ALGORITMOS DE APRENDIZAGEM PROFUNDA PARA DETECÇÃO DE OBJETOS DEFEITUOSOS PRODUZIDOS POR MANUFATURA ADITIVA

LUCAS MACEDO DA SILVA –luckasmacedo2@gmail.com
UNIVERSIDADE FEDERAL DE GOIÁS - UFG

SYMONE G. SOARES ALCALÁ - symone@ufg.br
UNIVERSIDADE FEDERAL DE GOIÁS - UFG

TALLES MARCELO G. DE A. BARBOSA - talles@pucgoias.edu.br
PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS – PUC GOIÁS

RUI ARAÚJO - rui@isr.uc.pt
UNIVERSIDADE DE COIMBRA, INSTITUTO DE SISTEMAS E ROBÓTICA (ISR-UC), DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES (DEEC-UC), PÓLO II, 3030-290 COIMBRA, PORTUGAL

ÁREA: *10. ENGENHARIA DO PRODUTO*
SUBÁREA: *10.1 - PROCESSO DE DESENVOLVIMENTO DO PRODUTO*

RESUMO: *A MANUFATURA ADITIVA É UMA TECNOLOGIA APLICADA NA INDÚSTRIA QUE PERMITE A CRIAÇÃO DE OBJETOS PERSONALIZADOS A BAIXO CUSTO. CONTUDO, É UM PROCESSO SUSCETÍVEL A ERROS QUE CAUSAM DEFEITOS E IMPACTAM NEGATIVAMENTE NO PRODUTO FINAL. A INSPEÇÃO DOS DEFEITOS É UMA ETAPA IMPORTANTE DO PROCESSO DE FABRICAÇÃO PARA EVITAR PERDAS FINANCEIRAS. ELA GERALMENTE É REALIZADA POR HUMANOS, MAS É UM PROCESSO LENTO, CANSATIVO E SUSCETÍVEL A ERROS. MODELOS DE APRENDIZAGEM PROFUNDA PODEM SER EMPREGADOS PARA ESSE FIM, POIS SÃO CONFIÁVEIS E MENOS SUSCETÍVEIS A ERROS. COM ISSO, O PRESENTE TRABALHO APRESENTA UM PROCESSO PARA A DETECÇÃO AUTOMÁTICA DE DEFEITOS EM OBJETOS MANUFATURADAS ADITIVAMENTE, A PARTIR DA APLICAÇÃO DE DUAS REDES NEURAIS DE APRENDIZAGEM PROFUNDA. A PRIMEIRA É UMA FASTER RCNN E REALIZA A DETECÇÃO DOS OBJETOS. A SEGUNDA É UMA CNN QUE REALIZA DE DETECÇÃO DE DEFEITOS. AS DUAS SÃO MESCLADAS PARA REALIZAR AS TAREFAS DE DETECÇÃO DE OBJETOS E DE DETECÇÃO DE DEFEITOS. A SEPARAÇÃO DAS TAREFAS EM DUAS REDES, PERMITE MAIOR FLEXIBILIDADE NO TREINO E NO TESTE DE CADA MODELO PARA TANTO, UM NOVO DATASET DE IMAGENS DE OBJETOS FOI CONSTRUÍDO, SENDO COMPOSTO POR PEÇAS PARA A MANUTENÇÃO DE ELEVADORES. NOS RESULTADOS EXPERIMENTAIS, FORAM OBTIDOS ERRO MAP DE 0,91 E 0,96 DE IOU PARA A FASTER RCNN NA DETECÇÃO DE OBJETOS, E ACURÁCIA DE 0,77 PARA A CNN NA DETECÇÃO DOS DEFEITOS.*

PALAVRAS-CHAVES: *DETECÇÃO DE DEFEITOS; DETECÇÃO DE OBJETOS; MANUFATURA ADITIVA; APRENDIZAGEM PROFUNDA; FASTER RCNN; CNN.*

DEEP LEARNING ALGORITHMS FOR DETECTING DEFECTIVE OBJECTS PRODUCED BY ADDITIVE MANUFACTURING

ABSTRACT: *ADDITIVE MANUFACTURING IS A TECHNOLOGY APPLIED IN INDUSTRY THAT ALLOWS THE BUILDING OF PERSONALIZED OBJECTS AT LOW COST. HOWEVER, IT IS A PROCESS SUSCEPTIBLE TO ERRORS THAT CAUSE DEFECTS AND NEGATIVELY IMPACT THE FINAL PRODUCT. DEFECT INSPECTION IS AN IMPORTANT STEP IN THE MANUFACTURING PROCESS TO AVOID FINANCIAL LOSS. IT IS USUALLY CARRIED OUT BY HUMANS, BUT IT IS A SLOW, TIRING PROCESS AND HIGHLY SUSCEPTIBLE TO ERRORS. HOWEVER, DEEP LEARNING MODELS CAN BE EMPLOYED FOR THIS PURPOSE, BECAUSE THEY ARE HIGHLY RELIABLE AND LESS PRONE TO ERRORS. THEREFORE, THIS WORK PRESENTS A PROCESS FOR THE AUTOMATIC DETECTION OF DEFECTS IN ADDITIVELY MANUFACTURED OBJECTS USING TWO NEURAL NETWORKS WITH DEEP LEARNING. THE FIRST IS A RCNN FASTER AND PERFORMS OBJECT DETECTION. THE SECOND IS A CNN THAT PERFORMS DEFECT OBJECT DETECTION. THE TWO NETWORKS ARE MERGED TO PERFORM THE OBJECT DETECTION TASK AND OBJECT DEFECT DETECTION TASK. THE SEPARATION OF TASKS INTO TWO NETWORKS ALLOWS GREATER FLEXIBILITY IN TRAINING AND TESTING EACH MODEL, WHICH CAN BE CARRIED OUT IN PARALLEL. THEREFORE, A NEW DATASET WAS BUILT, AND IT IS COMPOSED OF MAINTENANCE PARTS OF ELEVATORS. IN THE EXPERIMENTAL RESULTS, MAP OF 0.91 AND 0.96 OF IOU WERE OBTAINED FOR THE FASTER RCNN IN THE DETECTION OF OBJECTS, AND ACCURACY OF 0.77 FOR THE CNN IN THE DEFECT DETECTION.*

