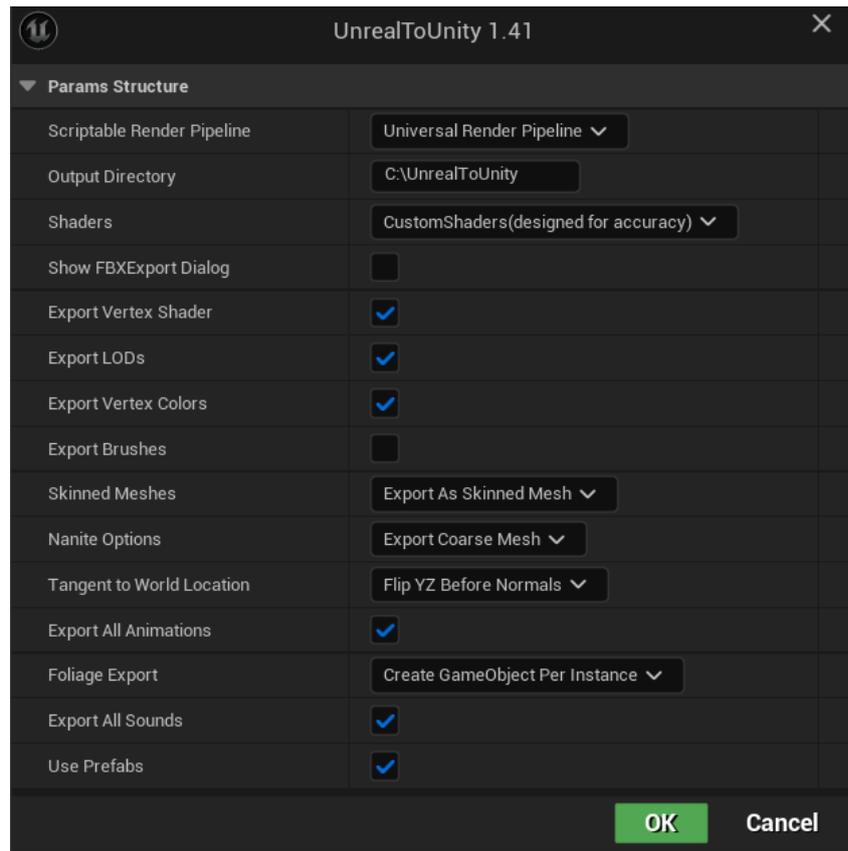


FIGURA 3.1 – Tutorial de acionamento da ferramenta *Unreal to Unity*, versão 1.41.

FIGURA 3.2 – Configurações de exportação da ferramenta *Unreal to Unity*, versão 1.41.

Nas configurações mostradas na figura 3.2, a *pipeline* de renderização automatizável por *scripts* pode variar como *Universal Render Pipeline*, *Built In Render Pipeline* e *High Definition Render Pipeline*. A *Built In Render Pipeline* é a *pipeline* de renderização padrão do *Unity*, oferecendo um sistema de renderização flexível capaz de ser personalizado por *shaders* escritos manualmente. Entretanto, não é otimizada para o aproveitamento de tecnologias modernas de renderização e pode ser menos eficiente em termos de desempenho quando comparada às *pipelines* mais recentes. Por outro lado, a *Universal Render Pipeline* foi projetada para o oferecimento de um equilíbrio entre qualidade gráfica e desempenho, sendo ideal para projetos que precisam rodar em múltiplas plataformas (incluindo dispositivos móveis, consoles e PCs). Por fim, o *HD Render Pipeline* é destinado a projetos que exigem gráficos de alta fidelidade, sendo ideal para aplicações em *hardware* de alto desempenho.

O projeto aqui descrito foi executado em um único computador pessoal *Notebook* que possui o seguinte modelo, configurações e características:

- Modelo *Acer Predator Helios 300*
- *Overclockable Geforce GTX 1660Ti*
- Processador Intel® Core™ i7-9750H CPU @ 2.60GHz 2,59 GHz

- RAM instalada 16,0 GB (utilizável: 15,8 GB)
- Sistema operacional de 64 bits
- Sistema operacional instalado em um disco local de 500 GB SSD. Além disso, possui um disco de 128 GB SSD e um HD de 1T.

Cabe destacar que, enquanto realizava a execução do cenário no *Unreal Engine*, foi comum que o sistema entrasse em estado de não responsividade devido a um travamento causado pela aplicação.

Além disso, para a escolha das configurações apropriadas de exportação, foram feitas todas as possíveis combinações dentre as renderizações descritas para avaliação de desempenho no *Unity*. Menciona-se, também, que outras modificações em relação à exportação padrão foram: acionamento da opção de exportação dos sons do cenário e exportação de todas as animações.

### 3.2.2 Agentes capazes de se deslocarem na simulação

Para a integração de agentes capazes de se deslocarem no ambiente de manutenção, traçando caminhos lógicos e condizentes com a física, duas etapas foram consideradas cruciais: o uso do pacote *Convai*, o qual é uma ferramenta de inteligência artificial conversacional capaz de ser integrada ao *Unity* que permite a criação de personagens virtuais e uma *NavMeshSurface*, sendo um componente usado para personalizar a área de navegação (*NavMesh*) onde os agentes de inteligência artificial podem se locomover em um ambiente tridimensional, permitindo a criação de caminhos possíveis nos quais os personagens controlados por IA, como os NPCs (Non-Playable Character).

Para o escopo desse trabalho, optou-se pelo uso inicial de 5 agentes personalizados e importados do *Convai*, que tiveram pontos intermediários de trajetória pré-programados e distintos, porém com variações entre esses pontos intermediários. Para o deslocamento dos agentes descritos (que completam este trabalho simulando os trabalhadores de manutenção aeronáuticos) foi utilizado um algoritmo norteador de movimento, cujos pontos intermediários e funcionamento são demonstrados na figura 3.3.

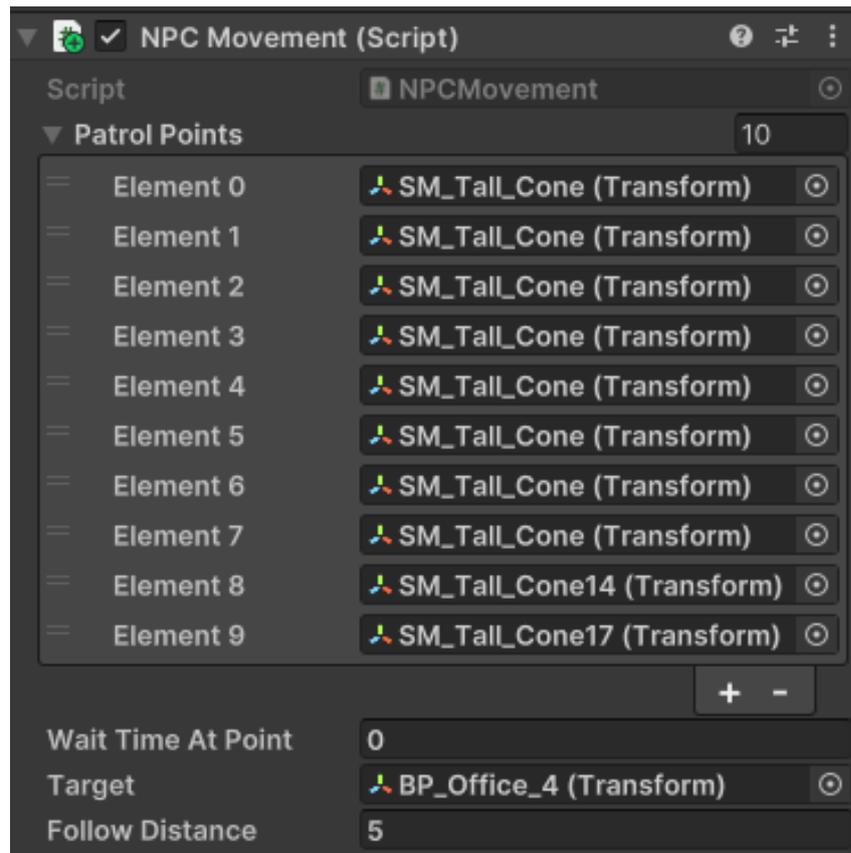


FIGURA 3.3 – Configuração de deslocamento de agentes no hangar

### 3.2.3 Posicionamento estratégico das câmeras

Conforme literatura, o número mínimo de câmeras para o imageamento eficiente de um ambiente são três câmeras. Entretanto, esse valor sofre variações que não são definidas normatizadamente, ao dependerem exclusivamente das características do ambiente mapeado (se existem oclusões, elementos altos, dimensão do lugar e separação entre as câmeras) e das características da câmera.

Dessa forma, não se é, portanto, possível a criação de uma metodologia definidora da quantidade de câmeras mínima capaz de se adaptar para todos os ambientes mapeados, pois estes sofrem variações.

Dessa maneira, no ambiente de estudo, foram usados os seguintes parâmetros para a definição do número de câmeras, em que seria utilizado o menor número capaz de cumprir a todos os requisitos:

- Não possuir oclusões no ambiente de operações do hangar
- Elementos que requerem atenção, tais como tanques de combustível, empilhadeiras e guindastes, devem ter detecção nítida

- Deverão ser consideradas as medidas do hangar e as características da câmera
- As áreas críticas deverão estar na área de captura

Cabe destacar nesta seção que, devido à captura de cada imagem e posterior tratamento com o algoritmo *YOLOv8* ser feita individualmente, quanto maior o número de câmeras, maior a carga de processamento.

### 3.2.4 Uso do *OpenCV* no *Unity*

Para a execução desse projeto, considerou-se necessária a utilização do *OpenCV*, que consiste em uma biblioteca de código aberto dedicada ao desenvolvimento de soluções voltadas à visão computacional e ao aprendizado de máquina. Para este projeto, foram fundamentais os conceitos e funcionalidades que serão descritos nos parágrafos seguintes.

A integração entre o *OpenCV* e o *Unity* foi feita por meio do pacote *OpenCV for Unity*, disponível na *Unity Assets Store*. Cabe destacar que essa biblioteca possui integração com realidade aumentada, realidade virtual e realidade mista que, embora não estejam no escopo desse trabalho, servem de fundamento para eventuais trabalhos futuros. Destaca-se também que possui licença Apache 2, caracterizada por uma maior permissividade e compatibilidade, permitindo maior flexibilidade no uso e distribuição do *software* devendo, porém, serem citados os devidos créditos.

Por meio do *OpenCV*, fez-se a leitura dos vídeos capturados pelas câmeras descritas na seção 3.2.3 para o processamento das imagens obtidas. Cabe destacar que, para o algoritmo funcionar adequadamente, foi necessária a conversão dos espaços de cores do RGB para GBR. Isso foi feito por meio de *scripts* usando a linguagem de programação *C#*.

Após a integração da biblioteca, foram feitas diversas variações entre conversões de espaço de cor e aplicação de filtros, objetivando-se uma melhor detecção de bordas.

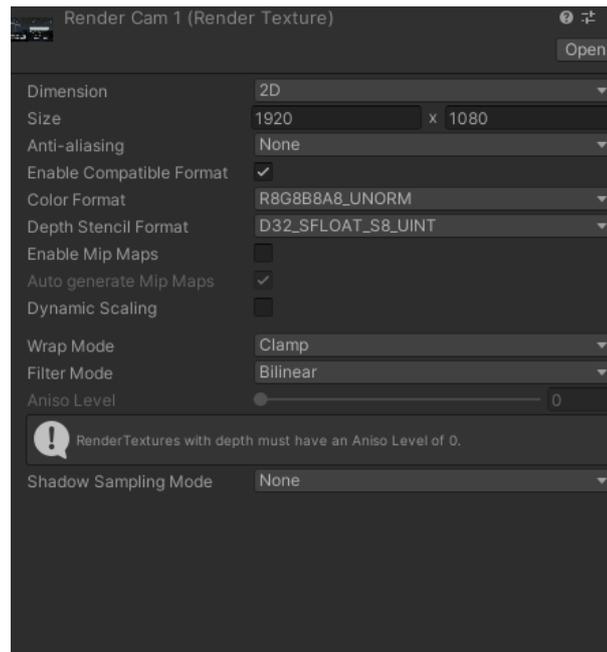


FIGURA 3.4 – Configurações de padrões de cores para estudo de detecção de bordas

### 3.2.5 Integração do *Unity* com o YOLO e o *OpenCV*

A etapa seguinte consiste na integração do *Unity* com o YOLO e o *OpenCV*, necessário para o reconhecimento de objetos. Inicialmente, foi feita a integração do *OpenCV* com o *Unity* e as versões mais antigas, porém consolidadas, V2 e V3 do YOLO. Entretanto, devido ao resultado insatisfatório obtido por esta metodologia, buscaram-se novas formas de se fazer o reconhecimento de objetos, uma vez que: os objetos não eram detectados instantaneamente e a detecção era falha.

A integração do YOLO ao *Unity* teve como objetivo uma detecção de objetos capaz de ser realizada tanto em um ambiente de simulação quanto em uma posterior integração à câmeras em ambientes reais.

Cabe destacar que para o teste do reconhecimento de objetos, foi utilizado o YOLO treinado com o banco de dados COCO, o qual é um dataset de reconhecimento de imagens em larga escala, desenvolvido para tarefas de detecção de objetos, segmentação e geração de legendas, composto por mais de 330.000 imagens, em que existem anotações de 80 categorias de objetos e cinco legendas descritivas da cena. Entretanto, ressalta-se um desafio deste projeto: uma vez que essas imagens são de ambientes distintos ao hangar de manutenção virtual do *Unity*, algumas informações e detecções possuem alto potencial de perda.

Dentre os algoritmos de visão computacional descritos na seção 2.1.4, após estudo das metodologias vigentes até o momento e descrições de modelos resultantes na sequência da integração com o *Unity* (além da possibilidade de mudança das câmeras virtuais por

câmeras reais após a condução deste trabalho, fator considerado um novo requisito), optou-se pela tentativa de integração com os modelos mais recentes do *YOLO*. Os modelos mais recentes foram comparados segundo a literatura, e o resultado foi disposto na tabela 3.1.

Modelo YOLO	Velocidade (Quadros por segundo)	Acurácia (mAP)
YOLOv6	60	0,75
YOLOv7	50	0,80
YOLOv8	70	0,85

TABELA 3.1 – Comparação dos Modelos YOLO por Velocidade e Acurácia

Pelo desempenho demonstrado na literatura, optou-se pelo modelo *YOLOv8*, através da integração do pacote *YOLOv8WithOpenCVForUnityExample*. Para o uso do pacote, foram feitas as configurações de imageamento necessárias para a captura eficiente da câmera para posterior detecção de objetos. Além disso, o pacote mencionado possibilita a entrada como *input* de qualquer modelo YOLOv8 treinado, em uma extensão *.onnx*, e do arquivo *.names* contendo as classes de detecção.

Após a configuração do ambiente e das configurações de imageamento, foi possível a detecção das classes “aeronave” e “pessoa”.

### 3.2.6 Treinamento de imagens para detecção de objetos típicos de hangares de manutenção inicialmente não reconhecidos

Como citado na seção 3.2.5, o modelo treinado com o banco de dados COCO integrado ao YOLO, *OpenCV* e ao *Unity* conseguiu identificar as pessoas e a aeronave do hangar. Entretanto, alguns elementos são imprescindíveis para que se possa fazer uma leitura inicial eficiente dos hangares e não são detectados inicialmente, sendo:

1. Empilhadeiras
2. Tanques de combustível
3. Gerador
4. Guindaste
5. Rebocador
6. Carrinhos de ferramentas

Como os elementos citados não são englobados no treinamento descrito, torna-se necessário um treinamento personalizado capaz de identificar as classes descritas no ambiente

mencionado. Entretanto, o processo de classificação e treinamento de imagens é extremamente trabalhoso e, em geral, feito por uma equipe de pessoas, uma vez que a qualidade do treinamento feito é proporcional à quantidade e qualidade de imagens que alimentam o modelo e, em geral, são recomendadas que se tenham centenas ou milhares de imagens.

As imagens devem ter diferentes ângulos, iluminações e contextos para garantir que o modelo aprenda a generalizar.

O treinamento de um modelo é feito da seguinte forma: é feito o *upload* de um banco de dados de imagens (quanto maior o número de imagens, melhor o modelo) e um arquivo .txt contendo as coordenadas da envoltória de cada objeto e a classe que a ele se atribui (ou seja, com caixas delimitadoras indicando cada objeto). Em seguida, essas imagens são divididas entre: imagens de treinamento (Cerca de 70–80% das imagens, usado para ajustar o modelo.), imagens de validação (Cerca de 10–15% das imagens, usado durante o treinamento para ajustar hiper parâmetros e evitar *overfitting*) e imagens de teste (Cerca de 10–15% das imagens, usado para avaliar o desempenho final do modelo após o treinamento).

Para facilitar o processo de classificação manual das imagens para a criação do modelo descrito, foi utilizada a plataforma *Roboflow*. Inicialmente, fez-se o treinamento de 24 imagens para a identificação das empilhadeiras e dos guindastes. Essas imagens receberam anotações conforme as classes contidas no *Roboflow*. O treinamento inicial foi feito da seguinte forma: as 24 imagens foram divididas em 17 imagens de treinamento, 5 imagens de validação e 2 imagens de teste. Essas imagens iniciais foram processadas com parâmetros como giro, brilho e saturação para que se tornassem 58 imagens, das quais: 51 são de treinamento, 5 de validação e 2 de teste. O tempo estimado de treinamento inicial foi de 16 minutos.

O passo seguinte consistiu na utilização da plataforma *Google Colab* para o treinamento das imagens conforme o modelo *YOLOv8*.



