



UFOP

Universidade Federal
de Ouro Preto

Universidade Federal de Ouro Preto - UFOP
Escola de Minas
Departamento de Engenharia Metalúrgica e de Materiais
Campus Morro do Cruzeiro
Ouro Preto – Minas Gerais – Brasil



MONOGRAFIA DE GRADUAÇÃO

EM ENGENHARIA METALÚRGICA

***DEEP LEARNING APLICADO À ANÁLISE GRANULOMÉTRICA
DE QUASE-PARTÍCULAS DO PROCESSO HYBRID PELLETIZED
SINTER (HPS)***

Natália Fernanda de Castro Meira

**Ouro Preto
2021**

Natália Fernanda de Castro Meira

***Deep Learning* Aplicado à Análise Granulométrica em Quase-
Partículas do *Processo Hybrid Pelletized Sinter (HPS)***

Monografia apresentada ao Curso de Engenharia Metalúrgica da Escola de Minas da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenharia Metalúrgica.

Orientador: Prof. Dr. Cláudio Batista Vieira

Coorientador: Prof. Dr. Ricardo Augusto Rabelo Oliveira

**Ouro Preto
2021**

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

M514d Meira, Natália Fernanda de Castro.

Deep Learning aplicado à análise granulométrica de quase-partículas do processo Hybrid Pelletized Sinter (HPS). [manuscrito] / Natália Fernanda de Castro Meira. - 2021.

60 f.

Orientador: Prof. Dr. Cláudio Batista Vieira.

Coorientador: Prof. Dr. Ricardo Augusto Rabelo Oliveira.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia Metalúrgica .

1. Inteligência artificial - Visão computacional. 2. Redes neurais (Computação). 3. Rede neural convolucional. I. Oliveira, Ricardo Augusto Rabelo. II. Vieira, Cláudio Batista. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 669:004.8

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526



MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto
Escola de Minas
Departamento de Engenharia Metalúrgica e de Materiais



ATA DE DEFESA DE MONOGRAFIA

Ao primeiro dia do mês de setembro de 2021, às 9 horas e 30 minutos, por meio da plataforma de videoconferência Google Meet, foi realizada a defesa de monografia pela aluna **Natália Fernanda de Castro Meira**, sendo a comissão examinadora constituída pelo Prof. Dr. Cláudio Batista Vieira (Orientador – DEMET/UFOP), pelo Prof. Dr. Ricardo Augusto Rabelo (Co-orientador – DECOM/UFOP), pela Prof^a. Dr^a. Andrea Gomes Campos Bianchi (DECOM/UFOP) e pelo engenheiro Roberto Okada Junior (ArcelorMittal Monlevade). A aluna apresentou a monografia intitulada: “*Deep learning* aplicado à análise granulométrica de quase-partículas do processo *Hybrid Pelletized Sinter (HPS)*”. A comissão deliberou, por unanimidade, pela aprovação da monografia, concedendo à aluna um prazo de 15 dias para incorporar, no texto final, as alterações sugeridas. Na forma regulamentar, lavrou-se a presente ata que vai devidamente assinada pelos membros da comissão e pela aluna.

Ouro Preto, 01 de setembro de 2021.

Prof. Dr. Cláudio Batista Vieira
Orientador – Presidente

Prof. Dr. Ricardo Augusto Rabelo
Co-orientador – DECOM/UFOP

Prof^a. Dr^a. Andrea Gomes Campos Bianchi
Membro – DECOM/UFOP

Eng. Roberto Okada Junior
Membro - ArcelorMittal Monlevade

Natália Fernanda de Castro Meira
Aluna

O que sabemos sobre o universo, e como sabemos? De onde ele veio e para onde está indo? O universo teve um começo? Se teve, o que aconteceu antes? Qual é a natureza do tempo? Um dia ele vai chegar ao fim?

Stephen Hawking

RESUMO

A indústria minero-metalúrgica busca se adaptar à Indústria 4.0 através da implementação da Inteligência Artificial nos processos. O objetivo deste trabalho é propor duas implementações de Redes Neurais Convolucionais: a primeira é o uso de um algoritmo de *deep learning* embarcado em um dispositivo de *edge computing* para classificar imagens de acordo com a presença ou não de amostras de quase-partículas. Em seguida, a implementação de um algoritmo de *deep learning* para realizar a segmentação das quase-partícula de acordo com sua faixa de distribuição granulométrica. A primeira abordagem consistiu em um modelo treinado utilizando a ferramenta aXeLeRate, usando a estrutura Keras-Tensorflow e a arquitetura MobileNet, em um sistema embarcado usando a placa SiPEED MaiX Dock. Os resultados mostraram uma acurácia no conjunto de validação do modelo de 98,60%. As métricas precisão, revocação e F1 para a validação foram, respectivamente, 98,60%, 100,00% e 99,34% e, os resultados do conjunto de teste no SiPEED foram, respectivamente, 100,00%, 70,00% e 82,35%. A abordagem de segmentação foi realizada com o algoritmo Mask R-CNN, em que as quase-partículas foram classificadas de acordo com a classe da faixa granulométrica correspondente. Foram realizados quatro treinamentos, com as arquiteturas de *backbone* ResNet101 e ResNet50, com 100 e 50 épocas de treinamento. Os resultados demonstraram boa concordância das máscaras com as bordas das partículas, individualizando as instâncias. Os valores de mAP obtidos foram entre 0,2333 e 0,2585.

Palavras chave: *AIoT*, Inteligência Artificial, *Edge Computing*, *Edge Learning*, Visão Computacional, Redes Neurais Convolucionais

ABSTRACT

The mining-metallurgical industry seeks to adapt to Industry 4.0 through the implementation of Artificial Intelligence in the processes. The objective of this work is to propose two implementations of Convolutional Neural Networks: the first is the use of a deep learning algorithm embedded in an edge computing device to classify images according to the presence or absence of quasi-particle samples. Then, the implementation of a deep learning algorithm to perform the segmentation of the quasi-particles according to their particle size distribution range. The first approach consisted of a trained model using the aXeRate tool, using the Keras-Tensorflow framework and the MobileNet architecture, in an embedded system using the SiPEED MaiX Dock board. The results showed an accuracy in the validation set of the model of 98.60%. The precision, recall, and F1 metrics for validation were, respectively, 98.60%, 100.00%, and 99.34%, and the test set results in SiPEED were, respectively, 100.00%, 70.00 %, and 82.35%. The segmentation approach was performed with the Mask R-CNN algorithm, in which the quasi-particles were classified according to the corresponding particle size range. Four training sessions were carried out, with the ResNet101 and ResNet50 backbone architectures, with 100 and 50 trainings epochs. The results showed good agreement of the masks with the edges of the particles, individualizing the instances. The mAP values obtained were between 0.2333 and 0.2585.

Keywords: AIoT, Artificial Intelligence, Edge Computing, Edge Learning, Computer Vision, Convolutional Neural Networks

LISTA DE FIGURAS

Figura 1 - SiPEED M1 Dock _____	23
Figura 2 - Diagrama de bloco da K210. _____	25
Figura 3 - Treinamento e compilação com o aXeLeRate seguindo a metodologia proposta por Klippel _____	31
Figura 4 - Anotação das quase-partículas utilizando a ferramenta VIA (VGG Image Annotator) _____	33
Figura 5 - Exemplo de curva precisão versus revocação _____	37
Figura 6 - Gráfico de treinamento _____	38
Figura 7 - SiPEED MaiX Dock - demonstração do teste _____	40
Figura 8 - Exemplo de reconhecimento de quase-partículas simulando amostragem em um ambiente industrial durante o teste usando SiPEED _____	41
Figura 9 - Exemplo de erro no reconhecimento de quase-partículas simulando amostragem em um ambiente industrial durante o teste usando SiPEED _____	41
Figura 10 - Exemplo de reconhecimento de quase-partículas com amostra desenvolvida em escala de bancada durante o teste usando SiPEED _____	42
Figura 11 - A predição demonstra corretamente as máscaras como instâncias da mesma classe, a bounding box e os resultados de confiança preditos _____	43
Figura 12 - A predição demonstra bordas bem definidas e resultados de confiança preditos _____	43
Figura 13 - Os gráficos representam a perda geral dos treinamentos: a) treinamento 1; b) treinamento 2; c) treinamento 3, e; d) treinamento 4 _____	46
Figura 14 - Os gráficos representam a perda geral da validação: a) treinamento 1; b) treinamento 2; c) treinamento 3, e; d) treinamento 4 _____	47
Figura 15 - Exemplo de imagem contendo partículas da classe '4-6' do treinamento 1. Em a) as partículas em destaque representam as partículas rotuladas e calculadas como verdade fundamental (ground truth), e em b) as partículas em destaque representam as partículas encontradas pelo modelo na predição (predictions) _____	49
Figura 16 - Curva precision-recall para a imagem da Figura 15 _____	49
Figura 17 - Exemplo de imagem contendo partículas da classe '8-9' do treinamento 4. Em a) as partículas em destaque representam as partículas rotuladas e calculadas como verdade fundamental (ground truth), e em b) as partículas em destaque representam as partículas encontradas pelo modelo na predição (predictions) _____	50
Figura 18 - Curva precision-recall para a imagem da Figura 17 _____	50

LISTA DE TABELAS

Tabela 1 - Números de desempenho da plataforma embarcada _____	23
Tabela 2 - Distribuição das imagens no dataset por classe _____	29
Tabela 3 - Número de imagens e anotações no dataset _____	33
Tabela 4 - Valores de hiperparâmetros ajustados no treinamento da Mask R-CNN para as quase partículas. As colunas TRAIN simbolizam cada um dos treinamentos _____	34
Tabela 5 - Matriz de Confusão do modelo - conjunto de validação _____	39
Tabela 6 - Desempenho do modelo treinado no conjunto de validação _____	39
Tabela 7 - Matriz de Confusão do modelo - conjunto de teste _____	40
Tabela 8 - Desempenho do modelo treinado no conjunto de teste _____	40
Tabela 9 - Valores de perda obtidos nos 4 treinamentos realizados. Os valores em destaque representam o menor valor para a perda selecionada _____	44
Tabela 10 - Tempo de execução do treinamento de acordo com o backbone e número de épocas selecionado _____	45
Tabela 11 - Valores da métrica mAP obtidas para cada treinamento _____	48

LISTA DE SIGLAS

AIOT - Artificial Intelligence of Things

ANN – Artificial Neural Networks

AP – Average Precision

CNN – Convolutional Neural Networks

DL – Deep Learning

FCN - Fully Convolutional Network

HPS – Hybrid Pelletized Sinter

IA – Inteligência Artificial

IOT – Internet of Things

IOU - Intersection over Union

MAP – Mean Average Precision

MEV – Microscópio Eletrônico de Varredura

RNN - Recurrent Neural Networks

RoI - RoI Align

SCAP - Sistema de Controle Avançado

SSD - Single Shot Detector

YOLO - You Only Look Once

SUMÁRIO

1 INTRODUÇÃO	12
2 REFERENCIAL TEÓRICO	15
2.1 Deep Learning	15
2.2 Convolutional Neural Networks (CNNs)	16
2.3 Deep Learning aplicado a cenas complexas	17
2.4 Mask R-CNN	18
2.5 Edge AI	20
2.5.1 Edge AI hardware	22
3 TRABALHOS RELACIONADOS	26
3.1 Controle automático da granulometria em processos de aglomeração de minério de ferro	26
3.2 Análise granulométrica em processos de aglomeração de minério de ferro por imagens	27
4 MATERIAIS E MÉTODOS	29
4.1 aXeLeRate com arquitetura MobileNet	29
4.1.1 Dataset	29
4.1.2 Treinamento do modelo Deep Learning	30
4.1.3 Métricas de Avaliação	31
4.1.4 Construção Edge AI	30
4.2 Arquitetura Mask R-CNN para segmentação das quase-partículas	32
4.2.1 Dataset	32
4.2.2 Hiperparâmetros e ajuste da arquitetura	33
4.2.3 Treinamento	35
4.2.4 Métricas de avaliação: Tensorboard e mAP	35
5 RESULTADOS E DISCUSSÕES	38
5.1 Resultados da classificação com aXeLeRate	38
5.1.1 Desempenho do modelo treinado	38
5.1.2 Desempenho do modelo em Edge AI	39
5.2 Resultados da segmentação de instâncias com a Mask R-CNN	42
5.2.1 Segmentação	43
5.2.2 Tensorboard	44
5.2.3 Desempenho do modelo: mAP	48
6 CONCLUSÃO	51
7 REFERÊNCIAS	54

1 INTRODUÇÃO

O setor minero-metalúrgico consiste em uma das áreas produtivas mais tradicionais e nos últimos anos, a inovação e a tecnologia desenvolveram novos métodos de produção e desenvolvimento (KINNUNEN; KAKSONEN, 2019; ROBBEN; WOTRUBA, 2019; MARDONOVA; CHOI, 2018; SINOVIEV *et al.*, 2016; CHEN *et al.*, 2016; SHIBUTA *et al.*, 2018). Assim, projetos inovadores são fundamentais para a modernização desses processos, pois são de alto interesse econômico. Na indústria siderúrgica, um dos principais parâmetros do processo é a distribuição granulométrica dos materiais (ZOBININ *et al.*, 2018). A distribuição granulométrica consiste na distribuição do tamanho das partículas presentes, o que permite sua empregabilidade no processo produtivo.

Ao transitar pela planta de produção, os engenheiros e os operadores precisam saber a distribuição granulométrica continuamente. Essas informações são essenciais como parâmetro de processo e para a tomada de decisões em condições críticas. Ao longo do processo siderúrgico, os materiais são transportados por meio de correias transportadoras em muitos estágios, o que contribui para variações nas características dos materiais, como por exemplo, devido à umidade. Essas variações impactam na distribuição granulométrica e podem comprometer o processo caso não estejam dentro das especificações necessárias (JANUZZI, 2008).

Desse modo, a implementação de um algoritmo em um sistema embarcado que classifique quase-partículas de acordo com sua distribuição granulométrica fornece um caminho para solucionar esse problema e melhorar o processo de produção. Quase-partículas são micro aglomerados de materiais formados no processo HPS (*Hybrid Pelletized Sinter*) (JANUZZI, 2008; LIMA, 2019).

Para tanto, o objetivo deste trabalho é propor duas implementações: a primeira é o uso de um algoritmo de *deep learning* (DL - aprendizado profundo) embarcado em um dispositivo de *edge computing* para classificar imagens de acordo com a presença ou não de amostras de quase-partículas. Em seguida, a implementação de um algoritmo de *deep learning* para realizar a segmentação das quase-partícula de acordo com sua faixa de tamanho. Estas duas etapas compreendem um estágio de pesquisa que busca obter a a faixa tamanho das quase-partículas através de métodos de reconhecimento por imagens.

Na primeira conjectura, o usuário deve fotografar uma amostra do material na correia transportadora ou deve manter o dispositivo fixo em uma área de amostragem em

que as partículas são coletadas em uma bandeja. O resultado é acessível através do display e também através de uma conexão de rede sem fio, que visa classificar aquela imagem como uma amostra de quase-partículas contendo ou não quase-partícula na bandeja, ou, se há outro objeto causando interferência.

O uso de inteligência artificial em dispositivos de *edge computing* ainda é um problema em aberto, e o uso de dispositivos *Edge AI* especializados permitem a expansão do aprendizado profundo (*deep learning*) para a IoT (*Internet of Things* – Internet das Coisas) (DENG *et al.*, 2020). A implementação de uma solução *edge computing* evita uma grande taxa de transferência de transmissão de dados. Essa tendência leva os recursos de computação e comunicação *edge*, com serviços e respostas mais rápidas ao usuário final (DENG *et al.*, 2020).

