



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO

ERICK ERATE DOS SANTOS

AUTOMATIZAÇÃO DE UM PROCESSO DE INSPEÇÃO VISUAL
MULTICLASSE COM O USO DE UM SISTEMA DE VISÃO
COMPUTACIONAL BASEADO EM DEEP LEARNING

CURITIBA - PR

2022

ERICK ERATE DOS SANTOS

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em
Engenharia da Computação pela Escola
Superior Politécnica do Centro
Universitário Internacional Uninter.

Orientadora: Prof^a. M^a. Dayane Perez
Bravo.

CURITIBA - PR
2022

AGRADECIMENTOS

À Deus, pelo direito a vida, e por me ajudar a superar todas as dificuldades encontradas ao longo da minha trajetória acadêmica.

Aos meus pais Edson e Roseli, junto às minhas irmãs Emanuely e Emilly, que me incentivaram nos momentos mais difíceis dessa caminhada. Sou grato à minha namorada Kelly, pelo apoio e compreensão nos momentos em que estive ausente me dedicando à realização deste trabalho. Agradeço a minha tia Hérica por contribuir para que este sonho se tornasse realidade. Aos demais familiares e amigos que de alguma forma tiveram a sua contribuição.

À empresa Sotes Sistemas pela oportunidade confiada a mim.

Às mestras Cristiane e Dayane, pela orientação. Sem vocês, a conclusão deste trabalho não atingiria tamanha excelência. Gratidão pelos encontros, ensinamentos e risadas que tornaram este momento ainda mais especial.

RESUMO

Graças a evolução no poder computacional e na quantidade de dados disponíveis, os sistemas de visão computacional baseados em técnicas de inteligência artificial conseguiram alcançar um desempenho superior aos sistemas de visão baseado em regras (COGNEX). Podemos citar como exemplo tarefas de inspeção visual, como classificação de imagens, vídeos e detecção de objetos no âmbito industrial. No entanto, após a avaliação dos sistemas de visão computacional disponíveis na indústria, identificou-se a inexistência da aplicação de um índice de capacidade de processo. Esse índice serve para quantificar o nível de desempenho do processo em atender às especificações definidas pelo projeto. O sistema implementado neste trabalho (Mosaic Office) gera um relatório técnico com o índice de capacidade potencial CP para avaliar a capacidade do processo inerente à inspeção visual multiclasse. Para construir um modelo de aprendizagem de máquina, este sistema utiliza a CNN ResNetV2 50 junto a duas implementações de algoritmos de treinamento disponíveis: Treinamento Clássico e Treinamento Contínuo. Foi aplicada uma ferramenta de geometria (ROI – Região de Interesse) para descartar informações de pixels irrelevantes às características de imagens avaliadas. Além disso, uma melhoria proposta pelo trabalho foi o uso de interfaces gráficas para expor ao usuário as técnicas de aprendizado profundo. O conjunto utilizado continha 28.287 imagens pré-classificadas em 6 classes. Em caráter de comparação de desempenho com um dos sistemas disponíveis na indústria, o Mosaic Office foi aproximadamente 99% mais rápido e aproximadamente 37% mais assertivo.

Palavras-chave: Inteligência Artificial; Aprendizado Profundo; Computer Vision.

LISTA DE FIGURAS

FIGURA 1 - EXEMPLOS DE TAREFAS INDUSTRIAIS	11
FIGURA 2 - REQUISITOS DE HARDWARE RECOMENDADOS	15
FIGURA 3 - ESTRUTURA DE UM NEURÔNIO	18
FIGURA 4 - MODELO DO NEURÔNIO ARTIFICIAL PROPOSTO POR MCCULLOCH E PITTS.....	19
FIGURA 5 - REDE NEURAL ARTIFICIAL COM UMA CAMADA OCULTA	20
FIGURA 6 - MODOS DE OPERAÇÃO DAS REDES NEURAIIS	21
FIGURA 7 - DISCRIMINANTES	23
FIGURA 8 - REDE NEURAL ARTIFICIAL DE TRÊS CAMADAS.....	24
FIGURA 9 - CAMADAS DA ARQUITETURA CNN.....	29
FIGURA 10 - TIPOS DE BLOCOS RESIDUAIS.....	31
FIGURA 11 - OPERAÇÕES INICIAIS DE CONVOLUÇÃO E SUMARIZAÇÃO	37
FIGURA 12 - ESTÁGIO 1.....	37
FIGURA 13 - ESTÁGIO 2.....	38
FIGURA 14 - ESTÁGIO 3.....	39
FIGURA 15 - ESTÁGIO 4.....	40
FIGURA 16 - OPERAÇÃO FINAL DE SUMARIZAÇÃO E CAMADAS TOTALMENTE CONECTADAS	40
FIGURA 17 - PIPELINE DE TREINAMENTO DO MODELO.....	41
FIGURA 18 - ALGORITMO TREINO CONTÍNUO.....	43
FIGURA 19 - WORKSPACE	45
FIGURA 20 – FLUXO DE UTILIZAÇÃO DAS INTERFACES GRÁFICAS.....	46
FIGURA 21 - INTERFACE GRÁFICA DE TREINO	48
FIGURA 22 - CONFIGURAÇÕES DE IDENTIFICAÇÃO	50
FIGURA 23 - CONFIGURAÇÕES DO CLASSIFICADOR.....	51
FIGURA 24 - CONFIGURAÇÕES DAS FERRAMENTA DE GEOMETRIA.....	52
FIGURA 25 - INTERFACE GRÁFICA DE TESTE	53
FIGURA 26 - INTERFACE GRÁFICA DE TESTE COM FILTRO DE SELEÇÃO	54
FIGURA 27 - INTERFACE GRÁFICA DE ERRO	55
FIGURA 28 - INTERFACE GRÁFICA DE OPERAÇÃO	56
FIGURA 29 - INTERFACE GRÁFICA DE REGISTROS	57
FIGURA 30 - FERRAMENTA DE GEOMETRIA APLICADA - ROI	62

FIGURA 31 - VALORES MÍNIMOS RECOMENDADOS PARA A CAPACIDADE DE PROCESSO	66
FIGURA 32 - RELATÓRIO DE CAPABILIDADE	81

LISTA DE QUADROS

QUADRO 1 - DESCRIÇÃO DOS CONTROLES.....	48
---	----

LISTA DE TABELAS

TABELA 1 - FUNCIONALIDADES DOS SISTEMAS DE VISÃO COMPUTACIONAL	13
TABELA 2 - ESPECIFICAÇÕES DE <i>HARDWARE</i> E <i>SOFTWARE</i>	57
TABELA 3 – IMAGENS CARACTERÍSTICAS DAS CLASSES AVALIDAS	58
TABELA 4 - HIPER PARÂMETROS DE TREINAMENTO DO SISTEMA IMPLEMENTADO.....	62
TABELA 5 - HIPER PARÂMETROS DE TREINAMENTO DO SISTEMA DE VISÃO MATROX.....	63
TABELA 6 - SISTEMA DE VISÃO MATROX - ESTATÍSTICA POR CLASSE.....	69
TABELA 7 - MOSAIC OFFICE ESTATÍSTICA POR CLASSE	69
TABELA 8 - ESTATÍSTICA KAPPA	70
TABELA 9 - COMPARAÇÃO ENTRE OS SISTEMAS	71

LISTA DE SIGLAS

CNN	Rede Neural Convolucional (Convolutional Neural Network)
DNN	Rede Neural Profunda (Deep Neural Network)
RNA	Rede Neural Artificial (Artificial Neural Network)
CP	Capabilidade potencial (Potencial Capability)
CPU	Unidade Central de Processamento (Central Processing Unit)
GPU	Unidade de processamento gráfico (Graphics Processing Units)
RAM	Memória de acesso aleatório (<i>Random Access Memory</i>)
SDK	Kit de desenvolvimento de Software (Software Development Kit)
ROI	Região de interesse (<i>Region of interest</i>)
USL	Limite superior de especificação (<i>Upper Limit Specification</i>)
LSL	Limite inferior de especificação (<i>Lower Limite Specification</i>)

LISTA DE ACRÔNIMOS

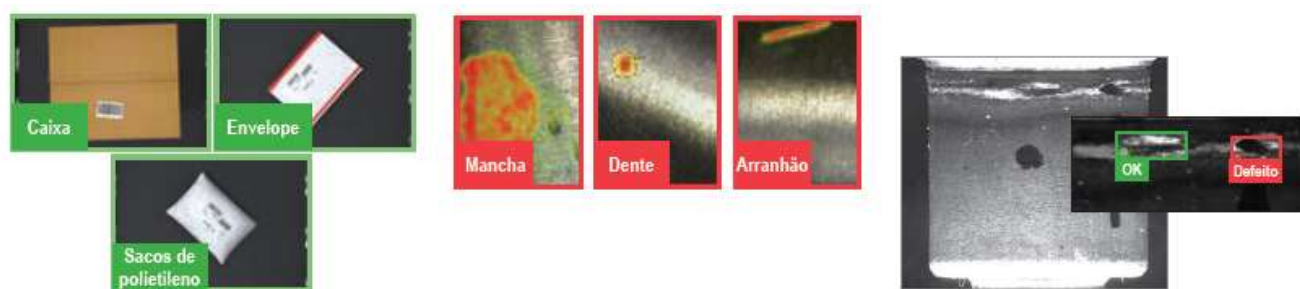
ResNet	Rede residual (<i>Residual Network</i>)
ReLu	Unidade linear retificada (<i>Rectified Linear Unit</i>)

1 INTRODUÇÃO

A inspeção visual automatizada tem sido amplamente utilizada no âmbito industrial, possuindo aplicações no controle de qualidade, classificação de produtos, dentre outros segmentos (GOES, 2014). Mediante as metodologias de reconhecimento de características existentes, destacam-se os classificadores orientados à imagem utilizando a tecnologia de aprendizado profundo, especificamente rede neural convolucional (CNN) (KHAN, RAHMANI, *et al.*, 2018).

De acordo com (GOES, 2014), atualmente o desenvolvimento de sistemas de inspeção visual automatizada baseada em aplicações da tecnologia de visão computacional desempenha um papel fundamental na busca da competitividade industrial. Com isso, a necessidade de automação da inspeção visual em tarefas industriais tem aumentado significativamente. Como exemplo das tarefas industriais em questão, tem-se a tarefa de classificação utilizada para distinguir entre diferentes tipos de objetos, identificar tipos de defeitos e inspecionar imagens, como ilustra a FIGURA 1 a seguir.

FIGURA 1 - EXEMPLOS DE TAREFAS INDUSTRIAIS



FONTE: (COGNEX, 2021).

A necessidade de automação da inspeção visual incentiva a implementação de sistemas inteligentes responsáveis pela inspeção de características com precisão e alta velocidade. Dentre os principais benefícios, tem-se o aumento da produtividade, melhoria na qualidade de produtos, redução de riscos ocupacionais, redução de custos, dentre outros.

Este trabalho tem como objetivo a implementação de um sistema de visão computacional baseado em técnicas de inteligência artificial, em especial aprendizado

profundo, para automatização do processo de inspeção visual na indústria. O sistema em questão (Mosaic Office) utiliza a arquitetura de rede residual ResNetV2 50 juntamente à biblioteca de desenvolvimento ML.NET¹, responsável por criar e treinar nativamente um modelo de aprendizagem de máquina.

O Mosaic Office fornece acesso às técnicas de aprendizado profundo através de interfaces gráficas que simplificam o processo de alimentação do modelo em 3 etapas: 1) Coletar e pré-classificar as amostras de imagens; 2) Treinar o modelo de aprendizagem de máquina com alguma das implementações de algoritmos de treinamento: Treinamento Contínuo e Treinamento Clássico e 3) Testar e aprimorar o modelo de aprendizagem de máquina.

Para (POLHEMUS, 2017), o controle da qualidade de processo diz respeito à utilização de métodos estatísticos que visam avaliar a condição de um processo em atender às especificações de uma característica. Sendo assim, o emprego do controle de qualidade no âmbito industrial traz benefícios como o aumento da produtividade, melhoria na qualidade de produtos e redução dos riscos ocupacionais. (KANE, 1986) afirma que a estimativa de qualidade de um processo é dada em função da capacidade deste processo, podendo ser especificada por meio de índices que quantificam o nível de desempenho do processo em atender os requisitos do projeto.

O sistema implementado neste trabalho inclui a funcionalidade de controle de qualidade de processo visando atender à demanda industrial. Para isso, o sistema emprega o uso do índice de capacidade potencial (CP) e permite a exportação de um relatório estatístico de capacidade da aplicação para visualização dos resultados obtidos na rotina de teste do modelo. Dentre as principais métricas, tem-se a quantidade de acertos e erros de cada classe, juntamente à quantidade de classificações com pontuações abaixo do limite de tolerância e o valor do índice de capacidade atribuído à cada classe. Como método de avaliação do sistema implementado neste trabalho, será feita uma comparação estatística com um dos sistemas disponíveis na indústria, a fim de verificar o desempenho destes sistemas em uma tarefa de classificação multiclasse.

As principais contribuições deste trabalho são: (i) fornecer acesso às técnicas de aprendizado profundo por meio das interfaces gráficas que simplificam o processo de implementação de aplicações no contexto de aprendizagem de máquina; (ii)

¹ Disponível em: <https://dotnet.microsoft.com/apps/machinelearning-ai>

implementar um algoritmo para otimizar o desempenho do modelo através de ciclos estratégicos de treino e teste; (iii) implementar um modelo probabilístico do processo para calcular a probabilidade de ocorrência de classificações com probabilidades inferiores ao limite estabelecido pelos requisitos do projeto; (iv) quantificar o nível de desempenho do processo em atender aos requisitos do projeto através do emprego do índice de capacidade do processo CP;

Este trabalho está organizado da seguinte forma: no Capítulo 2 descreveu-se os sistemas de visão computacional disponíveis no mercado industrial; no Capítulo 3 realizou-se uma breve revisão teórica; no Capítulo 4 apresentou-se o método proposto; no Capítulo 5 registrou-se os resultados obtidos; e no Capítulo 6 destacou-se as conclusões e sugestões para trabalhos futuros.

2 SISTEMAS DE VISÃO COMPUTACIONAL DISPONÍVEIS NA INDÚSTRIA

De acordo com (GALLON, 2013), existem diversos sistemas de visão computacional já consolidados na indústria capazes de realizar tarefas de classificação de imagens. Eles oferecem ferramentas que propõem soluções sobre as mais diversas necessidades de classificações, como verificação de defeitos no produto, distinção de grupos de características, etc.

A TABELA 1 exibe as principais funcionalidades dos sistemas de visão computacional disponíveis na indústria.

TABELA 1 - FUNCIONALIDADES DOS SISTEMAS DE VISÃO COMPUTACIONAL²

	Keyence	Cognex	Matrox
Software	IV2 Navigator	VisionPro Deep Learning	SDK Imaging Library Tools
Hardware	Hardware próprio	Computador apropriado	Computador apropriado
Custo	R\$40.000,00	R\$70.000,00 + Hardware	R\$50.000,00 + Hardware
Tipo do classificador	Binário	Multiclasse	Multiclasse

² Informações extraídas através dos representantes comerciais dos sistemas de visão computacional.

Treinamento	Limitado a 86 imagens	Ilimitado	Ilimitado
--------------------	--------------------------	-----------	-----------

FONTE: O autor (2022).

Algumas informações adicionais sobre esses sistemas são descritas a seguir.

2.1 KEYENCE: IV2 – NAVIGATOR

O sistema proposto pela fabricante *Keyence – IV2 Navigator* – consiste em um software com inteligência artificial integrada. Das opções de sistemas de visão computacional disponíveis na indústria, esta é a opção que oferece um conjunto de ferramentas inferior às demais e possui menor custo (KEYENCE, 2021).

O sistema de visão computacional consiste em uma câmera para fazer a captura das imagens, um monitor para oferecer a interface com o usuário e um amplificador responsável por armazenar configurações de cada linha de produto, dados de imagens, etc. Este sistema opera em conjunto, ou seja, não é possível realizar a classificação de uma imagem qualquer cuja origem não seja da própria câmera do sistema (KEYENCE, 2021).

O custo da licença vitalícia deste sistema é a partir de R\$40.000,00 com suporte técnico ao cliente. O treinamento é limitado a 86 imagens e a apenas duas classes: OK para imagens aprovadas e NG para imagens reprovadas. Essas características fazem deste uma solução não recomendada para aplicações na indústria que necessitam de uma tarefa de classificação de imagens multiclasse. Neste trabalho o sistema desenvolvido atende à necessidade de classificação de imagens multiclasse, assim como os sistemas descritos a seguir (KEYENCE, 2021).

2.2 COGNEX: VISIONPRO DEEP LEARNING

O sistema *VisionPro Deep Learning* oferece um software de análise de imagem com base em *Deep Learning*. Projetado para automação industrial, oferece quatro ferramentas para sistemas de visão computacional: localização da peça e verificação de montagem, detecção de defeitos; classificação de objetos e cenas, e leitura de caracteres (COGNEX, 2021).

O software de classificação de imagens multiclasse implementado por este sistema pode distinguir diferentes tipos de características, identificar tipos de defeitos e inspecionar imagens (COGNEX, 2021).

Os requisitos de hardware recomendados para garantir um bom desempenho do classificador de imagens são apresentadas na FIGURA 2 a seguir. Além disso, a licença vitalícia deste sistema é em torno de R\$70.000,00 com suporte técnico ao cliente (COGNEX, 2021).

FIGURA 2 - REQUISITOS DE HARDWARE RECOMENDADOS

ESPECIFICAÇÕES		
Requisitos de sistema e hardware	CPU	Intel Core i7 ou superior
	GPU	Compatível apenas com GPU's NVIDIA Memória GPU de 11GB ou superior (GTX 1080Ti, RTX 2080Ti, 3070, 3080, 3090)
	Memória RAM	32 GB ou mais
	Sistema	Windows 10 64-bit Windows Server 2019 64-bit
	Armazenamento	Unidade solid-state drive (SSD) com 100 GB ou mais de espaço livre
Formatos de arquivo de imagem suportados	PNG, BMP, TIFF, JPEG	
Propriedades de imagem compatíveis	1-4 canais, 8 ou 16 bits	

FONTE: Extraído de catálogo (2022).³

Destaca-se as especificações de CPU, GPU e memória RAM como indicativo de custo adicional para a utilização do sistema *VisionPro Deep Learning*. O sistema desenvolvido neste trabalho se beneficia de um hardware robusto como descrito na figura acima, porém não exige tal configuração recomendada para o seu funcionamento.

2.3 MATROX: SDK IMAGING LIBRARY TOOLS

A solução proposta pela fabricante *Matrox* consiste em uma coleção de ferramentas de software para o desenvolvimento de aplicativos de visão de máquina e análise de imagens. O kit de desenvolvimento oferece ferramentas como a

³ Disponível em <https://www.cognex.com/pt-br/downloads/visionpro-deep-learning-datasheet>

calibração de imagens; classificação para categorizar automaticamente o conteúdo da imagem; aprimoramento e transformação de imagens; localização de objetos; extração e medição de características; ler cadeias de caracteres e decodificação e verificação de marcas de identificação. Este sistema oferece um ambiente para facilitar e acelerar a avaliação e prototipagem de uma aplicação, como a aplicação de classificação de imagens multiclasse (MATROX, 2022).

O kit de desenvolvimento de Software inclui ferramentas que potencializam o aprendizado de máquina, incluindo o aprendizado profundo para classificar uma imagem. É uma solução já consolidada na indústria para a automatização de tarefas como classificação de imagens, detecção de presença de objetos, reconhecimento de caracteres e outros. O custo da licença vitalícia do kit de desenvolvimento é a partir de R\$50.000,00 com suporte técnico ao cliente.

Este sistema fornece diferentes classificadores predefinidos para resolver problemas distintos. As opções de contexto de classificadores disponíveis atendem desde aplicações de classificações simples até aplicações projetadas para lidar com problemas altamente complexos. Diferentes aspectos do problema precisam ser considerados para selecionar a arquitetura e modo de treinamento específicos deste sistema, como a quantidade de dados disponíveis, a velocidade necessária e a complexidade do problema (MATROX, 2022).

Os contextos de classificadores são definidos em três categorias: pequeno, médio e extra grande. A principal diferença entre estas categorias é o tamanho de imagem que o classificador é projetado para manipular. Problemas mais complexos geralmente exigem classificadores que lidam com imagens maiores e consequentemente levam mais tempo para serem treinados (MATROX, 2022).

Tendo analisado os sistemas de visão disponíveis no mercado, percebeu-se a oportunidade de criação de um novo sistema que facilite o processo de construção do modelo de aprendizagem de máquina utilizado nas tarefas de classificação de imagem. Convém também a implementação de um algoritmo de treinamento com o intuito de diminuir o tempo de treinamento da rede neural e otimizar o desempenho do modelo treinado. Bem como a elaboração da funcionalidade de capacidade do processo inerente à tarefa de classificação de imagem visando atender à demanda industrial (MATROX, 2022).

3 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho está organizada em sete seções: 1. Conceituação do elemento base de uma estrutura conexionista – O neurônio biológico; 2. Apresentação do modelo de neurônio artificial e seu funcionamento; 3. Características das Redes Neurais Artificiais; 4. Definição de Aprendizado profundo junto a sua utilização; 5. Apresentação das Redes Neurais Convolucionais e suas características; 6. Definição de Redes Residuais e sua topologia; e 7. Definição de técnicas estatísticas utilizadas para avaliar a capacidade do sistema através do modelo probabilístico do processo e índice de capacidade.