FIGURA 3.5 – Passo para anotação de guindaste na plataforma *Roboflow*



FIGURA 3.6 – Exibição das imagens anotadas, com as classificações de guindastes e empilhadeiras, na plataforma *Roboflow*

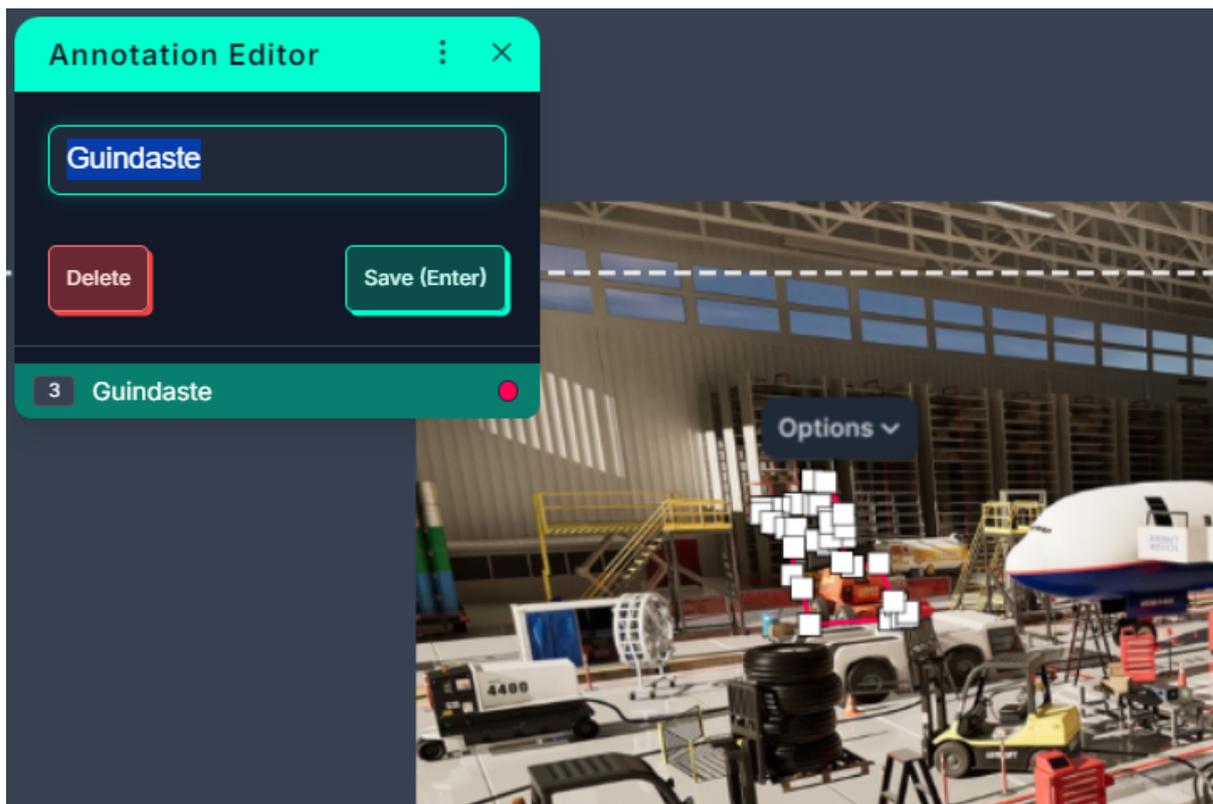


FIGURA 3.7 – Exibição de maneira em que é feita a etiquetagem do guindaste na imagem, na plataforma *Roboflow*

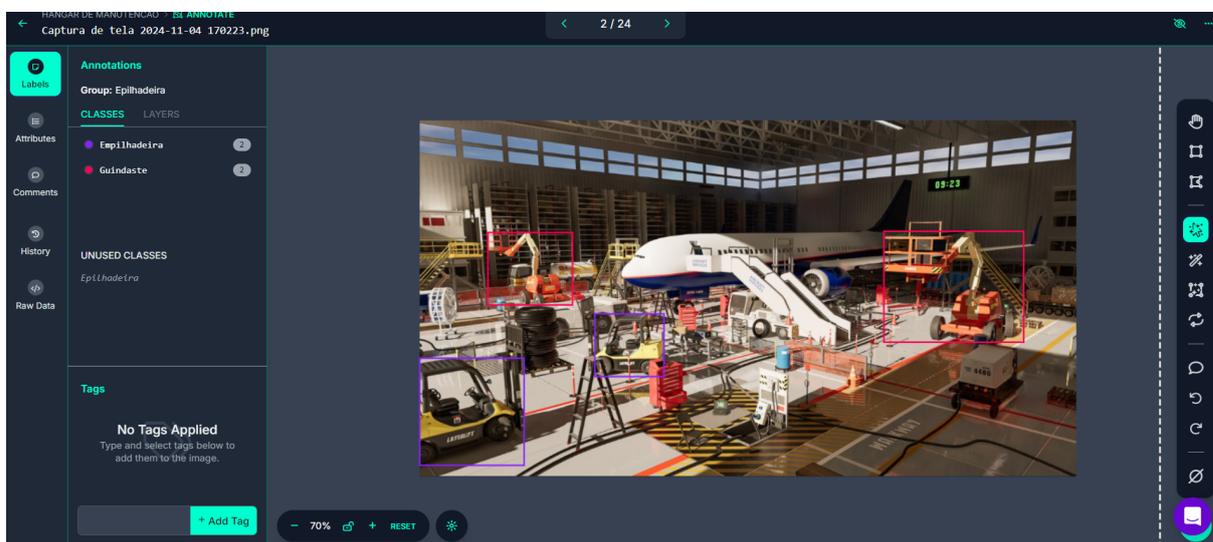


FIGURA 3.8 – Padrão de anotação de imagens na plataforma *Roboflow*

Posteriormente, foi feita uma automatização do processo de anotação para aumento do banco de dados: criou-se um novo projeto e foi feita a captura de um vídeo de um minuto e meio de duração, mostrando ângulos distintos e um panorama do hangar. Escolheu-se divisão de *frames* por minuto, de modo que foram obtidas 2067 imagens no total. Mesmo com automatizações feitas na plataforma, a checagem e correção de cada anotação é feita

manualmente, de maneira que se tornou inviável no escopo deste trabalho. Optou-se pela utilização de uma pequena amostragem dessa captura, contendo cerca de 130 imagens (que receberam anotações manualmente) e houve o seguinte aumento de dados:

- Saídas por exemplo de treinamento: 5
- Rotação de 90°: Sentido horário, anti-horário, de cabeça para baixo  
Brilho: Entre 0% e +69%
- Espelhamento: Horizontal, Vertical
- Desfoque: Até 0,9 px

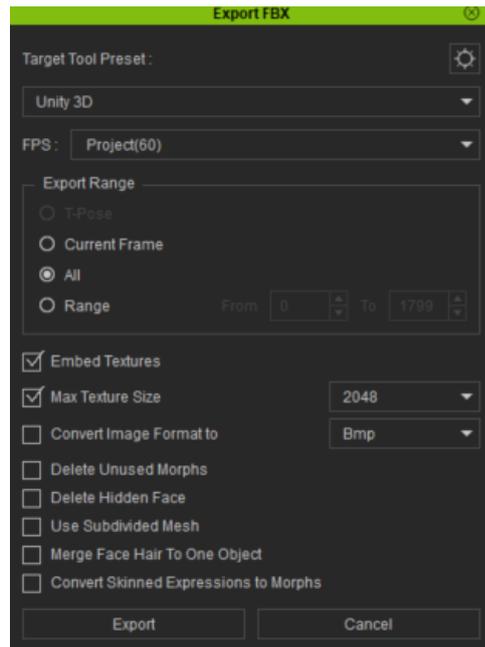
O dataset obtido foi treinado de duas formas: usando o *YOLOv8* no *Google Colab* e na plataforma *Roboflow* usando o modelo “Roboflow 3.0 Object Detection (accurate)” para fins comparativos. O treinamento feito no *Google Colab* retorna um arquivo .pt, convertido para .onnx e usado como *input* no *Unity*. O arquivo contendo as classes possui extensão .names, sendo feito manualmente.

### 3.2.7 Animação de agentes conforme rotinas pré-estabelecidas

Para a animação dos agentes conforme rotinas não estabelecidas pelo *Convai*, foram adquiridas as animações pertinentes na plataforma *Reallusion*. Essas animações foram as seguintes: movimento de pintura (para a pintura de aeronaves e peças altas) e o movimento de carregamento de objeto pesado nos ombros.

Para a implementação das animações no agente *Convai* do *Unity*, foram feitos os seguintes procedimentos. Inicialmente, utilizou-se o *Software IClone* para a implementação das animações pertinentes em um agente temporário. Feito essa etapa, fez-se a exportação das animações no formato .rlMotion. A etapa seguinte consistiu em se adquirir um novo agente temporário e importar as animações, anteriormente salvas no *Character Creator*.

As animações foram relacionadas ao agente temporário e o modelo foi exportado como um arquivo de formato .fbx, que poderá ser posteriormente importado no *Unity*.

FIGURA 3.9 – Configuração de exportação de modelo .fbx do *Character Creator*

Consecutivamente, faz-se a importação do modelo de formato .fbx no *Unity* e deve-se fazer a configuração do modelo importado em um avatar relacionando os ossos e membros do agente temporário aos parâmetros.

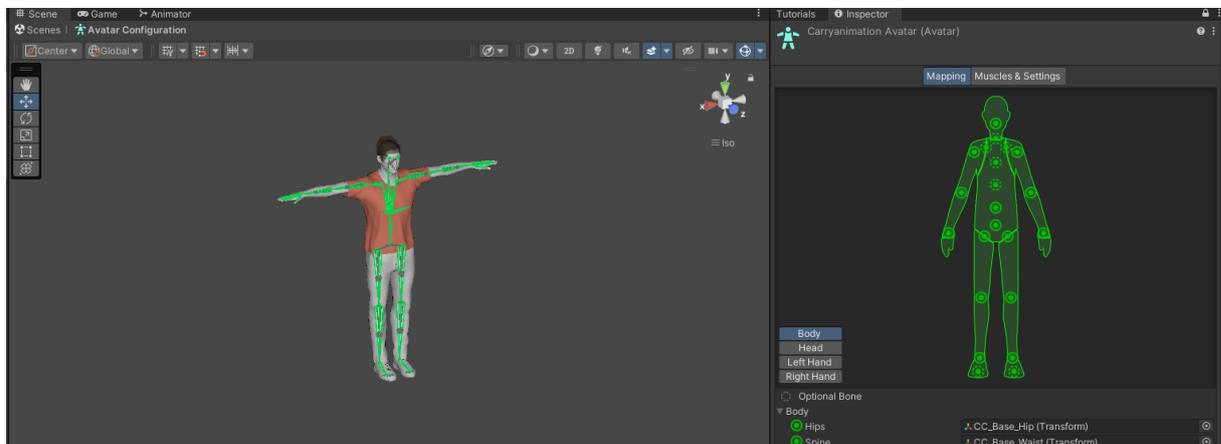


FIGURA 3.10 – Configuração de avatar

Feita a configuração, as animações importadas com o avatar temporário poderão ser relacionadas diretamente aos agentes do *Convai*, por meio da implementação da ação ao *Animator Controller* do agente que se deseja atribuir essa animação.

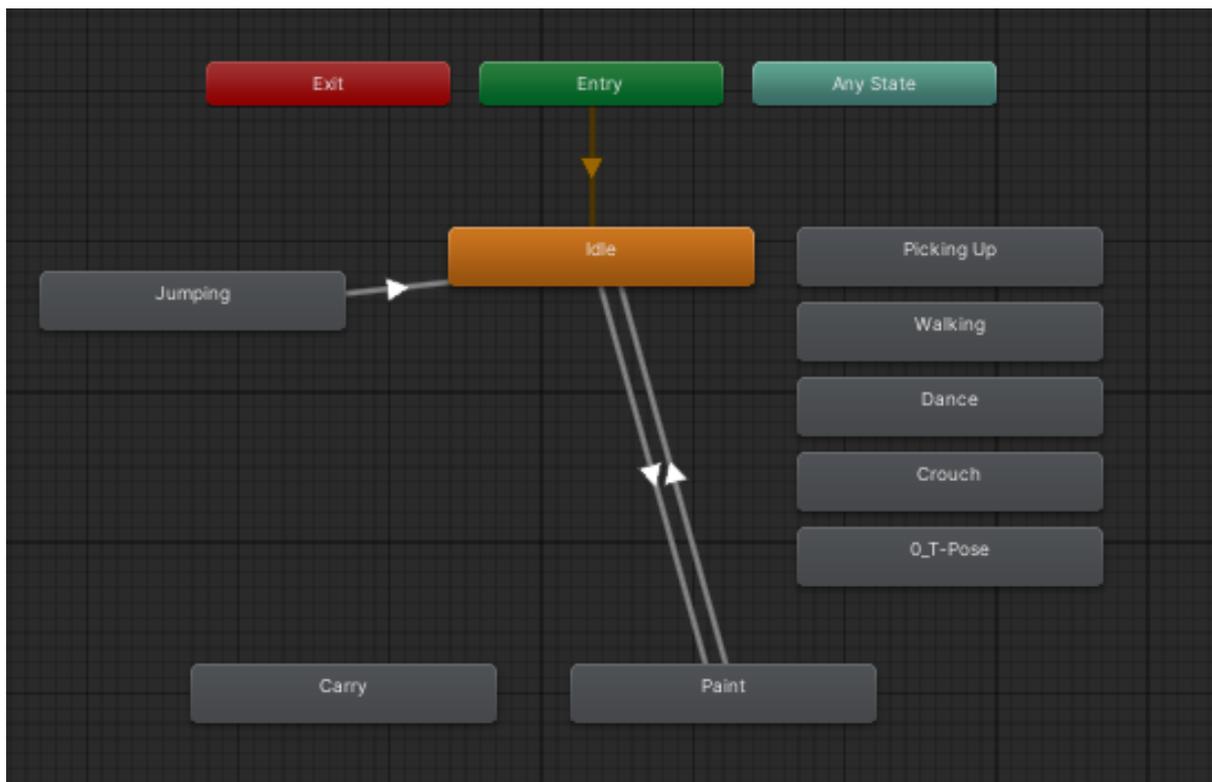


FIGURA 3.11 – Exemplificação de *Animator Controller* implementado com a função “pintar”.

As ações pretendidas resultantes em cada agente foram programadas por meio de *scripts*.

# 4 Apresentação dos resultados e discussão

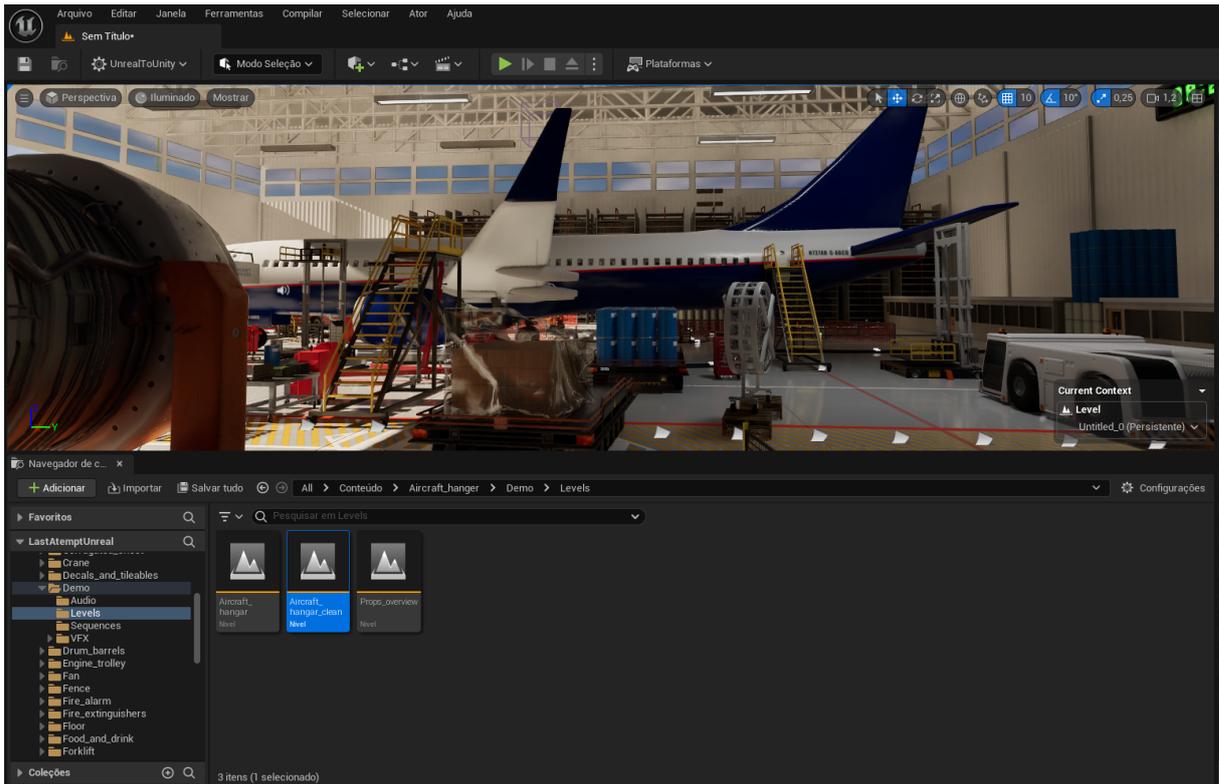
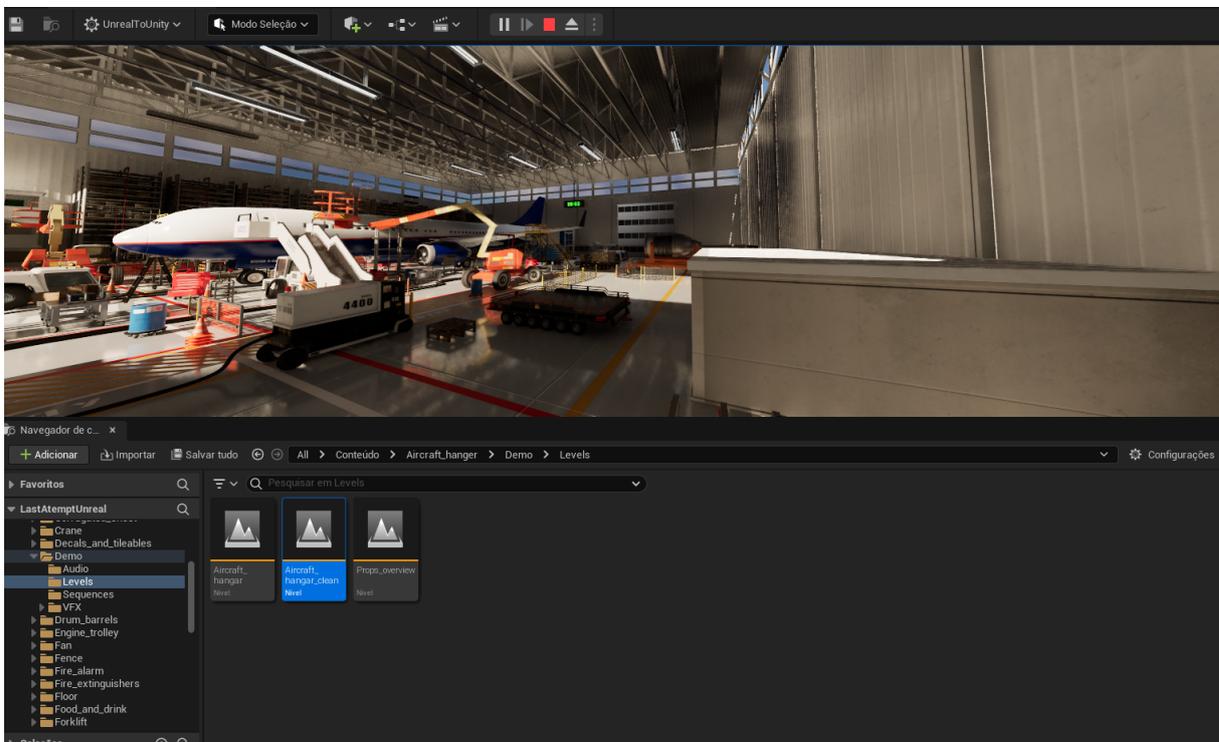
## 4.1 Ambiente concebido

Conforme descrito na seção 3.1.2, optou-se pela utilização do cenário criado por “Roy Sousa”, para fins de jogos recreativos, disponibilizado para venda na *Epic Games Store*. Inicialmente, o cenário descrito foi desenvolvido para uso no *Unreal*. Foi constatado que dispunha da construção do hangar em escala, presença de aeronaves e carros de combustível, detalhes no ambiente como cones, luzes e outras sinalizações, guindastes e plataformas elevatórias. Para uso no *Unreal*, notou-se a necessidade de animação de aeronaves e carros de combustível, além da importação de operadores com as devidas movimentações. Constatou-se que as plataformas elevatórias já continham movimentação vertical pré-programada.

