



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Rafael Ramos da Silva Quint

**Sistema inteligente para automação de testes em câmeras de segurança
aplicado ao processo produtivo**

Florianópolis
2022

Rafael Ramos da Silva Quint

**Sistema inteligente para automação de testes em câmeras de segurança
aplicado ao processo produtivo**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Rômulo Silva de Oliveira, Dr.
Supervisor: Bruno Rozza, Eng.

Florianópolis
2022

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

Quint, Rafael Ramos da Silva

Sistema inteligente para automação de testes em câmeras de segurança aplicado ao processo produtivo Local ano / Rafael Ramos da Silva Quint ; orientador, Rômulo Silva de Oliveira, 2022.

71 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2022.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Redes Neurais. 3. Câmera de segurança. 4. Jiga de Teste. I. Silva de Oliveira, Rômulo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Rafael Ramos da Silva Quint

**Sistema inteligente para automação de testes em câmeras de segurança
aplicado ao processo produtivo**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 02 de agosto de 2022.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Rômulo Silva de Oliveira, Dr.
Orientador
UFSC/CTC/DAS

Bruno Rozza, Eng.
Supervisor
Empresa Intelbras

Prof. Laio Oriel Seman, Dr.
Avaliador
Instituição UNIVALI

Prof. Eduardo Camponogara, Dr.
Presidente da Banca
UFSC/CTC/DAS

RESUMO

O processo de testes visuais aplicado às câmeras de segurança produzidas na indústria é pouco eficiente, pois, de forma isolada, não filtra todos os produtos defeituosos e demanda o retrabalho de câmeras íntegras. Isso gera custos na reinspeção de lotes de produção e permite a entrega de produtos que não atendem aos padrões de qualidade ao cliente final. Este trabalho propõe a reformulação da análise de imagem realizada por um operador para um sistema automatizado de classificação por Redes Neurais Artificiais. Os problemas de maior incidência no processo produtivo são avaliados por meio dos testes de: presença de tampa plástica, integridade da cor gerada, presença de mancha escura, acionamento do filtro ICR e do LED da câmera. A alternância entre os cenários de teste ocorre através de acionamentos pneumáticos e elétricos, a fim de simular diferentes condições em uma câmera que evidenciam a ocorrência do defeito. Com a mudança, o tempo total de teste reduziu em cerca de 20%, não sendo registrada a aprovação de câmeras defeituosas em três semanas de validação, para uma célula de produção.

Palavras-chave: Redes Neurais. Câmera de segurança. Jiga de Teste.

ABSTRACT

The visual testing process applied to the security cameras, manufactured in the industry, has low efficiency because, in isolation, it approves defective products and also requires the reworking of good ones. This causes the rising of expenses by demanding the inspection of batches. Furthermore, the current process allows costumers to receive a low quality product. This paper aims to change the way those tests are made by replacing the human visual test for an automated system based on an Artificial Neural Network approach. The most common problems that appear in the factory are evaluated through the following tests: plastic cover presence, integrity of the generated color, existence of dark spots, working of the infrared cut filter removal and LED activation. Pneumatic and electrical devices are used to switch between test scenarios in order to simulate the failure conditions. After this change, the time required to check each camera reduced about 20% and it was not registered the approval of defective products by the algorithm during the three weeks of validation.

Keywords: Artificial Neural Networks. Security cameras. Test Jig.

LISTA DE FIGURAS

Figura 1 – Número estimado de câmeras por 1000 habitantes.	10
Figura 2 – Matriz e filiais.	13
Figura 3 – Câmera VHD 3130 B e DVR Intelbras MHDX3104.	16
Figura 4 – Tipos de câmeras produzidas pela Intelbras.	16
Figura 5 – Esquema de componentes para câmera analógica.	17
Figura 6 – Comparação qualitativa entre ângulos nos modelos Bullet e Dome.	18
Figura 7 – Visão explodida do conjunto lentes, ICR e sensor de imagem.	18
Figura 8 – Esquema de funcionamento CMOS.	19
Figura 9 – Esquema: filtro de Bayer sobre sensor de imagem.	20
Figura 10 – Sinal de uma Câmera Analógica	20
Figura 11 – Matriz de confusão	21
Figura 12 – Representação Simplificada do Neurônio Biológico	23
Figura 13 – Representação Simplificada do Neurônio Artificial	23
Figura 14 – Plot de diferentes funções de ativação e suas derivadas	25
Figura 15 – Diagrama de uma Rede Neural	26
Figura 16 – Diagrama para conexão de camadas em uma CNN	30
Figura 17 – Compartilhamento de pesos entre conexões em uma CNN	30
Figura 18 – Compartilhamento de pesos entre conexões 3D em uma CNN	31
Figura 19 – Arquitetura da Le-Net5 para identificação de dígitos	32
Figura 20 – Blocos Residuais	34
Figura 21 – Arquitetura da ResNet18	34
Figura 22 – Tabela de requisitos	37
Figura 23 – Diagrama de classes	38
Figura 24 – Diagrama da estrutura do sistema de testes	40
Figura 25 – Desenho 3D da estrutura do sistema de testes	41
Figura 26 – Máquina de estados	41
Figura 27 – Diagrama de Circuito Pneumático	43
Figura 28 – Funcionamento de uma válvula solenoide 5/2	45
Figura 29 – Diagrama de circuito eletrônico	45
Figura 30 – Amostras do conjunto de dados para o teste de conexão	47
Figura 31 – Amostras do conjunto de dados para o teste de tampa	47
Figura 32 – Amostras do conjunto de dados para o teste de cor	48
Figura 33 – Amostras do conjunto de dados para o teste de mancha	49
Figura 34 – Câmera com o defeito de mancha escura	49
Figura 35 – Amostras do conjunto de dados para o teste de ICR	50
Figura 36 – Amostras do conjunto de dados para o teste de LED	50
Figura 37 – Dados de treinamento para o modelo da mancha	53

Figura 38 – Dados de refinamento para o modelo da mancha	53
Figura 39 – Gráfico dos dados de treinamento em função das épocas	54
Figura 40 – Matriz de confusão para o conjunto de validação	55
Figura 41 – Jetson Nano Developer Kit	56
Figura 42 – Carta branca em posição avançada	57
Figura 43 – Suporte das câmeras Dome	57
Figura 44 – Interface da Jiga de Testes	58
Figura 45 – Jiga de Testes implementada na fábrica em Manaus	59
Figura 46 – Projeto da placa de circuito impresso	60
Figura 47 – Matriz de confusão para as câmeras em produção	62
Figura 48 – Jiga de Testes em desenvolvimento	66
Figura 49 – Exemplos de padrões linearmente e não-linearmente separável respectivamente.	71

LISTA DE SÍMBOLOS

<i>ACC</i>	Acurácia
<i>VP</i>	Verdadeiro Positivo
<i>VN</i>	Verdadeiro Negativo
<i>FP</i>	Falso Positivo
<i>FN</i>	Falso Negativo
<i>P</i>	Precisão
<i>R</i>	Recall/Revocação
<i>F1</i>	F1-Score
<i>g</i>	Função de ativação
<i>x</i>	Entrada de um neurônio
<i>w</i>	Peso de uma conexão
<i>y</i>	Saída de um neurônio

SUMÁRIO

1	INTRODUÇÃO	10
1.1	PROBLEMA TRATADO E IMPORTÂNCIA	11
1.2	OBJETIVOS E SOLUÇÃO PROPOSTA	12
1.3	DESCRIÇÃO DA EMPRESA	12
1.4	ESTRUTURA DO DOCUMENTO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	TRABALHOS ANTERIORES	15
2.2	FUNCIONAMENTO DAS CÂMERAS ANALÓGICAS	16
2.2.1	Lentes e ICR	17
2.2.2	Sensor de Imagem e Processador	18
2.3	MÉTRICAS DE DESEMPENHO	21
2.4	REDES NEURAIS	22
2.4.1	Redes Neurais Convolucionais	29
2.4.2	ResNet	33
2.4.3	Conceitos utilizados no treinamento	35
3	REQUISITOS GERAIS E PROJETO	36
3.1	MODELAGEM	36
3.2	FUNCIONAMENTO	40
3.3	PROJETO DE CIRCUITOS	43
4	DESENVOLVIMENTO	46
4.1	TESTES	46
4.1.1	Teste de conexão	46
4.1.2	Teste de tampa	47
4.1.3	Teste de cor	48
4.1.4	Teste de mancha	48
4.1.5	Teste de ICR	49
4.1.6	Teste de LED	50
4.2	TREINAMENTO DO MODELO	51
4.2.1	Considerações sobre a função custo	51
4.2.2	Coleta de dados e treinamento	51
4.3	PROTÓTIPO	55
5	ANÁLISE DE RESULTADOS	61
6	CONSIDERAÇÕES FINAIS	64
6.1	IMPACTOS DO TRABALHO	64
6.2	CONCLUSÃO	66
	REFERÊNCIAS	68
	ANEXO A – LINEARIDADE DE PADRÕES	71

1 INTRODUÇÃO

Este trabalho está inserido no contexto de otimização de parte do processo fabril de câmeras analógicas, amplamente utilizadas em sistemas de monitoramento, através da automatização de testes visuais aplicados sobre as câmeras em produção.

A produção de uma câmera analógica passa por três etapas que compõem a subdivisão da fábrica da Intelbras, em Manaus. A primeira etapa consiste em montar os componentes eletrônicos sobre a placa do produto, além de associar o sensor de imagem à lente. Em seguida, destina-se o conjunto ao setor Ajuste de foco, aplicando-se o procedimento correspondente e testando-se o produto quanto à presença de manchas escuras na imagem. Por fim, a câmera é montada em sua versão definitiva no setor de Montagem Final, em que concentram-se os esforços deste trabalho.

O alto volume de produção de câmeras - aproximadamente 1200 por dia - e as possibilidades de expansão desse mercado permitem inferir acerca da relevância deste trabalho, pois os ganhos com a otimização do processo de testes, tanto em tempo como em qualidade, acabam sendo potencializados pela magnitude desse contexto. Esse potencial é reforçado por um estudo (BISCHOFF, 2021) o qual revisita uma pesquisa da IHS Markit, que projetava a existência de mais de um bilhão de câmeras de segurança instaladas até 2021, com aproximadamente metade delas concentradas na China. A Figura 1 ilustra uma visão geral da distribuição de câmeras de segurança nas 150 cidades mais populosas do mundo, indicando as possibilidades de expansão desse produto no Brasil.

Figura 1 – Número estimado de câmeras por 1000 habitantes.

The most surveilled cities in the world - cameras per 1,000 people



Fonte: www.comparitech.com

1.1 PROBLEMA TRATADO E IMPORTÂNCIA

O processo de produção de câmeras analógicas para sistemas de segurança eletrônica acaba gerando produtos defeituosos por diversas razões, como problemas de matéria-prima, erros de processo, desgaste de máquinas, entre outros. Com o objetivo de identificar os produtos defeituosos e evitar que o cliente final os receba, são executados procedimentos de testes manuais, definidos a partir do histórico de principais problemas registrados nas linhas de produção.

Após a conclusão da montagem de cada câmera são feitos cinco testes visuais por um operador, que consistem na verificação de:

- Ausência de tampa do produto;
- Mau-funcionamento do filtro ICR (*Infrared Cutfilter Removal*);
- Falha no acionamento do LED (*Light-emitting Diode*);
- Distorções na cor da imagem;
- Presença de mancha escura na imagem.

Vale ressaltar que o teste de mancha escura é realizado tanto na etapa de Ajuste de Foco quanto na etapa de Montagem Final, pois com o deslocamento da câmera pela fábrica, partículas sólidas podem atingir o sensor de imagem e gerar o defeito.

Entretanto, fatores humanos, como: distrações e limitação da capacidade de percepção de anomalias em imagens, podem implicar na aprovação de produtos que não atendem aos padrões de qualidade exigidos. Ressalta-se que a busca pela padronização dos testes executados também se faz necessária, pois atinge outras questões relacionadas a fatores humanos, já que em alguns dos testes realizados nas câmeras existe margem para interpretação por parte do operador quanto à conformidade do produto.

Ainda, uma segunda etapa de conferência é feita pelo setor de Controle de Qualidade, em que algumas amostras dos lotes produzidos são inspecionadas e, na ocorrência de defeitos, bloqueia-se o lote correspondente para que seja completamente inspecionado e retrabalhado.

Esse processo de identificação tardia de defeitos em câmeras é custoso, pois exige um maior investimento de recursos humanos para avaliar criteriosamente todo o lote produzido e reparar os produtos necessários. Por exemplo, a identificação de defeitos de mancha durante a etapa de Ajuste de foco é cerca de cinco vezes menos custosa que durante a etapa de Montagem Final - aproximadamente 30 segundos contra 150 segundos em termos de tempo de retrabalho - pois envolve o processo de

desmontar completamente uma câmera finalizada. Além disso, no pior cenário existe a possibilidade, embora pequena, de o problema somente ser identificado quando chega ao cliente final, desgastando a imagem da empresa frente aos consumidores.

1.2 OBJETIVOS E SOLUÇÃO PROPOSTA

Com base no problema apresentado, o objetivo deste trabalho é desenvolver um sistema inteligente para automatização dos testes realizados pelo operador, atuando em conjunto com o mesmo, de forma a apresentar um resultado claro que o oriente na destinação adequada da câmera. Ademais, o operador deve acompanhar visualmente a execução dos testes, atentando-se para eventuais questões não previstas.

Além disso, deseja-se aumentar o desempenho do teste, atingindo indicadores maiores que 90% para a métrica F1-score em todos os cinco testes. A escolha dessa métrica se dá devido ao desbalanceamento dos dados e a possibilidade de avaliação simultânea do desempenho em termos do tipo de erro cometido pelo modelo - Falso Positivo ou Falso Negativo. Pretende-se padronizar o resultado dos testes, removendo a subjetividade na interpretação de defeitos e a divergência que ocorre entre diferentes operadores.

Espera-se reduzir o tempo de duração total dos testes realizados pelo operador, em pelo menos 20%, ou seja, de 27 segundos para menos de 21.6 segundos.

O armazenamento dos dados da produção de forma padronizada para a integração futura com outras plataformas de análise também compõe os objetivos deste trabalho.

Ainda, tem-se o objetivo de adotar um procedimento de teste que tenha a execução garantida para todas as câmeras, aplicando-se um mecanismo de marcação automático aos produtos que indique a aprovação por parte do sistema.

1.3 DESCRIÇÃO DA EMPRESA

Nesta seção é feita uma descrição acerca da Intelbras, empresa em que o PFC foi realizado, introduzindo-se o contexto de suas atividades e algumas das soluções desenvolvidas.

A Intelbras S.A. Indústria de Telecomunicação Eletrônica Brasileira iniciou suas operações em 1976, em São José, desenvolvendo centrais telefônicas PABX. Entretanto, com o passar dos anos, as atividades da empresa se expandiram consideravelmente para outras áreas, assumindo frequentemente o papel de referência nas soluções entregues. Atualmente, sua estrutura é dividida em cinco unidades de negócio com um portfólio de produtos variado, são elas:

- Segurança Eletrônica: Focando em soluções de monitoramento, proteção e inibi-

ção de ações criminosas, através de soluções como: alarmes, câmeras (analógicas, IP, Wi-Fi), gravadores, entre outros;

- Comunicação: Visando a garantia de qualidade e eficiência da comunicação, por meio de produtos como: antenas, conversores, telefones, Headsets, Webcams;
- Energia: Facilitando conexões, promovendo a proteção de outros equipamentos, garantindo a manutenção de energia. Destacam-se: baterias, protetores de surto, fontes e Nobreaks;
- Energia Solar: Entregando alternativas sustentáveis e de eficiência energética, como, por exemplo: módulos fotovoltaicos, inversores, geradores;
- Redes: Oferecendo soluções de maior desempenho e robustez na conexão de dispositivos, através de: roteadores, *switches* e repetidores.

As atividades realizadas pelo autor se deram na unidade Matriz, em São José, visando a implementação de melhorias na filial de Manaus. Além das unidades citadas (Figura 2), existem outras três filiais: Filial São José, Filial Minas Gerais e Filial Nordeste, em Pernambuco. Ainda, atividades de pesquisa e inovação também são realizadas por mais de 60 colaboradores na China.

Figura 2 – Matriz e filiais.



Fonte: Adaptado de Logismat, Ado Rafael Evangelista

1.4 ESTRUTURA DO DOCUMENTO

Nesta seção apresenta-se uma visão geral do documento, descrevendo-se brevemente o conteúdo apresentado nos próximos capítulos, a fim de facilitar a compreensão do leitor acerca da estrutura deste trabalho.

No segundo capítulo são introduzidos os conceitos teóricos essenciais para uma melhor contextualização das atividades desempenhadas durante o PFC, abordando-se os trabalhos anteriores que motivaram este projeto, o funcionamento básico de uma câmera analógica, a escolha da métrica de desempenho utilizada e explicações sobre o funcionamento de redes neurais.

Na sequência, no terceiro capítulo, é apresentada a modelagem do sistema em termos de requisitos, diagramas, esquemas, entre outros. Assim, procura-se documentar e planejar o desenvolvimento do trabalho conforme as metodologias de modelagem propõem.

O desenvolvimento efetivo é exposto no capítulo quatro, tratando-se dos detalhes de implementação e das soluções adotadas.

Em seguida, no capítulo cinco abordam-se os resultados obtidos, bem como os impactos do trabalho em termos de reduções de custos e melhoria de processos.

Por fim, na conclusão é feita uma revisão geral do trabalho, ressaltando-se os principais pontos e possibilidades para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os trabalhos que motivaram este Projeto de Fim de Curso, contextualizando-se acerca do escopo em que o projeto está inserido. Além disso, são introduzidos os conceitos, teorias e modelos adotados no processo de desenvolvimento do sistema inteligente de automação de testes - também referido neste trabalho como Jiga de Testes.

2.1 TRABALHOS ANTERIORES

O projeto de Estágio Obrigatório realizado pelo autor (QUINT, 2022) pode ser entendido como parte fundamental dos primeiros passos para a compreensão e definição das estratégias adotadas neste trabalho. Através dele é que se pôde entender os requisitos de hardware, estudar a forma mais otimizada de sequenciamento de testes, definir o dimensionamento da estrutura da Jiga de testes e escolher o método de alternância de cenários.

