

# Deep Learning aplicado a inspeção visual da presença de um componente de conjunto de eixo

Lucas Ferreira Luchi e André Gustavo Adami

## Resumo

A evolução dos processos industriais, guiada pelos conceitos de fábrica inteligente da Indústria 4.0, e a necessidade de tornar as tarefas de tomada de decisão cada vez menos dependentes de humanos deve demandar cada vez mais a aplicação industrial do aprendizado de máquinas. Nesse sentido, esse trabalho propõe a utilização de aprendizado profundo para a identificação da presença ou falta de um anel de retenção montado na ponta de um eixo veicular a partir de imagens. Uma rede neural convolucional foi utilizada para aprender as características das imagens e realizar a classificação. O sistema foi avaliado utilizando uma base de imagens coletada em um ambiente real de uma empresa. Apesar do desbalanceamento do conjunto de dados, o método produziu resultados máximos em sensibilidade, especificidade e F1-score. Além disso, a arquitetura da rede foi otimizada (redução de 90% do número de parâmetros) a fim de aumentar a eficiência computacional.

## Palavras-chave

Aprendizado de máquina. Aprendizado profundo. Inteligência Artificial. Controle de Qualidade. Indústria 4.0. Manufatura Inteligente. Redes neurais convolucionais.

# Deep Learning applied to visual inspection of the presence of an axis assembly component

## Abstract

The evolution of industrial processes based on the concepts of smart factory in Industry 4.0 and the need to perform decision-making tasks less human-dependent should increasingly demand the industrial application of machine learning. In this sense, this work proposes the use deep learning to identify the presence or lack of a retaining ring at a vehicle axis end from images. A convolutional neural network was used to learn features from images e to perform classification. The system was evaluated using a dataset of images collected in a real industrial environment. Despite the dataset imbalance, the method yielded maximum results in sensitivity, specificity and F1-score. Thereafter, the neural network architecture was optimized (90% reduction of the number of parameters) to increase computational efficiency.

## Keywords

Machine Learning. Deep Learning. Artificial Intelligence. Quality Control. Industry 4.0. Smart Manufacturing. Convolutional Neural Networks.

## I. INTRODUÇÃO

Atualmente, tem-se observado uma crescente utilização da tecnologia no nosso dia a dia. Na indústria, é dito que a quarta revolução industrial ou Indústria 4.0 vai nos levar ao próximo nível de manufatura onde as máquinas irão se redefinir na forma como se comunicam e executam funções individuais [1]. Nesse contexto, muitas empresas recorrem à automação para superar desafios como problemas de qualidade, eficiência e custos [2]. Quando se trata da detecção de problemas de qualidade, tem sido cada vez mais comum o uso

de tecnologias que envolvem processamento de imagens e Inteligência Artificial (*Artificial Intelligence* - AI) [2][3].

A Indústria 4.0 pode ser vista como uma questão de difusão e adoção de tecnologia [4] que busca melhorar a indústria a fim de lidar com desafios globais [5]. Uma das formas de obter sucesso nesse novo ambiente industrial mais versátil é utilizar algoritmos avançados, incorporando AI nos processos de produção [6]. Um dos ramos da AI, o aprendizado de máquina (*Machine Learning* - ML) vem assumindo um importante papel na evolução da indústria [7]. A utilização de ML vem crescendo graças ao desenvolvimento de novos

Pós graduação em Engenharia 4.0 – Universidade de Caxias do Sul (UCS)

E-mails: lucas.luchi@gmail.com; agadami@ucs.br

Data de envio: 06/09/2020

Data de aceite: 17/10/2020

<http://dx.doi.org/10.18226/23185279.v8iss2p135>

algoritmos para resolver uma série de problemas relacionados ao uso intensivo de dados, como serviços ao consumidor, diagnóstico de falhas em sistemas complexos e controle de cadeias logísticas [8].

Entre os diversos algoritmos para ML, as redes neurais tem emergido novamente como uma das técnicas utilizadas para lidar com as complexidades dos padrões de forma similar ao sistema neural humano [7][9]. Um dos métodos que utilizam redes neurais artificiais para o aprendizado, o *Deep Learning* (DL) tem sido cada vez mais utilizado em aplicações envolvendo reconhecimento de objetos em imagens [10], pois demanda menor interferência manual no processo de aprendizado. A relação entre DL, ML e AI é ilustrada na Fig. 1.

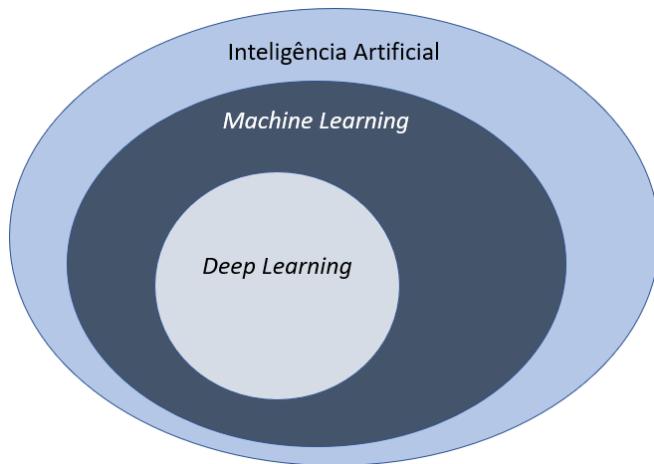


Fig. 1: *Deep Learning* no contexto da Inteligência Artificial, adaptado de [9][11]

Na indústria, uma das utilizações de ML (e de DL) que mostra grande potencial é o monitoramento da qualidade dos produtos, onde a técnica pode ser associada a sistemas de visão para o reconhecimento de padrões [12] [13]. Alguns exemplos desse tipo de aplicação são: um método proposto para detecção automática da presença de componentes em placas de circuito impresso e bombas de injeção utilizando a rede neural *Multi-Layer Perceptron* (MLP) [14]; método para detecção de componentes (*o'ring*) utilizando o classificador *Gaussian Mixture Models* (GMM) [3]; detecção de grampos de fixação em uma peça estampada utilizando um classificador *neuro-fuzzy* e um classificador baseado em limites [15]; e detecção de componentes de transformadores elétricos utilizando um modelo baseado em *Faster Region-based Convolutional Neural Network* [16]. O uso dessas tecnologias está em rápida expansão, muitas vezes criando novas formulações de problemas impulsionados por aplicações práticas [8], e devem fazer a Indústria 4.0 mais sofisticada [17].