Após exportação do cenário no *Unity*, conforme descreve a seção 3.2.1.1, verificou-se a demanda de animação dos guindastes, plataformas elevatórias, operadores, carros de combustível e aeronaves.

Por fim, salienta-se que, após a implementação do cenário descrito, foram exploradas todas as possibilidades de ações a serem executadas no local.

As figuras 4.1 e 4.2 mostram visões distintas do cenário exibido no *Unreal*.

FIGURA 4.1 – Visão do hangar obtida pelo *Unreal*FIGURA 4.2 – Visão do hangar obtida pelo *Unreal*, imagem capturada durante simulação do hangar

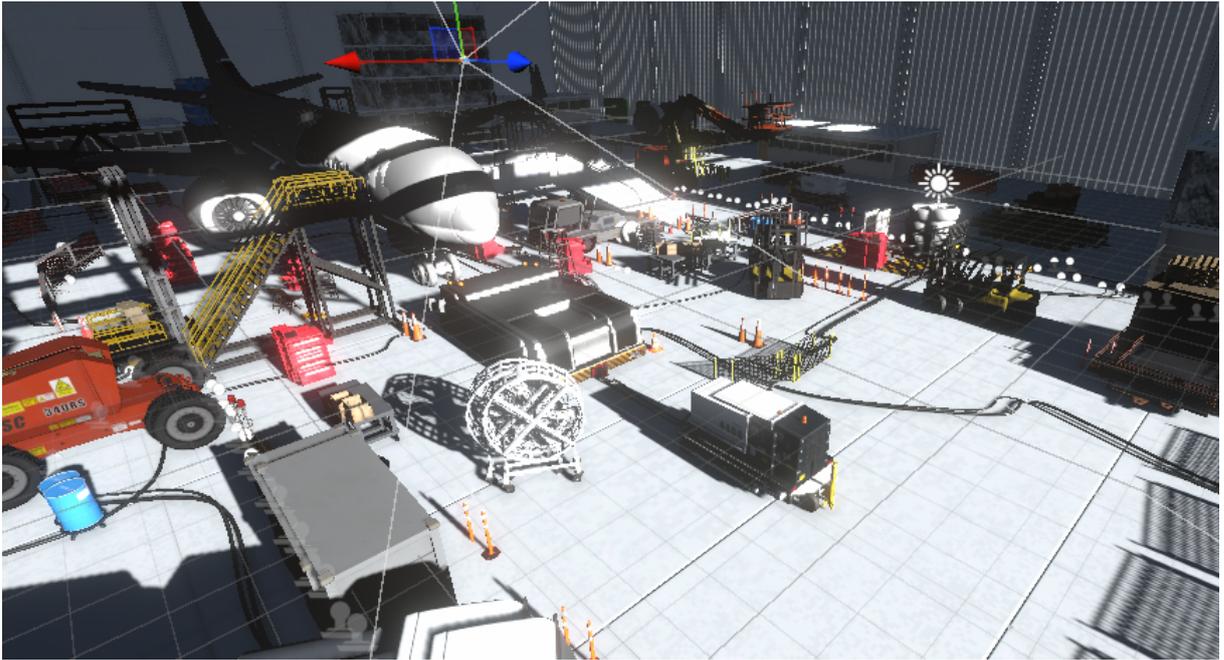


FIGURA 4.3 – Visão do hangar obtida pelo *Unity*, configurações do *Unity* conforme descrito na seção 3.2.1.1.

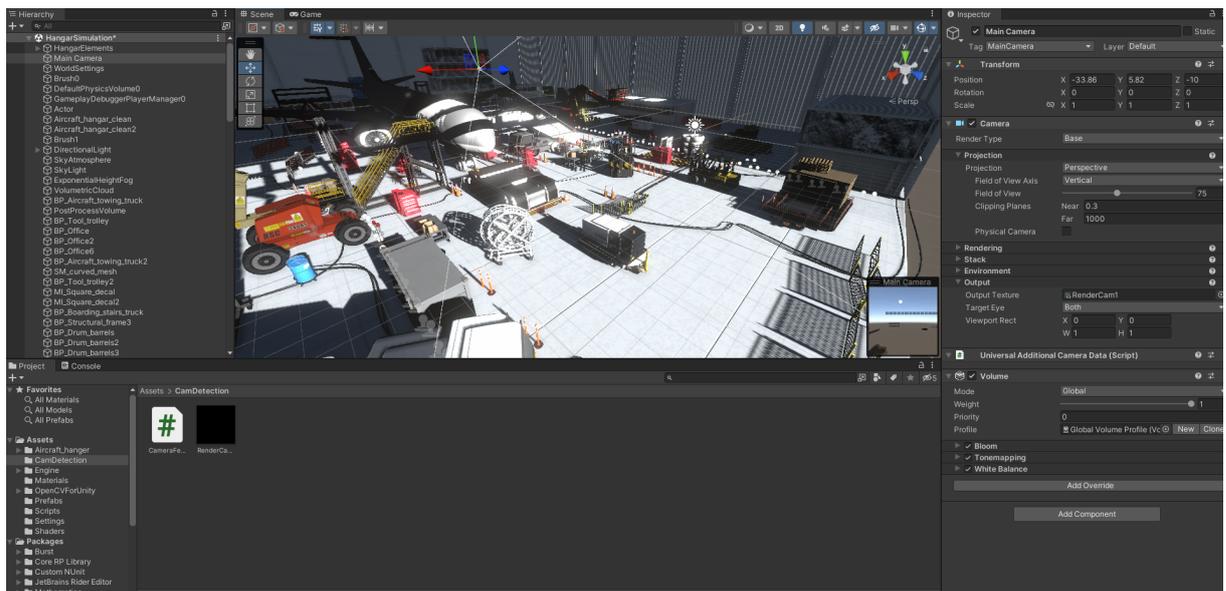


FIGURA 4.4 – Visão geral do cenário no *Unity*, configurações conforme descrito na seção 3.2.1.1.

Ao se variarem as configurações de exportação descrito na seção 3.2.1.1, constataram-se os seguintes fatos, para o sistema descrito para esse projeto: não responsividade crítica durante execução da aplicação usando o *HD Render Pipeline*. Entretanto, notou-se qualidade gráfica similar à apresentada pelo *Unreal*, ou seja, consideravelmente realística, característica considerada importante em termos de pesquisa em visão computacional. No *Universal Render Pipeline*, notou-se queda de qualidade gráfica considerável em relação à simulação do *Unreal*, conforme notado nas figuras 4.2 e 4.3. Entretanto, a velocidade

de execução do cenário foi notavelmente maior. A exportação em *Built In Render Pipeline* provou-se pior em todos os aspectos comparativamente às outras duas renderizações, sendo considerada irrelevante para os propósitos dessa pesquisa.

Dessa forma, optou-se pela utilização, no *Unity*, por critérios de configuração gráfica e velocidade de execução, no uso do *Universal Render Pipeline*. Destaca-se, entretanto, que, embora se tenha optado por fazer uma exportação contendo as animações e os sons do ambiente, ambos não funcionaram adequadamente após a exportação em todas as configurações testadas, devendo serem posteriormente adequadas para o propósito do projeto.

#### 4.1.1 Integração de agentes autônomos com o cenário

Após a aplicação do *NavMeshSurface*, obteve-se a superfície em azul dada pela figura 4.5.



FIGURA 4.5 – Resultado de aplicação de *NavMeshSurface* na superfície do hangar, em azul.

Ressalta-se da superfície obtida que: algumas superfícies *outliers*, acima do plano do chão, foram obtidas, como a superfície superior à aeronave, ao caminhão e ao rebocador. Entretanto, como não foram gerados caminhos ligando a superfície do chão com as superfícies superiores, estas foram tidas como isoladas e chegou-se à conclusão de que não irão influenciar o movimento dos agentes autônomos.

Para a configuração dos agentes, optaram-se por parâmetros que fossem considerados casos críticos para detecção (por exemplo, não contrastando no ambiente devido às cores de vestimentas). Dessa forma, programou-se o agente na plataforma do *Convai* segundo a figura 4.6. Foram utilizadas duplicatas dos agentes em toda a simulação, não se optando

por serem feitas variações fenotípicas devido às limitações do plano gratuito da plataforma.



FIGURA 4.6 – Agente usado como parâmetro na simulação.

Após a aplicação do algoritmo que norteia a movimentação do agente, verificou-se comportamento compatível com o esperado.

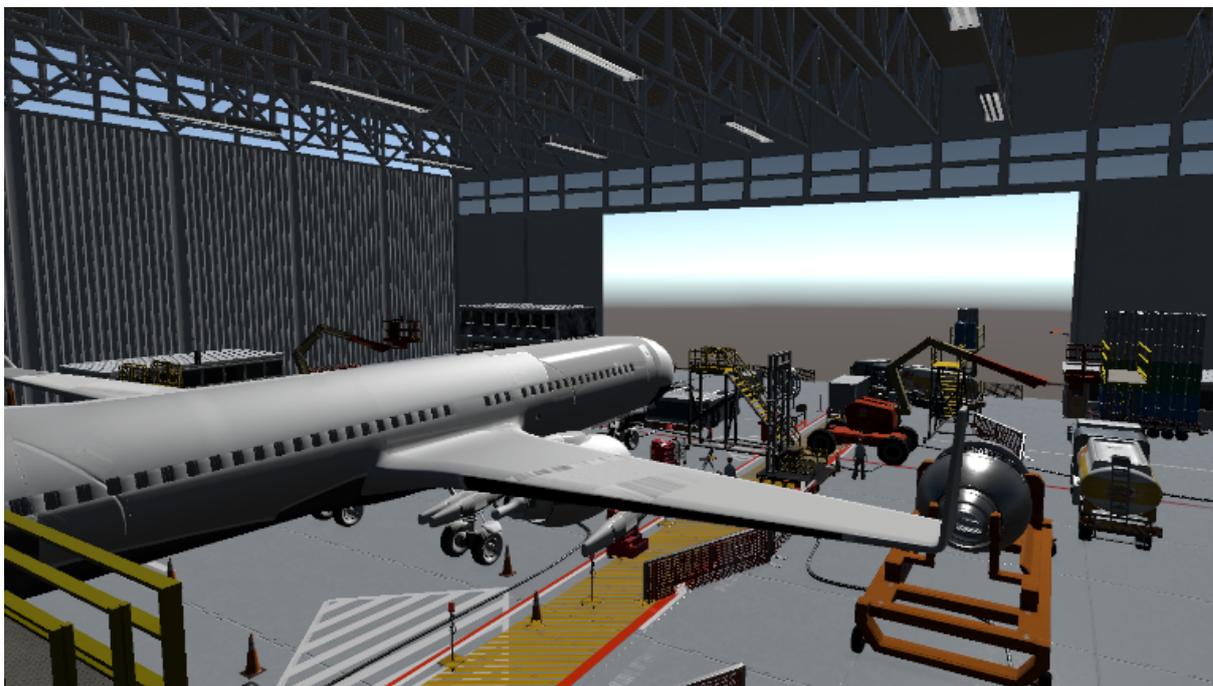


FIGURA 4.7 – *Unity* no modo *Play*, com cada agente (pessoas em proporção) espalhadas e simulando suas devidas tarefas.

Após a utilização de agentes autônomos na simulação, obteve-se uma simulação apta aos propósitos destinados, com futuras modificações e animações de objetos feitas conforme demanda.

## 4.2 Imageamento

Para cumprir os requisitos dados pela seção 3.2.3 e manter mínimo o número de câmeras, optou-se pela utilização de três câmeras no hangar de manutenção, dispostas inicialmente conforme as figuras 4.8, 4.9 e 4.10.

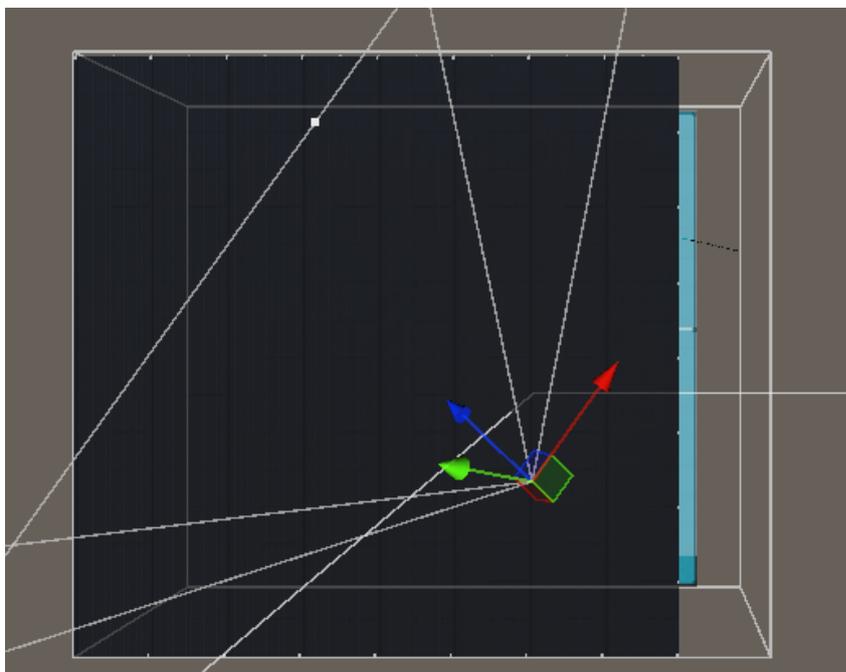


FIGURA 4.8 – Configuração inicial da primeira câmera.

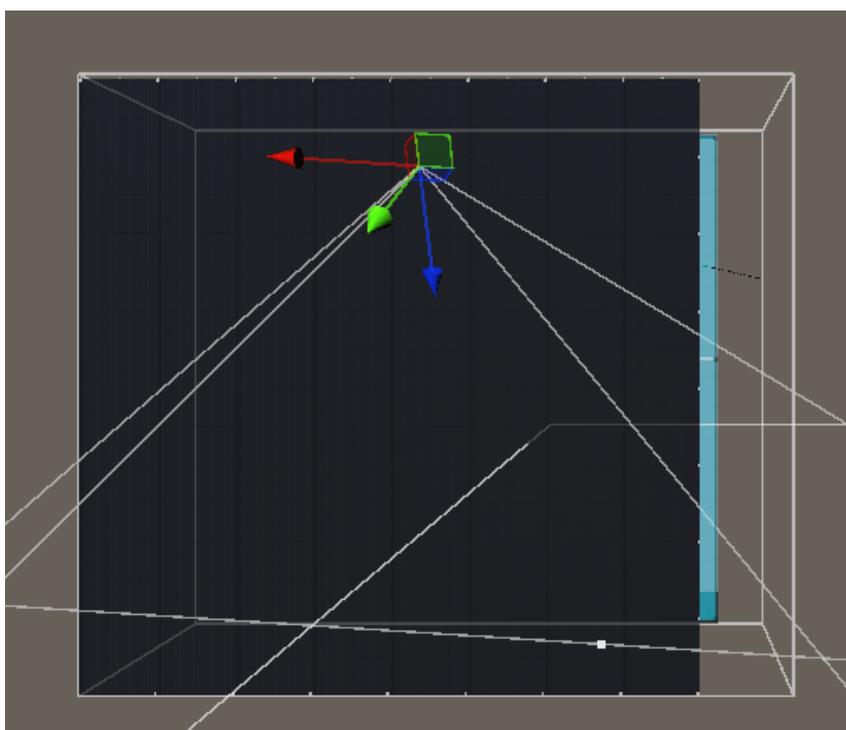


FIGURA 4.9 – Configuração inicial da segunda câmera.

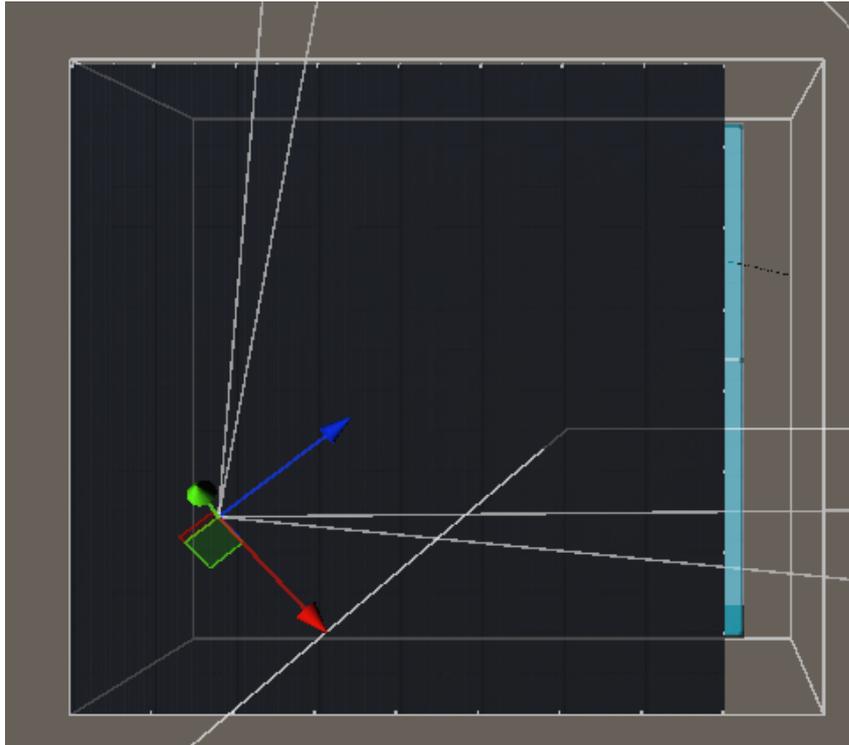


FIGURA 4.10 – Configuração inicial da terceira câmera

A configuração do posicionamento das câmeras sofrerá modificações conforme a escrita deste projeto, entretanto, a triangulação exibida se manterá sempre constante.