Essas etapas foram importantes para extrair os requisitos do problema que não são de fácil visualização no primeiro momento, ou seja, ao início do Projeto de Fim de Curso, já se tinha um protótipo inicial para servir de arcabouço do projeto definitivo. Assim, a definição da metodologia a ser adotada, o entendimento das particularidades do problemas a ser resolvido e dos detalhes de implementação a serem tratados, ocorreram de forma mais clara e certa.

Diversas modificações foram realizadas em relação a ideia inicialmente construída para o primeiro protótipo. O trabalho desenvolvido foi importante para validar o conceito principal, entender a visão do time da fábrica, coletar *feedbacks* e identificar problemas, além de servir como forma de convencimento para investimento de recursos e seguimento do projeto.

Além disso, o trabalho realizado pelo colega de curso Eduardo Camilo (CANTO, 2022) permitiu avaliar a viabilidade de aplicação de uma solução de Inteligência Artificial, através de Redes Neurais Convolucionais, para este projeto. O trabalho citado soluciona o problema de identificação de mancha escura na imagem de câmeras durante a etapa de produção "Ajuste de Foco", enquanto que este trabalho refere-se à automatização de cinco procedimentos de testes realizados durante a etapa "Montagem Final", incluindo a identificação de mancha escura. Assim, os resultados obtidos para o mesmo contexto de produção de câmeras motivaram o estudo da solução por meio de um modelo de classificação construído a partir de uma arquitetura de Redes Neurais Residuais.

2.2 FUNCIONAMENTO DAS CÂMERAS ANALÓGICAS

Visando possibilitar a compreensão dos testes que são aplicados às câmeras produzidas, faz-se necessária a introdução do funcionamento básico das mesmas. Ainda, em paralelo aos conceitos apresentados, contextualiza-se o papel do componente tratado com o teste correspondente feito pelo sistema.

As câmeras analógicas tratadas neste trabalho são comumente aplicadas em soluções de segurança eletrônica por meio de Circuitos Fechados de Televisão (CFTV), sendo conectadas a um dispositivo gravador Direct Video Recorder (DVR - Figura 3), através de um cabo coaxial. A partir da saída desse dispositivo pode-se acessar as imagens geradas, transmitindo-as para um monitor, por exemplo.

Figura 3 – Câmera VHD 3130 B e DVR Intelbras MHDX3104.



Fonte: Adaptado de www.intelbras.com

Pode-se dividir as câmeras em diferentes categorias (Figura 4), mas, no contexto deste trabalho, o foco é dado nos modelos correspondentes aos cases de maior volume de produção: Bullet e Dome.

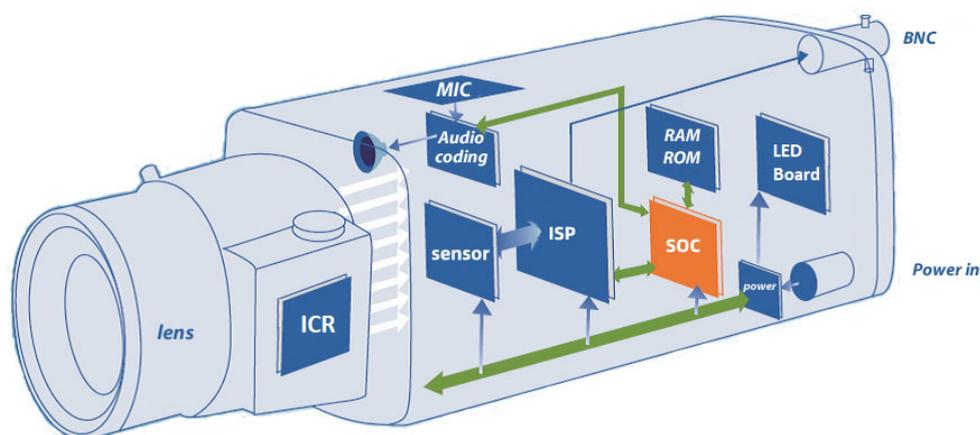
Figura 4 – Tipos de câmeras produzidas pela Intelbras.



Fonte: www.blog.intelbras.com.br

Os principais componentes de uma câmera analógica são apresentados no diagrama da Figura 5.

Figura 5 – Esquema de componentes para câmera analógica.



Fonte: Adaptado de www.ipsecu.com

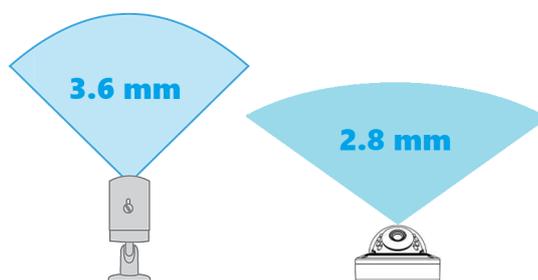
Apesar de não representar fielmente a disposição dos componentes que é empregada nas câmeras produzidas e até mesmo o formato dos *cases* utilizados, o diagrama auxilia no processo de descrição do funcionamento das mesmas. A seguir é feita uma explicação dos principais componentes associando-os ao teste correspondente, feito pelo sistema proposto.

2.2.1 Lentes e ICR

As lentes convergentes têm o papel de concentrar os raios luminosos provenientes de uma determinada cena sobre o sensor de imagem.

No contexto do sistema de automatização de testes, as diferenças no ângulo de abertura dado pela lente desses produtos e na altura dos mesmos são relevantes para a construção de um suporte único - por tipo de *case* - que atenda todos os modelos, evitando a realização de trocas frequentes (*setup*) na linha de produção. Na Figura 6 pode-se observar as implicações em termos de campo de visão das duas principais medidas de distância focal das lentes utilizadas nas câmeras Intelbras. Vale ressaltar que o tamanho do sensor de imagem também influencia o campo de visão da câmera. Assim, o suporte das câmeras deve compensar a diferença de altura entre os diferentes modelos, garantindo que o campo de visão cubra somente a área de interesse para os testes realizados.

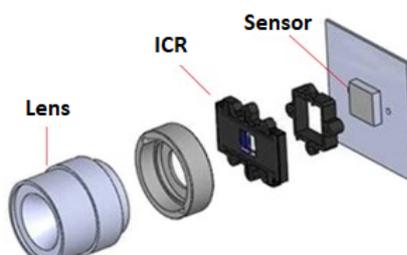
Figura 6 – Comparação qualitativa entre ângulos nos modelos Bullet e Dome.



Fonte: Adaptado de Lorex Technology

Entretanto, em cenários com luminosidade suficiente, um filtro denominado ICR (*Infrared Cutfilter Removal*) é aplicado entre esta trajetória para que a incidência de luz infravermelha sobre o sensor de imagem seja limitada, evitando a distorção das cores percebidas (LEARN CCTV, 2018). No modo noturno, com a baixa incidência de luz, é dado um comando para que uma válvula solenoide atue na remoção desse filtro e se possa capturar o máximo de sinais. A estrutura do conjunto citado pode ser observada na Figura 7.

Figura 7 – Visão explodida do conjunto lentes, ICR e sensor de imagem.



Fonte: Adaptado de www.e-consystems.com

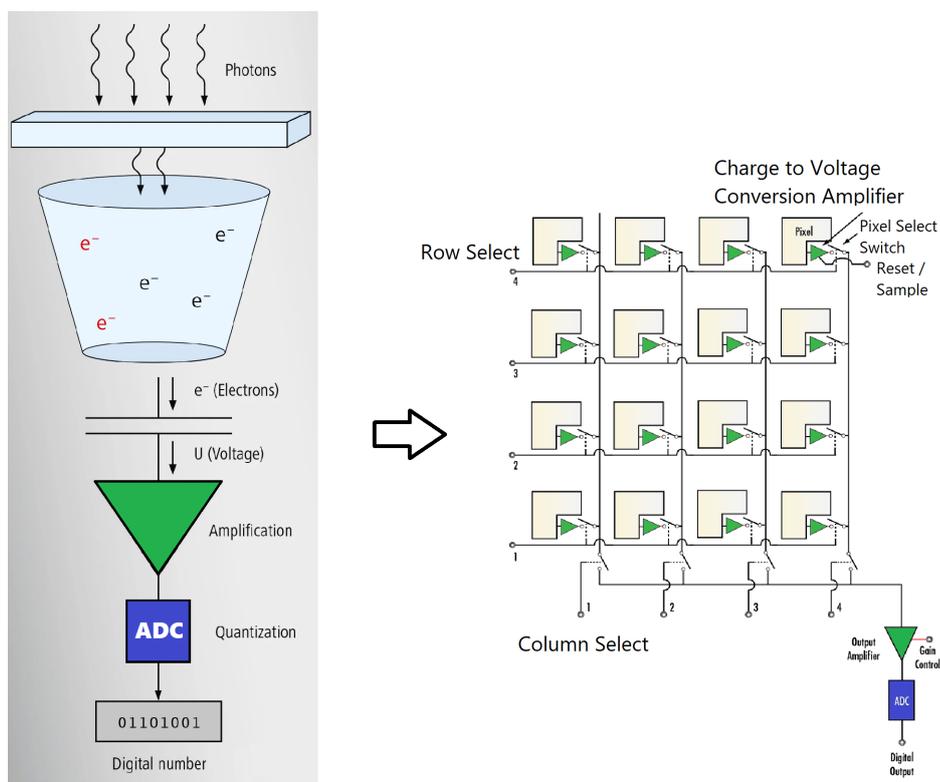
O mau funcionamento do ICR é percebido pela formação de imagens com aspecto rosado quando a câmera opera em modo Diurno. Assim, esse comportamento característico é utilizado como o principal indicativo de defeito durante um dos testes executados na etapa de Montagem Final.

2.2.2 Sensor de Imagem e Processador

Como sensor de imagem costuma-se empregar um componente do tipo CMOS (Complementary Metal-Oxide semiconductor - Figura 8), cuja função principal é converter o sinal luminoso em sinal elétrico (EDMUND, 2022).

A luz que incide sobre o sensor é armazenada através de matrizes compostas por fotodiodos (ou fotocapacitores), denominadas pixels. Eles podem ser entendidos

Figura 8 – Esquema de funcionamento CMOS.



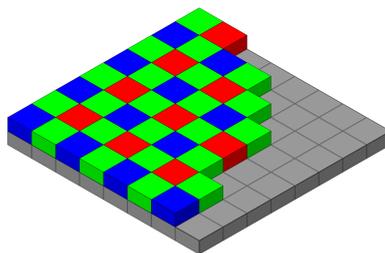
Fonte: Adaptado de Baumer e Edmund Optics

como reservatórios de fótons que se carregam proporcionalmente à incidência de luz, sendo convertidos em tensão e amplificados através de um circuito composto tipicamente por um fotodiodo e três transistores. Em seguida, o sinal gerado em cada pixel é multiplexado em linhas e colunas, conforme o esquema à direita da Figura 8, para ser enviado a um circuito conversor analógico-digital (ADC - Analog to Digital Converter).

Cabe ao ISP (Image Signal Processor) tratar o sinal digital recebido, interpretando as cores da imagem, além de corrigir o balanço de branco, a exposição e o foco (KLUG, 2013). Na prática cada pixel do sensor de imagem armazena a informação de apenas uma cor, não um sinal em RGB (Red, Green and Blue). Assim, utiliza-se um filtro de cores sobre o sensor - no padrão *Bayer* (Figura 9 - para que a interpolação das cores adjacentes possa ser calculada pelo algoritmo de interpretação, em um processo denominado *demosaicing*. Ainda, na saída do ISP tem-se um conversor DAC (Digital to Analog Converter) que entrega o sinal analógico característico do produto.

O Teste de Cor executado durante a etapa de Montagem Final, realizada na fábrica em Manaus, visa identificar distorções na cor apresentada pelas câmeras, possivelmente causadas por defeitos no sensor de imagem. Ainda, o Teste de Mancha visa identificar a presença de manchas nas imagens, que são ocasionadas pela deposição

Figura 9 – Esquema: filtro de Bayer sobre sensor de imagem.

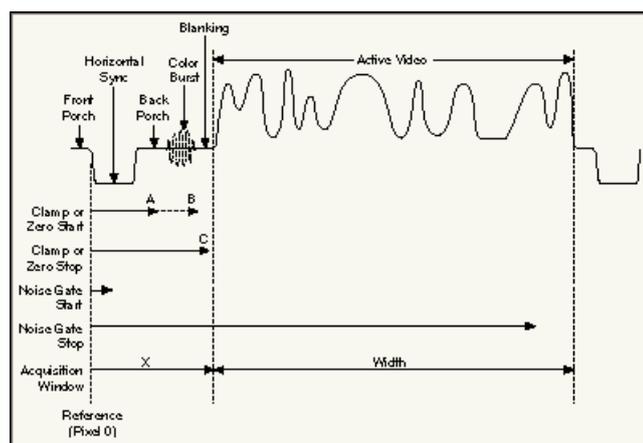


Fonte: Autor: Colin M.L. Burnett

de partículas sólidas sobre o sensor de imagem.

Além da conversão, o sinal de saída de uma câmera ainda precisa ser processado para se adequar ao padrão de operação de outros dispositivos (NTSC - *National Television Standard Committee*), possibilitando a visualização da imagem correspondente à cena que o gerou. Em um monitor apresentando o sinal de uma câmera, o vídeo é composto por um conjunto de quadros tal que a imagem apresentada é dividida em linhas de pixels (KAPATKER, 1997). Ao fim de cada linha tem-se um pulso, chamado de pulso de sincronismo horizontal, que indica o seu término e sinaliza aos demais dispositivos eletrônicos o início de uma nova linha. Além disso, tem-se o *burst de cor*, para a sincronização dos padrões de cor, e o sinal da imagem propriamente dito (Figura 10).

Figura 10 – Sinal de uma Câmera Analógica



Fonte: www.tistory.sunnyan.kr

Quando o nível de luminosidade percebido pelo ISP atinge um determinado patamar mínimo especificado, a placa LED comanda o acionamento de LEDs infravermelhos ou de luz branca, visando manter possível a visualização das imagens. O Teste de LED, realizado na Montagem Final, verifica a integridade dessa funcionalidade

através da submissão da câmera a um ambiente escuro que induz o acionamento dos LEDs. Com exceção dos modelos de câmeras Full Color, espera-se que durante esse teste a imagem capturada fique em preto e branco e os LEDs do produto mantenham-se ativos.

2.3 MÉTRICAS DE DESEMPENHO

Avaliar o desempenho do sistema de testes proposto é fundamental para entender os ganhos reais proporcionados. Embora amplamente utilizada, a métrica acurácia - Equação (1), por exemplo, de forma isolada pode levar a conclusões equivocadas, pois não considera o balanceamento dos dados (BROWNLEE, 2020).

$$ACC = \frac{VP + VN}{VP + VN + FP + FN} \quad (1)$$

VP (Verdadeiro Positivo) e VN (Verdadeiro Negativo) referem-se aos dados que foram classificados corretamente como pertencentes ao conjunto “Defeito” e “OK” respectivamente, enquanto que FP (Falso Positivo) e FN (Falso Negativo) representam os dados classificados incorretamente como “Defeito” e “OK” respectivamente.

Neste problema, a incidência de câmeras com defeito representa cerca de 4% do total produzido. Assim, um modelo hipotético que aprovasse todas as câmeras teria acurácia de 96%, ou seja, o valor elevado atingido nessa métrica induz à conclusão de que esse modelo seria útil.

A disposição dos dados em uma matriz de confusão (Figura 11) permite uma visualização mais clara da configuração dos mesmos. Além disso, derivam dela outras métricas que expressam de forma mais significativa o desempenho de um modelo classificador para o problema tratado neste trabalho.

Figura 11 – Matriz de confusão

	Real Positivo	Real Negativo
Predito Positivo	Verdadeiro Positivo	Falso Positivo
Predito Negativo	Falso Negativo	Verdadeiro Negativo

Fonte: Autoria própria

A Precisão - Equação (2) - pode ser entendida como a medida dos casos identificados corretamente como positivos/defeituosos em relação a todas as predições positivas, sendo particularmente útil quando o custo de um Falso Positivo é alto.

$$P = \frac{VP}{VP + FP} \cdot \quad (2)$$

Já a *Recall* (Revocação) - Equação (3) - pode ser entendida como a medida dos casos identificados corretamente como positivos/defeituosos em relação a todos os casos realmente positivos. Utiliza-se principalmente quando o custo de ocorrência de Falsos Negativos é alto. Esta métrica se mostra mais relevante, pois a aprovação de câmeras defeituosas é mais custosa que a reprovação indevida de câmeras boas. Ainda que ambos casos sejam indesejados, no primeiro tem-se o risco de bloqueio de todo o lote para reinspeção ou de entrega desse produto a um cliente, enquanto que no segundo apenas será retrabalhado um produto.

$$R = \frac{VP}{VP + FN} \cdot \quad (3)$$

Com o intuito de captar ambos os casos, utiliza-se a métrica F1-Score, apresentada na Equação (4) como a média harmônica entre a Precisão e a Revocação.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \cdot \quad (4)$$

No exemplo hipotético citado anteriormente, a *Recall* e conseqüentemente a F1-score seriam iguais a zero se o modelo sempre classificasse todos os dados como negativos.

2.4 REDES NEURAIS

O processo de classificação da imagem de uma câmera submetida a cada um dos testes no conjunto “Defeito” ou no conjunto “Correto”, também tratado neste trabalho como “NOK” ou “OK” respectivamente, se dá através da utilização de um modelo construído - a partir do trabalho de um colega de equipe - por meio de Redes Neurais Convolucionais.