Na aplicação que é objeto de estudo desse artigo, vários componentes são montados manualmente na ponta de um eixo veicular. O anel de retenção é o componente mais crítico, que ajuda a fixar os demais componentes para evitar que o rodado se solte na utilização do eixo. A Fig. 2 mostra imagens da ponta do eixo com anel (conforme apontado pela seta vermelha) e sem anel. Após a inspeção visual, é montada uma tampa na ponta do eixo, e não é mais possível verificar se o anel de retenção foi montado. O objetivo deste trabalho é utilizar um método de DL na inspeção para identificação da

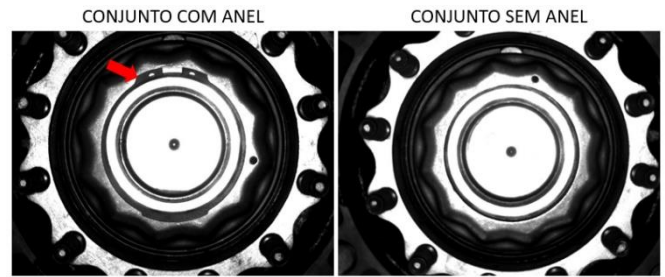


Fig. 2: Ponta do eixo com e sem anel de retenção

## II. REFERENCIAL TEÓRICO

### A. Indústria 4.0

A primeira revolução industrial iniciou-se na Inglaterra no século XVIII e teve como principais inovações máquinas modernas que substituíram as ferramentas manuais, a utilização do vapor como fonte de energia e a obtenção de novas matérias-primas [18]. O advento da energia elétrica na segunda metade do século XIX desencadeou a segunda revolução industrial [19], que também foi marcada pela introdução do aço barato, motor a combustão e a administração científica do trabalho visualizada na difusão das linhas de montagem [20]. A partir dos anos 70, uma nova onda de inovações tecnológicas, como a micro eletrônica e novos processos de telecomunicações, foi incorporada ao processo econômico e produtivo e os equipamentos passaram a processar, distribuir e transmitir informações através de redes no que ficou conhecido como terceira revolução industrial [21]. Ao relacionar o uso de novas tecnologias como os sistemas cyber-físicos e a internet das coisas com uma quarta revolução industrial, surge em 2011 na Alemanha na feira de Hannover o termo “Indústria 4.0” [22].

Desde que surgiu, o tema Indústria 4.0 vem ganhando destaque na indústria e na academia, sendo considerado como algo de extremo impacto no futuro da manufatura [1] [22]. Além da Alemanha, outros países como China, Estados Unidos, França e Brasil apresentaram planos de desenvolvimento industrial cientes dessa nova era da manufatura [22] [23] [4].

Apesar de toda atenção dada ao tema, não há um consenso sobre uma definição clara do que é a Indústria 4.0 [23]. No entanto, há a noção de que se trata de uma fusão entre o mundo virtual e real com ênfase em aplicações de engenharia como a robótica, digitalização e automatização [1]. A Indústria 4.0 pode ser descrita como um termo coletivo para conceitos e tecnologias da organização da cadeia de valor [23] que busca construir fábricas inteligentes através da integração de objetos físicos com tecnologias digitais [4], ou ainda, transformar a manufatura máquina-dominante em manufatura digital [24].

Alguns dos componentes ou técnicas chave da Indústria 4.0 são sistemas em nuvem ou computação em nuvem, comunicação máquina-a-máquina, simulação e realidade aumentada, *data mining*, *enterprise resource planning* e *business intelligence*, manufatura virtual, *big data analytics* e tecnologia da informação e comunicação [24] [25]. Entretanto em uma revisão de literatura foram também identificados os sistemas cyber-físicos (*Cyber-Physical Systems – CPS*), a

internet das coisas (*Internet of Things* – IoT), a internet dos serviços e a fábrica inteligente como componentes básicos da Indústria 4.0 [23].

### B. Fábrica inteligente

A manufatura inteligente busca aproveitar as tecnologias avançadas de manufatura e informação a fim de possibilitar a flexibilidade dos processos físicos, melhorando o *design*, produção, administração, e integração de todo o ciclo de um produto típico [25]. Isso é essencial para atender as demandas de lotes pequenos e customizadas dos clientes [5] e para operar em um mercado altamente dinâmico e global [24].

Espera-se que a Indústria 4.0 integre verticalmente os subsistemas hierárquicos para transferir a fábrica tradicional para o altamente flexível e reconfigurável sistema de manufatura que é a fábrica inteligente. A fábrica inteligente também é uma base chave para suportar outros dois tipos de integração, a integração horizontal pela cadeia de valor e a integração de engenharia digital de ponta a ponta [5]. Uma visão futura da fábrica inteligente descreve uma fábrica onde não somente todos os recursos de manufatura (sensores, atuadores, máquinas, robôs, esteiras, etc.) são conectados e trocam informação automaticamente, mas também a fábrica se tornará consciente e inteligente o suficiente para prever e manter as máquinas; para controlar o processo de produção; e para administrar o sistema fabril. Além disso, muitos processos de manufatura como o design de produto, planejamento de produção, engenharia de produção, de produtos e serviços, serão simulados como modulares e então conectados de ponta a ponta, o que significa que esses processos não serão apenas comandados por um sistema descentralizado mas também controlados independentemente [22]. Nessa proposta, há pouca intervenção humana, e isso ocorre principalmente nas fases de resolução de problemas [24]. A fábrica inteligente pode ser definida ainda como um fábrica onde o CPS comunica através da IoT e auxilia pessoas e máquinas na execução de suas tarefas [23]. Para que isso seja possível, é necessária a utilização de tecnologias como a AI e o ML, que também são vistos como pilares para a Indústria 4.0 [26].

### C. Machine Learning

ML é um ramo da AI que permite que sistemas computacionais busquem melhorar automaticamente através da experiência [8]. Ou seja, o ML busca desenvolver métodos para os computadores melhorarem o seu desempenho em certas tarefas com base em dados observados [27][11].

Os métodos para o ML podem ser divididos em quatro categorias: aprendizado não supervisionado, aprendizado semi-supervisionado, aprendizado supervisionado e aprendizado por reforço [28] [7]. No aprendizado supervisionado, o modelo de ML é treinado em uma base de dados que consiste em pares de entradas e saídas, chamadas de características e rótulos, respectivamente [13]. Nesse formato, geralmente os dados de treinamento assumem a forma de uma coleção de pares  $(x, y)$  e o objetivo é produzir uma previsão  $y$  em resposta a uma consulta  $x$  [8]. O aprendizado não supervisionado descobre a estrutura oculta em dados não rotulados [29]. O aprendizado semi-supervisionado é uma abordagem intermediária entre o supervisionado e não supervisionado, que utiliza uma base de

dados com dados rotulados e sem rótulos [30]. No aprendizado por reforço, um agente aprende o mapeamento ideal de situações para ações por meio de uma busca de tentativa e erro guiada por um sinal de recompensa a ser maximizado [31].

O aprendizado supervisionado é a forma mais comum de ML [10] [8]. Classificadores de spam em e-mails, reconhecimento facial através de imagens e sistemas de diagnóstico médico para pacientes são exemplos de aplicações de aprendizado supervisionado [8]. Na indústria, particularmente, exemplos de aplicações bem-sucedidas são classificação de falhas, desenvolvimento de sensores flexíveis e modelagem e previsões de regressão da qualidade do produto [7]. Sistemas de inspeção visual com ML geralmente utilizam algoritmos de aprendizado supervisionado [2].

Para executar o aprendizado supervisionado, existem muitas formas diferentes de mapear a função que relaciona os dados de entrada com os de saída, incluindo árvores de decisão, florestas de decisão, regressão logística, máquinas de vetores de suporte, redes neurais artificiais, aprendizado multi-instância, máquinas de *kernel* e classificadores bayesianos [8] [32] [2].