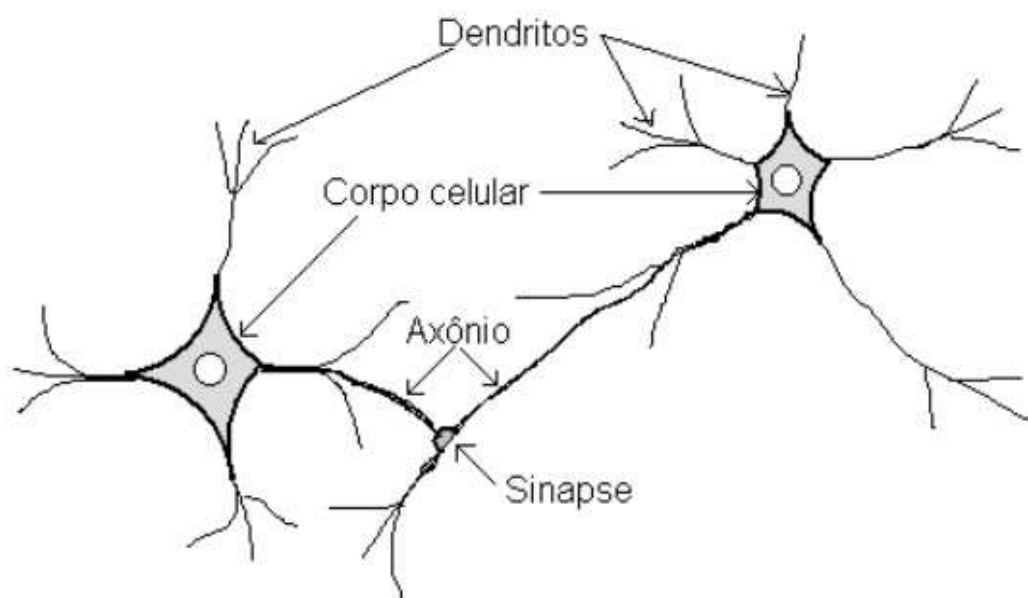
3.1 INSPIRAÇÃO BIOLÓGICA

O cérebro humano é um sistema cognitivo composto de neurônios interconectados por uma rede de sinapses e que cooperam entre si para processar informações de maneira distribuída (SILVA, 2003). Um neurônio ou célula nervosa se diferencia das outras células por ser capaz de responder à estímulos do meio em que se encontram, através de modificações da diferença de potencial elétrico existente entre a superfície interna e externa da membrana celular (BRAGA, CARVALHO e LUDEMIR, 2007).

Sua estrutura morfológica é formada por um corpo celular contendo um núcleo, que é o centro metabólico do neurônio (BRAGA, CARVALHO e LUDEMIR, 2007). A comunicação com outros neurônios se dá através de ramificações conhecidas como dendritos, responsáveis por receber as informações, ou impulsos nervosos, e conduzi-las até o corpo celular. Aqui, a informação é processada e novos impulsos são gerados.

Existe também uma ramificação prolongada única chamada axônio, responsável por transmitir os impulsos do corpo celular para suas extremidades. As extremidades do axônio são conectadas com dendritos de outros neurônios pelos pontos de contato chamados de sinapses, conforme ilustra a FIGURA 3 a seguir.

FIGURA 3 - ESTRUTURA DE UM NEURÔNIO



FONTE: Extraído de (MINUSSI, 2008).

É por meio das sinapses que os neurônios se unem funcionalmente, formando redes neurais. As sinapses funcionam como válvulas e são capazes de controlar a transmissão de impulsos - isto é, o fluxo da informação - entre os neurônios na rede neural (BRAGA, CARVALHO e LUDEMIR, 2007). Quando estes impulsos se propagam pelo axônio e chegam até as conexões sinápticas, provocam modificações nas membranas dessas terminações liberando moléculas de vários tipos com o nome genérico de neurotransmissores, que são transmitidos ao longo do terminal do axônio até o dendrito de um outro neurônio. Tais moléculas de neurotransmissores ao se unirem ao dendrito, provocam modificações em sua membrana, resultando em um potencial de ação ou dificultando seu aparecimento.

Em cada sinapse, o número de neurotransmissores que podem ser liberados para uma mesma frequência de impulsos do axônio representa a informação armazenada nesta sinapse. A cada vez que uma sinapse é ativada, aumenta-se o número de neurotransmissores liberados na conexão sináptica, reforçando a intensidade de conexão entre dois neurônios (BARRETO, 2002). Este processo é conhecido como facilitação, descrito na Lei de Hebb proposta pelo neuropsicólogo Donald Hebb. Essa lei afirma que a intensidade de uma conexão sináptica entre dois neurônios aumenta quando os dois neurônios estão excitados simultaneamente. Com

a possibilidade de variação dos potenciais sinápticos entre os neurônios, as redes neurais podem sofrer alterações estruturais no cérebro, levando à um novo conhecimento ou uma nova experiência.

Dado o entendimento biológico de um neurônio, segue-se para sua construção artificial.

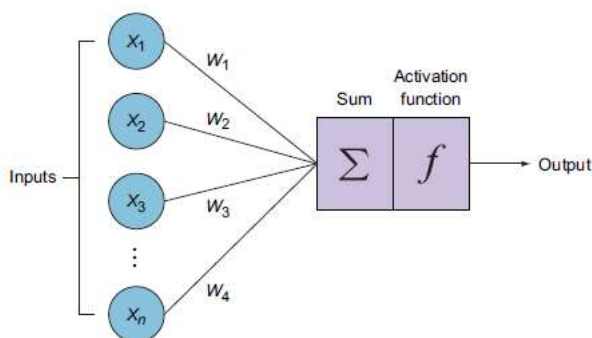
3.2 O NEURÔNIO ARTIFICIAL

As primeiras pesquisas para o desenvolvimento de um modelo computacional baseado no comportamento do cérebro humano datam em 1943, quando um psiquiatra e neuroanatomista Warren McCulloch e um matemático Walter Pitts publicaram um trabalho descrevendo sobre o cálculo lógico das redes neurais, unificando os estudos de neurofisiologia e lógica matemática (ANOCHI, 2010).

Os autores propuseram um modelo artificial matemático baseado nos neurônios biológicos, afirmando que uma rede neural com um número suficiente de neurônios, com conexões sinápticas ajustadas apropriadamente e operando de forma síncrona, realizaria a computação de qualquer função (FALQUETO, 2007).

Conforme ilustra a FIGURA 4 a seguir, o neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico, conforme descrito na sessão anterior. A grosso modo, pode-se associar os dendritos aos sinais entrada (*inputs*), o corpo celular à soma (*sum*) junto a sua função de ativação (*activation function*) e o axônio à saída (*output*).

FIGURA 4 - MODELO DO NEURÔNIO ARTIFICIAL PROPOSTO POR MCCULLOCH E PITTS



FONTE: Extraído de (ELGENDY, 2020).

Os dendritos são os sinais de entrada, cujas ligações com o corpo celular, associadamente chamado de neurônio artificial, são realizadas através de canais de comunicação que estão associados a um determinado peso sináptico. Os estímulos captados pelas entradas são processados pelo neurônio artificial e o limiar de ativação do neurônio biológico é substituído pela função de transferência (SILVA, 2003).

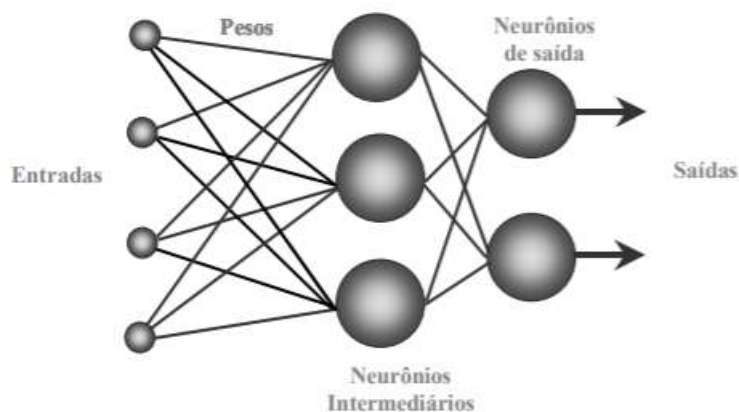
De acordo com (FERNEDA, 2006), o entusiasmo pela pesquisa neste campo cresceu durante os anos 50 e 60. Neste período, Frank Rosenblatt afirmou com o seu novo modelo, o Perceptron, que se fossem acrescentadas sinapses ajustáveis ao modelo desenvolvido por McCulloch e Pits, ele poderia ser treinado para classificar certos tipos de padrões (ANOCHI, 2010).

Com a definição do neurônio artificial, deu-se início às pesquisas sobre as redes neurais artificiais, que são descritas a seguir.

3.3 REDES NEURAIIS ARTIFICIAIS

As Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos, compostos por neurônios dispostos em uma topologia com várias camadas, que podem ser classificadas em: camadas de entrada, onde os padrões do exterior são expostos à rede; camada oculta, onde é feito o processamento; e a camada de saída, onde o resultado do processamento é apresentado. As diferentes possibilidades de conexões entre as camadas de neurônios podem gerar diversas topologias ou arquiteturas distintas (FURTADO, 2019). A FIGURA 5 a seguir representa um destes tipos de topologias, na qual a RNA possui apenas uma cada oculta.

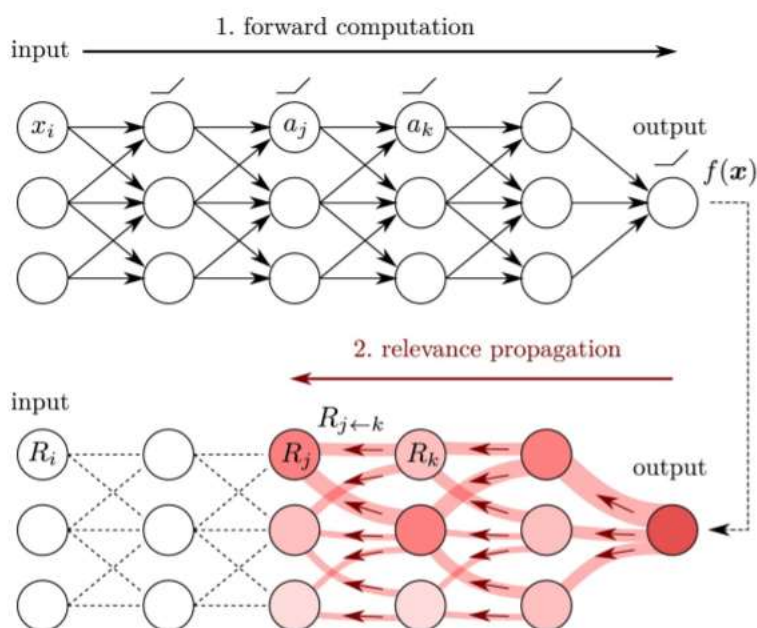
FIGURA 5 - REDE NEURAL ARTIFICIAL COM UMA CAMADA OCULTA



FONTE: Extraído de (FURTADO, 2019).

As redes neurais possuem dois modos de operação, conhecidos como *Feedforward* e *Propagation*, conforme ilustra a FIGURA 6 a seguir.

FIGURA 6 - MODOS DE OPERAÇÃO DAS REDES NEURAIS



FONTE: Extraído de (GHOLIZADEH e ZHOU, 2021).

O modo *Feedforward* consiste em apresentar um padrão para as unidades de entrada e propagar estes sinais pela rede, produzindo as saídas das unidades ocultas. Já o modo *Propagation* consiste em apresentar um padrão de entrada para as unidades de entrada, assim como um padrão de saída desejado para a camada de saída, e por fim alterar os parâmetros da rede para tornar a saída real mais próxima à de destino.

Para (GUALDA, 2008), o principal objetivo almejado com a estrutura de funcionamento de RNA com algoritmos de aprendizagem é a capacidade de generalização, ou seja, produzir saídas adequadas para entradas inexistentes durante o processo de aprendizagem.

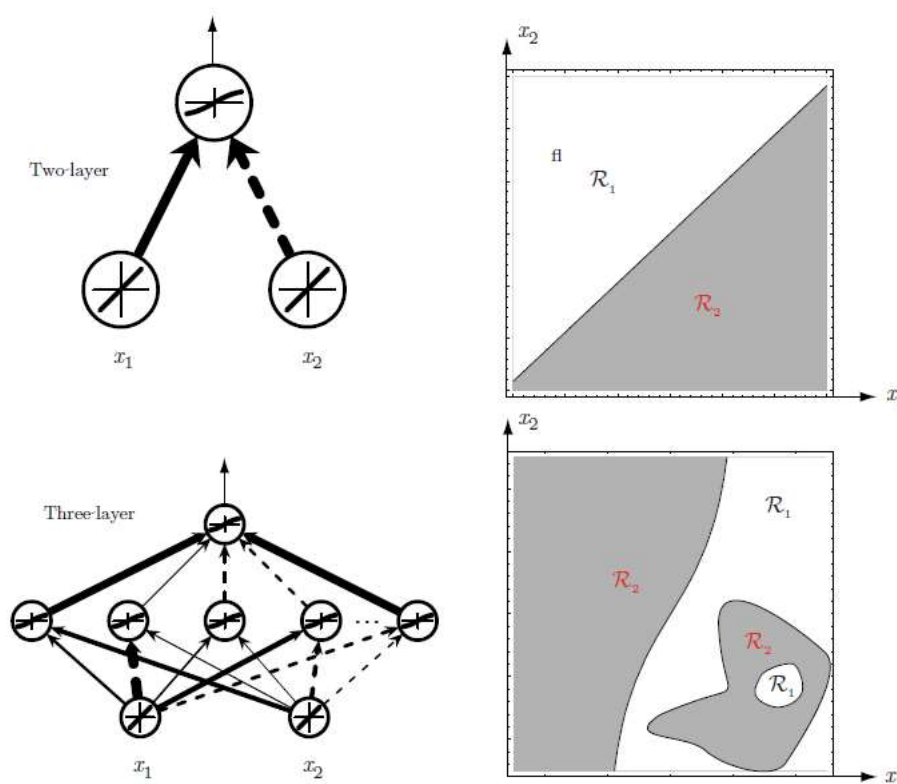
De acordo com (MINUSSI, 2008), denomina-se algoritmo de aprendizagem um conjunto de regras bem definidas para a solução de um problema específico. Isto é feito através de um processo iterativo de ajustes aplicados aos pesos das conexões sinápticas entre os neurônios que correspondem ao aprendido.

Neste contexto, existem diversos tipos de algoritmos específicos para determinadas topologias de RNA's, sendo os algoritmos de aprendizagem supervisionada e não supervisionada as abordagens mais comuns utilizadas. Para (OLIVEIRA, 2008), os algoritmos de aprendizagem não supervisionada recebem apenas os atributos de entrada e têm por objetivo construir um modelo que estabelece padrões em comum entre os atributos disponibilizados, classificando-os em grupos de características similares. Já os algoritmos de aprendizagem supervisionada recebem um conjunto de atributos com valores de entrada e saídas. Esse tipo de aprendizado tem como característica induzir o resultado do processamento.

Com isso, torna-se possível para as RNA's resolver problemas complexos, em especial problemas relacionados à classificação de imagens com duas classes (classificação binária) ou com três ou mais classes (classificação multiclasse). A razão desse feito se dá ao fato de que uma RNA, composta por uma ou mais camadas ocultas com um número adequado de neurônios e seus pesos sinápticos devidamente ajustados, possa aproximar praticamente qualquer função não linear contínua com restrições em um dado intervalo (HAYKIN, 2020).

De acordo com (DUDA, HART e STORK, 2001), a classe de soluções que podem ser obtidas de redes neurais com duas camadas não é suficiente em aplicações exigentes, pois há muitos problemas para os quais estes discriminantes lineares são insuficientes para induzir ao erro mínimo. Alternativamente, dado um número adequado de camadas ocultas com suas respectivas unidades, pode-se implementar limites de decisão arbitrários, conforme ilustra a FIGURA 7 a seguir.

FIGURA 7 - DISCRIMINANTES

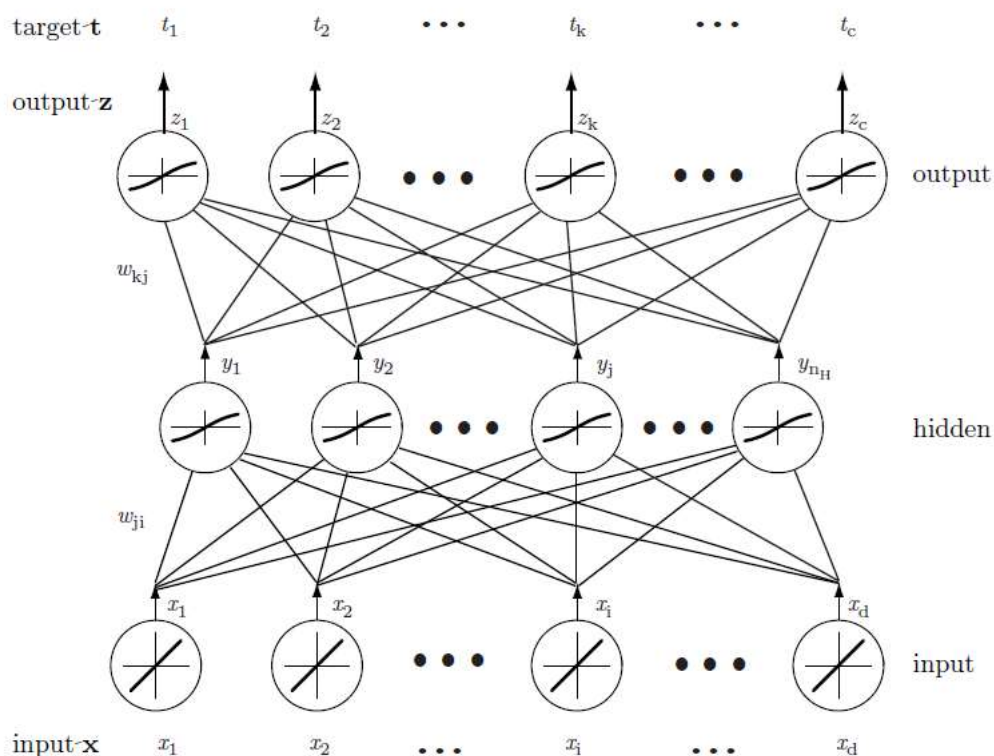


FONTE: Extraído de (DUDA, HART e STORK, 2001).

A dificuldade em aproximar qualquer função não linear consiste em encontrar os parâmetros corretos desta função. Um dos métodos mais populares para estimar os parâmetros das redes neurais multicamadas de forma supervisionada é baseado em gradiente descendente em erro – o algoritmo de retropropagação. Este método consiste em designar como os parâmetros das camadas ocultas devem ser aprendidos de forma a aproximar a saída obtida à saída desejada da rede. Através deste método, pode-se calcular um erro efetivo para cada unidade oculta e, portanto, derivar uma regra de aprendizado para os parâmetros das unidades das camadas ocultas (DUDA, HART e STORK, 2001).

A FIGURA 8 a seguir ilustra uma rede neural artificial de três camadas com d entradas representando o espaço de características, com nh unidades ocultas responsáveis por trazer a não linearidade à rede, e zc unidades de saída representando as classificações.

FIGURA 8 - REDE NEURAL ARTIFICIAL DE TRÊS CAMADAS



FONTE: Extraído de (DUDA, HART e STORK, 2001).

Conforme ilustra a FIGURA 8, durante o modo *Feedforward*, um padrão de entrada é apresentado à camada de entrada. Cada unidade de entrada emite seu componente x_i correspondente. Cada uma das n_H unidades ocultas computam a sua ativação líquida net_j como sendo o produto interno dos componentes x_i das entradas com seus respectivos pesos w_{ji} . As unidades ocultas emitem seus componentes y_j correspondentes, como sendo a aplicação da função de transferência não linear sob a sua ativação líquida (DUDA, HART e STORK, 2001).

Da mesma forma, cada uma das unidades de saída z_c computam net_k como sendo o produto interno dos componentes y_j com seus respectivos pesos w_{kj} . Os componentes z_k desta camada são obtidos através da aplicação de funções discriminantes para a classificação.

Durante o processo de treinamento da rede neural, as saídas reais são comparadas com as saídas desejadas através da função erro, e qualquer diferença entre as saídas é utilizada no treinamento dos parâmetros em toda a rede. Dessa forma, esta função é minimizada quando seus parâmetros são aprendidos de forma a aproximar a saída real da rede com a saída desejada. O processo de aprendizagem

dos parâmetros da rede neural se dá através do gradiente. O gradiente é definido como a velocidade com que os pesos e vieses de uma unidade se altera conforme o aprendizado da rede neural. O gradiente de uma unidade qualquer da rede neural pode ser encontrado através da Equação 1 adiante.

$$\Delta w_{mn} = -\eta \frac{\partial J}{\partial w_{mn}} \quad (1)$$

Onde:

- Δw_{mn} é o gradiente da unidade n da camada m ;
- η é a taxa de aprendizagem, onde diz respeito ao tamanho relativo da mudança nos pesos;
- ∂J é a derivada da função custo;
- A derivada parcial de ∂J em relação à ∂w_{mn} diz respeito à quanto uma mudança em ∂w_{mn} afeta ∂J .

Utilizando a rede neural de três camadas ilustrada na FIGURA 8 como demonstração, pode-se adaptar a Equação 2 à esta rede. Inicialmente, considerem-se os pesos w_{kj} das unidades j da camada k de saída. Como a função de erro J não é explicitamente dependente de w_{kj} , deve-se usar a regra da cadeia para diferenciação, conforme a Equação 2 a seguir.

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} = \delta_k \frac{\partial net_k}{\partial w_{kj}} \quad (2)$$

De acordo com (DUDA, HART e STORK, 2001), o último termo da Equação 2 pode ser simplificado conforme a Equação 3 a seguir.

$$\frac{\partial net_k}{\partial w_{kj}} = y_j \quad (3)$$

A sensibilidade da unidade k descreve como o erro geral se altera com a ativação desta unidade e é definida como:

$$\delta_k = -\frac{\partial J}{\partial net_k} \quad (4)$$

A Equação 4 pode ser simplificada como:

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k} \frac{\partial z_k}{\partial net_k} = (t_k - z_k) f'(net_k) \quad (5)$$

Com isso, tem-se a possibilidade de calcular o gradiente dos pesos da camada k , como definido na Equação 6 a seguir.

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (t_k - z_k) f'(net_k) y_j \quad (6)$$

O processo de aprendizagem do parâmetro peso da unidade j da camada k da rede ocorre em tendência a reduzir o erro, conforme a Equação 7 a seguir.

$$w'_{kj} = w_{kj} + \Delta w_{kj} \quad (7)$$

Onde:

- w'_{kj} é o novo valor do peso da unidade j da camada k ;
- w_{kj} é o valor do peso atual da unidade j da camada k ;
- Δw_{kj} é o gradiente da unidade j da camada k .