KEYWORDS: *DEFECT DETECTION; OBJECT DETECTION; ADDITIVE MANUFACTURING; DEEP LEARNING; FASTER RCNN; CNN.*

1. INTRODUÇÃO

Indústrias de manufatura estão empregando cada vez mais tecnologia na linha de produção para otimizar a produção e consequentemente aumentar os lucros (RODRIGUES *et al.*, 2016). Dessa forma, investimentos em manufatura inteligente têm crescido rapidamente. Nessa abordagem os dados são integrados em tempo real e a produção é totalmente controlada por computador. Este conceito é denominado de Indústria 4.0 e seu princípio básico é o uso de tecnologias disruptivas para o controle dos processos de manufatura (VAIDYA; AMBAD; BHOSLE, 2018). Entre essas tecnologias destaca-se a Internet das Coisas (em inglês *Internet of Things* - IoT), *big data*, *cloud computing* e Manufatura Aditiva (MA).

A MA é uma tecnologia disruptiva que permite a criação de objetos a partir da sobreposição de camadas, realizada por uma impressora 3D (ISO - INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2015). A modelagem do objeto que será impresso é realizada empregando *softwares* de computação assistida (em inglês *Computer-Aided Design* - CAD), e é a primeira etapa do processo. Após, a impressora 3D recebe o modelo e imprime o objeto camada a camada. A MA é considerada uma tecnologia disruptiva, pois permite a criação de objetos customizados sob demanda (WANG *et al.*, 2020).

A construção de objetos customizados pela MA permite o desenvolvimento de diversas áreas do conhecimento, como, por exemplo, a construção civil, a prototipagem e biomedicina (NGO *et al.*, 2018). Além disso, a MA permite criar objetos com formas complexas, reduz os gastos com energia, e ajuda na redução da emissão de gases do efeito estufa. Contudo, o processo é suscetível a falhas que podem consumir matéria-prima desnecessariamente, gerar custos e reduz os lucros e a competividade da indústria (WANG *et al.*, 2017). A presença de defeitos nos objetos dificulta a adoção da MA na indústria em geral (CHEN *et al.*, 2021). Portanto, ao final do processo de manufatura geralmente é realizado um processo de detecção de defeitos para evitar que produtos defeituosos cheguem ao consumidor.

Este processo geralmente é realizado por um humano especialista que inspeciona visualmente cada objeto e identifica erros ou irregularidades. Contudo, é um processo lento, cansativo e altamente suscetível a erros. Por outro lado, sistemas de detecção automática de defeitos podem rapidamente e de forma acurada detectar os defeitos, melhorando assim, a qualidade dos produtos e reduzindo os custos de produção. Neste caso, um modelo de

aprendizagem profunda pode ser aplicado para realizar essa tarefa (KIM; MOYLAN, 2018). Esses modelos são eficientes, menos suscetíveis a erros e podem operar de forma independente.

A detecção automática de defeitos empregando modelos de aprendizagem profunda identifica e localiza defeitos e irregularidades nos objetos. Estes modelos devem realizar duas operações na imagem: localizar os objetos presentes na imagem (ou seja, a tarefa de detecção de objetos); e determinar se o objeto é defeituoso ou não (ou seja, a tarefa de detecção de defeitos). A detecção de objetos é um problema complexo que envolve a localização do objeto na imagem e a classificação do objeto em uma das classes de objetos ao qual o modelo foi treinado (YIN; LI; FU, 2020). Com isso, uma vez que os objetos forem detectados, o modelo pode classificar cada objeto em defeituoso ou não defeituoso, que é um problema de classificação binária (BISHOP, 2016).

Os modelos de detecção de objetos podem ser categorizados em detectores de um estágio e detectores de dois estágios, dependendo de quantos passos o modelo leva para encontrar o objeto na imagem. Os detectores de um estágio podem detectar objetos e determinar a localização das caixas delimitadoras (em inglês *bounding boxes*) em uma única passagem pela imagem e são geralmente mais rápidos do que os detectores de dois estágios, mas menos precisos (YANG *et al.*, 2021). Exemplos desses modelos são YOLO (*You Only Look Once*) e SSD (*Single Shot Detector*). Por outro lado, os detectores de dois estágios primeiro geram as propostas de regiões e as refinam, para então detectar os objetos (YANG *et al.*, 2021). Exemplos de modelos de dois estágios incluem a RCNN (*Region-based Convolutional Neural Network*) e Faster R-CNN (*Faster Region-based Convolutional Neural Network*).

Normalmente, um único modelo detecta os objetos na imagem e também define ele é defeituoso ou não (WANG *et al.*, 2017) (YANG *et al.*, 2021). Nesta abordagem, um modelo precisa ser treinado com duas classes diferentes para o mesmo objeto (classe defeituosa e classe não defeituosa). Por outro lado, outra abordagem é configurar um processo que separe a inspeção em duas tarefas distintas: modelo de detecção de objetos que localiza e define a classe do objeto e modelo de detecção de defeitos que define se o objeto tem um defeito. Portanto, são dois modelos distintos, cada um responsável por uma tarefa. Esta abordagem permite treinar e testar ambos os modelos separadamente e em paralelo, bem como permite a integração de outros modelos para realizar outras tarefas e classificações.