### 4.3 Aplicação do *OpenCV* no *Unity*

Para este trabalho, foi desenvolvido um algoritmo detector de bordas que inicialmente obteve resultado insatisfatório. Em seguida, foram observadas as variações de aplicações de filtros e modificações de padrões de cores para verificação do impacto dessas modificações no objetivo de detecção deste trabalho. A gradação dos resultados obtidos foram dispostos nas figuras 4.11, 4.12, 4.15, 4.13 e 4.14.



FIGURA 4.11 – Aplicação do algoritmo identificador de bordas inicial, sem tratamento de imagem.



FIGURA 4.12 – Refinamento do algoritmo detector de bordas, mas ainda sem o tratamento da imagem.



FIGURA 4.13 – Uso do algoritmo detector de bordas atualizado e resultado obtido após processamentos de imagem, exibindo-se o resultado mais satisfatório.



FIGURA 4.14 – Foram feitos novos refinamentos, notando-se o aparecimento de novas bordas que são consideradas importantes para o modelo. A imagem exibe o resultado final.



FIGURA 4.15 – Vista de como é feita a exibição da imagem processada.

Dentro no hangar de manutenção, dispõe-se um quad, o qual é um plano, no qual aponta uma câmera e exibe o resultado do processamento. Dessa forma, são necessárias ao menos duas câmeras virtuais para essa simulação: uma câmera processada pelo OpenCV e pelo algoritmo detector de bordas, sendo refletida no quad, e uma câmera que aponta para o quad e mostra para o usuário o que o quad está exibindo.

#### 4.4 Integração entre *OpenCV*, *YOLO* e *Unity*

Após a integração entre o *OpenCV*, *YOLO* e *Unity*, foi possível a detecção de objetos das classes “aeronave” e “pessoa”.



FIGURA 4.16 – Detecção da aeronave do hangar em voo.

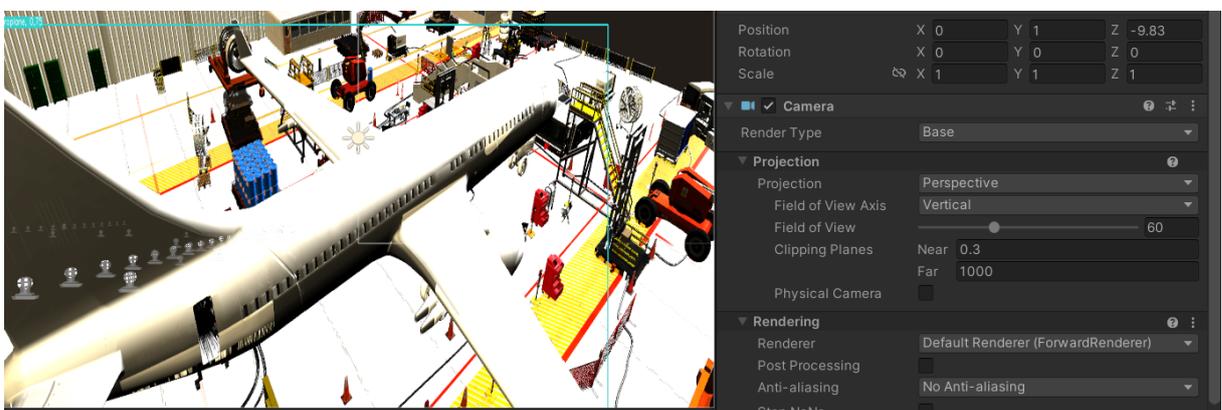


FIGURA 4.17 – Demonstração de detecção de aeronave, em tempo real, no hangar de manutenção.

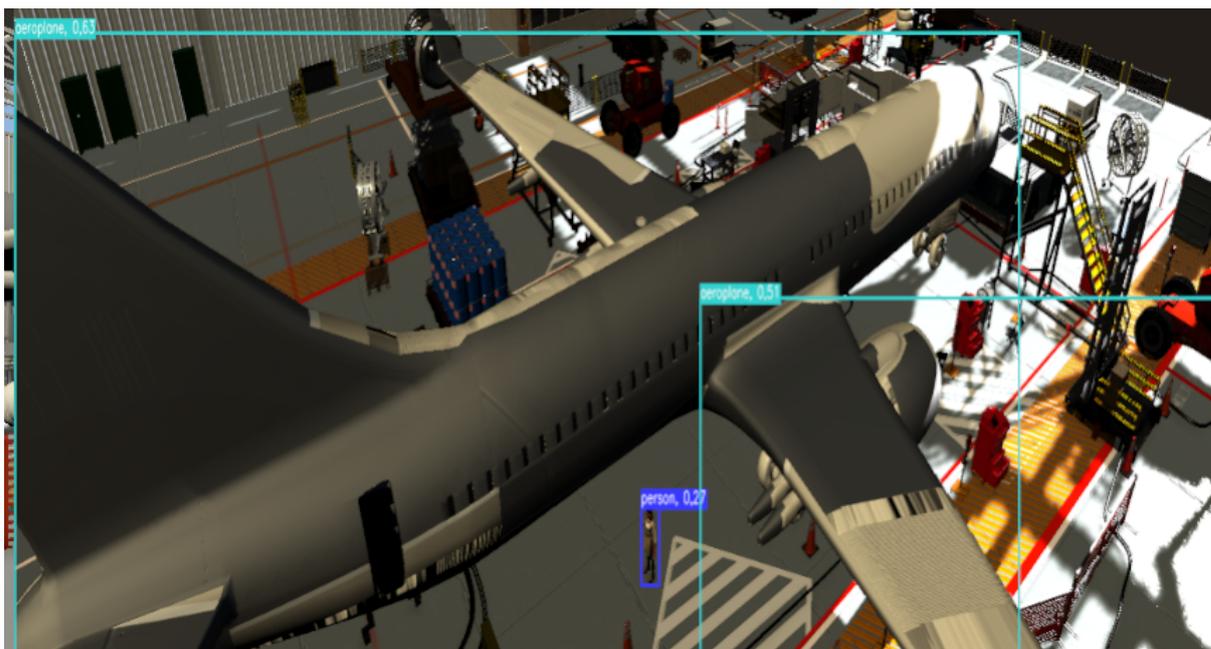


FIGURA 4.18 – Detecção de operador (classificado como pessoa) e aeronave em hangar de manutenção.

Ressalta-se que as imagens mostram diferentes níveis de iluminação durante simulação, para averiguação dos melhores parâmetros de detecção. Cita-se, também, que embora neste trabalho busquemos uma detecção em tempo real, e essa detecção seja descrita como alcançada em alguns trechos, esse termo não deve ser lido literalmente neste trabalho e na literatura de uma forma geral, uma vez que se trata de uma abstração impossível de ser alcançada (deveríamos ter precedimentos instantâneos e isso não existe, sempre haverá um tempo de execução, por menor que seja).

Foi feita uma configuração de 4 câmeras dispostas em painéis, cada uma com um processamento distinto usando o YOLOv8.

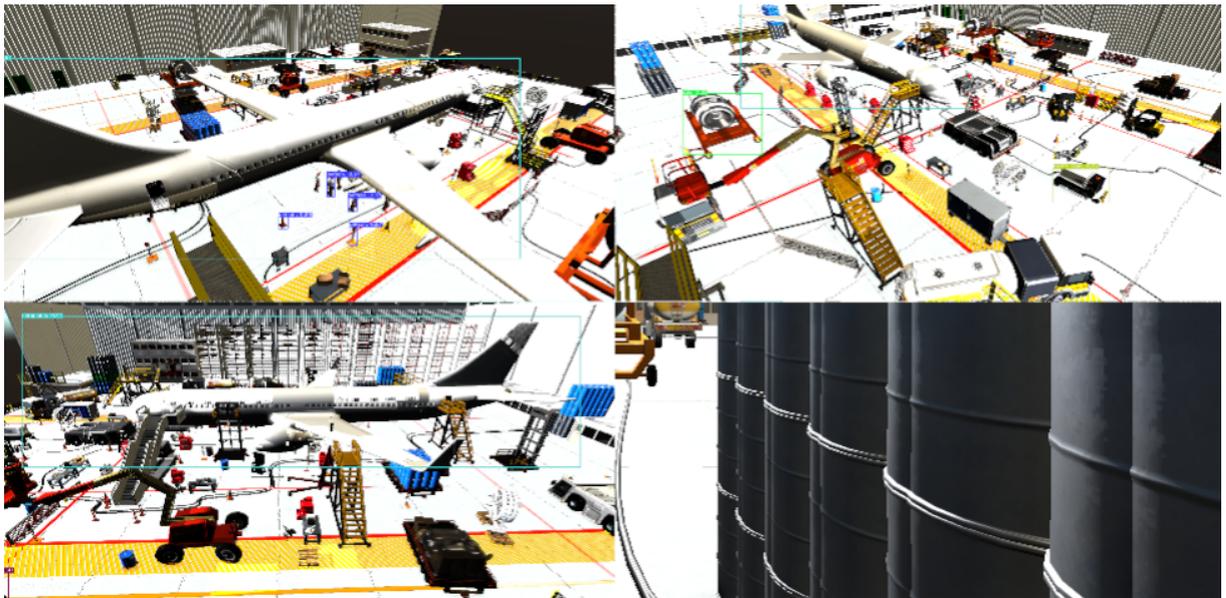


FIGURA 4.19 – O nível de acurácia da detecção foi variável conforme luminosidade.

A representação esquemática do funcionamento dessa metodologia pode ser vista na figura 4.43.

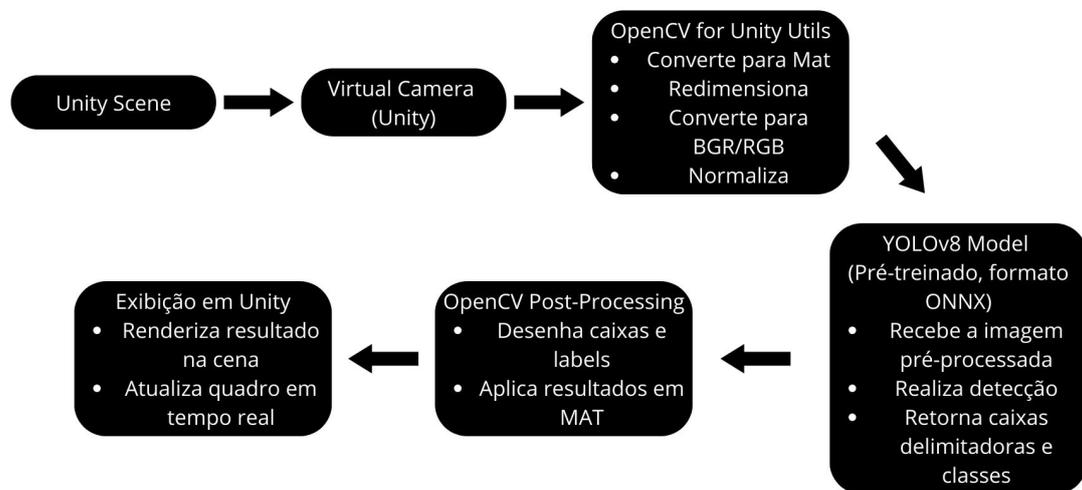


FIGURA 4.20 – Representação esquemática do funcionamento do algoritmo *YOLO* no *Unity* com o *OpenCV*.

Nessa solução, a imagem capturada é redimensionada para o tamanho de entrada esperado pelo YOLOv8, garantindo que a imagem corresponda ao formato de entrada do modelo para uma detecção precisa. Em seguida, o OpenCV faz a conversão de cores se a imagem capturada estiver em um formato diferente do aceito pelo *YOLO*. Por fim, como

o YOLOv8 geralmente espera que os valores de *pixel* estejam normalizados entre 0 e 1, os valores de *pixel* são divididos por 255 para padronizar a imagem de entrada, auxiliando o modelo a processar a imagem corretamente. Após o pré-processamento, a imagem ajustada é enviada ao modelo YOLOv8 para inferência. A detecção é então realizada, e os resultados (caixas delimitadoras, classes) são desenhados e exibidos na cena do Unity.

## 4.5 Treinamento de imagens usando o *Roboflow* e o *Google Colab*

Após o treinamento das imagens no *Google Colab*, obtiveram-se resultados satisfatórios no treinamento das imagens.



FIGURA 4.21 – Representação do funcionamento do algoritmo YOLO no Unity com o OpenCV.

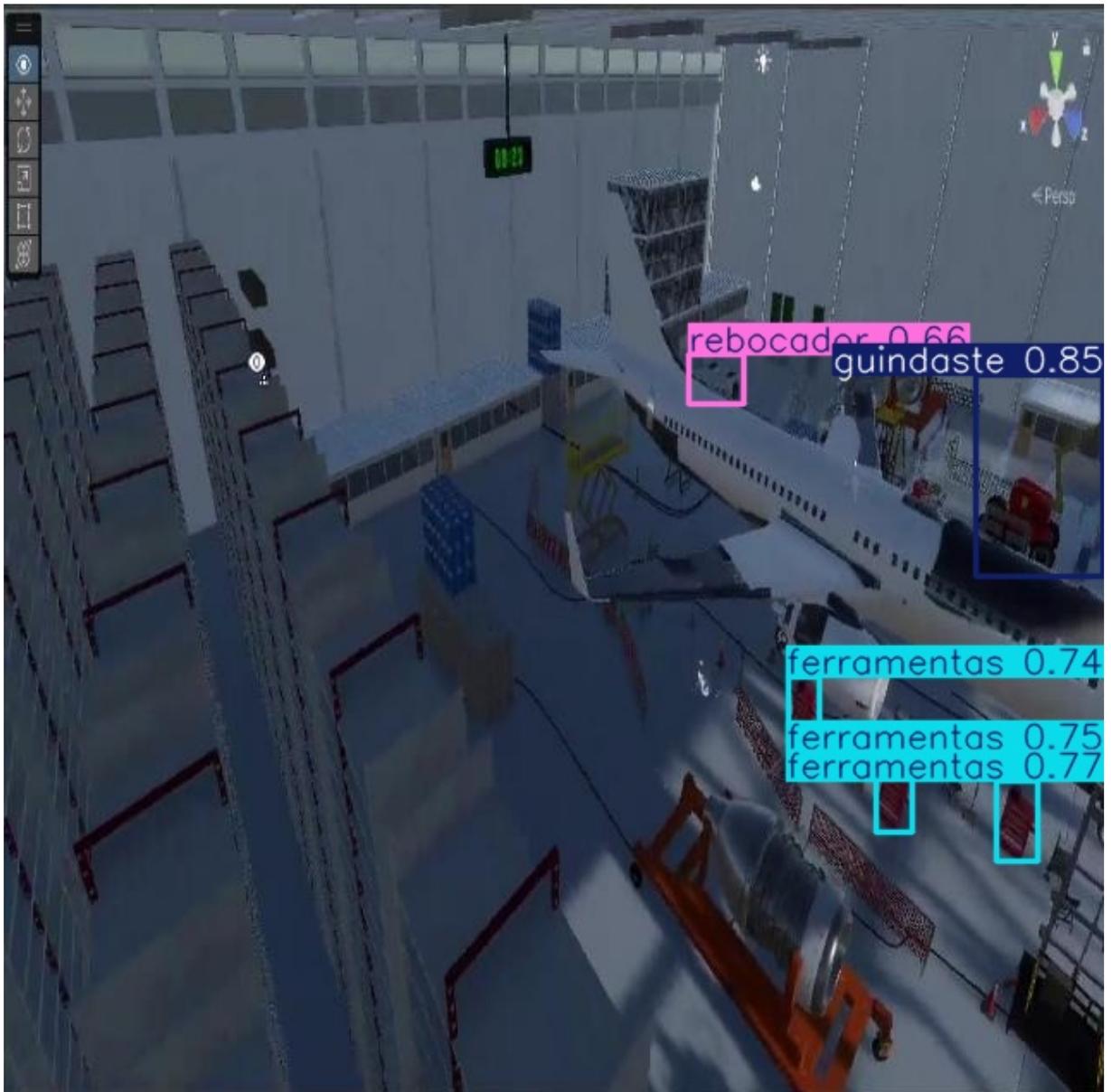


FIGURA 4.22 – Representação do funcionamento do algoritmo *YOLO* no *Unity* com o *OpenCV*.



FIGURA 4.23 – Representação do funcionamento do algoritmo *YOLO* no *Unity* com o *OpenCV*.