Da mesma forma, a regra de aprendizado para os pesos w_{ji} das unidades i da camada j pode ser encontrada utilizando a regra da cadeia para diferenciação, pois a função de erro J não é explicitamente dependente de w_{ji} . Tem-se:

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j} \frac{\partial y_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad (8)$$

$$\frac{\partial J}{\partial y_j} = \frac{\partial}{\partial y_j} \left[\frac{1}{2} \sum_{k=1}^c (t_k - z_k)^2 \right] \quad (9)$$

$$\frac{\partial J}{\partial y_j} = - \sum_{k=1}^c (t_k - z_k) \frac{\partial z_k}{\partial y_j} \quad (10)$$

$$\frac{\partial J}{\partial y_j} = - \sum_{k=1}^c (t_k - z_k) \frac{\partial z_k}{\partial net_k} \frac{\partial net_k}{\partial y_j} \quad (11)$$

$$\frac{\partial J}{\partial y_j} = - \sum_{k=1}^c (t_k - z_k) f'(net_k) w_{jk} \quad (12)$$

Na Equação 11 acima, teve-se que usar a regra da cadeia novamente, por não ser possível relacionar diretamente y_j com z_k . Na Equação 12, a soma final sobre as unidades de saída expressa como a saída da unidade oculta y_j afeta o erro em cada unidade de saída. Em analogia a Equação 4, a sensibilidade para uma unidade oculta é definida na Equação 13 a seguir.

$$\delta_j = f'(net_j) \sum_{k=1}^c w_{kj} \delta_k \quad (13)$$

A Equação 13 acima denota que a sensibilidade em uma unidade oculta é a soma das sensibilidades individuais nas unidades de saída ponderadas pelos pesos j da camada k multiplicados pela derivada da saída da função de ativação da unidade j . Portanto, a regra de aprendizado para os pesos das unidades j da camada oculta k é definida como:

$$\Delta w_{ji} = \eta x_j \delta_j = \eta x_j f'(net_j) \sum_{k=1}^c w_{kj} \delta_k \quad (14)$$

De forma análoga, o processo de aprendizagem do parâmetro peso da unidade i da camada j da rede ocorre em tendência a reduzir o erro, conforme definido na Equação 15 a seguir.

$$w'_{ji} = w_{ji} + \Delta w_{ji} \quad (15)$$

As Equações 6 e 14 descrevem o funcionamento do algoritmo de retropropagação de erros, pois a cada treinamento, o erro é propagado da camada de saída para as camadas anteriores, de forma a reajustar os pesos conforme a regra de aprendizagem. A essência deste algoritmo consiste na aplicação do conceito de gradiente descendente, onde através da regra da cadeia sob funções contínuas pode-se derivar a função de custo em relação a todos os parâmetros da rede neural.

Tendo conceituado a arquitetura de uma RNA, junto à um dos métodos existentes utilizado para o treinamento supervisionado, segue-se a apresentação das técnicas de aprendizado de máquina em arquiteturas com múltiplas camadas de processamento.

3.4 APRENDIZADO PROFUNDO

De acordo com (VALDATI, 2020), o aprendizado profundo, do inglês Deep Learning, refere-se a um conjunto de técnicas e arquiteturas de aprendizado de máquina, com a diferenciação de utilizar múltiplas camadas de processamento de informações não lineares. Cada camada contém unidades que transformam os dados de entrada em informações que a próxima camada pode usar para executar uma determinada tarefa preditiva.

À medida que novas camadas ocultas são adicionadas à arquitetura da rede neural, maior é a sua capacidade de aprendizagem. Este feito acontece, pois, redes neurais muito profundas são capazes de representar funções complexas, permitindo o aprendizado da rede em diversos níveis de abstração (ELGENDY, 2020).

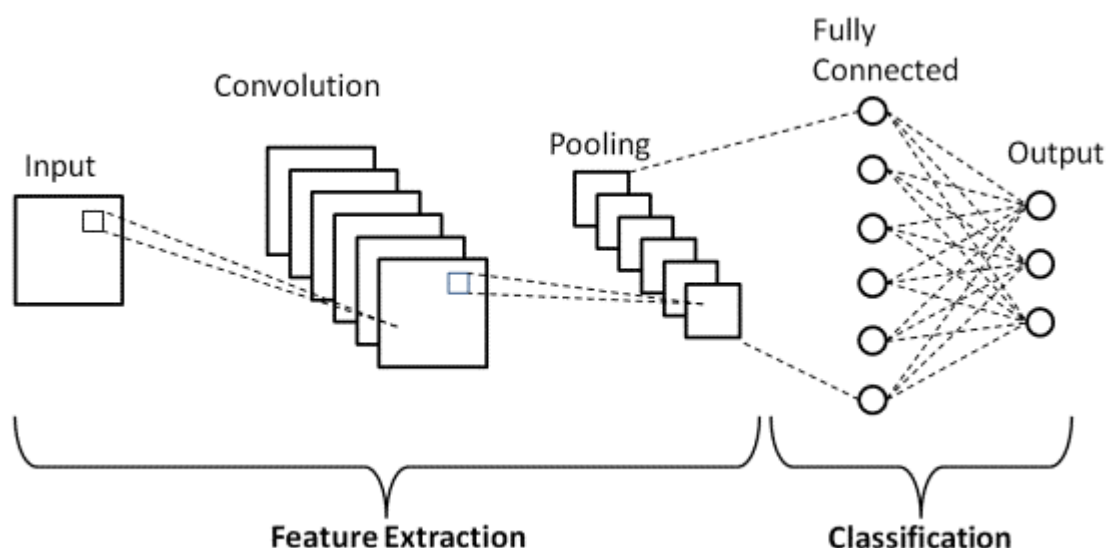
Para (BESSE, 2017), o uso das técnicas de aprendizado profundo permitiu um progresso significativo nas áreas de processamento de imagem. Na indústria, estas técnicas são utilizadas para resolver tarefas práticas em uma variedade de campos

como visão computacional, processamento de linguagem natural e reconhecimento automático de fala. Segundo (DU, RAO, *et al.*, 2019), existem diversas arquiteturas de redes neurais profundas popularmente utilizadas. Algumas delas são descritas a seguir.

3.5 REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais são arquiteturas de redes neurais artificiais muito utilizadas, especialmente para aplicações que utilizam dados de alto nível, como imagens e vídeos. Em contraste às redes neurais convencionais, as CNN's aplicam filtros de convolução aos dados brutos de entrada para extrair e aprender recursos de alto nível. O termo convolução denota a função matemática de convolução que é um tipo especial de operação linear em que duas funções são multiplicadas para produzir uma terceira função. Esta, por sua vez, expressa como a forma de uma função é modificada pela outra. Existem três tipos de camadas que compõe a CNN, conhecidas como camadas convolucionais, camadas de *pooling* e camadas totalmente conectadas, como ilustra a FIGURA 9 a seguir.

FIGURA 9 - CAMADAS DA ARQUITETURA CNN



FONTE: Extraído de (DU, RAO, *et al.*, 2019).

Nas camadas convolucionais, a operação matemática de convolução é realizada entre a imagem de entrada e um filtro de um determinado tamanho. Este

filtro é formado por pesos inicializados aleatoriamente, atualizando-se a cada nova entrada durante o processo. Ao aplicar o filtro sobre a imagem de entrada com *stride* (passos) de tamanho n , o produto escalar é obtido entre o filtro e as coordenadas espaciais da imagem de entrada em relação ao tamanho e posição do passo do filtro, denominado como mapa de recursos. As camadas convolucionais tipicamente aplicam uma função de ativação *ReLU* na saída para introduzir a não-linearidade ao modelo (DUDA, HART e STORK, 2001).

As camadas de *pooling* são responsáveis por diminuir o tamanho do mapa de recursos convoluído para reduzir os custos computacionais. Portanto, assim como na convolução, é escolhido um filtro a ser aplicado por toda a saída da camada anterior. Esse filtro é responsável por resumir toda a informação em um único valor através de métodos conhecidos como sumarização (DUDA, HART e STORK, 2001).

Por fim, as camadas totalmente conectadas recebem as saídas das camadas anteriores e suas saídas são N neurônios. Uma função de ativação *Softmax* é utilizada para gerar um valor entre 0 e 1 para cada neurônio. Esses valores representam as probabilidades de cada classe na tarefa de classificação (DUDA, HART e STORK, 2001).

Após a conceituação das características de uma CNN, deu-se início a abordagem das redes neurais residuais.

3.6 REDES RESIDUAIS

De acordo com (ZHANG, LIPTON, *et al.*, 2021), a intuição por trás da utilização de uma rede neural profunda é que as múltiplas camadas ocultas proporcionam uma vantagem conveniente em aprender a resolver problemas complexos de reconhecimento de padrões. No entanto, o acréscimo de múltiplas camadas ocultas à arquitetura da rede neural faz com que seu desempenho sature e eventualmente comece a degradar. Este fenômeno é conhecido como problema de Vanishing Gradient (gradiente de fuga).

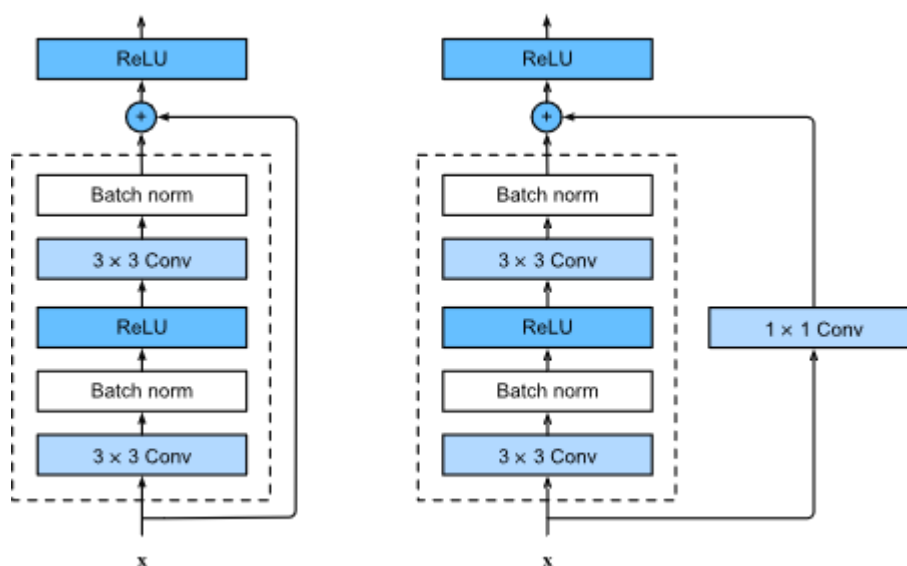
Este problema dificulta o aprendizado e o ajuste dos parâmetros das camadas anteriores da rede, pois o processo de determinação do gradiente e envio de volta para a próxima camada ocultada ocorre até que a camada de entrada seja alcançada. Com o gradiente tornando-se menor à medida que atinge as camadas próximas à extremidade de entrada do modelo, os pesos e vieses destas camadas serão

atualizados muito vagarosamente ou permanecerão os mesmos. Com isso, o treinamento do modelo não convergirá e a precisão começará degradar lentamente (DUDA, HART e STORK, 2001).

Para (DUDA, HART e STORK, 2001), existem vários métodos que ajudam a superar do gradiente de fuga, como Hierarquia de vários níveis, memória de longo prazo e redes neurais residuais. As redes neurais residuais não foram desenvolvidas para resolver o problema do gradiente de fuga, mas possuem uma característica que as diferenciam das outras redes neurais – as conexões residuais.

Através destas conexões, uma técnica chamada de “pular conexões” é utilizada para transmitir informações das camadas anteriores do modelo para camadas posteriores, conforme ilustra a FIGURA 10 a seguir. A possibilidade de saltar conexões gera o benefício de pular as camadas da rede que estão prejudicando o desempenho do modelo.

FIGURA 10 - TIPOS DE BLOCOS RESIDUAIS



FONTE: Extraído de (ZHANG, LIPTON, *et al.*, 2021).

A FIGURA 10 exibe os tipos de conexões residuais. O esquema esquerdo retrata o funcionamento do bloco Identidade, onde o mesmo transpõe diretamente o resíduo à saída, enquanto o esquema direito retrata o bloco Convolucional responsável por realizar uma convolução seguida de Normalização em lote no resíduo antes de adicioná-lo à saída.

Concluída a revisão teórica das redes neurais artificiais, segue-se para a teoria estatística utilizada neste trabalho.

3.7 CAPABILIDADE DE PROCESSO

De acordo com (KANE, 1986), a análise da Capabilidade de Processo é um procedimento para avaliar a condição de um processo em atender às especificações de determinada característica. Afirma ainda que esta análise visa determinar o comportamento do processo, de modo que as tolerâncias de cada projeto possam ser satisfeitas com os recursos disponíveis, ou então, na especificação de características de novos equipamentos produtivos ou na comparação de processos com diferentes equipamentos.

Em termos matemáticos, a análise da Capabilidade de Processo é desenvolvida mediante a aplicação de técnicas estatísticas que permitem determinar o modelo probabilístico da saída do processo, sua dispersão e localização, tendo como referência os limites de tolerância especificados em projeto. Com base neste modelo, é possível realizar inferências que descrevam adequadamente o seu comportamento (KANE, 1986). Para (POLHEMUS, 2017), a capacidade de um processo pode ser especificada por meio de índices que quantificam o nível de desempenho do processo em atender os requisitos do projeto.

De acordo com (BURY, 1999), a distribuição Beta é uma candidata natural para modelar razões de engenharia, por exemplo, medidas de eficiência, que variam ao longo da faixa de valores definidos entre 0 e 1 (ou 0% e 100%). Além disso, a flexibilidade da forma desta distribuição a torna eficiente como modelo de medições em geral.

Neste trabalho, utilizou-se a distribuição Beta para modelar a incerteza sobre a ocorrência de classificações com probabilidades inferiores ao limite de especificação definido pelos requisitos do projeto. Para quantificar o nível de desempenho de um processo, empregou-se o Índice de Capacidade Potencial (CP) determinado pelo quociente entre a dimensão da faixa de tolerância especificada e a variação natural do processo, afim de atender a demanda do controle estatístico de processo na indústria.

3.8 ESTATÍSTICA KAPPA

De acordo com (CONGALTON e GREEN, 2019), a análise Kappa é uma técnica multivariada discreta utilizada na avaliação da acurácia para determinar estatisticamente se uma matriz de confusão é significativamente diferente de outra. O resultado desta avaliação é uma estatística de concordância ou precisão. Esta medida de concordância é fundamentada na diferença entre a concordância real na matriz de confusão (concordância entre os dados da classificação e os dados de referência) e a concordância aleatória representada pelos totais marginais.

A definição do índice Kappa é descrita na Equação 22 a seguir.

$$Kappa = \frac{n \sum_{i=1}^k n_{ii} - \sum_{i=1}^k n_{i+} n_{+i}}{n^2 - \sum_{i=1}^k n_{i+} n_{+i}} \quad (22)$$

Onde:

- k é o número de linhas na matriz de confusão;
- n é o número total de observações;
- n_{ij} é o valor observado na linha e coluna i ;
- n_{i+} são os totais marginais da linha i ;
- n_{+i} são os totais marginais da coluna i .

A classificação do nível de precisão do índice Kappa é descrito por (LANDIS e KOCH, 2020), onde caracterizam as faixas possíveis para a estatística Kappa em três segmentos: 1) Forte concordância para um valor maior que 0,80; 2) Concordância moderada para um valor entre 0,40 e 0,80 e 3) Baixa concordância para um valor abaixo de 0,40.

Para (CONGALTON e GREEN, 2019), o fato de a estatística Kappa ser normalmente distribuída possibilita calcular os intervalos de confiança em torno do valor da estatística Kappa através da variância aproximada da amostra, assim como realizar o teste de significância para esta estatística, como descrito na Equação 23 e 28, respectivamente.

$$var(K) = \frac{1}{n} \left\{ \frac{\theta_1(1-\theta_1)}{(1-\theta_2)^2} - \frac{2(1-\theta_1)(2\theta_1\theta_2 - \theta_3)}{(1-\theta_2)^3} + \frac{(1-\theta_1)^2(\theta_4 - 4\theta_2^2)}{(1-\theta_2)^4} \right\} \quad (23)$$

Onde:

$$\theta_1 = \frac{1}{n} \sum_{i=1}^k n_{ii} \quad (24)$$

$$\theta_2 = \frac{1}{n^2} \sum_{i=1}^k n_{i+} n_{+i} \quad (25)$$

$$\theta_3 = \frac{1}{n^2} \sum_{i=1}^k n_{ii} (n_{i+} + n_{+i}) \quad (26)$$

$$\theta_4 = \frac{1}{n^3} \sum_{i=1}^k \sum_{j=1}^k n_{ij} (n_{j+} + n_{+i})^2 \quad (27)$$

O teste de significância Z determina se a concordância entre os dados da classificação e os dados de referência é significativamente melhor do que uma classificação aleatória.

$$Z = \frac{K}{\sqrt{var(K)}} \quad (28)$$

Dado a hipótese nula $H_0: K = 0$ em que a concordância entre os dados da classificação e os dados de referência é igual a uma classificação aleatória. A hipótese alternativa $H_1: K \neq 0$ em que a concordância entre os dados da classificação e os dados de referência é significativamente melhor do que uma classificação aleatória. Tem-se que H_0 é rejeitado se $Z \geq Z_\alpha$, onde Z_α é o nível de confiança do teste Z .

Para avaliar se dois valores Kappa independentes e, portanto, duas matrizes de confusão, são significativamente diferentes, utiliza-se a Equação 29 a seguir.

$$Z = \frac{|K_1 - K_2|}{\sqrt{\text{var}(K_1) + \text{var}(K_2)}} \quad (29)$$

Dado a hipótese nula $H_0: (K_1 - K_2) = 0$ em que os dois avaliadores são iguais, e a alternativa $H_1: (K_1 - K_2) \neq 0$ em que os dois avaliadores são significativamente diferentes, H_0 é rejeitado se $Z \geq Z_\alpha$, onde Z_α é o nível de confiança do teste Z .

4 MÉTODO PROPOSTO

A elaboração deste trabalho consiste em 4 etapas, conforme listado a seguir.

1ª Etapa: Pesquisa dos sistemas de visão computacional disponíveis na indústria e análise de suas funcionalidades;

2ª Etapa: Concepção da problematização e levantamento dos requisitos estabelecidos a partir da necessidade de aquisição de um sistema de visão computacional que disponha de simplicidade e desempenho no processo de construção do modelo de aprendizagem de máquina, juntamente à um controle estatístico da capacidade do processo.

3ª Etapa: Proposição de um sistema de visão computacional baseado em técnicas de inteligência artificial que contemple a solução da problematização definida;

4ª Etapa: Definição dos métodos de comparação e avaliação do *Mosaic Office* por meio de técnicas estatísticas como a estatística Kappa, Z e índices de capacidade;

4.1 ETAPA 1 – PESQUISA E ANÁLISE DE FUNCIONALIDADES

A implementação deste projeto se iniciou com uma pesquisa exploratória em sistemas de visão computacional disponíveis na indústria. Por meio desta e por pesquisas relacionadas à própria fabricante dos sistemas de visão em questão, pode-se conhecer suas funcionalidades conforme ilustrado na TABELA 1 do Capítulo 2.

4.2 ETAPA 2 – PROBLEMATIZAÇÃO

Após a análise das funcionalidades dos sistemas de visão computacional apresentados no Capítulo 2, identificou-se que os *softwares* classificadores de imagens implementados por estes sistemas não possuem a funcionalidade de capacidade do processo. Essa funcionalidade quantifica o nível de desempenho do processo em atender às especificações definidas pelo projeto através do índice de capacidade. Além disso, percebeu-se a oportunidade de criação de um sistema com interfaces gráficas que facilitem o processo de alimentação do modelo de aprendizagem de máquina, assim como implementar um algoritmo de treinamento rápido e eficiente.

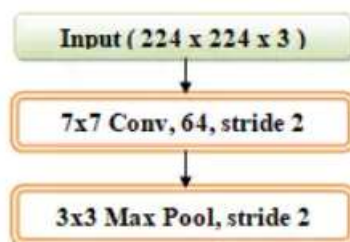
4.3 ETAPA 3 – SOLUÇÃO PROPOSTA

O sistema de visão computacional implementado neste projeto (*Mosaic Office*) teve por objetivo utilizar técnicas de inteligência artificial. Utilizou-se o aprendizado profundo para construir um modelo de aprendizagem de máquina capaz de realizar a tarefa de classificação de imagem multiclasse. Desenvolveu-se um algoritmo de treinamento denominado Treinamento Contínuo com o objetivo de otimizar o desempenho do modelo treinado. Para avaliar a capacidade do processo, o índice de capacidade potencial CP foi aplicado.

4.3.1 ARQUITETURA UTILIZADA

O sistema *Mosaic Office* utiliza a rede residual ResNetV2 50, um modelo de rede neural profunda que utiliza a arquitetura CNN, comumente empregada em visão computacional. Como ilustra a FIGURA 11, inicialmente a arquitetura executa uma convolução utilizando 64 filtros com o tamanho de 7x7. O resultado desta operação é resumido através do método de sumarização *Max Pooling* (agrupamento máximo) com somente um filtro de tamanho 3x3, no qual apenas o maior valor do mapa de recursos é passado para a saída. Nas duas operações, o tamanho de passo utilizado é 2 (SACHAN).

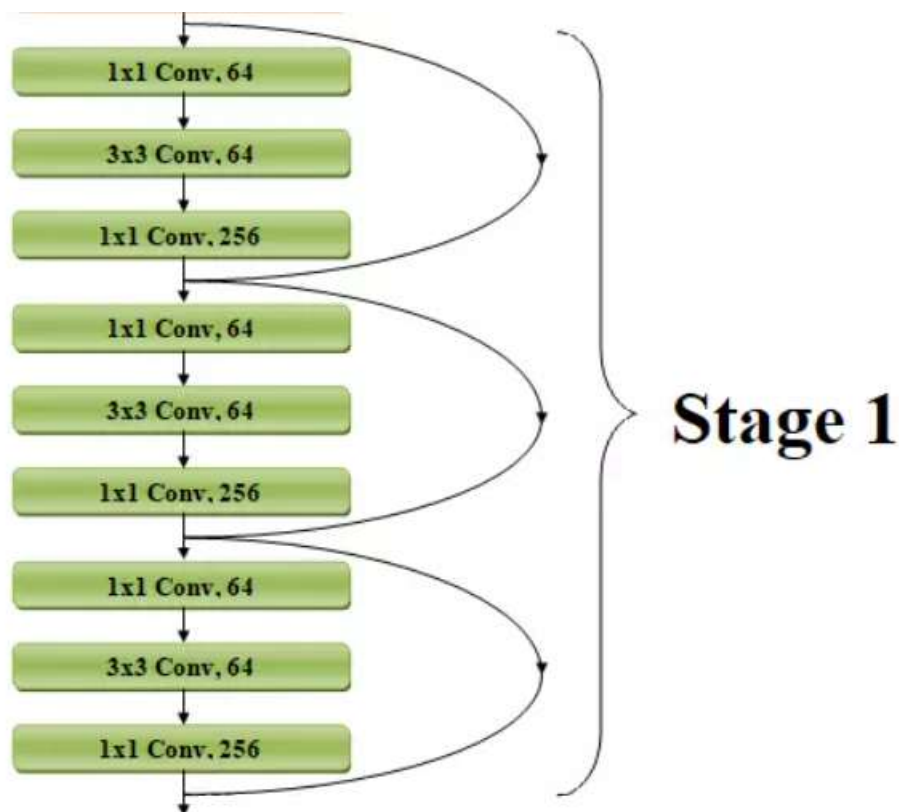
FIGURA 11 - OPERAÇÕES INICIAIS DE CONVOLUÇÃO E SUMARIZAÇÃO



FONTE: Extraído de (SACHAN).