### D. Reconhecimento de padrões de imagem com técnicas clássicas de Machine Learning

O reconhecimento de padrões é um processo de categorizar qualquer amostra de dados medidos ou observados como membro de uma de diversas classes ou categorias [33]. Em outras palavras, é a categorização de dados de entrada em classes identificáveis, através da extração de características ou atributos significativos dos dados de um fundo de detalhes irrelevantes [34]. É um campo que se preocupa com a descoberta automática de regularidades nos dados por meio do uso de algoritmos de computador e com o uso dessas regularidades para executar ações como classificar os dados em diferentes categorias [28].

Nas aplicações com imagens, como sistemas de visão automatizados, esse processo de análise pode ser dividido em quatro etapas básicas (Fig. 3): aquisição de imagens, processamento de imagens (filtros ou alterações para pré-processamento e localização da região de interesse), extração de características e classificação [15] [35]. Contudo, é importante ressaltar que, dependendo da aplicação, os processos e técnicas envolvidos podem variar [2].

Etapas comuns na detecção em uma inspeção com sistema de visão de máquina, o pré-processamento de imagens [36] tem por objetivo transformar a imagem a fim de manter as informações relevantes para o problema em questão. Nessa etapa podem ser utilizadas técnicas como a distribuição de pico do histograma (HPD), operação morfológica [36], filtro esquerdo bidimensional (2DLF) [37], supressão de ruído por *gaussian smoothing*, filtros medianos, filtros de modo, entre outros [38]. Outros métodos de processamento de imagens podem ser aplicados, como por exemplo a segmentação. O objetivo da segmentação é particionar uma imagem em regiões com uma aparência visual homogênea razoável, ou que corresponda a objetos ou partes de objetos [28] [39]. Entre as técnicas de segmentação estão: buscar a região de interesse e detectar anomalias usando limiar [37] e redes neurais artificiais como a arquitetura U-Net [40].

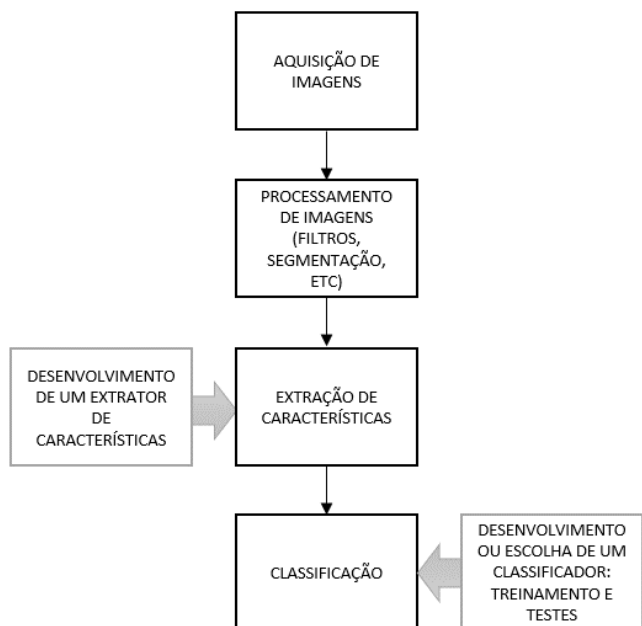


Fig. 3: Etapas básicas de reconhecimento de padrões de imagem

A extração de características constitui o cálculo de medidas específicas, que caracterizam o sinal [41] a fim de que as classes possam ser discriminadas. Essas características devem ser descritas numericamente, como por exemplo médias estatísticas, coeficientes de correlação, valores e vetores de matrizes de covariância [34]. Além disso, devem preservar informações úteis para distinguir os dados entre as classes [28]. Para extração podem ser usadas técnicas como o cálculo fracionário e diferencial Gaussiano [36], modelos de cores como HSI e RGB e histograma [42], análise de componentes principais [43].

Para a classificação de imagens, podem ser utilizados diversos modelos e algoritmos [38]. Uma das técnicas é o GMM, que agrupa dados através de estimativas de densidade de probabilidade. O GMM foi utilizado para classificar imagens de suportes com e sem a presença de um componente (o'ring) [3]. Outro classificador é a Máquina de Vetor Suporte (*Support Vector Machine* – SVM), o qual busca construir um hiperplano ou conjunto de hiperplanos, com base no qual diferentes tipos de amostras de dados podem ser separados [7]. O SVM foi utilizado para classificar imagens de suportes de rodas de trem de carga com e sem o componente chave de bloqueio de vagão [44]. Árvore de decisão é uma ferramenta de suporte à decisão que usa um modelo semelhante a uma árvore para descrever relações entre diferentes variáveis e tomar decisões [7]. Essa técnica foi utilizada para classificação de amostras de chapas sem defeitos ou com dois tipos de defeitos superficiais [41]. O aprendizado de múltipla instância busca relacionar ao seu respectivo rótulo um conjunto de instâncias caracterizado por um vetor de atributos, e foi utilizado para classificar imagens buscando detectar a presença de um parafuso montado em CPUs [2].

#### E. Reconhecimento de padrões de imagem com técnicas de Deep Learning

As redes neurais artificiais (*Artificial Neural Networks* – ANNs) são uma família de modelos inspirada nas redes neurais biológicas, que são apresentadas como sistemas de “neurônios” interconectados que trocam mensagens entre si

*SCIENTIA CUM INDUSTRIA*, V. 8, N. 2, PP. 135 – 144, 2020 [7] [45]. Tipicamente essas redes tem uma série de camadas (Fig. 4), que podem ser descritas como: camada de entrada, alimentada com os dados; camada de saída, que retorna as informações calculadas; e camadas intermediárias ou ocultas, que não têm saídas observáveis [46].

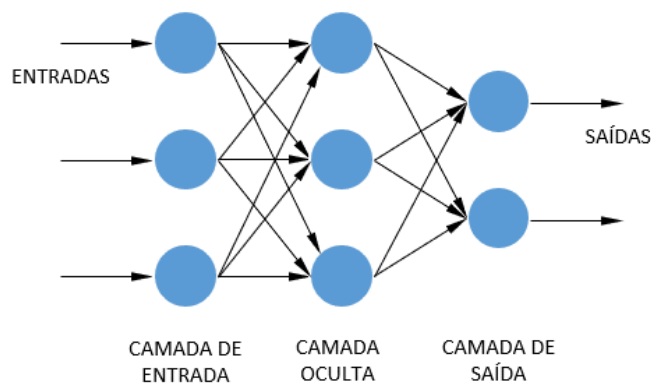


Fig. 4: Estrutura de uma rede neural de três camadas

De forma simplificada, um modelo de rede neural é uma função não linear de um conjunto de variáveis de entrada para um conjunto de variáveis de saída controladas por um vetor de parâmetros (pesos) ajustáveis [28]. O ajuste nos pesos da rede é feito após o cálculo da função custo, que mede o erro, ou seja, quanto a saída está distante do esperado [28][47]. O algoritmo de aprendizado utiliza um método chamado de *backpropagation* que permite que a informação de custo retorne através da rede e calcule o gradiente de erro para os pesos [11][28]. Um método como o gradiente descendente estocástico é usado para o aprendizado utilizando esse gradiente [11][39].