A segunda abordagem consiste na individualização das quase-partículas pelo método segmentação de instâncias em *deep learning*, detectadas por classe de acordo com a faixa de tamanho esperada. Assim, as quase-partículas que terão sido detectadas dentro da bandeja serão segmentadas de acordo com a faixa de tamanho da classe desejada, o que, neste trabalho, segue a série de classificação das peneiras amplamente utilizadas em laboratório de mineração e metalurgia para distribuição granulométrica.

A resposta rápida às condições detectadas permite um melhor controle do processo. Por exemplo, um padrão de granulometria acima do esperado é um indicador de umidade elevada, o que pode causar entupimento nas calhas (conhecidas como “chutes”) de transferência de material entre as correias transportadoras. Este evento pode paralisar todo o processo produtivo, expondo os operadores a condições de risco e perdendo produtividade. Na rotina do setor, esse processo pode levar muito tempo e não há garantia de qualidade.

Em muitos casos, esse processo leva mudanças de tempo substanciais, tornando impossível habilitar respostas rápidas devido às mudanças nas variáveis de produção. Nas aplicações atuais, a verificação da distribuição granulométrica de certos materiais ocorre por meio de um processo manual. Nesta tarefa, um operador coleta amostra de material do processo produtivo e analisa manualmente com auxílio de uma série de peneiras em um laboratório para obter a distribuição granulométrica. Este procedimento ocorre várias vezes ao dia, e a informação obtida é usada como parâmetro para tomada de decisões sobre o processo.

A relevância deste trabalho consiste na implementação de um método de *deep learning* em um dispositivo *edge computing* para aplicação voltado para o ambiente

industrial, incluindo testes práticos em hardware embarcado, bem como, oferece um *baseline* para a detecção e segmentação das quase-partículas do processo de acordo com a faixa granulométrica.

1.1 Objetivos

1.1.1 Objetivo Geral

Propor um método de reconhecimento de amostras de quase-partículas e do tamanho das quase-partículas por imagem.

1.1.2 Objetivos Específicos

- Implementação de um algoritmo de *deep learning* para reconhecimento da amostra de quase-partícula em uma plataforma embarcada;
- Implementação de um algoritmo de *deep learning* para a segmentação das quase-partículas por classes de tamanho.

1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma: Na Seção 2, consta o referencial teórico, que perpassa pela implementação de Redes Neurais Convolucionais no contexto Deep Learning, além do algoritmo Mask R-CNN implementado e a apresentação da plataforma embarcada utilizada. A Seção 3 apresenta o estado da arte dos trabalhos relacionados no controle de granulometria de aglomerados de minério de ferro e métodos por imagens.

Na Seção 4, encontra-se a metodologia experimental empregada das duas implementações, com a elaboração dos conjuntos de dados, ajuste de hiperparâmetros, treinamento dos modelos e métricas de avaliação. Na Seção 5, apresenta-se os resultados obtidos nas implementações, com a conclusão na Seção 6 e as referências na Seção 7.

2 REFERENCIAL TEÓRICO

Nesta seção, apresenta-se o referencial teórico sobre os conceitos aplicados para desenvolver a solução proposta. Primeiro, desenvolve-se os conceitos de Aprendizado Profundo (*Deep Learning*) e Rede Neural Convolutacional (*Convolutional Neural Network* - CNN), que abrangem as duas implementações propostas neste trabalho.

Em seguida, apresenta-se alguns trabalhos relacionados ao nível de complexidade da imagem das quase-partículas: problemas que envolvem objetos pequenos, com oclusão e sobreposição entre estes objetos.

Posteriormente, apresenta-se a tarefa de segmentação dentro do contexto *deep learning* e a arquitetura Mask R-CNN, implementada na etapa de segmentação das quase-partículas. Por fim, faz-se um fechamento com conceitos de *Edge AI* e as características do *hardware* utilizado na etapa de classificação das amostras de quase-partículas.

2.1 *Deep Learning*

O *deep learning*, através de modelos computacionais com diversas camadas de processamento e utilizando algoritmo de retropropagação para a alteração de parâmetros internos, trouxe grandes avanços no reconhecimento visual de objetos, reconhecimento de fala, detecção de objetos e outros problemas (LECUN *et al.*, 2015). Para os autores, as técnicas convencionais de *machine learning* (aprendizado de máquina) possuíam capacidade limitada para processar dados na forma bruta, enquanto que os métodos que compreendem a aprendizagem por representação pode ser alimentada com dados brutos e descobrir as características necessárias para detecção ou classificação.

Assim, os métodos de *deep learning* consistem em métodos de aprendizagem por representação com vários níveis/camadas, e isto permite que funções complexas (não – lineares) possam ser aprendidas (LECUN *et al.*, 2015).

Existem quatro categorias principais de algoritmos de *machine learning*: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado semi-supervisionado e aprendizado por reforço (CORNETTA e TOUHAFI, 2021). O aprendizado supervisionado (*supervised learning*) é a forma mais comum de *machine learning*, profundo ou não, enquanto que, uma arquitetura de *deep learning* é como uma pilha multicamadas de módulos simples sujeitas a aprendizagem e que computam entradas e saídas não-lineares (LECUN *et al.*, 2015).

No aprendizado supervisionado, o conjunto de dados é um conjunto finito de vetores de características, com rótulos atribuídos a cada elemento do conjunto de dados, em que o objetivo do problema é a prever o rótulo para um exemplo não rotulado (CORNETTA e TOUHAFI, 2021).

2.2 Convolutional Neural Networks (CNNs)

Tarefas inteligentes, como a percepção visual, requerem a construção de representações/características, que devem ser invariáveis em relação as outras informações e variações da entrada. As Redes Neurais Convolucionais (CNNs ou ConvNets) são arquiteturas treináveis capazes de aprender recursos invariantes (LECUN *et al.*, 2010).

As Redes Neurais Convolucionais são arquiteturas multiestágio treináveis, compostas por redes que processam dados na forma de *arrays*¹, como por exemplo, uma imagem colorida composta por três *arrays* 2D contendo intensidades de pixel nos três canais de cores (LECUN *et al.*, 2015; LECUN *et al.*, 2010).

A arquitetura típica de uma CNN consiste em uma estrutura de múltiplos estágios. Os dois primeiros estágios são compostos por camadas convolucionais, que possuem unidades organizadas em mapas de características, e camadas de *pooling* (LECUN *et al.*, 2015). Os estágios de convolução, não linearidade e *pooling* são empilhados e seguidos por mais camadas convolucionais e totalmente conectadas. As ConvNets obtiveram avanços significativos na última década, com o desenvolvimento de novas arquiteturas para tarefas especializadas.

¹ “Em computação científica é muito comum o uso de arrays. Arrays são estruturas de dados semelhantes às listas do Python, mas não tão flexíveis. Em um array todos os elementos devem ser de um mesmo tipo, tipicamente numérico, como int ou float. Além disso, o tamanho de um array não pode ser modificado, ao contrário de listas que podem crescer dinamicamente. Em contrapartida, o uso de arrays é muito mais eficiente e facilita a computação de grandes volumes de dados numéricos. Isso faz com que arrays sejam particularmente úteis em computação científica”. Disponível em: <https://panda.ime.usp.br/algoritmos/static/algoritmos/03-adt3-arrays.html>. Acesso em: Junho de 2021.

2.3 Deep Learning aplicado a cenas complexas

Lecun *et al.* (2015) afirmam que *deep learning* (DL) é um conjunto de técnicas do universo do *machine learning*, muitas vezes referido como Inteligência Artificial (IA). A formalização desses algoritmos vem das Redes Neurais Artificiais (*Artificial Neural Networks* - ANN), contendo várias camadas ocultas e conjuntos de dados de treinamento massivos. De acordo com Zhang *et al.* (2020), algoritmos DL representam o estado da arte em técnicas de aprendizado de máquina. No entanto, a detecção de objetos em cenas densas é particularmente desafiadora.

Alguns dos problemas enfrentados neste assunto são semelhantes a outros apresentados na literatura. Por exemplo, observa-se características semelhantes neste trabalho em aplicações de agricultura de precisão (KERESZTES *et al.*, 2018; SALEEM *et al.*, 2019), e até mesmo na contagem de pessoas em aglomeração (ZHANG *et al.*, 2020). Dentre os desafios apresentados, destacam-se alguns aspectos:

- a) Oclusão - muitas vezes, as quase-partículas se sobrepõem, causando oclusão parcial;
- b) Fundo complexo - homogeneidade na forma, textura ou cor do fundo e dos objetos;
- c) Rotação - as imagens são frequentemente giradas em diferentes ângulos;
- d) Alterações de iluminação - as imagens são expostas a diferentes níveis de luz durante o dia.

Zhang *et al.* (2020) separaram as cenas densas em duas classes diferentes: cenas densas de quantidade e cenas densas internamente. No primeiro, há um grande número de objetos de interesse na cena. O segundo ocorre quando os objetos possuem atributos internos densos. Em ambos os casos, rotular os dados é um desafio significativo, pois a classificação é afetada pelo ruído e pela resolução na detecção de pequenos objetos. Segundo esses autores, as melhores arquiteturas DL para classificação em cenas densas são VGGNet, GoogLeNet e ResNet. Além disso, as melhores arquiteturas para detecção de objetos são DetectNet e YOLO.

Gao *et al.* (2020) analisaram 220 trabalhos relacionados para compreender o processo de contagem sistemática de multidões. Esses autores apontam que o principal desafio é a detecção de pequenos objetos em uma cena. Essa característica em cenas de

multidão ocorre, pois, as cabeças dos indivíduos geralmente são muito pequenas. De acordo com os autores, as técnicas de maior sucesso para contar multidões com base na detecção são SSD, YOLO e RCNNs. Embora essas arquiteturas tenham tido sucesso em cenas esparsas, essas redes tiveram resultados insatisfatórios dadas as cenas com oclusão, desordem e fundo denso. Além disso, o SSD não é eficiente com pequenos objetos nas imagens, pois o mapeamento de recursos de suas camadas intermediárias pode diluir as informações do objeto detectado. Para o R-CNN, Zhou *et al.* (2019a) propuseram uma melhoria com base em PCA Jittering para aprimorar a detecção de pequenos objetos na arquitetura Faster R-CNN.

O trabalho apresentado mostra alguns dos desafios no desenvolvimento de Redes Neurais Convolucionais (CNNs) capazes de analisar cenas densas com objetos ocluídos. Esse problema é mais significativo quando a complexidade do conjunto de dados aumenta. Os desenvolvedores geralmente seguem um procedimento de banco de dados sintético para resolver esse problema, com validação adicional com dados reais. Os resultados obtidos são geralmente bons, exceto se houver um desvio substancial dos conjuntos de dados sintéticos e reais (ZHANG *et al.*, 2020).

2.4 Mask R-CNN

A comunidade de visão computacional, impulsionada por sistemas de *baseline* como Fast e Faster RCNN (GIRSHICK, 2015; REN, 2015) e Fully Convolutional Network (FCN) (LONG *et al.*, 2015), avançaram nas tarefas de detecção e segmentação semântica de objetos. A tarefa de segmentação semântica consiste em uma tarefa desafiadora, pois requer a detecção correta dos objetos na imagem e segmentar precisamente cada instância (HE *et al.*, 2017).

A segmentação semântica combina tarefas clássicas da visão computacional: detectar individualmente objetos, localizar com uma caixa delimitadora (*bounding box*) e fazer a segmentação semântica, no qual classifica cada pixel em um conjunto de categorias sem diferenciar instâncias de objetos (HE *et al.*, 2017). Dessa forma, He *et al.* (2017) propuseram o método Mask R-CNN de segmentação de instâncias, que estende a Faster R-CNN para a prever máscaras de segmentação em cada região de interesse (*Region of Interest* – RoI) com uma *branch* paralela para classificação e regressão da caixa delimitadora.

Um avanço do trabalho de He *et al.* (2017) foi o impacto do RoI, que melhorou a precisão da máscara de 10% à 50%, e ganhos ao desacoplar máscara e predição de classe, para que o RoI previsse a categoria individualmente sem competição de classes. Esse avanço se deve, sobretudo, ao contraste causado pelas FCNs, que combinavam segmentação e classificação, o que não funcionou para segmentação de instâncias.

A arquitetura de rede da Mask R-CNN foi instanciada em várias arquiteturas, divididas em: (i) arquitetura de *backbone* convolucional para a extração de recursos de uma imagem inteira e (ii) *head* da rede para o reconhecimento da caixa delimitadora (classificação e regressão) e previsão de máscara aplicada individualmente em cada RoI. Para a arquitetura de *backbone* foram avaliadas as redes ResNet (HE *et al.*, 2015) e ResNeXt (XIE *et al.*, 2017), com profundidades de 50 ou 101 camadas. Para a *head* da rede, a Mask R-CNN adicionou uma *branch* de predição de máscara totalmente convolucional (HE *et al.*, 2017).

Trabalhos com problemas desafiadores buscam implementar a Mask R-CNN, principalmente para objetos pequenos, o que configurou a Mask R-CNN como a estrutura de segmentação de instância mais influente segundo Zhang *et al.* (2020). Júnior e Rieder (2020) propuseram uma rotina para contagem e identificação automática de insetos em imagens, através armadilhas montadas para a captura em estações da Embrapa Trigo. Para os autores, a tarefa manual de contagem e identificação de pequenos insetos consiste em uma tarefa exaustiva, e a implementação da Mask R-CNN teve como resultado preliminar uma mAP (média da métrica *Average Precision* - precisão média) de 60,4%.

O trabalho de Cohn *et al.* (2021) implementou a Mask R-CNN para a análise de imagens de partículas de pó metálico de superliga de níquel atomizado a gás com potencial aplicação na manufatura aditiva. As imagens foram obtidas por Microscopia Eletrônica de Varredura (MEV) e após o treinamento com a Mask R-CNN, as máscaras demonstraram boa concordância com as partículas de pó presentes na imagem. Foram obtidos alguns falsos positivos em que partículas grandes foram divididas em partículas menores. Para os autores, este fenômeno pode ser considerado comum, uma vez que, as partículas possuem formatos irregulares ou partículas grandes parecem ter sido formadas pela união de partículas menores. A precisão (*precision*) alcançada foi de 0,938 e a revocação (*recall*) de 0,799.

O trabalho de Cohn *et al.* (2021) na área de Ciência dos Materiais pode ser comparado ao de Monchot *et al.* (2021), que propôs uma metodologia para automatizar a caracterização do tamanho das partículas de dióxido de titânio em imagens capturadas

por MEV. O trabalho considerou que a Mask R-CNN produziu máscaras de segmentação de alta precisão dentro dos aglomerados, sendo capaz de detectar de 84% à 97% das partículas com um mAP de 60.6%.

Chen *et al.* (2020) implementaram a Mask R-CNN em imagens metalográficas para a segmentação de uma microestrutura em liga de alumínio. Além disso, os autores implementaram cinco funções de perda diferentes e compararam a influência dessas funções de perda no desempenho da segmentação de imagens metalográficas. Como uma das contribuições do artigo, os autores sugerem que a implementação poderia alcançar segmentação de instâncias de microestruturas em imagens metalográficas de ligas de alumínio de forma automática, fornecendo uma ferramenta mais eficaz para a análise destas imagens.