Logo após, inicia-se o estágio 1 da rede. A primeira, segunda e terceira camada possuem, respectivamente, 64 filtros com o tamanho 1x1, 64 filtros com o tamanho 3x3 e 256 filtros com o tamanho 1x1. Todas essas camadas são repetidas três vezes, totalizando 9 camadas, conforme ilustra a FIGURA 12 a seguir (SACHAN).

FIGURA 12 - ESTÁGIO 1

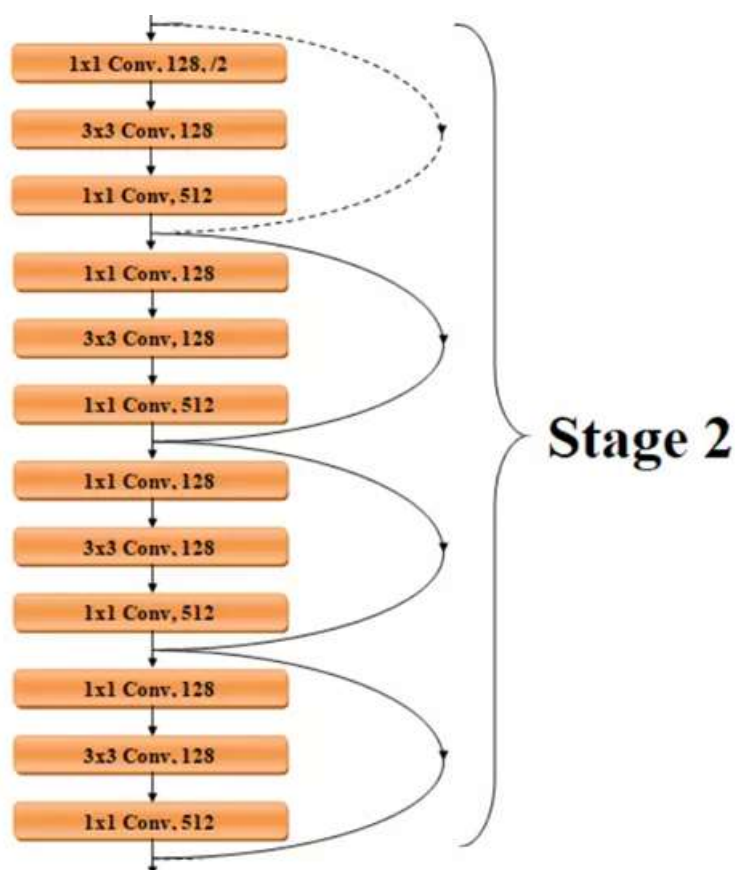


FONTE: Extraído de (SACHAN).

No estágio 2 da rede, a primeira, segunda e terceira camada possuem, respectivamente, 128 filtros com o tamanho 1x1, 128 filtros com o tamanho 3x3 e 512 filtros com o tamanho 1x1.

Com exceção da primeira camada que realiza a operação de convolução com passo 2 no bloco residual, todas essas camadas são repetidas quatro vezes, totalizando 12 camadas, conforme ilustra a FIGURA 13 a seguir.

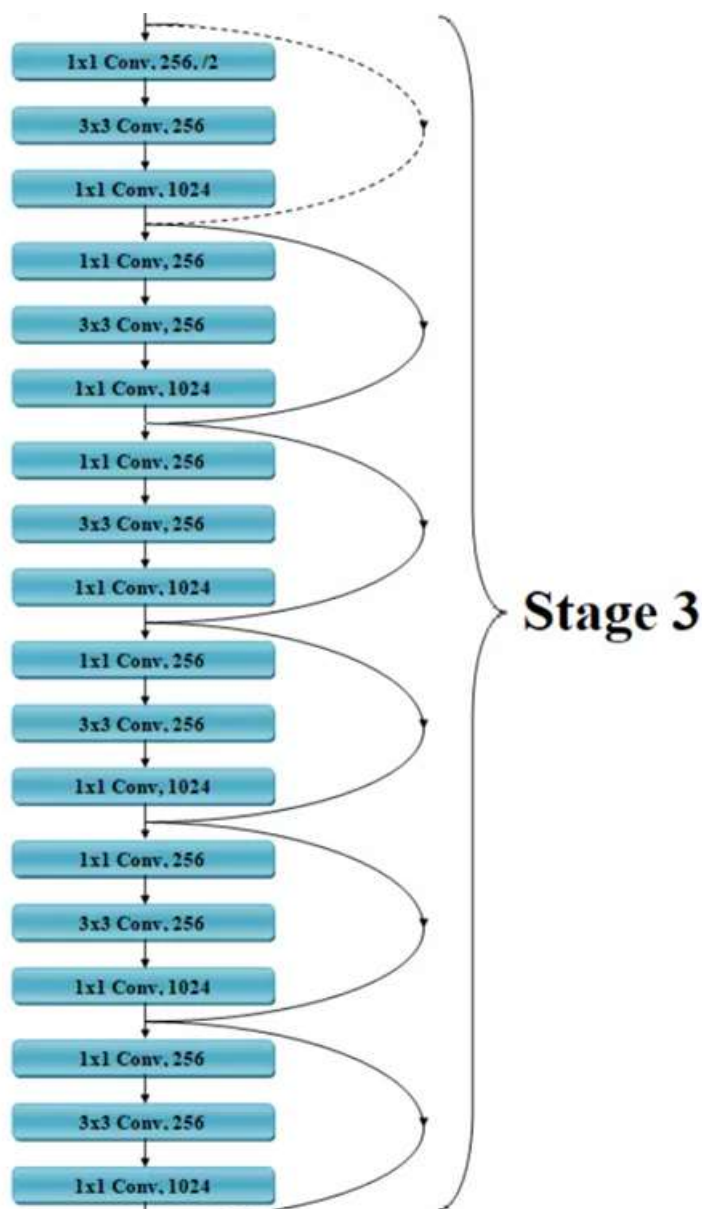
FIGURA 13 - ESTÁGIO 2



FONTE: Extraído de (SACHAN).

No estágio 3 seguinte, a primeira, segunda e terceira camada possuem, respectivamente, 256 filtros com o tamanho 1x1, 256 filtros com o tamanho 3x3 e 1024 filtros com o tamanho 1x1. Neste estágio também ocorre da primeira camada realizar a operação de convolução com passo 2 no bloco residual. Com exceção desta, todas essas camadas são repetidas seis vezes, totalizando 18 camadas, conforme ilustra a FIGURA 14 a seguir.

FIGURA 14 - ESTÁGIO 3

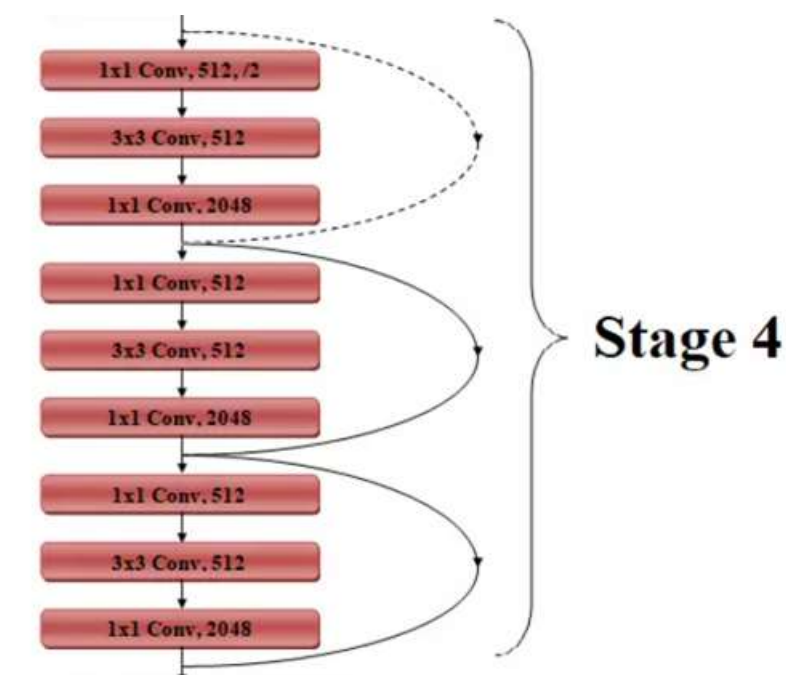


FONTE: Extraído de (SACHAN).

Em seu último estágio, o estágio 4, assim como nos estágios anteriores, as camadas estão agrupadas em trios. Esta característica está presente em todos os estágios devido à forma como as conexões residuais estão conectadas. A primeira, segunda e terceira camada possuem, respectivamente, 512 filtros com o tamanho 1x1, 512 filtros com o tamanho 3x3 e 2048 filtros com o tamanho 1x1.

Da mesma forma que no estágio 2 e 3, a primeira camada deste último estágio realiza a operação de convolução com passo 2 no bloco residual. Com exceção desta, todas essas camadas são repetidas três vezes, totalizando 9 camadas, conforme ilustra a FIGURA 15 a seguir.

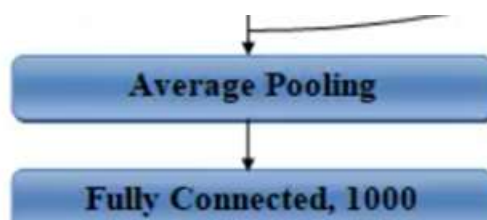
FIGURA 15 - ESTÁGIO 4



FONTE: Extraído de (SACHAN).

Por fim, o resultado da operação anterior é resumido através do método de sumarização *Average Pooling* (agrupamento médio), no qual o valor médio do mapa de recursos é passado para a camada final totalmente conectada contendo 1000 neurônios, como ilustra a FIGURA 16 a seguir.

FIGURA 16 - OPERAÇÃO FINAL DE SUMARIZAÇÃO E CAMADAS TOTALMENTE CONECTADAS



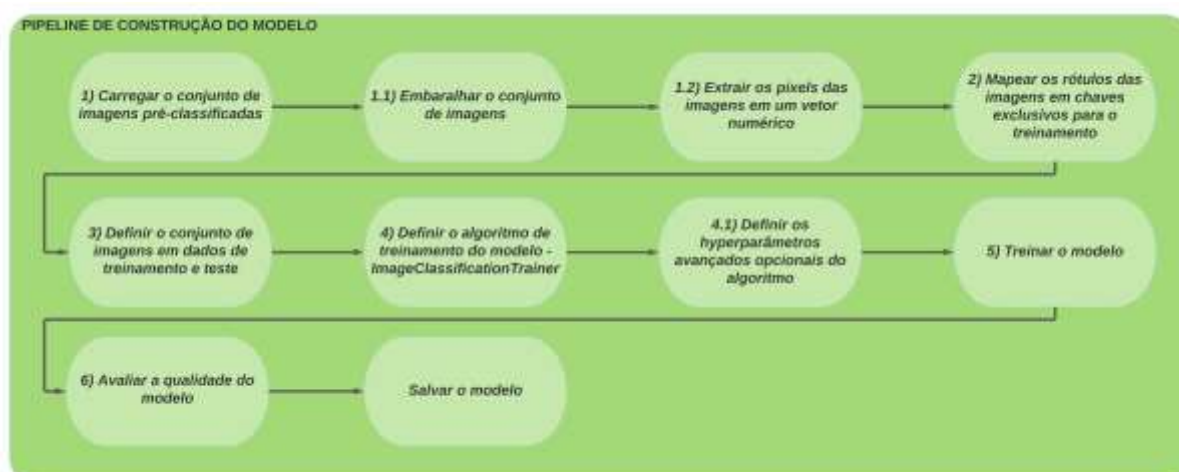
FONTE: Extraído de (SACHAN).

4.3.2 ALGORITMO DE TREINAMENTO

Para treinar esta arquitetura, o sistema de visão *Mosaic Office* utiliza a linguagem C# juntamente com a biblioteca de aprendizado de máquina de código aberto e multiplataforma ML.NET. Esta biblioteca treina nativamente um modelo da API ML.NET, contendo transformações a serem executadas em seus dados de entrada para chegar ao resultado previsto.

Cada operação de aprendizado de máquina utiliza um contexto, o qual possui catálogos responsáveis por carregar e salvar dados, realizar transformações, definir treinadores e componentes de operação de modelo. Para treinar um modelo de aprendizagem de máquina, é necessário um pipeline de treinamento do modelo, decorrido em 6 estágios, conforme ilustra a FIGURA 17 a seguir.

FIGURA 17 - PIPELINE DE TREINAMENTO DO MODELO



FONTE: O autor (2022).

No estágio 1) as imagens pré-classificadas são carregadas na memória, 1.1) embaralhadas para que o conjunto de dados de imagens seja melhor balanceado antes de ser dividido entre os conjuntos de dados de treinamento/teste, 1.2) os *pixels* são extraídos em um vetor numérico. A execução dos três primeiros estágios é necessária pois os dados das imagens precisam ser processados antes de serem utilizados para encontrar os parâmetros de seu modelo.

Em seguida, no estágio 2) é realizado um mapeamento valor-chave sobre os dados de treinamento. Onde é possível mapear rótulos de *string* em valores inteiros exclusivos para o treinamento.

No estágio 3) o conjunto de dados de imagens é dividido em dois: um para treinamento e outro para testar/validar a qualidade do modelo.

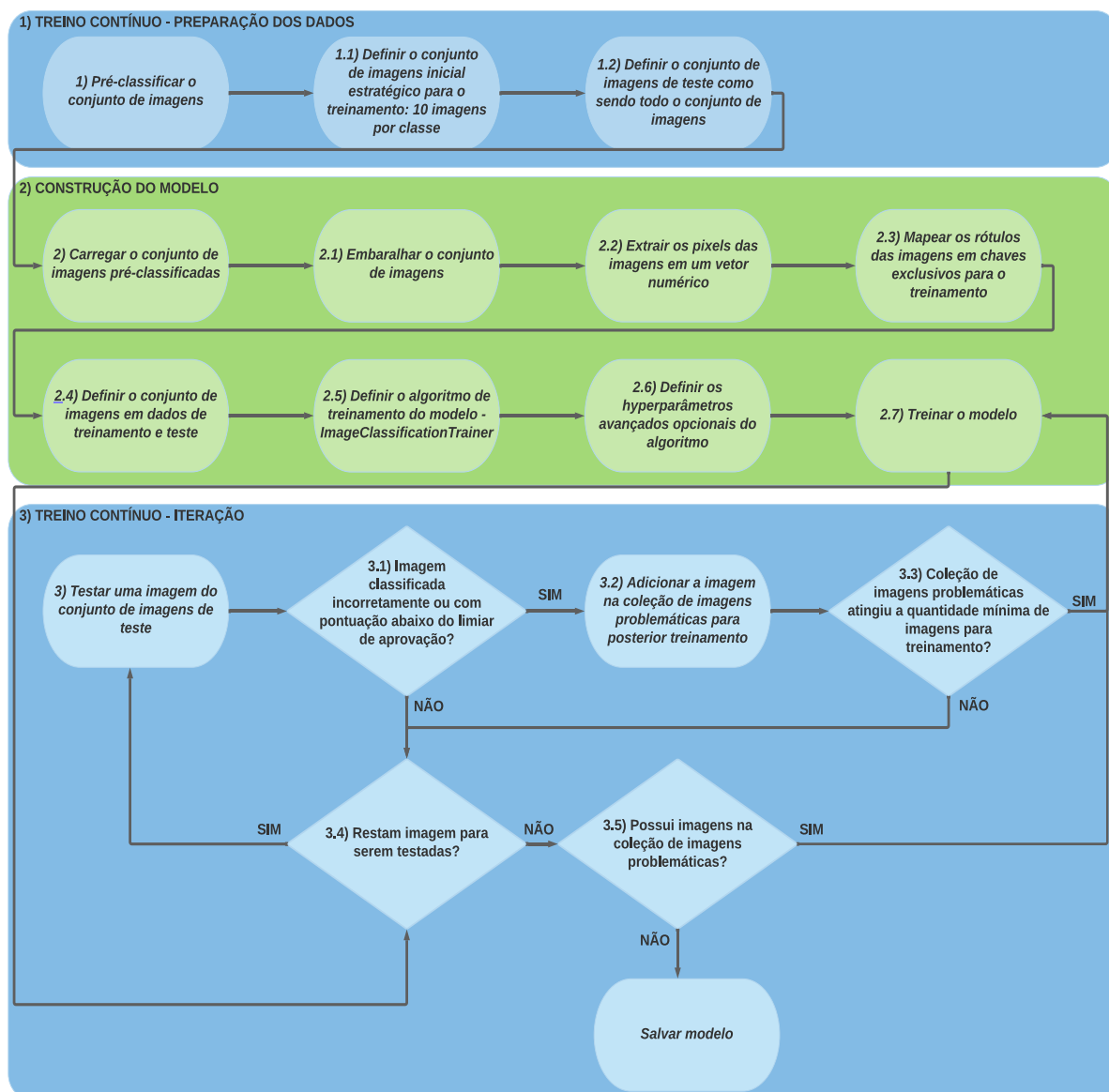
No estágio 4) é definido o algoritmo de treinamento do modelo de aprendizagem de máquina *ImageClassificationTrainer* (MICROSOFT), que é baseado em uma arquitetura padrão DNN. Este algoritmo possui hiperparâmetros avançados opcionais, os quais são definidos no estágio 4.1, tais como a quantidade de épocas, taxa de aprendizagem, dentre outros hiperparâmetros típicos de treinamento.

No estágio 5) após os objetos terem sido definidos, os dados de treinamento de entrada podem ser usados para estimar os parâmetros do modelo. O treinamento do modelo consiste em um processo de execução do algoritmo escolhido em dados de treinamento.

No estágio 6) a fim de avaliar a precisão do modelo, são adicionados dados de teste ao contexto de aprendizagem de máquina e após a realização da previsão, os valores previstos são comparados com o conjunto de dados de teste, retornando as métricas de desempenho do modelo.

Para otimizar o treinamento do modelo, neste trabalho desenvolveu-se um algoritmo denominado Treinamento Contínuo na tentativa de otimizar o desempenho do modelo através de ciclos estratégicos de treino e teste, como ilustra a FIGURA 18.

FIGURA 18 - ALGORITMO TREINO CONTÍNUO



FONTE: O autor (2022).

O algoritmo de Treinamento Contínuo pressupõe de que a retroalimentação automática das imagens problemáticas sobre os dados de treinamento irá aumentar a precisão do modelo treinado. Os benefícios deste algoritmo ficam ainda mais evidentes quando se utilizam imagens com as mesmas características. Nessas condições não é necessário ter um grande conjunto de dados de treinamento para se construir um modelo eficiente, visto que poucas imagens podem ter a capacidade de representar toda a população de sua classe.

O algoritmo de Treinamento Contínuo ocorre em 3 rotinas. Na rotina 1) ocorre a preparação dos dados para o treinamento, desde a pré-classificação das imagens, definição do conjunto de imagens inicial, até a definição do conjunto de imagens de

teste. Na rotina 2) ocorre a construção do modelo de aprendizado de máquina, ou seja, a execução do algoritmo escolhido sobre os dados de treinamento definidos na rotina anterior. Por fim, na rotina 3) ocorre a iteração do treinamento, onde o conjunto de imagens de teste são testados.

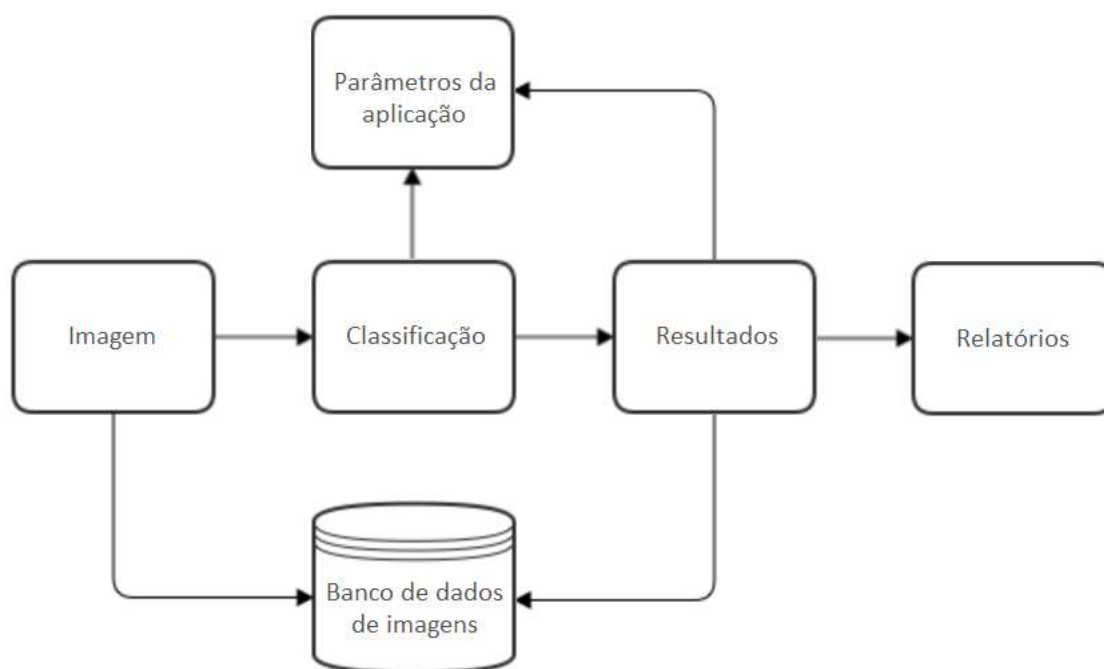
Quando uma imagem não é classificada corretamente, esta imagem é adicionada à uma coleção de imagens problemáticas. Esta coleção de imagens serve para armazenar as imagens que não foram devidamente classificadas, de acordo com sua classe e limite de pontuação mínima estabelecida. O ciclo de teste se repete até que haja uma quantidade mínima de imagens, definida empiricamente com o valor de 20 imagens. Quando isso ocorre, adiciona-se as imagens problemáticas ao conjunto de imagens de treinamento e inicia-se um novo treinamento do modelo. A última rotina do Treinamento Contínuo ocorre até que não haja mais imagens a serem testadas e não existam imagens na coleção de imagens problemáticas.

4.3.3 WORKSPACE

O *Mosaic Office* utiliza o conceito de *workspace* (área de trabalho) para cada aplicação. Dessa forma, o sistema poderá ter várias aplicações, cada uma com seu *workspace* operando de forma individual, com seus respectivos conjuntos de ferramentas e bancos de dados de imagens necessários para resolver a aplicação.