Os modelos de aprendizado profundo requerem um processo de treinamento para ajustar seus parâmetros e um processo de teste para verificar seu desempenho. Esses dois

processos usam conjuntos de dados de imagens (em inglês *datasets*), que permitem o teste de hipóteses, teorias e aplicações e ajudam a continuar o progresso dos modelos de aprendizado profundo (DOLLAR *et al.*, 2009). No entanto, os *datasets* devem conter imagens dos objetos no cenário de aplicação, e anotações das informações (por exemplo, tipo de objeto e localização do objeto) presentes nas imagens são necessárias para permitir seu uso no modelo. No entanto, na literatura, existem poucos conjuntos de dados anotados com objetos manufaturados aditivamente disponíveis para o uso público. Na literatura, existem diversos trabalhos que treinaram modelos de aprendizagem profunda em *datasets* para realizar a detecção de objetos e também de defeitos.

O trabalho Chen *et al.* (2017) apresentou uma melhoria de um modelo RCNN para resolver o problema de detecção de objetos pequenos, onde os objetos são menores que 30 centímetros. Um gerador de pequenas regiões foi utilizado para lidar com pequenos objetos. O modelo RCNN proposto teve uma melhoria de 29,8% em termos de precisão quando comparado ao modelo RCNN padrão. Mas os autores não se preocuparam com outras informações do conjunto de dados, como se o objeto é defeituoso, tipos de defeitos e etc. No entanto, o presente trabalho não se limita à detecção de objetos, mas também detecta essas outras informações do objeto.

Por outro lado, An, Wang e Liu (2020) melhoraram um modelo FAST-RCNN de arquitetura RESNET-101 para detectar defeitos em superfícies texturizadas. O trabalho foi testado no conjunto de dados de benchmark TILDA (ou seja, conjunto de dados de defeitos de tecido Tianchi). Como resultado, o trabalho alcançou uma precisão média de 94,66% no conjunto de dados e cerca de 4,35% superior ao modelo FAST-CNN padrão. A melhoria no modelo permitiu que ele realizasse a detecção de defeitos. No entanto, o trabalho emprega apenas um modelo para realizar duas tarefas diferentes, de modo que são necessários mais parâmetros para que o modelo aprenda as informações do conjunto de dados - o que faz com que o modelo necessite de mais tempo e recursos computacionais para ser treinado. O presente trabalho visa dividir as responsabilidades treinando um modelo específico para cada atividade, permitindo assim o treinamento de modelos menores e especialistas.

Portanto, o objetivo deste trabalho é investigar a aplicação de modelos de aprendizado profundo para a detecção automática de peças sobressalentes produzidas por MA e seus defeitos. A aplicação consiste em separar a responsabilidade das tarefas em dois modelos diferentes, onde um modelo (FASTER RCNN) é responsável por detectar objetos e outro modelo (CNN) é responsável por detectar defeitos. Ao final, os resultados de ambos os modelos são mesclados, gerando diversas informações (como, se o objeto é defeituoso, tipo

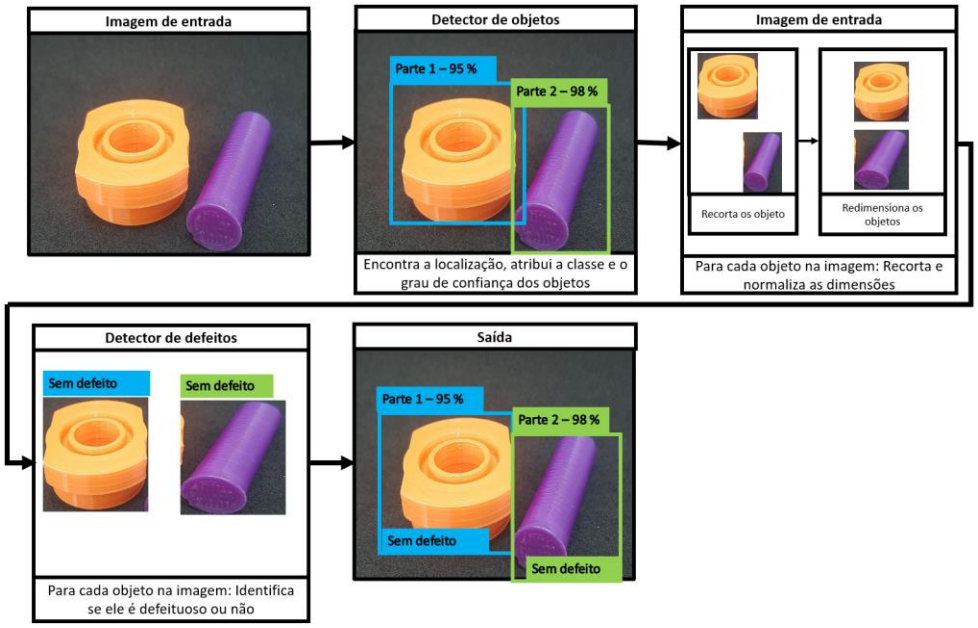
de defeito, nível de defeito, etc.) sobre os objetos presentes nas imagens. A abordagem proposta é avaliada e testada usando um novo conjunto de dados de objetos de peças de sobressalentes produzidas por MA, sendo que essas peças são utilizadas para manutenção de elevadores. O modelo FASTER RCNN obteve 0,91 de mAP e 0,96 de IoU; enquanto o modelo CNN obteve 0,77 de precisão. Portanto, os resultados demonstram as capacidades promissoras da abordagem e dos modelos propostos.

2. PROPOSTA DA ABORDAGEM PARA A DETECÇÃO DE OBJETOS E DEFEITOS

Esta seção descreve a abordagem proposta para detecção de objetos e defeitos. Como descrito anteriormente, ela mescla dois modelos distintos de aprendizado profundo, cada um realizando uma tarefa específica (ou seja, detecção de objetos e detecção de defeitos). Para isso, os modelos foram treinados e testados de forma independente e depois mesclados. O *dataset* foi coletado e anotado pelos autores, e sua descrição completa é fornecida na Subseção 3.1.

