



INSTITUTO TECNOLÓGICO VALE



**Programa de Pós-Graduação em Instrumentação, Controle e  
Automação de Processos de Mineração (PROFICAM)  
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)  
Associação Instituto Tecnológico Vale (ITV)**

**Dissertação**

**SISTEMA AUTOMÁTICO PARA A INSPEÇÃO VISUAL DE TRANSPORTADORES  
DE CORREIA POR MEIO DE REDES NEURAIS CONVOLUCIONAIS**

**André Almeida Santos**

**Ouro Preto  
Minas Gerais, Brasil  
2020**

**André Almeida Santos**

**SISTEMA AUTOMÁTICO PARA A INSPEÇÃO VISUAL DE TRANSPORTADORES  
DE CORREIA POR MEIO DE REDES NEURAIIS CONVOLUCIONAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Frederico Gadelha Guimarães, D.Sc.

Coorientador: Prof. Agnaldo José da Rocha Reis, D.Sc.

Ouro Preto  
2020



## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S237s Santos, André Almeida .

Sistema automático para a inspeção visual de transportadores de correia por meio de redes neurais convolucionais. [manuscrito] / André Almeida Santos. - 2020.

59 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Frederico Gadelha Guimarães.

Coorientador: Prof. Dr. Agnaldo José da Rocha Reis.

Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.

Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Aprendizado do computador . 2. Redes neurais (Computação). 3. Correias transportadoras. I. Guimarães, Frederico Gadelha. II. Reis, Agnaldo José da Rocha. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB: 1716



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
ESCOLA DE MINAS  
PROGR. POS GRAD. PROF. INST. CONT. E AUT. PROCESSOS DE MIN.



**FOLHA DE APROVAÇÃO**

**André Almeida Santos**

**Sistema Automático para a Inspeção Visual de Transportadores de Correia por meio de Redes Neurais Convolucionais**

Membros da banca

Frederico Gadelha Guimarães - Doutor - UFMG (orientador)  
Agnaldo José da Rocha Reis - Doutor - UFOP (coorientador)  
Sílvia Grasiella Moreira Almeida - Doutora - IFMG-OP (membro externo)  
Alan Kardek Rêgo Segundo - Doutor - UFOP (membro interno)  
Gustavo Pessin - Doutor - ITV (membro interno)

Versão final

Aprovada em 26 de Agosto de 2020

De acordo,

Professor Coorientador (p/ Professor Orientador).



Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis**, COORDENADOR DO CURSO DE POS-GRADUACAO EM INSTRUMENTACAO, CONTROLE E AUTOMACAO DE PROC DE MINERACAO, em 31/08/2020, às 09:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0078416** e o código CRC **D5C84512**.

**Referência:** Caso responda este documento, indicar expressamente o Processo nº 23109.006108/2020-55

SEI nº 0078416

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000  
Telefone: - www.ufop.br

*Aos meus pais, Luís Carlos e Vera  
Lúcia, por todo esforço realizado  
até hoje para que eu chegasse até  
aqui.*

# Agradecimentos

A Deus que iluminou o meu caminho durante toda a minha vida até este momento.

Aos meus pais, irmão e toda minha família que sempre deram apoio aos meus estudos

Aos amigos e colegas do ITV/UFOP e todos os demais que estiveram presente em reuniões, grupos de estudos, viagens e momentos compartilhados.

A todos os professores do ITV/UFOP pelo conhecimento transmitido durante esses dois anos.

Agradeço ao Frederico Gadelha e Agnaldo Reis pela orientação e coorientação, e contribuição positiva ao trabalho.

Agradeço aos pesquisadores do ITV pelo apoio nos projetos realizados e oportunidades de desenvolvimento.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

*“O orgulho vem antes da  
destruição; o espírito altivo, antes  
da queda.”(Provérbios 16:18)*

## Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

### SISTEMA AUTOMÁTICO PARA A INSPEÇÃO VISUAL DE TRANSPORTADORES DE CORREIA POR MEIO DE REDES NEURAIS CONVOLUCIONAIS

André Almeida Santos

Agosto/2020

Orientadores: Frederico Gadelha Guimarães

Agnaldo José da Rocha Reis

Os Transportadores de Correia são o meio de transporte mais difundido para grandes quantidades de materiais no setor de mineração. Portanto, métodos autônomos que podem ajudar os seres humanos a realizar a inspeção dos transportadores são uma grande preocupação para as empresas. Projetos que buscam solucionar problemas de segurança nas inspeções de transportadores são necessários e de grande valia para as empresas. Este trabalho apresenta um detector visual novo e automático que reconhece o acúmulo de sujeira nas estruturas dos transportadores de correia, uma das tarefas dos inspetores de manutenção. O sistema proposto envolve o treinamento de uma rede neural convolucional a partir de imagens RGB. O uso da técnica de aprendizado por transferência com redes consolidadas para classificação de imagens mostrou-se eficaz para este objetivo com a comparação de dois cenários diferentes. O melhor cenário apresentou uma acurácia média de 0,892 com um F-1 score de 0,836 para o reconhecimento de sujeira. Um experimento de validação de campo serviu para avaliar o desempenho do sistema proposto em uma tarefa de classificação em tempo real. Nesta etapa, foi avaliado um recall de 0,77 com F-1 score de 0,75 para o reconhecimento de sujeira.

**Palavras-chave:** Aprendizado de Máquina, Redes Neurais Artificiais, Transportador de Correia.

**Macrotema:** Mina; **Linha de Pesquisa:** Tecnologias da Informação, Comunicação e Automação Industrial; **Tema:** Inspeção Automática de Ativos.

## **Abstract**

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

# **AUTOMATIC SYSTEM FOR VISUAL INSPECTION OF BELT CONVEYORS BY CONVOLUTIONAL NEURAL NETWORKS**

André Almeida Santos

August/2020

Advisors: Frederico Gadelha Guimarães

Agnaldo José da Rocha Reis

Conveyor belts are the most widespread means of transportation for large quantities of materials in the mining sector. Therefore, autonomous methods that can help human beings carry out its inspection are a major concern for companies. Projects that seek to solve safety problems in the inspection of transporters are necessary and of great value for companies. This master's project features a new, automatic visual detector that recognizes the dirt buildup in the structures of belt conveyors, which is one of the tasks of maintenance inspectors. The proposed system involves the training of a convolutional neural network from RGB images. The use of the transfer learning technique with consolidated networks for image classification proved to be very effective for this purpose. Two different scenarios were analyzed. The best scenario presented an average accuracy of 0.892 with an F-1 score of 0.836 for dirt recognition. A field validation experiment was used to evaluate the performance of the proposed system in a classification task in real time. In this step, a recall of 0.77 was evaluated with an F-1 score of 0.75 for the recognition of dirt.

**Keywords:** Machine Learning, Artificial Neural Networks, Conveyor Belt.

**Macrotheme:** Mine; **Research Line:** Information Technologies, Communication and Industrial Automation; **Theme:** Automatic Asset Inspection.

# Lista de Figuras

Figura 3.1	RGB. . . . .	23
Figura 3.2	Sistema nervoso em três estágios. . . . .	23
Figura 3.3	Modelo não linear de um neurônio. . . . .	24
Figura 3.4	Rede Feedforward conectada. . . . .	25
Figura 3.5	Neurônios de um filtro $f$ que compartilham os mesmos pesos sinápticos. . .	28
Figura 3.6	Rede convolucional para processamento de imagem. . . . .	28
Figura 3.7	Campo receptivo local. . . . .	29
Figura 3.8	Deslocamento do campo receptivo local. . . . .	30
Figura 3.9	Visualização tridimensional de uma camada de convolução. . . . .	30
Figura 3.10	Operação de <i>Max Pooling</i> . . . . .	31
Figura 3.11	Três maneiras pelas quais a transferência pode melhorar o aprendizado. . . .	32
Figura 3.12	Matriz de confusão de um problema binário. . . . .	34
Figura 4.1	Estrutura do TC com acúmulo de resíduos. . . . .	35
Figura 4.2	Coleta de imagens em situações diversas. . . . .	36
Figura 4.3	Imagem da estrutura após extração da região central. . . . .	37
Figura 4.4	Quantidade de imagens por classe. . . . .	37
Figura 4.5	Divisão dos dados para o método de validação cruzada k-pastas. . . . .	38
Figura 4.6	Data Augmentation aplicados nas imagens. . . . .	38
Figura 4.7	Arquitetura original do modelo VGG16. . . . .	40
Figura 4.8	Arquitetura original do modelo Resnet18. . . . .	40
Figura 4.9	Cenários do treinamento. . . . .	41
Figura 4.10	Sequência do processo de validação. . . . .	43
Figura 4.11	GAP na última camada convolucional. . . . .	43
Figura 5.1	Boxplot com acurácia dos modelos no cenário 1. . . . .	46
Figura 5.2	Curva ROC para os modelos no cenário 1. . . . .	47
Figura 5.3	Boxplot com acurácia dos modelos no cenário 2. . . . .	49
Figura 5.4	Curva ROC para os modelos no cenário 2. . . . .	50
Figura 5.5	Boxplot. . . . .	51
Figura 5.6	Primeira validação em campo, com uma classificação <i>Limpo</i> em (a) e uma classificação <i>Sujo</i> em (b). . . . .	51



Figura 5.7	Quadros de validação de campo com probabilidades de classificação em tempo real. . . . .	52
Figura 5.8	Matriz de Confusão da Validação em Campo. . . . .	52
Figura 5.9	Localização discriminativa aplicada ao modelo. . . . .	53

# Lista de Tabelas

Tabela 4.1	parâmetros de treinamento para os modelos. . . . .	41
Tabela 5.1	Erro e acurácia dos modelos VGG16 and ResNet18 no cenário 1. . . . .	45
Tabela 5.2	Métricas para o cenário 1. . . . .	46
Tabela 5.3	Erro e acurácia dos modelos VGG16 and ResNet18 no cenário 2. . . . .	48
Tabela 5.4	Métricas para o cenário 2. . . . .	48

# Lista de Siglas e Abreviaturas

**AUC** Area Under the ROC Curve

**CAM** Mapas de Ativação de classe

**CE** Cross Entropy

**CNN** *Convolutional Neural Network*

**DA** *Data Augmentation*

**FN** False Negative

**FP** False Positive

**FPS** Quadros por Segundo

**GAP** Global Average Pooling

**GD** Gradiente Descendente

**IA** Inteligência Artificial

**ILSVRC** ImageNet Large Scale Visual Recognition Challenge

**PCA** Análise de Componentes Principais

**PDI** Processamento Digital de Imagem

**RNA** Rede Neural Artificial

**RNC** Rede Neural Convolucional

**ROC** Receiver Operating Characteristic

**ROI** Região de Interesse

**SBAI** Simpósio Brasileiro de Automação Inteligente

**SGD** *Gradiente Descendente Estocástico*

**SVM** *Support Vector Machine*

**TC** Transportador de Correia

**TL** *Transfer Learning*

**TN** True Negative

**TP** True Positive

# Lista de Unidades de Medidas

*GiB* gibibyte

*G* giba

*Hz* hertz

*km* quilômetro

*pp* pontos percentuais

# Sumário

<b>1</b>	<b>Introdução</b>	<b>16</b>
1.1	Objetivos . . . . .	17
1.1.1	Objetivos específicos . . . . .	17
1.2	Motivação . . . . .	17
1.3	Publicações e participação em eventos . . . . .	18
1.3.1	Simpósio Brasileiro de Automação Inteligente . . . . .	18
1.3.2	Mostra de Pós-Graduação do Encontro de Saberes . . . . .	18
1.4	Artigo publicado . . . . .	18
1.5	Contribuições internas no ITV . . . . .	18
1.5.1	Jornada de Inovação . . . . .	19
1.5.2	Treinamento Introdutório em Programação em Python . . . . .	19
1.6	Estrutura do trabalho . . . . .	19
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>20</b>
<b>3</b>	<b>Referencial Teórico</b>	<b>22</b>
3.1	Processamento Digital de Imagens . . . . .	22
3.2	Redes Neurais . . . . .	23
3.2.1	Modelo de um Neurônio . . . . .	24
3.2.2	Rede Feedforward Multicamada . . . . .	24
3.2.3	Treinamento de Rede Feedforward . . . . .	25
3.3	Rede Neural Convolucional . . . . .	27
3.3.1	Estrutura de uma rede neural convolucional . . . . .	29
3.4	Aprendizado por transferência . . . . .	31
3.5	Avaliação de um modelo de Inteligência Artificial . . . . .	32
<b>4</b>	<b>Materiais e Métodos</b>	<b>35</b>
4.1	Coleta de dados . . . . .	35
4.2	Pré-processamento dos dados . . . . .	37
4.3	Arquiteturas das redes neurais convolucionais . . . . .	39
4.3.1	VGG16 . . . . .	39

4.3.2	ResNet18 . . . . .	40
4.4	Treinamento . . . . .	41
4.5	Validação em campo . . . . .	42
4.6	Localização discriminativa . . . . .	42
<b>5</b>	<b>Resultados e Discussão</b>	<b>45</b>
5.1	Cenário 1: RNC como extrator fixo de características . . . . .	45
5.2	Cenário 2: RNC com ajuste fino . . . . .	47
5.3	Validação de Campo . . . . .	49
<b>6</b>	<b>Conclusão</b>	<b>54</b>
<b>7</b>	<b>Trabalhos Futuros</b>	<b>55</b>
	<b>Referências Bibliográficas</b>	<b>56</b>

# 1. Introdução

O Brasil possui uma das maiores reservas minerais do mundo, sendo um dos principais produtores e exportadores mundiais de minério de ferro. Nesse contexto, a Vale é a maior produtora mundial de minério de ferro e pelotas, matérias-primas essenciais para a fabricação de aço (VALE, 2018a). De acordo com Toffolo (2009), tecnologias que permitam melhorar atividades relacionadas à extração, transporte ou venda de minérios são de grande interesse no cenário nacional.