Outros trabalhos evidenciaram a generalização da implementação da Mask R-CNN e sua melhoria para a detecção automática de animais (TU *et al.*, 2021; BELLO *et al.*, 2021; XU *et al.*, 2020), detecção de aeronaves e prédios em imagens de sensoriamento remoto e imagens de satélite (WU *et al.*, 2021; ZHANG *et al.*, 2020; ZHAO *et al.*, 2018), imagens médicas, como por exemplo, para a segmentação de núcleos e tumores, núcleos de células e nódulos pulmonares (VUOLA *et al.*, 2019; ZHANG *et al.*, 2019; JOHNSON, 2018; LIU *et al.*, 2018), manutenção e controle de processos de manufatura (XI *et al.*, 2020) e mapeamento, quantização e distribuição granulométrica de clastos (SOLOY *et al.*, 2020).

Devido a generalização da Mask R-CNN para diversos problemas de natureza distintas, bem como, a necessidade de diferenciação e individualização das quase-partículas, motivou a implementação da Mask R-CNN para o problema apresentado neste trabalho.

2.5 Edge AI

Um aspecto crítico da solução é a persistência do algoritmo em aplicativos de *edge computing*. A evolução das tecnologias de computação embarcada traz o desafio de fornecer *deep learning* como serviços em aplicações *edge* com qualidade. Assim, a criação de modelos reduzidos e hardware especializado criam o conceito de um “Edge AI” (WANG *et al.*, 2019). Esta nova perspectiva visa o uso de *deep learning* em dispositivos *edge* com independência de aplicativos em nuvem.

No entanto, desenvolver *machine learning* e, especialmente, modelos DL para dispositivos *edge computing* é uma tarefa desafiadora. Redes Neurais Profundas (*Deep Neural Networks* - DNNs) geralmente são modelos computacionalmente intensivos e requerem elevado poder computacional (LI *et al.*, 2019). Mover este aplicativo para a nuvem requer um alto rendimento de dados por meio de uma infraestrutura de rede. O número crescente de dispositivos pode facilmente exceder as capacidades de rede (LIN *et al.*, 2019).

Outro aspecto a ser considerado no desenvolvimento de novas soluções de *edge computing* são as restrições de hardware. Como mencionado anteriormente, a maioria das arquiteturas DL requerem alto desempenho computacional. Um resultado para este problema é a integração de hardware dedicado para otimizar soluções Edge AI (MAZZIA *et al.*, 2020; OHBUCHI, 2018; KARRAS *et al.*, 2020; CORNETTA e TOUHAFI, 2021).

A abordagem e recente disponibilidade de soluções Edge AI contribuíram para a conciliação dos conceitos de *edge computing* e IA, e permite que aplicativos baseados em IA de computação e latência críticas sejam executados em tempo real (CAMPOLO *et al.*, 2021; WANG *et al.*, 2020). Os autores consideram que *edge* e nuvem são complementares: na divisão do fluxo de trabalho do ciclo de vida do AI, o treinamento do modelo pode ser implantado na nuvem enquanto a inferência pode ser realizada na borda.

Cornetta e Touhafi (2021) apresentaram uma revisão de algoritmos de *machine learning* mais populares para serem executados em dispositivos embarcados com recursos limitados. As técnicas de *deep learning* utilizadas nos dispositivos IoT foram as Redes Neurais Artificiais (Artificial Neural Networks - ANN) e as Redes Neurais Recorrentes (Recurrent Neural Networks - RNN) e, para os autores, soluções baseadas em TensorFlowLite ainda não são totalmente implementáveis em dispositivos embarcados.

O trabalho de Liu *et al.* (2021) propôs um método de detecção de anomalias de dados multissensor baseado em *edge computing* em mineração subterrânea. Em geral, a tecnologia IoT é amplamente utilizada no monitoramento e alerta precoce de segurança de construção de mineração subterrânea, porém, alguns problemas estão associados as anomalias dos dados, como: como falhas do sensor, mudanças ambientais e interferência de dados *wireless*. Outros problemas estão associados ao processamento em nuvem: quantidade de transmissão de dados inválidos e redundantes que desperdiçam recursos de rede limitados, alguns dados de sensor que possuem requisitos em tempo real para detecção de anomalias que podem sofrer atrasos, a latência para a nuvem pode ser

proibitiva para aplicativos sensíveis a atrasos e, a transferência de dados confidenciais recuperados por dispositivos IoT pode levantar questões de privacidade (LIU *et al.*, 2021; CAMPOLO *et al.*, 2021; WANG *et al.*, 2020).

Lin *et al.* (2021) implementaram um sistema de detecção de defeitos de pavimento baseado em YOLOv3 em um sistema embarcado Xilinx ZCU104. Os autores compactaram o modelo sem reduzir significativamente a precisão pelo método de quantização, reduzindo em 23% o tamanho do modelo original e compararam o desempenho no Xilinx ZCU104 com um sistema embarcado Nvidia TX2. A velocidade de execução do Xilinx ZCU104 foi de 27,4 FPS, que atendeu aos requisitos de baixo consumo de energia e resposta em tempo real.

Cob-Parro *et al.* (2021) apresentaram um sistema de videovigilância inteligente para detectar, contar e rastrear pessoas em tempo real em um sistema de hardware embarcado com módulos de unidades de processamento de visão (VPUs) na plataforma embarcada UpSquared2 e arquitetura MobileNet-SSD para a tarefa. O modelo atingiu um mAP (*Mean Average Precision*) de 72,7%. O desempenho Edge AI em CPU foi de 13,93ms enquanto que na VPU foi de 8,71ms.

2.5.1 Edge AI hardware

Neste trabalho, implementou-se a solução de classificação utilizando a placa SiPEED MAiX Dock, exibida na Figura 1. Alguns números de desempenho da placa são mostrados na Tabela 1. O trabalho de Klippel *et al.* (2020) demonstra a comparação entre SiPEED MaiX BiT, Raspberry Pi 3 e Jetson Nvidia Nano. Os autores implementaram o SiPEED MaiX BiT para a detecção de rasgos em correias transportadoras. A placa SiPEED MaiX BiT é semelhante a aplicada neste trabalho, que busca seguir a metodologia proposta por Klippel *et al.* (2020).



Figura 1 - SiPEED M1 Dock (própria autora)

Tabela 1 - Números de desempenho da plataforma embarcada (própria autora)

<i>Parâmetro</i>	<i>Características</i>
CPU	64-bit RISC-v processor and core
Chipset	K210 - RISC - V
Reconhecimento de Imagem	qvg at 60fps / vg at 30fps
Clock (GHz)	0,40
AI resources	KPU
OS/Language	uPython
Dimensões (mm)	60x43x5

Esta plataforma possui um dispositivo integrado com aceleração de hardware de inteligência artificial (AI). MAiX é o módulo explicitamente desenvolvido para o SiPEED, projetado para realizar IA. Oferece alto desempenho considerando uma pequena área física e consumo de energia, permitindo a implantação de IA de alta precisão e um preço competitivo. As principais vantagens deste dispositivo são (SEEEDSTUDIO, 2021):

- a) Infraestrutura completa de hardware e software para facilitar a implantação das soluções baseadas em IA;
- b) Bom desempenho, tamanho reduzido, baixo consumo de energia e baixo custo, o que permite ampla implantação de IA de alta qualidade em *edge computing*;
- c) Pode ser usado para um número crescente de casos de uso industrial, como manutenção preditiva, detecção de anomalias, visão de máquina, robótica e reconhecimento de voz.

O Sipeed MAIX atua como o controlador mestre e o hardware possui um KPU K210. O Maixpy foi projetado para a programação AIOT, executando em um chip AIOT K210 com base na sintaxe Micropython. O MicroPython é uma implementação enxuta e eficiente da linguagem de programação Python 3, que inclui um pequeno subconjunto da biblioteca padrão do Python, e é otimizada para ser executada em microcontroladores e em ambientes restritos, facilitando a programação no hardware do K210. O MAIX suporta um modelo de ponto fixo treinado pela estrutura, de acordo com regras de restrição específicas, e possui um compilador para modelos em seu próprio formato. Ele suporta tiny-yolo, mobile net-v1 e TensorFlow Lite (K210, 2018; MaixPy, _____).

O Kendryte K210 é um SoC RISC-V64 dual-core com capacidade de IA que possui recursos de visão de máquina e pode realizar cálculos de Redes Neurais Convolucionais (CNNs) de baixo consumo de energia, com recursos para detecção de objetos, classificação de imagem, detecção e reconhecimento de face, obtenção de tamanho de alvo e coordenadas em tempo real e obtenção do tipo de alvo detectado em tempo real. A Figura 2 exibe o diagrama de blocos K210 (K210, 2018). O KPU é um processador de rede neural de uso geral com convolução interna, normalização e ativação. De acordo com o fabricante, também possui as seguintes características:

- a) Suporta o modelo de ponto fixo que a estrutura de treinamento convencional treina de acordo com regras de restrição específicas;
- b) Não há limite direto no número de camadas da rede, e cada camada dos parâmetros da Rede Neural Convolucional pode ser configurada separadamente, incluindo o número de canais de entrada e saída, a largura da linha de entrada e saída e a altura da coluna;
- c) Suporte para kernels de convolução 1x1 e 3x3;
- d) Suporte para qualquer forma de função de ativação;
- e) O tamanho máximo do parâmetro da rede neural suportada para trabalho em tempo real é de 5MiB a 5.9MiB.

A principal contribuição desta implementação é de um método de *deep learning* em um dispositivo de *edge computing* para aplicação voltada ao ambiente industrial, incluindo testes práticos em hardware embarcado.

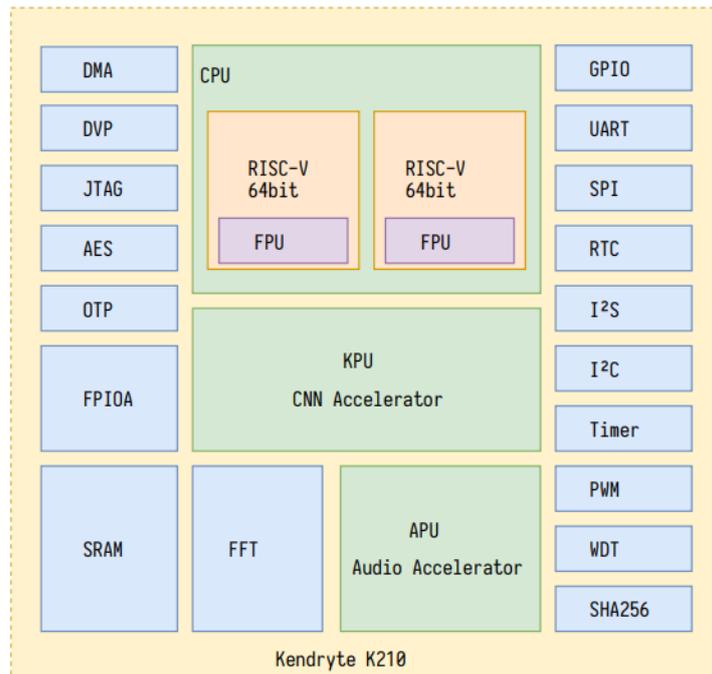


Figura 2 - Diagrama de bloco da K210 (K210, 2018).

3 TRABALHOS RELACIONADOS

3.1 Controle Automático da Granulometria em Processos de Aglomeração de Minério de Ferro

Devido à importância da etapa de aglomeração do minério de ferro para as etapas posteriores do processo, diversos estudos têm sido realizados a fim de controlar e monitorar as variáveis que interferem nos processos de sinterização e pelletização.

Dias (2018) propôs um sistema de controle granulométrico de pelotas de minério de ferro através do controle da injeção de água no tambor de pelletamento que, até então, era feita manualmente pelos operadores de acordo com a necessidade do processo. Os resultados mostraram que a adição de água tende a aumentar a granulometria das pelotas e que o controle tende a homogeneizar as pelotas, contudo, para que a variável controlada apresentasse estabilização, incorreria de se estudar outros parâmetros, como por exemplo, a saturação de água devido a recirculação de pelotas fora da faixa granulométrica requerida.

Estudos sobre a influência das matérias primas no processo de aglomeração a frio do processo HPS também foram estudadas, como mostrado no trabalho de Januzzi (2008). O trabalho teve como objetivo caracterizar as matérias primas, estudar a contribuição de cada uma delas no processo de aglomeração a frio e adequar os parâmetros visando a melhoria na performance do processo. Algumas das medidas tomadas foram as alterações nas curvas de distribuição granulométrica do serpentinito, do calcário calcítico e do minério de manganês, o que promoveu melhora no tamanho médio das quase-partículas e, conseqüentemente, “um efeito positivo na pressão de sucção na máquina de sinter permitindo o aumento da altura de camada, ganho na produtividade e na produção de sinter” (JANUZZI, 2008, p.122), demonstrando mais uma vez a importância da distribuição granulométrica no processo de aglomeração de minério de ferro.

Para o caso em que o controle manual dependia dos operadores de área para obter a granulometria adequada da pelota crua, Passos *et al.* (2014) desenvolveu seu trabalho na implementação de um Sistema de Controle Avançado (SCAP) com o objetivo de controlar a granulometria das pelotas cruas atuando na velocidade e na alimentação dos discos. Os resultados demonstraram estabilidade do processo produtivo, principalmente no controle da distribuição granulométrica das pelotas, estabilidade da dosagem de insumos e aumento da permeabilidade do forno de endurecimento.

Souza (2020) propôs a utilização de algoritmos de *deep learning* para a identificação de partículas de minério de ferro e medição de suas dimensões lineares a partir de imagens obtidas na operação de britagem primária. Foram avaliados os algoritmos SSD, Faster R-CNN, YOLOv3 e U-NET. As partículas das imagens de bancada consistiam em fragmentos de 4,8mm à 19mm e os fragmentos das imagens de vídeo da área industrial possuíam dimensões superiores à 200mm.

Os resultados obtidos no treinamento das redes SSD, Faster R-CNN e YOLOv3 demonstraram baixa precisão e baixo índice de assertividade. A rede U-NET apresentou acurácia de 91,3%. A partir das máscaras geradas, foi desenvolvida uma rotina com a biblioteca de visão computacional OpenCV para a geração de uma caixa delimitadora sobre a máscara e fornecimento do comprimento lateral da caixa para a medida do objeto.

3.2 Análise Granulométrica por Imagens em Processos de Aglomeração de Minério de Ferro

Para caracterizar os materiais ultrafinos e o consumo de tamanho médio, Gontijo (2018) realizou tratamento prévio de imagens em Microscópio Eletrônico de Varredura (MEV). As partículas das imagens foram digitalizadas, escalonadas em software, classificadas por cores em faixas de tamanho (intervalos) e, após a classificação, foram gerados gráficos de distribuições granulométricas.

O trabalho de Santos *et al.* (2019) propôs uma rotina de análise automática de imagens, para identificar as quase-partículas de sinterização e classificar em três classes, calcular a fração da área de classe, circularidade e espessura da camada aderente e, por fim, quantificar as fases minerais presentes nos núcleos das quase-partículas. Os autores utilizaram amostras produzidas em planta piloto de sinterização, que foram classificadas nas seguintes faixas de tamanho: > 4,76 mm, 2,83–4,76 mm e 1,00–2,83 mm e a fração de tamanho de < 1,0 mm foi descartada.