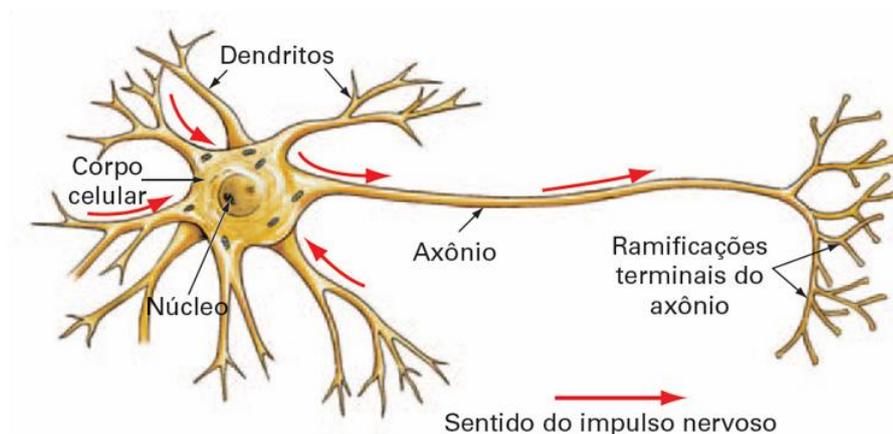
A seguir é feita uma explicação dos conceitos gerais que contemplam o tópico Redes Neurais. A ideia é esclarecer o seu funcionamento básico e justificar a sua utilização neste trabalho. Em seguida, abordam-se os conceitos associados ao funcionamento de uma Rede Neural Convolucional e da arquitetura *ResNet18*.

A estratégia adotada na criação das Redes Neurais Artificiais baseia-se no funcionamento do cérebro humano, imitando a forma como a informação é transmitida entre os neurônios (IBM, 2020).

O neurônio biológico (Figura 12) pode ser dividido em três principais partes: dendritos, corpo celular e axônio (ACADEMY, 2022). Os dendritos recebem os impulsos nervosos provenientes de outros neurônios, direcionando-os ao corpo celular, onde são processados e geram-se novos impulsos. Se esses sinais forem superiores a um limiar de disparo (*threshold*), então eles percorrem a ramificação maior denominada

axônio e são transmitidos para outros neurônios através de suas terminações, em um processo conhecido como sinapse.

Figura 12 – Representação Simplificada do Neurônio Biológico

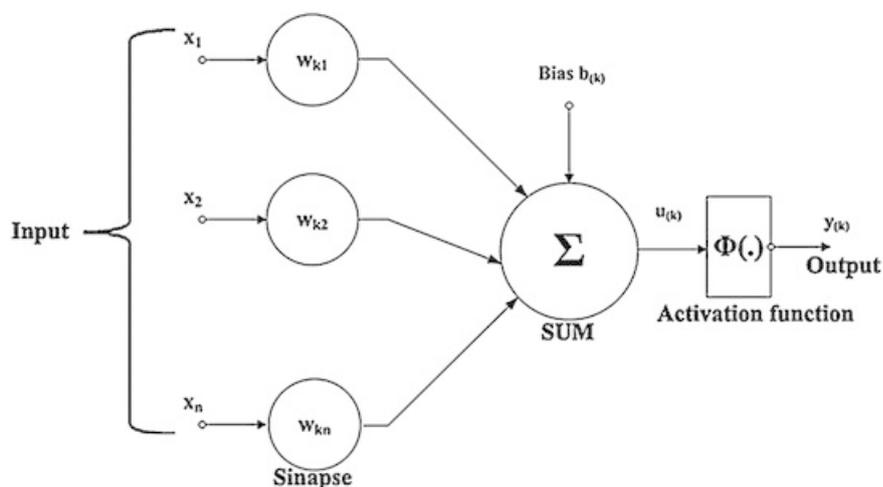


Fonte: www.deeplearningbook.com.br

O sinal proveniente de diferentes dendritos é transmitido com um peso sináptico diferente, denominado memória, dependendo do dendrito de origem. Esse peso é ajustado por meio do treinamento realizado pelo cérebro ao longo da vida.

Com base no funcionamento simplificado de Redes Neurais Biológicas apresentado é que Warren McCulloch e Walter Pitts desenvolveram o modelo mais aceito de um neurônio matemático, em 1943.

Figura 13 – Representação Simplificada do Neurônio Artificial



Fonte: www.deeplearningbook.com.br

Na Figura 13 apresenta-se a estrutura de um neurônio matemático, tal que:

- Sinais de entrada (x_1, x_2, \dots, x_n): Referem-se aos dados que são fornecidos ao modelo de predição, geralmente normalizados para aumentar a eficiência da aprendizagem.
- Pesos sinápticos (w_1, w_2, \dots, w_n): Valores aprendidos durante o treinamento de uma rede que definem o quanto do sinal de entrada será passado adiante.
- Bias (b): Variável incluída no somatório para aumentar o grau de liberdade da função e possibilitar a entrega de uma saída não-nula mesmo que todas as entradas sejam nulas.
- Combinador Linear (Σ): Agrupa os valores provenientes do produto entre os sinais de entrada e os pesos sinápticos correspondentes.
- Potencial de ativação (u): É dado pela diferença entre o valor calculado pelo Combinador Linear e o limiar mínimo para disparo de ativação.
- Função de ativação (Φ ou g): Utiliza-se uma função não-linear crescente para ampliar a capacidade de representação. Tem o papel de limitar a saída do neurônio em um determinado intervalo.
- Sinal de saída (y): Representa o valor de saída de um neurônio, comumente conectado à entrada de outro neurônio visando a formação de uma rede.

Assim, o corpo celular é representado pela multiplicação do somatório dos estímulos (entradas x) pelos pesos w juntamente com a função de ativação, que é responsável por definir qual a saída do neurônio. A saída y simboliza o axônio, o qual transmite o sinal processado.

A função de ativação deve garantir que não ocorra o somatório indefinido dos sinais ao longo das camadas da rede, limitando a saída do neurônio. Além disso, a derivabilidade da função de ativação é essencial para o processo de aprendizagem de uma rede (WANGENHEIM, 2004), apresentado posteriormente nesta seção. Ainda, a escolha de funções de ativação lineares impossibilita a representação de padrões linearmente inseparáveis (Anexo - A) e, por esse motivo, costumam-se utilizar funções com o comportamento similar ao da tangente hiperbólica, apresentada na Equação (5). Além disso, segundo o teorema da aproximação universal, basta uma camada oculta para que a rede neural seja capaz de aproximar qualquer função contínua, o que implica na inutilização de um maior número de camadas internas em uma rede dessa topologia.

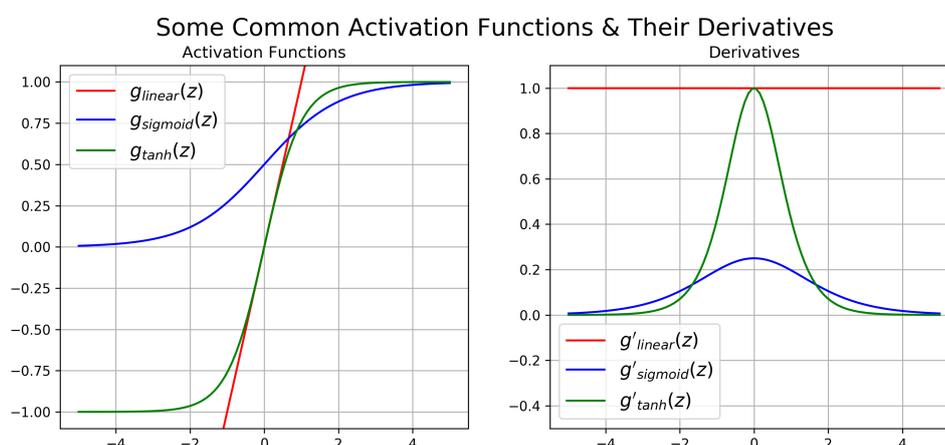
$$g(x) = \tanh(\beta \cdot x). \quad (5)$$

Aplicando-se a regra do quociente para o cálculo da derivada, obtém-se:

$$g'(x) = \beta \cdot (1 - g^2(x)). \quad (6)$$

Note que a tangente hiperbólica, em verde na Figura 14, compõe o cálculo da própria derivada. Isso facilita a realização de cálculos computacionais, já que a equação será utilizada recursivamente no cálculo de trás para frente dos erros, durante o processo de ajuste dos pesos de cada conexão. Ademais, a função possui comportamento assintótico para entradas muito pequenas ou muito grandes, e comportamento aproximadamente linear em valores mais próximos à origem. Essas características são desejáveis, pois limitam o crescimento abrupto dos pesos para entradas de maior magnitude.

Figura 14 – Plot de diferentes funções de ativação e suas derivadas



Fonte: dustinstansbury.github.io

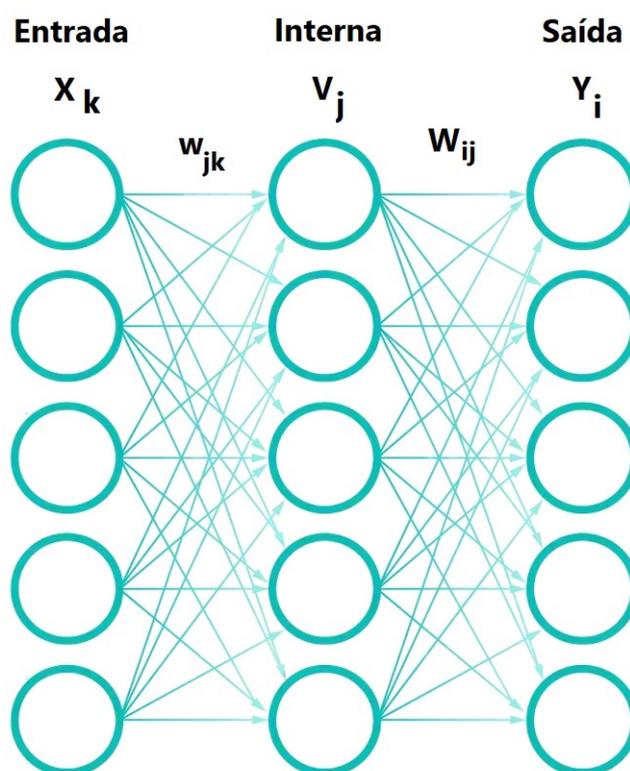
Outra função de ativação amplamente utilizada é a ReLU (Rectified Linear Unity), definida na Equação (7).

$$f(x) = \max(0, x). \quad (7)$$

Pode-se perceber que a ReLU também atende o requisito de não-linearidade e é amplamente utilizada por permitir a ativação de alguns neurônios isoladamente, já que essa função retorna valores nulos nos casos de entradas negativas. Assim, com essas características tem-se uma rede esparsa, o que pode ser útil em termos de eficiência computacional. Ainda, o problema de Gradientes Evanescentes, apresentado posteriormente, tende a ser mais frequente em topologias que adotam a tangente hiperbólica em relação as que adotam a ReLU. Além disso, o cálculo da região linear é menos custoso computacionalmente que o cálculo da tangente hiperbólica. Essa função é derivável parcialmente, separando-a na região constante em 0 e na região

linear, o que possibilita o cálculo do algoritmo de Descida em Gradiente, apresentado posteriormente nesta seção.

Figura 15 – Diagrama de uma Rede Neural



Fonte: Adaptado de www.ibm.com

Na Figura 15, cada círculo representa um neurônio, associado a um peso, viés e função de ativação. A informação da camada de entrada é processada e passada para as camadas seguintes através de transformações lineares (pela multiplicação de pesos) e não-lineares (pela função de ativação) em um processo denominado propagação direta. Em seguida, aplica-se o *backpropagation*, ou seja, compara-se o valor de saída com o valor esperado, calculando-se um erro que é utilizado de trás para frente para corrigir os pesos da rede e os bias de cada camada, com o intuito de minimizar o erro.

Um dos algoritmos de otimização responsáveis por encontrar os parâmetros w (peso) e b (bias) que minimizam uma função custo associada aos erros é chamado de Gradiente Descendente (ou Descida em Gradiente).

A Equação (8) apresenta uma possível função custo baseada na soma dos quadrados dos erros entre a saída esperada e a calculada. Vale ressaltar que o desenvolvimento apresentado na sequência baseou-se nas notas de aula da disciplina de Visão Computacional (WANGENHEIM, 2004).

$$E(w) = \frac{1}{2} \sum_{i\mu} (Yr_i^\mu - Y_i^\mu)^2, \quad (8)$$

na qual:

- $E(w)$ corresponde à função custo;
- i é o índice referente a cada neurônio da camada de saída;
- μ refere-se ao padrão de treinamento apresentado à camada de entrada;
- Yr^μ é o vetor referência de saída, ou seja, o vetor de saída esperado (para o padrão μ);
- Yr_i^μ é o valor de saída esperado (para o padrão μ);
- Y_i^μ é o sinal de saída (para o padrão μ) referente ao neurônio 'i' da camada de saída.

Entretanto, deseja-se uma expressão para a função custo que permita o cálculo da Descida em Gradiente para três camadas, não duas. Assim, para a dedução dessa expressão deve-se propagar o sinal até se obter o vetor Y_i de saída da rede.

Com base na Figura 13, obtém-se a expressão para a saída V_j^μ de um neurônio 'j' da camada interna, em termos dos pesos e das entradas.

$$V_j^\mu = g\left(\sum_k w_{jk} \cdot X_k^\mu\right), \quad (9)$$

tal que o índice 'j' refere-se a cada neurônio da camada interna e X_k^μ corresponde à entrada dessa camada. Propagando-se a ativação novamente para a camada de saída da rede e aplicando a expressão obtida para V_j^μ , obtém-se:

$$Y_i^\mu = g\left(\sum_j W_{ij} \cdot V_j^\mu\right) = g\left(\sum_j W_{ij} \cdot g\left(\sum_k w_{jk} \cdot X_k^\mu\right)\right), \quad (10)$$

onde W_{ij} refere-se aos pesos que conectam a camada interna com a camada de saída e V_j^μ refere-se ao sinal de saída do neurônio j da camada interna.

Substituindo a Equação (10) na Equação (8), obtém-se a seguinte expressão diferenciável para a função custo:

$$E(w) = \frac{1}{2} \left(Yr_i^\mu - \sum_{i\mu} \left(g\left(\sum_j W_{ij} \cdot g\left(\sum_k w_{jk} \cdot X_k^\mu \right) \right) \right) \right)^2. \quad (11)$$

O cálculo da derivada parcial dessa função é importante para entender a variação do erro em função dos pesos, apontando na direção de maior crescimento dos erros. Assim, desloca-se no sentido contrário com o intuito de encontrar o ajuste

adequado dos pesos que minimizam essa função. A ideia, portanto, é encontrar a regra-delta que descreve a equação de adaptação dos pesos. Segue:

$$\Delta W_{i,j} = -\eta \cdot \frac{\partial E}{\partial W_{i,j}} = \eta \cdot \sum_{\mu} (Yr_i^{\mu} - Y_j^{\mu}) \cdot g'(h_i^{\mu}) \cdot V_j^{\mu}. \quad (12)$$

O parâmetro η é chamado de taxa de aprendizado e indica o ritmo com que os pesos são atualizados. Já o parâmetro h_i^{μ} refere-se à entrada total do neurônio i . Rescrevendo-se em função de δ , tem-se:

$$\Delta W_{i,j} = \eta \cdot \sum_{\mu} \delta_i^{\mu} \cdot V_j^{\mu}. \quad (13)$$

onde:

$$\delta_i^{\mu} = (Yr_i^{\mu} - Y_j^{\mu}) \cdot g'(h_i^{\mu}). \quad (14)$$

A Equação (14) é utilizada para a última camada da rede (M) em um algoritmo de *backpropagation*. Nas camadas anteriores deve-se utilizar as seguintes expressões:

$$\Delta w_{p,q}^m = \eta \cdot \sum_{\mu} \delta_p^{m,\mu} \cdot V_q^{m-1,\mu}. \quad (15)$$

$$\delta_p^{m,\mu} = \sum_r (w_{r,p}^{m+1} \cdot \delta_r^{m+1,\mu}) \cdot g'(h_p^{m,\mu}), \quad (16)$$

em que $m = 0, \dots, M$ refere-se a cada camada e os índices genéricos q, p, r referem-se a cada neurônio de camadas anteriores, atuais e subsequentes. O desenvolvimento para a obtenção das Equações 15 e 16 foi omitido por se distanciar do propósito deste trabalho.

Então, o algoritmo para o treinamento de uma rede *backpropagation* pode ser sumarizado nas seguintes etapas:

1. Inicializam-se os pesos das camadas 'm' com valores pequenos e aleatórios (ou utiliza-se uma heurística de inicialização apropriada);
2. Escolhe-se um padrão μ do conjunto de treinamento, apresentando-o para todos os neurônios de camada de entrada;
3. Propaga-se a função de ativação até que o cálculo seja feito na camada de saída;
4. Calculam-se os δ para camada de saída, através da Equação (14);
5. Percorrem-se as camadas anteriores, aplicando-se a retropropagação dos erros com a Equação (16);
6. Calcula-se a variação dos pesos Δw

7. Ajustam-se os novos pesos das conexões, por $w^m = w^m + \Delta w^m$;
8. Retorna-se ao segundo passo, para outro padrão de treinamento.

Ao fim dessas etapas, se o conjunto de dados for representativo, a rede aprende uma função que mapeia a entrada à saída, quando existente.

2.4.1 Redes Neurais Convolucionais

A primeira apresentação bem estabelecida das Redes Neurais Convolucionais (CNN) se deu com o artigo “*Gradient-based learning applied to document recognition*” (LECUN *et al.*, 1998), abordando-se o problema de reconhecimento de caracteres. Desde então essa estratégia tem sido utilizada nas mais diversas aplicações envolvendo visão computacional.

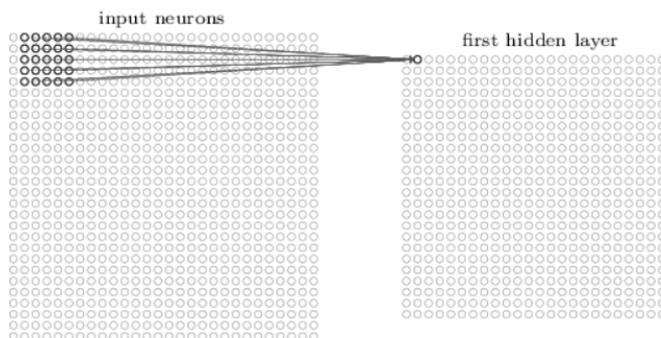
Tratando-se do problema de classificação de imagens, objeto de estudo deste trabalho, o uso de diferentes arquiteturas em que a rede não está totalmente conectada (como ocorria na *Feed Forward* tradicional, apresentada anteriormente) passa a fazer mais sentido. Isso se deve ao fato de o contexto dos pixels adjacentes ser relevante, e o tratamento de pixels distantes da mesma forma acaba não sendo tão eficiente.