No setor minero-metalúrgico, observa-se que o Transportador de Correia (TC) constitui o meio mais difundido de transporte para grandes quantidades de materiais a granel, reduzindo, desta forma, a quantidade de caminhões e o custo deste tipo de serviço (MALAGONI e DOS SANTOS, 2014). A título de exemplo, a Vale possui aproximadamente 1200 quilômetros (*km*) de TC. Somente em Carajás, a maior mina de minério de ferro do mundo, são 85 *km* desta estrutura (VALE, 2018b). Guastala (2013) cita o TC como um dos ativos mais importantes da empresa Vale. Para Malagoni e dos Santos (2014), o TC detém uma posição dominante no transporte de materiais devido às suas inerentes vantagens, como economia e segurança de operação, confiabilidade, versatilidade e enorme gama de capacidades.

Muitos materiais a granel transportados nas correias são um pouco pegajosos. Partes do material aderem à superfície de transporte da correia e não são descarregadas com o restante da carga no ponto de descarga. O material residual eventualmente cai em vários pontos ao longo da linha da correia, acumulando e exigindo limpeza para evitar falhas. A recuperação pode levar ao desgaste excessivo, acúmulo de roldanas de retorno, possíveis danos ao forçar a correia contra alguma parte da estrutura de suporte e efeitos adversos na operação do TC e na eficiência da planta. Além disso, o acúmulo de material no solo ou nuvens de poeira no ar representa um risco à saúde e à segurança (CONVEYOR EQUIPMENT MANUFACTURES ASSOCIATION, 2007).

O TC pode ter falhas operacionais, como rasgo longitudinal, desvio e danos na superfície da borracha da correia. Independentemente da causa, quando um TC está fora de serviço de maneira não planejada, pode-se esperar altos custos de manutenção e perda de produção (YANG *et al.*, 2014). Para Carvalho Júnior (2018), 96 horas por *km* de TC é o tempo médio de inatividade por evento. Para Matthee (2009), as empresas estão sob enorme pressão para reduzir custos e manter a produção. As plantas com TC geralmente contêm milhares de rolamentos que são difíceis de inspecionar de maneira rentável usando apenas um único método de inspeção.

Neste sentido, o projeto ROSI (Dispositivo Robótico para Inspeção de Rolos de Correia Transportadora) busca solucionar os problemas mencionados com o desenvolvimento de uma plataforma robótica equipada com um braço manipulador, onde é instalado um conjunto de sensores (microfone, acelerômetro, laser, câmeras térmica e RGB) para realizar as inspeções necessárias. O objetivo é usar a plataforma robótica móvel para remover o operador de tarefas perigosas (GARCIA *et al.*, 2019). O protótipo inicial foi composto com os seguintes equi-



pamentos: (i) uma plataforma móvel com esteiras laterais e dois pares de braços de alavanca rotativos para superar obstáculos e ajustar a altura da plataforma; (ii) um manipulador robótico SCHUNK LWA 4; (iii) uma câmera FLIR AX8 RGB e Térmica com proteção IP-67; (iv) um scanner a laser planar de alta velocidade Hokuyo UTM-30LX; (v) um microfone convencional sensível a sons de 20 a 20k Hz e (vi) sistema de IMU + GPS aXsens Mti-G-710.

Com a identificação correta de possíveis anomalias, os problemas podem ser evitados com maior precisão pelas equipes de manutenção. Para Minhas e Zelek (2019), a avaliação de defeitos visuais por meio de imagens RGB é uma das formas de detecção de anomalias, sendo muito importante em tarefas de inspeção de superfícies. Nesse contexto, propõe-se neste trabalho o desenvolvimento de um detector visual novo e automático que reconheça o acúmulo de sujeira nas estruturas dos TC, utilizando Rede Neural Convolutacional (RNC) (do inglês, *Convolutional Neural Network* - CNN), para ser acoplado ao projeto ROSI como um dos sistemas de inspeção.

## **1.1. Objetivos**

O objetivo principal deste trabalho é realizar uma pesquisa no campo da Inteligência Artificial (IA) de forma a treinar um detector que reconheça a aglomeração de sujeira nas estruturas de rolos de TC com o uso de RNC.

### **1.1.1. Objetivos específicos**

- Pesquisar e entender como a IA pode ser aplicada no contexto da mineração;
- Coletar dados representativos do problema definido: Aglomeração de sujeira nas estruturas do TC;
- Conceber modelos de classificadores baseados em IA;
- Validar e experimentar os sistemas;
- Aplicar modelos em campo com avaliação em tempo real.

## **1.2. Motivação**

Segundo Nascimento (2018), em processos de mineração é extenso o uso de TC para a movimentação de grandes quantidades de material. O autor indica que apenas na Vale estão presentes mais de 2.000 TC, com aproximadamente 1.200 km de extensão de correia e aproximadamente 1,6 milhões de rolos, sendo que alguns componentes estão fisicamente espalhados ao longo do TC, em grande quantidade. Isto dificulta o uso de técnicas eficientes para o seu monitoramento.

Falhas nos rolos podem gerar impactos operacionais significativos, somando prejuízos materiais e estruturais. Para Carvalho Júnior (2018), os rolos dos TC geralmente falham de duas maneiras distintas: por travamento ou falha no rolamento. Para os dois casos, a identificação é feita de maneira visual em um processo que deixa o inspetor exposto a diversos riscos.

Desta forma, sistemas de IA que viabilizem uma inspeção automatizada dos equipamentos proporcionam ganhos para a empresa e, além disso, segurança para os inspetores. Como o projeto ROSI é uma demanda da Vale e possui os mesmos princípios de automatizar e melhorar a segurança das inspeções, entende-se que esta solução possa ser incorporada no sistema deixando-o mais robusto.

### **1.3. Publicações e participação em eventos**

#### **1.3.1. Simpósio Brasileiro de Automação Inteligente**

Artigo<sup>1</sup> aceito e apresentado no 14º Simpósio Brasileiro de Automação Inteligente (SBAI), 2019, na categoria poster, com título: Sistema automático para a inspeção visual de transportadores de correia.

#### **1.3.2. Mostra de Pós-Graduação do Encontro de Saberes**

Resumo<sup>2</sup> aceito e apresentado na Mostra de Pós-Graduação - UFOP do Encontro de Saberes 2019 na categoria poster, com título: Sistema automático para a inspeção visual de transportadores de correia.

### **1.4. Artigo publicado**

Artigo<sup>3</sup> submetido e publicado na revista Sensors com título “Automatic System for Visual Detection of Dirt Buildup on Belt Conveyors Using Convolutional Neural Networks”.

### **1.5. Contribuições internas no ITV**

Treinamentos internos do ITV, realizados para os colaboradores da Vale, alinhados ao tema objetivo do trabalho de mestrado.

---

<sup>1</sup><https://proceedings.science/sbai-2019/papers/sistema-automatico-para-a-inspecao-visual-de-transportadores-de-correia>

<sup>2</sup>[https://www.encontrodesaberes.ufop.br/exibir\\_trabalho.php?id=9887](https://www.encontrodesaberes.ufop.br/exibir_trabalho.php?id=9887)

<sup>3</sup><https://www.mdpi.com/1424-8220/20/20/5762>

### **1.5.1. Jornada de Inovação**

No dia 03/12/2019 foi apresentado um minicurso com título “Inteligência Artificial: Classificação (manutenção) com Imagens” na Jornada de Inovação da Vale, na mina do Sossego em Canaã do carajás, Pará.

O minicurso abordou o conceito geral de IA com uma base em classificação de problemas com imagens, abordagem relacionada à pesquisa desenvolvida neste trabalho de mestrado.

### **1.5.2. Treinamento Introductório em Programação em Python**

Treinamento realizado em quatro módulos para a Gerência de Automação Sul/Sudeste da Vale, durante março e abril, totalmente online.

Responsável pelos três módulos iniciais, foram ministradas 18h de curso com os seguintes tópicos:

1. Noções gerais de programação em Python;
2. Processamento de imagens (Python e OpenCV);
3. Deep Learning com Rede Neural Convolutacional.

## **1.6. Estrutura do trabalho**

A seguir, são apresentados trabalhos relacionados à área de visão computacional com IA no campo da mineração no Capítulo 2 e o referencial teórico com o conteúdo necessário para a realização do trabalho no Capítulo 3. A metodologia do trabalho é descrita no Capítulo 4. Os resultados e discussão são apresentados no Capítulo 5. Por fim, é apresentado a conclusão da pesquisa no Capítulo 6 com os possíveis trabalhos futuros descritos no Capítulo 7.

## 2. Trabalhos Relacionados

Aplicação de Rede Neural Artificial (RNA) para detecção de emendas de correias em TC foi o objetivo de pesquisa de Alport *et al.* (2001). A identificação automática de emendas de um TC móvel de alta velocidade, a partir de vídeo gravado, foi alcançada com um grau promissor de precisão usando coeficientes wavelet como entradas para uma RNA feedforward. Foi concluído no trabalho que o algoritmo wavelet discrimina com maior precisão as emendas e as características da correia, possibilitando que a saída da RNA (escalada entre 0 e 1) forneça uma medida direta da confiança dessa classificação particular.

A visão computacional é utilizada também com RNC para a identificação visual de problemas em superfícies defeituosas. Masci *et al.* (2012) apresentam uma abordagem de RNC Max-Pooling com treinamento supervisionado para classificação de defeitos em superfície de aço. Com uma base de dados de uma linha de produção real, contendo 2927 imagens, os testes foram comparados com algoritmos clássicos de extração de características: Local Binary Patterns (LBP), Local Binary Patterns Histogram Fourier (LBP-HF), Monogenic-LBP, Rotation invariant measure of local variance (VAR), Histogram of Oriented Gradients (HOG) e Pyramid of Histograms Orientation Gradients (PHOG). Foi obtido uma taxa de erro na classificação de 7 % com o modelo de RNC, trabalhando diretamente nas intensidades dos pixels de defeito do aço detectados e segmentados, enquanto que o melhor resultado dos modelos clássicos foi de 15 % com o PHOG.

No trabalho de Qiao *et al.* (2013) foi apresentado uma proposta para testar eficientemente a falha do rasgo longitudinal de uma correia. Os autores propõem um método de detecção por imagem infravermelha utilizando um sistema de *Support Vector Machine* (SVM). No trabalho foi pesquisado a relação entre o rasgo longitudinal e o número de pixels da área rasgada detectada na imagem. Desse modo, eles estabeleceram uma tabela de limiar para análise de imagens infravermelhas anti-coloridas com resolução de  $256 \times 256$  pixels.

Hamzeloo *et al.* (2014) utilizaram a visão computacional para estimar a distribuição de tamanho de partículas em circuitos de britagem de um concentrador de cobre, análise essencial para melhorar a eficiência energética e o desempenho metalúrgico. Várias imagens foram tiradas do material em um TC e processadas para identificação e segmentação de partículas. A distribuição do tamanho de partícula do material no TC foi estimada com precisão pelas técnicas de Análise de Componentes Principais (PCA) e RNA. É concluído no trabalho que sensores virtuais podem, atualmente, ser usados como ferramentas confiáveis, baratas e não intrusivas para a determinação do tamanho de partículas na indústria de mineração.

Com o objetivo de identificar TC com falhas mecânicas, Yang *et al.* (2014) desenvolveram um algoritmo de visão computacional para segmentar imagens de TC e detectar desvios longitudinais e das correias, que são uma séria ameaça à segurança de minas, a partir de imagens binárias que representam as potenciais falhas. Após o processamento binário, a imagem do TC é representada por 0 e 1. Assim, são extraídas as características da falha do TC de acordo

com a distribuição de 1 na imagem binária.

Para Yang *et al.* (2016), os principais componentes mecânicos de um TC sofrem com a falta de um monitoramento efetivo. Eles propõem um programa de robô de inspeção com termômetro infravermelho para TC. A proposta apresentada foi realizar a identificação automática dos elementos típicos após a segmentação por infravermelho, extração e classificação das características nas imagens. Após a extração dos dados, um classificador SVM foi utilizado para treinar as amostras e realizar a classificação automática da imagem infravermelha. A taxa de reconhecimento correta foi de até 96,7%.

Uma análise de problemas por meio de imagens térmicas também foi objeto de estudo de Carvalho Júnior (2018). O autor trabalhou no desenvolvimento de um método para a identificação dos defeitos com o uso de um drone com uma câmera térmica embarcada. A proposta é baseada na identificação das regiões de interesse, em um processamento morfológico das imagens, e em dados radiométricos. Foram testados dois algoritmos para a identificação da região dos rolos, o Viola Jones e o Canais de Características Agregados. Como conclusão, o algoritmo para a detecção de rolos atingiu taxas de falsos negativos de até 5% e o processo morfológico se mostrou eficiente em eliminar falsos positivos. A plataforma de transporte com drone se mostrou extremamente eficiente com ganhos de produtividade estimados de até 93% no tempo de inspeção.

Olivier *et al.* (2019) utilizaram a RNC de estrutura VGG para caracterizar tamanho do minério na alimentação de moinho por meio de imagens. Esta detecção é importante pois uma variação muito grande no tamanho das partículas de alimentação requer uma intervenção no sistema. Com 223 imagens categorizadas em 4 classes, e usando técnicas de aprendizado por transferência e *data augmentation* os resultados alcançados foram na ordem de 0,97 para a métrica de F1-Score. Os autores concluem que modelos de RNC podem superar os métodos tradicionais quando se trata de extrair das imagens a distribuição de tamanho de alimentação em um transportador industrial de minério.