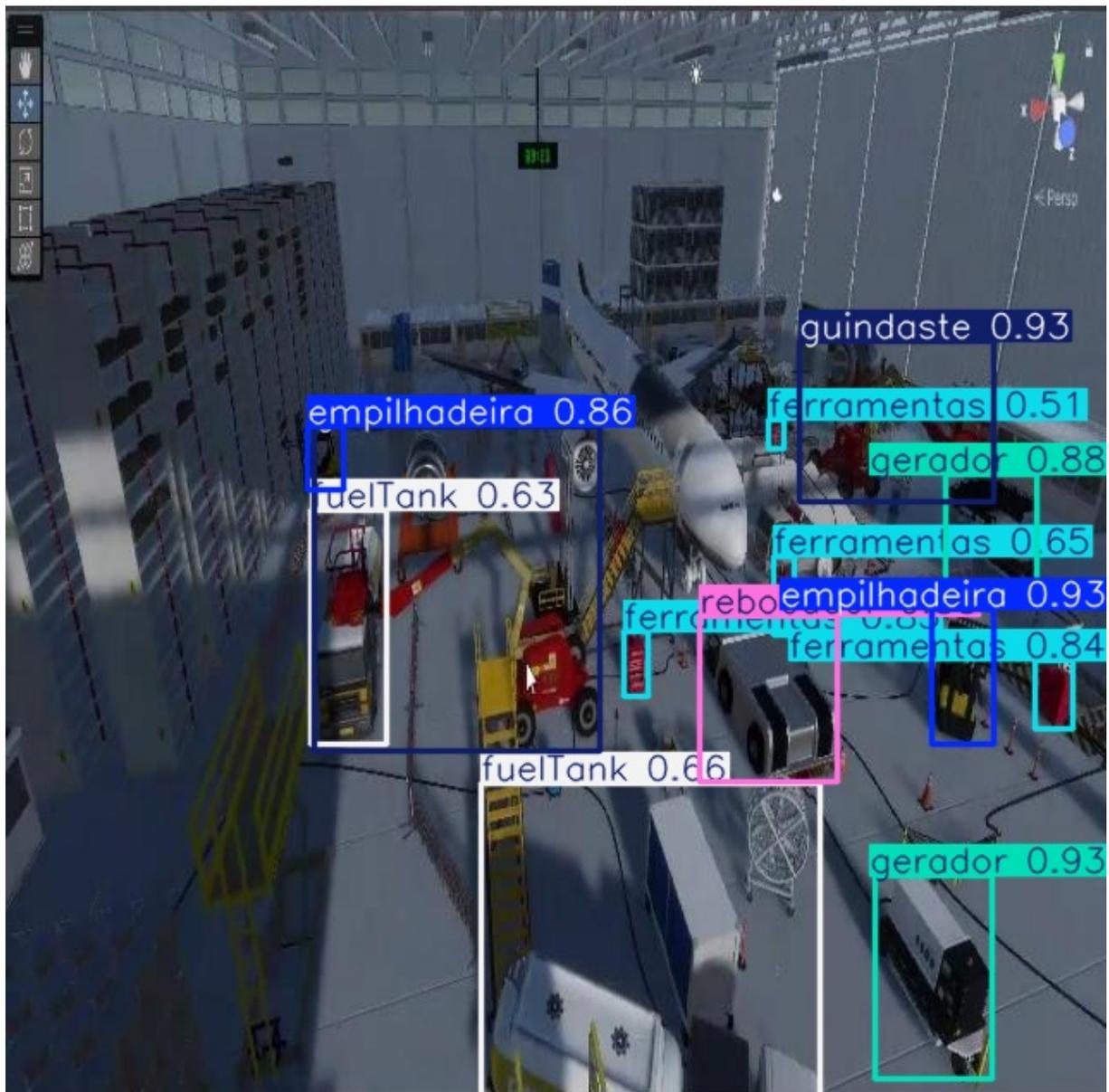


FIGURA 4.24 – Representação do funcionamento do algoritmo *YOLO* no *Unity* com o *OpenCV*.

## 4.6 Estudo de caso

O presente estudo de caso foi realizado para mapear tarefas e identificar objetos críticos em um hangar de manutenção aeronáutica simulado, usando para isso visão computacional. A finalidade principal é a compreensão dos fluxos de trabalho e definição de viabilidade de um sistema equivalente em um hangar de manutenção real, para a devolutiva de um painel gerencial contendo informações rápidas sobre os principais elementos de um hangar. Para este fim, foi utilizada a metodologia descrita na seção 3.

Para a execução deste projeto, foram definidas 10 classes, das quais se têm 7 objetos e 3 ações/tarefas. Os objetos são:

- aeronave
- combustível
- empilhadeira
- ferramentas
- gerador
- guindaste
- rebocador

As ações e tarefas são:

- Andando
- Carregando Peso
- Pintando

A escolha das ações buscou a representação de duas criticidades no ambiente de manutenção. A primeira é a lesão no trabalho decorrente do carregamento em excesso e repetitivo de materiais pesados. A segunda é o esforço repetitivo por pinturas de peças grandes, feita de maneira não ergonômica no hangar representado.

Após a animação dos agentes para executarem as ações desejadas conforme o relatado em 3, mantiveram-se as mesmas configurações de câmeras dispostas em 4.2.

Para o treinamento das imagens, visando uma conversão com o menor número possível de imagens (devido às limitações temporais obtidas pelo treinamento de um grande número de imagens), optou-se por fazer a geração de imagens por meio de uma gravação de cerca de um minuto em cada uma das câmeras, totalizando cerca de 3 minutos. Para a formação do *dataset*, foram extraídos entre dois a três *frames* por minuto, totalizando 288 imagens.

Usando a ferramenta de etiquetagem do *Roboflow*, fez-se a etiquetagem manual de todos os objetos, ações e tarefas relatados.

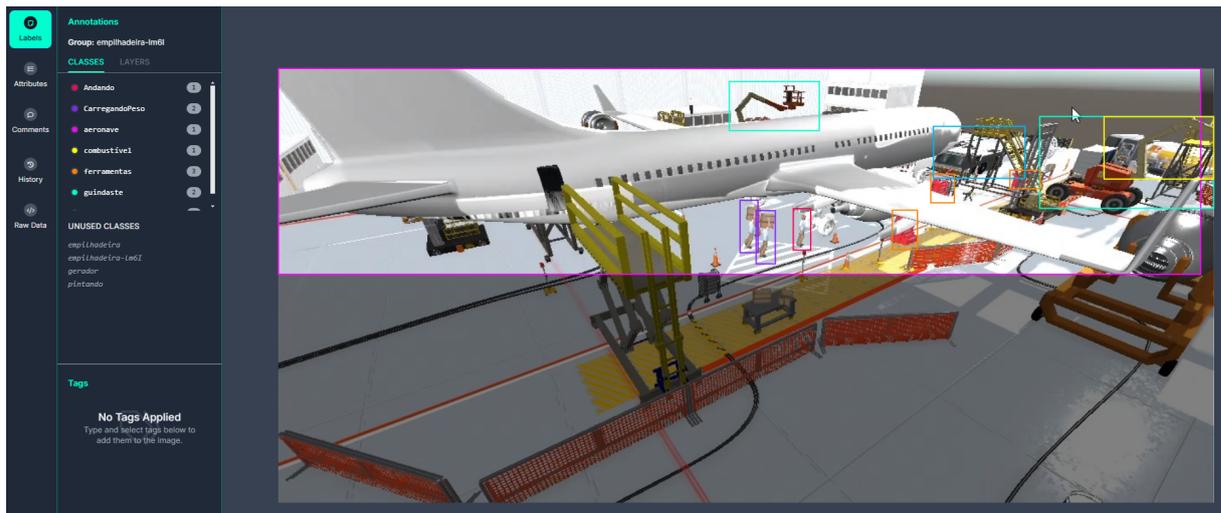


FIGURA 4.25 – Representação de etiquetagem do cenário descrito no *RoboFlow*.

Após a etiquetagem, aumentou-se e configurou-se o banco de imagens da seguinte forma:

- Auto-orientação: aplicado
- Redimensionamento: ajuste (com bordas pretas) em  $640 \times 640$
- Saídas por exemplo de treinamento: 5
- Brilho: entre 15% e +15%

Ao término desse aumento, obtiveram-se 1050 imagens para treinamento. O treinamento do modelo foi executado de duas formas: por meio do *Google Colab*, utilizando-se o YOLOv8, e na própria plataforma *RoboFlow*, usando-se o modelo “*RoboFlow 3.0 Object Detection (Accurate)*”, para fins de comparação. O dataset foi dividido em: 964 imagens para treinamento, 57 imagens para validação e 29 imagens para teste.

O treinamento realizado no *Google Colab* obteve os seguintes resultados:

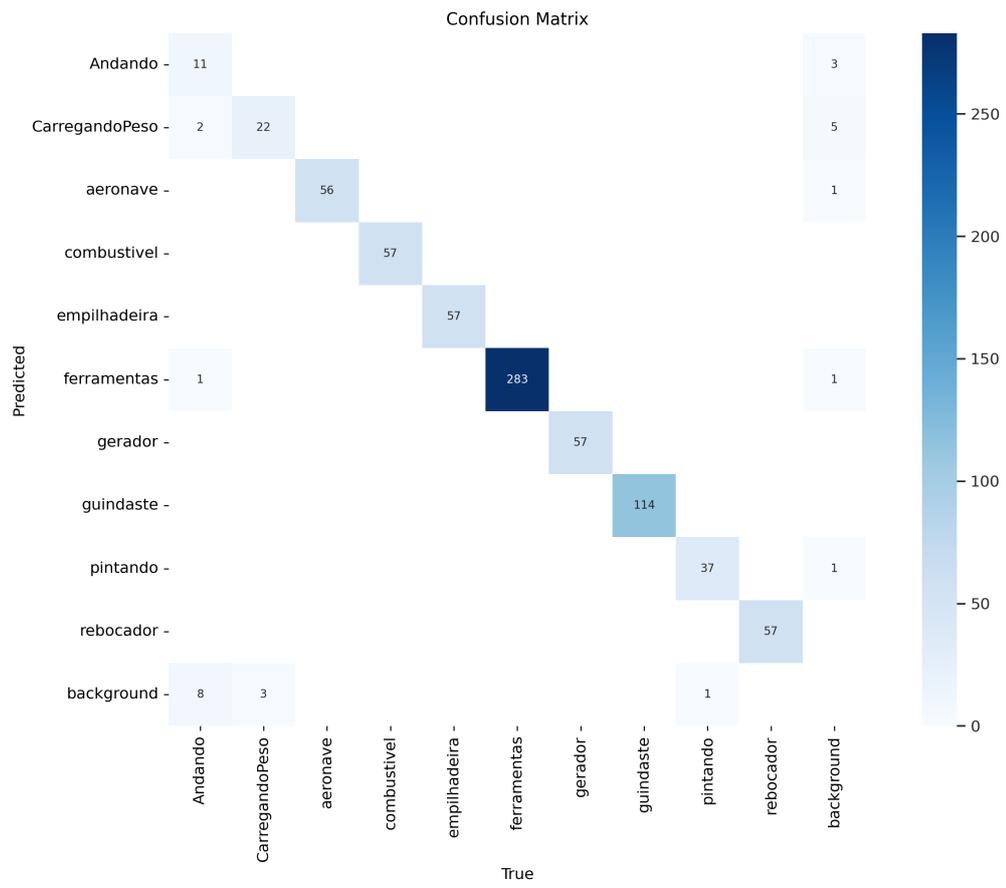


FIGURA 4.26 – Matriz de confusão do modelo treinado no *Google Colab*.

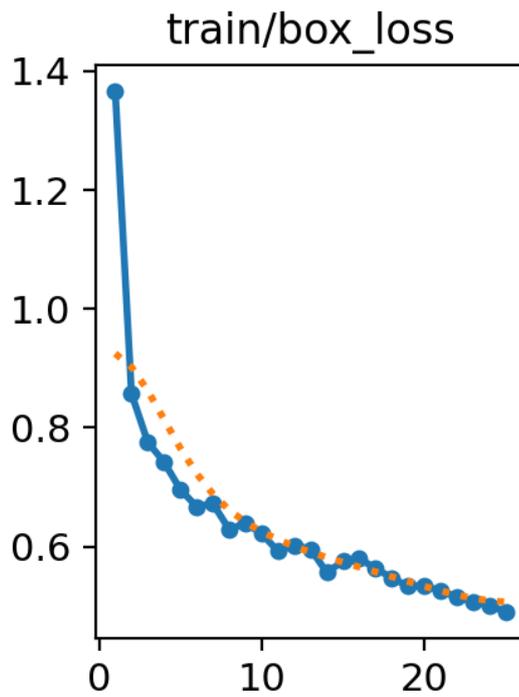
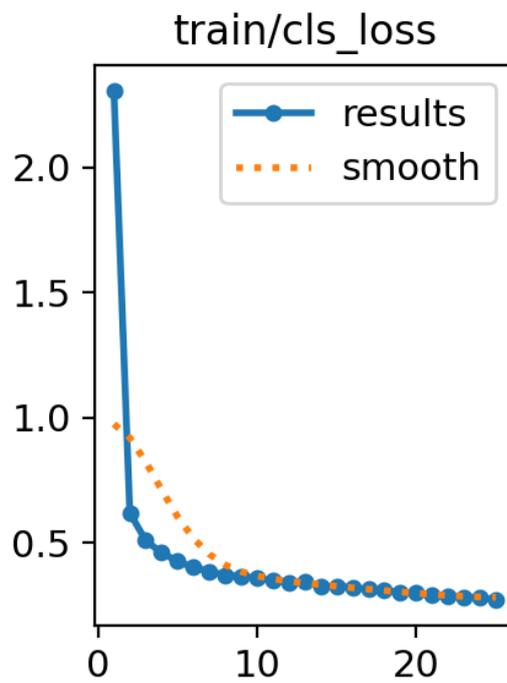
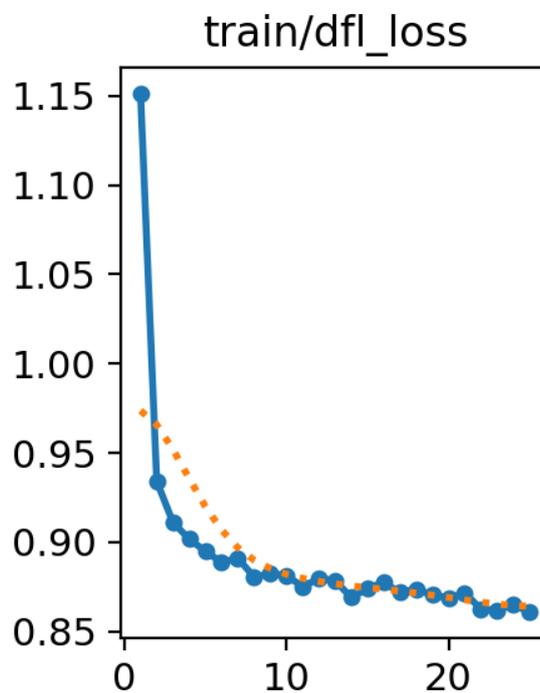
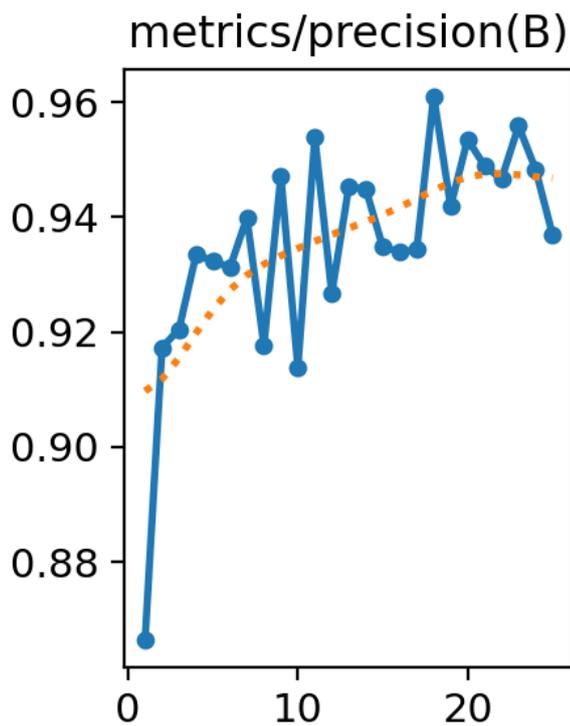
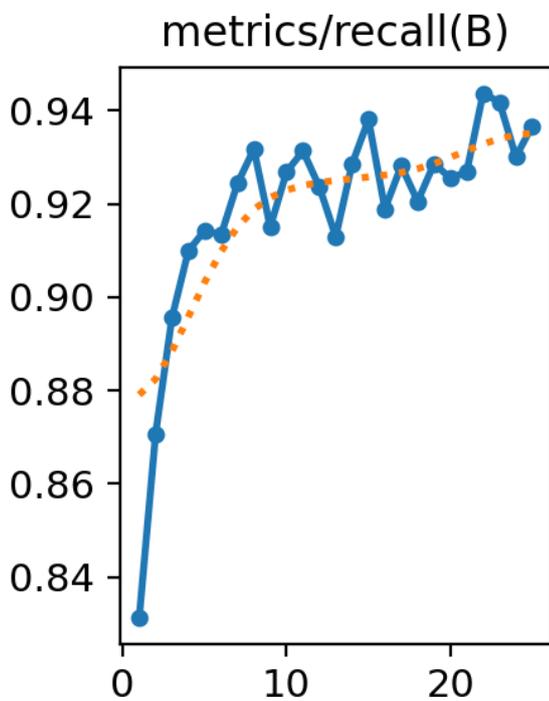


FIGURA 4.27 – Perda relacionada à localização das caixas delimitadoras durante o treinamento no *Google Colab*.

FIGURA 4.28 – Perda de classificação durante o treinamento no *Google Colab*.FIGURA 4.29 – Perda Focal Distribuída durante o treinamento no *Google Colab*.

FIGURA 4.30 – Precisão do modelo durante o treinamento no *Google Colab*.FIGURA 4.31 – Capacidade do modelo de identificar corretamente todos os objetos durante treinamento no *Google Colab*.

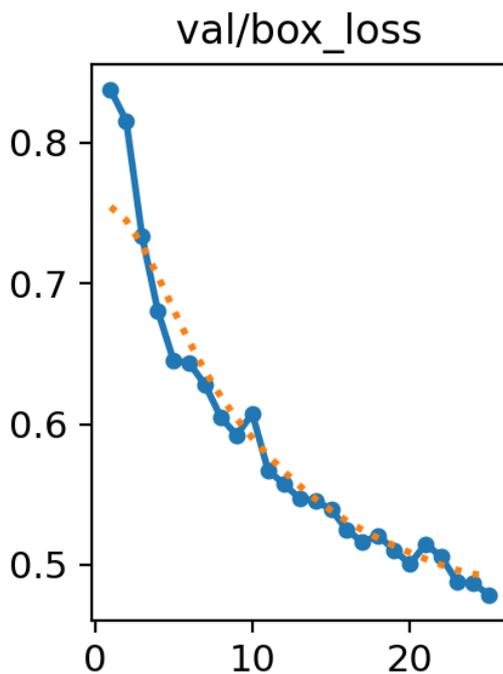


FIGURA 4.32 – Perda relacionada à localização das caixas delimitadoras durante a validação do treinamento no *Google Colab*.

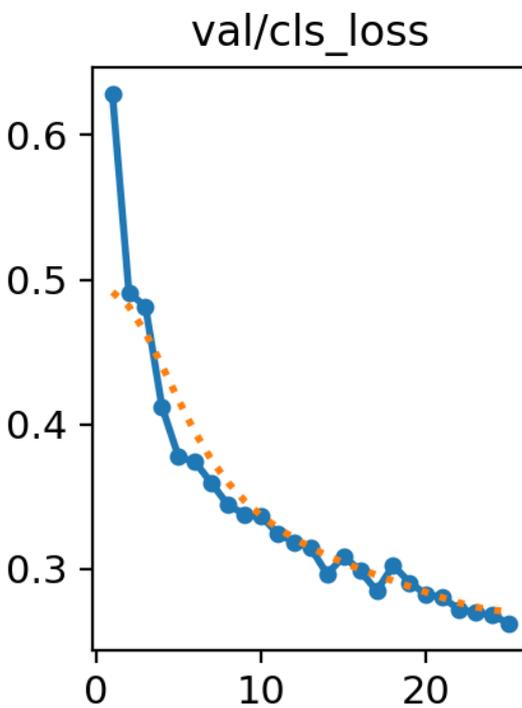


FIGURA 4.33 – Perda de classificação no conjunto de validação durante treinamento no *Google Colab*.