A aquisição das imagens foi por microscopia ótica de luz refletida com aumento de aproximadamente 50x e resolução de 2,05 $\mu\text{m}/\text{pixel}$. Para o processamento e análise digital de imagens, os autores utilizaram o pacote de processamento de imagem Fiji. Com o computador utilizado, a rotina desenvolvida foi capaz de processar uma imagem de granulometria de 4,76 mm em cerca de 6 min, enquanto uma imagem de granulometria de 1,00 mm levou cerca de 18 min devido ao aumento número de partículas.

No resultado final, os autores consideraram que a rotina desenvolvida forneceu bom desempenho e velocidade, comparado com o desempenho humano, pois o sistema foi capaz de processar amostras de 1,00 mm em cerca de 20 min, enquanto um operador pode demorar até 6 horas. Santos *et al.* (2019) concluíram o trabalho cogitando um trabalho futuro com a utilização de Redes Neurais Convolucionais (CNN) para a segmentação, uma vez que, “as CNNs são capazes de alcançar alta eficiência em problemas de classificação e segmentação, combinando e, às vezes, excedendo o desempenho humano, pois são capazes de processar recursos altamente abstratos” (SANTOS *et al.*, 2019, p.9).

4 MATERIAIS E MÉTODOS

Esta seção descreve a metodologia utilizada para os modelos de classificação e de segmentação das quase-partículas. Primeiro, apresenta-se a metodologia empregada para o funcionamento do dispositivo, de acordo com o hardware alvo. Para isso, apresenta-se o conjunto de dados (*dataset*) empregado, o processo de treinamento e as métricas de avaliação. Apresenta-se também o método de avaliação do desempenho do classificador e validação da transferência do modelo para o hardware mencionado. Em seguida, descreve-se a metodologia empregada para o modelo de segmentação das quase-partículas. Inclui a elaboração do conjunto de dados (*dataset*), do ajuste dos hiperparâmetros, treinamento e métricas de avaliação.

4.1 aXeLeRate com Arquitetura MobileNet

4.1.1 Dataset

Elaborou-se um *dataset* com 1368 imagens para o desenvolvimento e treinamento, contendo 1140 para treinamento e 228 para validação. O conjunto de dados foi dividido em três classes: quase partícula (*quase-particle*), sem-categoria (*non-category*) e vazio (*empty*). A classe principal, da bandeja contendo amostra de quase-partículas, foi composta por imagens reais do ambiente industrial, obtidas de uma empresa setor de minero-metalúrgico, e imagens sintéticas produzidas em escala de bancada. A distribuição das imagens no conjunto de dados é apresentada na Tabela 2:

Tabela 2 - Distribuição das imagens no dataset por classe (própria autora)

Conjunto	Classe	Número de imagens por classe	Origem	Número total de imagens
Treinamento	quase_particula	686	343 reais 343 sintéticas	1140
	sem_categoria	150		
	vazio	304		
Validação cruzada	quase_particula	76	38 reais 38 sintéticas	228
	sem_categoria	76		
	vazio	76		

4.1.2 Treinamento do modelo Deep Learning

O treinamento do modelo de *deep learning* foi realizado na plataforma Google Colaboratory², com a ferramenta aXeLeRate³. O aXeLeRate é uma ferramenta para treinar modelos de classificação e detecção, desenvolvido usando a estrutura Keras / Tensorflow.

Para realizar a tarefa desejada, seguiu-se a metodologia proposta por Klippel *et al.* (2020). Utilizou-se a MobileNet como arquitetura CNN, na versão 0.75 MobileNet-224 v1, configurada como classificador, com 224 entradas, duas camadas totalmente conectadas com 100 e 50 neurônios e *dropout* de 0,5. O treinamento foi realizado em 10 épocas de treinamento e a taxa de aprendizagem (*learning rate*) adotada de 0,001. Os pesos iniciais do modelo foram carregados, considerando o treinamento anterior com o conjunto de dados ImageNet. Além disso, o aumento de dados (*data augmentation*) foi aplicado durante o treinamento.

4.1.4 Construção Edge AI

Para o treinamento, implementou-se a estrutura aXeLeRaTe, uma estrutura baseada em Keras para *Edge AI*, para executar aplicativos de visão computacional (classificação de imagem, detecção de objetos, segmentação semântica) em dispositivos *edge* com aceleração de hardware. O AXeLeRate simplifica o treinamento e a conversão de modelos de visão computacional e é otimizado para o fluxo de trabalho na máquina local e no *Google Colab*. Suporta conversão de modelo treinado para: formatos: *.kmodel* (K210) e *.tflite4*.

A Figura 3 mostra o processo de fluxo da implementação do aXeLeRate, com as principais etapas indicadas pelos círculos azuis e segue a metodologia proposta por Klippel *et al.* (2020). Em (1), o conjunto de dados é carregado no Google Drive para treinamento na estrutura Keras-Tensorflow. Em seguida, (2), os pesos do modelo são entregues no formato *.h5* para classificação e retorna ao Tensorflow (3) para ser convertido no formato *.tflite* (4). Assim, é entregue ao *nncase* (5) para ser compilado no formato *.kmodel* (6), que é executado pelo KPU (7).

2 Disponível em: <<https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR>>

3 Disponível em: <<https://github.com/AIWintermuteAI/aXeLeRate>>

4 Disponível em: <<https://www.tensorflow.org/lite?hl=pt-br>>

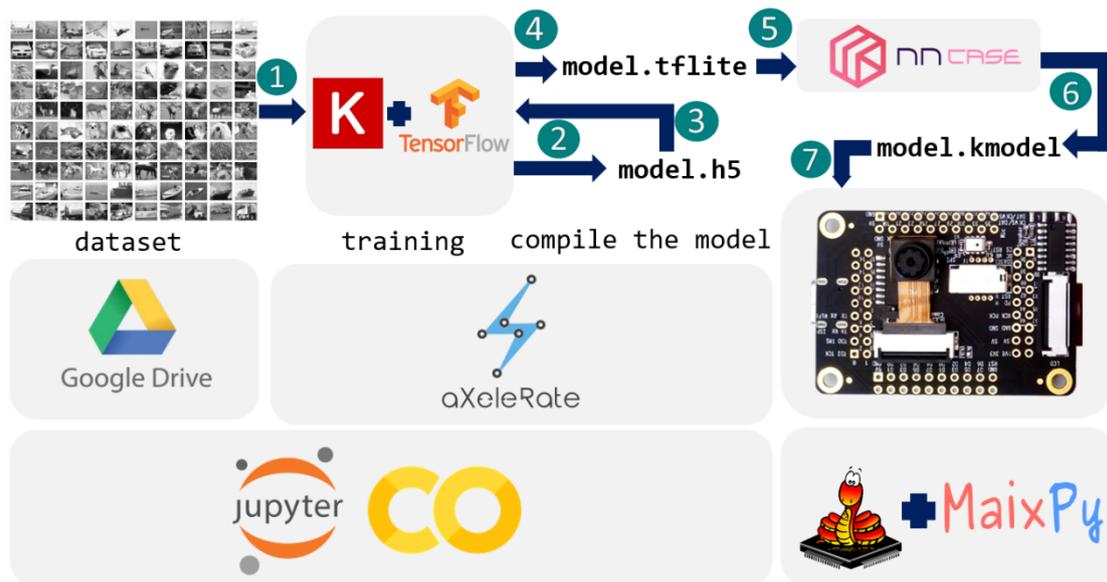


Figura 3 - Treinamento e compilação com o aXeLeRate seguindo a metodologia proposta por Klippel *et al* (2020)(própria autora)

A execução da inferência do modelo foi realizada no dispositivo embarcado SiPEED MaiX Dock em escala de bancada com imagens sintéticas. Para este teste, utilizou-se dois *scripts* Python, em que o primeiro faz a captura das fotos com resolução 224x224 e armazena no cartão SD. O segundo testa o modelo armazenado no cartão SD.

4.1.3 Métricas de Avaliação

Em um primeiro momento, o desempenho do modelo de classificação foi calculado por meio da Matriz de Confusão, que mostra as frequências de classificação para cada classe do modelo. A partir desses dados, obtêm-se as métricas: precisão (*precision* - Equação 4.1), revocação (*recall* - Equação 4.2) e *F1* (Equação 4.3). Esses parâmetros definem, respectivamente, quão bem o modelo funcionou, quão bom o modelo é para prever positivos e o balanço entre a precisão e a revocação do modelo.

As definições utilizadas são: TP como verdadeiro positivo (True Positive), FP é falso positivo (False Positive) e FN é falso negativo (False Negative). A previsão TP ocorre quando a previsão da classe principal está correta, FP ocorre quando a previsão detectada como positiva está errada (mal prevista) e FN ocorre quando a classe principal não foi prevista.

$$precision = \frac{TP}{TP+FP} \quad (4.1)$$

$$recall = \frac{TP}{TP+FN} \quad (4.2)$$

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (4.3)$$

Após o treinamento, o modelo fornece um gráfico da acurácia pelo número de épocas. A métrica acurácia é definida pela Equação 4.4 (CHEN, 2020).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.4)$$

4.2 Arquitetura Mask R-CNN para Segmentação das Quase-partículas

4.2.1 Dataset

As imagens dos conjuntos de dados foram elaboradas a partir de amostras reais de quase-partículas, obtidas no ambiente industrial. Após a amostragem ser realizada em uma bandeja com o auxílio de um operador, as partículas foram fotografadas na bandeja e peneiradas. As partículas de cada peneira foram dispostas novamente em bandejas e fotografadas. Assim, cada imagem possuía uma faixa granulométrica de partículas conhecida, de tal forma que, essas faixas granulométricas foram consideradas como classes para a segmentação no algoritmo Mask R-CNN. As classes foram nomeadas de acordo com o intervalo de tamanho das partículas em milímetros: '2-3', '3-4', '4-6', '8-9' e '>9', totalizando 5 classes. O algoritmo Mask R-CNN considera o *background* da imagem como uma classe, totalizando 6 classes para o treinamento.

Para a realização dos treinamentos utilizando a Mask R-CNN modificando os hiperparâmetros, elaborou-se um *dataset* contendo 81 imagens para treinamento, com 4801 regiões anotadas (rotuladas) e 46 imagens para validação, contendo 460 anotações, ou seja, aproximadamente 10% das anotações de treinamento (Tabela 3). Os rótulos das imagens seguem as classes supracitadas. As imagens foram redimensionadas para 1488x1488x3 antes da anotação, a fim de conciliar com a capacidade do hardware disponível.

Tabela 3 - Número de imagens e anotações no *dataset* (própria autora)

	Imagens	Regiões anotadas
Treinamento	81	4801
Validação	46	460

As anotações foram realizadas manualmente utilizando a ferramenta VIA (VGG Image Annotator⁵), que consiste em um projeto *open source* desenvolvido pelo *Visual Geometry Group (VGG)* para anotação manual. A ferramenta gera um arquivo do tipo *.json* com as informações das imagens anotadas. A Figura 4 mostra um exemplo de anotação de imagens sendo realizada com a ferramenta VIA.

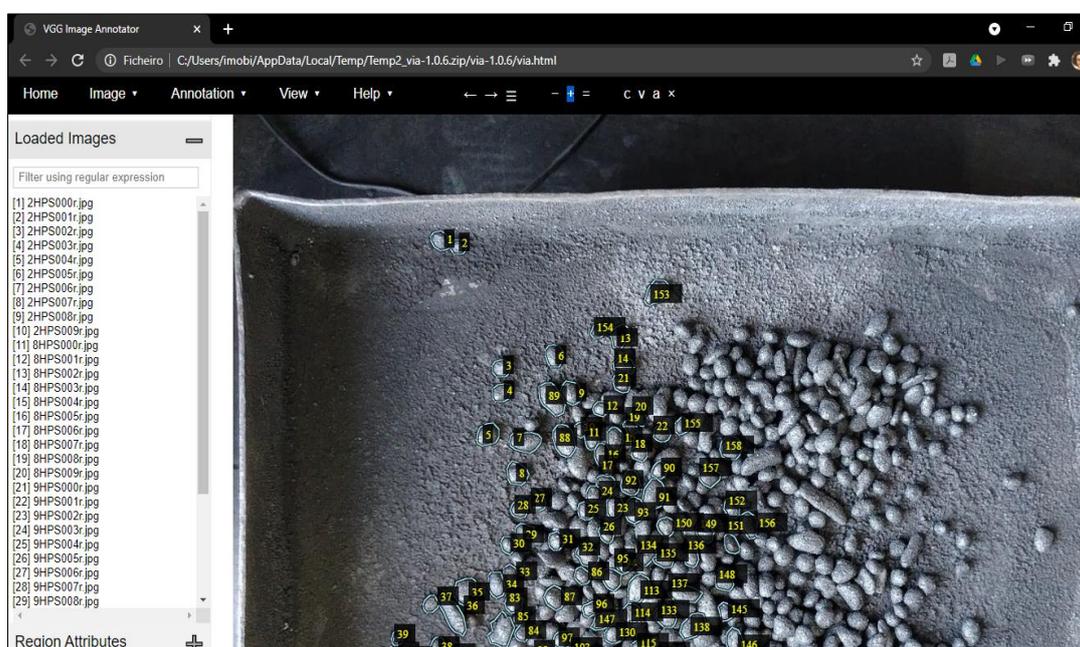


Figura 4 - Anotação das quase-partículas utilizando a ferramenta VIA (VGG Image Annotator) (própria autora)

4.2.2 Hiperparâmetros e ajuste da arquitetura

Implementou-se a Mask R-CNN a partir do repositório original disponível no GitHub⁶. Alguns hiperparâmetros foram ajustados a fim de conciliar com o modelo proposto neste estudo, baseado nas explanações de Bobba (2019) e De Cesaro Júnior (2020). O primeiro hiperparâmetro consiste no *backbone*, arquitetura convnet da primeira etapa da Mask R-CNN.

⁵ Disponível em: <<https://www.robots.ox.ac.uk/~vgg/software/via/>>.

⁶ Disponível em: <https://github.com/matterport/Mask_RCNN>

Foram realizados treinamentos utilizando dois *backbones*, ResNet50 e ResNet101, a fim de comparar as diferenças de tempo de treinamento e precisão. Espera-se que o ResNet101 realize maior tempo de treinamento devido ao maior número de camadas, e que seja mais preciso com os hiperparâmetros bem ajustados. Todos os treinamentos foram executados com o mesmo *dataset* apresentados na seção anterior, a fim de se comparar somente os resultados e ajustes dos hiperparâmetros.

Foram utilizados os valores padrão para taxa de aprendizagem (*learning rate*) e decaimento de peso (*weight decay*), com valores de 0,001 e 0,0001, respectivamente.

Os hiperparâmetros ajustados em cada treinamento são mostrados na Tabela 4. O *backbone* é a arquitetura ConvNet utilizada na primeira etapa da Mask R-CNN. O TRAIN_ROIS_PER_IMAGE é o número máximo de ROI's (*Region of Interest* – regiões de interesse) que a RPN irá gerar para a imagem. O MAX_GT_INSTANCES é o número de instâncias que podem ser detectadas em uma imagem. O DETECTION_MIN_CONFIDENCE é o limite de confiança além do qual ocorrerá a classificação de uma instância. Os hiperparâmetros IMAGE_MIN_DIM e IMAGE_MAX_DIM controlam a resolução de entrada da imagem que, por padrão, são redimensionadas para os tamanhos 1024x1024. Podem ser utilizadas imagens menores (512 x 512) para reduzir requisitos de memória e tempo de treinamento, porém, devido ao tamanho pequeno dos objetos, optou-se por utilizar 1024 x1024. Além destes hiperparâmetros, os pesos que foram inicializados com o valor padrão 1 (BOBBA, 2019; CESARO JÚNIOR, 2020).