Assim, pode-se entender uma Rede Neural Convolucional como um algoritmo de Aprendizado Profundo que atribui diferentes relevâncias para aspectos de uma imagem de entrada, condensando a informação através de filtros de convolução que facilitam o processamento voltado para imagens.

A utilização de imagens nesse contexto se dá através da representação em termos de matrizes em que as entradas referem-se à intensidade de cada pixel em um determinado espaço de cor, como RGB. Assim, a simples transformação de matrizes em vetores, por exemplo: 3x3 em 9x1, para aplicação em uma Rede Neural totalmente conectada, seria pouco eficiente em termos de processamento, já que não considera as dependências entre pixels. Além disso, em imagens existem variações, como: translações e rotações, as quais geram padrões diferentes que teriam que ser aprendidos para uma rede neural tradicional.

Desse modo, a ideia por trás de uma CNN consiste em conectar os neurônios através de campos receptivos locais (também chamado de matriz de convolução ou kernel), em que cada neurônio da camada convolucional se conecta a uma região de pixels da camada de entrada. A Figura 16 ilustra o conceito da conexão entre camadas em uma Rede Neural Convolucional para uma imagem 28x28 e um campo receptivo 5x5, que se desloca pela imagem com passo (stride) unitário, implicando em 24x24 neurônios na camada seguinte. Essa perda de resolução ocorre pois o kernel colide com a borda direita e com a borda inferior quando atinge os limites da imagem de entrada. Com isso, um neurônio da primeira camada oculta (camada convolucional) aprende 25 pesos, além do bias, para as conexões que compõem essa matriz 5x5.

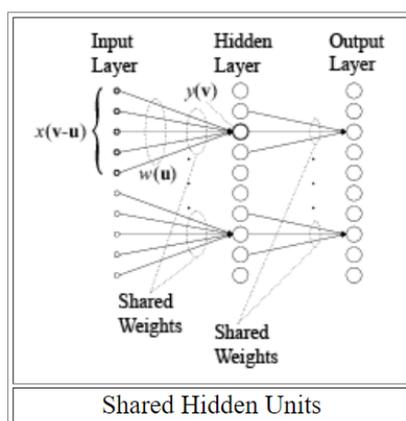
Figura 16 – Diagrama para conexão de camadas em uma CNN



Fonte: Deep Learning Book

Na prática, no processo de retropropagação dos erros, aprende-se a mesma matriz de convolução que relaciona cada região 5x5 da imagem de entrada com a próxima camada, permitindo a detecção do mesmo padrão em diferentes locais da imagem. Assim, os pesos que compõem essas matrizes iguais são chamados de pesos compartilhados, conforme apresentado na Figura 17. Isso faz sentido, pois a detecção de uma borda, por exemplo, pode ser útil tanto no canto esquerdo como no canto direito de uma imagem, independente da rotação.

Figura 17 – Compartilhamento de pesos entre conexões em uma CNN

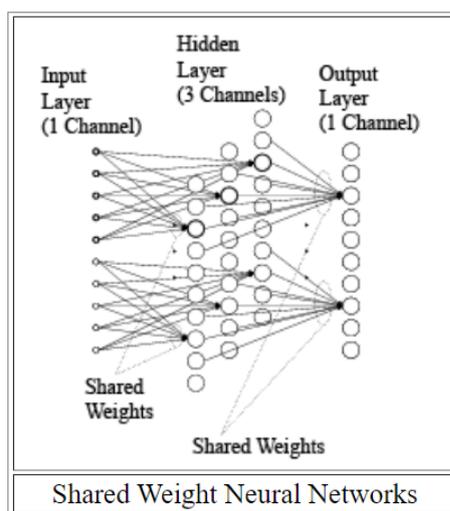


Fonte: (OTUYAMA, 2009)

Ainda, no processamento de imagens coloridas as camadas internas de uma Rede Neural Convolucional são geralmente tridimensionais, ou seja, são compostas por planos de neurônios. Essa estrutura pode ser observada na Figura 19, na qual é apresentada a arquitetura da Le-Net5, proposta por Yann LeCun no contexto de identificação de dígitos. Assim, aprende-se uma matriz diferente que mapeia a entrada

para cada plano, composta por um conjunto de pesos compartilhados 5x5 e um bias compartilhado. A Figura 18 ilustra a construção de três planos de forma menos abstrata. Neste exemplo, a rede seria capaz de identificar três características diferentes, em toda a imagem.

Figura 18 – Compartilhamento de pesos entre conexões 3D em uma CNN



Fonte: (OTUYAMA, 2009)

Com isso, a vantagem do uso de operadores de convolução se manifesta também através da redução no número de parâmetros de uma rede. Já que, em cada um dos seis planos da Le-Net5, cada matriz é composta por 26 parâmetros (5x5 pesos compartilhados acrescidos de um bias), ou seja, existem 156 (6x26) parâmetros no mapeamento entre a entrada e a primeira camada. Enquanto que em uma rede totalmente conectada tem-se 784 (28x28) neurônios de entrada associados a, por exemplo, 10 neurônios da camada oculta com 10 bias, resultando em 7850 (784x10+10) parâmetros. A comparação não considera a equivalência de desempenho entre as redes. Porém, através dela, pode-se compreender as diferenças na magnitude de cada estrutura em termos de número de conexões.

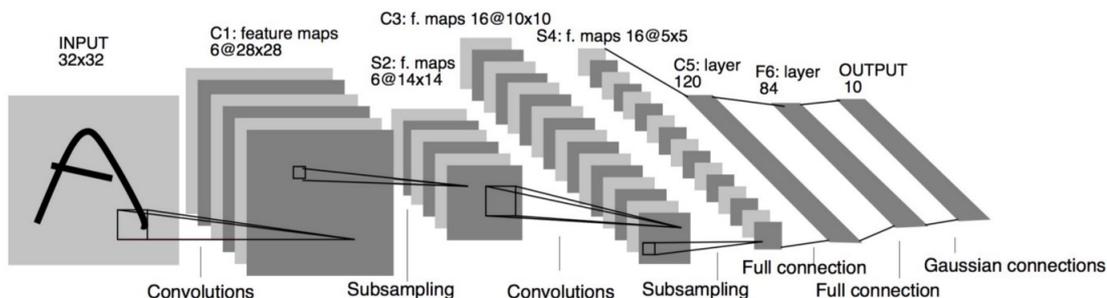
O cálculo de propagação do sinal em uma Rede Neural Convolutional, com kernel $m \times m$, é apresentado na Equação (17)

$$y^l = \sigma\left(\sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} \cdot y_{i+a, j+b}^{l-1}\right), \quad (17)$$

em que y^{l-1} é a saída do neurônio da camada anterior, w_{ab} refere-se aos pesos compartilhados que compõem cada posição da matriz (kernel), i refere-se ao índice do neurônio da camada atual, j refere-se ao índice do neurônio da camada anterior,

ou seja, a saída do neurônio j é conectada à entrada do neurônio i . σ é a função de ativação aplicada à saída do neurônio da camada atual.

Figura 19 – Arquitetura da Le-Net5 para identificação de dígitos



Fonte: (LECUN *et al.*, 1998)

Após o cálculo das operações de convolução, aplica-se na saída da camada convolucional, uma operação denominada *subsampling*, que consiste em reduzir a resolução dessa saída e simplificar as informações obtidas. Em geral, utiliza-se a operação *Max-Pooling*, que escolhe a maior ativação em uma região 2x2 da camada de convolução, por exemplo, e a resume em uma unidade da camada seguinte, descartando as demais ativações. Portanto, com a aplicação do *subsampling* à primeira camada de convolução (28x28), apresentada na Figura 19, obtém-se uma camada de *Pooling* mais condensada (14x14). Na prática isto faz sentido, pois pode-se entender essa operação como o armazenamento da informação de presença de uma característica qualquer e descarte da informação de sua localização exata.

Em seguida, a saída de todos os planos que compõem a camada tridimensional de *Pooling* é conectada a uma camada convolucional, tal como foi feito para a imagem na camada de entrada. Aprende-se novos filtros de convolução (ou matrizes) e, da mesma forma, aplica-se o *subsampling*. Esse processo é feito repetidamente, obtendo-se matrizes de convolução independentes ao longo da rede. Após sucessivas aplicações, espera-se obter filtros que aprendem características de alto nível de uma imagem, diferentemente das características elementares, como: bordas, ruídos, cantos, arestas, que eram aprendidas nas primeiras camadas.

Essas características de mais alto nível são apresentadas para o treinamento de um classificador, através de uma Rede Neural totalmente conectada com uso de um *backpropagation* padrão.

Finalmente, pode-se aplicar uma função *Softmax* (Função Exponencial Normalizada) à saída da Rede *backpropagation* para transformá-la em um vetor cuja soma dos elementos é igual a 1. Assim, cada elemento desse vetor formado pela transformação *Softmax*, apresentada na Equação (18), pode ser interpretado como o grau de certeza

da classificação da imagem de entrada a cada uma das possíveis classes.

$$s(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \quad (18)$$

z representa o vetor composto pelas saídas da última camada, com a mesma dimensão do número de classes K . s é a saída da função *Softmax* e j refere-se a classe sobre a qual deseja-se calcular o grau de certeza.

2.4.2 ResNet

Os modelos de classificação utilizados neste trabalho foram construídos a partir da arquitetura ResNet (Rede Neural Residual), a qual teve sua implementação baseada nas Redes Neurais Convolucionais apresentadas anteriormente (HE *et al.*, 2015), visando a correção do problema de Gradientes Evanescentes.

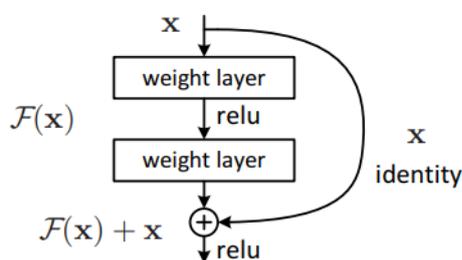
Esse problema se manifesta em redes profundas - com elevado número de camadas - em que o gradiente do erro calculado tende a zero, dificultando o processo de treinamento (WANG, 2019). Isso acontece porque, ao utilizar uma função de ativação como a tangente hiperbólica, as entradas são mapeadas em um intervalo reduzido de saída (-1 até +1). Assim, grandes alterações na entrada dessa função resultam em pequenas mudanças em sua saída e , portanto, a derivada calculada é próxima de zero. Durante a retropropagação dos erros, essas derivadas pequenas são multiplicadas das camadas finais até as iniciais para que se possa atualizar os pesos das primeiras conexões. Assim, a atualização dos pesos ocorre de forma lenta, pois o gradiente calculado decai conforme as sucessivas multiplicações desses valores próximos de zero são feitas. Portanto, o treinamento de redes profundas, com esse tipo de função de ativação, acaba sendo lento e pouco eficiente para pequenos valores de erro.

Com o intuito de corrigir esse problema foi proposta uma estratégia de “atalhos” que pulam camadas convolucionais, contornando as funções de ativação que comprimiam esses valores. Nota-se, a partir da Figura 20, que a saída passa a ser calculada pelo sinal resultante do processamento de duas camadas convolucionais somado ao sinal de entrada anterior ao processamento. Isso faz com que a perda de gradiente quando o erro é retropropagado seja muito menor.

Neste trabalho utilizou-se uma ResNet18 - uma arquitetura com dezoito camadas ocultas - para o treinamento de cada um dos seis modelos de classificação de imagens. Na Figura 21, pode-se perceber a utilização de camadas convolucionais, camadas de pooling, softmax, além dos “atalhos” característicos de uma Rede Residual.

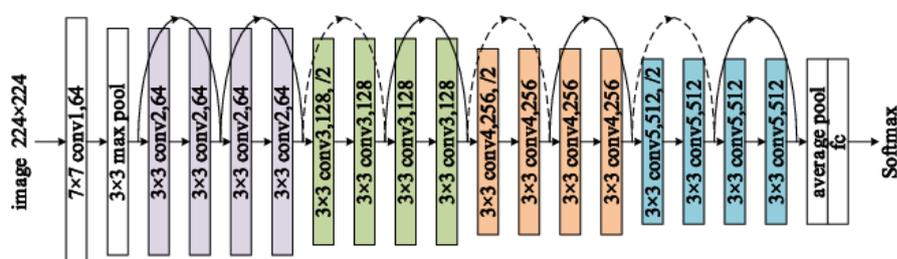
Uma vantagem das Redes Convolucionais é a otimização de treinamentos através da utilização de camadas pré-treinadas. Como as camadas iniciais dessa rede destinam-se à classificação de elementos básicos de uma imagem, pode-se reaprovei-

Figura 20 – Blocos Residuais



Fonte: (HE *et al.*, 2015))

Figura 21 – Arquitetura da ResNet18



Fonte: (OU *et al.*, 2019)

tar essas informações para obter um processo de treinamento mais assertivo. Divide-se, portanto, o treinamento em duas etapas: Aprendizado por Transferência e Ajuste Fino (YOSINSKI *et al.*, 2014).

No Aprendizado por Transferência, utiliza-se um modelo pré-treinado para outra tarefa similar como ponto de partida do treinamento voltado para o problema abordado. Desse modo, a inicialização dos pesos costuma ser mais certa, permitindo o treinamento mais eficiente da rede. Costuma-se partir de um modelo treinado para milhares de imagens, capaz de generalizar padrões do mundo visual. Isso faz sentido, pois as camadas iniciais, responsáveis por detectar recursos elementares, como bordas e formas, tendem a satisfazer às necessidades de diferentes problemas. Assim, treina-se apenas as camadas totalmente conectadas que se situam ao fim da rede, ou seja, não é feita a retropropagação nas camadas convolucionais. Deve-se, também, ajustar a dimensão da camada de entrada para a resolução das imagens referentes ao conjunto de dados desejado. Esse processo é vantajoso, pois reduz o tempo de treinamento e possibilita a utilização de conjuntos de dados menores, além de viabilizar o uso de hardware mais acessível.

Quando o treinamento atinge um resultado satisfatório, passa-se para a etapa de Ajuste Fino, que consiste em “descongelar” os pesos das camadas convolucionais,

propagar a ativação dos dados, calcular o erro e retreinar os pesos de toda a rede.

2.4.3 Conceitos utilizados no treinamento

O treinamento de uma Rede Neural tem o intuito de reduzir o erro de classificação que um modelo possui quando apresentado aos dados de entrada. Em geral, divide-se esses dados em conjunto de treino, conjunto de validação e conjunto de teste. O primeiro é utilizado para ajustar os pesos das conexões durante o processo de treinamento, o segundo permite uma avaliação imparcial de um modelo ajustado enquanto corrige-se os hiperparâmetros (parâmetros especificados manualmente, como a taxa de aprendizado), já o último também consiste em dados não usados no treinamento, mas é utilizado na verificação do resultado do modelo final. Encontra-se também, em algumas literaturas, a utilização de apenas dois conjuntos em que o conjunto de teste é denominado conjunto de validação (BROWNLEE, 2017).

No entanto, é possível que durante o processo de treinamento o modelo se ajuste perfeitamente ao conjunto de dados, perdendo sua capacidade de generalização. Esse fenômeno é chamado de *overfitting* (ou sobreajuste) e faz com que a performance do modelo seja inferior quando apresentado a dados nunca vistos. Por outro lado, o *underfitting* (ou subajuste) ocorre quando o modelo não consegue aprender satisfatoriamente, ou seja, tem-se um valor de erro elevado tanto no conjunto de treino quanto no conjunto de teste.

Uma época de treinamento consiste na apresentação de todo o conjunto de dados no processo de treinamento, incluindo propagação e retropropagação. O número de épocas é um hiperparâmetro que deve ser escolhido a fim de evitar subajuste e sobreajuste, já que com poucas épocas o modelo pode não aprender o suficiente, enquanto que o oposto pode acarretar na especialização do modelo sobre o conjunto de teste.

O *Batch Size* (tamanho do lote) é um hiperparâmetro que define o número de amostras que são apresentadas à rede por vez. Quando o tamanho do lote é igual ao tamanho do conjunto de dados garante-se a convergência para o ótimo global da função alvo em uma rede padrão, ao custo de um processo mais lento e a tendência de uma generalização pior quando comparado a valores menores de Batch Size.

Na prática, não existe uma regra geral definitiva para definição dos hiperparâmetros e a sintonização dos mesmos acaba ocorrendo de forma empírica, com base nos conceitos apresentados.

3 REQUISITOS GERAIS E PROJETO

O objetivo deste capítulo é apresentar as etapas de modelagem do projeto desenvolvido, incluindo tabelas de requisitos, diagramas e modelos.

O levantamento de requisitos gerais, o projeto de software por diagrama de classes, a especificação de componentes, a elaboração do projeto de circuito pneumático e eletrônico foram majoritariamente realizados pelo autor. Já as tarefas como: definição da estratégia de classificação de imagens, do diagrama estrutural e da máquina de estados, foram executadas em conjunto com um colega de equipe. A confecção do desenho 3D do protótipo foi realizada pelo time industrial, a partir das especificações fornecidas.

3.1 MODELAGEM

O problema tratado neste trabalho envolve a reformulação de um processo consolidado na fábrica, em Manaus. O intuito da alteração é garantir maior eficácia em filtrar câmeras defeituosas, padronizar os testes entre diferentes células de produção e operadores, além de reduzir o tempo total do procedimento. Entretanto, a distância física acaba sendo um empecilho na coleta de informações detalhadas acerca do funcionamento desse processo. Já que o processo envolve diferentes equipes e os dados da produção são de difícil acesso.

Nesse sentido, a definição dos requisitos se deu a partir do levantamento das cinco principais causas de lotes de produção bloqueados e do alinhamento com as equipes de Engenharia Industrial e Manutenção acerca das necessidades da fábrica, focando-se no ganho de qualidade e redução do tempo total de teste.