Na abordagem proposta, o modelo detector de objetos é responsável por detectar objetos em imagens, enquanto o modelo detector de defeitos identifica se o objeto é defeituoso ou não. A Figura 1 mostra a abordagem completa.

FIGURA 1 - Diagrama da abordagem proposta com dois modelos (ou seja, detector de objetos e detector de defeitos) e suas responsabilidades.



Fonte: Autoral (2023).

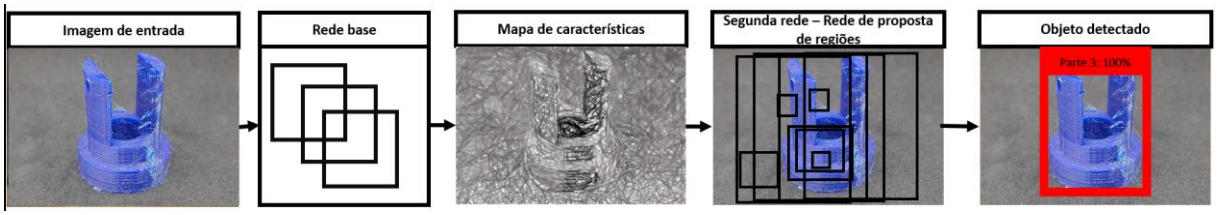
Primeiro, uma imagem de entrada passa pelo modelo detector de objetos (FASTER-RCNN) que localiza os objetos na imagem e define a classe do objeto. A partir da localização do objeto definida por uma caixa delimitadora, é possível recortar cada objeto da imagem e analisá-lo separadamente. O algoritmo de pré-processamento é responsável por esta tarefa. Ou seja, ele recorta cada objeto de imagem com base em sua caixa delimitadora e o redimensiona para dimensões de 128x128 pixels. Cada objeto detectado é definido como uma entrada para o modelo detector de defeitos (CNN), que realiza uma classificação binária definindo se o objeto é defeituoso (com defeito) ou não defeituoso (sem defeito). Finalmente, informações sobre a classe do objeto, sua localização na imagem (caixa delimitadora), o nível de confiança do detector de objetos em sua previsão e se o objeto possui defeitos são gerados.

Assim, no final de todo o processo, é possível obter informações relevantes do objeto, nomeadamente: localização dos objetos - posição de cada objeto na imagem, descrita em caixas retangulares, denominadas de caixas delimitadoras; classe - a categoria a qual o objeto pertence; confiança - nível de confiança que o modelo possui para determinar se o objeto pertence à classe à qual foi classificado; e defeituoso - determina se o objeto é defeituoso ou não.

2.1 Detector de objetos

O modelo escolhido para a detecção de objetos foi o modelo FASTER-RCNN por ser preciso, flexível e eficiente. Ele é composto por duas redes distintas. A rede base produz um mapa de características. A segunda rede, chamada Rede de Proposta de Regiões (em inglês *Region Proposal Network* - RPN), que usa um mapa de recursos para propor regiões potenciais que contêm um objeto. A técnica de supressão não máxima (em inglês *non-maximum supression*) é usada para determinar se a região proposta possui um objeto (LEMOS *et al.*, 2019). Nesta técnica, a melhor caixa delimitadora é encontrada avaliando cada proposta em um conjunto de caixas sobrepostas. A Figura 2 apresenta uma representação da arquitetura FASTER RCNN, adaptada de (LEMOS *et al.*, 2019).

FIGURA 2 – Representação da arquitetura da FASTER RCNN.



Fonte: Adaptado de Lemos *et al.* (2019).

Para o estudo atual, o modelo FASTER-RCNN foi implementado usando a arquitetura ResNet50. A arquitetura ResNet50 consiste em 50 camadas de Redes Neurais Convolucionais (CNNs) profundas, organizadas em 48 camadas de convolução, uma camada de *pool* máximo e uma camada de *pool* médio. A camada de *pool* máximo seleciona o elemento máximo do mapa de recursos, enquanto a camada de *pool* médio seleciona um valor de elemento do mapa de recursos. Apesar de conter mais de 23 milhões de parâmetros, a arquitetura ResNet50 é considerada compacta em comparação com outras arquiteturas existentes. Além disso, essa arquitetura tem demonstrado desempenho superior em aplicativos baseados em imagem (MANDAL; OKEUKWU; THEIS, 2021).

2.2 Algoritmo de pré-processamento dos objetos detectados

O algoritmo de pré-processamento corta cada objeto de imagem e o redimensiona. Esta etapa é necessária porque o detector de defeitos requer uma entrada de dados com um tamanho limitado. Como cada caixa delimitadora tem um tamanho diferente, o redimensionamento é necessário para que a rede de detecção de defeitos opere corretamente. A primeira etapa do algoritmo de pré-processamento é receber a caixa delimitadora do objeto e a imagem como entrada. Em seguida, cada objeto é cortado pela área definida pela caixa delimitadora. Depois, cada parte recortada é redimensionada para 128x128 *pixels* e retornada. O algoritmo é aplicado a cada objeto da imagem, podendo ser visualizado no Algoritmo 1.