Naixun *et al.* (2019) foram além da identificação de problemas específicos no campo de mineração. Eles foram capazes de identificar poços abertos de maneira rápida e precisa usando modelos de RNC combinados com um método de segmentação multi-escala. Foi realizada uma comparação entre o reconhecimento da RNC com um modelo SVM e foi concluído que a RNC possui menos classificação incorreta, identificando quase todos os detalhes. Com uma avaliação para 6.927 objetos aleatoriamente selecionados como amostras, foi alcançado 91,18% de acerto com um coeficiente Kappa de 0,89 para o modelo de RNC, enquanto que o SVM alcançou 84,58% e 0,81, respectivamente.

### 3. Referencial Teórico

#### 3.1. Processamento Digital de Imagens

A maior parte das informações recebidas pelos seres humanos é visual. Uma imagem é uma representação bidimensional (2-D) de uma cena tridimensional (3-D). Uma certa quantidade de informações pode ser transmitida de maneira rápida e eficaz por uma imagem. Portanto, o Processamento Digital de Imagem (PDI) pode ser usado em muitas aplicações da ciência e engenharia. Uma imagem é um sinal 2D e pode ser interpretada como um sinal discreto de amplitude  $x(m, n)$ , onde  $m$  e  $n$  são as duas variáveis independentes. A amplitude da imagem  $x(m, n)$  em cada ponto é o valor do píxel, que representa o elemento da imagem. De acordo Sundarajan (2017), os três maiores objetivos do PDI são: (i) melhorar a qualidade da imagem para a percepção humana, (ii) melhorar a qualidade e representar a imagem adequada para a percepção automática da máquina e (iii) comprimir a imagem para reduzir os requisitos de armazenamento e transmissão. Na Equação 3.1 é apresentado o domínio espacial de uma imagem digital  $x(m, n)$  e demonstra que uma imagem pode ser representada por uma matriz. Desta forma, operações em PDI são compostas basicamente de operações vetoriais e matriciais.

$$x(m, n) = \begin{bmatrix} x(0,0) & x(0,1) & x(0,2) & \cdots & x(0,N-1) \\ x(1,0) & x(1,1) & x(1,2) & \cdots & x(1,N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(M-1,0) & x(M-1,1) & x(M-1,2) & \cdots & x(M-1,N-1) \end{bmatrix} \quad (3.1)$$

Para imagens coloridas, no espaço de cores RGB (Red, Green, Blue), cada canal de cor é uma matriz única. Logo, cada píxel de uma imagem RGB possui três componentes, que podem ser organizados em um vetor coluna conforme Equação 3.2, onde  $Z_1$  é a intensidade do canal vermelho,  $Z_2$  é a intensidade do canal Verde e  $Z_3$  é a intensidade do canal Azul, respectivamente.

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (3.2)$$

Na Figura 3.1 é demonstrado os três canais que formam a imagem RGB e como os píxeis de cada canal podem ser organizados no vetor coluna. Extrapolando esse mesmo conceito para imagens com  $n$  canais, obtêm-se as imagens multiespectrais, (GONZALEZ e WOODS, 2017). No campo da robótica, aplicações com visão computacional são essenciais para solução de problemas, assim como em processos de inspeção de componentes na indústria (SUNDARAJAN, 2017).

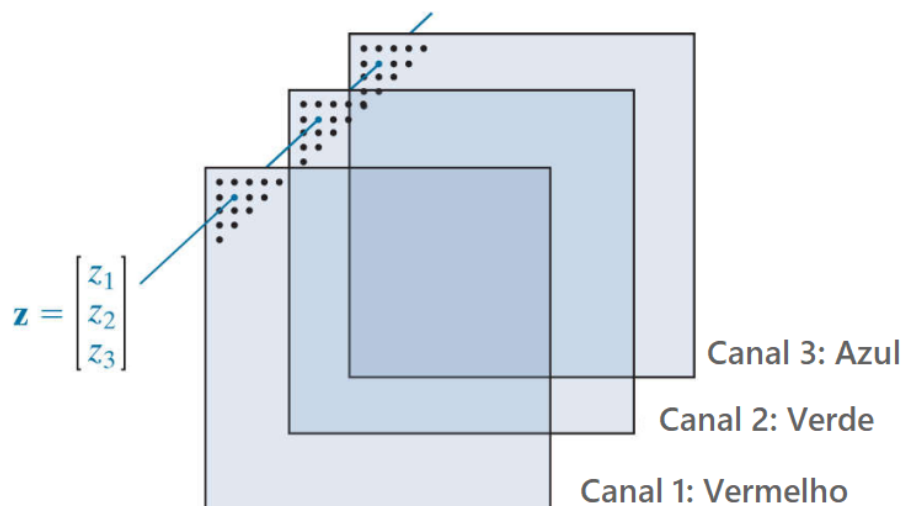


Figura 3.1: RGB.

Fonte: Adaptado de Gonzalez e Woods (2017)

## 3.2. Redes Neurais

O campo do reconhecimento de padrões preocupa-se com a descoberta automática de regularidades em dados por meio do uso de algoritmos de computador para classificar os dados em diferentes categorias (BISHOP, 2006). O cérebro é um computador altamente complexo, não linear e paralelo que trabalha de uma forma diferente dos computadores digitais. Essa complexidade e alta capacidade de reconhecimento de padrões tem motivado os trabalhos envolvendo RNA ao longo dos anos. Em termos gerais, uma RNA é uma máquina projetada para modelar a forma como o cérebro realiza uma tarefa ou função particular (HAYKIN *et al.*, 2009).

Um sistema nervoso pode ser visto como uma vasta rede de neurônios arranjados em estruturas com interconexões complexas. Essa rede recebe dados de entrada dos receptores, que convertem estímulos do corpo ou do mundo externo em padrões de impulsos elétricos que contêm informação para a rede. Esses dados interagem com a rede e resultam em impulsos que controlam os operadores terminais, como por exemplo os músculos, completando um sistema de três estágios (ARBIB, 1987). Os estágios citados são ilustrados na Figura 3.2.

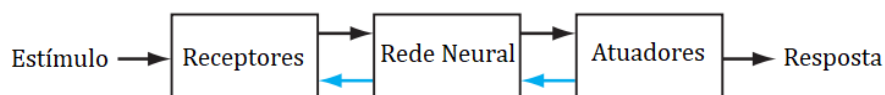


Figura 3.2: Sistema nervoso em três estágios.

Fonte: adaptado de Haykin *et al.* (2009)

### 3.2.1. Modelo de um Neurônio

De acordo com Haykin *et al.* (2009), um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma RNA, sendo a base para uma vasta família de tipos de redes existentes. Na Figura 3.3 é apresentado um modelo não linear de um neurônio, onde é possível observar algumas das características citadas por Arbib (1987): O sinais de entradas (*Input signals*) e saída (*Output*) podem ser considerados, respectivamente, como os estímulos e a resposta do neurônio ao estímulo. A rede em si é composta pelo conjunto dos pesos sinápticos (*Synaptic weights*), junção de soma (*Summing junction*) e função de ativação (*Activation function*), que vão determinar como os estímulos serão interpretados pelo neurônio e como essa informação será passada adiante.

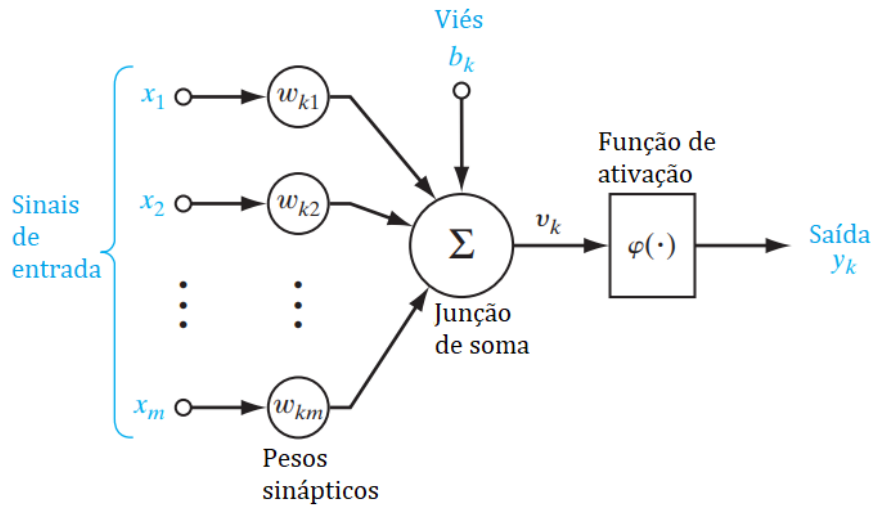


Figura 3.3: Modelo não linear de um neurônio.

Fonte: adaptado de Haykin *et al.* (2009)

Matematicamente, o neurônio artificial pode ser formulado conforme Equação 3.3:

$$y = f(\mathbf{x}) = \varphi(\mathbf{w}\mathbf{x}^T + b) \quad (3.3)$$

em que  $\mathbf{w}$  é o vetor de pesos,  $\mathbf{x}$  é o vetor de entrada e  $b$  é o termo de interceptação, também chamado de viés (*Bias*).

### 3.2.2. Rede Feedforward Multicamada

Uma RNA é composta por um conjunto de neurônios interligados. A forma como esses neurônios são organizados entre si define a arquitetura da rede. Essa ligação é estruturada essencialmente direcionando as conexões sinápticas dos neurônios (DA SILVA *et al.*, 2017).

Uma estrutura utilizada para problemas de classificação de padrões, identificação de sistemas, otimização, entre outros, é a rede Feedforward de multicamadas (DA SILVA *et al.*,



2017). Ela é composta por uma camada de entrada de dados, uma camada de saída, e uma ou mais camadas intermediárias, também chamadas camadas escondidas, conforme Figura 3.4. A adição dessas camadas intermediárias possibilita a rede extrair estatísticas de ordem superior dos dados de entrada (HAYKIN *et al.*, 2009). Olhando para a Equação 3.3, em uma rede Feedforward, a saída de um neurônio se torna a entrada da próxima camada. Logo, verifica-se que esses dados estão encadeados em função das camadas intermediárias.

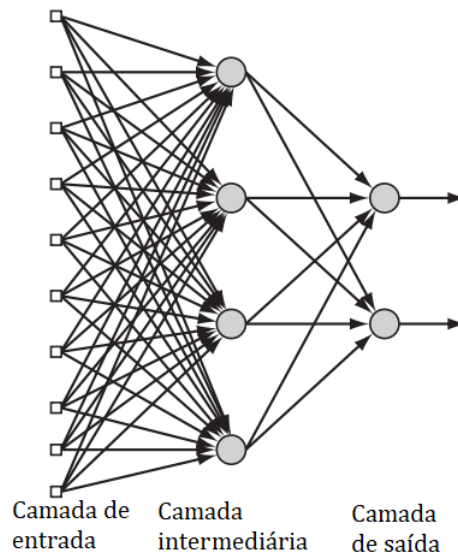


Figura 3.4: Rede Feedforward conectada.  
Fonte: adaptado de Haykin *et al.* (2009)

Considerando a Figura 3.4, pode-se formular a Equação 3.4 que descreve a relação existente entre a saída da rede com as camadas intermediária e de entrada. Neste caso a saída  $\mathbf{y}$  se torna um vetor, pois tem-se mais de um neurônio de saída.

$$\mathbf{y} = f(\mathbf{x}) = \varphi(\varphi(\mathbf{w}^1 \mathbf{x} + b^1) \mathbf{w}^2 + b^2) \quad (3.4)$$

À medida que novas camadas intermediárias são adicionadas na estrutura, esta formulação vai crescendo e ficando cada vez mais profunda. Aghdam e Heravi (2017) dizem que, em termos de transformação, as camadas intermediárias atuam como uma função de transformação de características, que é uma função composta. Em seguida, a camada de saída atua como o classificador linear. O modelo Feedforward pode ser considerado um aproximador universal, sendo muito importante em problemas de classificação.

### 3.2.3. Treinamento de Rede Feedforward

O treinamento das redes Feedforward, por meio de dados de amostras de entrada, possibilita o aprendizado da relação entre as entradas e as saídas previamente conhecidas, estabelecendo, assim, uma identificação do comportamento de um padrão. Desta forma, a rede

generaliza uma solução para produzir uma saída que seja próxima à desejada, dado um certo valor de entrada (DA SILVA *et al.*, 2017).

Buduma e Locascio (2017) descrevem esse processo como a tentativa de minimizar uma função de custo  $E$ , ou função de erro, para todas as amostras de treinamento  $i$ . Assumindo a função de custo do erro quadrático na Equação 3.5, e sabendo que  $t^{(i)}$  é a resposta real conhecida da amostra e  $y^{(i)}$  é o valor computado pela rede, quanto mais próximo de zero o resultado de  $E$  chegar, melhor.

$$E = \frac{1}{2} \sum_i (t^{(i)} - y^{(i)})^2 \quad (3.5)$$

As funções de ativação inserem não linearidades em uma RNA deixando-a mais robusta para solucionar problemas não lineares. Logo, uma estratégia de otimização da função de custo é importante. Para Buduma e Locascio (2017), o método Gradiente Descendente (GD) soluciona esta questão. Com a direção do gradiente é possível dar um passo na direção correta que minimiza a função de custo. O gradiente é a derivada parcial da função de custo com relação à cada um dos pesos da rede. Em suma, a equação do gradiente é definida como

$$\Delta \mathbf{w} = -\varepsilon \frac{\partial E}{\partial \mathbf{w}} \quad (3.6)$$

em que  $\varepsilon$  é a taxa de aprendizado e define o fator de multiplicação aplicado sobre o gradiente.