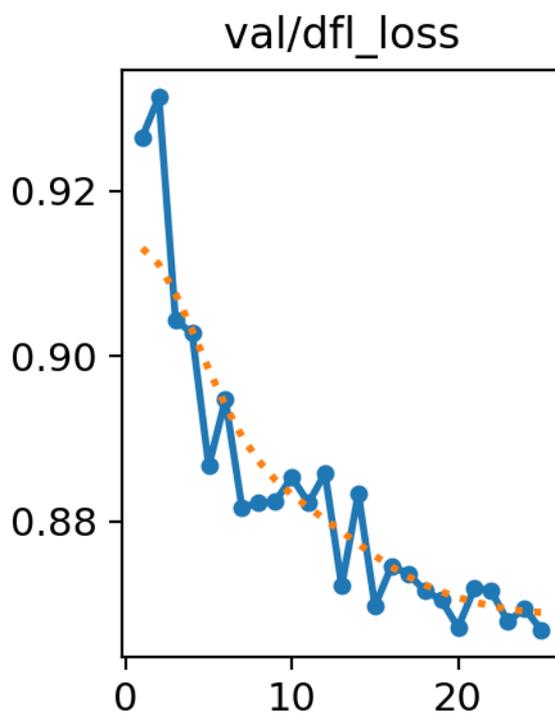
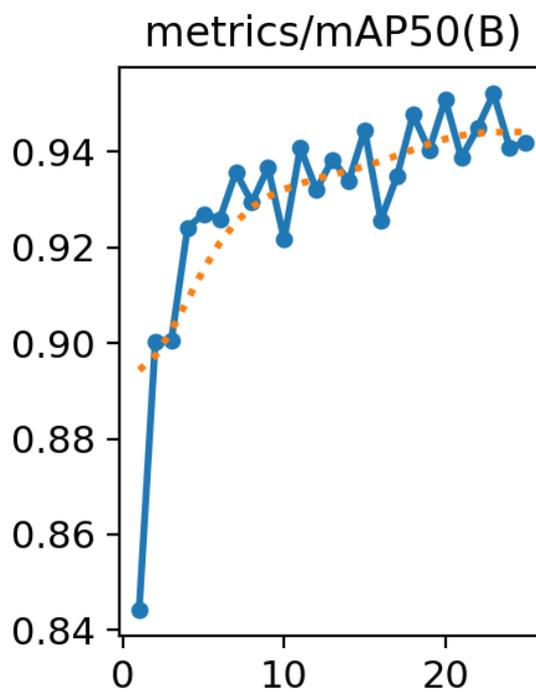
FIGURA 4.34 – *Distributed Focal Loss* (DFL) durante validação no *Google Colab*.

FIGURA 4.35 – A métrica mean Average Precision (mAP) com IoU (Intersection over Union) de 50%.

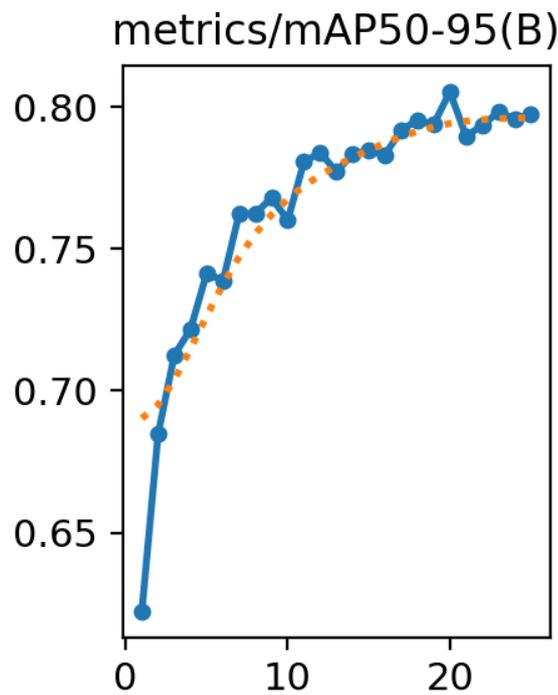


FIGURA 4.36 – Média das precisões calculadas em vários limiares de IoU, de 50% a 95%.

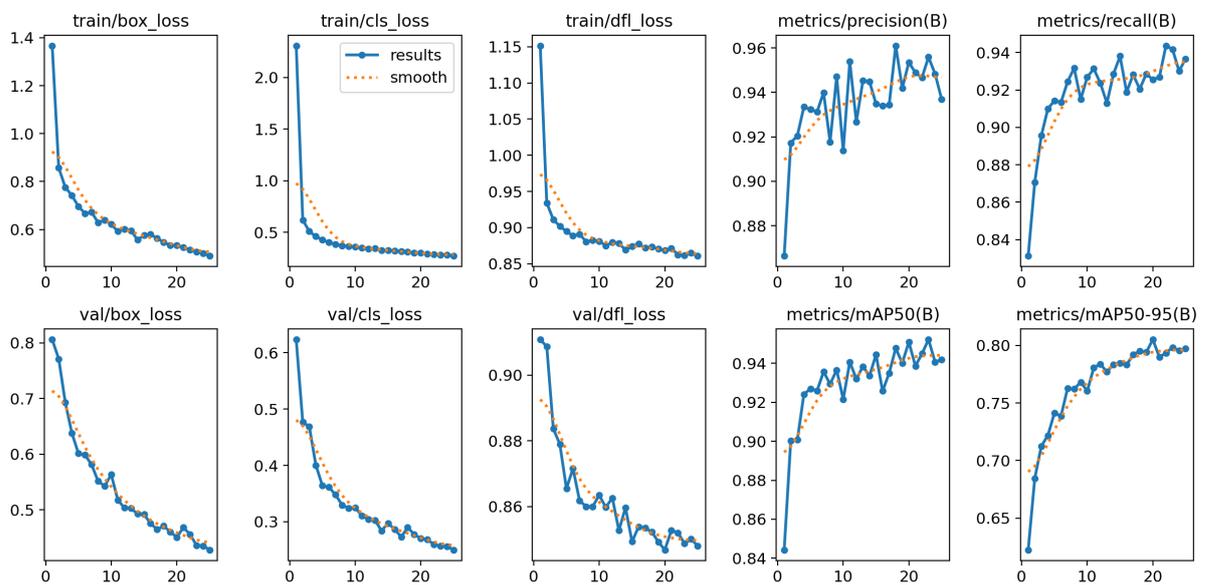
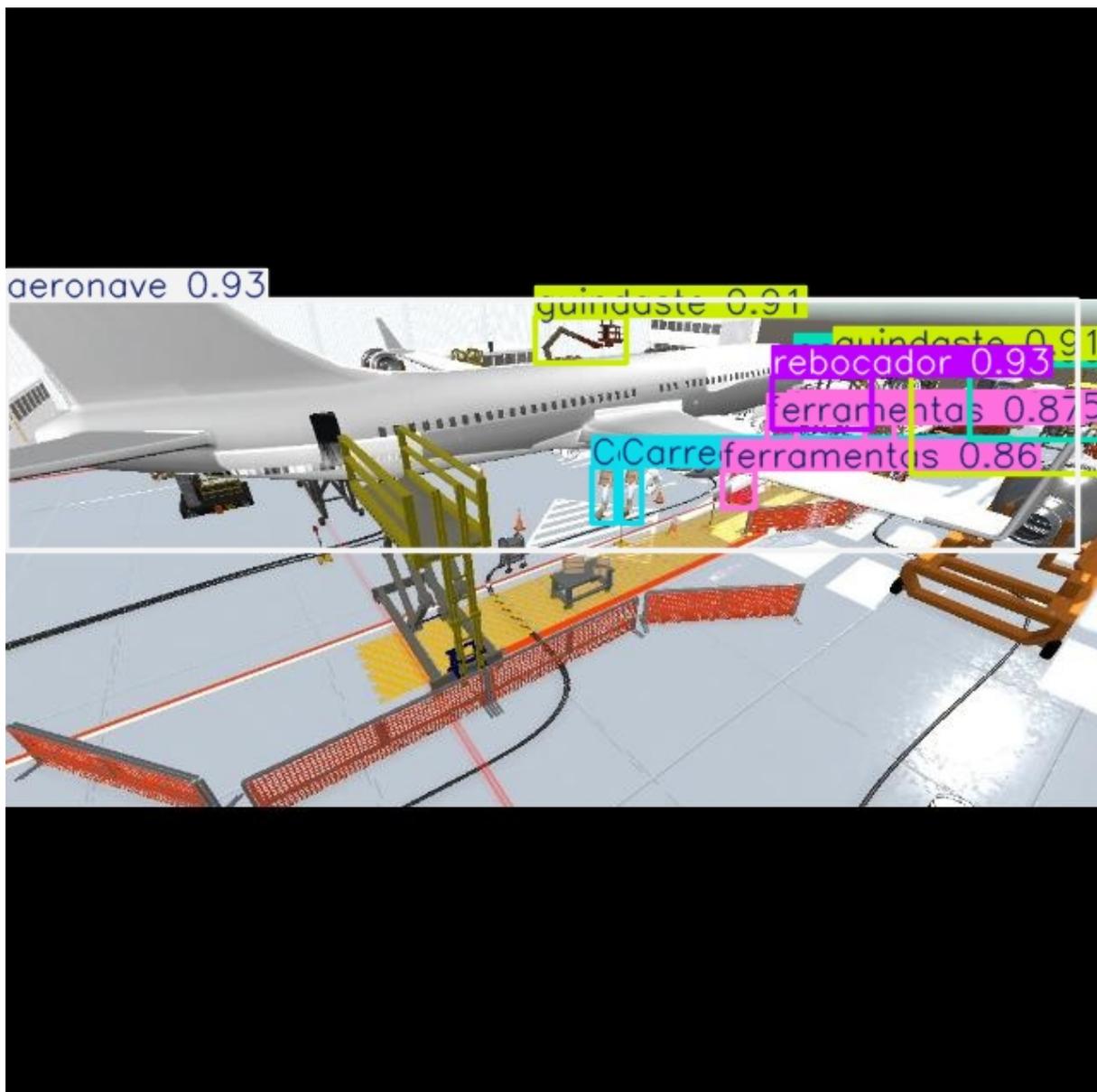
FIGURA 4.37 – Síntese dos gráficos com métricas de treinamento feito no *Google Colab*.



FIGURA 4.38 – Teste de modelo executado no *Google Colab*

FIGURA 4.39 – Resultado de treinamento no *Google Colab*.

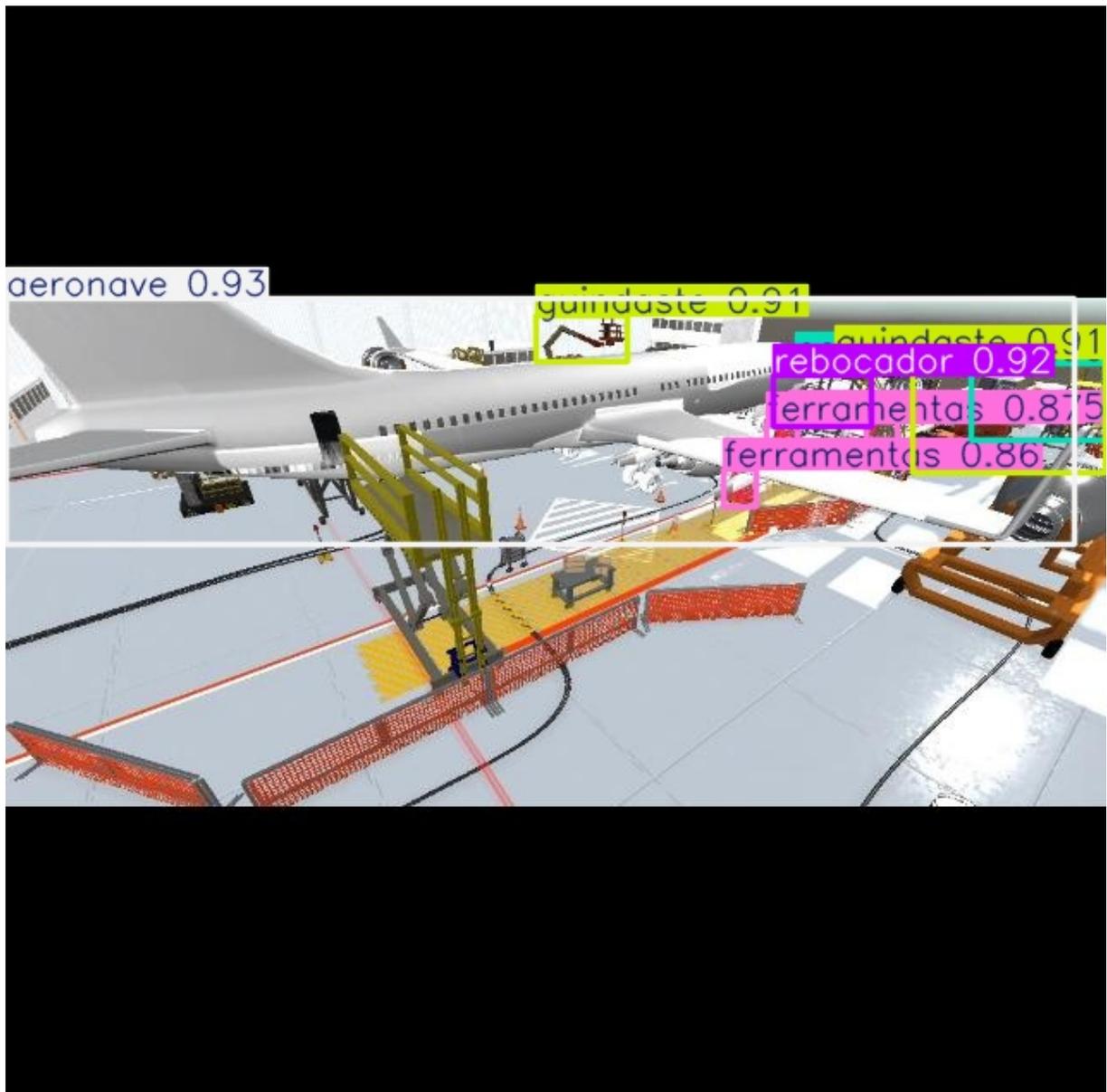
FIGURA 4.40 – Resultado de treinamento no *Google Colab*.

FIGURA 4.41 – Resultado de treinamento no *Google Colab*.

FIGURA 4.42 – Validação de detecção no *Roboflow*.

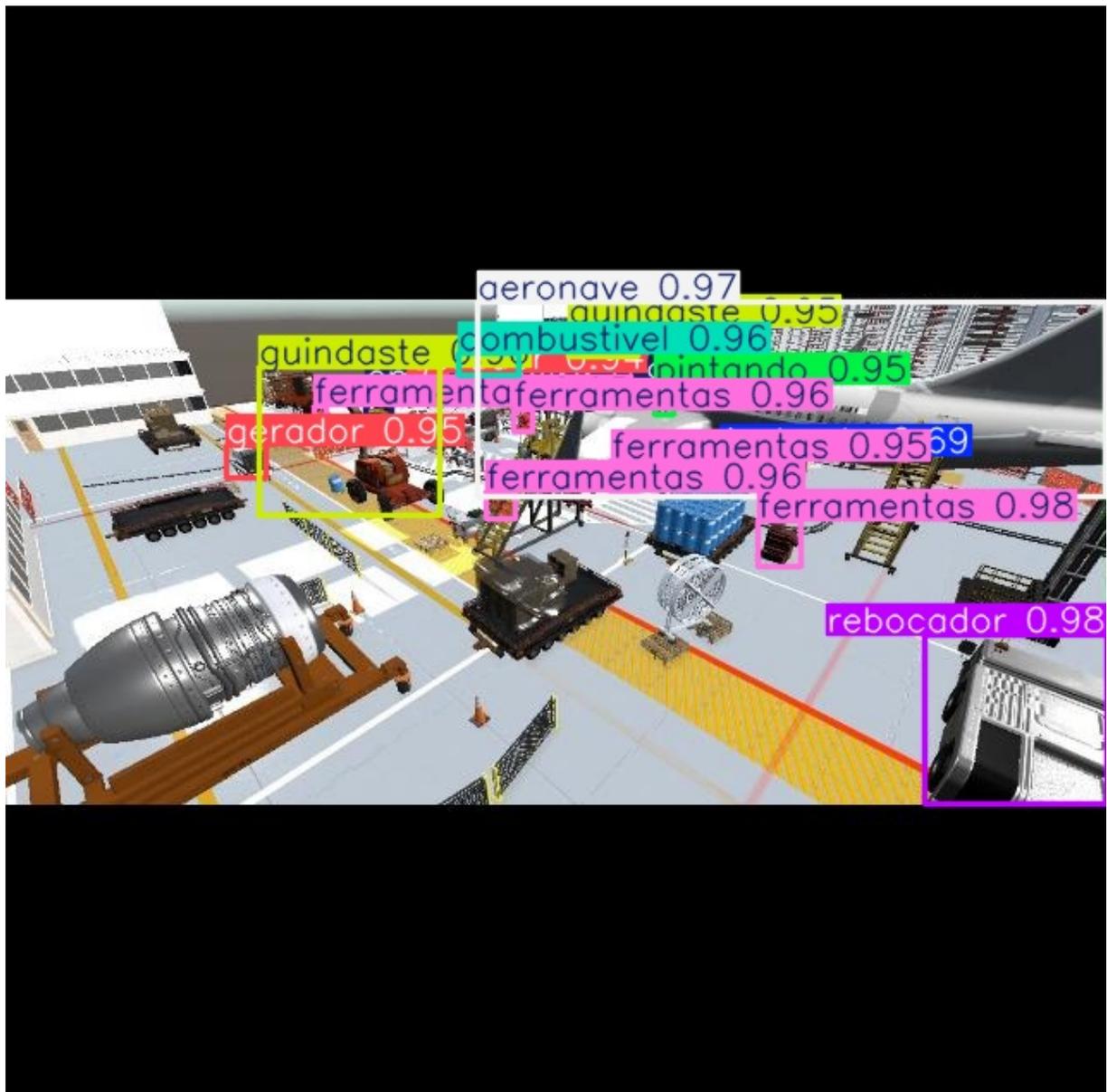
FIGURA 4.43 – Validação de detecção no *Roboflow*.



FIGURA 4.44 – Resultado de treinamento no *Google Colab*, usando o YOLOv8.



FIGURA 4.45 – Resultado de treinamento no *Google Colab*, usando o YOLOv8.