Tabela 4 - Valores de hiperparâmetros ajustados no treinamento da Mask R-CNN para as quase partículas. As colunas TRAIN simbolizam cada um dos treinamentos (própria autora)

HIPERPARÂMETROS	TRAIN 1	TRAIN 2	TRAIN 3	TRAIN 4
Backbone	ResNet101	ResNet50	ResNet101	ResNet50
TRAIN_ROIS_PER_IMAGE	500	500	500	500
MAX_GT_INSTANCES	300	300	400	400
DETECTION_MAX_INSTANCES	500	500	500	500
DETECTION_MIN_CONFIDENCE	0.7	0.7	0.7	0.7
IMAGE_MIN_DIM	800	800	800	800
IMAGE_MAX_DIM	1024	1024	1024	1024
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)	(32, 64, 128, 256, 512)	(32, 64, 128, 256, 512)	(32, 64, 128, 256, 512)
EPOCHS	50	50	100	100

4.2.3 Treinamento

Foram realizados 4 treinamentos, e para a transferência de aprendizado (*transfer learning*), utilizou-se os pesos do conjunto MS COCO7. Os treinamentos foram realizados com 50 épocas e 100 épocas, e 100 passos por época.

Toda a implementação foi realizada na linguagem de programação Python, e as bibliotecas OpenCV8 para o redimensionamento das imagens, e para treinamento, as bibliotecas Tensorflow9 e Keras10. O hardware disponível para os treinamentos foi um computador com processador Intel Core I7-6950X, 32 GB de RAM e a unidade de processamento gráfico (GPU) GTX2080 com 8 GB de VRAM.

4.2.4 Métricas de avaliação: Tensorboard e mAP

Para a visualização gráfica das perdas do modelo, utilizou-se o *Tensorboard*11, um kit de ferramentas de visualização do *Tensorflow* que permite visualizar e acompanhar métricas como perda e precisão, grafos do modelo, histogramas, entre outros. A perda da Mask R-CNN é calculada de acordo com a fórmula 4.5 abaixo. Na perda multitarefa definida, L_{cls} é a perda de classificação, L_{box} é a perda da caixa delimitadora e L_{mask} é a perda da máscara (He *et al.*, 2017; BOBBA, 2018).

$$L = L_{cls} + L_{box} + L_{mask} \quad (4.5)$$

Para a avaliação da precisão do modelo utilizou-se a métrica mAP, que é uma métrica frequentemente utilizada nas tarefas de reconhecimento de objetos. Durante a detecção, busca-se prever caixas delimitadoras que se sobreponham a verdade fundamental rotulada. A previsão do quão bom esta sobreposição ocorre pode ser obtida dividindo-se a área da sobreposição pela área total de ambas as caixas delimitadoras, fornecendo a métrica IoU (*Intersection over Union* - interseção sobre união) como mostrado na Equação 4.6 (BANSAL, 2019; SHAH, 2018). É comum nos conjuntos de

7 Disponível em: <https://github.com/matterport/Mask_RCNN/releases/tag/v2.0>

8 Disponível em: <<https://opencv.org/>>

9 Disponível em: <<https://tensorflow.org>>

10 Disponível em: <<https://keras.io>>

11 Disponível em: <<https://www.tensorflow.org/tensorboard?hl=pt-br>>

dados predefinir um limite IoU de 0,5 ao classificar se a previsão é um verdadeiro positivo ou falso positivo (HUI, 2018).

$$IoU = \frac{\text{área de interseção}}{\text{área de união}} \quad (4.6)$$

As métricas apresentadas na seção 4.1.3, *precision* (precisão) e *recall* (revocação) avaliadas separadamente não são suficientes para expressar se um modelo de detecção é suficientemente bom:

Quando um modelo tem alta revocação, mas baixa precisão, o modelo classifica a maioria das amostras positivas corretamente, mas tem muitos falsos positivos (ou seja, classifica muitas amostras negativas como positivas). Quando um modelo tem alta precisão, mas baixa revocação, o modelo é preciso ao classificar uma amostra como positiva, mas pode classificar apenas algumas das amostras positivas (GAD, 2020 – tradução livre).

Na detecção de imagens, a precisão refere-se a porcentagem de caixas delimitadoras previstas corretamente ($IoU > 0,5$) em relação a todas as caixas delimitadoras previstas na imagem, enquanto que a revocação é a porcentagem de caixas delimitadoras previstas corretamente ($IoU > 0,5$) de todos os objetos na imagem (BANSAL, 2019).

Considera-se que a métrica IoU é o limite para uma previsão correta. Logo, pode-se plotar uma curva precisão versus revocação pelo limite de IoU de 0,5. Essa representação fornece uma curva com comportamento zigue-zague para modelos de detecção, embora possa variar para outros modelos (ANDREW NG, ____; HUI, 2018).

Então, maximiza-se o valor da revocação para cada valor de precisão, a fim de suavizar o comportamento da curva (HUI, 2018; BANSAL, 2019), como pode ser observado na Figura 6. A área abaixo da curva fornece o valor médio da precisão, ou seja, a precisão média, métrica AP (*Average Precision*). A média da métrica AP em todas as imagens em um conjunto de dados fornece a métrica mAP (*Mean Average Precision*) (HUI, 2018; BANSAL, 2019; TAN, 2019).

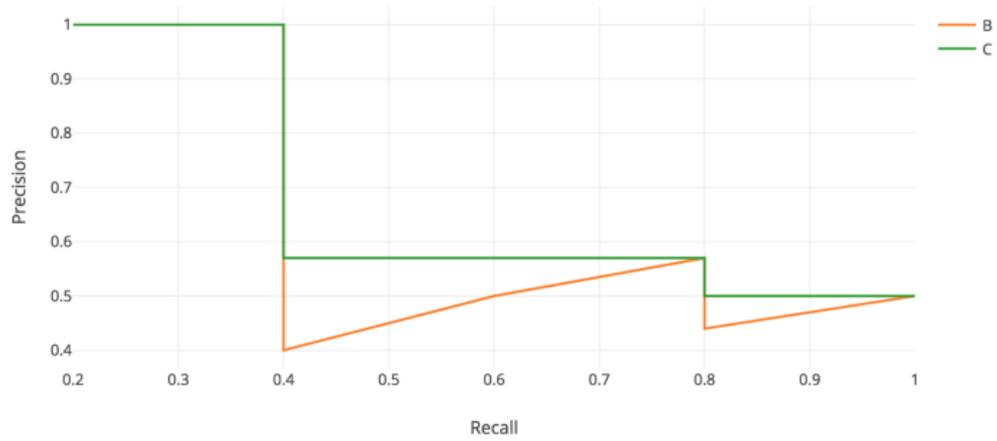


Figura 5 - Exemplo de curva precisão versus revocação (HUI, 2018).

A métrica mAP foi implementada para a avaliação do modelo em relação ao conjunto de dados de validação rotulado.

5 RESULTADOS E DISCUSSÕES

5.1 Resultados da Classificação com aXeLeRate

Nesta seção, apresenta-se os resultados obtidos com a implementação do modelo de classificação. Os resultados preliminares indicam a viabilidade do sistema e mostram as restrições para transportar o modelo para o dispositivo *Edge AI*.

5.1.1 Desempenho do modelo treinado

O tempo decorrido de treinamento foi de 54 minutos, atingindo uma acurária de 98,60%. A Figura 6 mostra a evolução da acurácia ao longo das épocas no treinamento. Conforme mostrado, o treinamento do modelo convergiu em apenas dez épocas, indicando que o modelo não teve grande dificuldade em diferenciar as classes de imagens presentes no conjunto de dados.

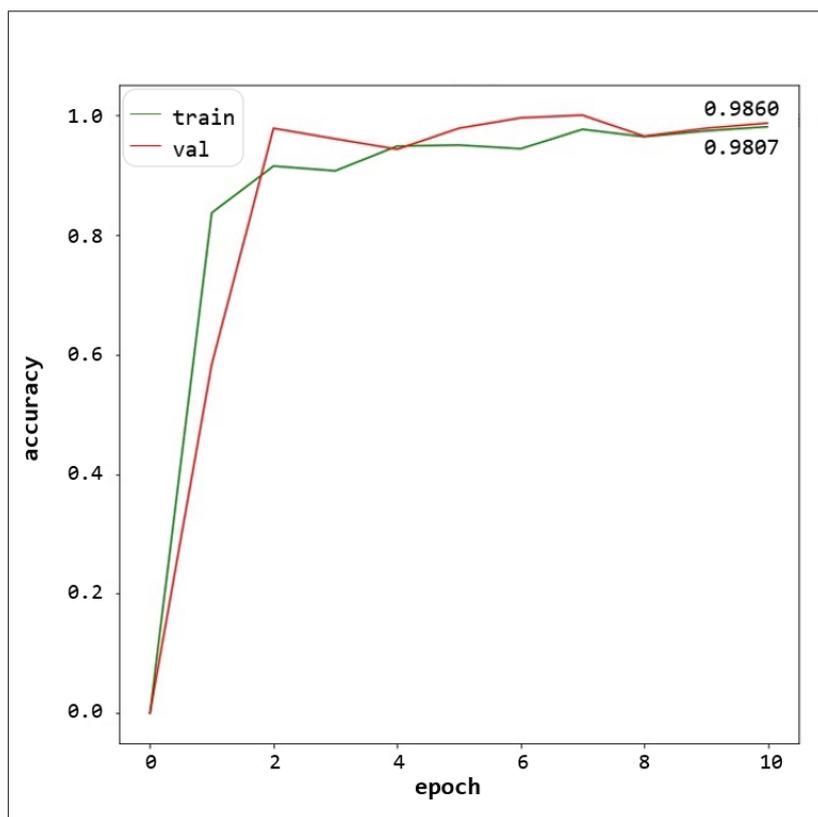


Figura 6 - Gráfico de treinamento (própria autora)

Para validar o modelo, elaborou-se um *dataset* de validação com 228 imagens. As imagens foram divididas em três classes, contendo 76 imagens cada classe:

quase_particle (quase-partícula), *non_category* (sem categoria, para o caso de uma imagem aleatória qualquer) e *empty* (vazio refere-se à mesma bandeja, mas sem a presença de quase-partículas). A Tabela 5 apresenta a matriz de confusão considerando quase-partículas como classe principal e a Tabela 6 mostra os indicadores de desempenho.

Tabela 5 - Matriz de Confusão do modelo - conjunto de validação (própria autora)

		Predict		
		quase_particle	non_category	empty
Real	quase_particle	76	0	0
	non_category	1	74	1
	empty	0	0	76

Tabela 6 - Desempenho do modelo treinado no conjunto de validação (própria autora)

Indicador	Valor
Precision	98,60%
Recall	100,00%
F1	99,34%

A precisão do modelo foi de 98,60%. A aplicação apresentou problemas na classificação de algumas imagens da classe *non_category* com bandejas de quase partículas e vazias. Os dados sugerem um bom *recall*, o que significa que o modelo teve uma pequena taxa de erro na classificação das quase-partículas quando elas eram de fato quase-partículas. Esses resultados demonstram a viabilidade do processo de reconhecimento usando o conjunto de dados proposto. Esse valor possibilitou um equilíbrio na pontuação F1.

5.1.2 Desempenho do modelo em Edge AI

O desempenho do classificador foi testado na plataforma proposta para *edge computing*. Após o treinamento, o modelo foi carregado no SiPEED Maix Dock para teste, conforme mostrado na Figura 8. Para isso, testou-se o sistema usando imagens das três classes (*quase_particle*, *non_category* e *empty*). A Tabela 7 mostra a matriz de confusão e a Tabela 8 mostra os indicadores de desempenho.

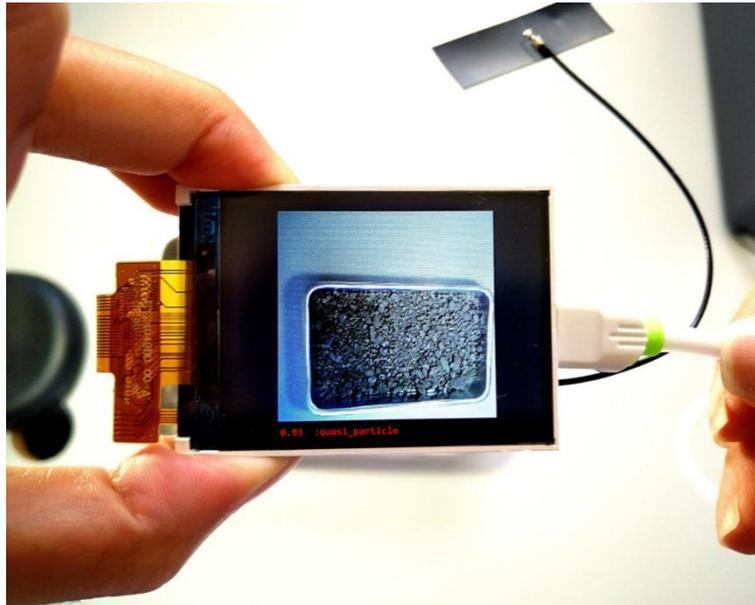


Figura 7 - SiPEED MaiX Dock - demonstração do teste (própria autora)

Tabela 7 - Matriz de Confusão do modelo - conjunto de teste (própria autora)

		Predict		
		quase_particle	non_category	empty
Real	quase_particle	7	2	1
	non_category	0	9	1
	empty	0	0	10

Tabela 8 - Desempenho do modelo treinado no conjunto de teste (própria autora)

Indicador	Valor
Precision	100,00%
Recall	70,00%
F1	82,35%

Em contraste com o valor alcançado no conjunto de validação, o *recall* no conjunto de teste caiu para 70%, avaliado a partir do sistema embarcado SiPEED. Esse resultado indica que o modelo teve dificuldades em testar positivamente para simulações de imagens semelhantes a imagens no ambiente industrial, conforme mostrado nas Figuras 8 e 9, embora para imagens sintéticas com partículas espaçadas não tenha havido dificuldade, como mostra a Figura 10.

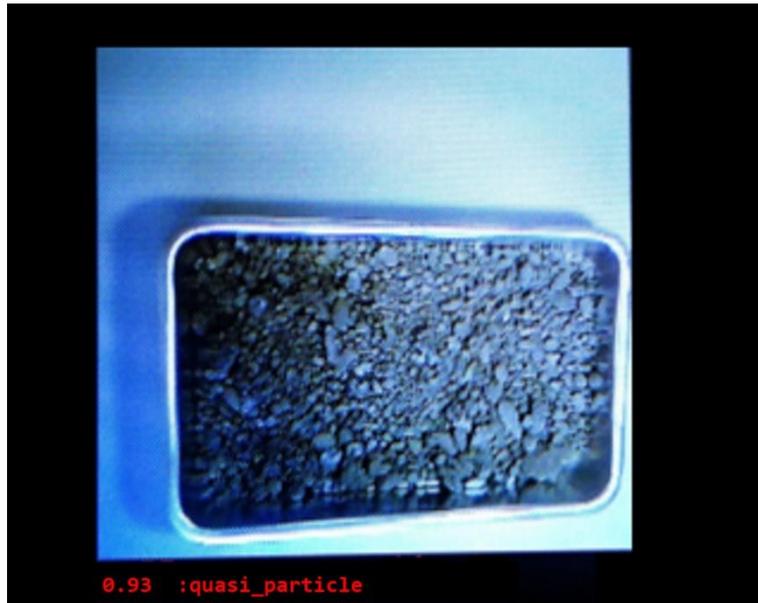


Figura 8 - Exemplo de reconhecimento de quase-partículas simulando amostragem em um ambiente industrial durante o teste usando SiPEED (própria autora)



Figura 9 - Exemplo de erro no reconhecimento de quase-partículas simulando amostragem em um ambiente industrial durante o teste usando SiPEED (própria autora)



Figura 10 - Exemplo de reconhecimento de quase-partículas com amostra desenvolvida em escala de bancada durante o teste usando SiPEED (própria autora)

O trabalho de Klippel *et al.* (2020) implementou o SiPEED MaiX BiT para detectar falhas em correias transportadoras. Os resultados para o desempenho do treinamento são semelhantes aos resultados obtidos por Klippel *et al.* (2020). No desempenho do teste, obteve-se um *recall* inferior, conforme mostrado.