Um *workspace* é o contêiner básico que contém todas as informações relevantes sobre uma aplicação de classificação de imagens multiclasse específico. Mais concretamente, isso inclui todos os componentes exibidos na FIGURA 19 a seguir.

FIGURA 19 - WORKSPACE



FONTE: O autor (2022).

Uma aplicação de classificação é definida em torno de um conjunto de imagens, tanto na parametrização e treinamento da rede neural, como também no ambiente de operação.

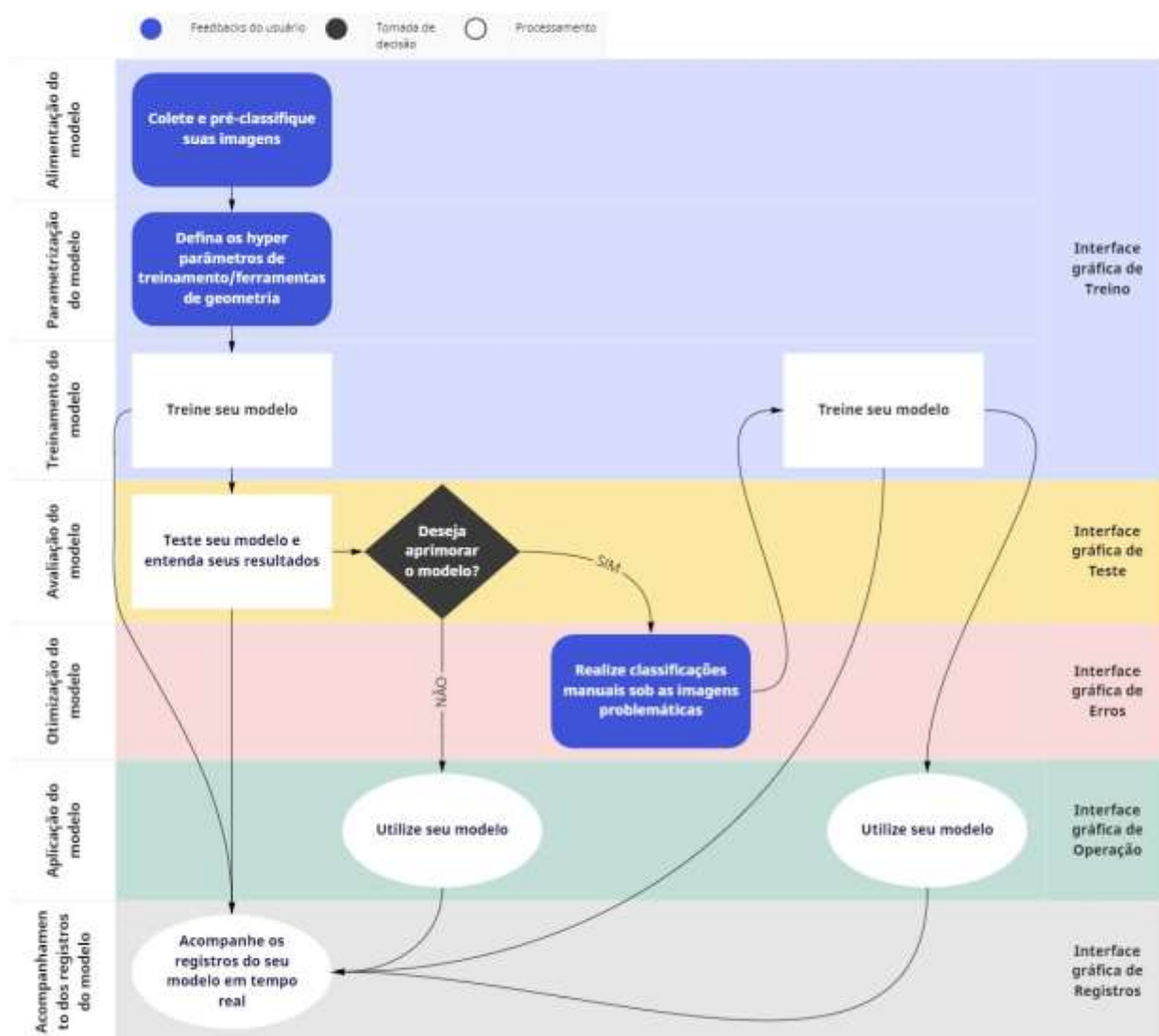
Pelo fato de as técnicas de aprendizado profundo serem baseadas em aprendizado, se faz necessário um banco de dados de imagens para a aplicação. Estas imagens estão dispostas localmente nos diretórios de aprendizagem de treino e teste, assim como nos diretórios de resultados.

Durante a tarefa de classificação, as imagens são processadas pelo sistema com base nas parametrizações da aplicação e os resultados da tarefa são encaminhados para o diretório de resultados. Se o resultado da conclusão da tarefa estiver dentro dos requisitos da aplicação, uma marcação em forma de sobreposição com os resultados da classificação será adicionada na imagem. Caso o resultado esteja fora dos requisitos da aplicação, a imagem será encaminhada para o diretório de imagens com erros da aplicação. Os resultados de teste de uma aplicação podem ser exportados através de um relatório estatístico de capacidade, situados em um diretório específico para essa finalidade.

4.3.4 INTERFACES GRÁFICAS

O *Mosaic Office* fornece acesso às técnicas de aprendizado profundo por meio das interfaces gráficas que simplificam o processo de alimentação do modelo de aprendizagem de máquina. A FIGURA 20 a seguir ilustra o fluxo de utilização das interfaces gráficas.

FIGURA 20 – FLUXO DE UTILIZAÇÃO DAS INTERFACES GRÁFICAS



FONTE: O autor (2022).

O *Mosaic Office* implementa cinco interfaces gráficas: 1) Interface gráfica de Treino; 2) Interface gráfica de Teste; 3) Interface gráfica de Erros; 4) Interface gráfica

de Operação e 5) Interface gráfica de Registros. A seguir a descrição dessas interfaces:

1) pode-se coletar e pré-classificar as imagens da aplicação, assim como também definir os parâmetros de identificação da aplicação, os hiperparâmetros de treinamento e as ferramentas de geometria. Para estimar os parâmetros do modelo, pode-se iniciar o treinamento deste com alguma das implementações de algoritmos de treinamento: Treinamento Contínuo e Treinamento Clássico.

2) oferece a possibilidade de avaliar o modelo através da rotina de testes. Com isso, pode-se compreender os pontos fracos e fortes do modelo através dos resultados das tarefas de classificação. Além disso, a rotina de testes encaminha as imagens problemáticas para o diretório de erros da aplicação.

3) exibe as imagens do diretório de erros da aplicação. Com isto, tem-se a possibilidade de realizar classificações manuais sob estas imagens, encaminhando as mesmas para o diretório de treinamento da classe escolhida. Desta forma, o desempenho da rede neural será aprimorado em treinamentos subsequentes.

4) consiste em realizar tarefas de classificação de amostras de imagens avulsas utilizando o modelo treinado, com o objetivo de simular um ambiente de operação realizando tarefas de classificações com imagens nunca utilizadas no processo de aprendizagem dos parâmetros do modelo.

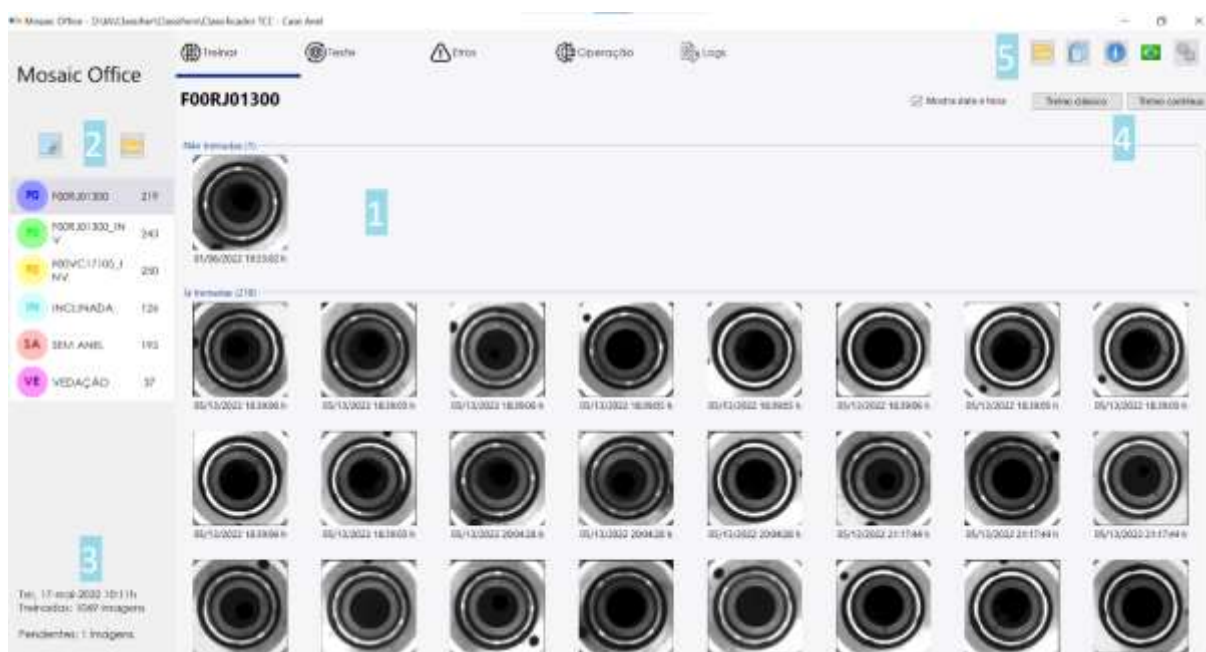
5) oferece um registro detalhado das atividades executadas pelo sistema, como o processo de treinamento/teste do modelo.

Mais detalhes sobre as interfaces implementadas neste trabalho estão disponíveis a seguir.

4.3.5 INTERFACE GRÁFICA DE TREINO

A interface gráfica de treino oferece a possibilidade de coletar, pré-classificar e inspecionar imagens, assim como treinar a rede neural com alguma das implementações de algoritmos de treinamento: Treinamento Contínuo e Treinamento Clássico, conforme ilustra a FIGURA 21 seguir.

FIGURA 21 - INTERFACE GRÁFICA DE TREINO



FONTE: O autor (2022).

O QUADRO 1 a seguir exibe a descrição dos controles da interface de treino ilustrada na FIGURA 21.

QUADRO 1 - DESCRIÇÃO DOS CONTROLES

Controle		Descrição
1	Área de exibição das imagens da classe	O espaço onde as imagens da classe são exibidas. Tem-se o agrupamento das imagens treinadas e não treinadas, junto a data e hora de modificação das mesmas. Contém a opção de duplo clique com o botão esquerdo para inspecionar a imagem.
2	Classes e ações	Fornece acesso às classes da aplicação, permitindo a inserção e remoção destas de acordo com os requisitos da aplicação. Contém o menu de clique com o botão direito sob cada classe para edição, assim como a opção de abrir o diretório da classe localizado na área de trabalho da aplicação.

3	Informações do modelo treinado	Exibe a data do último treinamento da rede neural, junto a quantidade de imagens treinadas e pendentes.
4	Treinamento	Exibe os botões para iniciar um treinamento com as implementações de algoritmo existentes.
5	Controle de ações	Fornecer acesso à possibilidade de alterar a área de trabalho do projeto em execução. Contém a opção de exportar um relatório de capacidade, que fornece estatísticas gerais e por cada classe da aplicação. O acesso às informações do software se encontra neste controle, bem como a possibilidade de alterar o idioma da aplicação. As configurações do classificador também podem ser alteradas.

FONTE: O autor (2022).

4.3.6 INTERFACE GRÁFICA DE CONFIGURAÇÕES DA APLICAÇÃO

Para realizar as configurações da aplicação, o sistema implementa interfaces que expõem as configurações de identificação e parametrizações do classificador, como ilustra a FIGURA 22, FIGURA 23 e FIGURA 24.

FIGURA 22 - CONFIGURAÇÕES DE IDENTIFICAÇÃO

The image shows a software window titled 'Configurações da aplicação' with a close button in the top right corner. It has three tabs: 'Identificação' (selected), 'Classificador', and 'Geometria'. Under the 'Identificação' tab, there is a section labeled 'Sobre' containing several input fields and a text area. The fields are: 'Elaborador' (containing 'Erick'), 'Revisão' (containing '00'), 'Empresa' (empty), 'Linha de produção' (empty), 'Estação' (empty), and 'Aplicação' (containing 'Classificação de anel'). Below these is a text area for 'Observações' containing the text 'Aplicação de classificação multiclasse implementada como trabalho de conclusão de curso'. To the right of these fields is a logo for 'UNINTER' and a button labeled 'Alterar logo'. At the bottom of the window are 'Ok' and 'Cancelar' buttons.

Configurações da aplicação	
Identificação	
Sobre	
Elaborador	Erick
Revisão	00
Empresa	
Linha de produção	
Estação	
Aplicação	Classificação de anel
Observações	Aplicação de classificação multiclasse implementada como trabalho de conclusão de curso

FONTE: O autor (2022).

As configurações de identificação servem para atribuir parâmetros de identificação específicos à aplicação. Esses parâmetros são utilizados na exportação de relatórios de capacidade, e servem como um mecanismo de descrição da aplicação. Através disso, é possível estabelecer critérios de identificação sobre a perspectiva de avaliação do decorrer de desenvolvimento da aplicação.

FIGURA 23 - CONFIGURAÇÕES DO CLASSIFICADOR

Configurações da aplicação

Identificação Classificador Geometria

Classificador

☐ Usar configurações padrões

Limite de pontuação mínima 70 %

Quantidade de épocas 200

Taxa de aprendizagem 0,1

Tamanho do lote 10

☐ Reutilizar conjunto de treinamento armazenados em cache

☐ Reutilizar conjunto de teste armazenados em cache

☐ Salvar imagens classificadas no diretório

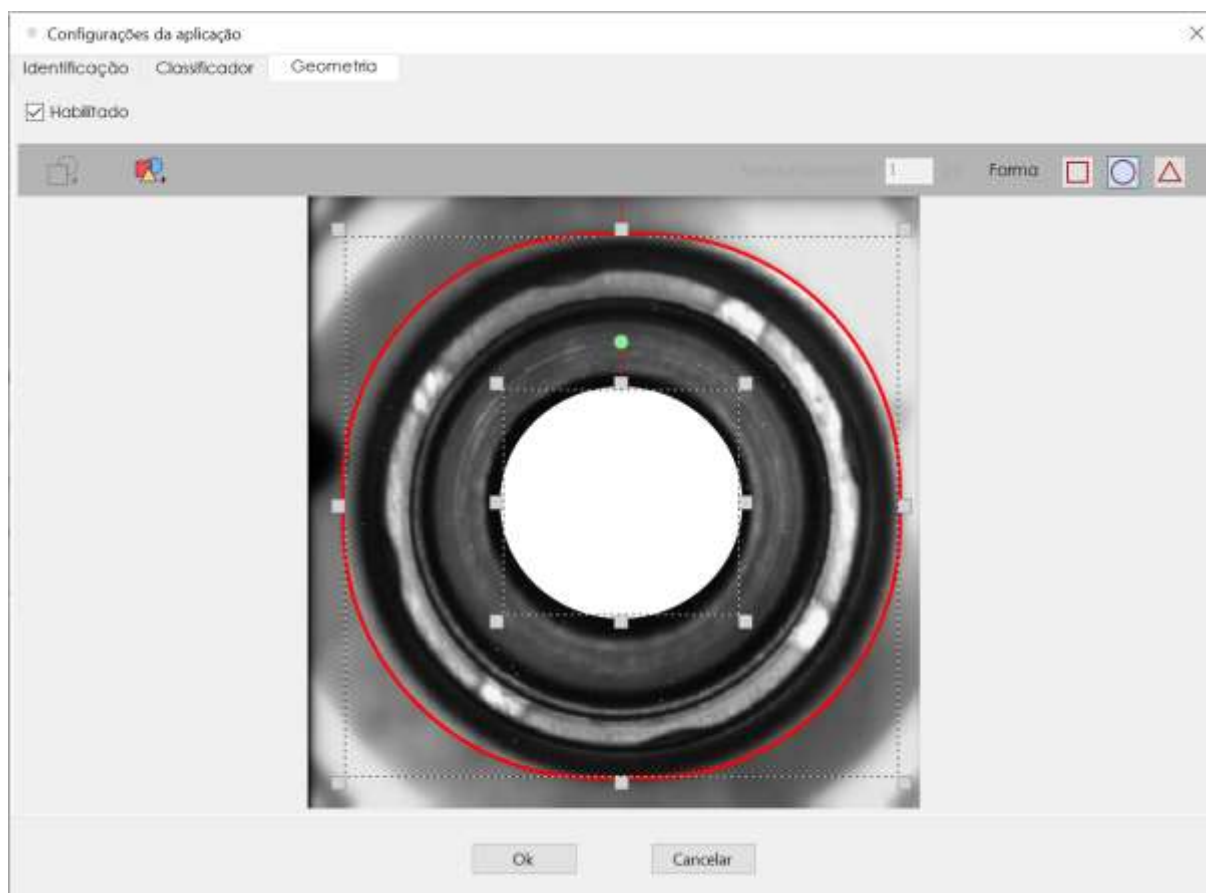
Caminho D:\IA\Classifier\Classifiers\Classificador TCC - C

Ok Cancelar

FONTE: O autor (2022).

Através das parametrizações do classificador, pode-se definir os hiperparâmetros inerentes ao processo de treinamento da rede neural e também os parâmetros de avaliação de desempenho do modelo de aprendizagem de máquina treinado. Diferentemente dos parâmetros da rede neural, os hiperparâmetros não são naturalmente aprendidos durante o treinamento e, portanto, devem ser configurados manualmente.

FIGURA 24 - CONFIGURAÇÕES DAS FERRAMENTA DE GEOMETRIA



FONTE: O autor (2022).

As configurações das ferramentas de geometria incluem a capacidade de delimitar regiões de interesse e aplicar máscaras nas imagens utilizadas pela aplicação. A delimitação manual da região de interesse nos conjuntos de imagens de treino e teste garante a utilização de uma menor quantidade de *pixels*, na premissa de que haja um ganho de desempenho no processo de análise feita pelo classificador, pois consiste em recortar a imagem deixando somente características inerentes à classe, excluindo informações irrelevantes do ponto de vista do classificador.

Quando uma máscara é aplicada à uma imagem, os valores dos *pixels* da imagem na região de amostragem são sobrepostos com os valores dos *pixels* da máscara. Dessa forma, qualquer informação de imagem disposta na região de amostragem terá o seu valor redefinido de tal forma que seu novo valor não influenciará nas tomadas de decisões do classificador.

A FIGURA 24 ilustra uma possível configuração de ferramentas de geometria onde é delimitado uma região de interesse circular destacada na cor vermelha em torno do anel, junto a aplicação de uma máscara no interior do anel.

Essas ferramentas de geometria serão aplicadas a todo o conjunto de imagens, com base nas coordenadas da amostragem definida. Portanto, é necessário que todo o conjunto de imagens seja padronizado, ou seja, tenham as mesmas características nas mesmas coordenadas.

4.3.7 INTERFACE GRÁFICA DE TESTE

A interface gráfica de teste consiste em realizar uma rotina de testes com as amostras de imagens da aplicação. Ao iniciar esta rotina, o classificador realiza a tarefa de classificação para cada amostra de imagem junto a avaliação do resultado da tarefa mediante o parâmetro de limite de pontuação mínima exigida pela aplicação. Caso a tarefa não seja bem sucedida, a amostra de imagem é encaminhada ao diretório de erros da aplicação, onde poderá ser classificada manualmente e adicionada à coleção de imagens de treinamento. A FIGURA 25 exibe o resultado da rotina de teste de um modelo treinado em uma aplicação.

FIGURA 25 - INTERFACE GRÁFICA DE TESTE

Imagem	Pré classificação	Classificado como	Pontuação	Tempo
F00R01300	F00R01300	F00R01300	99.00	2279 ms
1_Anel_INCLINADO_NG001...	F00R01300	F00R01300	98.78	488 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	98.89	211 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	99.16	214 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	77.21	117 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	97.30	117 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	98.87	121 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	93.85	125 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	97.35	114 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	96.19	126 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	99.32	116 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	98.79	120 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	99.20	115 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	96.24	111 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	95.24	126 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	99.37	125 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	97.20	143 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	95.62	128 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	94.60	123 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	98.85	126 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	99.01	120 ms
1_Anel_INCLINADO_NG002...	F00R01300	F00R01300	98.47	130 ms

Total de 28287 imagens
99.88% OK
0.12% NOK

FONTE: O autor (2022).

Conforme ilustra a FIGURA 25, a interface em questão expõe os resultados da rotina de teste através de uma listagem contendo todas as tarefas de classificação realizadas. Cada item da lista representa a tarefa de classificação de uma amostra de imagem da aplicação, contendo o nome da amostra de imagem, sua pré-classificação e classe predita, junto a sua pontuação e tempo decorrido na tarefa em milissegundos.

A interface de teste dispõe de uma caixa de seleção para filtrar as amostras de imagens e exibir somente as amostras com tarefas de classificações incorretas, conforme ilustra a FIGURA 26 a seguir.

FIGURA 26 - INTERFACE GRÁFICA DE TESTE COM FILTRO DE SELEÇÃO

Imagem	Pré-classificação	Classificado como	Pontuação	Tempo
FOUR01300				
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	76,63	169 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	73,23	128 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	68,15	131 ms
Image06234_3.png	FOUR01300	FOUR01300_INV	65,43	187 ms
Image06219_5.png	FOUR01300	FOUR01300_INV	64,27	159 ms
Image06148_5.png	FOUR01300	SEM ANEL	57,69	130 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	55,49	140 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	55,33	120 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	54,98	136 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	54,92	148 ms
Image06072_5.png	FOUR01300	FOUR01300_INV	54,85	139 ms
Image06224_8.png	FOUR01300	FOUR01300_INV	54,01	154 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	53,96	119 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	52,82	250 ms
Image06109_3.png	FOUR01300	FOUR01300_INV	52,62	138 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	52,27	123 ms
Image06093_4.png	FOUR01300	FOUR01300_INV	50,67	138 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	50,47	140 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	50,25	117 ms
1_Anel_INCLINADO_NG905...	FOUR01300	FOUR01300_INV	50,02	136 ms
1_Anel_INCLINADO_NG903...	FOUR01300	FOUR01300_INV	49,33	210 ms
1_Anel_INCLINADO_NG904...	FOUR01300	FOUR01300_INV	45,00	160 ms

FONTE: O autor (2022).