No DL, normalmente são utilizadas ANNs com múltiplas camadas, e embora alguns autores considerem DL redes com mais de três camadas [9], não há um consenso sobre a profundidade do modelo [11]. Entretanto, o DL pode ser considerado o estudo de modelos que envolvem uma quantidade maior de composição de funções aprendidas ou conceitos aprendidos do que o ML tradicional [11]. Esses modelos realizam o chamado aprendizado de características, que permite que um sistema seja alimentado com dados brutos e descubra automaticamente as características necessárias para detecção ou classificação [10]. Dessa forma, pode-se dizer que o DL é um tipo específico de ML que alcança grande poder e flexibilidade ao aprender a representar o mundo como uma hierarquia de conceitos, com cada conceito definido em relação a conceitos mais simples e características mais abstratas calculadas em relação a outras menos abstratas [11].

As redes neurais que se enquadram no conceito de DL são chamadas de *deep neural networks* (DNNs) [9]. O reconhecimento de imagens com essas redes tem a vantagem de não distinguir a segmentação, extração de características, e classificação, aprendendo das imagens as características, diferente da abordagem clássica de reconhecimento de padrões, onde há uma engenharia de características projetada por humanos [17] [48] [10]. Além disso, é possível que o DL produza um melhor desempenho de classificação, pois muitas vezes as técnicas clássicas de limiar são insuficientes para segmentar defeitos de fundo se não houver um ambiente controlado e com iluminação estável [49]. Entretanto, o aprendizado da DNN depende da disponibilidade de grande

quantidade de dados de treinamento e demanda certa carga de processamento computacional [47] [7].

Um tipo de DNN são as redes neurais convolucionais (*Convolutional Neural Network - CNN*), que são projetadas para processar dados no formato de múltiplas matrizes, como por exemplo sinais e sequências de uma dimensão, imagens ou espectrogramas de áudio em 2D, vídeo ou imagens volumétricas em 3D [10]. As CNNs usadas para classificação de imagens são formadas por duas partes: começam com uma série de camadas de convolução e *pooling* e terminam com um classificador densamente conectado [47]. A Fig.5 mostra uma representação desse tipo de rede.

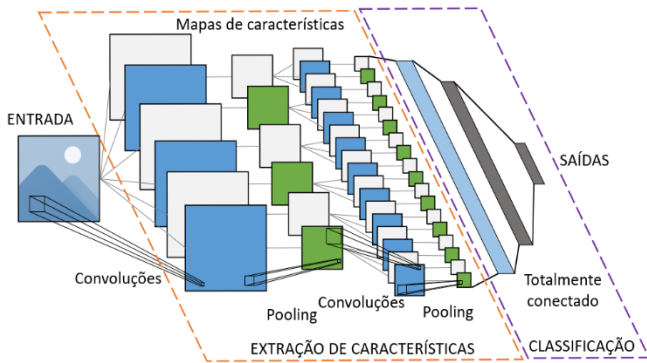


Fig. 5: Representação de uma CNN

Nas camadas de convolução de uma CNN, os neurônios são organizados em planos, chamados de mapas de características [28]. Nessas camadas, os neurônios são conectados somente a uma pequena sub-região (campo receptor) da imagem ou do mapa de características da camada anterior, de onde podem extrair características visuais como por exemplo bordas e cantos [28] [10] [46]. Em cada mapa de características, todos os neurônios são restritos a compartilhar os mesmos valores de pesos ou banco de filtros [28] [10]. Dessa forma, são extraídos da imagem ou do mapa de características de entrada diferentes fragmentos aos quais é aplicada uma mesma transformação, gerando um mapa de características de saída, onde cada localização corresponde à mesma localização no mapa de características de entrada [47]. Uma camada de convolução completa é composta por vários mapas de características, e em cada mapa é utilizado um banco de filtros diferente [46] [10]. Matematicamente, a operação de filtro executada por um mapa de características é uma convolução discreta, daí o nome da rede [10]. Como essa operação é espacialmente invariante, permite que as CNNs reconheçam padrões aprendidos em qualquer local na imagem, enquanto as redes neurais comuns teriam que aprender novamente caso o padrão aparecesse em um novo local [38] [47].

Para cada mapa de características, na camada de convolução, há um plano de unidades na camada de *pooling* e cada unidade recebe entradas de um pequeno campo receptivo no mapa de características correspondente da camada de convolução [28]. Nessa camada, de forma similar à camada de convolução, são extraídas janelas dos mapas de características de entrada, mas aqui a transformação reduz a resolução dos mapas de características, por exemplo, tendo como saída somente o valor máximo de cada canal (*max-pooling*) [47]. Isso reduz a precisão da localização de cada uma das diferentes características, mas, para identificar o padrão, é mais relevante a posição relativa em relação a outras

características e informações sobre a posição exata podem ser prejudiciais [46].

A classificação ocorre nas camadas finais da CNN, que normalmente são totalmente ou densamente conectadas (densas) e adaptáveis, como as utilizadas nas redes neurais comuns [28] [49] [47].

### III. MATERIAL E MÉTODO

O objetivo nesse trabalho é utilizar um método de DL para identificação da presença ou falta do anel de retenção no conjunto de eixo. Uma CNN (Fig. 6) foi utilizada para realizar a extração e classificação de imagens brutas de conjuntos de eixo.