Durante essa etapa foi verificada a possibilidade de utilizar um sistema de Visão Computacional orientada a Redes Neurais Artificiais, haja vista os resultados positivos obtidos em (CANTO, 2022) e a natureza do problema abordado. Além disso, a robustez desse tipo de solução e a abrangência para diferentes modelos de câmeras, sem a necessidade de reparametrização, foi fator decisivo na escolha desta abordagem em detrimento de soluções de *thresholding* e Morfologia Matemática, por exemplo. Esse fator é importante pela necessidade de redução no número de ações por parte dos operadores. Pois, a dependência dos mesmos para realizar ações como: escolher entre diferentes executáveis ou selecionar parâmetros específicos de cada produto na interface, poderia gerar novos problemas.

Desse modo, foram levantados os requisitos funcionais e não funcionais, apresentados na Figura 22, com o intuito de estabelecer os principais pontos necessários ao desenvolvimento assertivo do projeto.

Espera-se, a partir do primeiro requisito, que a imagem gerada por uma câmera, após o término de sua montagem, seja aplicada em um modelo de classificação e

Figura 22 – Tabela de requisitos

Nome: F1 Detectar defeitos nas câmeras				
Descrição: O sistema deve analisar as imagens geradas pelas câmeras e identificar a presença de defeitos nas mesmas				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Tempo de teste	O tempo total de teste deve ser menor ou igual ao realizado atualmente pelo operador (Em torno de 27 segundos)	Temporal	()	(X)
NF1.2 Métrica do classificador	A métrica F1-score do sistema para identificação de defeitos nas imagens deve ser superior a 90%.	Performance	()	()
NF1.3 Setup	A quantidade de configurações necessárias por parte do usuário deve ser reduzida.	Interface	()	(X)
Nome: F2 Apresentar o resultado do teste ao operador				
Descrição: O resultado de cada teste deve ser apresentado e mantido na tela até o início da próxima sequência de testes				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Armazenamento	A imagem do defeito, o operador responsável e o código da câmera devem ser armazenadas permanentemente.	Armazenamento	()	(X)
NF2.2 Sobreposição de falhas	Se uma câmera for reprovada, o sistema deve exigir a identificação do operador antes de permitir a realização de novos testes	Interface	()	(X)
NF2.3 Notificação	A identificação de um defeito deve ser apresentada de forma clara, destacada em vermelho na interface	Interface	()	(X)
Nome: F3 Armazenar os dados da produção				
Descrição: Armazenar dados como: número de câmeras produzidas por hora e taxa de defeitos				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Integração dos dados	Armazenar os dados em formato padronizado viabilizando a integração com as demais células	Armazenamento	(X)	()
Nome: F4 Iniciar o processo de identificação				
Descrição: O sistema deve iniciar a identificação a partir do comando bimanual do usuário.				
Requisitos Não Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Identificação	O início do processo requer a configuração do código da câmera por parte do operador.	Especificação	()	(X)

Autoria própria

tenha-se um resultado objetivo quanto à sua integridade. Nesse sentido, o desempenho do teste realizado pelo sistema inteligente deve ser superior ao obtido com a inspeção visual do operador, evitando a aprovação de câmeras defeituosas. Ainda, o tempo necessário para executar esse procedimento, incluindo a manipulação das câmeras pelos operadores, deve ser inferior - ou igual - ao realizado atualmente. Além disso, deseja-se o mínimo de interação possível do usuário com a interface, focando-se nas demandas da produção.

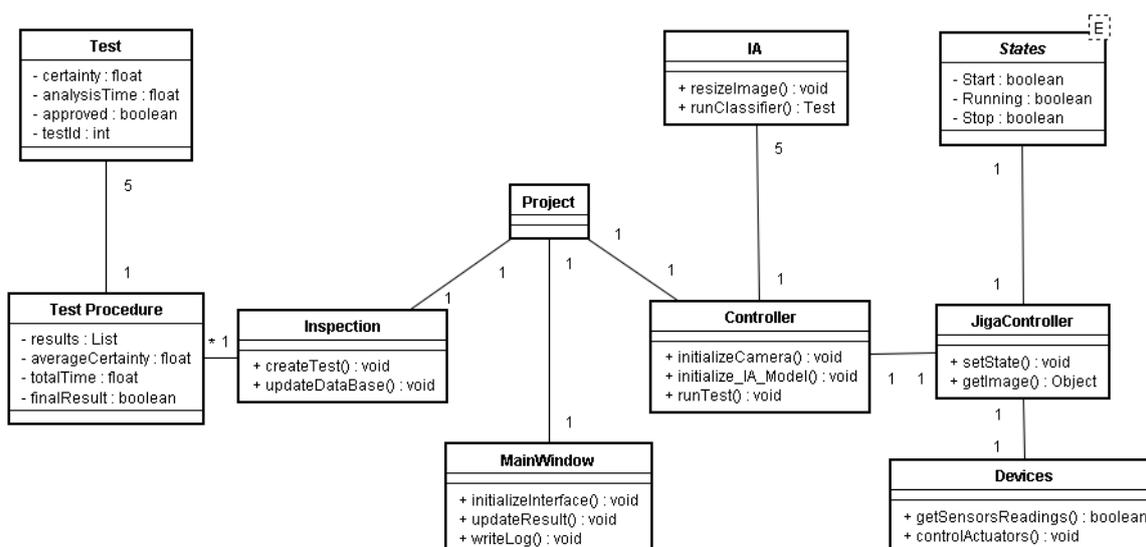
O segundo requisito é importante para mitigar os problemas de desatenção que ocorrem na fábrica, propondo-se pontos de atenção para destacar a ocorrência de um evento adverso. Desse modo, na presença de um defeito, o crachá do operador deve ser apresentado a um leitor, por exemplo, antes que ele possa destinar essa câmera à

próxima etapa. Além disso, alterações de cor devem ocorrer na interface para destacar o problema.

No terceiro requisito prepara-se o sistema para a integração futura com outras interfaces de análise de dados, permitindo-se otimizações no processo produtivo.

O sistema deve permitir a alternância de cenários percebidos pela câmera, já que a execução de alguns testes exige a captação de padrões diferentes. Isto é, ora aponta-se a câmera para regiões com padrões de cor, ora aponta-se para regiões totalmente brancas. Por conta disso, devem ser utilizadas estratégias de acionamentos que atendam a essa necessidade, o que acaba demandando adequações de segurança. Com o intuito de preservar a integridade dos operadores, para o quarto requisito deve-se empregar botões com acionamento bimanual, garantindo a ocupação simultânea de ambas as mãos.

Figura 23 – Diagrama de classes



Autoria própria

Com base nos requisitos levantados e nas soluções estudadas durante a graduação, mais especificamente na disciplina de Visão Computacional (WANGENHEIM, 2019), optou-se pela utilização de Redes Neurais Convolucionais na construção de um modelo para classificação de câmeras quanto à integridade da imagem gerada. Nesse sentido, a familiaridade do autor com a biblioteca *fastai* (HOWARD; THOMAS, 2016) e a linguagem de programação *Python* também orientaram a abordagem empregada para o desenvolvimento do software. Vale ressaltar que a programação efetiva das funções em Python, no que diz respeito à interface e armazenamento de dados, foram realizadas por um colega de equipe.

Assim, com o intuito de mapear a estrutura do projeto e entender as associações necessárias entre diferentes módulos, elaborou-se o modelo conceitual e, posteriormente, o diagrama de classes, apresentado na Figura 23.

A estrutura de software desenvolvida é baseada no padrão MVC (Model View Controller (MEDEIROS, 2013)). Na classe MainWindow (View) tem-se a interface com o usuário, onde apresenta-se os botões, os campos para preenchimento e os resultados provenientes da controladora. A classe Controller é responsável por interpretar as entradas do usuário, transformando-as em ações para as demais classes, como, por exemplo, comandar o início da execução do algoritmo de predição. Já a classe Inspection (Model) refere-se a parte de dados do sistema, instanciando os testes e organizando o seu armazenamento.

Na classe IA estão implementadas as funções de tratamento da imagem coletada, como: redimensionamento e conversão para um formato específico. Encontram-se também as funções referentes ao carregamento do modelo de Redes Neurais e sua execução.

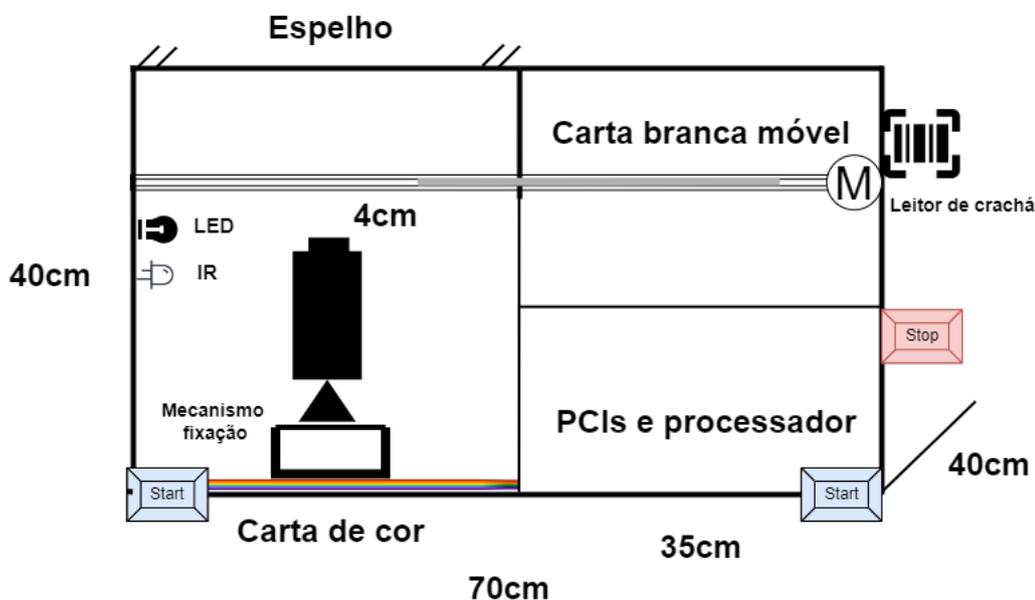
Funções como: acionamento de atuadores, leitura de sensores, botões, e identificação do crachá, são implementadas na classe Devices. A coordenação desses comandos se dá através de uma máquina de estados, apresentada posteriormente nesta seção, implementada na classe JigaController.

A estrutura da Jiga de Testes foi projetada com o objetivo de minimizar o número de graus de liberdade necessários para alternância da câmera entre os cenários de testes. Os primeiros modelos propostos eram baseados no deslocamento da câmera, porém percebeu-se que o deslocamento do cenário seria mais eficiente devido à agilidade na fixação de diferentes modelos de câmeras e o tempo necessário para a estabilização da imagem após a movimentação da câmera. A Figura 24 ilustra o esboço da estrutura prevista para o sistema.

O espelho fixado ao teto permite que a imagem captada pela câmera seja utilizada tanto para analisar a presença da tampa de acrílico no próprio produto quanto para identificar anomalias de cor, já que dispõe-se de uma carta com diferentes padrões de cor embaixo do mecanismo de fixação. Isso reduz a necessidade de diferentes cenários para a execução dos testes de tampa e de cor, otimizando-se o tempo de teste.

As câmeras de *case Bullet* captam um campo de visão mais fechado e possuem altura maior, quando comparadas com as câmeras de *case Dome*. Em um mesmo nível de referência, isso impacta a cobertura completa das imagens pela carta branca. Assim, definiu-se a compensação de altura necessária para os dois mecanismos de fixação, incluindo todos os modelos de câmeras produzidos e as variações existentes. O requisito de redução do setup da linha de produção foi fator determinante para a aplicação dessas compensações, influenciando no tamanho necessário para a carta

Figura 24 – Diagrama da estrutura do sistema de testes



Fonte: Repositório do projeto

de cor e, conseqüentemente, no tamanho total da Jiga.

Como forma de reduzir os riscos à segurança na operação do sistema de testes, optou-se por construir um equipamento totalmente fechado que impeça o contato com o atuador pneumático. Além disso, o início dos testes - seguido do travamento da porta de acesso - é condicionado ao pressionamento simultâneo dos dois botões Start.

A partir dessas definições, solicitou-se ao time industrial a confecção de um projeto de desenho 3D, apresentado na Figura 25, visando a construção do primeiro protótipo.

3.2 FUNCIONAMENTO

O diagrama de máquina de estados, apresentado na Figura 26, foi utilizado para descrever as saídas desejadas do sistema a partir da sequência de eventos correspondente (IBM, 2021).

Nota-se que os estados estão relacionados com o sequenciamento de testes previsto, aplicando-se as saídas correspondentes aos seus requisitos. Tal que “1” e “0” referem-se ao acionamento e desligamento de cada atuador, respectivamente.

- LED: Variável correspondente ao acionamento da fita LED, responsável por alterar o cenário dentro da Jiga de Testes e permitir que a câmera entre em modo noturno;

- Ar: Variável associada à válvula pneumática que comanda o deslocamento da alimentação de ar da Válvula do Cilindro. Emprega-se esse dispositivo para garantir que o cilindro não possa se deslocar na presença de um operador;
- Trava: Variável relativa ao sistema de retenção da porta de acesso à Jiga. Deve coibir o acesso a área de testes durante a execução dos mesmos;
- Laser: Variável correspondente ao sistema de marcação das câmeras;
- sensorInicio: Evento correspondente à chegada do cilindro pneumático à posição de avanço, ou seja, cobrindo a lente da câmera com a carta branca;
- sensorFim: Evento referente à chegada do cilindro pneumático à posição de recuo, ou seja, permitindo que a lente da câmera visualize completamente o espelho fixado no teto da Jiga de Testes.

O disparo do evento “botãoStop” leva para o estado “Parada emergência” independente do estado em que o sistema se encontra. Na prática, nesse estado mantém-se a última posição do cilindro e corta-se a alimentação de ar comprimido e do laser.

O sistema inicia seu funcionamento no estado início, e executa o Teste de Conexão quando o operador pressiona os botões simultaneamente com a porta da Jiga fechada. Esse teste foi acrescentado para verificar a existência de sinal de imagem na câmera e garantir o aguardo do tempo de inicialização da mesma. Caso a câmera ainda não tenha inicializado ou apresente alguma falha, retorna-se ao estado início. Caso contrário, disparam-se os testes de tampa e de cor. No primeiro é verificada a presença da tampa de acrílico no produto, enquanto que no segundo detectam-se anomalias na disposição de cores na imagem gerada, como, por exemplo, a apresentação de faixas laranjas em vez de faixas amarelas.

Seguido do disparo desses testes, comanda-se o avanço da carta branca para a cobertura completa da lente das câmeras, através de um atuador pneumático linear. Aguarda-se o atingimento da posição desejada, indicado pelo evento “sensorInicio” e, então, inicia-se o Teste de Mancha. Na ocorrência desse defeito interrompe-se o teste precocemente, pois as próximas ações envolvem a transição relativamente demorada da câmera para o modo noturno.

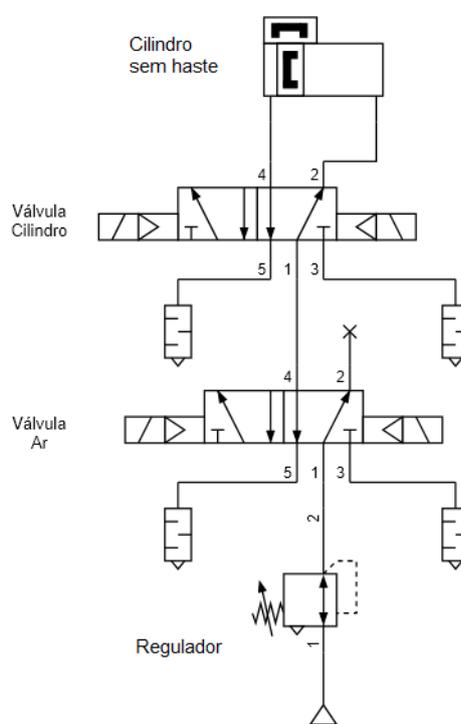
Do contrário, aciona-se o recuo do cilindro pneumático e desliga-se a fita LED. Nessa condição, o aparecimento de defeitos no filtro ICR torna-se evidente, já que a imagem adquire aspecto rosado com o seu mau funcionamento. Após a execução desse teste, ainda deve-se aguardar alguns segundos para que a câmera opere em modo noturno. Espera-se que nesse modo de operação, com exceção das câmeras com tecnologia Full Color, o produto acione o LED Infravermelho e passe a gerar imagens em preto e branco.

Com a finalização de todas as classificações, o produto deve ser retirado da Jiga e destinado adequadamente. No caso de aprovação, dispara-se um feixe de laser ultravioleta para marcação do conector e libera-se a trava da porta, cabendo ao operador a inserção da câmera na caixa do produto. Por outro lado, a reprovação exige a identificação do operador através de um leitor de crachá RFID para que o produto seja liberado e destinado ao retrabalho. Esse procedimento foi adicionado para despertar a atenção do operador quanto a ocorrência de um defeito, além de registrar os responsáveis pela operação em cada turno. Com isso, o processo pode ser reiniciado após o término da montagem de outra câmera.

3.3 PROJETO DE CIRCUITOS

Com os objetivos de: especificar o comportamento desejado para o sistema, definir os componentes necessários e permitir a comunicação padronizada com as demais áreas, desenvolveu-se o projeto de circuito pneumático, apresentado na Figura 27.

Figura 27 – Diagrama de Circuito Pneumático



Autoria própria

Nesta solução foram utilizados os seguintes componentes:

- Válvula Reguladora de Pressão;

- Válvula direcional de 5 portas e 2 posições com controle solenoide de piloto interno;
- Cilindro sem haste de dupla ação;
- Silenciadores pneumáticos.

A válvula reguladora conectada ao suprimento de ar comprimido é utilizada para manter a pressão constante independente do consumo do atuador e das variações na linha, além de reduzir pulsações provenientes do compressor. Na sequência, a conexão segue para a primeira válvula solenoide de duas posições que, ao ser acionada, desloca o fluxo de ar para a alimentação da válvula responsável por comandar a movimentação do cilindro. Essa abordagem foi adotada como forma de garantir que o cilindro esteja despressurizado sempre que a porta da Jiga de Teste estiver aberta, minimizando os riscos ao operador. O comando da bobina de ambas as válvulas se dá através de relés conectados às saídas digitais do computador responsável pelo controle do sistema.