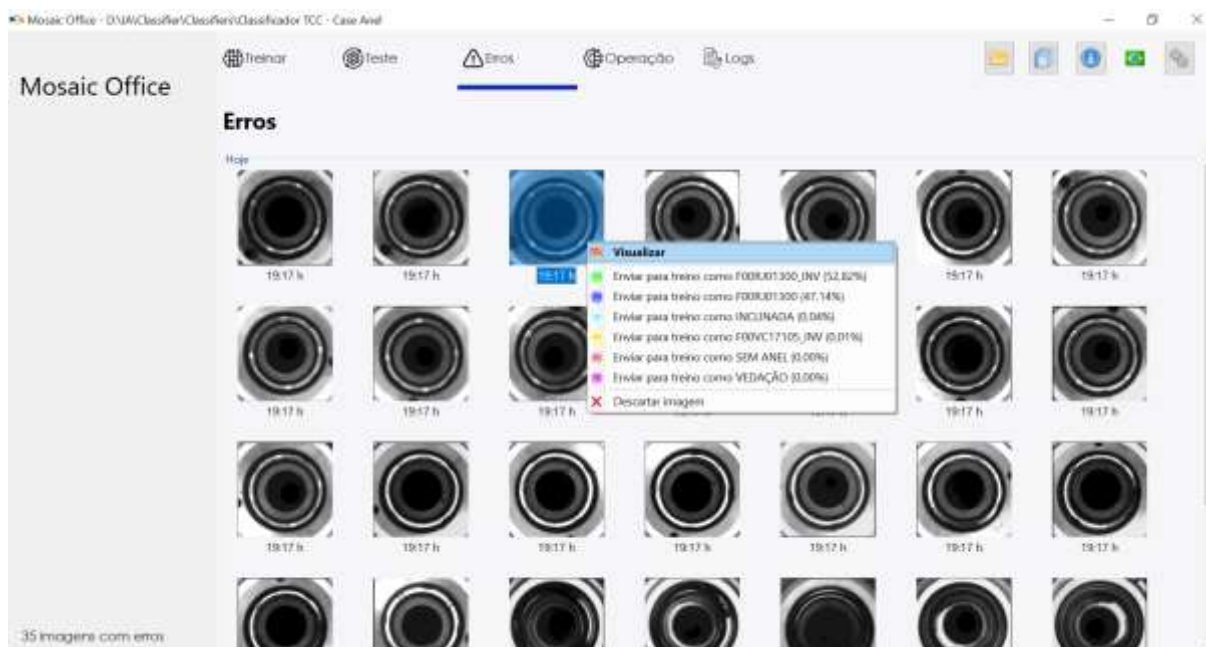
Em geral, o objetivo da interface gráfica de teste é ter a possibilidade de testar o modelo treinado com as amostras de imagens do diretório de teste da aplicação, e encaminhar as amostras de imagens que tiveram sua tarefa de classificação mal sucedida para o diretório de erros da aplicação.

4.3.8 INTERFACE GRÁFICA DE ERRO

A interface gráfica de erros é utilizada para aprimorar o desempenho do modelo treinado. Ela consiste em visualizar as amostras de imagens do diretório de erros da

aplicação e oferecer a possibilidade de o usuário realizar *feedbacks* de classificação manual sob estas amostras de imagens, conforme ilustra a FIGURA 27 seguir.

FIGURA 27 - INTERFACE GRÁFICA DE ERRO



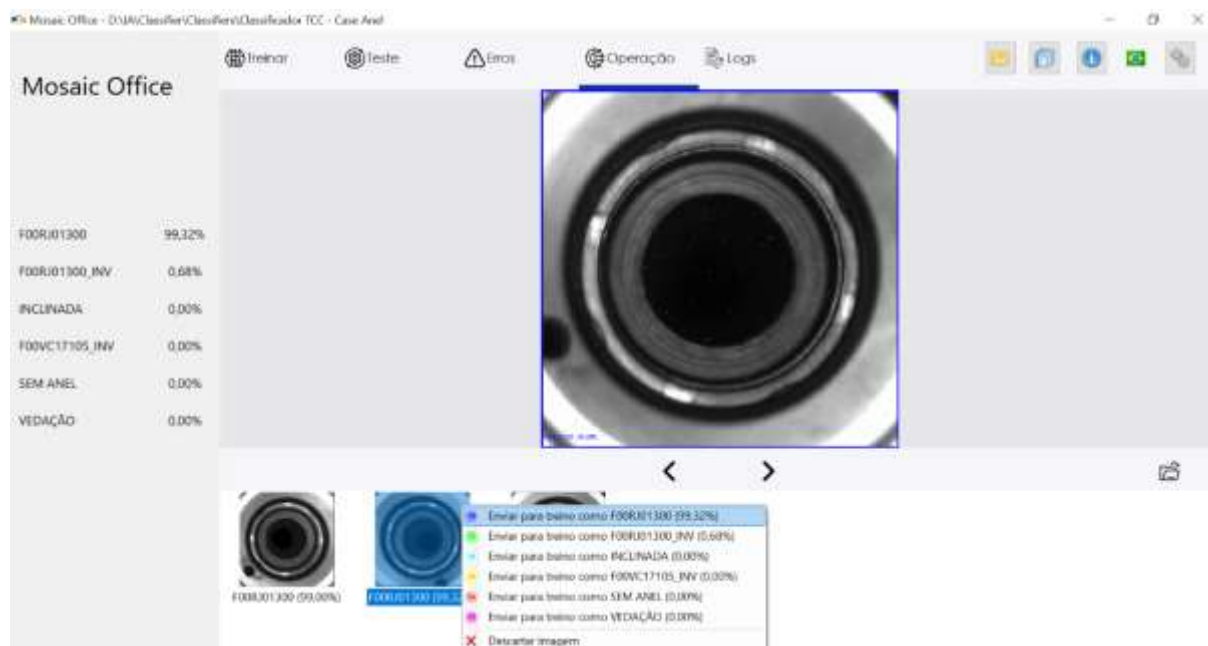
FONTE: O autor (2022).

Através do mecanismo de classificação manual da imagem, tem-se a possibilidade de encaminhar a imagem em questão para o diretório de treinamento da classe escolhida. Desta forma, o desempenho da rede neural será aprimorado em treinamentos subsequentes. Este feito parte da premissa de que esta grandeza é favorecida à medida que se aumenta o tamanho das amostras de imagens de treinamento da classe.

4.3.9 INTERFACE GRÁFICA DE OPERAÇÃO

A interface gráfica de operação consiste em realizar tarefas de classificação de amostras de imagens avulsas utilizando o modelo treinado, conforme ilustra a FIGURA 28. O objetivo desta interface é simular um ambiente de operação onde é possível realizar tais tarefas com amostras de imagens nunca utilizadas no processo de aprendizagem dos parâmetros do modelo.

FIGURA 28 - INTERFACE GRÁFICA DE OPERAÇÃO



FONTE: O autor (2022).

Esta interface também implementa o aprimoramento de desempenho do modelo através de *feedbacks* de classificação manual sob as amostras de imagens.

4.3.10 INTERFACE GRÁFICA DE REGISTROS

A interface gráfica de registros expõe as atividades realizadas pelo sistema. Durante a execução de atividades como treinamento e teste da rede neural, o sistema registra o progresso de execução destas atividades, suas métricas de desempenho, dentre outras informações relevantes, conforme ilustra a FIGURA 29 a seguir.

FIGURA 29 - INTERFACE GRÁFICA DE REGISTROS

ID	Data	Tempo	Classificador	Caso	Descrição
1701	05/06/2022	12:11:53	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1702	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1703	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,91%
1704	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 130 ms
1705	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_1.png' (22 de 28287)
1706	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_1.png'
1707	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1708	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_1.png'
1709	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1710	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1711	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,47%
1712	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 96 ms
1713	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_2.png' (23 de 28287)
1714	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_2.png'
1715	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1716	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_2.png'
1717	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1718	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1719	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,70%
1720	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 121 ms
1721	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_3.png' (24 de 28287)
1722	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_3.png'
1723	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1724	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_3.png'
1725	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1726	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1727	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,30%
1728	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 125 ms
1729	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_4.png' (25 de 28287)
1730	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_4.png'
1731	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1732	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_4.png'
1733	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1734	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1735	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,30%
1736	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 125 ms
1737	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_5.png' (26 de 28287)
1738	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_5.png'
1739	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1740	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_5.png'
1741	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1742	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1743	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,77%
1744	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 131 ms
1745	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_6.png' (27 de 28287)
1746	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_6.png'
1747	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1748	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_6.png'
1749	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1750	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1751	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,77%
1752	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 131 ms
1753	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Testando imagem '1_Anel_INCLINADO_MG00287_7.png' (28 de 28287)
1754	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificando imagem '1_Anel_INCLINADO_MG00287_7.png'
1755	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem classificada corretamente!
1756	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Imagem: '1_Anel_INCLINADO_MG00287_7.png'
1757	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Pre-classificação: 'F00RJ01300'
1758	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Classificado como: 'F00RJ01300'
1759	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Portuguesa: 99,77%
1760	05/06/2022	12:11:52	Classificador TCC	Caso Anel	Tempo: 131 ms

FONTE: O autor (2022).

O objetivo desta interface é oferecer a possibilidade de acompanhamento das atividades realizadas pelo sistema. Por meio disso, pode-se capacitar o usuário com o conhecimento prévio de informações inerentes à atividade em execução.

4.4 ETAPA 4 – AVALIAÇÃO DO SISTEMA PROPOSTO

Para a execução do processo de treinamento e teste do modelo de aprendizagem de máquina, utilizou-se uma amostra com 28.287 imagens pré-classificadas por um usuário especialista na fabricação do produto (anel) em 6 classes: *F00RJ01300* (4610 imagens), *F00RJ01300_INV* (5670 imagens), *F00VC17105_INV* (4580 imagens), *INCLINADA* (5170 imagens), *SEM ANEL* (5757 imagens), *VEDAÇÃO* (2500 imagens). As especificações de *hardware* e *software* em que foram executadas tais processos são descritas na TABELA 2 **Erro! Fonte de referência não encontrada..**

TABELA 2 - ESPECIFICAÇÕES DE *HARDWARE* E *SOFTWARE*

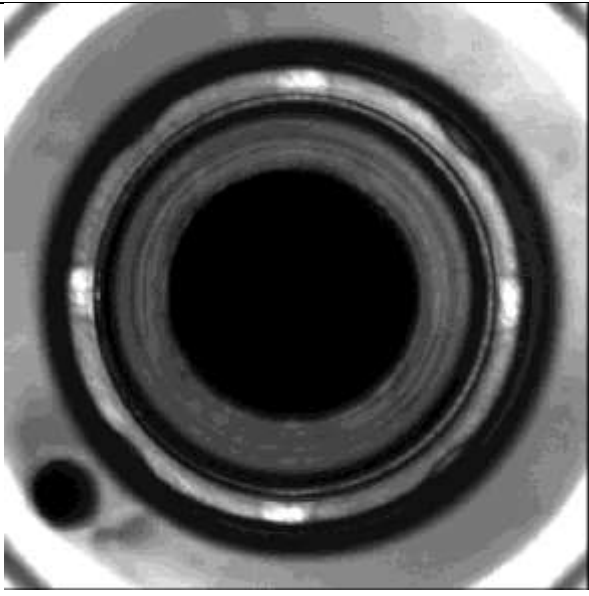
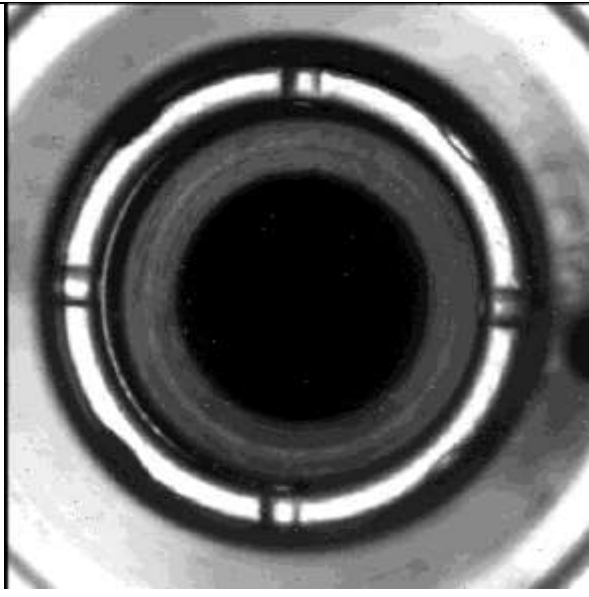
CPU	Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
GPU	NVIDIA MX230 4GB GDDR5

Memória RAM	16 GB
Sistema	Windows 10 pro 64 bits (10.0, Compilação 19044)
Armazenamento	SSD 240 GB

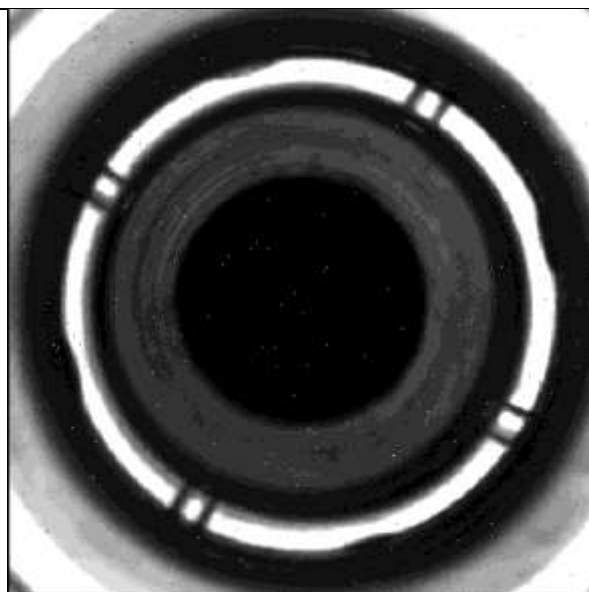
FONTE: O autor (2022).

Com a definição das classes e suas respectivas imagens, definiu-se que a tarefa de *Machine Learning* está centrada em realizar a classificação destas imagens com base nas características inerentes ao anel, como o tipo, a existência ou inexistência do mesmo, conforme ilustra a TABELA 3 seguir.

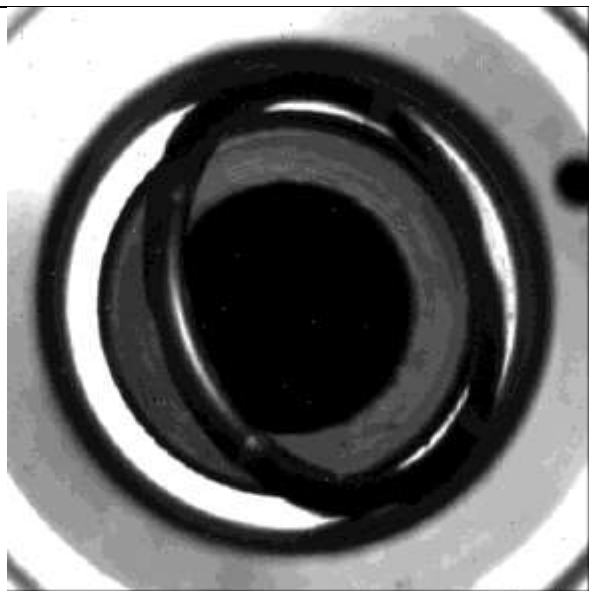
TABELA 3 – IMAGENS CARACTERÍSTICAS DAS CLASSES AVALIDAS

<i>F00RJ01300</i>	
<i>F00RJ01300_INV</i>	

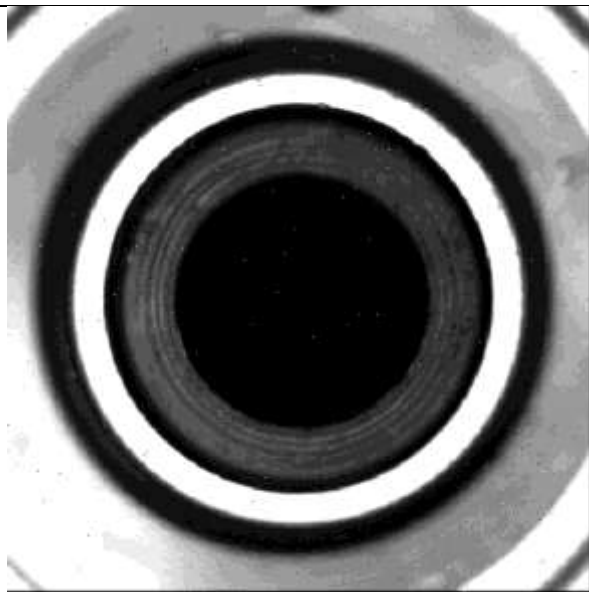
F00VC17105_INV

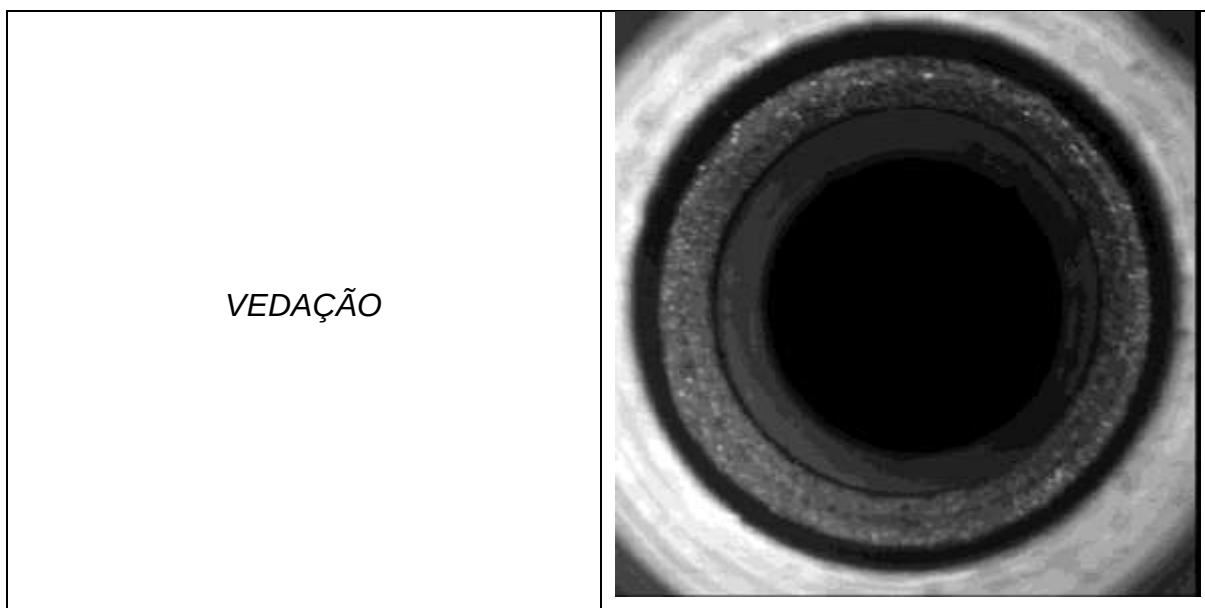


INCLINADA



SEM ANEL





FONTE: O autor (2022).

Portanto, ao realizar uma tarefa de classificação sobre uma imagem de entrada, o modelo treinado retornará as probabilidades de a imagem pertencer a cada classe.

Para construir o modelo de aprendizagem de máquina, é necessário um *pipeline* de treinamento do modelo. Esse processo ocorre inicialmente com a definição do conjunto de imagens de treinamento e teste. O conjunto de imagens de treinamento pode ser definido através de um especialista responsável pela aquisição de imagens estratégicas ou através de um mecanismo de carga inicial de imagens aleatórias retiradas do conjunto de imagens de teste. O mecanismo de carga inicial é executado somente quando o conjunto de imagens de treinamento não está previamente definido. Neste caso, a quantidade de imagens a serem definidas para o conjunto de imagens de treinamento e teste depende do algoritmo de treinamento utilizado.

No algoritmo de Treinamento Clássico, as imagens de treinamento representam 90% da amostra de imagens de cada classe, restando 10% da amostra de imagens de cada classe para o conjunto de imagens de teste. Já no algoritmo de Treinamento Contínuo, as imagens do conjunto de treinamento são definidas como sendo 10 imagens por classe da amostra, enquanto o conjunto de imagens de teste possui todas as imagens da amostra. Em ambos os casos as imagens são escolhidas através de um algoritmo que embaralha o conjunto de imagens e as escolhe aleatoriamente.

Após os conjuntos de imagens terem sido definidos e posteriormente carregados em memória, os dados de treinamento de entrada podem ser utilizados

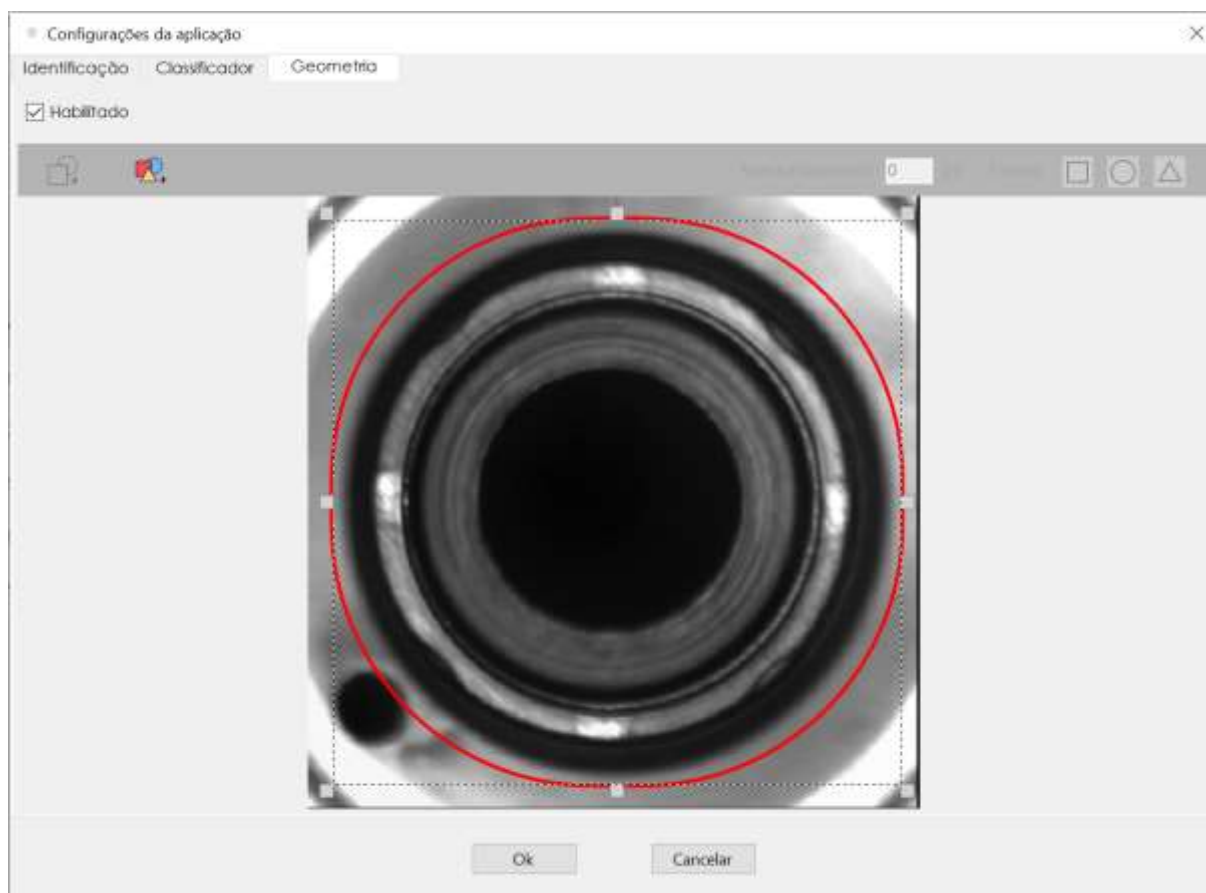
para estimar os parâmetros do modelo. Em caráter de comparação de desempenho do sistema implementado neste trabalho com o sistema de visão Matrox, utilizou-se o algoritmo de Treinamento Clássico afim de garantir a máxima semelhança no tamanho dos conjuntos de imagens utilizados nos sistemas. Já o algoritmo de Treinamento Contínuo foi utilizado como fonte de comparação de desempenho entre os algoritmos de treinamento implementados pelos sistemas. Para avaliação deste algoritmo, partiu-se do modelo treinado através do Treinamento Clássico afim de aprimorar o seu desempenho.