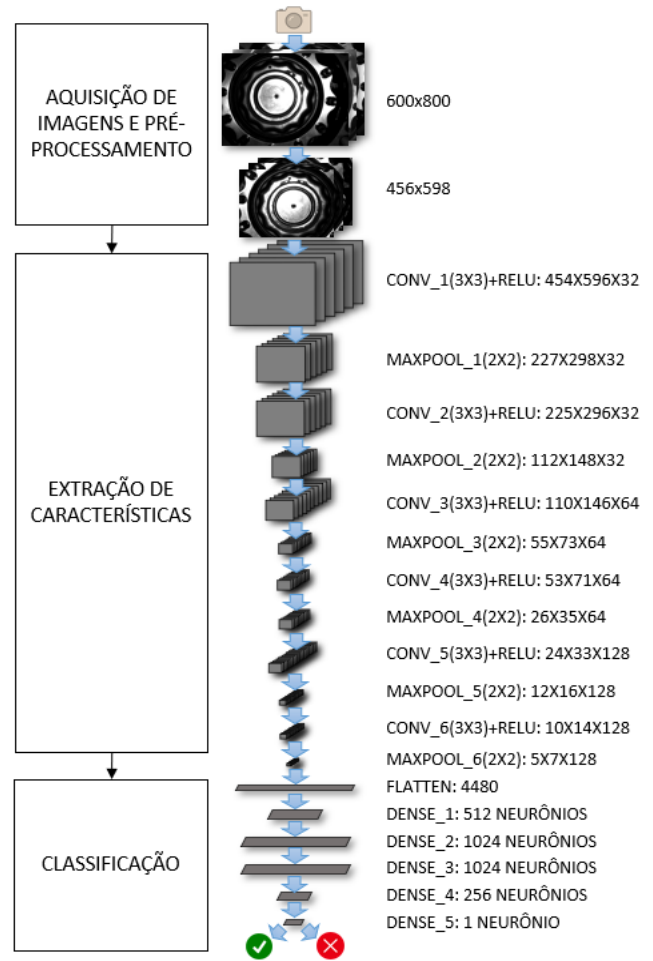


Fig. 6: Método e arquitetura inicial propostos

#### A. Aquisição de imagens

A etapa de aquisição de imagens foi realizada por duas câmeras (uma para cada ponta do eixo) de um sistema de visão comercial instalado em um ponto de inspeção na fábrica onde os conjuntos são montados. Nesse sistema o eixo é posicionado manualmente até uma trava mecânica de indexação; são posicionadas manualmente proteções para diminuir a influência da iluminação externa; e as câmeras são acionadas. Em cada câmera um flash é disparado automaticamente no momento da captura da imagem. As imagens capturadas tem escala de cinza e tamanho de 800 x 600 pixels. A trava de indexação do sistema permite uma variação na posição do eixo, portanto há um pequeno deslocamento na posição do conjunto em relação à câmera no momento da captura da imagem. Como as proteções para o

controle da iluminação são posicionadas manualmente, podem haver também variações na iluminação. A Fig. 7 mostra imagens que exemplificam essas variações. É possível observar que há variação de um conjunto para outro na posição radial do anel montado.

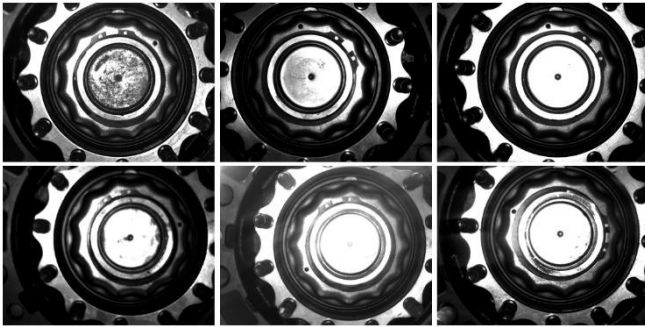


Fig. 7: Exemplos de imagens de conjuntos de eixo

### B. Pré-processamento

Considerando que a região de interesse é a parte central das imagens, as bordas das imagens foram removidas. Para isso, foram selecionadas aleatoriamente diversas imagens da base com o anel de retenção não centralizado na imagem. Nessas imagens, foi verificado manualmente o quanto de cada borda da imagem poderia ser cortada sem afetar o conjunto. Para garantir que parte de nenhum conjunto fosse removida por estar deslocado do centro mais do que foi estimado com as imagens aleatórias, cada borda foi reduzida em torno de 10% do tamanho estimado. Ao final, todas as imagens foram tiveram as suas dimensões alteradas para 598 x 456 pixels.

Para facilitar o aprendizado da rede neural, todos os pixels foram normalizados para o intervalo [0;1]. Esta normalização facilita a convergência de redes neurais [47].

### C. Extração de características

Como projetar bons extratores de características requer considerável perícia e habilidades de engenharia [10], neste trabalho foi optado por utilizar uma CNN para a extração de características das imagens.

A extração de características na CNN é realizada pelas camadas de convolução alternadas com as camadas de *max-pooling*. As camadas de convolução utilizaram núcleo de 3x3. Esse é o menor núcleo de convolução que pode perceber informações espaciais locais e é capaz de obter bom desempenho através da regularização, além de reduzir o armazenamento da rede se comparado a núcleos maiores [50] [51]. A função de ativação utilizada nas camadas de convolução foi a unidade linear retificada (*Rectified Linear Unit – ReLU*), que é a mais utilizada em CNNs por proporcionar rápido aprendizado [9]. As camadas de *max-pooling* (onde o valor máximo de cada canal é extraído) utilizaram janelas 2x2.

A parte convolucional da rede foi construída do princípio, com a inclusão de camadas com aumento progressivo do número de mapas de características e redução do tamanho dos mapas, que é um padrão comum na maioria das CNNs [47]. A quantidade de camadas foi definida buscando reduzir consideravelmente o número de parâmetros a ser processado na etapa de classificação. Quanto maior o número de camadas de convolução e *pooling*, menor a quantidade de neurônios e o tamanho dos mapas de características. Foram adicionadas 6

### D. Classificação

Para a etapa de classificação, foram adicionadas na rede uma camada para planificar, que transforma para uma dimensão as informações da última camada de *max-pooling*, seguida de camadas densas. Foram utilizadas 5 camadas densas (com 512, 1024, 1024, 256 e 1 neurônios, respectivamente) com base na arquitetura utilizada para classificar imagens de doenças em plantas, que obteve um bom desempenho com uma base de dados sem balanceamento [52]. A função de ativação utilizada nas camadas densas foi a ReLU, exceto pela camada final que utilizou a função de ativação *sigmoide*. Tal função é uma escolha comum para problemas de classificação binários, e retorna valores no intervalo [0;1] que podem ser interpretados como uma probabilidade para a amostra pertencer a uma classe ou outra [47]. A classificação é feita separando esses valores calculados para cada amostra acima ou abaixo do limite de 0,5.

### E. Avaliação de desempenho

Como o problema de classificação é binário, o desempenho dos modelos pode ser avaliado utilizando a matriz de confusão (Tabela 1) [53]. As linhas da matriz correspondem à classe, de fato, da amostra, enquanto as colunas correspondem à classe predita pelo classificador. No problema de duas classes, a matriz é composta por quatro medidas: Verdadeiro Positivo (VP), que se refere ao número de amostras positivas classificadas corretamente; Verdadeiro Negativo (VN), que se refere ao número de amostras negativas classificadas corretamente; Falso Positivo (FP), que se refere ao número de amostras positivas classificadas incorretamente como negativas; e Falso Negativo (FN), que se refere ao número de amostras negativas classificadas incorretamente como positivas [54]. Neste trabalho amostras positivas são imagens do conjunto de eixo sem o anel e negativas são as imagens do conjunto de eixo completo.

Tabela 1 - Matriz de confusão

	Valor classificado positivo	Valor classificado negativo
Valor real positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Valor real negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

A partir da matriz de confusão, é possível extrair métricas de desempenho de classificação. A mais comum é a acurácia, que mede a razão entre o número de amostras classificadas corretamente e o número total de amostras. Entretanto, ela não é uma métrica adequada quando não há balanceamento de amostras entre as classes [53]. A sensibilidade, *recall* ou taxa de positivo verdadeiro, é uma métrica que mede a porcentagem do grupo positivo que foi corretamente classificada [53], e é dada por

$$Sensibilidade = \frac{PV}{PV + FN}$$

A especificidade mede a porcentagem de amostras da classe negativa classificadas corretamente [53], e é dada por

$$Especificidade = \frac{NV}{NV + FP}$$

A precisão mede a porcentagem de amostras classificadas como positivas que realmente são positivas [53] e é dada por

$$Precisão = \frac{PV}{PV + FP}$$

O F1-score é uma métrica que relaciona a sensibilidade e a precisão e um único valor [53] e é dado por

$$F1\text{-score} = \frac{2 \times Precisão \times Sensibilidade}{Precisão + Sensibilidade}$$

Diversos trabalhos utilizaram o F1-score, precisão e sensibilidade como métricas na classificação de imagens com classes desbalanceadas [55] [56] [52] [57]. A sensibilidade é mais penalizada pela classificação incorreta de uma amostra como negativo, o que nesse caso é mais crítico, pois significa aprovar um conjunto sem o anel de retenção. A especificidade e o F1-score podem ser utilizados para complementar a análise de desempenho. Portanto, optou-se por utilizar como métricas neste trabalho a sensibilidade, a especificidade e o F1-score.