FIGURA 4.46 – Resultado de treinamento no *Google Colab*, usando o YOLOv8.FIGURA 4.47 – Resultado de treinamento no *Google Colab*, usando o YOLOv8.



FIGURA 4.48 – Resultado de treinamento no *Google Colab*, usando o YOLOv8.

O resultado da validação foi:

- Número de Imagens: 57
- Número de Classes: 10
- Precisão Global (mAP50): 0.956
- mAP50-95 (média ponderada): 0.807

As classes avaliadas foram:

Classe	Imagens	Instâncias	Box(P)	R	mAP50	mAP50-95
Geral ( <i>all</i> )	57	766	0,956	0,928	0,953	0,807
Andando	18	22	0,811	0,455	0,647	0,226
Carregando Peso	15	25	0,893	0,880	0,952	0,431
Aeronave	56	56	0,956	0,928	0,995	0,811
Combustível	57	57	0,990	1,000	0,995	0,911
Empilhadeira	38	57	0,990	1,000	0,995	0,933
Ferramentas	57	283	0,990	1,000	0,995	0,911
Gerador	38	57	0,990	1,000	0,995	0,973
Guindaste	57	114	0,990	1,000	0,995	0,961
Pintando	38	38	0,938	0,947	0,963	0,815
Rebocador	57	57	0,990	1,000	0,995	0,949

TABELA 4.1 – Resultados de validação por classe

A velocidade de execução foi:

- Pré-processamento: 0,5 ms
- Inferência: 25,6 ms
- Pós-processamento: 14,7 ms por imagem

Desse treinamento, a maioria das classes apresentou um desempenho satisfatório, com mAP50 próximo de 1,0. Entretanto, as classes “Andando” e “Carregando Peso” têm desempenho inferior (mAP50-95 de 0.226 e 0.431, respectivamente), sugerindo que melhorias no dataset ou refinamento do modelo podem ser necessárias para essas classes específicas.

O treinamento feito no *Roboflow* obteve os seguintes resultados:

- mAP = 93,1%
- Precision = 93,5%
- Recall = 93,5

Como o *mAP* foi maior do que 90%, tem-se um indicativo de que o modelo é preciso na indicação e localização de objetos nas imagens. Além disso, o alto valor de precisão indica menor detecção de falsos positivos. Por fim, o valor de *recall* denota que o modelo identifica a maioria dos objetos relevantes nas imagens, minimizando os falsos negativos. A quantidade de imagens obtidas após aumento é, conforme literatura, suficiente para um treinamento inicial.

O modelo apresenta resultados bons para detecção de objetos em termos de mAP, precisão e recall. Para uso prático, ele está em um estágio maduro e pronto para implementação para as configurações de imageamento descritas. Visando implementação em outros hangares de manutenção sem que seja necessária configuração conforme a metodologia e novos treinamentos, tornam-se necessários treinamentos com imagens dinâmicas (e não estáticas), com uma variação de ângulos maiores e diversidade de modelos de componentes.

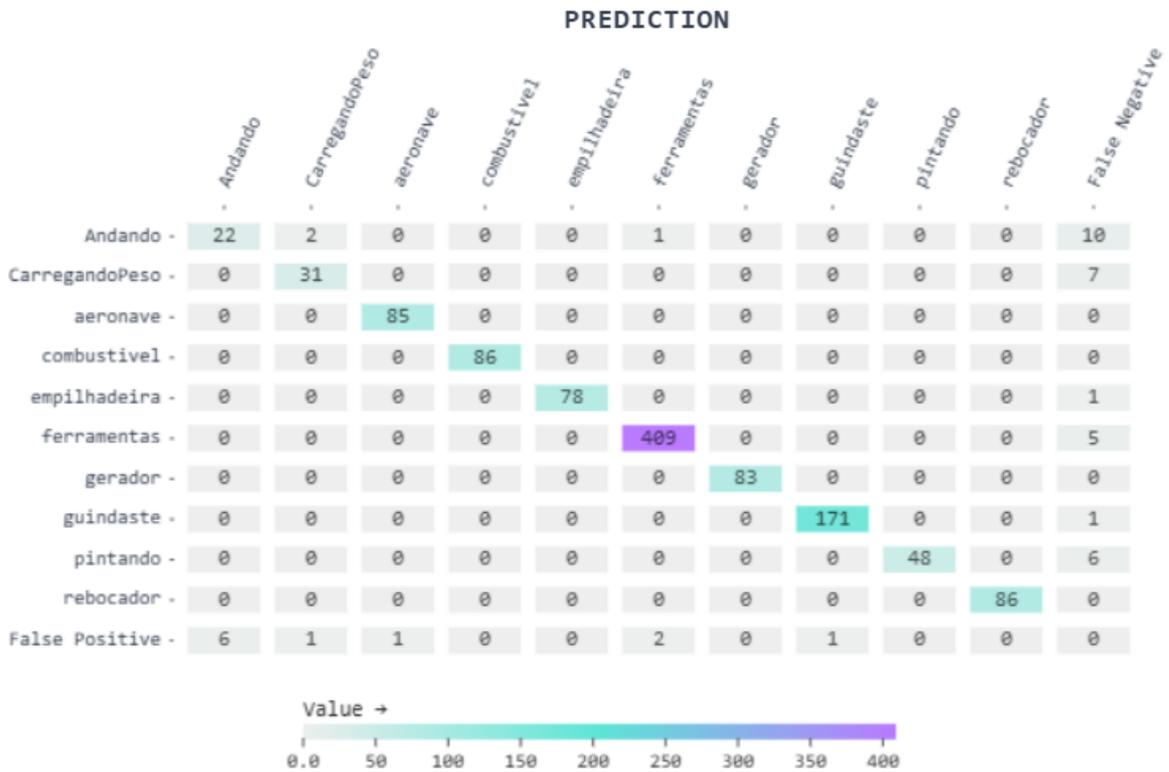


FIGURA 4.49 – Matriz de confusão do modelo treinado no *Roboflow* (dados de teste e validação).

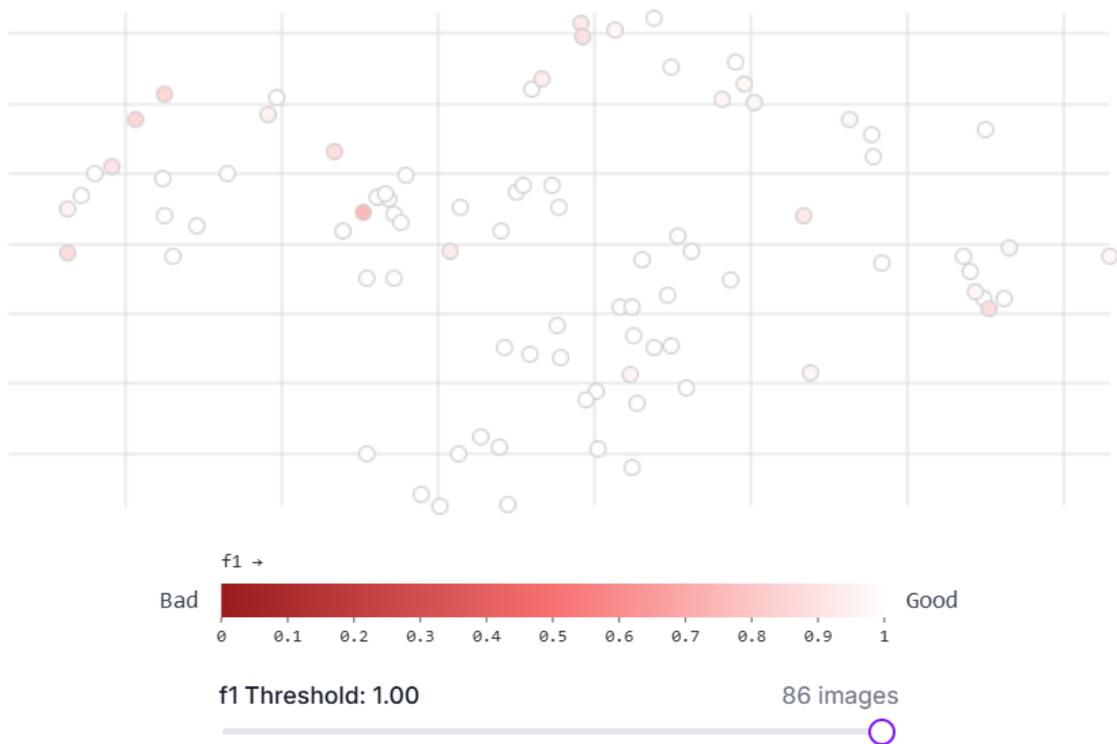
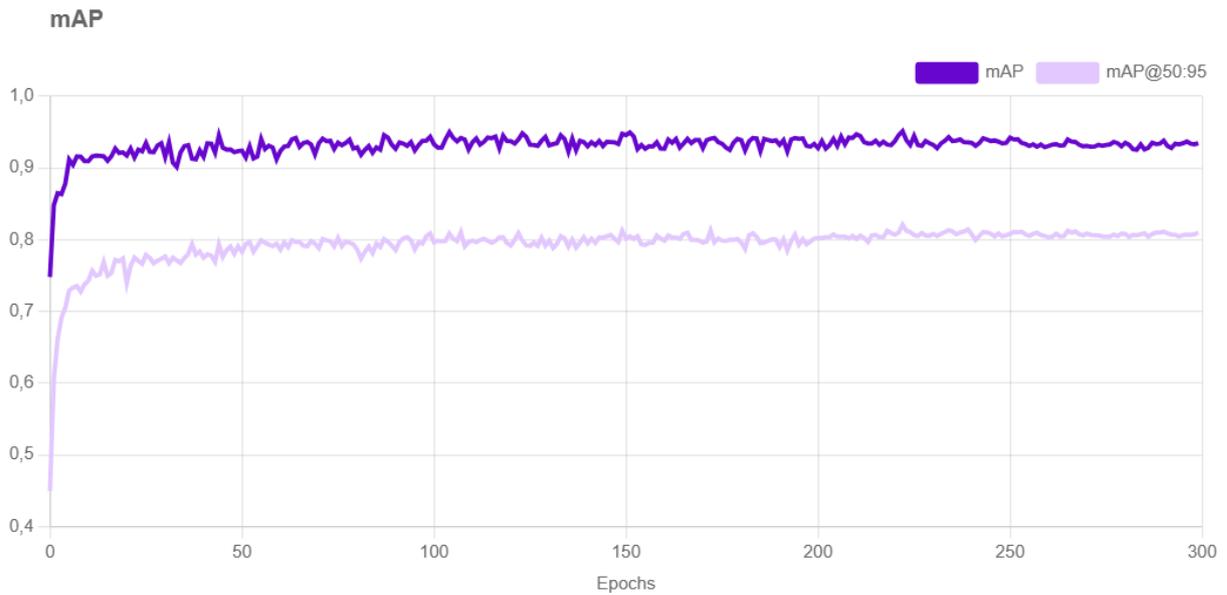
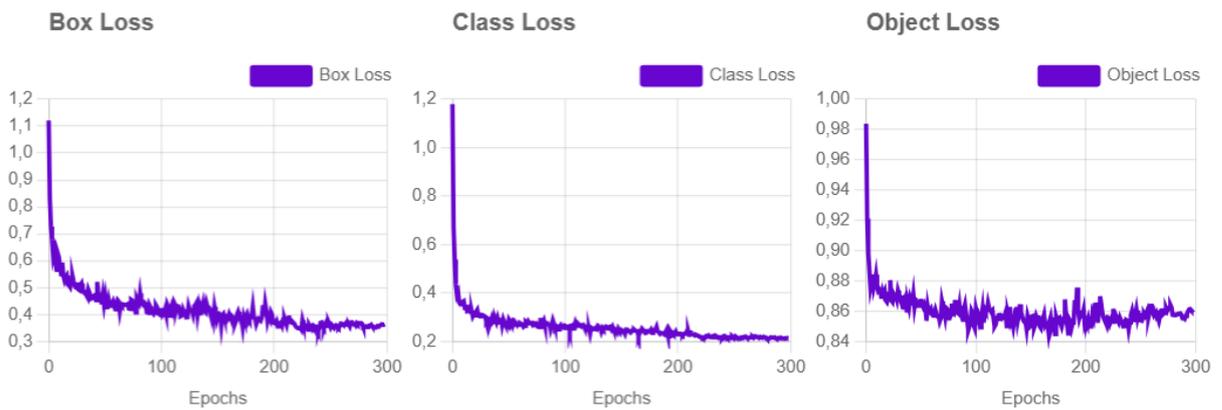
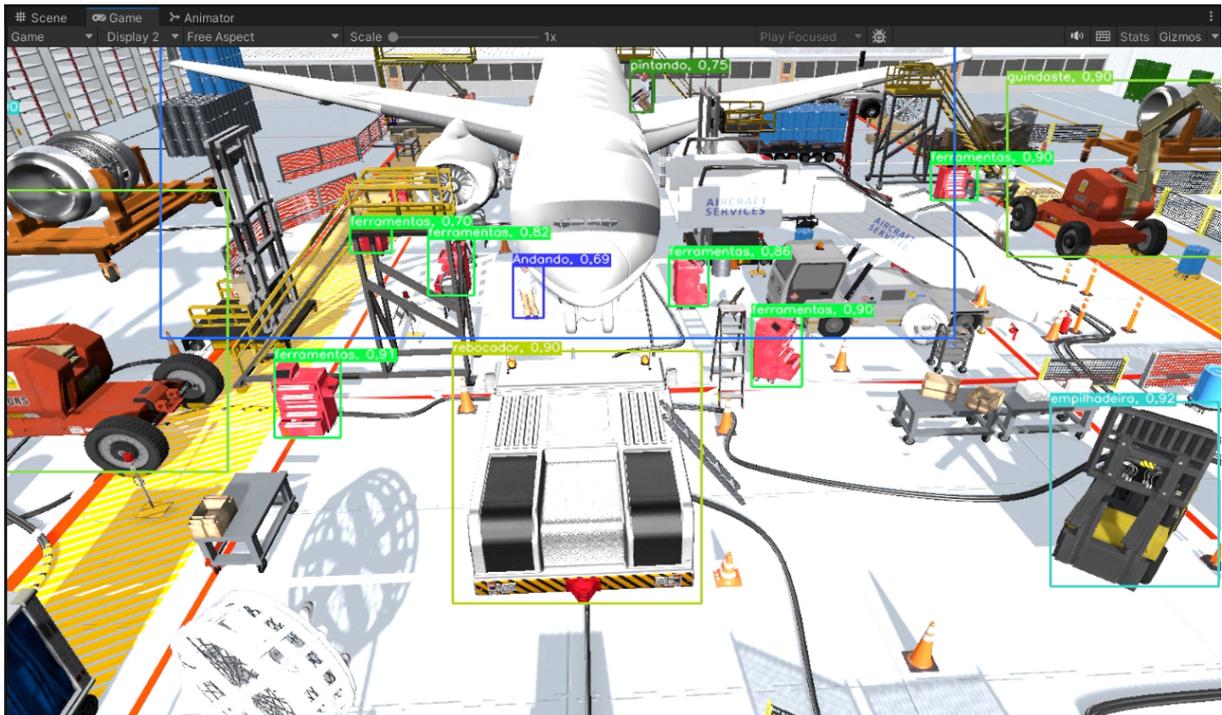


FIGURA 4.50 – Análise vetorial do modelo treinado no *Roboflow* (dados de teste e validação).

FIGURA 4.51 – Gráfico mAP do modelo treinado no *Roboflow* (dados de teste e validação).FIGURA 4.52 – Gráficos do modelo treinado no *Roboflow* (dados de teste e validação).

FIGURA 4.53 – Execução do modelo treinado no *Unity*.

A próxima etapa do estudo de caso consistiu na programação e estudo de um carro de ferramentas autônomo, pré-programado conforme as características do hangar de manutenção e capaz de se locomover entre pontos comandados, evitando obstáculos ao ter um gêmeo digital.

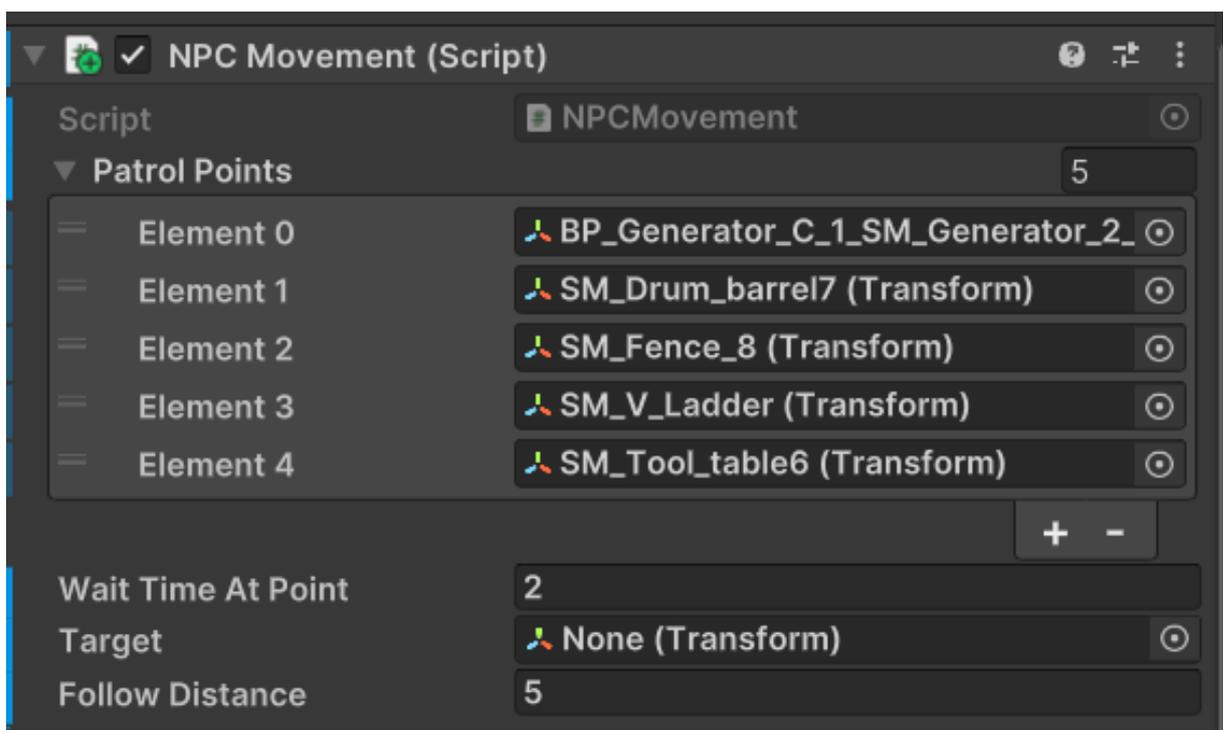


FIGURA 4.54 – Parâmetros que regem a movimentação do carro de ferramentas.