O algoritmo chamado *Backpropagation* busca encontrar o caminho do GD para definir o valor de ajuste de cada peso individual da rede. Isto é possível pois, como a rede está em camadas, cada elemento de uma camada influencia diretamente na resposta do elemento na camada seguinte. Para Data Science Academy (2019), o *Backpropagation* é indiscutivelmente o algoritmo mais importante na história das redes neurais. Ele pode ser dividido em duas partes:

1. Fase de propagação, na qual os dados passam pela rede e a previsão é obtida;
2. Fase de retro-propagação: na qual é calculado o gradiente da função de custo na camada final e calculado, de forma recursiva, os ajustes de atualização para cada peso da rede.

Logo, usando o algoritmo *Backpropagation* é possível mapear os gradientes entre todas as camadas. Buduma e Locascio (2017) demonstram que, com o *Backpropagation*, chega-se à seguinte fórmula de modificação dos pesos:

$$\Delta w_{ij} = - \sum_k \varepsilon y_i^{(k)} y_j^{(k)} (1 - y_j^{(k)}) \frac{\partial E^{(k)}}{\partial y_j^{(k)}} \quad (3.7)$$

em que  $i$  representa uma camada anterior à camada  $j$  e  $k$  representa o  $k$ -ésimo neurônio na camada.

Uma abordagem para melhorar a capacidade de encontrar o mínimo global de uma função é usar o *Gradiente Descendente Estocástico* (SGD) com pequenos lotes. Neste caso, o

ajuste dos pesos na Equação 3.7 é função de um lote de dados. Durante as épocas de treinamento as atualizações ocorrem de lote em lote, e não para o conjunto completo de dados. Este método tem aplicações muito importantes além do contexto de aprendizado profundo e é a principal maneira de treinar grandes modelos não lineares em conjuntos de dados muito grandes (YANG *et al.*, 2014).

### 3.3. Rede Neural Convolutional

Quando a finalidade de uma RNA é a identificação de padrão por imagens, as redes Feedforward necessitam de um número muito grande de neurônios, o que dificulta o seu uso prático. Uma solução para tal problema é a RNC (AGHDAM e HERAVI, 2017).

Em uma imagem, cada píxel possui uma alta correlação com seus vizinhos e uma baixa correlação com os píxeis distantes. Uma RNC aproveita essa característica para extrair informações de uma imagem por meio de filtros, também chamados kernels, que são aplicados em regiões distintas da mesma. Ao invés de cada neurônio estar relacionado a um píxel, cada neurônio é relacionado ao resultado da aplicação de um filtro em um conjunto de píxeis de uma região da imagem, conforme Figura 3.5. Logo, trata-se de uma operação de convolução em dois eixos que pode ser escrita de acordo Equação 3.8.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n) \quad (3.8)$$

em que  $I$  representa a imagem de entrada,  $(i, j)$  a dimensão da operação de convolução e  $K$  o filtro de dimensão  $(m, n)$  (YANG *et al.*, 2014).

Para vários filtros aplicados sobre uma imagem, pode-se alterar a Equação 3.8 para:

$$S^l(i, j) = (I * K^l)(i, j) = \sum_m \sum_n I(i + m, j + n) K^l(m, n) \quad (3.9)$$

em que  $l$  representa o índice de um filtro específico.

A aplicação desses filtros reduz drasticamente a quantidade de neurônios e de parâmetros necessários para processamento da RNA, pois os pesos sinápticos de cada neurônio são compartilhados por um mesmo filtro (AGHDAM e HERAVI, 2017). Na Figura 3.5 é exemplificado o uso de 50 filtros, ( $f_0$  até  $f_{49}$ ), aplicados em uma imagem. Verifica-se que cada filtro possui seus neurônios e pesos sinápticos específicos. Relacionando a Figura 3.5 com a Equação 3.9,  $f_0$  representa o  $S^0$  e  $(a, b, c, d)$  os pesos do filtro  $K^0$ . Esta relação pode ser estabelecida para todos os 50 filtros do exemplo.

De acordo Haykin *et al.* (2009), uma RNC é uma rede multicamadas projetada especificamente para reconhecer formas bidimensionais com um alto grau de invariância à translação, escalonamento, inclinação e outras formas de distorção. A aplicação da RNC possui as seguintes características:

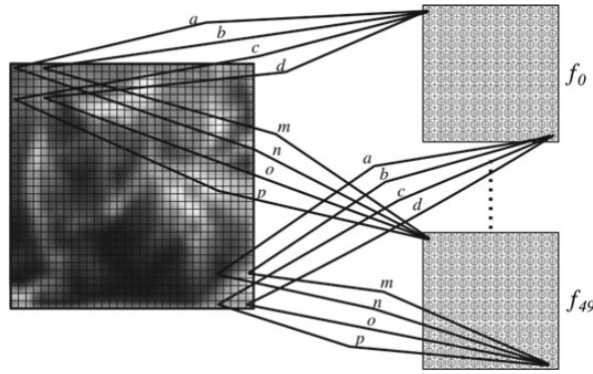


Figura 3.5: Neurônios de um filtro  $f$  que compartilham os mesmos pesos sinápticos.  
Fonte: Aghdam e Heravi (2017)

1. **Feature extraction:** cada neurônio obtém suas entradas sinápticas de um campo receptivo local na camada anterior, forçando-o a extrair características locais.
2. **Feature mapping:** cada camada computacional da rede é composta de múltiplos *feature maps*, ou filtro de características, os quais são na forma de um plano onde cada um dos seus neurônios individuais são forçados a compartilhar o mesmo conjunto de pesos sinápticos.
3. **Subsampling:** cada camada convolucional é seguida de uma camada computacional que realiza uma média local e subamostragem, onde a resolução do *feature map* é reduzida. Essa operação tem o efeito de reduzir a sensibilidade da saída do *feature map* em relação às mudanças e outras formas de distorção.

Todos os pesos em todas as camadas da RNC são aprendidos por meio de treinamento com o *Backpropagation*. Assim, a rede extrai suas próprias características automaticamente. Na Figura 3.6 é apresentado um esquema de RNC para reconhecimento de caracteres escritos a mão com uma camada de entrada, quatro camadas intermediárias responsáveis por extrair as características principais das imagens e uma camada de saída.

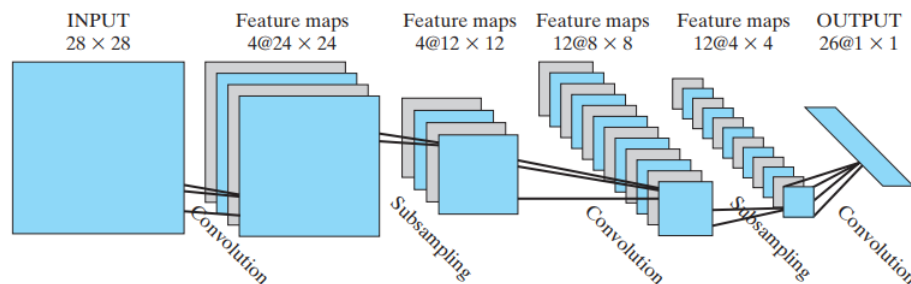


Figura 3.6: Rede convolucional para processamento de imagem.  
Fonte: Haykin *et al.* (2009)

### 3.3.1. Estrutura de uma rede neural convolucional

Durante o treinamento de uma RNC, a imagem passa por algumas camadas conectadas aos neurônios. Os pesos dos neurônios estão associados aos filtros criados e são aprendidos durante o treinamento. As etapas que extraem características dos dados são constituídas da convolução e pooling (YANG *et al.*, 2014).

#### Convolução

Na Figura 3.7 é possível visualizar uma camada de entrada e uma camada intermediária. A camada de entrada representa uma imagem, com duas dimensões e um canal. Na intermediária, há um conjunto de neurônios onde cada um será associado a uma região da imagem original. Cada região seleccionada na imagem original é o campo receptivo local para o respectivo neurônio intermediário (DATA SCIENCE ACADEMY, 2019).

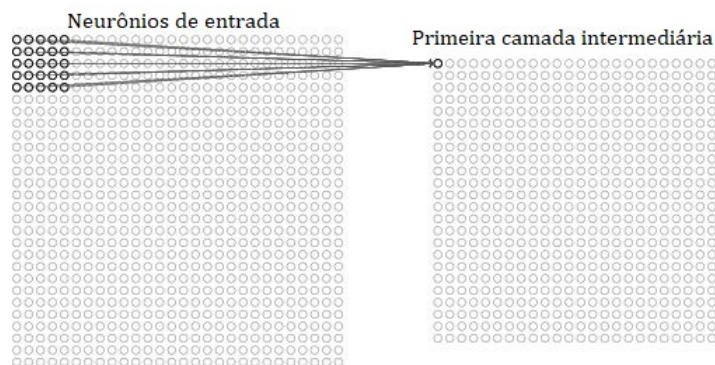


Figura 3.7: Campo receptivo local.

Fonte: adaptado de Data Science Academy (2019)

O filtro, de dimensão (5X5), é deslizado sobre toda a imagem, da esquerda para direita e de cima para baixo, passando por vários campos receptivos locais, processo matematicamente representado na Equação 3.9. Nesta etapa acontece, de fato, a convolução e para cada campo receptivo local, existe um neurônio oculto diferente na primeira camada oculta, conforme Figura 3.8.

Cada neurônio da primeira camada intermediária compartilham os mesmos pesos. Isso caracteriza o resultado dos neurônios na camada intermediária, ou também como é chamado, mapa de recursos. Para fazer o reconhecimento de imagem, é necessário mais de um mapa de recursos. Assim, uma camada convolucional completa consiste em vários mapas de recursos diferentes que criam estruturas espaciais de detecção, (DATA SCIENCE ACADEMY, 2019). Um neurônio no mapa de características é ativado se o filtro detectar uma característica apropriada na posição correspondente do campo receptivo local (BUDUMA e LOCASCIO, 2017).

Esta operação pode ser estendida para imagens tridimensionais, desta forma os filtros também são tridimensionais e a operação é realizada sobre todo o volume da imagem em uma

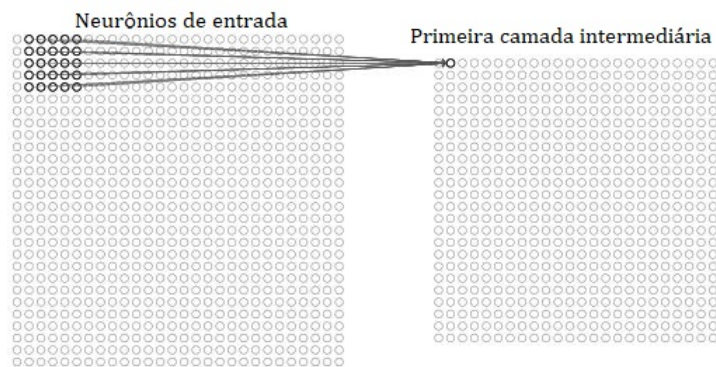


Figura 3.8: Deslocamento do campo receptivo local.  
Fonte: adaptado de Data Science Academy (2019)

determinada camada da rede. Isto é importante pois cada mapa de recursos é responsável por uma característica, logo, a análise combinada de múltiplos mapas leva a evidências importantes para uma classificação (BUDUMA e LOCASCIO, 2017). Na Figura 3.9 é exemplificado este processo, sendo possível identificar uma entrada com três canais RGB, seis filtros aplicados sobre a entrada e uma saída da convolução com seis canais representando o mapa de recursos. Em uma estrutura conectada de uma rede, este mapa pode ser entrada para uma nova camada de convolução ou de Pooling.

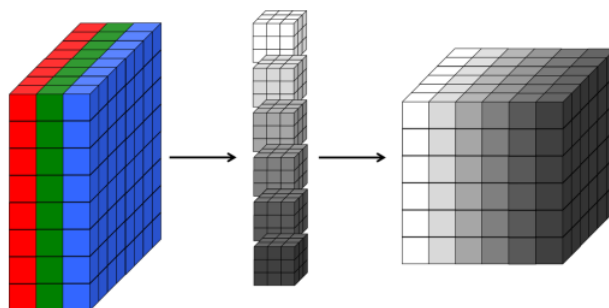


Figura 3.9: Visualização tridimensional de uma camada de convolução.  
Fonte: Buduma e Locascio (2017)

## Pooling

A próxima etapa de uma RNC é a etapa de *pooling*, ou agrupamento. Ela simplifica as informações na saída da camada convolucional, (DATA SCIENCE ACADEMY, 2019). Esta etapa caracteriza o *subsampling* citado por Haykin *et al.* (2009). Assim, o dado de entrada se torna mais invariante a pequenas translações. Invariância à translação significa que, ao se alterar a entrada em uma pequena quantidade, os valores da maioria das saídas combinadas não serão alterados (YANG *et al.*, 2014).

A etapa de pooling recebe cada saída do mapa de características na etapa convolucional e prepara um mapa de características condensadas. Cada unidade na etapa de pooling pode

resumir uma região de neurônios na camada anterior. Um procedimento comum para o pooling é conhecido como agrupamento pelo máximo (ou *Max-Pooling*). No Max-Pooling, uma unidade de pooling simplesmente gera a ativação máxima na região de entrada. Na Figura 3.10 é ilustrado a operação de Max-Pooling considerando uma região  $2 \times 2$  da camada anterior. Esta operação reduz a imagem da camada anterior pela metade e deve ser aplicada em cada mapa de recursos separadamente, (DATA SCIENCE ACADEMY, 2019).