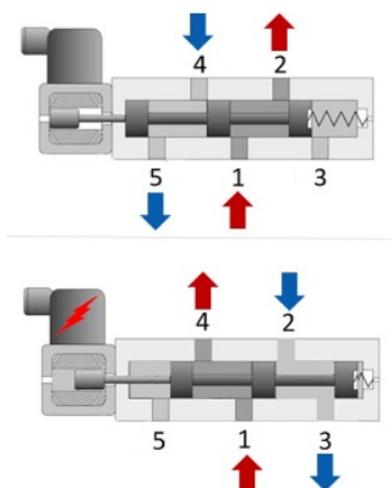
Nas válvulas solenoides de ação direta, o campo magnético da bobina energizada atua diretamente sobre o elemento responsável pela liberação da passagem do fluido (BELTON, 2020). Já nas válvulas servo-operadas que foram utilizadas neste projeto, o campo magnético libera uma pequena parte do fluxo de ar para um canal que atua sobre um êmbolo. Por sua vez, este atua sobre o elemento encarregado de obstruir ou liberar a passagem do fluido principal para o restante do circuito. A vantagem da topologia empregada é a possibilidade de utilização de bobinas menores, já que o trabalho de deslocamento dos elementos principais passa a ser realizado pelo fluido intermediário.

Na Figura 28 ilustra-se o funcionamento de uma válvula de cinco portas e duas direções. Na imagem superior, a bobina não está energizada e a posição do elemento de bloqueio é dada pela ação de uma mola. Já na imagem inferior, o campo magnético da bobina faz com que o elemento se desloque e as portas pelas quais o fluido passa sejam alteradas.

Além disso, projetou-se um diagrama de circuito eletrônico para a interação do computador - que executa o software em Python - com os atuadores, sensores e botões presentes na Jiga de Testes, através de entradas e saídas digitais.

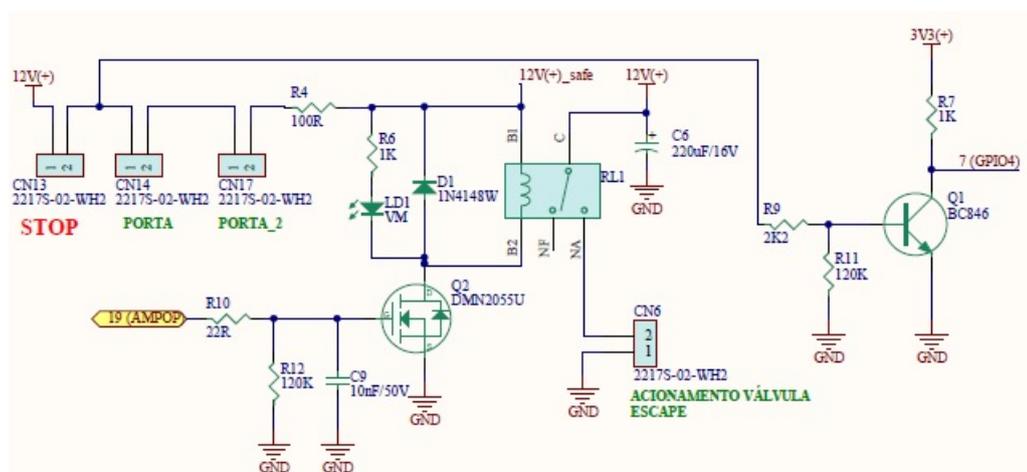
A título de curiosidade, na Figura 29, é apresentada uma pequena parte do projeto elaborado no software Altium Designer. Nota-se a utilização de diferentes componentes de chaveamento, como: MOSFET (Metal Oxide Semiconductor Field Effect Transistor), relé e transistor. Recorreu-se a essa topologia para acionamento da válvula pneumática devido às limitações de corrente do controlador. Na prática, esse circuito aciona a válvula de liberação de ar comprimido quando a saída digital “19” encontra-se em nível lógico alto e, também, lê o estado do botão *Stop* na entrada digital “7”.

Figura 28 – Funcionamento de uma válvula solenoide 5/2



Fonte: www.mtibrasil.com.br

Figura 29 – Diagrama de circuito eletrônico



Autoria própria

4 DESENVOLVIMENTO

Neste capítulo são descritos os detalhes de implementação do projeto. Aborda-se a explicação sobre os testes realizados, o processo de treinamento da *ResNet*, a construção do protótipo e a validação do mesmo na linha de produção.

A usinagem das partes estruturais da Jiga de Testes foi executada pelo time industrial, na filial São José. Já as atividades de instrumentação do sistema, montagem de placa de circuito e programação de software foram realizadas em conjunto com um membro da equipe.

4.1 TESTES

Nesta seção explica-se cada um dos procedimentos de teste que são aplicados às câmeras, abordando-se os fatores que caracterizam um produto como defeituoso ou íntegro. As imagens situadas na parte superior e com bordas verdes são de produtos íntegros e pertencem ao conjunto “OK”, enquanto que as imagens localizadas na parte inferior e com bordas vermelhas foram captadas por câmeras defeituosas e pertencem ao conjunto “NOK”.

4.1.1 Teste de conexão

O teste tem o objetivo de garantir que a câmera esteja devidamente conectada antes que os demais testes possam ser disparados. Verificou-se a necessidade de sua inclusão, após os primeiros experimentos, em virtude da tendência do usuário em apertar os botões de inicialização antes do carregamento da imagem no monitor. Além disso, nas câmeras Dome pode-se alterar manualmente a direção de apontamento da lente, ou seja, ao fixá-la no suporte da Jiga não necessariamente tem-se a visão centralizada do espelho, posicionado no teto. Assim, o teste de conexão verifica a presença de imagem e o alinhamento correto da câmera para que possam ser iniciados os demais testes.

Na Figura 30, pode-se observar as diferenças entre os dois conjuntos de dados (“OK” na parte superior e “NOK” na parte inferior).

Figura 30 – Amostras do conjunto de dados para o teste de conexão

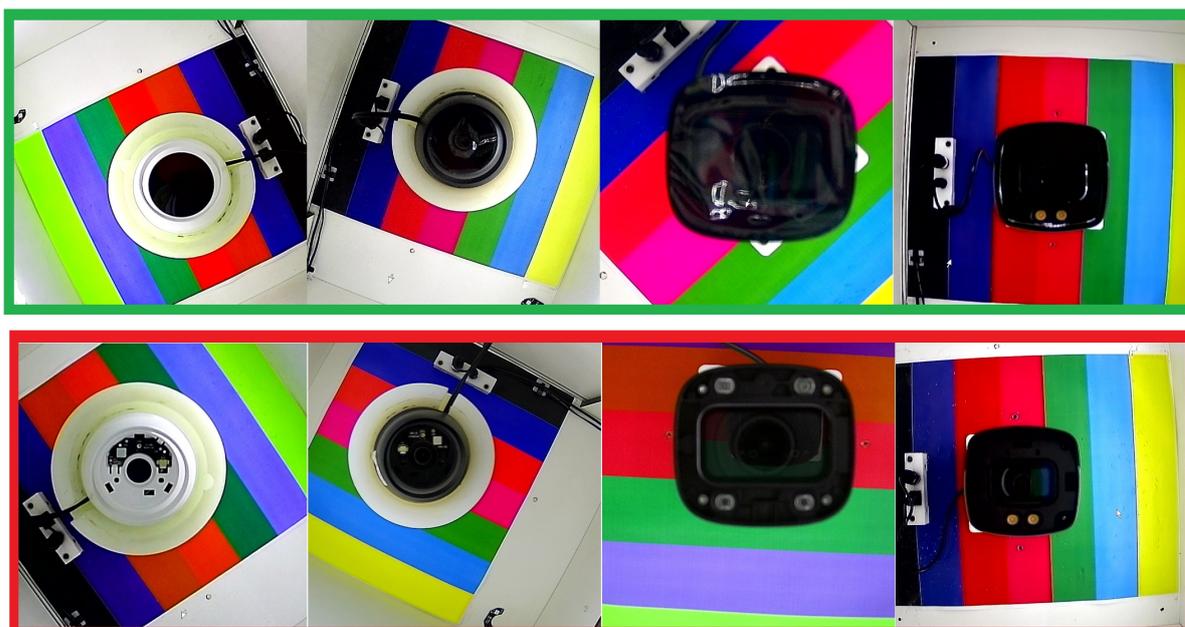


Autoria própria

4.1.2 Teste de tampa

O teste de tampa visa filtrar os casos em que o produto é montado sem a tampa de acrílico. A ausência desse componente é de fácil percepção para alguns produtos, porém em outros necessita-se de maior atenção para identificar a falha. Essas diferenças podem ser observadas na Figura 31.

Figura 31 – Amostras do conjunto de dados para o teste de tampa



Autoria própria

4.1.3 Teste de cor

Com o intuito de identificar anomalias na cor apresentada na imagem de uma câmera de segurança, aplica-se o teste de cor. A ideia é aceitar pequenas distorções e reprovar as câmeras com alterações mais evidentes. Em geral, este defeito é causado por problemas em um capacitor associado ao circuito oscilador ou por um sensor de imagem danificado. Alguns exemplos desse problema podem ser observados na Figura 32.

Figura 32 – Amostras do conjunto de dados para o teste de cor



Autoria própria

4.1.4 Teste de mancha

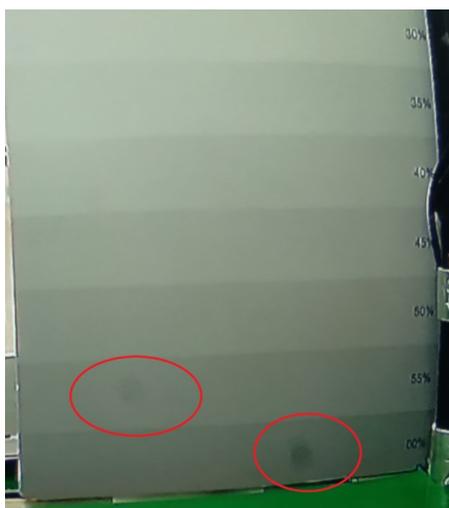
O procedimento de verificação da existência de manchas é feito em duas etapas diferentes da produção de câmeras. Pois, no deslocamento entre os setores da fábrica pode ocorrer de pequenas partículas se desprenderem e serem depositadas sobre o sensor de imagem. Esse problema é apontado como a principal preocupação na fábrica devido à sua alta incidência. Além disso, existe um grau de tolerância para as manchas identificadas após a Montagem Final. Manchas escuras com até 30% na escala de cinza (Figura 34), padronizada pelo Controle de Qualidade, devem ser consideradas como aprovadas pelo teste. Na Figura 33 podem ser observadas as imagens que caracterizam esse defeito, com as câmeras apontadas para o cenário referente à carta branca.

Figura 33 – Amostras do conjunto de dados para o teste de mancha



Autoria própria

Figura 34 – Câmera com o defeito de mancha escura



Repositório do projeto

4.1.5 Teste de ICR

O ICR atua na filtragem de ondas na faixa do infravermelho para evitar a saturação do sensor de imagem quando exposto à luz do dia. A simulação desse defeito se dá através do acionamento de um LED infravermelho dentro da Jiga, com o intuito de induzir a formação de uma imagem distorcida (rosada) nos casos de mau funcionamento do componente. Esse fenômeno pode ser observado na Figura 35.

Figura 35 – Amostras do conjunto de dados para o teste de ICR



Autoria própria

4.1.6 Teste de LED

Quando a luminosidade percebida pela câmera atinge um patamar mínimo, espera-se a transição para o modo noturno. Isto é, os LEDs devem ser acionados para que se tenha uma imagem visível em preto e branco (ou colorida para os modelos Full Color). A Figura 36 ilustra as divergências nesse comportamento para modelos funcionais e não funcionais.

Figura 36 – Amostras do conjunto de dados para o teste de LED



Fonte: Repositório do projeto

4.2 TREINAMENTO DO MODELO

4.2.1 Considerações sobre a função custo

A equação da função custo, ou função de perda, baseada na soma dos quadrados dos erros foi apresentada na seção 2.4 deste trabalho com o intuito de simplificar a explicação dos conceitos. Porém, no processo de treinamento efetivamente realizado, optou-se pela escolha da entropia cruzada do erro como função custo.

Essa função é amplamente utilizada para corrigir o problema de desaceleração do aprendizado (ACADEMY, 2022). Isso ocorre quando a saída obtida está distante da saída esperada e, mesmo assim, os pesos continuam a ser atualizados lentamente, por conta das características da função de ativação. Por exemplo, utilizando-se a função sigmoide, em azul na Figura 14, como função ativação, em um neurônio com saída próxima de “1” tem-se uma derivada muito pequena para essa função. Consequentemente, a derivada da função custo em relação aos pesos também será pequena e, portanto, o aprendizado será lento.

Assim, define-se a função da entropia cruzada para um neurônio por:

$$E(w) = \frac{-1}{n} \sum_x [y_r \cdot \ln(y) + (1 - y_r) \cdot \ln(1 - y)], \quad (19)$$

onde n refere-se ao número de amostras do conjunto de treinamento, o somatório é aplicado sobre os dados de entrada, y_r e y representam a saída esperada e calculada, respectivamente. Nota-se que os logaritmos são negativos, pois são aplicados em valores limitados entre “0” e “1”. Portanto, a multiplicação por “ $\frac{-1}{n}$ ” resulta em uma função custo positiva.

Calcula-se a derivada parcial de $E(w)$ em relação aos pesos, aplicada à função sigmoide, dada por $y = \sigma(z) = \frac{1}{1+e^z}$, tal que z refere-se ao somatório das entradas multiplicadas pelos pesos correspondentes. Simplificando a expressão, tem-se:

$$\frac{dE(w)}{dw_j} = \frac{1}{n} \sum_x (\sigma(z) - y), \quad (20)$$

As etapas de desenvolvimento da Equação (20) foram suprimidas neste trabalho, mas podem ser encontradas no capítulo 17 de (ACADEMY, 2022). Na prática, a equação mostra que a derivada da função custo, ou seja, a taxa de atualização dos pesos é dada pelo erro entre a saída do neurônio e a saída esperada. Portanto, com a utilização dessa função, corrige-se o problema de desaceleração do aprendizado.

4.2.2 Coleta de dados e treinamento

Para que o processo de treinamento de uma Rede Neural Convolutiva seja efetivo, faz-se necessária a coleta de um banco de dados representativo e com as

devidas rotulações. Isto é, separa-se as imagens em pastas que correspondem às classes sob as quais o modelo final deve informar o grau de pertencimento. Neste caso, a imagem de uma câmera para um dado teste pertence à classe “OK” para câmeras íntegras ou “NOK” em caso de câmeras defeituosas. A obtenção das imagens se dá através de um dispositivo - denominado placa de captura - que converte o sinal de saída HDMI do DVR, conectado à câmera, para uma entrada USB do computador.

Como forma de expandir o número de imagens coletadas, espelhou-se verticalmente, horizontalmente e de ambas as formas cada imagem. Essa estratégia, além de ser utilizada para aumento do banco de dados, de certa maneira também representa os dados reais, pois o operador pode encaixar a câmera no suporte em diferentes orientações. Com o objetivo de não inserir vieses nos dados coletados, atentou-se para a variabilidade de modelos para cada defeito e aleatoriedade na separação dos dados entre os conjuntos de treino e validação. Na prática, deseja-se que a Rede Neural aprenda somente o padrão de interesse, sem levar em consideração informações como: data e hora do DVR, posição do mouse, luminosidade, ruídos de um exemplar de câmera, sombras, entre outros.

Foram coletadas: 200 imagens para o teste de conexão, 887 para o teste de tampa, 1518 para o teste de mancha, 780 para o teste de cor, 310 para o teste de ICR e 460 para o teste de LED. Esses valores foram obtidos conforme a complexidade estimada do teste, a disponibilidade de amostras e a necessidade percebida durante o treinamento. No teste de mancha reparte-se a imagem em quatro regiões e dispara-se os testes em threads. Isso faz sentido, pois a imagem apontada para uma carta branca possui pouca dependência de contexto entre pixels vizinhos quando comparada com a imagem do teste de tampa, por exemplo. Desse modo, a dimensão das imagens é reduzida, mas dispõe-se de mais amostras no banco de dados.

O processo de treinamento foi executado no Google Colab, através da biblioteca *fastai* (HOWARD; THOMAS, 2016). Assim, as etapas para a obtenção do modelo consistem em:

- Separar o banco de dados em 80% para o conjunto de treino e 20% para o conjunto de validação (escolhendo-se o batch size);
- Carregar a arquitetura da Rede desejada, no caso: ResNet18 (escolhendo-se a métrica);
- Disparar o processo de treinamento por transfer learning, importando-se a inicialização de pesos conveniente e escolhendo-se o número de épocas;
- Avaliar os resultados em termos das funções de perda;
- Salvar o treinamento;

- Aplicar o Fine Tuning (escolhendo-se o número de épocas);
- Avaliar o desempenho em termos das funções de perda e da matriz de confusão.

A escolha de hiperparâmetros se deu de forma empírica, verificando-se o comportamento da rede a partir das métricas adotadas. Em geral, nos seis modelos construídos optou-se pela utilização das ResNet com 18 camadas, o tamanho do lote igual a 12 e a taxa de aprendizado variável entre 0,0005 e 0,003. Vale ressaltar que as análises deste trabalho estão centralizadas no problema de identificação de mancha, pois a taxa de incidência deste defeito é muito maior que a dos demais (cerca de 4,1% contra menos de 0,4%), além da maior dificuldade em sua identificação por parte dos operadores.

Na Figura 37 são apresentados resultados obtidos em 8 épocas de treinamento das últimas camadas totalmente conectadas (Transfer Learning) para o problema de mancha escura. O desempenho inicial satisfatório se deve ao pré-treinamento da rede facilitar a identificação das *features* de interesse já nas primeiras iterações do algoritmo.