#### IV. EXPERIMENTOS E RESULTADOS

A CNN foi desenvolvida em linguagem Python, na plataforma de computação em nuvem Google Colaboratory<sup>1</sup> com ambiente de execução *graphics processing unit* (GPU). Foram utilizadas as bibliotecas abertas de DL Keras e TensorFlow [58]. A biblioteca Scikit-Learn foi utilizada para avaliação dos modelos.

##### A. Treinamento

A base de imagens foi dividida em três conjuntos: 60% para treinamento, 20% para validação e 20% para teste, mantendo a proporção entre as classes. A Tabela 2 mostra a quantidade de imagens por conjunto para cada classe. A divisão da base em nesses três conjuntos é comum em outros problemas de classificação com DL [49] [51].

Tabela 2 – Quantidade de imagens por conjunto (treinamento, validação e teste) para cada classe

Anel	Treinamento	Validação	Teste	Total
Presente	6.068	2.022	2.022	<b>10.112</b>
Em falta	106	36	36	<b>178</b>

Uma preocupação durante o treinamento foi evitar que ocorresse o *overfitting* para os dados de treinamento, que é quando o modelo se ajusta tão bem aos dados de treinamento que não consegue lidar com as variações existentes em novos dados [38]. Isso pode ocorrer principalmente quando a rede tem mais parâmetros que o necessário [38]. É possível

verificar se isso está ocorrendo ao comparar o desempenho da classificação nos dados de treinamento e de validação [47]. Além disso, o conjunto de validação permite ajustes do modelo ao longo do treinamento sem que informações dos dados de teste sejam transferidas indiretamente para o modelo, o que poderia ocasionar um *overfitting* para os dados de teste [47].

Buscou-se utilizar o maior número de imagens por lote possível para ser processado durante o treinamento, a fim de reduzir o risco de lotes sem amostras da classe minoritária, o que não permitiria o aprendizado do modelo nesses lotes. O tamanho de lote utilizado foi de 63 imagens. Ao longo da pesquisa foram encontrados trabalhos que utilizaram de 15 a 1999 ciclos de treinamento [49] [48] [51] [55] [52] [59] [17]. Definiu-se inicialmente realizar o treinamento de 20 ciclos, o que poderia ser ajustado conforme o desempenho.

Para o controle de desempenho do modelo durante o treinamento, foi utilizada a função de custo *crossentropy* (CE) binária, que é considerada a melhor escolha em problemas de classificação binários [49] [47]. Essa função mede a distância entre a probabilidade prevista pelo modelo e a real. Sua fórmula é dada por

$$CE = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

onde  $y$  é a classe e  $p(y)$  a probabilidade prevista para cada amostra  $N$ . Foram avaliados os métodos de otimização gradiente descendente estocástico (*stochastic gradient descent* – SGD), Adam [60] e RMSprop [61] (sendo que os últimos dois são variantes do SGD).

##### B. Resultados da Classificação

As redes treinadas utilizando os métodos Adam e RMSprop produziram os melhores resultados, conforme é mostrado nas matrizes de confusão (Tabela 3) e nas métricas de avaliação de desempenho calculadas (Fig. 8). O pior resultado foi obtido para a rede treinada utilizando SGD (Sensibilidade de 0%). Nesta rede, todas as amostras foram classificadas como imagens que possuíam o anel de retenção. Visto que essa é a classe que possui mais amostras no treinamento, o desbalanceamento das classes afetou o aprendizado da rede. Assim, os métodos Adam e RMSprop foram selecionados para os demais experimentos e não foi necessário aplicar nenhuma técnica de compensação de desbalanceamento.

Tabela 3 - Matrizes de confusão com diferentes métodos de otimização

		SGD classificado		Adam classificado		RMSprop classificado	
		VP	FN	VP	FN	VP	FN
real	Presente	0	36	36	0	36	0
	Em falta	0	2022	0	2022	0	2022
		FP	VN	FP	VN	FP	VN

<sup>1</sup> <https://colab.research.google.com/notebooks/intro.ipynb>.

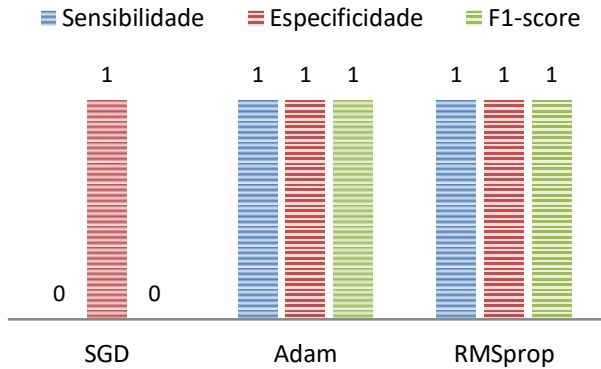


Fig. 8: Resultados com diferentes métodos de otimização

C. Redução do número de parâmetros do modelo

Para reduzir a complexidade e consequentemente aumentar a eficiência da rede, experimentos foram realizados reduzindo o número de parâmetros da arquitetura. Como o método de otimização da rede Adam apresentou o menor tempo de processamento (1.247 segundos contra 1.855 segundos do RMSprop), optou-se por utilizar esse método nos experimentos.

O número de camadas densas foi reduzido de cinco para três. Além disso, foram realizados experimentos com diferentes números de neurônios nas duas primeiras camadas densas (a última camada só possui um neurônio): 8, 16, 32, 64 e 128 neurônios. As demais configurações do modelo não foram alteradas. A Tabela 4 mostra o desempenho das diferentes configurações das camadas densas.

Tabela 4 – Desempenho para diferentes configurações das camadas densas antes da camada final

Configuração das camadas densas	Sensibilidade	Especificidade	F1-score
512+1024(2x)+256	1	1	1
128(2x)	1	1	1
64(2x)	1	1	1
32(2x)	1	1	1
16(2x)	0,972	1	0,986
8(2x)	1,000	0,999	0,960

Os resultados mostram que as arquiteturas com mais de 16 neurônios nas duas primeiras camadas densas produziram os mesmos resultados que a arquitetura original. Assim, pode-se concluir que a arquitetura estava superdimensionada. Apesar da redução do desempenho nas arquiteturas com 16 ou menos neurônios nas camadas densas, estes resultados mostram uma redução não significativa.

Com a redução do número de neurônios nas camadas densas, o número de parâmetros foi significativamente reduzido. A Fig. 9 mostra o número de parâmetros por configuração da arquitetura da rede. Comparando o número de parâmetros da arquitetura original (4.418.849) com a arquitetura com 32 neurônios em cada uma das duas camadas densas (431.489), a redução foi de aproximadamente 90%.