Em ambas as implementações de algoritmos de treinamento, utilizou-se um limite de pontuação mínima de 70% nas classificações definido pelo responsável da aplicação na indústria, junto ao algoritmo de treinamento do modelo de aprendizagem de máquina chamado *ImageClassificationTrainer*. Neste algoritmo é necessário definir as classes e os conjuntos de imagens a serem utilizados durante a estimação dos parâmetros do modelo.

Para a validação das ferramentas de geometria implementadas neste trabalho, utilizou-se um ROI em torno do anel, conforme ilustra a FIGURA 30 seguir. O desempenho do algoritmo de Treinamento Clássico foi comparado com o algoritmo de Treinamento Clássico com esta ferramenta de geometria aplicada.

A delimitação manual desta região de interesse pressupõe que haja um ganho de desempenho nas tarefas de classificações de imagens, uma vez que as informações dos *pixels* externos à esta região não possuem características inerentes às classes e, portanto, podem ser descartados. Para encontrar um valor ótimo, realizou-se uma estimativa empírica da delimitação manual da região de interesse.

FIGURA 30 - FERRAMENTA DE GEOMETRIA APLICADA - ROI



FONTE: O autor (2022).

O algoritmo de treinamento do modelo de aprendizagem de máquina utilizado neste trabalho possui hiperparâmetros avançados opcionais, assim como o algoritmo de treinamento utilizado pelo sistema de visão Matrox. Foram utilizados os valores padrões pré-definidos ilustrados na TABELA 4 e TABELA 5 a seguir.

TABELA 4 - HIPER PARÂMETROS DE TREINAMENTO DO SISTEMA IMPLEMENTADO

Quantidade de épocas	60
Taxa de aprendizagem	0,005
Tamanho do lote	5
Tarefa de Machine Learning	Classificação multiclasse
Engine	NVIDIA MX230
Algoritmo	ImageClassificationTrainer
Arquitetura	Resnet V2 50

FONTE: O autor (2022).

TABELA 5 - HIPER PARÂMETROS DE TREINAMENTO DO SISTEMA DE VISÃO MATROX

Quantidade de épocas	60
Taxa de aprendizagem	0,005
Tamanho do lote	5
Tarefa de Machine Learning	Classificação multiclasse
Engine	NVIDIA MX230
Arquitetura	FCNET Small

FONTE: O autor (2022).

Depois de ter finalizado o processo de construção do modelo de aprendizagem de máquina, iniciou-se a rotina de testes a fim de avaliar o desempenho do classificador. Esta rotina consiste em realizar predições sobre o conjunto de imagens e comparar as classes preditas com a classe verdadeira conhecida, assim como a verificação se as pontuações atenderam o limite inferior de especificação de 70% (LSL). Estes resultados são utilizados como base de cálculo na comparação e avaliação do *Mosaic Office* por meio de técnicas estatísticas e índice de capacidade como descrito a seguir.

4.4.1 MODELO PROBABILÍSTICO DO PROCESSO

Ao realizar a tarefa de classificação de uma imagem qualquer, o modelo treinado retornará as probabilidades de a imagem pertencer a cada classe. O *Mosaic Office* considera a classe predita como sendo a classe que contém a maior probabilidade dentre as demais probabilidades retornadas pelo modelo.

No que diz respeito ao modelo probabilístico do sistema implementado neste trabalho, uma amostra é considerada como o conjunto de classificações extraídas de uma rotina de teste aleatória. Cada classificação de uma amostra é considerada como sendo o elemento amostral, e deste são extraídas características inerentes ao processo, como a classe predita e sua probabilidade. A variável aleatória por sua vez, é definida como sendo a característica observada no elemento amostral, neste caso a probabilidade.

Tendo em vista o conjunto de valores possíveis definidos no intervalo de $[0, 1]$ de uma variável aleatória qualquer, definiu-se a distribuição Beta para modelar a incerteza sobre a probabilidade de ocorrência de classificações com probabilidades inferiores ao limite estabelecido pelos requisitos do projeto.

A função densidade de probabilidade da distribuição beta em que o limite inferior e superior são 0 e 1, respectivamente, é definida na Equação 16 a seguir (FORBES, EVANS, *et al.*, 2011).

$$f(x) = \frac{x^{v-1}(1-x)^{w-1}}{B(v, w)} \quad (16)$$

Na função acima, $B(v, w)$ é a função beta com parâmetros de forma v e w definida como na Equação 17 a seguir (FORBES, EVANS, *et al.*, 2011).

$$B(v, w) = \int_0^1 u^{v-1}(1-u)^{w-1} du \quad (17)$$

A função beta incompleta, uma generalização da função beta, é a função de distribuição cumulativa definida como na Equação 18 a seguir (GUPTA e NADARAJAH, 2004).

$$B(v, w) = \int_0^x u^{v-1}(1-u)^{w-1} du \quad (18)$$

Os parâmetros de forma v e w podem ser estimados pelo método de máxima verossimilhança ou através do método dos momentos. De acordo com (FORBES, EVANS, *et al.*, 2011), as equações de máxima verossimilhança para a distribuição beta não têm solução fechada.

Utilizou-se o método dos momentos para estimar os parâmetros v e w , como descrito na Equação 18 e 19 a seguir.

$$v = \bar{x} \left(\frac{\bar{x}(1-\bar{x})}{s^2} - 1 \right) \quad (18)$$

$$w = (1 - \bar{x}) \left(\frac{\bar{x}(1 - \bar{x})}{s^2} - 1 \right) \quad (19)$$

De acordo com (FORBES, EVANS, *et al.*, 2011), a função de distribuição cumulativa é a probabilidade de que a variável aleatória obtenha um valor menor ou igual a x , como define a Equação 20 a seguir.

$$f(x) = Pr[X \leq x] = \alpha \quad (20)$$

Desta forma, assumindo as variáveis aleatórias como sendo as probabilidades das classificações obtidas de uma rotina de teste aleatória, e x como sendo o limite de probabilidade mínima estabelecida pelo projeto, utilizou-se a função de distribuição cumulativa para calcular a probabilidade de ocorrência de classificações com probabilidades inferiores a x .

4.4.2 ÍNDICE DE CAPABILIDADE DE PROCESSO CP

De acordo com (SAFDAR, AHMED, *et al.*, 2019), índices de capacidade projetados para processos normalmente distribuídos fornecem resultados errôneos para processo não normais. Sobre estas circunstâncias, inúmeros métodos foram propostos para estimar estes índices. Este trabalho utiliza a estimação do índice de capacidade potencial CP proposta por (SAFDAR, AHMED, *et al.*, 2019) para processos não normais, como define a Equação 21 a seguir.

$$CP = \frac{USL - LSL}{U_p - L_p} \quad (21)$$

Onde:

- Cp é a capacidade de processo;
- USL é o limite superior de tolerância;

- LSL: é o limite inferior de tolerância;
- U_p é o percentil 99,865 da distribuição;
- L_p é o percentil 0,135 da distribuição.

Os valores mínimos para a capacidade de um processo são relativos às características inerentes ao processo. Para (MONTGOMERY, 2016), diversos critérios de avaliação podem ser adotados, sendo os valores mínimos recomendados ilustrado na FIGURA 31 a seguir.

FIGURA 31 - VALORES MÍNIMOS RECOMENDADOS PARA A CAPACIDADE DE PROCESSO

	Especificações Bilaterais	Especificações Unilaterais
Processos existentes	1,33	1,25
Novos processos	1,50	1,45
Segurança, força, ou parâmetro crítico, processo existente	1,50	1,45
Segurança, força, ou parâmetro crítico, processo novo	1,67	1,60

FONTE: O autor (2022).

Neste trabalho, pelo fato de o projeto estar em fase de implementação, junto a característica de especificação unilateral, adotou-se o valor mínimo de 1,45 para avaliar a capacidade potencial inerente à tarefa de classificação de imagem multiclasse.

4.4.3 ESTATÍSTICA KAPPA

Utilizando a Equação 28 do Capítulo 3.8 com os dados da classificação de cada sistema e os parâmetros de referência, dado a hipótese nula $H_0:K = 0$ em que a concordância entre os dados da classificação e os dados de referência é igual a uma classificação aleatória. A hipótese alternativa $H_1:K \neq 0$ em que a concordância entre os dados da classificação e os dados de referência é significativamente melhor do que uma classificação aleatória. Tem-se que H_0 é rejeitado se $Z \geq Z_\alpha$, onde Z_α é o nível

de confiança do teste Z. Empregando-se um nível de confiança de 95%, o valor crítico seria 1,96. Portanto, se o valor absoluto do teste de significância Z for maior que 1,96, conclui-se que o resultado da estatística Kappa é significativo e melhor do que uma classificação aleatória (CONGALTON e GREEN, 2019).

Para avaliar se dois valores Kappa independentes são significativamente diferentes, utilizou-se a Equação 29 do Capítulo 3.8 com um nível de confiança de 95%. Dado a hipótese nula $H_0: (K_1 - K_2) = 0$ em que os dois avaliadores são iguais, e a alternativa $H_1: (K_1 - K_2) \neq 0$ em que os dois avaliadores são significativamente diferentes. Tem-se que H_0 é rejeitado se $Z \geq Z_\alpha$, onde Z_α é o nível de confiança do teste Z.

5 RESULTADOS DA AVALIAÇÃO DE DESEMPENHO

A viabilidade de um sistema de visão computacional mediante a tarefa de classificação de imagens depende de o sistema satisfazer os requisitos estabelecidos a partir das especificações do projeto. A seguir, serão descritos os resultados obtidos na etapa de avaliação e comparação de desempenho do sistema implementado neste trabalho.

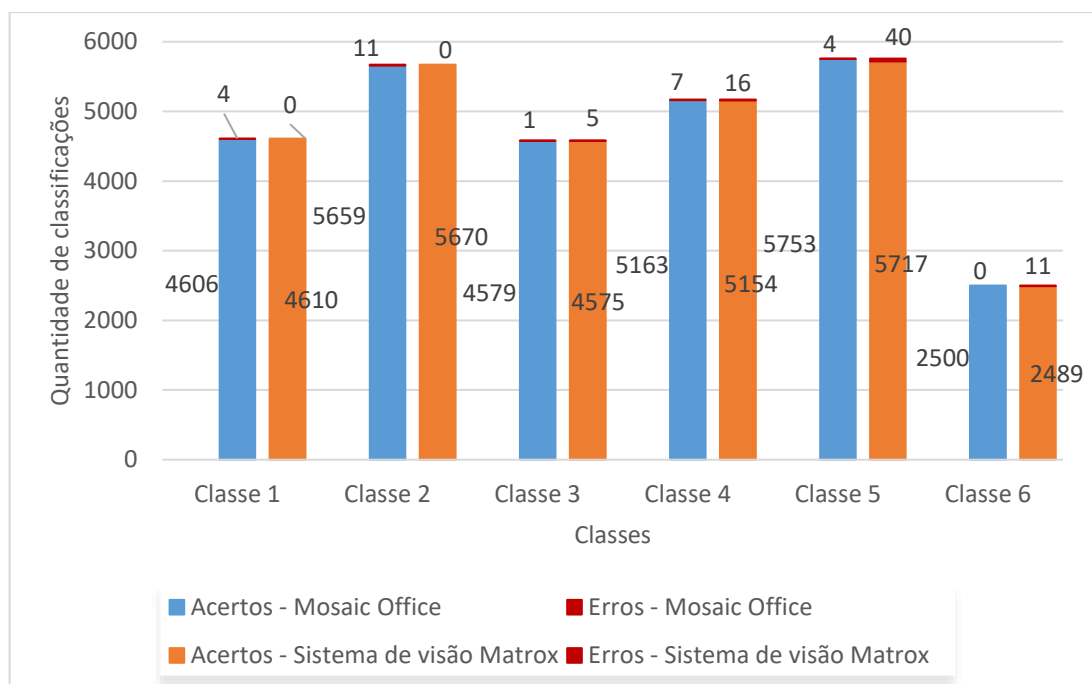
5.1 COMPARAÇÃO ENTRE OS SISTEMAS

Para treinar o modelo utilizando o conjunto de imagens descrito no Capítulo 4.4, foram necessárias 122h para sistema de visão Matrox e 1,16h para o *Mosaic Office*, uma redução de aproximadamente 99,04%. No *Mosaic Office* o treinamento foi realizado por completo. Porém, no sistema de visão Matrox, o treinamento foi interrompido durante a 21ª época de treinamento, visto que o tempo de treinamento do sistema já havia excedido em mais de 100 vezes o tempo do sistema proposto. Além disso, na 21ª época de treinamento a taxa de erro de treinamento já havia sido minimizada ao valor ideal de acordo com um representante técnico do sistema de visão Matrox. Portanto, a diferença entre os sistemas neste aspecto torna-se ainda mais evidente.

Afim de verificar qual sistema se destaca nas tarefas de classificações de imagens, o GRÁFICO 1 a seguir exibe o comparativo de acertos/erros de cada classe entre os sistemas. Referente a quantidade de erros cometidos pelos sistemas, pode-

se observar que o *Mosaic Office* teve um desempenho superior ao sistema de visão Matrox. No total, o sistema de visão Matrox realizou 72 classificações incorretas, enquanto o *Mosaic Office* realizou somente 27.

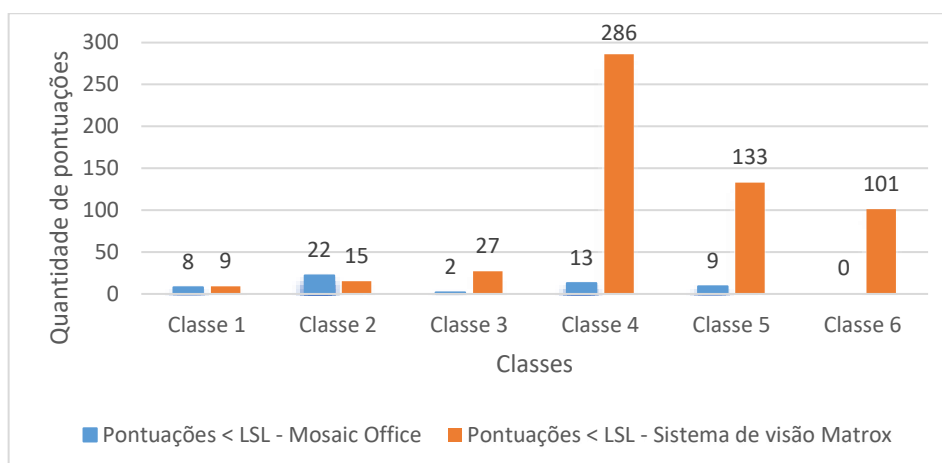
GRÁFICO 1 - QUANTIDADE DE ACERTOS/ERROS DOS SISTEMAS



FONTE: O autor (2022).

Para atender as especificações do projeto com excelência, é preciso minimizar a quantidade de classificações incorretas e a quantidade de classificações com pontuações abaixo do LSL. O GRÁFICO 2 a seguir ilustra o comparativo dessa métrica de desempenho entre os sistemas. O sistema de visão Matrox totalizou 571 classificações com pontuação abaixo de LSL, enquanto o Mosaic Office totalizou 54 classificações. O Mosaic Office realizou aproximadamente dez vezes menos classificações abaixo de LSL quando comparado ao sistema de visão Matrox. Isso evidencia novamente o melhor desempenho do sistema proposto neste trabalho, além de obter os melhores resultados para atender ao requisito de limite de pontuação mínima especificado pelo projeto.

GRÁFICO 2 - QUANTIDADE DE PONTUAÇÕES ABAIXO DO LSL DOS SISTEMAS



FONTE: O autor (2022).

Outras métricas estatísticas dos sistemas são descritas na TABELA 6 E TABELA 7 a seguir.

TABELA 6 - SISTEMA DE VISÃO MATROX - ESTATÍSTICA POR CLASSE

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Revocação	100%	100%	99,89%	99,69%	99,30%	99,56%
Precisão	99,97%	99,93%	99,63%	99,32%	99,80%	99,84%
Pontuação mínima	46,46%	38,78%	31,86%	37,84%	25,40%	33,61%
Pontuação média	96,13%	95,24%	95,24%	90,77%	93,48%	95,74%
Pontuação máxima	99,22%	99,51%	99,22%	99,70%	99,92%	99,81%
Desvio padrão	0,0170	0,0221	0,0291	0,0645	0,0510	0,0513

FONTE: O autor (2022).

TABELA 7 - MOSAIC OFFICE ESTATÍSTICA POR CLASSE

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Revocação	99,82%	99,92%	99,76%	99,98%	99,94%	100%
Precisão	99,91%	99,80%	99,97%	99,86%	99,93%	100%
Pontuação mínima	40,15%	50,00%	50,65%	50,81%	51,80%	82,70%
Pontuação média	99,64%	99,45%	99,72%	99,66%	99,60%	99,92%

Pontuação máxima	100%	100%	100%	100%	100%	100%
Tempo médio	153 ms	152 ms	152 ms	152 ms	152 ms	153 ms
Desvio padrão	0,0060	0,0090	0,0043	0,0062	0,0065	0,0013

FONTE: O autor (2022).

Com base nos resultados das tabelas acima, pode-se observar que a métrica de Revocação e Precisão de ambos os sistemas são semelhantes. Já a pontuação mínima atinge 25,40% na classe 5 no sistema de visão Matrox e 40,15% na classe 1 no Mosaic Office, enquanto a média mínima é 90,77% na classe 4 no sistema de visão Matrox e 99,45% na classe 2 no Mosaic Office. Além disso, o desvio padrão máximo do sistema de visão Matrox é 0,0645 na classe 4 e 0,0090 na classe 2 no Mosaic Office. Portanto, em todos os resultados o Mosaic Office possui melhor desempenho.

Para provar se os classificadores dos sistemas avaliados são estatisticamente melhores do que um classificador aleatório, os resultados da estatística Kappa descritos na TABELA 8 podem ser utilizados.

TABELA 8 - ESTATÍSTICA KAPPA

Sistema	Kappa	Variância	Estatística Z
Mosaic Office	0,9988	$4,9600 \times 10^{-8}$	4484,90
Sistema de visão Matrox	0,9969	$1,3206 \times 10^{-7}$	2743,27

FONTE: O autor (2022).

Com base nos valores da estatística Kappa encontrados para cada sistema, pode-se observar que ambos apresentam forte concordância entre as previsões do modelo e os valores de referência. Utilizando um nível de confiança de 95% para a estatística Z, tem-se que o valor de Z tabelado é 1,96. Como ambos os valores do teste Z calculados para cada sistema são maiores que o valor de Z tabelado, pode-se afirmar que ambos são estatisticamente melhores do que um classificador aleatório.

Para provar que os classificadores são estatisticamente diferentes, o resultado do teste da estatística Z descrito na TABELA 9 pode ser utilizado.

TABELA 9 - COMPARAÇÃO ENTRE OS SISTEMAS

Comparação	Estatística Z
Mosaic Office vs. Sistema de visão Matrox	4,4578

FONTE: O autor (2022).

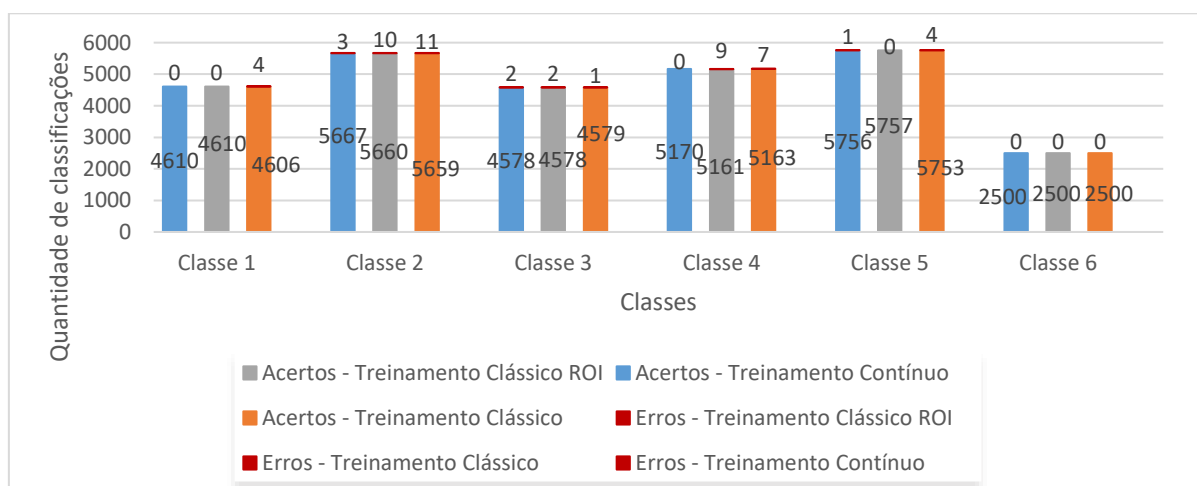
Com base no valor da estatística Z encontrado, assumindo um nível de confiança de 95%, tem-se que o valor de Z tabelado é 1,96. Como o valor calculado do teste Z é maior que o valor de Z tabelado, pode-se afirmar que os classificadores são estatisticamente diferentes.