Por meio da discretização do hangar de manutenção e do uso do NavMeshSurface, possibilitou-se o monitoramento de um carrinho autônomo no hangar de manutenção, com a sua detecção em um ambiente controlado e controle dinâmico de posição.

#### 4.6.1 Limitações de Processamento em Tempo Real

Durante a execução deste trabalho, constatou-se que o processamento em tempo real, embora seja fundamental para muitas aplicações práticas, apresentou limitações significativas. Essas limitações decorreram, principalmente, das demandas computacionais elevadas para a execução simultânea de simulação e detecção no hangar de manutenção aeronáutica.

Os algoritmos empregados, como o *YOLOv8* integrado ao *OpenCV*, requerem um elevado poder de processamento gráfico para garantir detecção de objetos eficiente e em alta velocidade. Apesar de o modelo *YOLOv8* apresentar 70 quadros por segundo (FPS) e mAP de 0,85, observou-se que a complexidade do ambiente e a quantidade de elementos presentes na simulação impuseram desafios que afetaram a fluidez e a responsividade da aplicação em tempo real.

Além disso, a integração de múltiplas câmeras para cobrir áreas críticas do hangar contribui para o aumento da carga computacional, dado que cada imagem capturada é processada de forma independente pelo modelo de visão computacional. Com o hardware disponível, as taxas de quadros podem variar significativamente dependendo da complexidade da cena, o que resultou em pequenos atrasos na detecção de objetos e no acionamento de respostas autônomas do sistema.

Em aplicações práticas, a tolerância ao erro varia consideravelmente dependendo do contexto. No ambiente simulado deste trabalho, a exigência de precisão é alta para tarefas críticas, como a detecção de perigos em tempo real (por exemplo, colisões entre elementos em movimento) ou o desvio de agentes autônomos de obstáculos. Nesses casos, atrasos ou erros na classificação podem comprometer a eficácia e a segurança do sistema.

Por outro lado, em aplicações menos críticas, como a análise posterior de movimentos, níveis mais altos de erro podem ser toleráveis, especialmente se o foco estiver na extração de padrões gerais em vez de uma acurácia absoluta em tempo real.

# 5 Conclusão

O trabalho aqui disposto se encerra, após os resultados e discussões apresentados no capítulo 4, como uma abordagem eficaz para atender ao problema de pesquisa descrito no tópico 1.3. Por meio da metodologia e das ferramentas detalhadas no capítulo 3 utilizadas para a realização da ferramenta idealizada, obteve-se um fluxo sistemático capaz de garantir uma replicação ferramental em hangares de manutenção de uma forma geral, fornecendo instruções claras e específicas sobre: quais os principais itens que devem ser mapeados, forma de imageamento, captura de ações e forma de simulação. Além disso, apresenta as técnicas de visão computacional disponíveis na literatura, indicando um processo ordenado de treinamento de algoritmo e combinação de técnicas capazes de fornecerem tanto subsídios para uma implementação em ambiente controlado e simulado, quanto uma extrapolação para ambientes reais. Por fim, foi apresentada a metodologia de implementação de agente autônomo, dado gêmeo digital, capaz de ser monitorado pelo imageamento das câmeras. Dessa forma, todos os objetivos específicos, dispostos em 1.4.1, foram atingidos, de uma maneira gradual, para se atingir ao objetivo geral do trabalho. Os resultados obtidos destacaram o potencial da visão computacional para transformar o gerenciamento de manutenção, com impacto direto na segurança e na eficiência das operações.

## 5.1 Contribuição Acadêmica

O trabalho ofereceu uma solução competente e interdisciplinar no campo de manutenção aeronáutica, ao combinar fatores humanos, simulações computacionais e visão computacional em um ambiente de hangar de manutenção aeronáutica objetivando-se a apresentação de uma ferramenta capaz de viabilizar a implementação das principais tendências tecnológicas apresentadas no capítulo 2.1. Este trabalho contribui de maneira significativa para a área ao abordar uma possibilidade de captura do fluxo de trabalho por meio de monitoramento inteligente (ou seja, capaz de interpretar o ambiente) utilizando câmeras, apresentando técnicas modernas de visão computacional e definindo uma metodologia que aplica, inovadoramente frente à literatura atualmente vigente, a combinação

de YOLOv8, *OpenCV* e *Unity* em um hangar de manutenção aeronáutica simulado.

Além disso, ao preencher as lacunas descritas no 1.3, se torna ferramenta capaz de:

1. Fornecer um panorama gerencial às pessoas tomadoras de decisões, indicando movimentação de aparatos pesados, capazes de colidir com estruturas importantes, e a rápida detecção de objetos;
2. Indicar atividades, tarefas e tempos de duração; e
3. Compor outras ferramentas, capazes de fornecer sugestões de atuação com base na ferramenta aqui descrita.

## 5.2 Aplicação na indústria

O trabalho possui aplicabilidade na indústria de manutenção aeronáutica (com viés de expansão para outros segmentos de manutenção na totalidade) permitindo não só a captura, o mapeamento e a interpretação do cenário descrito como também a identificação dos gargalos e a melhoria da logística interna, proporcionando um maior controle e previsibilidade dos processos.

A introdução dessa ferramenta, adaptada ao contexto de cada hangar de manutenção, compõem um diferencial competitivo e uma necessidade para atender às exigências de um mercado em crescente expansão que demanda cada vez mais um aumento na operacionalidade dos serviços que o engloba.

Dessa forma, possui potencial para diminuir eventuais riscos de colisões, possibilitar a integração de um agente autônomo e fornecer um panorama capaz de possibilitar uma rápida tomada de decisões, fatores diretamente relacionados à saúde econômica da companhia.

## 5.3 Praticabilidade Operacional

Verifica-se, ainda que este trabalho possui potencial para ser aplicado em hangares de defesa, uma vez que a metodologia descrita pode ser implementado nas forças armadas para aumentar a sua operacionalidade, diminuindo riscos de colisões e possibilitando uma melhoria na alocação de receita e gerenciamento de recursos, fornecendo um painel gerencial capaz de contabilizar os elementos presentes no hangar para tomadores de decisão que podem tanto estar no ambiente do hangar de manutenção, como em outra cidade.

Além disso, a implementação da metodologia, com rotulagem específicas das aeronaves que compõem a Força Aérea Brasileira e as ações mais usuais ou causadoras de

conflito, fornece um aparato valioso capaz de aumentar a operacionalidade dos hangares de manutenção de defesa.

## 5.4 Sugestão de trabalhos futuros

O presente trabalho explora e define uma tecnologia capaz de fazer a captura e interpretação de um hangar de manutenção, viabilizando o desenvolvimento e a implementação de ferramentas capazes de proporcionar um hangar inteligente. Dessa forma, a partir do momento em que as lacunas descritas no 1.3 foram preenchidas, as etapas subsequentes consistem no uso e mensuração de eficiência do impacto que o uso de tecnologias como realidade aumentada, realidade virtual e treinamentos com base no hangar mapeado por visão computacional poderiam proporcionar.

Dessa forma, sugere-se que futuros trabalhos incorporem a metodologia aqui presente e enfoquem na possibilidade de tomada de decisões com base na interpretação da captura do hangar, disponibilizando um algoritmo tomador de decisões que possa: reduzir riscos, aliviar a carga de trabalho e fornecer um painel gerencial mais amplo acerca das possibilidades existentes no cenário. Após a criação do algoritmo descrito, sugere-se a sua implementação e alerta aos usuários de interesse por meio de mensagens visuais e áudio, com posterior mensuração do impacto da ferramenta gerada em um ambiente real.

# Referências

ASSOCIATION, I. A. T. **Airport Development Reference Manual**. International Air Transport Association, 1995. ISBN 9789290357292. Available at: <https://books.google.com.br/books?id=tm9PQQAACAAJ>.

BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.

CASEY, L.; DOOLEY, J.; CODD, M.; DAHYOT, R.; COGNETTI, M.; MULLARKEY, T.; REDMOND, P.; LACEY, G. A real-time digital twin for active safety in an aircraft hangar. **Frontiers in Virtual Reality**, 2024.

CONTRIBUIDORES, M. **YOLOv8 by MMYOLO**. 2023. <https://github.com/open-mmlab/mmyolo/tree/main/configs/yolov8>. Acesso em: 15 Março 2024.

E-CON SYSTEMS. **What are depth sensing cameras & how do they work?** 2024. Available at: <https://www.e-consystems.com/blog/camera/technology/what-are-depth-sensing-cameras-how-do-they-work/>.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. **arXiv preprint arXiv:1311.2524**, 2014.

GKAGIANNIS, K.; KOMNINOS, A. Spatial multimodal alert cues for virtual reality control environments in industry 4.0. **International Conference of the ACM Greek SIGCHI Chapter**, 2023.

GONZAGA, L. M.; ALMEIDA, G. M. de; SANTOS, C. M. C. V. Identificação e medição de defeitos em produtos automotivos utilizando visão computacional. **Peer Review**, 2023.

HOBBS, A. **An Overview of Human Factors in Aviation Maintenance**. Canberra, Australia: Australian Transport Safety Bureau, 2008. AR-2008-055. ISBN 978-1-921490-93-4. Available at: <http://www.atsb.gov.au/>.

Jeevan S. Devagiri; SIDIKE, P.; PAHEDING, S.; Quamar Niyaz; Xiaoli Yang; SMITH, S.; Samantha Smith. Augmented reality and artificial intelligence in industry. **Expert Systems With Applications**, p. 118002–118002, mAG ID: 4283802212 S2ID: 4a8e83e01d83f9f3c796fda7d07588995ed65e19, jul. 2022.

- JULIANI, A.; BERGES, V.-P.; VCKAY, E.; GAO, Y.; HENRY, H.; MATTAR, M.; LANGE, D. Unity: A general platform for intelligent agents. **arXiv: Learning**, 2018.
- KRISHNA, K. M.; CHOWDARI, K. A.; ANUSHA, J.; SRAVANI, S. L.; CHOWDARY, S. Object detection and screen presence time estimation using opencv and yolo algorithm. **International Journal of Food and Nutritional Science**, 2023.
- LIU, Z.; MEYENDORF, N.; MRAD, N. The role of data fusion in predictive maintenance using digital twin. **null**, 2018.
- MELLEMA, G. M. Application of dupont's dirty dozen framework to commercial aviation maintenance incidents. **null**, 2018.
- MYERS, K.; BERRY, P. M.; BLYTHE, J.; CONLEY, K.; GERVASIO, M.; MCGUINNESS, D. L.; MORLEY, D. N.; PFEFFER, A.; POLLACK, M. E.; TAMBE, M. An intelligent personal assistant for task and time management. **Ai Magazine**, 2007.
- NILSSON, N. E.; JACOBSSON, J. Operator - robot collaboration - creation and evaluation of a collaborative operator - robot workstation. **null**, 2017.
- OCIMF. Fatores humanos. **OCIMF**, acessado em: 04-set-2024, 2024. Available at: <https://www.ocimf.org/pt/fatores-humanos/~:text=Os%20fatores%20humanos%20s%C3%A3o%20caracter%20ADsticas, indiv%20duos%20ou%20equipes%20de%20trabalho>.
- OLAGANATHAN, R. Human factors in aviation maintenance: understanding errors, management, and technological trends. **Global Journal of Engineering and Technology Advances**, 2024.
- OLORUNSHOLA, O.; JEMITOLA, P.; ADEM UWAGUN, A. Comparative study of some deep learning object detection algorithms: R-cnn, fast r-cnn, faster r-cnn, ssd, and yolo. **Nile Journal of Engineering and Applied Science**, 2023.
- OSIGBESAN, A.; BARRAT, S.; SINGH, H.; XIA, D.; SINGH, S.; XING, Y.; GUO, W.; TSOURDOS, A. Vision-based fall detection in aircraft maintenance environment with pose estimation. **International Conference on Multisensor Fusion and Integration for Intelligent Systems**, 2022.
- PHAND, S. R.; PATHARE, S. V.; BHAVSAR, S. A.; ANDHARE, R. V.; BHANDARI, G. M. Human fall detection using cnn algorithm. **International Journal of Advanced Research in Science, Communication and Technology**, 2023.
- QIU, W.; ZHONG, F.; ZHANG, Y.; QIAO, S.; XIAO, Z.; KIM, T. S.; YU, Y. Unrealcv: Virtual worlds for computer vision. **ACM Multimedia**, 2017.
- RANTANEN, E. M.; RANTANEN, E. M.; RANTANEN, E. M.; BUTTS, T. A.; BUTTS, T. A.; WOJTOWICZ, D. S.; WOJTOWICZ, D. S.; WEBB, M. L.; WEBB, M. L.; MARIN, A. K.; MARIN, A. K. Human factors considerations in the design of an aircraft maintenance hangar. **null**, 2003.
- ROMERAL, P. A. D. A. F.; ZANCUL, E. de S.; NASCIMENTO, P. T. de S. Visão computacional na indústria: tendências e exemplos práticos. **Anais do Encontro Nacional de Engenharia de Produção**, 2023.

- ROMERO, D.; STAHR, J.; WUEST, T.; NORAN, O. S.; BERNUS, P.; FAST-BERGLUND Åsa; GORECKY, D. Towards an operator 4.0 typology: A human-centric perspective on the fourth industrial revolution technologies. **Computers in Entertainment**, 2016.
- SAFI, M.; CHUNG, J.; PRADHAN, P. Review of augmented reality in aerospace industry. **Aircraft Engineering and Aerospace Technology**, v. 91, n. 9, p. 1187–1194, mAG ID: 2958645313 S2ID: 6b31b3fde662d85567f7c530ab5b7537f410129f, out. 2019.
- STEINBERG, A. N.; BOWMAN, C.; WHITE, F. E. Revisions to the jdl data fusion model. **Defense, Security, and Sensing**, 1999.
- STUSTER, J. **The Human Factors and Ergonomics Society: Stories From the First 50 Years**. Santa Monica, CA: The Human Factors and Ergonomics Society, 2006.
- SUN, S.; ZHENG, X.; GONG, B.; PAREDES, J.; ORDIERES-MERÉ, J. Healthy operator 4.0: A human cyber-physical system architecture for smart workplaces. **Sensors**, 2020.
- SYLLA, N.; BONNET, V.; COLLEDANI, F.; FRAISSE, P. Ergonomic contribution of able exoskeleton in automotive industry. **International Journal of Industrial Ergonomics**, 2014.
- SZELISKI, R. Computer vision - algorithms and applications. **Texts in Computer Science**, 2011.
- Sá, A. G. de; ZACHMANN, G. Virtual reality as a tool for verification of assembly and maintenance processes. **Computers & Graphics**, v. 23, n. 3, p. 389–403, mAG ID: 1982954541 S2ID: 62567b885a9a936bd57a9e753c231f5046ac68d8, jun. 1999.
- TERVEN, J. R.; ESPARZA, D. M. C.; ROMERO-GONZÁLEZ, J. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. **Machine Learning and Knowledge Extraction**, 2023.
- VARGAS, A. C. G.; CARVALHO, A. M. P.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. **Instituto de Computação, Universidade Federal Fluminense**, disponível online em <http://www.uff.br> (Acessado em 13 de novembro de 2024), 2023.
- WUEST, T.; HRIBERNIK, K.; THOBEN, K. Can a product have a facebook? a new perspective on product avatars in product lifecycle management. **Product Lifecycle Management**, 2012.
- WUEST, T.; WEIMER, D.; IRGENS, C.; THOBEN, K. Machine learning in manufacturing: advantages, challenges, and applications. **Production and Manufacturing Research: An Open Access Journal**, 2016.
- XU, S.; XU, S.; WANG, D.; WANG, J.; WANG, X.; WANG, X.; SHOU, W.; SHOU, W. Computer vision techniques in construction, operation and maintenance phases of civil assets: A critical review. **Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)**, 2019.

---

ZHAO, R.; TIAN, Y. Application of computer vision and image processing technology in bridge condition monitoring system. **2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)**, 2023.

## FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO <p style="text-align: center;">TC</p>	2. DATA <p style="text-align: center;">28 de novembro de 2024</p>	3. DOCUMENTO Nº <p style="text-align: center;">DCTA/ITA/TC-149/2024</p>	4. Nº DE PÁGINAS <p style="text-align: center;">114</p>
5. TÍTULO E SUBTÍTULO: Uso de Visão Computacional no Mapeamento de Tarefas de Hangares de Manutenção Aeronáutica			
6. AUTORA(ES): <b>Rafaella Carmo dos Santos</b>			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
8. PALAVRAS-CHAVE SUGERIDAS PELA AUTORA: Visão Computacional; Hangares de Manutenção; Unity; YOLO; Simulação; OpenCV			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: 1. Visão por computador 2. Realidade aumentada 3. Mapeamento 4. Manutenção 5. Hangares 6. Computação 7. Engenharia Aeronáutica			
10. APRESENTAÇÃO: <span style="float: right;"><input checked="" type="checkbox"/> Nacional    <input type="checkbox"/> Internacional</span> ITA, São José dos Campos. Curso de Engenharia Civil-Aeronáutica. Orientador: Henrique Costa Marques; Apresentação em 18/11/2021. Publicada em 2024.			
11. RESUMO: <p>A disponibilização de novas ferramentas relacionadas à visão computacional e detecção de objetos possibilita uma ampla gama de aplicações capazes de melhorar o mundo contemporâneo. Nesse contexto, o mapeamento de atividades em ambientes industriais e de defesa para a identificação de contexto de manutenção aeronáutica e a diminuição de riscos é objeto de estudo que, antes, seria considerado inviável.</p> <p>Busca-se, nesse relatório, a integração de ferramentas de visão computacional para o mapeamento de ações e objetos que compõem os hangares de manutenção, com o objetivo de alcançar o reconhecimento computacional das etapas críticas (temporal e probabilisticamente com maior risco) para que, posteriormente, sejam indicadas arquiteturas tecnológicas capazes de suprir a criticidade das operações.</p> <p>Dessa forma, esse trabalho fez a integração entre <i>OpenCV</i>, <i>YOLOv8</i> e uma simulação de um hangar de manutenção no <i>Unity</i> com subsídios para a implementação em um hangar real, possibilitando uma captura e interpretação nesse ambiente, assim como uma análise de viabilidade do mapeamento de tarefas.</p>			
12. GRAU DE SIGILO: <p style="text-align: center;"> <input checked="" type="checkbox"/> <b>OSTENSIVO</b>                      <input type="checkbox"/> <b>RESERVADO</b>                      <input type="checkbox"/> <b>SECRETO</b> </p>			