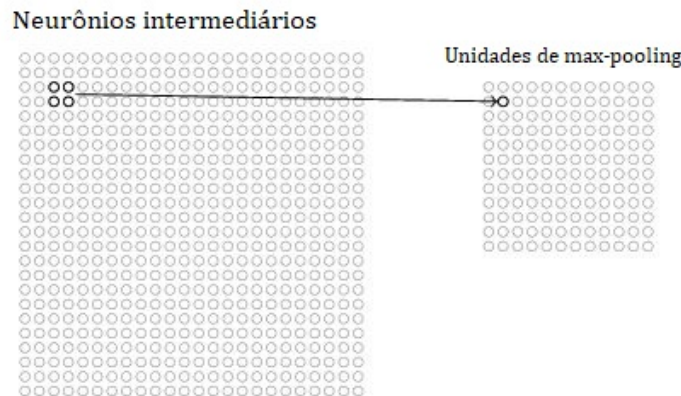


Figura 3.10: Operação de *Max Pooling*.  
Fonte: adaptado de Data Science Academy (2019)

A etapa final é conectar a saída do *Max Pooling* em uma rede totalmente conectada, ou seja, uma rede Feedforward multicamadas apresentada na Figura 3.4.

Uma RNC complexa possui várias camadas convolucionais e de *Pooling*. Isto define a capacidade de extração de características da rede. Desta forma, toda a informação é extraída das camadas convolucionais e vetorizada, servindo como entrada para a etapa final da rede que realizará a classificação dos dados.

### 3.4. Aprendizado por transferência

Humanos parecem ter formas inerentes de transferir conhecimento entre tarefas. Ou seja, reconhecem e aplicam conhecimento relevante de experiências de aprendizado anteriores quando encontram novas tarefas. Quanto mais uma nova tarefa é relacionada à experiência anterior, mais facilmente ela é dominada, (TORREY e SHAVLIK, 2010).

O aprendizado por transferência, do inglês *Transfer Learning* (TL), é um método de aprendizado de máquina em que um modelo desenvolvido para uma tarefa é reutilizado como ponto de partida para um modelo em uma segunda tarefa (BROWNLEE, 2017). Para Goodfellow *et al.* (2016), TL refere-se à situação em que o que foi aprendido em um ambiente é explorado para melhorar a generalização em outro ambiente.

Técnicas que permitem a transferência de conhecimento representam um progresso no sentido de tornar a aprendizagem de máquina tão eficiente quanto a aprendizagem humana,



(TORREY e SHAVLIK, 2010). Na Figura 3.11 são apresentadas três maneiras pelas quais o TL pode melhorar o aprendizado: um desempenho superior no início do aprendizado; uma inclinação mais íngreme na curva de aprendizado; um desempenho assintótico superior.

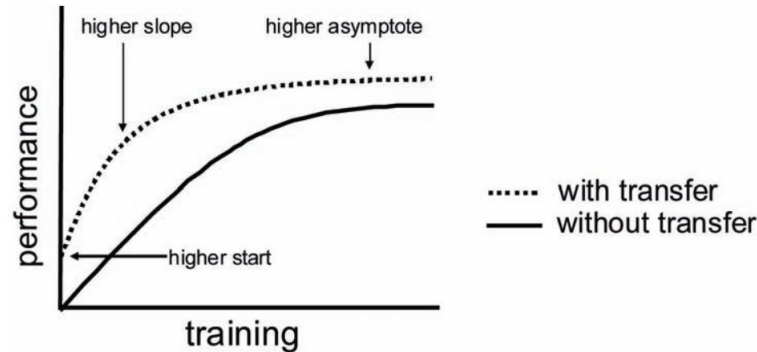


Figura 3.11: Três maneiras pelas quais a transferência pode melhorar o aprendizado.

Fonte: Torrey e Shavlik (2010)

Bhatia *et al.* (2019) realizaram um trabalho para detectar buracos de rua usando IA. No seu trabalho, os autores estabeleceram um comparativo entre uma RNC construída do zero e o uso de modelos de redes *ResNet* pré-treinadas para TL. O resultado observado foi que a transferência de conhecimento de redes existentes viabiliza a construção de modelos precisos de redes de uma maneira que economiza tempo e processamento.

### 3.5. Avaliação de um modelo de Inteligência Artificial

Para Aghdam e Heravi (2017), uma das dificuldades na RNC é encontrar uma boa arquitetura que produza resultados precisos e seja computacionalmente eficiente. A maneira prática é escolher uma arquitetura, implementar e treiná-la no conjunto de treinamento. Em seguida, o modelo é avaliado no conjunto de validação.

Avaliar e escolher o melhor modelo de IA treinado não é fácil. Russell e Norvig (2009) definem esse processo como a escolha de uma hipótese, dentro de um espaço de hipóteses, que generaliza bem em um conjunto de teste e que seja distinto do conjunto de treino. Isto significa escolher uma hipótese que melhor se ajuste a dados futuros. A abordagem mais simples é dividir aleatoriamente os dados disponíveis em um conjunto de treinamento, a partir do qual o modelo de IA aprende a hipótese, e um conjunto de testes no qual a precisão da hipótese é avaliada. Esta abordagem falha em usar os dados disponíveis, pois os resultados são influenciados pela qualidade e quantidade das amostras.

Para extrair mais informações dos dados e, ainda assim, obter um bom resultado, Russell e Norvig (2009) sugere o uso da técnica de validação cruzada *k-pastas*. A ideia é que cada exemplo tenha duas funções – como dados de treinamento e dados de teste. Primeiro, divide-se os dados em  $k$  subconjuntos iguais. Em seguida, realiza-se  $k$  rodadas de aprendizado; em cada rodada  $1/k$  dos dados é apresentado como um conjunto de testes e os exemplos restantes são



usados como dados de treinamento. A pontuação média do conjunto de testes das  $k$  rodadas deve ser uma estimativa melhor do que uma única pontuação. Yang *et al.* (2014) apresenta o algoritmo da validação cruzada  $k$ -pastas conforme a seguir:

---

**Algorithm 1** Algoritmo da validação cruzada  $k$ -pastas

---

**Defina**  $\text{validacaoKpastas}(\mathbb{D}, A, L, k)$

**Requer:**  $\mathbb{D}$ , o dataset completo com  $z^{(i)}$  elementos, onde  $z^{(i)} = (x^{(i)}, y^{(i)})$

**Requer:**  $A$ , o algoritmo de aprendizado

**Requer:**  $L$ , a função de custo utilizada

**Requer:**  $k$ , o número de pastas (divisões)

Separe  $\mathbb{D}$  em  $k$  subconjuntos mutuamente exclusivos  $\mathbb{D}_i$ , os quais a união é  $\mathbb{D}$

**for**  $i$  de 1 até  $k$  **do**

$f_i = A(\mathbb{D}/\mathbb{D}_i)$

**for**  $z^{(j)}$  em  $\mathbb{D}_i$  **do**

$e_j = L(f_i, z^{(j)})$

**end for**

**end for**

**Retorna**  $e$

---

A avaliação de um modelo de classificação no conjunto de teste pode ser feita com algumas métricas especiais. Aghdam e Heravi (2017) citam a Matriz de Confusão como uma ferramenta poderosa para avaliar, de forma criteriosa, os modelos de RNC. Para um problema de classificação com  $C$  classes, a matriz de confusão,  $M$ , é uma matriz  $C \times C$ , na qual o elemento  $M_{ij}$  da matriz mostra o número de amostras no conjunto de teste cujo rótulo real de classe é  $i$ , mas são classificados como classe  $j$  usando a RNC. O  $M_{ij}$  mostra o número de amostras classificadas correta e incorretamente.

Na Figura 3.12 é representada a matriz de confusão para um problema binário, em que o eixo horizontal representa a informação real e o eixo vertical a informação predita pela rede. Os elementos  $M_{ij}$  podem ser caracterizados conforme a seguir:

- True Positive (TP) - Verdadeiro Positivo: Elementos da classe positiva classificados corretamente;
- False Negative (FN) - Falso Negativo: Elementos da classe positiva classificados incorretamente;
- False Positive (FP) - Falso Positivo: Elementos da classe negativa classificados incorretamente;
- True Negative (TN) - Verdadeiro Negativo: Elementos da classe negativa classificados corretamente;

É importante observar que,  $TP + FN + FP + TN = n$ , sendo que  $n$  é a quantidade do conjunto de dados avaliado com a matriz. Um modelo pode ser dito ideal quando  $FP = FN = 0$ . A partir da matriz de confusão várias medidas de desempenho podem ser avaliadas.

		Predicted Label	
		1	0
Actual Label	1	True Positive	False Negative
	0	False Positive	True Negative

Figura 3.12: Matriz de cnfusão de um problema binário.  
Fonte: Adaptada de Aghdam e Heravi (2017)

De acordo Aghdam e Heravi (2017), a acurácia, na Equação 3.10, é a métrica mais simples de ser avaliada. Ela indica a fração das amostras que são classificadas corretamente. No entanto, avaliar unicamente a acurácia não é recomendado por causa de suas limitações principalmente para classes desbalanceadas.

$$acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.10)$$

Para solucionar este problema, as métricas quantitativas a seguir ajudam a ter uma observação mais confiável de um modelo, se comparadas com a acurácia. A precisão, na Equação 3.11, calcula a fração dos preditos positivos. A revocação, na Equação 3.12, calcula a fração dos positivos reais. Faceli *et al.* (2011) indicam que a precisão pode ser vista como uma medida de exatidão do modelo e a revocação como uma medida de sua completude.

$$precisão = \frac{TP}{TP + FP} \quad (3.11)$$

$$revocação = \frac{TP}{TP + FN} \quad (3.12)$$

Aghdam e Heravi (2017) citam a métrica F1-Score, ou seja, a média harmônica entre a precisão e a revocação, como uma alternativa eficiente para avaliar um modelo de RNC com apenas um valor. Faceli *et al.* (2011) indicam que essa análise é importante pois, uma precisão de 1 para uma classe  $C$  significa que cada item rotulado como pertencente à classe  $C$  realmente pertencem a essa classe, mas não fornece nenhuma informação a respeito do número de exemplos da classe  $C$  que não foram classificados corretamente.

$$F1Score = \frac{2 \times precisão \times revocação}{precisão + revocação} \quad (3.13)$$

## 4. Materiais e Métodos

Nesta seção são apresentados os procedimentos desenvolvidos durante a pesquisa e as etapas de validação. Os métodos utilizados desde a coleta de dados até a etapa de validação em campo dos modelos de IA serão abordados.

Durante as etapas de construção dos modelos e validação a linguagem de programação *Python* foi utilizada como ferramenta de desenvolvimento (MADHAVAN, 2015) com as bibliotecas OpenCV (BRADSKI, 2000) e *Pytorch*, uma biblioteca construída para permitir pesquisa com modelos de IA (PASZKE *et al.*, 2017).

### 4.1. Coleta de dados

A primeira coleta de dados foi realizada na Mina de Alegria, na cidade de Mariana, MG, no dia 15 de março de 2019.

Neste primeiro momento, foi verificado que um problema específico relacionado ao acúmulo de materiais aglutinantes nas estruturas do TC é muito comum. Se esse acúmulo não for detectado durante uma inspeção, problemas de travamento podem ser gerados nos rolos do TC aquecendo-os e danificando-os. Assim, foi realizada a captura de dados com o objetivo de caracterizar e classificar o problema do acúmulo de materiais aglutinantes nas estruturas.

Os TC foram fotografados com uma câmera comum RGB, nas quais as fotos possuem dimensões de  $4000 \times 2000$  pixels. As imagens foram coletadas em ângulos diferentes sempre com foco na parte central das estruturas, local onde há mais acúmulo dos resíduos como indicado na Figura 4.1.



Figura 4.1: Estrutura do TC com acúmulo de resíduos.  
Fonte: Própria e registrada no dia 15 de março de 2019.

A segunda coleta de dados foi realizada entre os dias 08 e 12 de julho de 2019 no porto de Tubarão na cidade de Vitória, ES. Nesta visita foi possível uma coleta mais completa

sendo possível registrar os TC em diversas situações, como por exemplo imagens fotografadas à noite, com elementos na estrutura que podem dificultar a visualização, e imagens com diferentes exposição à luz, pois algumas fotos foram capturadas em ambientes com cobertura externa. Algumas imagens com essas características podem ser visualizadas na Figura 4.2.

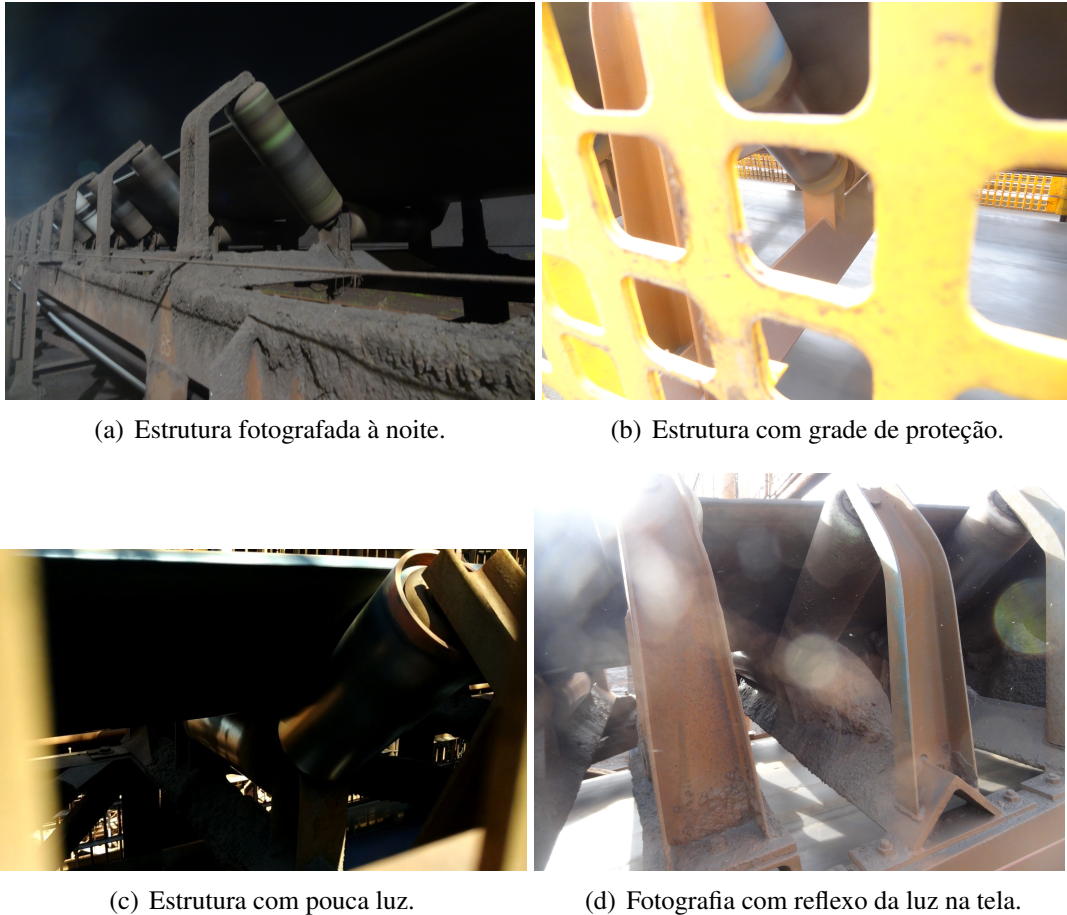


Figura 4.2: Coleta de imagens em situações diversas.