O valor de *recall* nos testes não condiz com os resultados obtidos nos testes realizados por Klippel *et al.* (2020). Para justificar o valor de 70%, justifica-se que o conjunto de dados pode ser aprimorado para apenas imagens reais em análises futuras com as classes equilibradas. Além disso, existe a possibilidade de ter ocorrido *overfitting* durante o treinamento e influência das condições testadas, como por exemplo, iluminação do ambiente. Esses dados demonstram a dificuldade de conciliar resultados obtidos em escala de bancada com resultados próximos de ambientes reais.

5.2 Resultados da Segmentação de Instâncias com a Mask R-CNN

Nesta seção, apresenta-se os resultados obtidos com a implementação do modelo de segmentação. Os resultados preliminares indicaram que a segmentação de instâncias se mostrou a abordagem mais adequada para a individualização das quase-partículas e separação por classes de tamanho. A seguir, são descritos os resultados das máscaras de segmentação, das classes, das métricas de avaliação e dos histogramas gerados pelo Tensorboard.

5.2.1 Segmentação

Os treinamentos foram realizados de acordo com ajuste dos hiperparâmetros demonstrados na Seção 4.2.2. As máscaras de segmentação se mostraram convergentes com as bordas das quase-partículas e com as instâncias segmentadas separadamente, ressaltando a classe e a confiança da classe, como é possível observar na Figura 11. As bordas ficaram bem definidas, principalmente para as partículas maiores, como mostrado na Figura 12.



Figura 11 – A predição demonstra corretamente as máscaras como instâncias da mesma classe, a *bouding box* e os resultados de confiança preditos (própria autora)

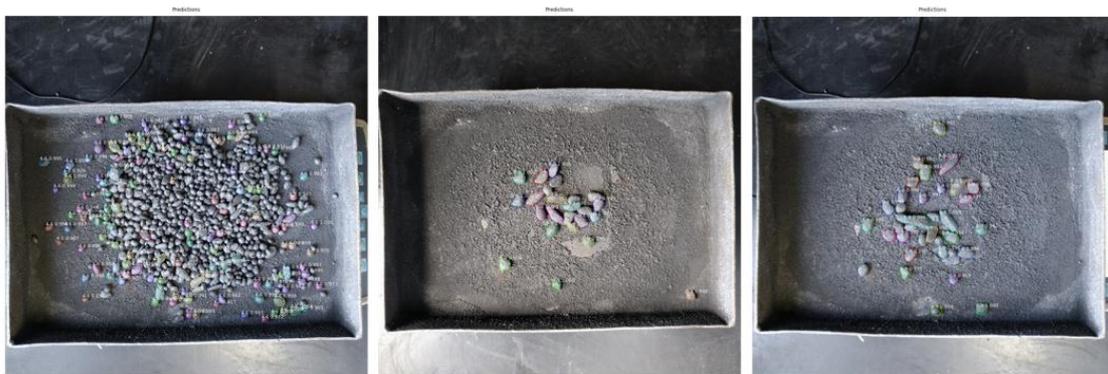


Figura 12 - A predição demonstra bordas bem definidas e resultados de confiança preditos (própria autora)

Os modelos também foram precisos em evitar a detecção e segmentação de partículas oclusas e sobrepostas umas às outras, fator que poderia levar ao erro na identificação da classe corretamente.

5.2.2 Tensorboard

A visualização da perda em geral e das perdas específicas da Mask R-CNN foram observadas com o auxílio da ferramenta Tensorboard. Nos gráficos, o eixo x representa o número de épocas de treinamento e o eixo y representa os valores de perda. Os valores de perda obtidos pela Mask RCNN podem ser observados na Tabela 9.

Tabela 9 - Valores de perda obtidos nos 4 treinamentos realizados. Os valores em destaque representam o menor valor para a perda selecionada (própria autora)

Perdas	TRAIN 1	TRAIN 2	TRAIN 3	TRAIN 4
	50 épocas	50 épocas	100 épocas	100 épocas
loss	0.7702	0.7541	0.5663	0.5728
mrcnn_bbox_loss	0.06478	0.06426	0.031	0.03347
mrcnn_class_loss	0.1252	0.1221	0.077	0.0768
mrcnn_mask_loss	0.1567	0.1616	0.1215	0.1267
rpn_bbox_loss	0.3335	0.3062	0.2627	0.259
rpn_class_loss	0.08497	0.08507	0.07416	0.0768
val_loss	0.8825	0.8783	0.8222	0.7977
val_mrcnn_bbox_loss	0.1002	0.1001	0.0879	0.08419
val_mrcnn_class_loss	0.1121	0.1259	0.1646	0.1446
val_mrcnn_mask_loss	0.1657	0.1649	0.1684	0.1641
val_rpn_bbox_loss	0.3875	0.3715	0.2952	0.3012
val_rpn_class_loss	0.1168	0.1159	0.1061	0.1035

Os valores totais das perdas do treinamento ficaram entre 0,5663 e 0,7702, no qual a menor perda foi registrada no treinamento 3, com o *backbone* ResNet101 e 100 épocas de treinamento. As perdas para detecção por caixas delimitadoras, geração das máscaras e classificação foram bastante positivos, sendo a maior perda para as caixas delimitadores da etapa de refinamento RPN.

Os valores de perda para a validação aparentaram bom resultado. Porém, a observação dos gráficos gerados pelo Tensorboard demonstrou flutuações, principalmente na previsão das classes na validação. Este comportamento pode estar associado ao pequeno número de rotulações por imagens e pela taxa de aprendizado. Assim, sugestões de melhoria deste comportamento de flutuação da validação seriam: anotação de maior quantidade de imagens e de objetos por imagem e diminuição da taxa de aprendizado durante o treinamento.

Os valores gerais de perda de validação ficaram entre 0,7977 e 0,8825, sendo a menor perda registrada também no treinamento 3, com o *backbone* ResNet101 e 100 épocas de treinamento.

O tempo de cada treinamento de acordo com o número de épocas é demonstrado na Tabela 10. Observa-se que número de épocas foi decisivo no tempo de execução do tempo de treinamento, enquanto que o número de camadas inicializadas pelo *backbone* das Mask RCNN não teve impacto significativo. Comparando-se com os valores de perda geral “loss” para o treinamento e “val_loss” para a validação, considera-se que houve uma diminuição da perda com o aumento do número de épocas treinadas, com a única contrapartida do aumento do tempo de execução do modelo.

Tabela 10 - Tempo de execução do treinamento de acordo com o *backbone* e número de épocas selecionado (própria autora)

Treinamento	Backbone	Épocas	Tempo de execução
TRAIN 1	Resnet101	50	1h 33m 29s
TRAIN 2	Resnet50	50	1h 33m 7s
TRAIN 3	Resnet101	100	3h 9m 13s
TRAIN 4	Resnet50	100	3h 9m 29s

Os valores gerais de perda para o treinamento (loss) e para a validação (val_loss) podem ser observados nos gráficos gerados pelo *Tensorboard*, nas Figuras 13 e 14. Os gráficos de treinamento demonstram uma descida suave para a convergência do modelo. O mesmo comportamento não foi observado em alguns dos gráficos de perda da etapa de validação, com flutuações constantes e aumento após determinado número de épocas.

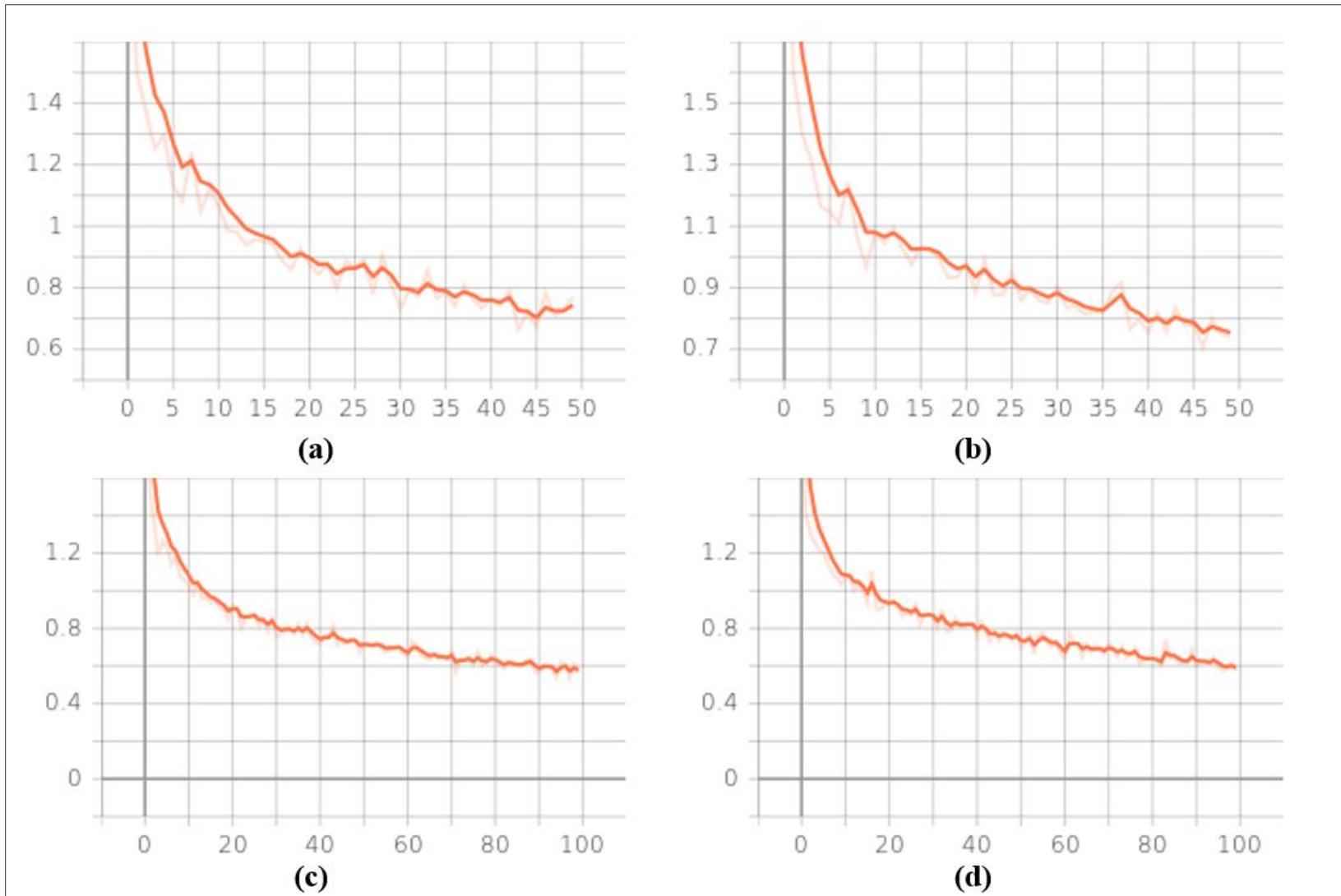


Figura 13 - Os gráficos representam a perda geral dos treinamentos: a) treinamento 1; b) treinamento 2; c) treinamento 3, e; d) treinamento 4 (própria autora)

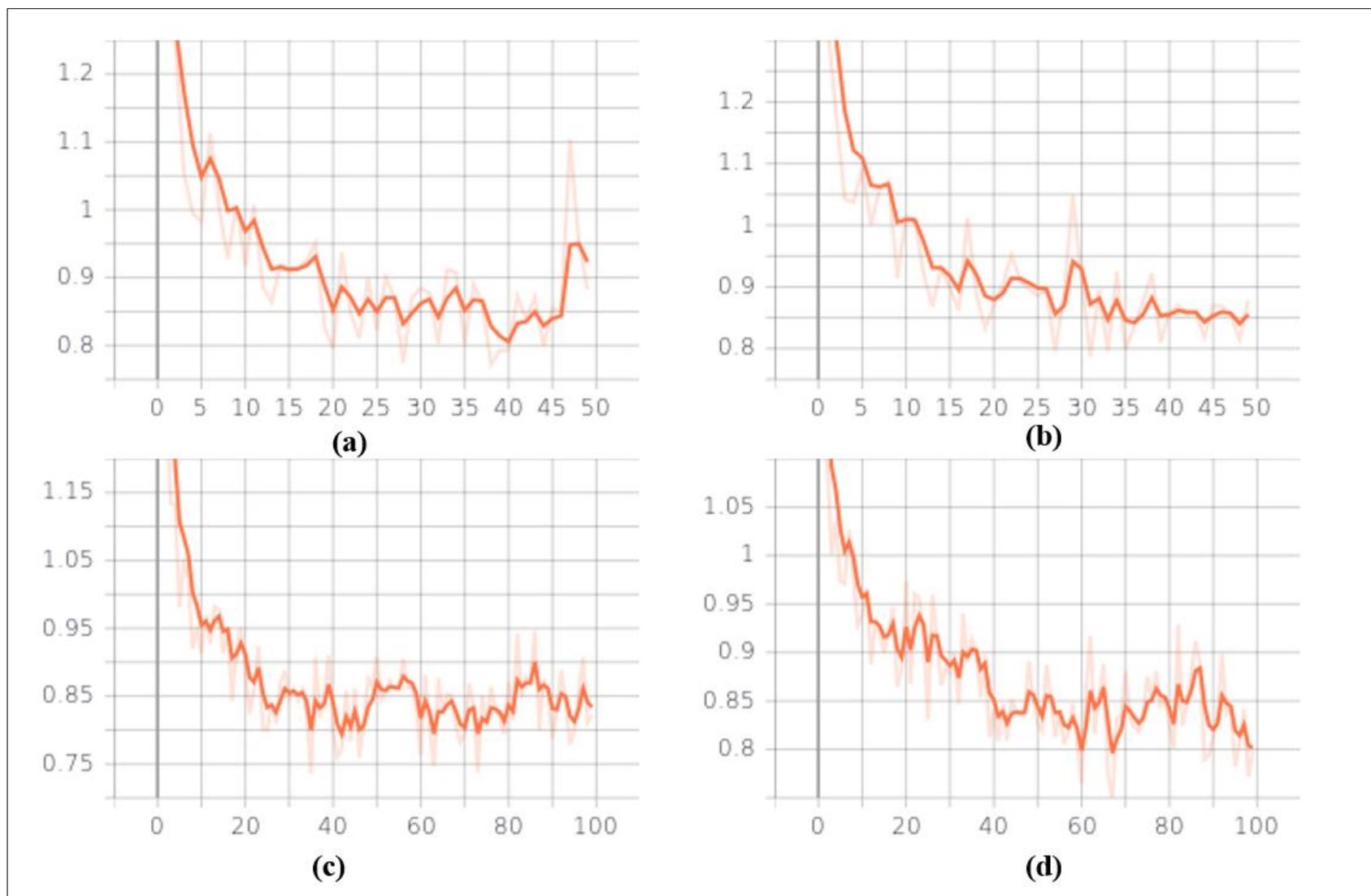


Figura 14 - Os gráficos representam a perda geral da validação: a) treinamento 1; b) treinamento 2; c) treinamento 3, e; d) treinamento 4 (própria autora)

5.2.3 Desempenho do modelo: mAP

Para avaliar o desempenho do modelo, utilizou-se a métrica mAP (média das métricas *Average Precision*, como apresentado na seção 4.2.4) com um IoU de 0,5. As métricas foram obtidas para as 46 imagens do conjunto de validação. Os valores obtidos em cada treinamento são mostrados na Tabela 11 a seguir:

Tabela 11 - Valores da métrica mAP obtidas para cada treinamento (própria autora)

Treinamento	Backbone	Épocas	mAP
TRAIN 1	Resnet101	50	0,2334
TRAIN 2	Resnet50	50	0,2585
TRAIN 3	Resnet101	100	0,2333
TRAIN 4	Resnet50	100	0,2511

Os valores da métrica mAP ficaram próximos para todos os treinamentos, e o treinamento 2 com a ResNet50 e 50 épocas obteve maior valor. Contudo, observou-se uma discrepância que impactou os valores de mAP. As imagens contendo partículas menores, das 3 primeiras classes de tamanho, continham muito mais partículas (objetos por imagem) do que as imagens com partículas maiores, pois as representações seguiram a amostragem: para cada amostra contendo 1Kg de material, proporcionalmente, há menos partículas de granulometria maior que 6mm.