5.2 AVALIAÇÃO DO SISTEMA PROPOSTO

Utilizando o conjunto de imagens e os modelos descritos no Capítulo 4.4, um comparativo foi realizado entre estes modelos a fim de verificar o melhor desempenho. Espera-se que o modelo treinado com o algoritmo de Treinamento Clássico seja aprimorado pelo algoritmo de Treinamento Clássico com ROI e também pelo algoritmo de Treinamento Contínuo.

O algoritmo de Treinamento Contínuo adicionou 14,07h de treinamento ao modelo descrito no Capítulo 5.1, totalizando 15,23h. O GRÁFICO 3 a seguir exibe o comparativo de acertos/erros de cada classe para cada implementação de algoritmo de treinamento.

GRÁFICO 3 - QUANTIDADE DE ACERTOS/ERROS



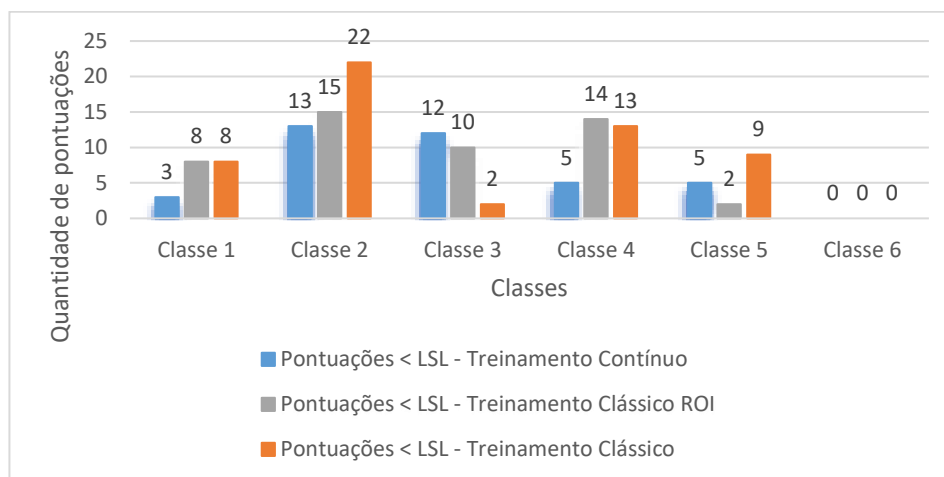
FONTE: O autor (2022).

Com base nos resultados acima, na tentativa de aprimoramento do modelo houve uma perda de desempenho apenas na classe 3. No modelo treinado pelo algoritmo de Treinamento Clássico ocorreu 1 erro e no algoritmo de Treinamento Contínuo ocorreram 2 erros. A quantidade erros do modelo treinado com o ROI aplicado se assemelha com a quantidade de erros do modelo treinado sem a utilização de ferramentas de geometria.

Portanto, pode-se afirmar que através da retroalimentação automática das imagens problemáticas sobre os dados de treinamento houve um ganho de precisão do modelo treinado pelo algoritmo de Treinamento Contínuo. Visto que não houve um ganho desempenho significativo do modelo treinado com o ROI aplicado, pode-se afirmar que não existe a necessidade de delimitação manual da região de interesse sobre as imagens desta aplicação.

O GRÁFICO 4 ilustra a quantidade de classificações com pontuações abaixo do LSL realizadas pelos modelos treinados pelos algoritmos de treinamento.

GRÁFICO 4 – QUANTIDADE DE PONTUAÇÕES ABAIXO DO LSL



FONTE: O autor (2022).

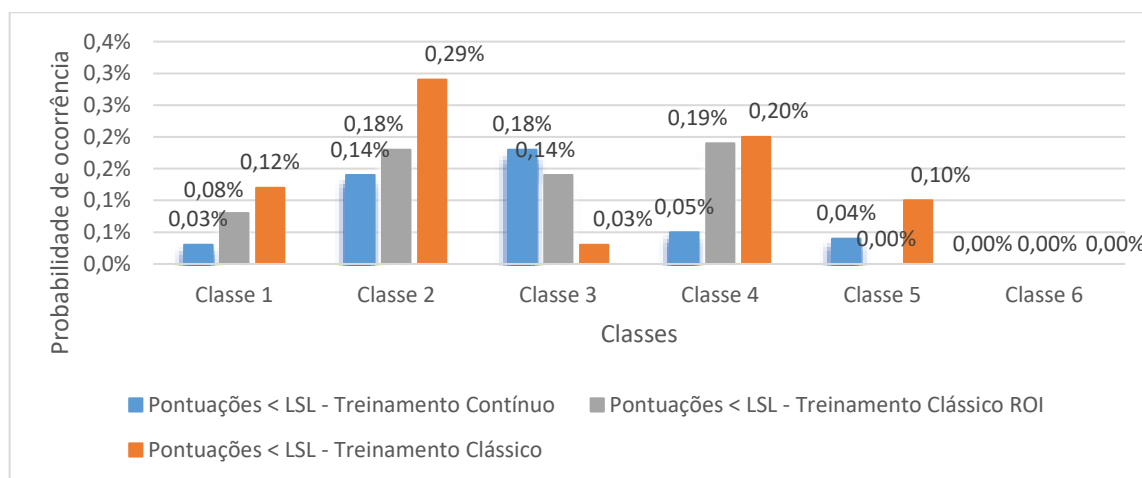
Conforme ilustrado no gráfico acima, o modelo aprimorado pelo algoritmo de Treinamento Contínuo sofreu uma perda de desempenho somente na classe 3, passando a ter 12 classificações com pontuação abaixo do LSL. No restante, houve uma menor quantidade de classificações com pontuações abaixo do LSL. O modelo treinado com o ROI aplicado teve uma menor quantidade de classificações com

pontuações abaixo do LSL nas classes 2 e 5. No restante, seus resultados se assemelham com os resultados do modelo treinado sem ferramentas de geometria.

Os resultados obtidos no GRÁFICO 4 concordam com os resultados obtidos anteriormente pois as classes que possuem a menor quantidade de pontuações abaixo do LSL também possuem a menor quantidade de classificações incorretas. Com base na quantidade de pontuações abaixo do LSL e na quantidade de erros para a classe 3, tanto no modelo aprimorado pelo Treinamento Contínuo como também no modelo treinado com o ROI aplicado, pode-se concluir que estes modelos não conseguiram minimizar o erro com excelência para esta classe. Acredita-se que o motivo pelo qual isso ocorre é por conta da baixa qualidade na amostra de imagens desta classe.

O GRÁFICO 5 a seguir ilustra as probabilidades de ocorrência de classificações com pontuações abaixo do LSL em cada implementação de algoritmo de treinamento.

GRÁFICO 5 - PROBABILIDADE DE OCORRÊNCIA DE PONTUAÇÕES ABAIXO DO LSL



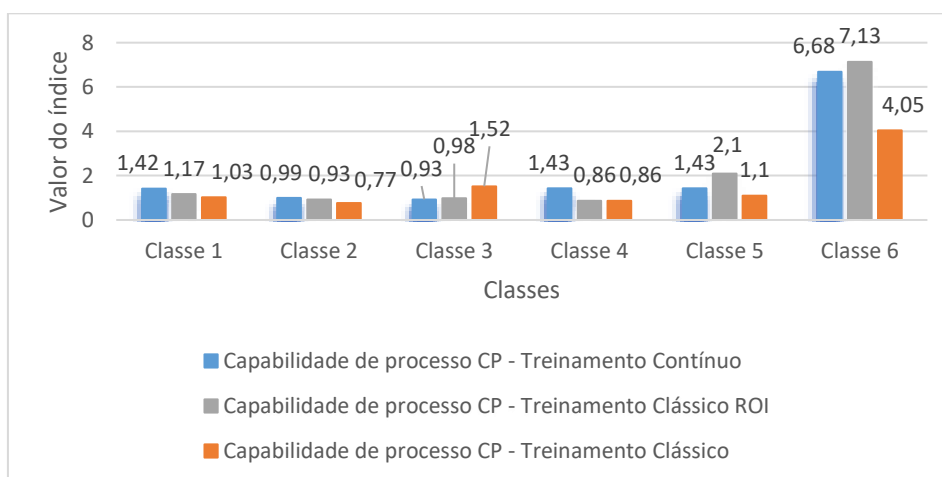
FONTE: O autor (2022).

De acordo com os resultados acima, a probabilidade de ocorrência de classificações com pontuações abaixo do LSL é minimizada com o algoritmo de Treinamento Contínuo, com exceção da classe 3. A probabilidade de ocorrência de classificações com pontuações abaixo do LSL do modelo treinado com ROI aplicado é minimizada significativamente na classe 2. Nas classes restantes, o seu desempenho se assemelha com o modelo treinado sem ferramentas de geometria. Isso evidencia o ganho de desempenho promovido pelo algoritmo de Treinamento

Contínuo e a ausência da necessidade de delimitação manual da região de interesse sobre as imagens desta aplicação.

O GRÁFICO 6 a seguir ilustra os valores do índice de capacidade de processo para cada classe em cada implementação de algoritmo de treinamento.

GRÁFICO 6 - CAPABILIDADE DE PROCESSO



FONTE: O autor (2022).

De acordo com o gráfico acima, pode-se observar um aumento no índice de capacidade de processo com o aprimoramento do modelo realizado pelo algoritmo de Treinamento Contínuo. Com exceção da classe 3, todas as classes tiveram o valor de seu índice de capacidade maximizado. Este ganho de desempenho se dá em função da minimização da quantidade de classificações com pontuações abaixo do LSL.

O valor do índice de capacidade do modelo treinado com ROI aplicado teve um desempenho superior ao modelo treinado sem ferramentas de geometria, com exceção da classe 3. Isso ocorre, pois, a quantidade de classificações com pontuações abaixo do LSL do modelo treinado com ROI aplicado é inferior a quantidade do modelo treinado sem as ferramentas de geometria.

O valor mínimo de 1,45 para o índice de capacidade de processo CP é atendido somente na classe 6 em todos os modelos treinados. Na classe 3 somente no modelo treinado com o algoritmo de treinamento clássico. Já na classe 5 com o modelo treinado com o algoritmo de Treinamento Clássico com ROI aplicado.

Com exceção da classe 3, o modelo treinado com o algoritmo de Treinamento Clássico apresentou os menores índices de capacidade. Esse mesmo algoritmo

treinamento com o ROI aplicado apresentou o segundo maior índice nas classes 1, 2 e 3. O modelo treinado com o algoritmo de Treinamento Contínuo apresentou o melhor desempenho entre os modelos. Ele atendeu o valor mínimo na classe 6 e apresentou valores próximos ao valor mínimo nas classes 1, 4 e 5. Já na classe 2 e 3 ele apresentou valores próximo a 1.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo a elaboração de um sistema de visão computacional (*Mosaic Office*) baseado em técnicas de inteligência artificial para automatização do processo de inspeção visual na indústria. Para oferecer acesso às técnicas de inteligência artificial, o sistema implementa interfaces gráficas que podem ser utilizadas no processo de construção e aprimoramento do modelo. De acordo com os testes funcionais aplicados, pode-se afirmar que as interfaces gráficas contribuíram com (i) o processo de coletar e pré-classificar as imagens; (ii) treinamento do modelo e entendimento dos resultados; (iii) teste e aprimoramento do modelo.

Para treinar o modelo de aprendizagem de máquina, o *Mosaic Office* implementou dois algoritmos de treinamento: Treinamento Contínuo e Treinamento Clássico. Utilizando o conjunto de imagens descrito no Capítulo 4.4, o algoritmo de Treinamento Clássico mostrou-se ser aproximadamente 99,04% mais rápido que o sistema de visão Matrox, além de ter um desempenho superior como descrito no Capítulo 5.1.

Com os resultados obtidos, pode-se afirmar que através da retroalimentação automática das imagens problemáticas sobre os dados de treinamento, teve-se um ganho de desempenho do modelo aprimorado pelo algoritmo de Treinamento Contínuo. Analisando o conjunto de imagens utilizado, não se caracteriza a necessidade de aplicação de um ROI nas imagens. Esta observação está em concordância com os resultados obtidos com o treinamento do modelo com ROI aplicado, onde não se obteve melhorias significativas em comparação aos resultados do modelo treinado sem ferramentas de geometria.

O *Mosaic Office* utiliza a distribuição Beta para modelar a incerteza sobre a probabilidade de ocorrência de classificações com probabilidades inferiores ao limite estabelecido pelos requisitos do projeto. Com base na rotina de teste avaliada, pode-

se concluir que os resultados obtidos apresentaram forte aderência ao modelo probabilístico. Analisando os níveis de desempenho das classes quantificadas pelo índice de capacidade CP, à medida que se aumenta a densidade de classificações com probabilidades superiores ao LSL, tem-se um aumento no nível do índice de capacidade. Portanto, pode-se afirmar que estes níveis estão em concordância com a variação natural do processo.

Mediante a necessidade de exportação dos resultados da rotina de teste através de um documento padronizado, o Mosaic Office implementa a exportação de um relatório estatístico de capacidade da aplicação. Conforme ilustrado na FIGURA 32 no ANEXOS deste trabalho, esse relatório contém (i) os campos de identificação da aplicação; (ii) os hiperparâmetros do classificador utilizados no processo de treinamento do modelo; (iii) um resumo geral dos resultados da rotina de teste e (iv) os resultados específicos de cada classe. Através da avaliação do sistema proposto, onde para cada modelo treinado tem-se um relatório estatístico de capacidade, a utilização destes documentos favoreceu a visualização e comparação dos resultados entre os modelos.

Com base nos resultados obtidos no processo de comparação e avaliação do sistema proposto, pode-se afirmar que o mesmo apresentou resultados superiores ao sistema de visão Matrox. Mediante ao desempenho na tarefa de classificação de imagem, o Mosaic Office apresentou um desempenho superior ao sistema de visão Matrox nos seguintes aspectos: (i) menor quantidade de erros de classificação, (ii) menor quantidade de classificações com pontuações abaixo do LSL; (iii) maior pontuação mínima atingida nas classes; (iv) menor desvio padrão nas classificações em todas as classes. Portanto, além da exclusividade de implementar ferramentas estatísticas para avaliação da capacidade do processo e implementar um algoritmo de otimização de um modelo, o Mosaic Office mostrou-se ser superior ao sistema de visão Matrox.

Através da inspeção visual das imagens utilizadas no processo de avaliação de desempenho dos sistemas, observou-se que estas imagens estão com baixa qualidade, algumas delas apresentam a característica de ruído. Portanto, acredita-se que se as imagens a serem utilizadas na construção do modelo apresentarem maior qualidade, o desempenho do sistema será maximizado.

As sugestões para trabalhos futuros são: (i) adequação do sistema para permitir a utilização de mais de um ROI, permitindo que uma única imagem possa conter

diversas classes. Portanto, para cada ROI teria um modelo treinado para identificar tal característica; (ii) exportar e analisar as matrizes de confusão da aplicação para identificar possíveis vieses de classificação das imagens; (iii) inserir opções de gráficos de desempenho; (iv) explorar outras arquiteturas e parametrizações de redes neurais disponíveis na literatura.

REFERÊNCIAS

ANOCHI, J. A. MODELOS BASEADOS EM REDES NEURAIAS PARA O ESTUDO DE PADRÕES CLIMÁTICOS SAZONAIS A PARTIR DE DADOS TRATADOS COM A TEORIA DOS CONJUNTOS APROXIMADOS. **Ministério da ciência e tecnologia**, São José dos Campos, 2010.

BARRETO, J. M. Introdução às Redes Neurais Artificiais, Florianópolis, abr. 2002.

BESSE, P. Neural Networks and Introduction to Deep Learning, 20 out. 2017.

BRAGA, A. D. P.; CARVALHO, A. C. P. D. L. F. D.; LUDEMIR, T. B. **Redes Neurais Artificiais - Teoria e Aplicações**. 2º. ed. Rio de Janeiro: LTC, 2007.

BRAIN, GOOGLE. Classificação de imagens com TensorFlow Hub. **TensorFlow**, 2022. Disponível em: <https://www.tensorflow.org/hub/tutorials/image_classification>. Acesso em: 13 jun. 2022.

BURY, K. **Statistical Distributions in Engineering**. Cambridge: Cambridge University Press, 1999.

COGNEX. VISIONPRO DEEP LEARNING. **Cognex**, 2021. Disponível em: <<https://www.cognex.com/downloads/literaturemain?event=de3bc3c6-3d9e-43a7-9c4a-42790cc1ea76>>. Acesso em: 01 ago. 2022.

COGNEX. DEEP LEARNING VS. MACHINE VISION EBOOK. **Cognex**. Disponível em: <<https://www.cognex.com/pt-br/resources/white-papers-articles/whitepaperandarticlemain?event=1389055c-4adf-45b7-94a7-e129abe70f1d&cm-campid=7014W000000ufH0QAI>>. Acesso em: 21 ago. 2022.

CONGALTON, R. G.; GREEN, K. **Assessing the Accuracy of Remotely Sensed Data: Principles and Practices**. 3ª. ed. Boca Raton: CRC Press, 2019.

DU, W. et al. Review on the Applications of Deep Learning in the Analysis of Gastrointestinal Endoscopy Images. **IEEE Access**, v. 7, p. 142053-142069, 2019. ISSN 10.1109/ACCESS.2019.2944676.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Patter Classification**. 2ª. ed. Nova York: Wiley, 2001.

ELGENDY, M. **Deep Learning for Vision Systems**. Shelter Island: MANNING, 2020.

FALQUETO, D. Rede Neural artificial para reconhecimento de tabela de horários de arme/desarme no sistema sigma, São José, jun. 2007.

FERNEDA, E. Redes neurais e sua aplicação em sistemas de recuperação de informação, Ribeirão Preto, Janeiro 2006.

FORBES, C. et al. **Statistical Distributions**. 4^a. ed. Canada: Jonh Wiley, 2011.

FURTADO, M. I. V. Redes Neurais Artificiais: Uma abordagem para sala de aula. **Atena**, Belo Horizonte, 2019. ISSN 10.22533/at.ed.262191504.

GALLON, L. Sistema de visão computacional para classificação de pedras naturais através de vídeo em tempo real, Lajeado, 2013.

GHOLIZADEH, S.; ZHOU, N. Model Explainability in Deep Learning Based, 14 jun. 2021. 12.

GOES, S. Desenvolvimento de sistema de inspeção automatizada de componentes manufaturados baseada em visão de máquina. **XXXIV Encontro nacional de engenharia de produção**, 7 out. 2014. 12.

GUALDA, I. P. Aplicação de Redes Neurais Artificiais na Ciência e Tecnologia de alimentos: Estudo de casos, Londrina, 2008.

GUPTA, A. K.; NADARAJAH, S. **Handbook of Beta Distribution and Its Applications**. New York: CRC Press, 2004.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. 2^o. ed. Canadá: Bookman, 2020.

KANE, V. E. Process capability indices. **Journal of quality technology**, v. 18, n. 1, p. 41--52, 1986.

KEYENCE. Sensor de Visão com IA integrada - Série IV2. **Keyence**, 2021. Disponível em: <<https://www.keyence.com.br/mykeyence/?ptn=002>>. Acesso em: 01 ago. 2022.

KHAN, S. et al. **A Guide to Convolutional Neural Networks for Computer Vision**. California: Morgan & Claypool publishers, v. SYNTHESIS LECTURES ON COMPUTER VISION #15, 2018.

LANDIS, J. R.; KOCH, G. G. The Measurement of Observer Agreement for Categorical Data. **International Biometric Society**, 06 fev. 2020.

MATROX. Ferramentas da Biblioteca de Imagens Matrox (MIL). **Matrox**, 2022. Disponível em: <<https://www.matrox.com/en/imaging/products/software/sdk/mil/tools/classification>>. Acesso em: 01 ago. 2022.

MICROSOFT. VisionCatalog.ImageClassification Method. **docs.microsoft**. Disponível em: <<https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.visioncatalog.imageclassification?view=ml-dotnet>>. Acesso em: 06 ago. 2022.

MINUSSI, C. R. Redes Neurais: Introdução e Principais Conceitos, Ilha Solteira, Maio 2008.

MONTGOMERY, D. C. **Introdução ao controle estatístico da qualidade**. 7^a. ed. Porto Alegre: LTC, 2016.

OLIVEIRA, L. G. D. S. Utilização de Rede Neural Para Predição de Proteínas de Bactérias Secretadas Por Vias Não Clássicas, Uberlândia, 2008.

POLHEMUS, N. W. **Process Capability Analysis: Estimating Quality**. New York: CRC Press, 2017.

SACHAN, A. Detailed Guide to Understand and Implement ResNets. **CV-Tricks.com**. Disponível em: <<https://cv-tricks.com/keras/understand-implement-resnets/>>. Acesso em: 06 ago. 2022.

SAFDAR, D. S. et al. Process Capability Indices under Non-Normality Conditions using Johnson Systems. **International Journal of Advanced Computer Science and Applications**, Nottingham, 2019. 292-294.

SILVA, J. M. N. Redes Neurais Artificiais: Rede Hopfield e Redes. **Universidade Federal Fluminense**, Niterói, jul. 2003.

VALDATI, A. D. B. **Inteligência artificial - IA**. 1^a. ed. Santa Catarina: Contentus, 2020.

ZHANG, A. et al. **Dive into Deep Learning**. Pequim: arXiv preprint arXiv:2106.11342, 2021.

ANEXOS

FIGURA 32 - RELATÓRIO DE CAPABILIDADE

Capabilidade de classificação de imagem

CCI-Classificador TCC - Treino contínuo-Rev00

Identificação

Nome:

Linha:

Estação:

Observações:

Elaborador: erick

Data: 07/04/2022

Resumo geral

Parâmetros

Limite de pontuação mínima 70%

Quantidade de épocas: 90

Taxa de aprendizagem: 0,005

Tamanho do lote: 3

Mecanismo: 14, 1&7

Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

Classes

1

F00LJ0100



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

2

F00LJ0100_INV



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

3

F00VC17105_INV



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

4

INCLINADA



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

5

SEM ANEL



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

6

VEDAÇÃO



Total de imagens

Classificações corretas

Classificações incorretas

Pontuações < 70%

Tempo médio de classificação

Pontuação mínima

Pontuação média

Pontuação máxima

Desvio padrão

Probabilidade de pontuações < 70%

Capabilidade de processo CP

FONTE: O autor (2022).