Fonte: Própria.

Após a coleta, algumas imagens passaram por um processo manual de extração da Região de Interesse (ROI) com o objeto de focar ao máximo na região central da estrutura. A ROI exemplo extraído da Figura 4.2(d) pode ser vista na Figura 4.3.

Ao final destes procedimentos foram obtidas 392 imagens para compor o dataset da pesquisa. Assim, foram definidas duas classes principais para análise, caracterizando um problema classificação binário: (i) *Limpo* e (ii) *Sujo*. Então, o dataset foi montado com 228 imagens da classe *Limpo* e 164 da classe *Sujo*, como indicado na Figura 4.4. De acordo Fernández *et al.* (2018), o desequilíbrio de classe está presente em muitos conjuntos de dados de classificação do mundo real. Sabe-se que esse problema dificulta o desempenho dos classificadores, o que geralmente faz com que a classe minoritária seja negligenciada. Com a coleta, os dados estão apresentados com razão de 1,39 : 1 da classe *Limpo* em relação à classe *Sujo*.



Figura 4.3: Imagem da estrutura após extração da região central.  
Fonte: Própria.

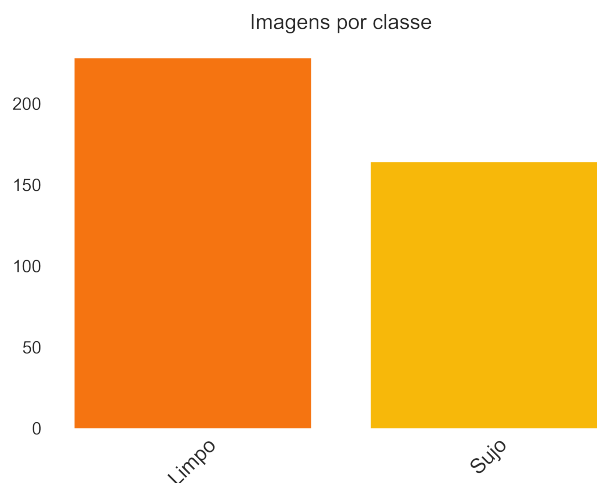


Figura 4.4: Quantidade de imagens por classe.  
Fonte: Própria.

## 4.2. Pré-processamento dos dados

A divisão dos dados em conjuntos de treino e validação é de extrema importância para a observação da eficiência do modelo. No entanto, dividir o dataset em conjuntos fixos pode ser problemático se o conjunto de dados for pequeno demais (GOODFELLOW *et al.*, 2016). De acordo Russell e Norvig (2009), o método de validação cruzada *k-pastas* é uma das opções para se ter uma estimativa precisa de um modelo de IA. Com esta metodologia é necessário realizar *k* rodadas de treinamento. Em cada rodada,  $1/k$  dos dados é selecionado como o conjunto de teste e o restante dos dados como o conjunto de treinamento.

Durante esta etapa, os dados foram organizados conforme indicado na Figura 4.5. O dataset foi dividido em dez partes, sem sobreposição, com a proporção aproximada de 1,39 : 1 entre as classes. Para cada rodada foram escolhidos dois *k-pastas* para o conjunto de teste. Desta forma, em cada rodada foi possível ter, aproximadamente, 80% dos dados para treino e

20% para teste.

	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Limpo	23	23	23	23	23	23	23	23	22	22
Sujo	17	17	17	17	16	16	16	16	16	16

Rodada 1	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 2	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 3	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 4	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 5	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 6	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 7	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 8	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 9	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
Rodada 10	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10

Conjunto de treino
  Conjunto de Teste

Figura 4.5: Divisão dos dados para o método de validação cruzada k-pastas.

Fonte: Própria.

Após a etapa de divisão dos dados as imagens foram redimensionadas para uma resolução de  $224 \times 224$  pixels. Para o processo de treinamento, a técnica de *Data Augmentation* (DA) foi utilizada para inserir pequenas alterações nos dados, com o objetivo de diversificar o conjunto original de treino, conforme mostrado na Figura 4.6. De acordo Bhatia *et al.* (2019), o uso de DA possibilita o melhor uso das imagens quando se tem poucos dados.

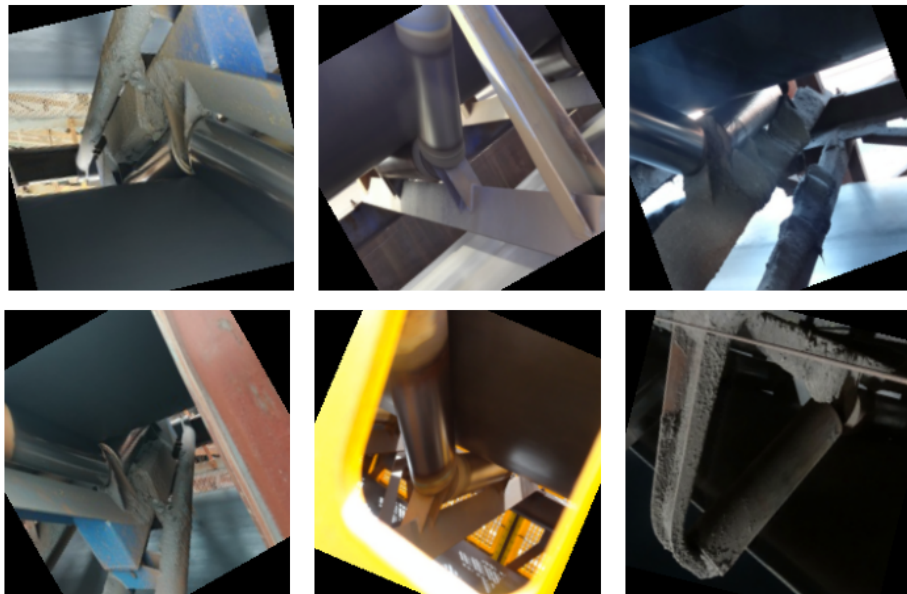


Figura 4.6: Data Augmentation aplicados nas imagens.

Fonte: Própria.

Quatro tipos de alterações foram inseridas nos dados:

1. **Random Horizontal Flip**: um giro horizontal aplicado nas imagens com uma probabilidade de ocorrência de 0,3 - o objetivo desta alteração é, especificamente, inserir ruído na imagem;
2. **Random Vertical Flip**: um giro vertical foi aplicado nas imagens com uma probabilidade de ocorrência de 0,3 - o objetivo desta alteração é evitar que a rede se especialize em um único sentido do rolo na estrutura;
3. **Random Rotation**: uma inclinação aleatória com ângulo de até 30 foi aplicada nas imagens - o objetivo desta alteração é, especificamente, inserir ruído na imagem;
4. **Color Jitter**: uma alteração aleatória de até 0,05 na saturação e matiz foi aplicada nas imagens - o objetivo desta alteração alterar o grau de pureza da cor.

### 4.3. Arquiteturas das redes neurais convolucionais

Na literatura existem muitas arquiteturas diferentes de RNC. Segundo Goodfellow *et al.* (2016), é útil aumentar a quantidade de camadas convolucionais e reduzir a quantidade de camadas de pooling à medida que a estrutura da rede cresce, pois as camadas de pooling são destrutivas e as várias camadas convolucionais em sequência possibilitam uma extração mais rica de características.

Para a modelagem do problema, a definição de modelos adequados para reconhecimento de padrões em imagens foi importante. Assim, duas arquiteturas foram selecionadas como base para desenvolvimento e comparação dos desempenhos:

1. Visual Geometry Group network (VGG);
2. Residual Network (ResNet).

Estas arquiteturas possuem modelos com variações na sua construção. Desta forma é importante deixar claro que os modelos usados foram: **VGG16** e **ResNet18**.

#### 4.3.1. VGG16

De acordo Simonyan e Zisserman (2014), o modelo VGG16 foi treinado originalmente com o conjunto de dados *ImageNet* e apresenta ótima capacidade de generalização para muitos problemas. Em 2014, o modelo alcançou o primeiro e segundo lugar nos desafios de localização e rastreo, respectivamente, no ImageNet Large Scale Visual Recognition Challenge (ILSVRC). O conjunto de dados *ImageNet* possui mais de 14 milhões de imagens com 1000 classes e o modelo VGG16 alcançou uma acurácia de 92.7% no conjunto de teste dos dados.

O modelo tem uma estrutura com pequenos filtros de tamanho  $(3 \times 3)$ , em todas as camadas convolucionais, com uma profundidade de 16 camadas com pesos ajustáveis. As camadas



totalmente conectadas são compostas de duas camadas com 4096 neurônios cada uma, seguidas de uma camada com 1000 neurônios na saída. Originalmente o modelo possui 138 milhões de parâmetros e ocupa um espaço de armazenamento de aproximadamente 528 MegaBytes (MB). Na Figura 4.7 é apresentada a arquitetura original do modelo.

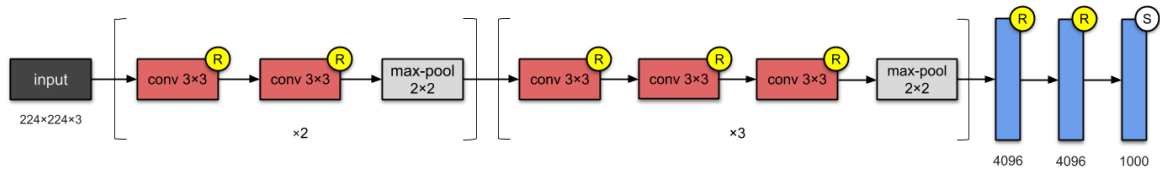


Figura 4.7: Arquitetura original do modelo VGG16.  
Fonte: (KARIM, 2019)

### 4.3.2. ResNet18

No trabalho de He *et al.* (2016) é apresentado a estrutura de aprendizado residual. Foi mostrado que este tipo de rede é mais fácil de otimizar e podem se beneficiar consideravelmente do aumento de profundidade do modelo. Um *ensemble* das redes residuais alcançou um erro de 3.57% no teste do *ImageNet* e ganhou o primeiro lugar no desafio de classificação do ILSVRC em 2015. O modelo ResNet18 é uma RNC com 18 camadas de profundidade. O modelo possui aproximadamente 11 milhões de parâmetros e ocupa um espaço em disco de 43MB. Na Figura 4.8 é apresentada a arquitetura original do modelo.

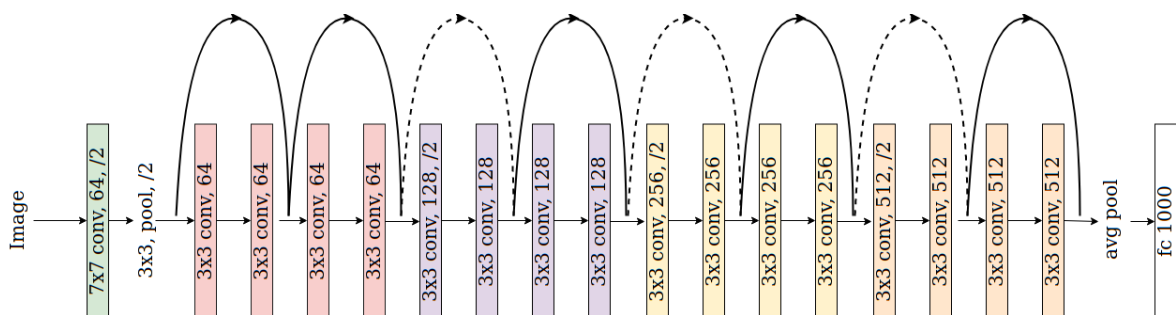


Figura 4.8: Arquitetura original do modelo Resnet18.  
Fonte: Adaptado de (HE *et al.*, 2016)

Como o problema proposto possui duas classes, as estruturas de ambos os modelos, *VGG16* e *ResNet18*, foram alteradas de forma a satisfazer as condições de classificação. A camada de saída dos modelos foi alterada para dois neurônios, que representam as classes *Limpo* e *Sujo*.



Tabela 4.1: parâmetros de treinamento para os modelos.