Desta forma, com apenas 10 anotações em cada imagem no conjunto de validação, mesmo que o modelo previsse muito mais partículas do que anotado, o mAP destas imagens eram valores muito pequenos. Um exemplo pode ser visto na Figura 15, em que foram previstas muito mais partículas do que as rotuladas como verdade fundamental:

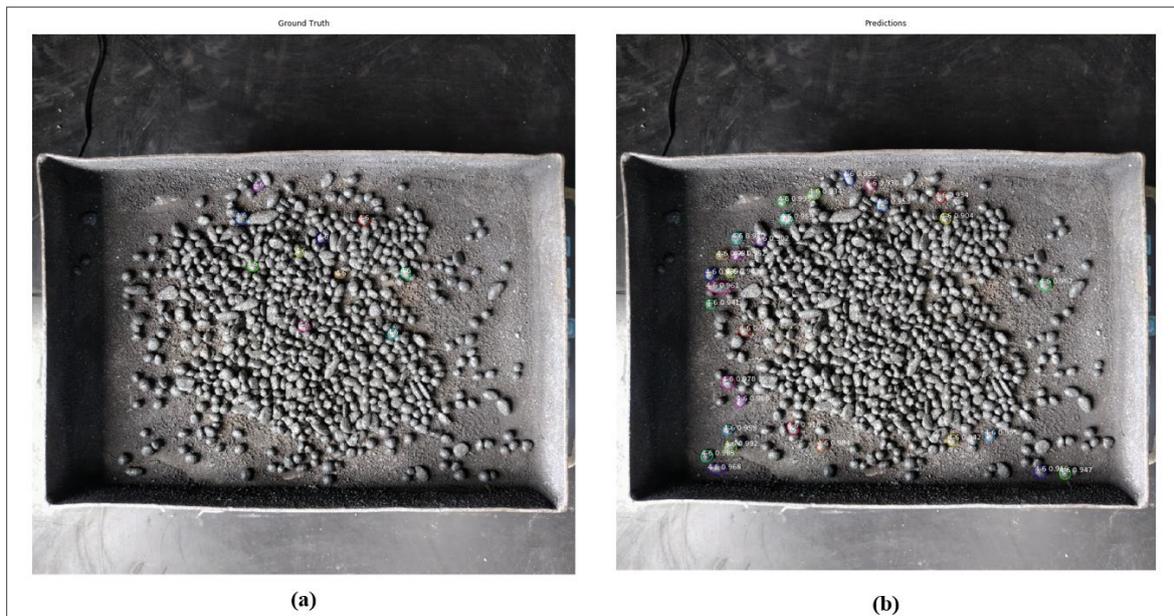


Figura 15 - Exemplo de imagem contendo partículas da classe '4-6' do treinamento 1. Em a) as partículas em destaque representam as partículas rotuladas e calculadas como verdade fundamental (*ground truth*), e em b) as partículas em destaque representam as partículas encontradas pelo modelo na predição (*predictions*) (própria autora)

O resultado prático desta discrepância é evidenciado na Figura 16. A imagem atingiu um AP de apenas 0,003 para um IoU de 0,5.

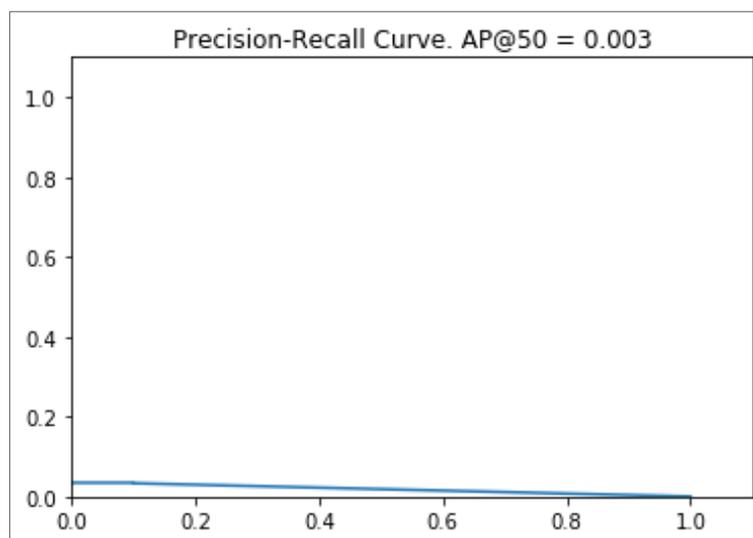


Figura 16 - Curva *precision-recall* para a imagem da Figura 15 (própria autora)

Para os casos de imagens contendo partículas maiores, como por exemplo, da classe '>9', os valores de AP por imagem foram superiores, chegando a 90% para algumas imagens. Isto se deve ao fato de que quase todas as partículas da imagem estavam rotuladas, logo, a chance de estar contida na predição também era maior. Um exemplo por ser visto na Figura 17:



Figura 17 - Exemplo de imagem contendo partículas da classe '8-9' do treinamento 4. Em a) as partículas em destaque representam as partículas rotuladas e calculadas como verdade fundamental (*ground truth*), e em b) as partículas em destaque representam as partículas encontradas pelo modelo na predição (*predictions*) (própria autora)

O resultado da métrica AP desta imagem é mostrado na curva *precision-recall* na Figura 18, com valor de AP igual a 0,551, ou seja, $AP_{50} = 55,1\%$, revelando a discrepância com o demonstrado na Figura 16.

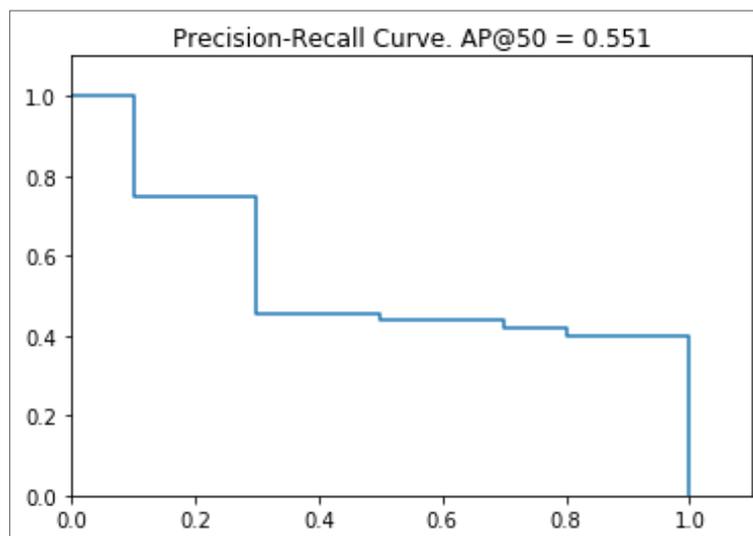


Figura 18 - Curva *precision-recall* para a imagem da Figura 17 (própria autora)

Assim, a arquitetura Mask R-CNN foi capaz de detectar e segmentar as quase-partículas, bem como, classificar corretamente as classes rotuladas, proporcionando resultados coerentes.

6 CONCLUSÃO

Neste trabalho, foram propostas duas aplicações em Redes Neurais Convolucionais para as quase-partículas do processo HPS: uma proposta de classificação implementada em um dispositivo *Edge AI*, e uma proposta de segmentação.

A primeira proposta consistiu na implementação de um algoritmo de classificação com inferência em um dispositivo *Edge AI* para classificar imagens do processo *Hybrid Pellet Sinter (HPS)* da indústria siderúrgica. Para isso, apresentou-se uma aplicação que visa reconhecer bandejas contendo quase-partículas, permitindo que se diferenciem de outros objetos e bandejas vazias. Os resultados indicam a viabilidade do sistema e a possibilidade de carregar o modelo produzido em um hardware especializado *Edge AI*.

Esta aplicação tem sua relevância ao implementar uma CNN em um dispositivo *edge computing*, em que o algoritmo deve analisar uma cena densa, com problemas como oclusão, sobreposições e alterações complexas de fundo. Apesar de haver amplas aplicações de *deep learning* em cenas densas, ainda há questões em aberto a serem resolvidas no processo de pesquisa. O uso de *hardware* de *edge computing* especializado melhora o desempenho da solução *edge*.

Para testar este aplicativo, usou-se o dispositivo embarcado SiPEED MaiX Dock, que possui infraestrutura de *hardware* e *software* para aprimorar o desenvolvimento de aplicativos *Edge AI*. A solução restringe custos e recursos, devido ao seu *hardware* especializado para criar aplicativos avançados de *deep learning*. Ele também se integra com modelos criados usando as principais estruturas de DL.

Testes com SiPEED permitiram o reconhecimento das amostras de quase-partículas em imagens sintéticas sem dificuldades, mesmo com a distribuição espaçada das quase-partículas/microaglomerados. Porém, testes com imagens simulando ambiente industrial obtiveram algumas falhas, evidenciadas pela queda no *recall* para 70%. Este resultado pode ser consequência de *overfitting* durante o treinamento ou, durante os testes, influências associadas principalmente ao brilho do dia, à oclusão entre partículas, homogeneidade de cores e sobreposição entre objetos. Esses problemas foram relatados em outros trabalhos com problemas semelhantes. Para melhorar este resultado, propõe-se em trabalhos futuros, elaborar um conjunto de dados contendo apenas imagens reais na classe principal o equilíbrio de dados entre as classes.

A segunda implementação consistiu na implementação do algoritmo Mask R-CNN para a detecção e segmentação das quase-partículas de acordo com a classe desejada. Para tanto, o

dataset foi rotulado com classes de acordo com a faixa granulométrica especificada. Foram realizados 4 treinamentos, com hiperparâmetros ajustados e customizados para o problema. A avaliação do modelo demonstrou boa detecção e segmentação, classes previstas corretamente e bordas bem definidas das quase-partículas. O modelo também teve bons resultados ao evitar a segmentação de partículas sobrepostas e oclusas, fator que poderia levar a previsão errada da classe.

A métrica mAP utilizada na avaliação do modelo da Mask R-CNN customizada para esse problema teve resultados entre 0,2333 e 0,2585 (em um intervalo de 0 a 1), para um IoU de 0,5. Na avaliação individual da métrica *Average Precision* de cada imagem, observou-se que os valores de AP foram menores para imagens que continham muitas partículas presentes na amostra e valores de AP maiores para imagens que continham poucas partículas presentes. Este fator pode estar associado a pequena quantidade de rótulos do conjunto de validação (apenas 10 por imagem), aumentando a probabilidade para imagens com poucas partículas de que o valor predito estivesse associado a uma verdade fundamental rotulada. Para resultados melhores de AP para estas classes, sugere-se em um trabalho futuro, que uma maior proporção de partículas seja rotulada no conjunto de validação e que os conjuntos de treinamento e validação possuam partículas de diversas classes rotuladas na mesma imagem.

As implementações sugerem um caminho importante para o desenvolvimento de uma implementação automática de análise granulométrica de quase-partículas. O algoritmo em *Edge AI* permite o reconhecimento imediato de uma amostra. Em seguida, uma implementação por segmentação fornece os intervalos granulométricos das partículas presentes na imagem, fornecendo um *baseline* para a classificação granulométrica automatizada das quase-partículas, processo que até o momento em que este trabalho foi escrito, é executado de forma manual por um operador da planta.

Este trabalho também teve a contribuição da elaboração do banco de imagens reais coletadas na empresa e rotuladas manualmente pela técnica de polígonos para treinamento e implementações de modelos *deep learning* de detecção e segmentação. Corrobora-se a esta justificativa, a dificuldade em se obter esses dados, uma vez que, não foi encontrada nenhuma base de dados dessa natureza. Além disso, os modelos amplamente apresentados na literatura se baseiam em base de dados disponíveis e, em grande parte, elaboradas em escala de bancada.

A utilização de dados industriais incrementam dificuldades, pois não são geradas em ambiente controlado e trazem consigo os desafios citados: fundo complexo; densidade, sobreposição e oclusão de objetos; variabilidade nas condições feitas, como iluminação; entre outras.

A partir dos resultados obtidos nesta etapa, foi possível levantar novas hipóteses de abordagens para melhorar o algoritmo a fim de se obter a o tamanho das quase-partículas presentes em uma amostra por faixas de tamanho. O desenvolvimento de um dispositivo Edge AI com *deep learning* pode trazer benefícios significativos, tanto na melhoria de processos quanto na inserção de processos siderúrgicos na Indústria 4.0.

Como contribuição deste trabalho, obteve-se a publicação do artigo “*Edge Deep Learning Applied to Granulometric Analysis on Quasi-particles from the Hybrid Pelletized Sinter (HPS) Process*” no *23rd International Conference on Enterprise Information Systems (ICEIS)* em 2021.

7 REFERÊNCIAS

ABDULLA, W. **Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow**. Github, 2017. Disponível em: <https://github.com/matterport/Mask_RCNN>. Acesso em: agosto de 2021.

ANDREW, NG. Lecture 11.4: **Machine Learning System Design | Trading Off Precision and Recall**. Disponível em: <<https://www.youtube.com/watch?v=W5meQnGACGo>>. Acesso em: agosto de 2021.

BASAL, N. **Computer Vision Techniques: Implementing Mask-R CNN on Malaria Cells Data**. 2019. Medium. Disponível em: <<https://medium.com/analytics-vidhya/computer-vision-techniques-implementing-mask-r-cnn-on-malaria-cells-data-e03b9fbeb6be>>. Acesso em: agosto de 2021.

BELLO, R.; MOHAMED, A. S. A.; TALIB, A. Z. Contour Extraction of Individual Cattle From an Image Using Enhanced Mask R-CNN Instance Segmentation Method. **IEEE Access**, v. 9, p. 56984-57000, 2021.