Algoritmo 1: Algoritmo de pré-processamento dos objetos detectados

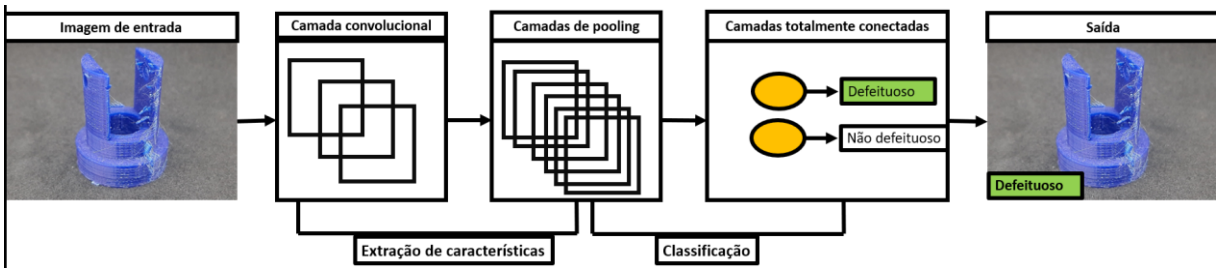
Entradas: I = Imagem contendo os objetos
Cxy = Posições que definem a caixa delimitadora do objeto atual
Saída: Op – Objeto processado
Ic ← Corte o objeto definido pela caixa delimitadora Cxy na imagem I
Op ← Redimensione o objeto Ic recortado para 128x128 pixels
Retorne: Op

2.3 Detector de defeitos

A detecção de defeitos é realizada por um modelo CNN, já que ele tem boa precisão, tolerância a variações de imagem e pode lidar com grandes conjuntos de dados. Ele é um tipo especial de rede neural *feed-forward* com o princípio de compartilhamento de peso (PATHAK; PANDEY; RAUTARAY, 2018), consistindo principalmente em três camadas: convolucional, *pooling* e totalmente conectada (VOULODIMOS *et al.*, 2018). As camadas convolucionais aplicam a operação de convolução, deslizando um *kernel* (matriz) na imagem e gerando os vetores de características. A segunda camada é o agrupamento de camadas que

realiza a redução dimensional das imagens, onde um mapa de recursos é reduzido a partir de uma função de agregação, como por exemplo, valores médios ou máximos. Em seguida, as regiões próximas no mapa de recursos são agrupadas em uma única unidade. Finalmente, as camadas totalmente conectadas são responsáveis por converter os vetores de recursos 2D em vetores 1D e gerar a classificação (VOULODIMOS *et al.*, 2018). A Figura 3 apresenta uma representação da arquitetura CNN (MANDAL; OKEUKWU; THEIS, 2021).

FIGURA 3 – Representação da arquitetura da CNN.



Fonte: Adaptado de Mandal, Okeukwu e Theis (2021).

3. RESULTADOS EXPERIMENTAIS

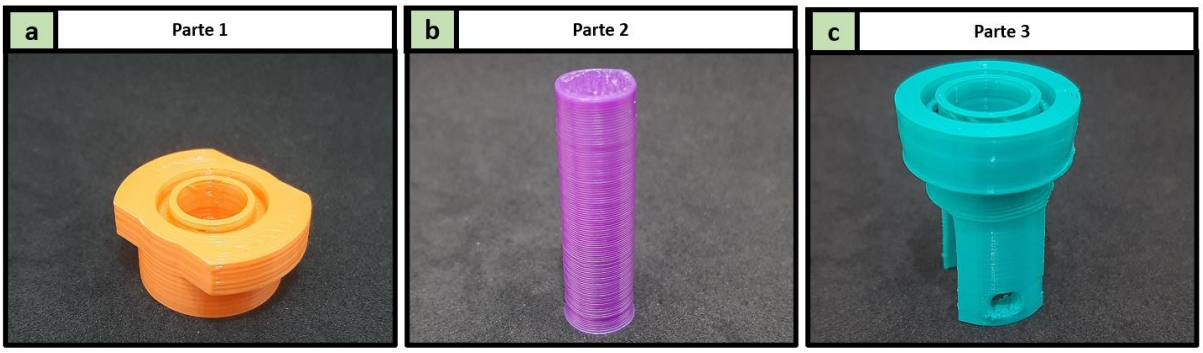
O processo proposto foi desenvolvido na Plataforma Google Colab, por ser um site de fácil configuração e não requer instalação. Além disso, possui uma unidade gráfica integrada (em inglês *Graphical Processing Unity* - GPU). O ambiente de plano gratuito consiste em 13 GB de RAM, 79 GB de disco rígido e uma GPU NVIDIA Tesla K80 com 12 GB de memória. Os modelos foram desenvolvidos usando a Interface de Programação de Aplicação (em inglês *Application Programming Interface* - API) TensorFlow *Object Detection* e a linguagem de programação Python 3.9.

3.1 Descrição do *dataset*

Para o desenvolvimento do *dataset*, um conjunto de imagens de objetos fabricados aditivamente (peças sobressalentes) foi criado para treinar e testar os modelos. Os objetos são fabricados de forma aditiva e foram definidos no projeto FASTEN (*Flexible and Autonomous Manufacturing Systems for Custom-Designed Products*) cujo objetivo é desenvolver sistemas inteligentes para projetar objetos manufaturados e personalizados a baixo custo. Os objetos pertencem a três classes distintas de objetos, chamados de partes, que são usados para construir um botão de elevador. Para montar um botão, são necessários um objeto de cada

classe, chamados de parte 1, parte 2 e parte 3. A Figura 4 apresenta exemplos de imagens de objetos e suas classes.

FIGURA 4 – Exemplos de objetos de cada classe.



Fonte: Autoral (2023).

O conjunto de dados é composto por 981 imagens, onde cada uma contém de 1 a 3 objetos, totalizando 1093 objetos. Destes, 436 (39,9%) não apresentam defeito e 657 (60,1%) apresentam algum tipo de defeito. A anotação dos dados foi realizada por meio da ferramenta VIA (VGG *Image Annotation*), na qual foram anotadas diversas informações do objeto, como cor, contorno e classe. Contudo, para o presente trabalho foram utilizadas as informações da localização do objeto e se ele é defeituoso ou não. As demais informações podem ser úteis para pesquisas futuras usando o conjunto de dados.