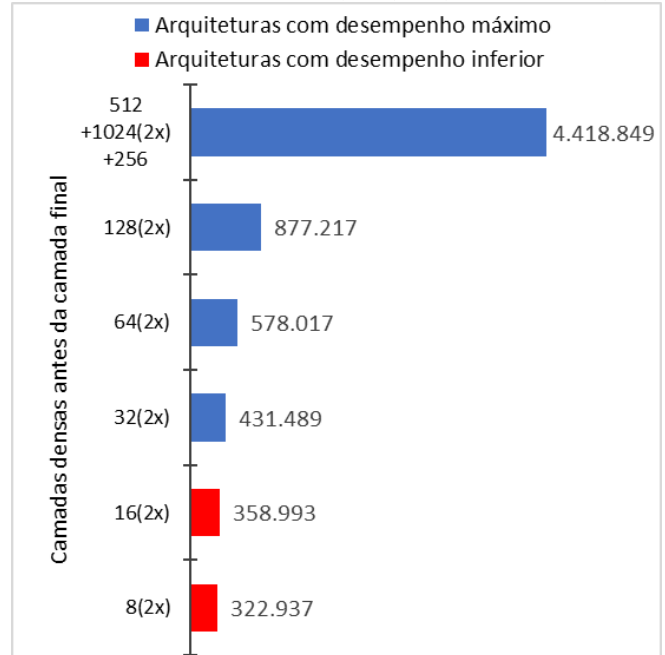


Fig. 9: Número de parâmetros nas diferentes arquiteturas avaliadas

V. CONCLUSÕES

A evolução dos processos industriais guiada pelos conceitos de fábrica inteligente da Indústria 4.0 e a necessidade de tornar as tarefas de tomada de decisão cada vez menos dependentes de humanos deve demandar cada vez mais a aplicação industrial de ML. Nesse sentido, esse trabalho apresenta um método que permite realizar de forma automática uma inspeção visual dos componentes montados na ponta de um eixo veicular a fim de detectar a falta de um anel de retenção.

O método utiliza uma CNN que recebe como entrada imagens brutas do eixo veicular. A rede neural realiza a extração de características e a classificação. Para realizar o treinamento da rede (adaptação dos pesos) foram avaliados três métodos de otimização: SGD, Adam e RMSprop. O processo de treinamento e avaliação das arquiteturas utilizou uma base de dados de imagens reais coletadas em uma empresa. Os melhores resultados (detectou corretamente todas as amostras com e sem o anel de retenção) foram obtidos utilizando os métodos de otimização Adam e RMSprop.

Apesar do desbalanceamento das classes (somente 1,7% das amostras pertencem a classe sem o anel de retenção), o método proposto obteve o desempenho máximo. O método de otimização SGD foi o único que não obteve o desempenho máximo. Conforme esperado, todas as amostras da classe minoritária foram classificadas como pertencessem à classe majoritária.

Em seguida, a arquitetura da rede foi otimizada (redução do número de parâmetros) a fim de aumentar a sua eficiência. Foram realizados diversos experimentos com alterações nas camadas densas. Os resultados mostraram que é possível atingir o desempenho máximo reduzindo em até 90% do número de parâmetros da rede (somente 2 camadas densas de 32 neurônios e a de saída com 1 neurônio).

Tratando-se de inspeção visual, a automação utilizando DL ao invés de soluções comerciais padronizadas de sistema de visão tem o potencial de reduzir o investimento necessário



para a solução do problema. O estudo apresentado nesse artigo contribui trazendo um método que pode servir como base para esse tipo de aplicação de DL.

Para trabalhos futuros, sugere-se incluir a identificação automática da região de interesse nas imagens ou de um modelo que utilize as imagens sem a remoção de bordas. Dentre essas duas alternativas, cabe ponderar a que terá menor custo computacional.