BOBBA, R. **Taming the Hyper-Parameters of Mask RCNN**. 2019. Analytics Vidhya - Community of Analytics and Data Science professionals. Disponível em: <<https://medium.com/analytics-vidhya/taming-the-hyper-parameters-of-mask-rcnn-3742cb3f0e1b>>. Acesso em: agosto de 2021.

CAMPOLO, C.; GENOVESE, G.; IERA, A.; MOLINARO, A. Virtualizing AI at the distributed edge towards intelligent IoT applications. **Journal of Sensor and Actuator Networks**, v. 10, n. 1, p. 13, 2021.

CHEN, D.; GUO, D.; LIU, S.; LIU, F. Microstructure instance segmentation from aluminum alloy metallographic image using different loss functions. **Symmetry**, v. 12, n. 4, p. 639, 2020.

CHEN, Z.; LIU, L.; QI, X.; GENG, J. Digital Mining Technology-Based Teaching Mode for Mining Engineering. **ijET**, v. 11, n. 10, p. 47-52, 2016.

COB-PARRO, A. C.; LOSADA-GUTIÉRREZ, C.; MARRÓN-ROMERA, M.; GARDELVICENTE, A.; BRAVO-MUÑOZ, I. Smart Video Surveillance System Based on Edge Computing. **Sensors**, v. 21, n. 9, p. 2958, 2021.

COHN, R.; ANDERSON, I.; PROST, T.; TIARKS, J.; WHITE, E.; HOLM, E. Instance Segmentation for Direct Measurements of Satellites in Metal Powders and Automated Microstructural Characterization from Image Data. **arXiv preprint arXiv:2101.01585**, 2021.

CORNETTA, G.; TOUHAFI, A. Design and Evaluation of a New Machine Learning Framework for IoT and Embedded Devices. **Electronics**, v. 10, n. 5, p. 600, 2021.

DE CESARO JÚNIOR, T. **InsectCV: um sistema para detecção de insetos em imagens digitais**. Dissertação (Mestrado em Computação Aplicada). 2020. 82f. Universidade de Passo Fundo, Passo Fundo, 2020.

DENG, S.; ZHAO, H.; FANG, W.; YIN, J.; DUSTDAR, S.; ZOMAYA, A. Y. Edge intelligence: the confluence of edge computing and artificial intelligence. **IEEE Internet of Things Journal**, v. 7, n. 8, p. 7457-7469, 2020.

DIAS, I. S. M. **Sistema de controle granulométrico de pelotas de minério de ferro**. 2018. 95f. Dissertação (mestrado profissional em Instrumentação, Controle e Automação de Processos de Mineração). Universidade Federal de Ouro Preto, Ouro Preto, 2018.

GAD, A. F. **Evaluating Object Detection Models Using Mean Average Precision (mAP)**. 2020. PaperspaceBlog. Disponível em: <<https://blog.paperspace.com/mean-average-precision/>>. Acesso em: agosto de 2021.

GAO, G.; GAO, J.; LIU, Q.; WANG, Q.; WANG, Y. CNN-based density estimation and crowd counting: A survey. **arXiv preprint arXiv:2003.12783**, 2020.

GIRSHICK, R. Fast R-CNN. *In: Proceedings of the IEEE international conference on computer vision*. 2015. p. 1440-1448.

GONTIJO, M. D. Análise granulométrica por imagem de amostras ultrafinas. **Revista Engenharia de Interesse Social**, v. 1, n. 3, 2018.

HE, K.; GKIOXARI, G.; DÓLLAR, P.; GIRSHHICK, R. Mask r-cnn. *In: Proceedings of the IEEE international conference on computer vision*. 2017. p. 2961-2969.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 770-778.

HUI, J. **mAP (mean Average Precision) for Object Detection**. 2018. Medium. Disponível em: <<https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>>. Acesso em: agosto de 2021.

JANUZZI, A. **Análise da aglomeração a frio no processo HPS (Hybrid Pelletized Sínter) com ênfase nas matérias-primas envolvidas**. 2008. 145 f. Dissertação (mestrado em Engenharia Metalúrgica e de Minas). Universidade Federal de Minas Gerais, Belo Horizonte, 2008.

JOHNSON, J.W. Adapting MASK-RCNN for automatic nucleus segmentation. *arXiv* **2018**, arXiv:1805.00500.

JÚNIOR, T. C; RIEDER, R. Uma Implementação Baseada em Mask R-CNN para Detecção de Insetos em Imagens Digitais. *In: Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*. SBC, 2020. p. 139-142.

K210 Datasheet. 2018. Disponível em: <https://s3.cn-north-1.amazonaws.com.cn/dl.kendryte.com/documents/kendryte_datasheet_20181011163248_en.pdf>. Acesso em: Abril de 2021.

KARRAS, K.; PALLIS, E.; MASTORAKIS, G.; NIKOLOUDAKIS, Y.; BATALLA, J. M.; MAVROMOUSTAKIS, C. X.; MARKAKIS, E. A hardware acceleration platform for AI-based inference at the edge. **Circuits, Systems, and Signal Processing**, v. 39, n. 2, p. 1059-1070, 2020.

KERESZTES, B.; ABDELGHAFOR, F.; RANDRIAMANGA, D.; DA COSTA, J.-P.; GERMAIN, C. Real-time fruit detection using deep neural networks. *In: 14th International Conference on Precision Agriculture*. 2018.

KINNUNEN, P. H.-M.; KAKSONEN, A. H. Towards circular economy in mining: Opportunities and bottlenecks for tailings valorization. **Journal of cleaner production**, v. 228, p. 153-160, 2019.

KLIPPEL, E.; OLIVEIRA, R.; MASLOV, D.; BIANCHI, A.; SILVA, S. E.; GARROCHO, C. Towards to an Embedded Edge AI Implementation for Longitudinal Rip Detection in Conveyor Belt. *In: Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. SBC, 2020. p. 97-102.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. *In: Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE, 2010. p. 253-256.

LECUN, Y; BENGIO, Y; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436-444, 2015.

LI, E.; ZENG, L.; ZHOU, Z.; CHEN, X. Edge AI: On-demand accelerating deep neural network inference via edge computing. **IEEE Transactions on Wireless Communications**, v. 19, n. 1, p. 447-457, 2019.

LIMA, A. J. A. S. Caracterização tecnológica de uma mistura de *sinter feed* e *pellet feed* para uso em processo HPS. 125 f. 2019. Dissertação (Mestrado em Engenharia de Materiais) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2019.

LIN, X.; LI, J.; WU, J.; LIANG, H.; YANG, W. Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach. **IEEE Transactions on Industrial Informatics**, v. 15, n. 12, p. 6367-6378, 2019.

LIN, Y. C.; CHEN, W. H.; KUO, C. H. Implementation of Pavement Defect Detection System on Edge Computing Platform. **Applied Sciences**, v. 11, n. 8, p. 3725, 2021.

LIU, C.; SU, X.; LI, C. Edge Computing for Data Anomaly Detection of Multi-Sensors in Underground Mining. **Electronics**, v. 10, n. 3, p. 302, 2021.

LIU, M.; DONG, J.; DONG, X.; YU, H.; QI, L. Segmentation of lung nodule in CT images based on mask R-CNN. In: **2018 9th International Conference on Awareness Science and Technology (iCAST)**. IEEE, 2018. p. 1-6.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2015. p. 3431-3440.

MAIXPY: Introduction to MaixPy documentation. Disponível em: <https://wiki.sipeed.com/soft/maixpy/en/>. Acesso em: agosto de 2021.

MARDONOVA, M.; CHOI, Y. Review of wearable device technology and its applications to the mining industry. **Energies**, v. 11, n. 3, p. 547, 2018.

MAZZIA, V.; KHALIQ, A.; SALVETTI, F.; CHIABERGE, M. REAL-time apple detection system using embedded systems with hardware accelerators: an edge AI application. **IEEE Access**, v. 8, p. 9102-9114, 2020.

MEIRA, N.; SILVA, M.; OLIVEIRA, R.; SOUZA, A.; D'ANGELO, T.; VIEIRA, C. Edge Deep Learning Applied to Granulometric Analysis on Quasi-particles from the Hybrid Pelletized Sinter (HPS) Process. *In: Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS*. 2021. ISBN 978-989-758-509-8 ISSN 2184-4992. P. 527-535. DOI: 10.5220/0010458805270535. Disponível em: <<https://www.scitepress.org/Link.aspx?doi=10.5220/0010458805270535>>. Acesso em: agosto de 2021.

MONCHOT, P.; COQUELIN, L.; GUERROUDJ, K.; FELTIN, N.; DELVALLÉE, A.; CROUZIER, L.; FISCHER, N. Deep Learning Based Instance Segmentation of Titanium Dioxide Particles in the Form of Agglomerates in Scanning Electron Microscopy. *Nanomaterials*, v. 11, n. 4, p. 968, 2021.

OHBUCHI, E. Low power AI hardware platform for deep learning in edge computing. *In: 2018 IEEE CPMT Symposium Japan (ICJS)*. IEEE, 2018. p. 89-90.

PASSOS, L. A. S.; MOREIRA, J. L.; JORGE, A.; CAVALCANTE, M. V. S. Melhoria no desempenho do processo de produção de pelotas de minério de ferro em discos de pelotização pela utilização de sistemas otimizantes com lógica nebulosa*. **44º Seminário de Redução de Minério de Ferro e Matérias-primas, 15º Simpósio Brasileiro de Minério de Ferro e 2º Simpósio Brasileiro de Aglomeração de Minério de Ferro**. Belo Horizonte, 2014.

REN, S.; HE, K.; GIRSHICK, R.; SUN, R. Faster R-CNN: Towards real-time object detection with region proposal networks. **arXiv preprint arXiv:1506.01497**, 2015.

ROBBEN, C.; WOTRUBA, H. Sensor-based ore sorting technology in mining—Past, present and future. *Minerals*, v. 9, n. 9, p. 523, 2019.

SALEEM, M. H.; POTGIETER, J.; ARIF, K. M. Plant disease detection and classification by deep learning. *Plants*, v. 8, n. 11, p. 468, 2019.

SANTOS, R. B. M.; AUGUSTO, K. S.; PACIORNICK, S.; DOMINGUES, A. L. A. An image analysis system for automatic characterisation of iron ore sintering quasiparticles. *Mineral Processing and Extractive Metallurgy*, p. 1-9, 2019.

SEEDSTUDIO, 2021. Disponível em: <<https://www.seedstudio.com/Sipeed-M1-dock-suit-M1-dock-2-4-inch-LCD-OV2640-K210-Dev-Board-1st-RV64-AI-board-for-Edge-Computing.html>>. Acesso em: agosto de 2021.

SHAH, T. Measuring Object Detection models — mAP — What is Mean Average Precision? 2018. Medium. Disponível em: <<https://towardsdatascience.com/what-is-map-understanding->

[the-statistic-of-choice-for-comparing-object-detection-models-1ea4f67a9dbd](#)>. Acesso em: agosto de 2021.

SHIBUTA, Y.; OHNO, M.; TAKAKI, T. Advent of Cross-Scale Modeling: High-Performance Computing of Solidification and Grain Growth. **Advanced Theory and Simulations**, v. 1, n. 9, p. 1800065, 2018.

SINOVIEV, V. V.; OKOLNISHNIKOV, V. V.; STARODUBOV, A. N.; DOROFEEV, M. U. Approach to effectiveness evaluation of robotics technology in mining using discrete event simulation. **International Journal of Mathematics and Computers in Simulation**, v. 10, p. 123-128, 2016.

SOLOY, A.; TURKI, I.; FOURNIER, M.; COSTA, S.; PEUZIAT, B.; LECOQ, N. A Deep Learning-Based Method for Quantifying and Mapping the Grain Size on Pebble Beaches. **Remote Sensing**, v. 12, n. 21, p. 3659, 2020.

SOUZA, L. E. R. **Medição de granulometria de minério de ferro através de imagens em circuito de britagem primária**. 2020. 99 f. Dissertação (Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração) - Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2020.

TAN, R. J. **Breaking Down Mean Average Precision (mAP)**. 2019. Medium. Disponível em: <<https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>>. Acesso em: agosto de 2021.

TU, S.; YUAN, W.; LIANG, Y.; WANG, F.; WAN, H. Automatic detection and segmentation for group-housed pigs based on PIGMS R-CNN. **SENSORS**, V. 21, N. 9, P. 3251, 2021.

VUOLA, A. O.; AKRAM, S. U.; KANNALA, J. Mask-RCNN and U-net ensembled for nuclei segmentation. *In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019. p. 208-212.

WANG, X.; HAN, Y.; LEUNG, V. C.; NIYATO, D.; YAN, X.; CHEN, X. Convergence of edge computing and deep learning: A comprehensive survey. **IEEE Communications Surveys & Tutorials**, v. 22, n. 2, p. 869-904, 2020.

WANG, X.; HAN, Y.; WANG, C.; ZHAO, Q.; CHEN, X.; CHEN, M. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. **IEEE Network**, v. 33, n. 5, p. 156-165, 2019.

WU, Q.; FENG, D.; CAO, C.; XIAODONG, Z.; FENG, Z.; WU, J. Improved Mask R-CNN for Aircraft Detection in Remote Sensing Images. **Sensors**, v. 21, n. 8, p. 2618, 2021.

XI, D.; QIN, Y.; WANG, Y. Vision Measurement of Gear Pitting Under Different Scenes by Deep Mask R-CNN. **Sensors**, v. 20, n. 15, p. 4298, 2020.

XIE, S.; GIRSHICK, R.; DOLLÁR, P.; TU, Z.; HE, K. Aggregated residual transformations for deep neural networks. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 1492-1500.

XU, B.; WANG, W.; FALZON, G.; KWAN, P.; GUO, L.; SUN, Z.; LI, C. Livestock classification and counting in quadcopter aerial images using Mask R-CNN. **International Journal of Remote Sensing**, v. 41, n. 21, p. 8121-8142, 2020.

ZHANG, L.; WU, J.; FAN, Y.; GAO, H.; SHAO, Y. An efficient building extraction method from high spatial resolution remote sensing images based on improved mask R-CNN. **Sensors**, v. 20, n. 5, p. 1465, 2020.

ZHANG, Q.; LIU, Y.; GONG, C.; CHEN, Y.; YU, H. Applications of deep learning for dense scenes analysis in agriculture: A review. **Sensors**, v. 20, n. 5, p. 1520, 2020.

ZHANG, R.; CHENG, C.; ZHAO, X.; LI, X. Multiscale Mask R-CNN–Based Lung Tumor Detection Using Pet Imaging. **MOL. IMAGING** 2019, 18, 1536012119863531.

ZHAO, K.; KANG, J.; JUNG, J.; SOHN, G. Building extraction from satellite images using mask R-CNN with building boundary regularization. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018. p. 247-251.

ZHOU, X., FANG, B., QIAN, J., XIE, G., DENG, B., QIAN, J. Data Driven Faster R-CNN for Transmission Line Object Detection. *In: Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health*. Springer, Singapore, 2019. p. 379-389. 2019-a.

ZHOU, Z.; CHEN, X.; LI, E.; ZENG, L.; LUO, K.; ZHANG, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. **Proceedings of the IEEE**, v. 107, n. 8, p. 1738-1762, 2019-b.

ZOBNIN, N. N.; TORGOVETS, A. K.; PIKALOVA, I. A.; YUSSUPOVA, Y. S.; ATAKISHIYEV, S. A. Influence of Thermal Stability of Quartz and the Particle Size distribution of Burden Materials on the Process of Electrothermal Smelting of Metallurgical Silicon. **Oriental Journal of Chemistry**, v. 34, n. 2, p. 1120, 2018.