O detector de objetos recebe uma imagem completa para realizar a detecção. Por outro lado, o detector de defeitos recebe apenas o objeto. Este *dataset* foi dividido em outros dois *datasets*, um para o detector de objetos (denominado *dataset* 1) e outro para o detector de defeitos (denominado *dataset* 2). O *dataset* 1 contém as imagens completas e a separação em conjuntos de treinamento e de teste foi realizada por meio de seleção aleatória de imagens. A seleção aleatória de imagens consiste em um algoritmo que lê todas as imagens salvas e as divide aleatoriamente em dois conjuntos distintos (conjunto de treino e conjunto de teste). Para o *dataset* 2, as informações do objeto que está com defeito também são lidas e, em seguida, o algoritmo divide o *dataset* com base nessas informações. A proporção da divisão dos dois *datasets* (1 e 2) é de 70% para treinamento e 30% para teste. Dessa forma, o *dataset* 2 foi obtido cortando e redimensionando cada objeto da imagem. A Tabela 1 apresenta a distribuição dos objetos por classes e defeitos.

TABELA 1 – Distribuição dos objetos do dataset por classe e se possui ou não defeito.

Classe	Defeituoso	Total	Total por objeto
Parte 1	Não	181	350
	Sim	169	
Parte 2	Não	103	222
	Sim	119	
Parte 3	Não	152	521
	Sim	369	

Fonte: Autoral (2023).

3.2 Métricas de avaliação

Quatro métricas de desempenho foram usadas para avaliar os modelos. Interseção sobre união (em inglês *Intersection Over Union* - IoU) e precisão média (*Mean Avarage Precision* - mAP) foram empregadas para avaliar o detector de objetos. Por outro lado, a matriz de confusão e a precisão foram usadas para avaliar o detector de defeitos. A IoU avalia a precisão da caixa delimitadora prevista pelo modelo (área de sobreposição) e a caixa delimitadora anotada/real (a área de união). Ou seja, o IoU avalia a precisão da sobreposição de regiões entre as duas caixas delimitadoras. O resultado é um número entre 0 e 1, 0 indica que não há sobreposição, e sendo que valores próximos de 1 indicam melhor os resultados da previsão (ULLAH; ULLAH, 2020):

$$IoU = \frac{\text{Área de sobreposição}}{\text{Área de união}} \tag{1}$$

A métrica mAP avalia o grau de precisão e exatidão da técnica de detecção de objetos. Ela é contínua como a média das precisões médias para cada classe de objeto (ZHU *et al.*, 2020). Matematicamente, a métrica mAP é definida como:

$$mAP = \sum_{q=1}^{Q} \frac{\text{MédiaP}(q)}{Q}, \tag{2}$$

onde Q é o número de classes e MédiaP(q) é a precisão média para cada classe (AHMAD *et al.*, 2020). A precisão média resulta em um valor entre 0 e 1, também, valores próximos a 1 indicam bons resultados preditivos. O mAP é calculado considerando todas as classes e limites de IoU. Dessa forma, quanto melhor o IoU, melhor será o mAP, pois o objeto é detectado corretamente na imagem.

Sobre as métricas para avaliar a classificação do objeto, uma matriz de confusão permite avaliar um modelo de classificação binária. Ela avalia a previsão categórica do modelo em relação a falsos positivos (FP), falsos negativos (FN), verdadeiros positivos (VP)

e verdadeiros negativos (VN). VP indica que o modelo classificou corretamente um objeto com defeito; VN indica que o modelo classificou corretamente um objeto sem defeito; FP indica que o modelo classificou que o objeto é defeituoso, mas ele não possui defeitos; e FN indica que o objeto não possui defeito, mas ele é defeituoso. Nessa métrica, as linhas representam as classes verdadeiras dos objetos e as colunas as classes previstas pelo modelo (AHMAD *et al.*, 2020). Essa matriz fornece uma visão clara do desempenho do modelo, permitindo a avaliação de sua precisão e taxa de erro.

Por fim, a acurácia (A) avalia a proporção de predições verdadeiras/corretas (VP e VN) em relação ao total de predições realizadas (SCHNEIDER, 2018). Um resultado é um número entre 0 e 1, onde valores próximos a 1 indicam bons resultados preditivos:

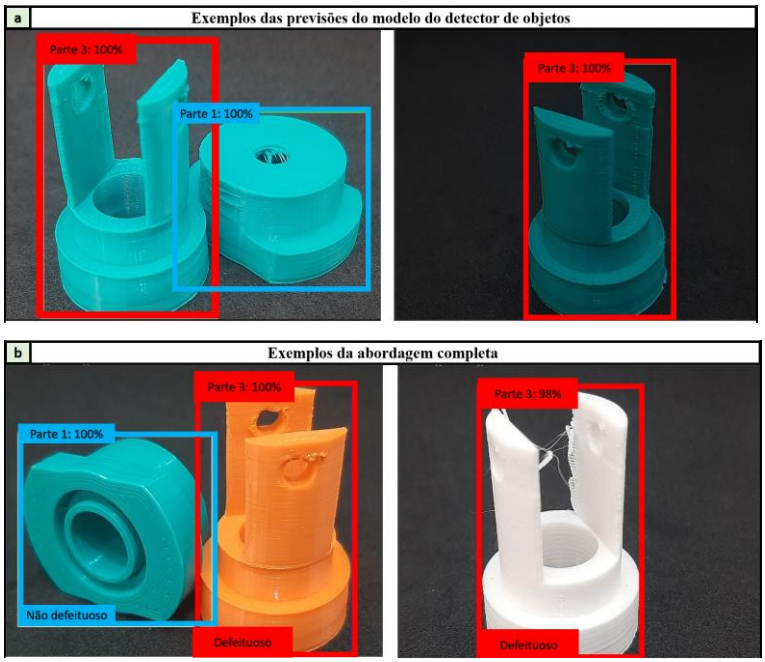
$$A = \frac{VP+VN}{VP+VN+FP+FN} \tag{3}$$

3.3 Resultados do detector de objetos (FASTER RCNN)

O modelo de detector de objetos foi construído com base no FASTER-RCNN com arquitetura ResNet50, previamente treinado com o conjunto de dados Microsoft Common Objects in Context (MS COCO) (LIN *et al.*, 2015). Ele foi treinado usando aprendizado de transferência por 25.000 épocas com o novo conjunto de dados deste trabalho. O treinamento durou cerca de 4 horas no ambiente Google Colab. Um total de 687 imagens foram usadas para treinamento e 294 para teste.