Figura 37 – Dados de treinamento para o modelo da mancha

Epoch	Train_loss	Valid_loss	Accuracy	Error_rate	Precision	Recall	FtScore	Time
0	1.1956	0.7003	0.6991	0.3009	0.6115	0.6141	0.6013	0.01
1	0.8814	0.9050	0.6991	0.3009	0.7731	0.7241	0.7478	0.01
2	0.6606	0.4328	0.8080	0.1920	0.8269	0.8153	0.8210	0.01
3	0.5183	0.3919	0.8309	0.1691	0.8412	0.8631	0.8520	0.01
4	0.3999	0.3674	0.8510	0.1490	0.8407	0.8756	0.8578	0.01
5	0.3636	0.3141	0.8625	0.1375	0.8552	0.8788	0.8668	0.01
6	0.3210	0.2904	0.8739	0.1261	0.9006	0.8805	0.8904	0.01
7	0.2821	0.2549	0.8883	0.1117	0.9037	0.9038	0.9038	0.01

Autoria própria

Descongelando-se as camadas convolucionais e aplicando-se o processo de *Fine Tuning* - para 6 épocas - foram obtidos os resultados apresentados na Figura 38.

Figura 38 – Dados de refinamento para o modelo da mancha

Epoch	Train_loss	Valid_loss	Accuracy	Error_rate	Precision	Recall	FtScore	Time
0	0.3132	0.2449	0.9026	0.0974	0.8443	0.8774	0.8568	0.01
1	0.4055	0.2124	0.9398	0.0602	0.8937	0.9443	0.9183	0.01
2	0.2652	0.1354	0.9542	0.0458	0.9278	0.8803	0.9034	0.01
3	0.1848	0.0523	0.9885	0.0115	0.9784	0.9889	0.9836	0.01
4	0.0758	0.0394	0.9771	0.0229	0.9518	0.8724	0.9104	0.01
5	0.0639	0.0203	0.9943	0.0057	0.9817	0.9946	0.9881	0.01

Autoria própria

A função de perda para o conjunto de validação (`valid_loss`) apresentou resultados menores que os obtidos para o conjunto de treinamento (`train_loss`). A partir disso, pode-se esperar que não tenha ocorrido sobreajuste dos dados de treino durante o processo de treinamento. Para os demais modelos foram obtidos resultados similares em termos das métricas utilizadas.

Pode-se observar o progresso do treinamento da Rede graficamente a partir da Figura 39. A linha em verde divide as etapas de Transfer Learning e Fine Tuning.

Figura 39 – Gráfico dos dados de treinamento em função das épocas



Autoria própria

Nesse sentido, a escolha do número de épocas adequado se mostra fundamental para que seja possível extrair as características dos dados de treino e ao mesmo tempo evitar o ajuste específico para este conjunto, que implicaria na perda da capacidade de generalização.

Expressando os dados em termos de uma matriz de confusão pode-se entender de forma mais direta o desempenho do modelo. Através da Figura 40 pode-se perceber que não foram feitas previsões incorretas para o conjunto de validação. Entretanto, o acesso a amostras de câmeras defeituosas no ambiente de trabalho, na Unidade Matriz, é limitado.

Assim, com o intuito de expandir o banco de dados para além das câmeras disponíveis, foram simulados defeitos através da deposição proposital de partículas sólidas sobre o sensor de imagem de outras câmeras. Ainda que essa abordagem contribua para a coleta de mais dados, sabe-se que não representa de forma totalmente fiel os problemas que podem ocorrer na linha de produção. Por isso, é esperado que

o modelo não apresente um desempenho tão bom quando aplicado na fábrica. Pensando nisso, planejou-se a coleta de mais dados e retreinamento do modelo durante a implementação do protótipo, em Manaus.

Figura 40 – Matriz de confusão para o conjunto de validação

	Real: Mancha	Real: OK
Predito: Mancha	103	0
Predito: OK	0	143

Autoria própria

4.3 PROTÓTIPO

No primeiro protótipo do sistema utilizou-se um motor de passo acoplado ao suporte das câmeras que as movimentava entre duas posições de teste. Essa abordagem foi adotada por conta dos primeiros requisitos informados pela fábrica que consistiam na necessidade de movimentar o produto da posição inicial (à esquerda) para uma posição final (à direita). Entretanto, a partir da melhor compreensão do processo, optou-se por utilizar um sistema pneumático responsável por movimentar o cenário, em vez da câmera. A existência de outros dispositivos pneumáticos na fábrica e a velocidade de deslocamento, em relação à solução adotada inicialmente, corroboraram para essa mudança. No entanto, essa etapa inicial foi importante para se ter um sistema de deslocamento funcional enquanto os esforços se concentravam no estudo da estratégia de classificação de imagens.

A partir do refinamento dos requisitos e definição quanto às necessidades da estrutura em termos de ângulo de visão, luminosidade interna e dimensionamento, solicitou-se o auxílio do time industrial da filial São José para a construção da mesma.

Com a entrega da estrutura da Jiga, pôde-se iniciar o processo de instalação de componentes e montagem dos cenários. Essa etapa é importante para permitir a coleta de um banco de imagens fiel, ou muito próximo, do cenário real de aplicação, visando o treinamento do modelo por Redes Neurais.

Optou-se pela utilização do Jetson Nano, apresentado na Figura 41, como computador responsável por executar o software do sistema, processando o modelo, lendo botões e sensores, e comandando o acionamento de atuadores. A utilização do Raspberry PI 4 Model B - disponível no local de trabalho - foi descartada pela necessidade de redução do tempo do sistema e a unidade de processamento gráfica do Jetson permitir a execução de testes cerca de quatro vezes mais rápida, nos testes realizados. A

capacidade de processamento e a disponibilidade de entradas e saídas digitais foram os primeiros pontos a serem considerados na escolha da solução, porém inicialmente foram abordadas outras estratégias como, por exemplo, o uso de um computador convencional conectado a um Arduino. Entretanto, o atendimento dos requisitos por parte do Jetson Nano e o desempenho satisfatório para aplicações de Inteligência Artificial tornaram conveniente a sua escolha.

Figura 41 – Jetson Nano Developer Kit



Fonte: www.devobox.com

Com o intuito de reduzir as dimensões da estrutura, comportando todo o conjunto pneumático na parte interna, optou-se pela utilização de um cilindro sem haste. Esse dispositivo é aplicado na movimentação da carta branca sobre a lente conforme a Figura 42.

Na figura apresentada também pode-se observar o encaixe e conexão de uma câmera Bullet. Os suportes, construídos pelo time industrial, possuem diferentes níveis para contemplar o encaixe de todos os cases de câmeras, além de compensar a altura para atender os requisitos de campo de visão. Na Figura 43 pode-se observar os detalhes construtivos do suporte para câmeras Dome.

Em paralelo à construção e instalação da estrutura desenvolveu-se o software da aplicação em Python. O autor contribuiu na programação de funções como: leitura de sensores e botões bimanuais, tratamento do leitor de crachá, acionamento de atuadores e implementação da máquina de estados. Porém, funções relacionadas à interface, criação de logs, tratamento da imagem, entre outras, foram elaboradas por um colega de equipe.

Na interface do sistema são apresentados os resultados dos testes realizados, além da imagem utilizada pelo modelo para a classificação. Desenvolveu-se essa estrutura com o objetivo de apresentar um resultado de forma clara e minimizar o número de ações do operador. Na Figura 44 é apresentada a visualização disponível para o operador, à esquerda tem-se a interface do sistema com o resultado dos testes. Ainda, pode-se visualizar a imagem fornecida pela câmera no monitor à direita. Essa

Figura 42 – Carta branca em posição avançada



Fonte: Repositório do projeto

Figura 43 – Suporte das câmeras Dome

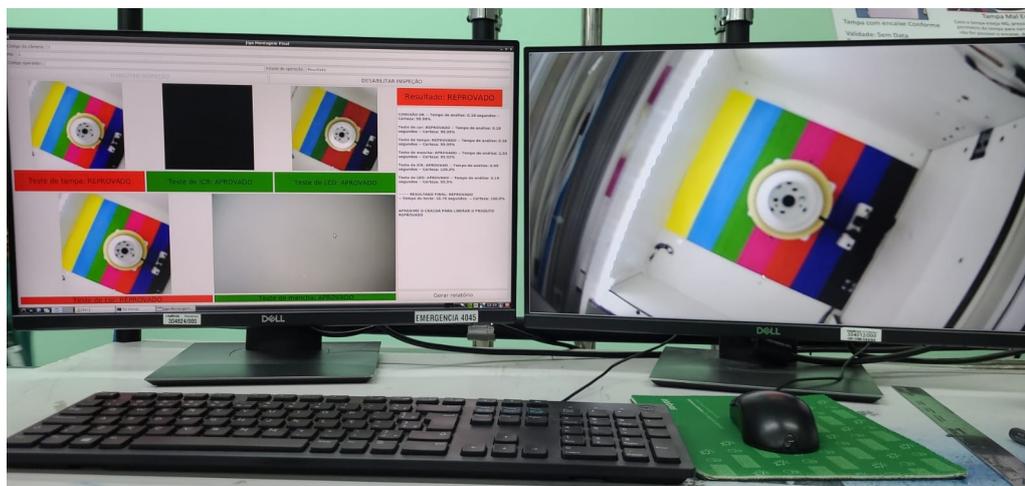


Fonte: Repositório do projeto

configuração permite que o operador identifique eventuais problemas não previstos e contribua para o melhor funcionamento do sistema.

Com a construção do modelo de classificação para cada um dos testes pôde-se validar o funcionamento do sistema como um todo. A coordenação de tarefas como: acionamento da fita LED e LED infravermelho, avanço de atuador pneumático e fechamento da trava, permitiu a realização de todos os testes em aproximadamente onze segundos. Somado a isso, tem-se dez segundos de manipulação da câmera por parte

Figura 44 – Interface da Jiga de Testes



Fonte: Repositório do projeto

do operador, ou seja, conectar os cabos de alimentação e vídeo, além de posicioná-la no suporte.

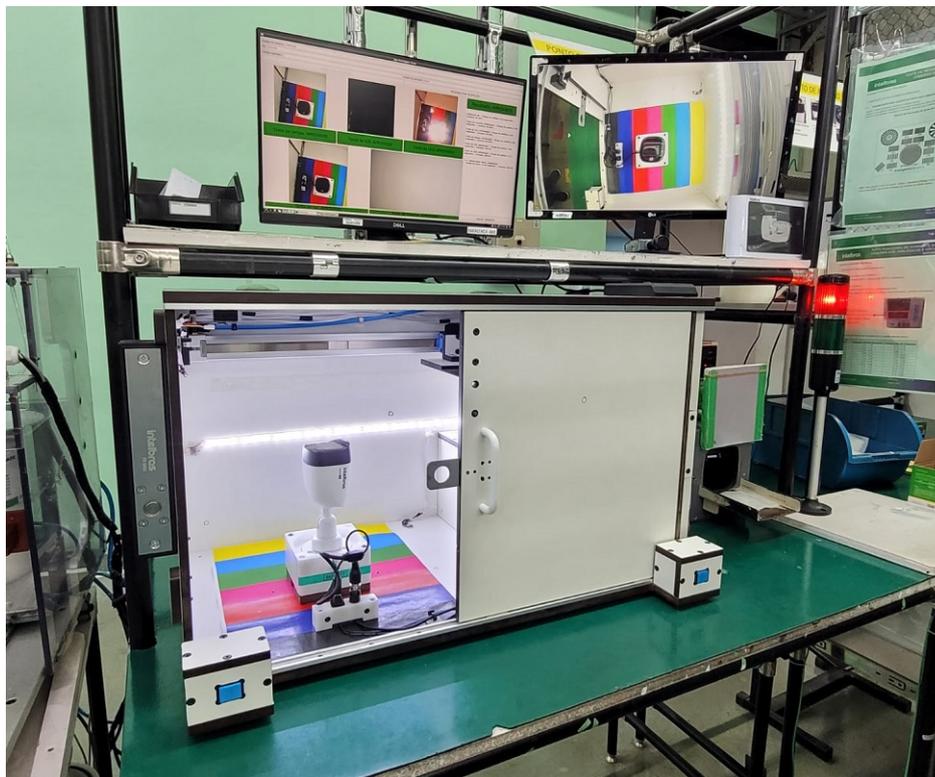
Assim, após o alcance do desempenho desejado, transportou-se o sistema para a implementação na fábrica, em Manaus. Esse processo foi realizado por um colega de equipe, em uma das células de produção. Através dessa etapa, foi possível verificar o desempenho real do sistema, coletar dados e sugestões das partes envolvidas. Com isso, foram considerados os pontos levantados em termos de ergonomia no posicionamento de botões e conectores, por exemplo, para a implementação definitiva da Jiga em todas as linhas de produção da fábrica. Na Figura 45 pode-se observar o sistema de testes operando na fábrica, em Manaus.

Nessa etapa de implementação foram coletados os dados referentes ao desempenho do sistema em termos das métricas tratadas até aqui e também quanto à repetibilidade do modelo. Verificou-se o atendimento dos requisitos de projeto, alinhou-se o planejamento das próximas etapas com os times envolvidos e documentou-se as adaptações propostas.

No desenvolvimento desse protótipo procurou-se utilizar componentes acessíveis que atendessem o funcionamento desejado. Porém, visando a implementação definitiva do projeto, faz-se necessária a adequação quanto às normas regulamentadoras. A movimentação pneumática da carta branca enquadra o equipamento desenvolvido na Norma Regulamentadora número 12, conforme apresentado no parágrafo 12.1.4 (NR12, 1978). Portanto, faz-se necessário o estudo da utilização de dispositivos de segurança além dos inicialmente incluídos, como, por exemplo, a partida do sistema por acionamento bimanual (prevista no parágrafo 12.4.3).

A alimentação da válvula solenoide, que comanda o funcionamento do cilin-

Figura 45 – Jiga de Testes implementada na fábrica em Manaus



Fonte: Repositório do projeto

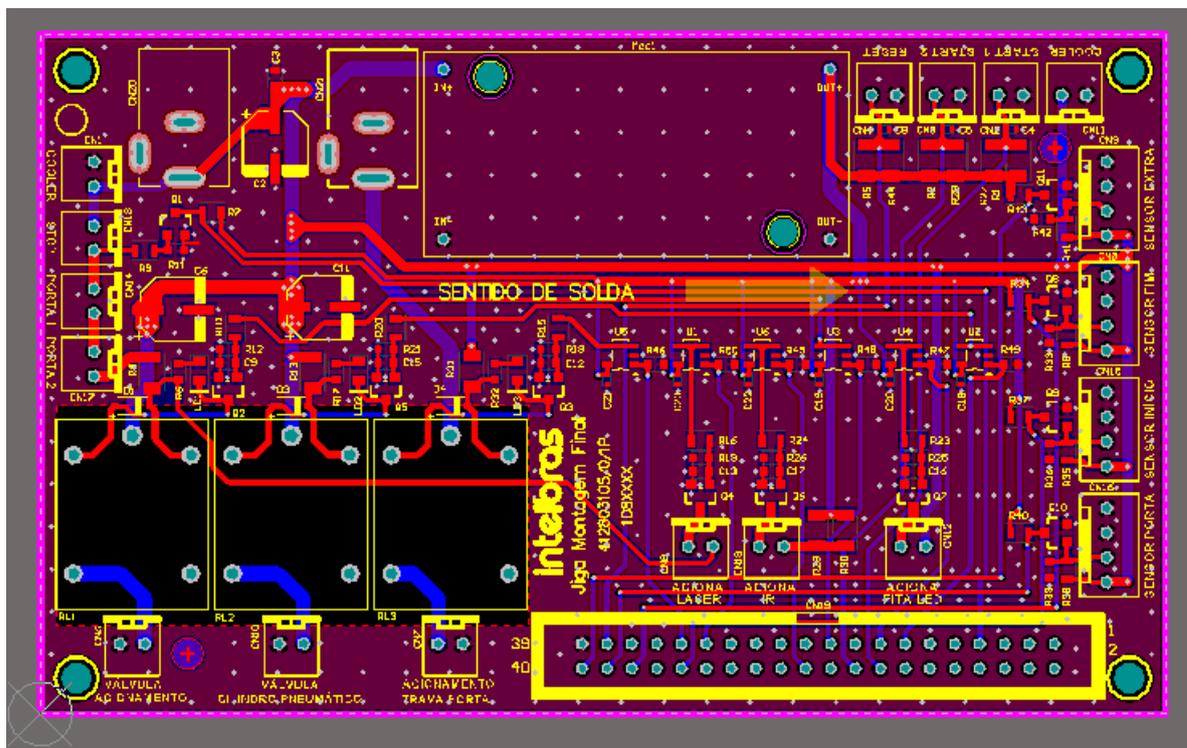
dro, condicionada ao fechamento da porta, através de um sensor mecânico, foi outra abordagem adotada no projeto que priorizou a minimização dos riscos de acidentes. Entretanto, a alteração do modelo de sensor utilizado e a necessidade de redundância são questões em estudo para que se possa definir a versão final do projeto.

Além disso, por conta da disponibilidade de recursos, utilizou-se, como trava de segurança da porta do sistema, um produto interno destinado a aplicações residenciais. Assim, nessa etapa de avaliação de melhorias também planeja-se readequar esse ponto.

Ademais, visitou-se o setor de montagem de placas SMD (Surface Mounted Device) da filial São José e avaliou-se a viabilidade de confeccionar as placas de circuito da Jiga de Testes, através desse processo automatizado. Requisitou-se ao time de layout a construção do projeto de circuito impresso, apresentado na Figura 46, e elaborou-se os processos internos para cadastro de estruturas de montagem que são interpretadas pelas máquinas SMD. Desse modo, a replicação das partes eletrônicas do projeto ocorre de forma mais ágil e menos suscetível a erros.

Na data de escrita deste documento, as Jigas para as demais bancadas de cada uma das linhas de produção encontram-se em fase de replicação. Após a finalização da construção das doze Jigas de Testes cabe ao autor acompanhar e garantir

Figura 46 – Projeto da placa de circuito impresso



Fonte: Repositório do projeto

a implementação adequada na filial Manaus, entendendo as necessidades da fábrica, ajustando detalhes no software e fornecendo treinamento às equipes.

5 ANÁLISE DE RESULTADOS

Com base nos registros de dados da produção e nas anotações feitas por um colega de equipe durante a viagem para Manaus, pôde-se levantar as informações acerca da incidência de defeitos na fábrica. Das câmeras que chegam a etapa de Montagem Final, aproximadamente:

- 0,3% não possuem a tampa plástica;
- 0,2% apresentam anomalias na cor das imagens geradas;
- 4,1% apresentam o defeito de mancha escura;
- 0,2% possuem mau funcionamento do filtro ICR;
- 0,2% não acionam o LED quando submetidas ao modo noturno.