Parâmetros	Descrição
Tamanho do lote	8
Otimizador	SGD
Taxa de Aprendizado	0.001
Decaimento	0.1
Tamanho do passo	65
Função de Custo	Cross Entropy (CE)

## 4.4. Treinamento

A técnica de TL foi utilizada com os modelos como um ponto de partida para a solução proposta. Dois cenários, como descritos em Karpathy *et al.* (2016), foram usados para comparação dos resultados:

1. **RNC como extrator fixo de características:** os pesos são congelados para toda a rede, exceto os das camadas totalmente conectadas (FC). A última camada é substituída por uma nova com dois neurônios e pesos aleatórios. Somente as camadas FC são treinadas. Uma representação desse cenário é mostrada na Figura 4.9(a), em que o bloco azul indica que apenas os pesos de classificação são treinados.
2. **RNC com ajuste fino:** em vez da inicialização aleatória, a rede é inicializada com os pesos pré-treinados do modelo. Somente a última camada FC é inicializada aleatoriamente com dois neurônios. Durante o treinamento, todos os pesos da rede (camadas convolucionais e classificadoras) são re-treinados. Uma representação desse cenário é mostrada na Figura 4.9(b), onde os blocos azuis indicam que todos os pesos na rede são treinados.

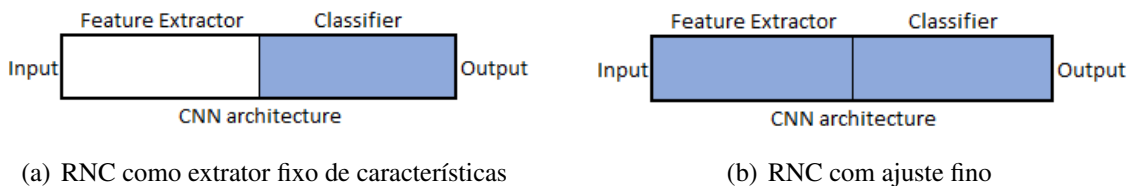


Figura 4.9: Cenários do treinamento.

Fonte: Própria.

O treinamento das redes foi fixado em 80 épocas, e a técnica de validação cruzada foi usada para observação de cada cenário. Os principais parâmetros de treinamento utilizados são apresentados na Tabela 4.1.

As imagens são carregadas nos modelos em pequenos lotes com 8 imagens. A cada lote processado, os pesos são corrigidos com o otimizador SGD aplicado em função do erro médio do lote. O erro avaliado com a função CE mede o desempenho do modelo cuja saída é um valor

de ativação que indica o neurônio mais acionado para a imagem de entrada. A função de custo CE pode ser descrita conforme Equação 4.1.

$$loss(x, class) = -x[class] + \log \left( \sum_j \exp(x[j]) \right) \quad (4.1)$$

onde  $x$  é o dado de entrada;  $class$  é a classe *target*;  $x[class]$  pode ser interpretado como o valor de ativação para a classe positiva e  $x[j]$  o valor de ativação para todas as  $j$  classes do modelo (*Limpo e Sujo*). Este critério combina as funções de erro *Log Softmax* e a *Negative Log Likelihood*.

## 4.5. Validação em campo

Com o sistema treinado, foi possível a validação em campo simulando o que será um sistema em tempo real de inspeção. Para tal, uma nova visita foi realizada no porto de Tubarão entre os dias 24 e 27 de novembro de 2019.

O processo de validação foi realizado com uma webcam modelo *logitech c920 Pro Full HD* conectada a um notebook para processar as informações com a rede já treinada e salva. Na Figura 4.10 é possível observar uma sequência que representa o processo simplificado.

Primeiro, é realizado a captura do frame do vídeo com a câmera. Em seguida, o frame é processado para ser inserido na rede de acordo às configurações necessárias de entrada. Nesta etapa é realizado uma normalização e o ajuste da imagem para uma resolução de  $224 \times 224$  pixels.

A seguir, a imagem passa pela RNC e a resposta de classificação é armazenada. Assim, esta informação é inserida em tempo real no frame do vídeo para que o usuário possa ver a resposta. Todo o processo é executado em um loop e pode ser interrompido a qualquer momento, caso o usuário entre com um comando específico no teclado.

## 4.6. Localização discriminativa

Após a validação em campo, algumas observações de localização discriminativa foram realizadas como forma de localizar os pontos de ativação nos dados usado no modelo para classificação. Esta etapa foi usada para corroborar os resultados de validação e testes em campo.

A maioria dos modelos de RNC é treinada com rótulos no nível da imagem e as camadas convolucionais aprendem representações hierárquicas para tomar sua decisão de classificação (BUDUMA e LOCASCIO, 2017). No entanto, trabalhos como o de Bazzani *et al.* (2016), mostraram que uma RNC tem a capacidade de localizar objetos nas imagens sem nem mesmo saber sua localização com antecedência.

Para aprender recursos profundos para localização discriminativa, Zhou *et al.* (2016)

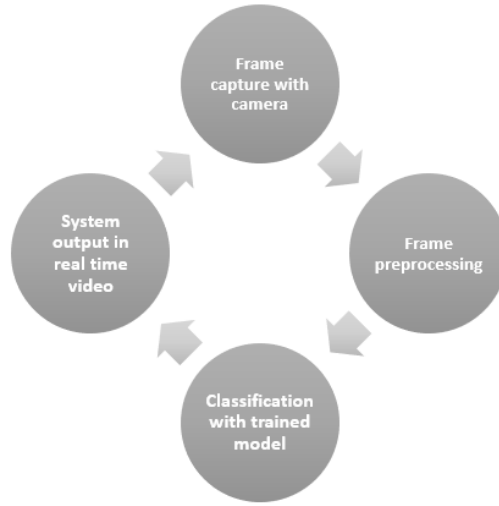


Figura 4.10: Sequência do processo de validação.  
Fonte: Própria.

propôs uma técnica para gerar Mapas de Ativação de classe (CAM) usando o Global Average Pooling (GAP) na RNC, Figura 4.11. O CAM permite a visualização das pontuações de classe previstas em qualquer imagem, destacando as partes discriminativas dos objetos detectados pelo modelo da RNC. Isto é possível pois o GAP gera um vetor de valores no qual cada item é a média global de um mapa de recurso da camada anterior. Logo, cada peso que conecta o vetor de saída do GAP com a camada de classificação guarda uma informação de contribuição daquele mapa para a classe do objeto detectado.

O CAM é fornecido pela Equação 4.2, em que  $w_k^c$  é o peso correspondente à classe  $C$  para o  $k$ -ésimo valor do vetor após o GAP e  $f_k(x,y)$  representa o  $k$ -ésimo valor de ativação após o GAP, referente à última camada convolucional na localização espacial  $(x,y)$ . Isto é uma soma linear ponderada das contribuições de cada padrão detectado nos mapas de recursos. De acordo Zhou *et al.* (2016), com apenas uma interpolação do CAM para o tamanho da imagem de entrada é possível identificar visualmente as regiões mais relevantes na classificação.

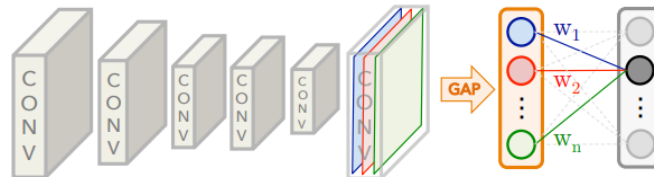


Figura 4.11: GAP na última camada convolucional.  
Fonte: Adaptado de Zhou *et al.* (2016)

$$M_c(x,y) = \sum_k w_k^c f_k(x,y) \quad (4.2)$$

Uma implementação de localização discriminativa mais robusta é encontrada em Selvaraju *et al.* (2017). Neste caso, a saída do mapa de recursos é capturada na última camada convolucional da rede, e cada canal da camada é ponderado pelo gradiente da classe ativada com relação ao respectivo canal. Para os autores esta abordagem generaliza o CAM e pode ser aplicada em modelos de RNC mais complexos com várias camadas de classificação.

## 5. Resultados e Discussão

Neste capítulo serão apresentados os resultados dos testes realizados e as discussões da validação do sistema.

Os resultados dos modelos foram avaliados em cada cenário treinado com base nas métricas de precisão, revocação e F-1 Score. A observação da capacidade de reconhecimento da aglomeração de sujeira, objetivo da pesquisa, pôde ser avaliada com uma validação em campo e aplicação do método de localização discriminativa.

Para as métricas, assumindo-se a classe *Sujo* como a classe positiva na classificação, pode-se dizer que: a precisão representa a proporção de acerto nas predições realizadas da classe *Sujo* pelos modelos. O recall representa a proporção de imagens com acúmulo de sujeira que foram classificadas corretamente nas amostras que devem ser classificadas como com acúmulo de sujeira.

### 5.1. Cenário 1: RNC como extrator fixo de características

Os resultados de teste para as dez rodadas do cenário 1 estão apresentados na Tabela 5.1. Pode-se observar os valores de erro, acurácia, média e desvio padrão de cada modelo.

Tabela 5.1: Erro e acurácia dos modelos VGG16 and ResNet18 no cenário 1.

Rodadas	VGG16		ResNet18	
	Erro	Acurácia	Erro	Acurácia
1	0.61607	0.75000	0.55296	0.76250
2	0.57405	0.68750	0.49225	0.81250
3	0.52146	0.77500	0.41456	0.83750
4	0.63685	0.68355	0.52222	0.74684
5	0.66657	0.62821	0.58369	0.66667
6	0.31370	0.91026	0.49037	0.73077
7	0.21140	0.94872	0.21918	0.96154
8	0.24979	0.94805	0.19412	0.98701
9	0.47380	0.81579	0.32815	0.89474
10	0.43595	0.89744	0.39296	0.89744
<b>Média</b>	0.46996	0.80445	0.41905	0.82975
<b>Desvio Padrão</b>	0.15565	0.11162	0.12870	0.09952

Para cada modelo, é necessário ter uma observação das métricas de avaliação. Desta forma, foram selecionados aqueles com acurácia mais próxima à média calculada dos resultados, conforme itens a seguir: (i) Round 9 do modelo VGG16 e (ii) Round 3 do modelo ResNet18. Comparando estes valores com o gráfico de boxplot das rodadas na Figura 5.1, observa-se que o valor médio dos dois modelos avaliados está de acordo com as suas respectivas medianas.

Os resultados das métricas estão apresentados na Tabela 5.2, assim como os resultados considerando a classe *Limpo* como positiva. Desta forma pode-se ter uma observação mais

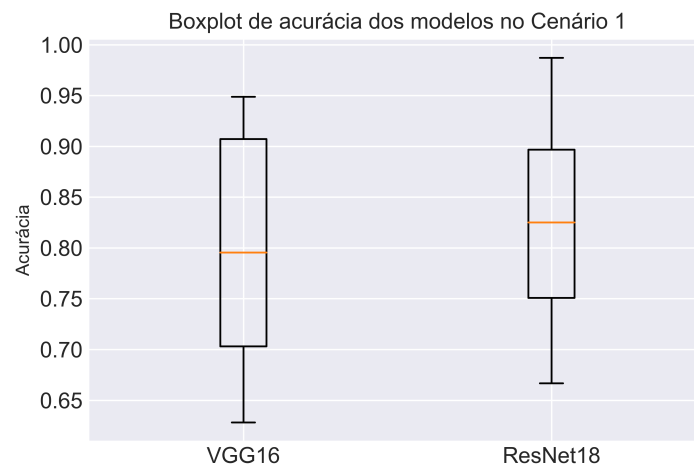


Figura 5.1: Boxplot com acurácia dos modelos no cenário 1.  
Fonte: Própria.

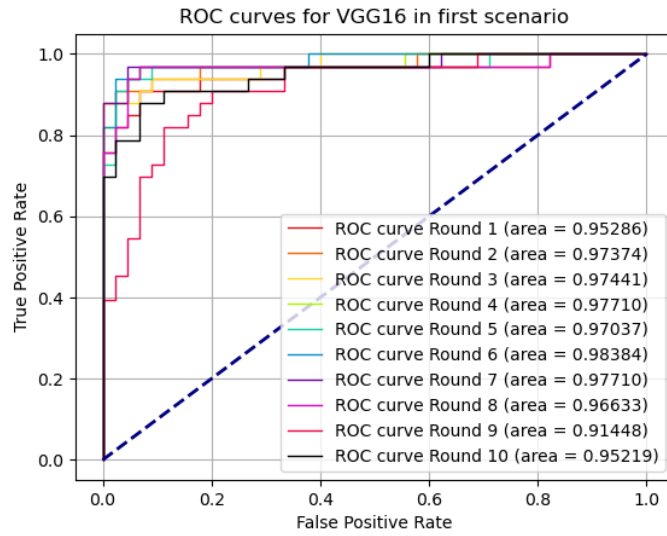
completa dos modelos com relação à detecção das duas classes. Percebe-se que, para estas duas avaliações, o modelo VGG16 apresentou melhor desempenho na identificação correta das imagens da classe que se busca identificar de fato, a classe *Sujo*, com uma revocação de 0.71875. O F-1 Score dos dois modelos foram próximos em função da precisão máxima do modelo ResNet18.

Tabela 5.2: Métricas para o cenário 1.