O modelo proposto alcançou 0,91 de mAP, o que indica que o modelo está detectando e localizando corretamente os objetos em grande proporção das imagens do conjunto de dados. Além disso, o modelo obteve 0,96 de IoU, ou seja, o modelo é capaz de detectar caixas delimitadoras corretas de objetos, detectando com precisão a localização do objeto na imagem. Nota-se que o modelo conseguiu operar bem, mesmo quando treinado com um pequeno conjunto de dados. Além disso, foi capaz de realizar corretamente a previsão de classe de objeto e determinar caixas delimitadoras. A Figura 5a apresenta exemplos de imagens de detecção de objetos realizadas com o modelo proposto.

FIGURA 5 – Exemplos de detecção de objetos com (a) o detector de objetos, (b) a abordagem



Fonte: Autoral (2023).

Como pode ser visto, ele foi capaz de detectar todos os objetos da imagem, mesmo imagens que possuem mais de um objeto. Além disso, o modelo é capaz de prever corretamente as classes e caixas delimitadoras dos objetos.

3.3 Resultados do detector de defeitos (CNN)

Um modelo CNN foi usado como detector de defeitos para realizar a classificação binária dos objetos. A rede é formada por 35 camadas, sendo uma camada de entrada com 128 neurônios, um neurônio para cada pixel da imagem, uma camada de saída com as duas possíveis classes e 33 camadas ocultas responsáveis por extrair as características. As camadas ocultas são formadas por blocos de convolução 2D, normalização de lotes (*batch normalization*), *pooling* máximo e *dropout*. A função de ativação utilizada é a sigmoide que permite a classificação em duas possíveis classes, defeituoso ou não defeituoso. A rede possui ao total 2.233.889 parâmetros, sendo 2.231.585 parâmetros treináveis e 2.304 parâmetros não treináveis.

As entradas são objetos cortados e redimensionados em tamanho de 128 x 128 pixels. A rede foi treinada com 717 imagens e testada com 307 imagens. O treinamento foi realizado por 2.000 épocas. O modelo obteve 0,99 de precisão no conjunto de treinamento e 0,77 no conjunto de teste. A Tabela 2 apresenta a matriz de confusão da CNN.

TABELA 2 – Matriz de confusão para o detector de defeitos

Classe	Defeituoso	Não defeituoso
Defeituoso	140	42
Não defeituoso	26	99

Fonte: Autoral (2023).

A rede classificou incorretamente apenas 68 exemplos. Portanto, o modelo obteve bons resultados mesmo tendo sido treinado com poucos passos e um conjunto de dados com poucas imagens. As imagens cortadas reduzem o custo computacional, pois as dimensionalidades das imagens são reduzidas e, consequentemente, a rede precisará de menos neurônios. Isso mostra o uso potencial de um modelo CNN como uma rede de classificação de defeitos. Espera-se que o treinamento da rede em um conjunto de dados maior e com mais etapas melhore ainda mais o desempenho e os resultados.

3.4 Resultados com a abordagem completa e discussão final dos resultados

O processo completo da abordagem proposta é composto pela fusão dos dois modelos, treinados separadamente. Após o treinamento, eles são salvos em disco para serem usados, o que possibilita aplicá-los a qualquer momento. Assim, a saída do detector de objetos é a entrada do algoritmo de pré-processamento que retorna cada objeto recortado e redimensionado, que é a entrada do detector de defeitos. Ao final, é possível obter informações sobre o objeto e se ele está com defeito ou não. Essas informações podem subsidiar estratégias de tomada de decisão em indústrias de manufatura. A Figura 5b apresenta exemplos de detecção após a aplicação da abordagem completa.

O estudo realizado por (CHEN *et al.*, 2017) teve como objetivo detectar pequenos objetos, utilizando melhorias do modelo RCNN, que apresentou resultados satisfatórios. No entanto, os pesquisadores não identificaram outras características das imagens, como a presença de defeitos. Por outro lado, o presente estudo identifica outras informações sobre os objetos, incluindo a detecção de defeitos. Por outro lado, os autores (AN *et al.*, 2020) empregaram uma melhoria do modelo FAST-RCNN com aprendizado de transferência em um novo conjunto de dados de objetos manufaturados. Além da detecção de objetos, este estudo também se concentra na identificação de defeitos em objetos, separando essas responsabilidades em dois modelos diferentes. Dessa forma, é possível treinar outros modelos para realizar outras tarefas a fim de obter informações adicionais do conjunto de dados. Por

exemplo, é possível treinar um modelo para prever o nível de defeito dos objetos e outro modelo para prever o tipo de defeito. Esses modelos podem ser treinados e testados independentemente para fornecer informações precisas e detalhadas sobre o conjunto de dados.

4. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta para detecção de objetos e detecção de defeitos, separando essas responsabilidades em dois modelos diferentes. Um modelo era responsável por detectar objetos e o outro, com base nos objetos encontrados, define se o objeto é defeituoso ou não. O detector de objetos conseguiu operar bem, obtendo 0,91 de mAP e 0,96 de IoU. Por outro lado, a combinação desses dois modelos permitiu a localização do objeto na imagem e define se o objeto é defeituoso, ampliando as possíveis aplicações desse processo.