Os impactos desses defeitos se tornam evidentes ao se considerar uma produção média diária de 15000 câmeras. Assim, a filtragem desses problemas é de suma importância para aumentar a eficiência do processo, reduzindo custos de retrabalho.

A redução de cerca de 6 segundos no tempo total de teste, com a inclusão da Jiga de Testes, pode viabilizar o aumento do número de câmeras produzidas por hora se associada a outras melhorias de processo em execução, como alteração de maquinário e de procedimentos internos. Assim, por meio desse projeto pode-se aumentar a produtividade, reduzir a necessidade de retrabalho e garantir maior qualidade ao produto desenvolvido. Além disso, viabiliza-se o levantamento de dados da produção, permitindo a construção de análises que embasam as decisões de otimização dos processos fabris.

Entretanto, os ganhos maiores se manifestam na qualidade do teste realizado, pois, na prática, existem metas de produção de câmeras por hora que impactam na realização do teste manual. Isso ocorre porque, em câmeras de alto volume de produção, o operador acaba negligenciando a etapa de conferência da qualidade do produto com o intuito de atingir o objetivo de produção. Assim, a alteração do procedimento atual para incluir o sistema inteligente de testes - composto por um marcador de produtos aprovados - faz com que se tenha uma forma padronizada de se avaliar todas as câmeras produzidas.

Avaliando-se o desempenho do sistema proposto com os dados coletados durante a implementação, verificou-se que:

- 1 em cada 250 câmeras são classificadas incorretamente como defeituosas no teste de tampa, ou seja, aponta-se a ausência de tampa de forma indevida.
- 4 em cada 650 câmeras apresentam manchas que poderiam ser aprovadas segundo o critério de 30%, mas são reprovadas no teste mancha pelo algoritmo.

- Não foi constatada a ocorrência de predições incorretas para os testes de cor, ICR e LED.

A presença de uma proteção plástica sobre as tampas das câmeras dificulta o processo de classificação, podendo ser interpretada como uma das causas dessa imprecisão. Já a classificação entre diferentes níveis de mancha acaba sendo problemática e de difícil distinção até mesmo para o setor de Controle de Qualidade. Assim, a criticidade de se aprovar uma câmera com defeito é muito maior do que se reprovar uma câmera boa, pois implica no risco de retrabalho de todo um lote de produção ou de entrega de um produto final problemático ao cliente. Por isso, pode-se considerar os resultados práticos alcançados com a implementação do novo sistema de testes como satisfatórios.

A matriz de confusão dos dados coletados durante a produção de 650 câmeras pode ser observada na Figura 47. Ainda, percebeu-se que os operadores reprovam qualquer mancha percebida, ignorando a tolerância de 30% e, algumas vezes por desatenção, deixam de reprovar manchas que ultrapassam esse critério. Assim, entende-se que nas amostras coletadas o desempenho da Jiga foi superior por não implicar em problemas de qualidade, com a aprovação de câmeras ruins e, também, garantiu a tolerância para boa parte dos casos.

Figura 47 – Matriz de confusão para as câmeras em produção

	Real: Mancha	Real: OK
Predito: Mancha	28	4
Predito: OK	0	618

Autoria própria

Aplicando-se as métricas abordadas neste trabalho para a matriz de confusão, tem-se:

- Precision: 87,50%;
- Recall: 100%;
- F1-Score: 93,33%;
- Acurácia: 99,38%.

A discussão levantada na seção 2.3 acerca do desbalanceamento de dados se aplica à comparação entre o resultado obtido com a métrica F1-Score e a Acurácia.

Pois, ao observar-se exclusivamente a acurácia pode-se interpretar incorretamente a capacidade do modelo em classificar defeitos. A aprovação de câmeras defeituosas poderia ser ainda mais penalizada com a utilização da métrica F2-Score, por exemplo.

Além disso, visando entender a capacidade do modelo em apresentar resultados consistentes, calculou-se a variação na saída a partir do fornecimento das mesmas entradas sucessivamente. A avaliação desse ponto, tratado aqui como repetibilidade, se deu através de cinco repetições de cada um dos testes para 30 câmeras. Ao fim desse procedimento, não foram verificadas variações nos resultados de classificação fornecidos pelo sistema. Entende-se que um levantamento estatístico definitivo exige um número específico de amostras para que se possa inferir os valores de repetibilidade com uma margem de erro determinada. Porém, o intuito desse cálculo é apenas estimar a qualidade do modelo a partir dos recursos disponíveis.

Os ganhos do projeto são potencializados quando aplicados a todas as 12 linhas de produção. Assim, implementando-se o sistema em todo o setor de Montagem Final e considerando a produção diária de aproximadamente 15000 câmeras, espera-se uma filtragem de praticamente todos os defeitos que seriam posteriormente identificados pelo Controle de Qualidade.

O desempenho do operador na identificação de defeitos se mostra difícil de mensurar porque a inspeção feita pelo Controle de Qualidade é aplicada em apenas uma pequena amostra de cada lote, composto por 120 peças. Entretanto, estima-se que, para o problema de mancha, 0,05% das câmeras defeituosas não são filtradas. Além disso, cerca de 1% das câmeras possuem manchas toleráveis e a maior parte é reprovada indevidamente.

A expressividade desses números se dá através da análise dos registros de lotes bloqueados dos últimos 6 meses. Constatou-se que 28 lotes de produção foram bloqueados por conta do problema de mancha escura, totalizando 3360 câmeras que precisaram ser inspecionadas, sendo que 34 delas realmente apresentavam esse defeito. Assim, o tempo despendido na conferência de todas essas câmeras poderia ter sido resumido ao retrabalho de apenas 34 unidades (com 150 segundos por unidade), caso a Jiga atuasse filtrando-as previamente. Vale ressaltar que o número de câmeras com manchas que não foram filtradas pelo teste da Montagem Final provavelmente é maior que esse valor verificado, em virtude das questões de amostragem mencionadas.

6 CONSIDERAÇÕES FINAIS

6.1 IMPACTOS DO TRABALHO

Este trabalho visou redefinir os procedimentos manuais de testagem de câmeras analógicas, executados durante a produção, em Manaus. Isso é feito utilizando-se modelos treinados por meio de Redes Neurais Convolucionais, que são aplicados às imagens geradas pelas câmeras, identificando a presença dos cinco defeitos mais comuns em termos de lotes bloqueados pelo Controle de Qualidade.

Além disso, para simular adequadamente as condições que permitem induzir a ocorrência de falha nos produtos, é preciso alternar os cenários dentro do sistema de testes. Por exemplo, a verificação do funcionamento do LED da câmera só é possível quando o nível de luminosidade, captado pelo sensor de imagem do produto, é baixo o suficiente. Assim, fez-se necessária uma sequência de acionamentos coordenada com o disparo de cada modelo classificador para que sejam captadas as imagens correspondentes em cada cenário de teste.

O sistema proposto atua, portanto, em conjunto com os operadores, que possuem a função de inserir as câmeras para o início do teste e removê-las ao fim, destinando-as com base no resultado apresentado na interface.

Após as etapas de estudo, desenvolvimento e implementação realizadas na unidade Matriz, em São José, foi feita a validação do projeto piloto na fábrica, em Manaus, por um colega de equipe. Os dados levantados ao longo de três semanas indicam uma redução no tempo médio de teste em cerca de 20%, já que o procedimento, antes executado em aproximadamente 27 segundos, passou a ser executado em cerca de 21 segundos. Com isso, pode-se adequar as demais etapas do processo produtivo, visando o aumento do número de câmeras produzidas por hora.

Além dos ganhos no tempo de execução dos testes, estimam-se melhorias na eficácia do processo, já que cerca de 4,1% das câmeras que chegam à etapa de Montagem Final apresentam o problema de mancha escura e pelo menos 0,05% delas não eram identificadas pelos operadores. Considerando-se uma produção diária de 1200 câmeras na célula acompanhada e a ausência de classificações incorretas para Falso Negativos, ou seja, falha do sistema quanto à percepção de um problema, durante as três semanas citadas, pode-se considerar como atendido o objetivo de aprimoramento do desempenho desse processo.

A automatização, já em andamento, de outras etapas do processo produtivo juntamente com a etapa impactada por este projeto permite o aumento da produtividade e a redução de custos com trabalho manual. Com a replicação do projeto para todas as células pode-se evitar a inspeção de cerca de 500 produtos por mês, reduzindo-se custos com retrabalho. Espera-se também a diminuição do retrabalho de câmeras com manchas aceitáveis. Além disso, com o aumento do desempenho do teste, minimizam-

se as chances de entregar um produto que não atende os padrões de qualidade ao cliente.

Ainda, os ganhos do projeto extrapolam a questão econômica, pois as atividades desempenhadas pelos operadores nessa etapa de testagem das câmeras acabam sendo repetitivas e podem trazer algum desgaste aos mesmos. Assim, espera-se reduzir também o número de ações realizadas por um operador durante essa etapa, com o intuito de minimizar ainda mais a possibilidade de riscos à sua saúde e segurança.

6.2 CONCLUSÃO

A alteração do sistema de testes para toda a linha de produção de câmeras analógicas se mostra como um dos primeiros passos para a expansão do uso de técnicas de Visão Computacional na referida fábrica. A ampla existência de processos manuais e os benefícios a longo prazo desta mudança são fatores que indicam a tendência de modernização e aperfeiçoamento como um todo.

Tratando-se de trabalhos futuros, o autor já tem desenvolvido o software de outro sistema de identificação de defeitos para um produto produzido na fábrica da filial São José. Além das técnicas aqui descritas também contempla-se a abordagem por meio de Morfologia Matemática. O projeto em andamento herda várias características deste trabalho e demonstra a abrangência de aplicações desse tipo de solução. Em suma, analisa-se a integridade dos canais da placa de um produto responsável pela transmissão de sinais de vídeo e alimentação de câmeras. Isso é feito através da identificação de defeitos típicos pela abordagem de Deep Learning, como: ruídos, formação de imagens em preto e branco e desfoque, além da contagem do número de LEDs indicativos do produto, por técnicas tradicionais de Visão Computacional.

Figura 48 – Jiga de Testes em desenvolvimento



Fonte: Autoria própria

Por meio do desenvolvimento deste projeto explorou-se a aplicação do ferramental teórico, construído ao longo do curso, para a solução de diversos problemas de engenharia. O contato com outros profissionais da área durante o processo foi enriquecedor devido ao empenho no compartilhamento do conhecimento. Além disso, as visitas ao parque fabril, em São José, contribuíram para a compreensão do funcionamento das atividades no ambiente industrial, analisando-se a implementação de muitas das estratégias apresentadas durante o curso.

Por fim, entende-se que o trabalho engloba vários aspectos conceituais estu-

dados no curso de Engenharia de Controle e Automação, como: circuitos eletrônicos, inteligência artificial, acionamentos pneumáticos, programação concorrente, modelagem de software, entre outros. Assim, a aplicação multidisciplinar desses conceitos no ambiente fabril encontra-se no cerne das atividades desenvolvidas durante o curso.

REFERÊNCIAS

ACADEMY, Data Science. **Deep Learning Book**. [S./], 2022. Disponível em: <https://www.deeplearningbook.com.br/>. Acesso em: 3 jul. 2022.

BELTON. **Válvula solenoide: o que é e como funciona**. [S./], 2020. Disponível em: <https://www.belton.com.br/valvula-solenoide-o-que-e-e-como-funciona>. Acesso em: 13 jul. 2022.

BISCHOFF, Paul. **Surveillance camera statistics: which cities have the most CCTV cameras?** [S./], 2021. Disponível em: <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/>. Acesso em: 27 jun. 2022.

BROWNLEE, Jason. **Tour of Evaluation Metrics for Imbalanced Classification**. [S./], 2020. Disponível em: <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/>. Acesso em: 26 jun. 2022.

BROWNLEE, Jason. **What is the Difference Between Test and Validation Datasets?** [S./], 2017. Disponível em: <https://machinelearningmastery.com/difference-test-validation-datasets/>. Acesso em: 10 jul. 2022.

CANTO, Eduardo Camilo do. **Software com inteligência artificial para detecção automática de manchas na imagem de câmeras durante processo produtivo**. [S./], 2022. Disponível em: <https://repositorio.ufsc.br/handle/123456789/233318>.. Acesso em: 21 jun. 2022.

EDMUND, Optics. **Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications**. [S./], 2022. Disponível em: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/>. Acesso em: 21 jun. 2022.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. **Deep Residual Learning for Image Recognition**. [S./], 2015. Disponível em: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf. Acesso em: 8 jul. 2022.

HOWARD, Jeremy; THOMAS, Rachel. **fastai**. [S./], 2016. Disponível em: <https://github.com/fastai/fastai>. Acesso em: 12 jul. 2022.

IBM. **Modelando o Comportamento do Objeto Utilizando Diagramas de Máquina de Estado**. [S./], 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=diagrams-state-machines>. Acesso em: 12 jul. 2022.

IBM. **What are Neural Networks?** [S./], 2020. Disponível em: <https://www.ibm.com/cloud/learn/neural-networks>. Acesso em: 3 jul. 2022.

KAPATKER, Jayant. **UNDERSTANDING THE VIDEO SIGNAL**. [S./], 1997. Disponível em: <https://clearview-communications.com/wp-content/uploads/2020/04/Understanding-the-Video-Signal.pdf>. Acesso em: 21 jun. 2022.

KLUG, Brian. **Understanding Camera Optics Smartphone Camera Trends**. [S./], 2013. Disponível em: <https://www.anandtech.com/show/6777/understanding-camera-optics-smartphone-camera-trends>. Acesso em: 21 jun. 2022.

LEARNCCTV. **What is ICR (Infrared Cutoff Removal) for Day/Night cameras?** [S./], 2018. Disponível em: <https://learncctv.com/what-is-icr/>. Acesso em: 21 jun. 2022.

LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. **Gradient-based learning applied to document recognition**. [S./], 1998. Disponível em: <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>. Acesso em: 6 jul. 2022.

MEDEIROS, Higor. **Introdução ao Padrão MVC**. [S./], 2013. Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>. Acesso em: 12 jul. 2022.

NR12. **SEGURANÇA NO TRABALHO EM MÁQUINAS E EQUIPAMENTOS**. [S./], 1978. Disponível em: <https://www.gov.br/trabalho-e-previdencia/pt-br/composicao/orgaos-especificos/secretaria-de-trabalho/inspecao/seguranca-e-saude-no-trabalho/normas-regulamentadoras/nr-12.pdf>. Acesso em: 18 jul. 2022.

OTUYAMA, Júlio M. **Introduction to Convolutional Neural Networks**. [S.l.], 2009. Disponível em: <https://www.ime.usp.br/~otuyama/academic/cnn/index.html>. Acesso em: 11 jul. 2022.

OU, XIANFENG; YAN, PENGCHENG; ZHANG, YIMING; TU, BING; ZHANG, GUOYUN; WU, JIANHUI; LI, WUJING. **Moving Object Detection Method via ResNet-18 With Encoder–Decoder Structure in Complex Scenes**. [S.l.], 2019. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8781779>. Acesso em: 11 jul. 2022.

QUINT, Rafael Ramos S. **Desenvolvimento de Câmeras Analógicas: Aprimoramento no Processo de Testes**. [S.l.], 2022. Acesso em: 21 jun. 2022.

WANG, Chi-Feng. **The Vanishing Gradient Problem: The Problem, Its Causes, Its Significance, and Its Solutions**. [S.l.], 2019. Disponível em: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>. Acesso em: 8 jul. 2022.

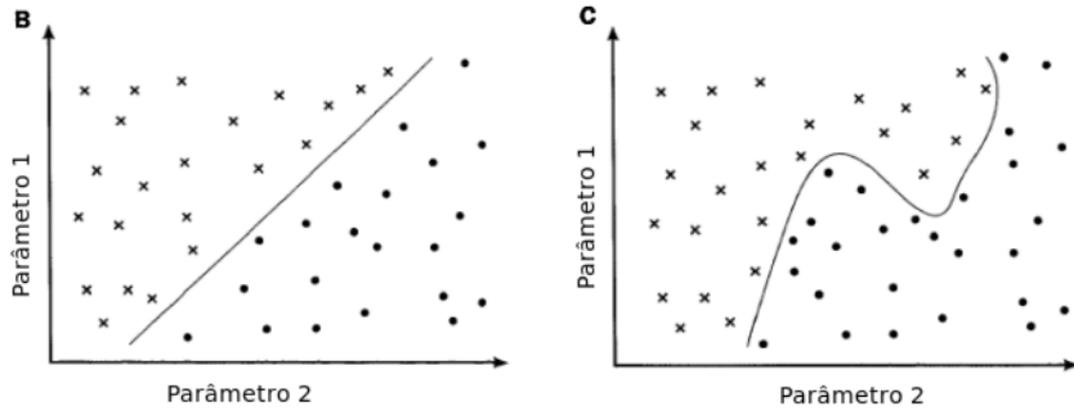
WANGENHEIM, Aldo von. **Introdução a Redes Neurais Convolucionais e Deep Learning em Visão Computacional**. [S.l.], 2019. Disponível em: <https://www.youtube.com/watch?v=sWqsLKxOLYc&list=PLmDIGfkfgKy1SBjXA0kBk4DAhIaN1vQOS&index=13>. Acesso em: 12 jul. 2022.

WANGENHEIM, Aldo von. **Técnicas Subsimbólicas: Redes Neurais**. [S.l.], 2004. Disponível em: <https://www.inf.ufsc.br/~aldo.vw/patrec/subsimbolicas1.pdf>. Acesso em: 3 jul. 2022.

YOSINSKI, Jason; CLUNE, Jeff; BENGIO, Yoshua; LIPSON, Hod. **How transferable are features in deep neural networks?** [S.l.], 2014. Disponível em: <https://arxiv.org/pdf/1411.1792.pdf>. Acesso em: 10 jul. 2022.

ANEXO A – LINEARIDADE DE PADRÕES

Figura 49 – Exemplos de padrões linearmente e não-linearmente separável respectivamente.



Autor: André Ricardo Gonçalves