## VI. BIBLIOGRAFIA

- [1] P. K. Muhuri, A. K. Shukla and A. Abraham, "Industry 4.0: A bibliometric analysis and detailed overview," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 218-235, 2019.
- [2] N. Rahmatov, A. Paul, F. Saeed, W.-H. Hong, H. Seo and J. Kim, "Machine learning-based automated image processing for quality management in industrial Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 15, no 10, pp. 1-12, 2019.
- [3] V. Chauhan and B. Surgenor, "Fault detection and classification in automated assembly machines using machine vision," *The International Journal of Advanced Manufacturing Technology*, vol. 90, no 9-12, pp. 2491-2512, 2017.
- [4] L. S. Dalenogare, G. B. Benitez, N. F. Ayala and A. G. Frank, "The expected contribution of Industry 4.0 technologies for industrial performance," *International Journal of Production Economics*, vol. 204, pp. 383-394, 2018.
- [5] S. Wang, J. Wan, D. Li and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," *International Journal of Distributed Sensor Networks*, vol. 12, no 1, p. 3159805, 2016.
- [6] J. A. Carvajal Soto, F. Tavakolizadeh and D. Gyulai, "An online machine learning framework for early detection of product failures in an Industry 4.0 context," *International Journal of Computer Integrated Manufacturing*, vol. 32, no 4-5, pp. 452-465, 2019.
- [7] Z. Ge, Z. Song, S. X. Ding and B. Huang, "Data Mining and Analytics in the Process Industry: The Role of Machine Learning," *IEEE Access*, vol. 5, pp. 20590-20616, 2017.
- [8] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives and prospects," *Science*, vol. 349, no 6245, pp. 255-260, 2015.
- [9] V. Sze, Y. H. Chen, T. J. Yang and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no 12, pp. 2295-2329, 2017.
- [10] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, vol. 521, no 7553, pp. 436-444, 2015.
- [11] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, Cambridge: MIT press, 2016.
- [12] D. Dechow, "The engine in engineering: David Dechow at Integro Technologies, discusses vision system integration, a discipline that remains as strong as ever as new imaging technologies become more readily available.," *Imaging and Machine Vision Europe*, pp. 34-36, 2018.
- [13] A. Mayr, D. Kibkalt, M. Meiners, B. Lutz, F. Schäfler, R. Seidel, A. Selmaier, J. Fuchs, M. Metzner, A. Blank and J. Franke, "Machine Learning in Production - Potentials, Challenges and Exemplary Applications," *Procedia CIRP*, vol. 86, pp. 49-54, 2019.
- [14] G. Acciani, G. Brunetti, E. Chiarantoni and G. Fornarelli, "An Automatic Method to Detect Missing Components in Manufactured Products," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2324-2330, 2005.
- [15] J. Killing, B. W. Surgenor and C. K. Mechefske, "A machine vision system for the detection of missing fasteners on steel stampings," *The International Journal of Advanced Manufacturing Technology*, vol. 41, no 7-8, pp. 808-819, 2009.
- [16] Z. Liu and H. Wang, "Automatic Detection of Transformer Components in Inspection Images Based on Improved Faster R-CNN," *Energies*, vol. 11, no 12, p. 3496, 2018.
- [17] C.-C. Huang and X.-P. Lin, "Study on Machine Learning Based Intelligent Defect Detection System," *MATEC Web of Conferences*, vol. 201, pp. 1-10, 2018.
- [18] L. B. Canêdo, *A revolução industrial*, São Paulo: Atual, 1994.
- [19] A. M. Nicolaci-da-Costa, "Revoluções tecnológicas e transformações subjetivas," *Psicologia: teoria e pesquisa*, vol. 18, pp. 193-202, 2002.
- [20] A. T. Rauen, "Ciência, Tecnologia e Economia: Características frente à primeira e segunda Revoluções Industriais," *Espaço Acadêmico*, pp. 1-3, 2006.
- [21] M. F. F. Júnior, "A terceira revolução industrial e o novo paradigma produtivo: algumas considerações sobre o desenvolvimento industrial brasileiro nos anos 90," *Revista da FAE*, vol. 3, nº 2, 2000.
- [22] J. Qin, Y. Liu and R. Grosvenor, "A Categorical Framework of Manufacturing for Industry 4.0 and Beyond," *Procedia CIRP*, vol. 52, pp. 173-178, 2016.
- [23] M. Hermann, T. Pentek and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," *2016 49th Hawaii international conference on system sciences (HICSS)*, pp. 3928-3937, 2016.
- [24] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, no 1, pp. 127-182, 2020.
- [25] R. Y. Zhong, X. Xu, E. Klotz and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no 5, pp. 616-630, 2017.
- [26] S. J. Qin and L. H. Chiang, "Advances and opportunities in machine learning for process data analytics," *Computers and Chemical Engineering*, vol. 126, pp. 465-473, 2019.
- [27] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no 7553, pp. 452-459, 2015.
- [28] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.
- [29] S. Zerdoumi, A. Q. M. Sabri, A. Kamsin, I. A. T. Hashem, A. Gani, S. Hakak, M. A. Al-garadi and V. Chang, "Image pattern recognition in big data: taxonomy and open challenges: survey," *Multimedia Tools and Applications*, vol. 77, no 8, pp. 10091-10121, 2018.
- [30] O. Chapelle, B. Scholkopf and A. Zien, "Semi-supervised Learning," *IEEE Transactions on Neural Networks*, vol. 20, no 3, p. 542, 2006.
- [31] R. Nian, J. Liu and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Computers & Chemical Engineering*, vol. 139, p. 106886, 2020.
- [32] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.
- [33] S.-T. Bow, *Pattern Recognition and Image Preprocessing*, Boca Raton: CRC press, 2002.
- [34] J. T. Tou and R. C. Gonzalez, *Pattern recognition principles*, Massachusetts: Addison-Wesley Publishing Company, 1974.
- [35] J. Sun and Q. Sun, "Further development of adaptable automated visual inspection—part I: concept and scheme," *The International Journal of Advanced Manufacturing Technology*, vol. 81, no 5-8, pp. 1067-1076, 2015.
- [36] H. Liu, C. Zhang and D. Huang, "Extreme Learning Machine and Moving Least Square Regression Based Solar Panel Vision Inspection," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1-10, 2017.
- [37] A. A. Moghaddam and L. Rangarajan, "Classification of Welding Defects in Radiographic Images," *Pattern Recognition and Image Analysis*, vol. 26, no 1, pp. 54-60, 2016.
- [38] R. E. Davies, *Computer and machine vision: theory, algorithms, practicalities*, Academic Press, 2012.
- [39] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*, Prentice Hall Professional Technical Reference, 2003.
- [40] R. S. Pahwa, J. Chao, J. Paul, Y. Li, M. T. L. Nwe, S. Xie, A. James, A. Ambikapathi, Z. Zeng and V. R. Chandrasekhar, "FaultNet: Faulty Rail-Valves Detection using Deep Learning and Computer Vision," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 559-566, 2019.
- [41] S. Ravikumar, K. I. Ramachandran and V. Sugumaran, "Machine learning approach for automated visual inspection of machine components," *Expert Systems with Applications*, vol. 38, no 4, pp. 3260-3266, 2011.
- [42] P. Malaca, L. F. Rocha, D. D. M. Gomes, J. Silva and G. Veiga, "Online inspection system based on machine learning techniques: real case study of fabric textures classification for the automotive industry," *Journal of Intelligent Manufacturing*, vol. 30, no 1, pp. 351-361, 2019.
- [43] J. Sun and Q. Sun, "Further development of adaptable automated visual inspection—part II: implementation and evaluation," *The International Journal of Advanced Manufacturing Technology*, vol. 81, no 5-8, pp. 1077-1096, 2015.
- [44] L. Liu, F. Zhou and Y. He, "Vision-based fault inspection of small mechanical components for train safety," *IET Intelligent Transport Systems*, vol. 10, no 2, pp. 130-139, 2016.
- [45] C. M. Bishop, *Neural Networks for pattern recognition*, Oxford university press, 1995.
- [46] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no 11, pp. 2278-2324, 1998.
- [47] F. Chollet, *Deep learning with Phyton*, Manning publications, 2017.
- [48] H.-H. Kim, J.-K. Park, J.-H. Oh and D. J. Kang, "Multi-task Convolutional Neural Network System for License Plate Recognition," *International Journal of Control, Automation and Systems*, vol. 15, no 6, pp. 2942-2949, 2017.
- [49] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Meré, M. Buchwitz and W. Wellbrock, "Deep Learning for Industrial Computer Vision

- quality Control in the printing Industry 4.0,” *Sensors*, vol. 19, no 18, p. 3987, 2019.
- [50] J. Wang, P. Fu and R. X. Gao, “Machine vision intelligence for product defect inspection based on deep learning and Hough transform,” *Journal of Manufacturing Systems*, vol. 51, pp. 52-60, 2019.
- [51] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1506*, 2015.
- [52] G. Sambasivam and G. D. Opiyo, “A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks,” *Egyptian Informatics Journal*, 2020.
- [53] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *Journal of Big Data*, vol. 6, no 1, p. 27, 2019.
- [54] J. Han, J. Pei and M. Kamber, *Data mining: Concepts and techniques*, Elsevier, 2011.
- [55] W. Ding, D.-Y. Huang, Z. Chen, X. Yu and W. Lin, “Facial action recognition using very deep networks for highly imbalanced class distribution,” *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1368-1372, 2017.
- [56] H. Hamledari, B. McCabe and S. Davari, “Automated computer vision-based detection of components of under-construction indoor partitions,” *Automation in Construction*, vol. 74, pp. 78-92, 2017.
- [57] Y. Wu, D. Guo, H. Liu and Y. Huang, “An end-to-end learning method for industrial defect detection,” *Assembly Automation*, vol. 40, pp. 31-39, 2019.
- [58] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, “TensorFlow: A System for Large-Scale Machine Learning,” *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265-283, 2016.
- [59] Y. Wu, D. Guo, H. Liu and Y. Huang, “An end-to-end learning method for industrial defect detection,” *Assembly Automation*, pp. 31-39, 2019.
- [60] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations (ICLR)*, 2015.
- [61] Y. Dauphin, H. De Vries, Y. Bengio, “RMSprop and equilibrated adaptive learning rates for non-convex optimization,” *Advances in neural information processing systems*, pp. 1504-1512, 2015.