Dessa forma, a separação em dois modelos permite maior flexibilidade, pois cada uma precisa aprender apenas as informações necessárias. Além disso, permite a criação de conjuntos de dados separados para treinamento e teste de cada modelo de forma independente. Assim, ambos os modelos podem ser treinados ao mesmo tempo, pois esta é a etapa mais demorada.

A principal limitação do presente trabalho é o número limitado de imagens/objetos devido à complexidade da coleta de imagens de objetos manufaturados com e sem defeitos e à anotação das informações dos objetos. Para resolver isso, técnicas de aumento de dados podem ser aplicadas para gerar novas imagens. Além disso, são sugeridos os seguintes trabalhos futuros: (*i*) realizar a classificação de objetos de acordo com seu nível de defeito; (*ii*) explorar a aplicação dos modelos propostos em outras áreas, como o desenvolvimento de dispositivos vestíveis que auxiliam pessoas com deficiência visual; e (*iii*) analisar a influência do material na detecção do defeito.

Agradecimentos

Este trabalho financiado pelo CNPq (Processo nº 428142/2018-4, Edital MCTIC/CNPq Nº 28/2018 – Universal 018), essencial para a realização desta pesquisa. Os autores agradecem ao Projeto TOOLING4G "*Advanced Tools for Smart Manufacturing*" (TOOLING4G/2016/24516), cofinanciado pelo COMPETE 2020, LISBOA2020, Portugal 2020 (PT2020), e pelo FEDER.

REFERÊNCIAS

ADARSH, P.; RATHI, P.; KUMAR, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. **2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)**, mar. 2020.

AHMAD, T. *et al.* Object Detection through Modified YOLO Neural Network. **Scientific Programming**, v. 2020, p. 1–10, 6 jun. 2020.

AN, M. *et al.* Fabric defect detection using deep learning: An Improved Faster R-approach. **2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)**, jul. 2020.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [s.l.] Springer, 2016.

CHEN, C. *et al.* R-CNN for Small Object Detection. **Computer Vision – ACCV 2016**, p. 214–230, 2017.

CHEN, Y. *et al.* Defect inspection technologies for additive manufacturing. **International Journal of Extreme Manufacturing**, v. 3, n. 2, p. 022002, 3 mar. 2021.

DOLLAR, P. *et al.* Pedestrian detection: A benchmark. **2009 IEEE Conference on Computer Vision and Pattern Recognition**, jun. 2009.

ISO - INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/ASTM 52900:2015**. Disponível em: <https://www.iso.org/standard/69669.html>.

KIM, F. H.; MOYLAN, S. P. Literature review of metal additive manufacturing defects. maio 2018.

LEMOS, C. B. *et al.* Convolutional Neural Network Based Object Detection for Additive Manufacturing. **2019 19th International Conference on Advanced Robotics (ICAR)**, dez. 2019.

LIN, T.-Y. *et al.* **Microsoft COCO: Common Objects in Context**. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/1405.0312.pdf>.

MALHOTRA, P.; GARG, E. Object Detection Techniques: A Comparison. **2020 7th International Conference on Smart Structures and Systems (ICSSS)**, jul. 2020.

MANDAL, B.; OKEUKWU, A.; THEIS, Y. **Masked Face Recognition using ResNet-50**. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2104.08997.pdf>.

NGO, T. D. *et al.* Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. **Composites Part B: Engineering**, v. 143, p. 172–196, jun. 2018.

PATHAK, A. R.; PANDEY, M.; RAUTARAY, S. Application of Deep Learning for Object Detection. **Procedia Computer Science**, v. 132, p. 1706–1717, 2018.

RODRIGUES, E. R. *et al.* INOVAÇÃO TECNOLÓGICA EM PRODUTOS E PROCESSOS: ESTUDO DE CASO EM EMPRESA DE AUTOMAÇÃO INDUSTRIAL. **Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782)**, v. 1, n. 1, 24 ago. 2016.

SCHNEIDER, C. F. **MACHINE LEARNING APLICADO NA PREVISÃO DE RESULTADOS DE PARTIDAS DE FUTEBOL: UM ESTUDO DE CASO PARA COMPARAÇÃO DE DIFERENTES CLASSIFICADORES**. [s.l: s.n.].

ULLAH, M. B.; ULLAH, MD. B. CPU Based YOLO: A Real Time Object Detection Algorithm. **2020 IEEE Region 10 Symposium (TENSYMP)**, 2020.

VAIDYA, S.; AMBAD, P.; BHOSLE, S. Industry 4.0 – A Glimpse. **Procedia Manufacturing**, v. 20, n. 20, p. 233–238, 2018.
VOULODIMOS, A. *et al.* Deep Learning for Computer Vision: A Brief Review. **Computational Intelligence and Neuroscience**, v. 2018, p. 1–13, 2018.

WANG, C. *et al.* Machine learning in additive manufacturing: State-of-the-art and perspectives. **Additive Manufacturing**, v. 36, p. 101538, dez. 2020.

WANG, T. *et al.* A fast and robust convolutional neural network-based defect detection model in product quality control. **The International Journal of Advanced Manufacturing Technology**, v. 94, n. 9-12, p. 3465–3471, 15 ago. 2017.

YANG, H. *et al.* Towards improving classification power for one-shot object detection. **Neurocomputing**, v. 455, p. 390–400, set. 2021.

YIN, Y.; LI, H.; FU, W. Faster-YOLO: An accurate and faster object detection method. **Digital Signal Processing**, v. 102, p. 102756, jul. 2020.

ZHU, H. *et al.* A Review of Video Object Detection: Datasets, Metrics and Methods. **Applied Sciences**, v. 10, n. 21, p. 7834, 4 nov. 2020.