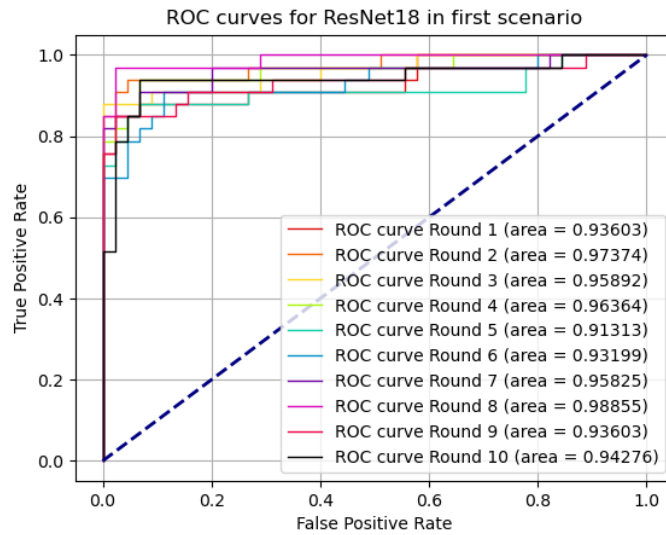
Classe positiva	Rodada 9 - VGG16			Rodada 3 - ResNet18		
	Precisão	Revocação	F-1 Score	Precisão	Revocação	F-1 Score
<i>Limpo</i>	0.81250	0.88636	0.84783	0.77966	1.00000	0.87619
<i>Sujo</i>	0.82143	0.71875	0.76667	1.00000	0.61765	0.76364

Para todos os testes, as curvas características de Operação do receptor, do inglês, Receiver Operating Characteristic (ROC), foram plotadas. Ela é uma representação gráfica que ilustra o desempenho (ou performance) de um sistema classificador binário à medida que o seu limiar de discriminação varia. O desempenho do modelo pode ser plotado em um gráfico bidimensional com a Taxa de Verdadeiro Positivo (True Positive Rate) em função da Taxa de falso positivo (False Positive Rate). Uma curva ROC traça em diferentes limiares de classificação. De acordo Faceli *et al.* (2011) seu uso foi inicialmente relatado na comparação de algoritmos e estendido mesmo na proposição de novos algoritmos. O ranking de classificação para gerar os gráficos foi realizado com base na probabilidade do resultado pertencer à classe *Sujo* e estão dispostos na Figura 5.2.

Observa-se nos gráficos da Figura 5.2 que os modelos possuem curvas bem próximas. A informação de área para cada rodada, Area Under the ROC Curve (AUC), quantifica a eficácia dos resultados, ou seja, quanto maior o valor de AUC, melhor. Comparando os dados, observa-



(a) Curva ROC para VGG16.



(b) Curva ROC para ResNet18.

Figura 5.2: Curva ROC para os modelos no cenário 1.

se valores bem elevados com AUC mínimo de 0.91448 para o modelo VGG16 e 0.91313 para o modelo ResNet18.

## 5.2. Cenário 2: RNC com ajuste fino

Os resultados para as rodadas do cenários 2 estão presentes na Tabela 5.3. Os dois modelos alcançaram resultados melhores, se comparados ao cenário 1. Essa melhora é relacionada ao treino dos filtros nas camadas de extração de características junto aos pesos das camadas de classificação dos modelos.

Tabela 5.3: Erro e acurácia dos modelos VGG16 and ResNet18 no cenário 2.

Rodadas	VGG16		ResNet18	
	Erro	Acurácia	Erro	Acurácia
1	0.34981	0.83750	0.47387	0.83750
2	0.46762	0.83750	0.37702	0.83750
3	0.18515	0.93750	0.18151	0.93750
4	0.74877	0.77215	0.64232	0.78481
5	0.71532	0.76923	0.74643	0.62821
6	0.07791	0.98718	0.22678	0.93590
7	0.12124	0.98718	0.05031	0.98718
8	0.07030	0.97403	0.03769	0.98701
9	0.28833	0.88158	0.16743	0.96053
10	0.21761	0.93590	0.18890	0.93590
<b>Média</b>	0.32421	0.89198	0.30923	0.88320
<b>Desvio Padrão</b>	0.23511	0.08021	0.23071	0.10721

Os resultados mostram uma acurácia média de 0,89198 para o modelo VGG16 e 0,88320 para o modelo ResNet18. Uma melhora de 8,8530 pontos percentuais (*pp*) e 5,3450 *pp*, respectivamente. O desvio padrão do modelo VGG16 reduziu de 0,11162, no cenário 1, para 0,08021, enquanto que o desvio padrão do ResNet18 aumentou de 0,09952, no cenário 1, para 0,10721.

Considerando a média da acurácia como parâmetro para determinar a análise das métricas de um dos modelos, tem-se que os modelos mais próximos da média são: (i) Round 9 do modelo VGG16 e (ii) Round 2 do modelo ResNet18. Novamente tem-se um modelo VGG16 com média próximo à mediana. No entanto, a média do modelo ResNet18 não está mais próximo à mediana, pois de acordo com o gráfico de boxplot na Figura 5.3 há um resultado considerado como outlier que influencia neste resultado. Desta forma, foi analisado também o modelo da rodada 3, mais próximo à mediana. As comparações estão na Tabela 5.4.

Considerando o resultado de outlier para a acurácia de 0.62821 do modelo ResNet18 na rodada 5, foi verificado que os dados de teste da rodada possuem muitas imagens à noite e estruturas com as grades de proteção (Ver Figura 4.2) e essas características não estão presentes no conjunto de treino de maneira equivalente. Isso evidencia a necessidade de dados de treino com essas características para deixar os modelos mais robustos.

Tabela 5.4: Métricas para o cenário 2.

Classe Positiva	Rodada 9 - VGG16			Rodada 2 - ResNet18			Rodada 3 - ResNet18		
	Precisão	Revocação	F-1 Score	Precisão	Revocação	F-1 Score	Precisão	Revocação	F-1 Score
<i>Limpo</i>	0.83019	1.00000	0.90722	0.77966	1.00000	0.87619	0.91837	0.97826	0.94737
<i>Sujo</i>	1.00000	0.71875	0.83636	1.00000	0.61765	0.76364	0.96774	0.88235	0.92308

As rodadas observadas na Tabela 5.4 indicam um índice de revocação muito alto para a classe *Limpo*, enquanto que o recall da classe *Sujo* parece estar no mesmo patamar do cenário 1. A observação da curva ROC e a análise da AUC para estes casos, plotados na Figura 5.4, também corroboram para o entendimento mais completo dos modelos. Percebe-se que todas



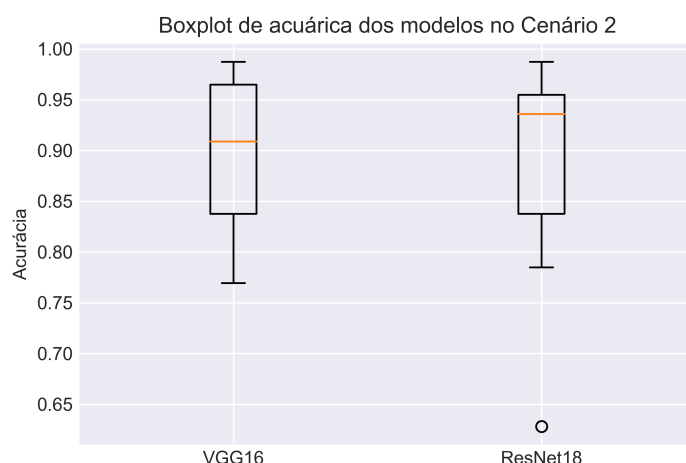


Figura 5.3: Boxplot com acurácia dos modelos no cenário 2.  
Fonte: Própria.

as curvas estão mais próximas da borda superior esquerda, região que, de acordo Faceli *et al.* (2011), é onde deve estar os resultados, pois quanto maior o valor de AUC, melhor.

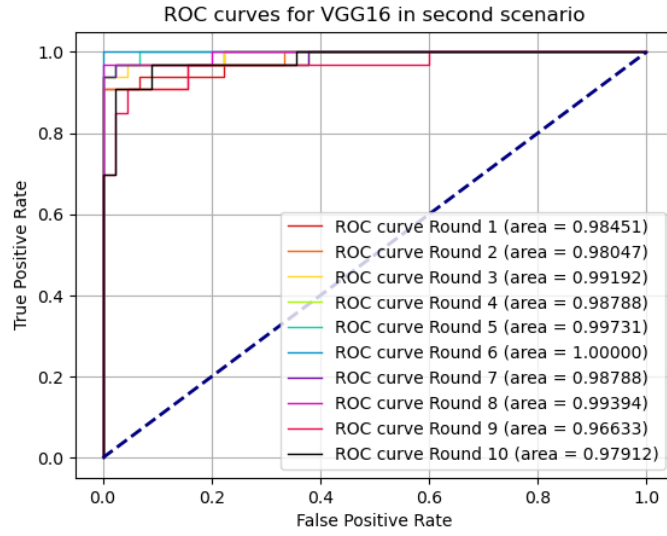
Com estes resultados, é interessante fazer uma observação quanto à rodada 9. Em função da divisão do dados, esta rodada teve um conjunto de treino com imagens da classe *Sujo* predominantemente do porto de Tubarão, apenas 3 imagens da classe *Sujo* pertenciam à mina de Alegria. Já o conjunto de teste desta rodada, 32 imagens, foi construído com imagens da classe *Sujo* pertencentes apenas à mina de Alegria. Para o cenário 2, o recall dessa rodada foi de 0.71875 para o modelo VGG16 e 0.90625 para o modelo ResNet18. Com isso é importante observar essa capacidade de generalizar o reconhecimento de um local para outro, pois as características do ambiente do porto são diferentes do ambiente da mina.

Para comparar o reconhecimento de sujeira de todos os modelos, a revocação da classe *Sujo* de todas as rodadas foi observado em boxplot, indicado na Figura 5.5. Pode-se perceber que o reconhecimento dos modelos no cenário 2 foi superior aos modelos no cenário 1, com uma mediana muito próxima 0.90 e valores do 3<sup>o</sup> quartil muito próximos de 1.0 para os dois modelos. O modelo VGG16 teve mais variações em função das rodadas, e o modelo ResNet18 também apresenta apenas um resultado considerado como outlier.

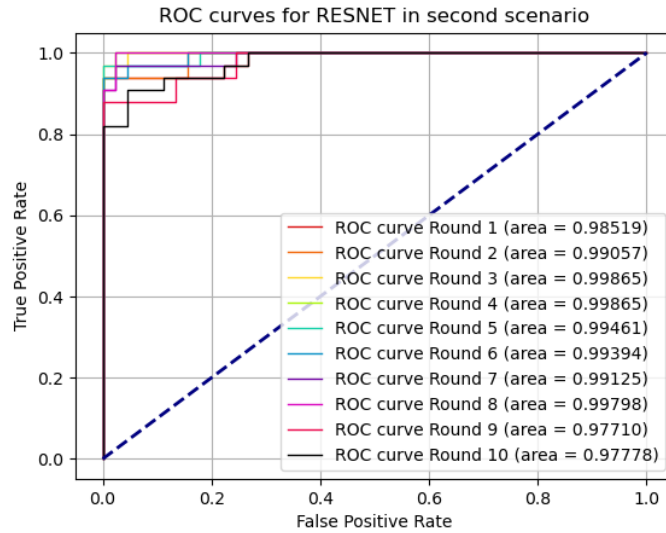
### 5.3. Validação de Campo

A validação de campo serviu como prova de conceito para o sistema ser integrado como um serviço em um sistema de inspeção robótica. Como o melhor resultado foi alcançado com o segundo cenário de treinamento, um modelo treinado em ajuste fino foi utilizado na validação.

No porto de Tubarão, foram realizadas 30 gravações para classificação em tempo real.



(a) Curva ROC para VGG16.



(b) Curva ROC para ResNet18.

Figura 5.4: Curva ROC para os modelos no cenário 2.

Foi definido a realização de 15 gravações em estruturas com características de *Limpo* e 15 gravações em estruturas com características de *Sujo*, de forma a manter os dados balanceados para as duas classes. Alguns dos primeiros testes podem ser vistos na Figura 5.6, que mostra um frame de duas gravações diferentes.

Foi interessante observar a resposta em tempo real da Figura 5.6(b), pois o rolo em questão possui uma característica que não estava presente nos dados de treinamento: foco no rolo lateral com sujeira entre a superfície superior do rolo e a superfície inferior da borracha do TC. A maioria das imagens de treinamento tem sujeira na parte inferior dos rolos.

Para obter mais informações sobre as classificações, o sistema foi implementado para mostrar a probabilidade de classificação como um indicador de confiança do sistema para o

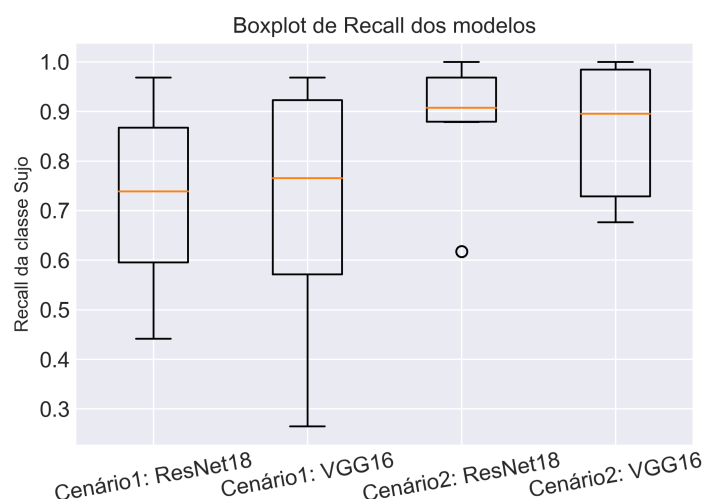


Figura 5.5: Boxplot.



(a) Classificado como *Limpo*



(b) Classificado como *Sujo*

Figura 5.6: Primeira validação em campo, com uma classificação *Limpo* em (a) e uma classificação *Sujo* em (b).

usuário, conforme mostrado na Figura 5.7.

Na Figura 5.7(a), é possível ver uma classificação como *Limpo* com uma alta probabilidade de 91,46%. As figuras 5.7(b) e 5.7(c) também apresentam classificações com alta probabilidade, mas desta vez reconhecendo o acúmulo de sujeira nas estruturas, respectivamente 87,66% e 97,85%. A figura 5.7(d) indica uma classificação de *Sujo*, com uma probabilidade de 75,81%. Mesmo a sujeira em um nível não tão acumulado, como nas figuras 5.7(b) e 5.7(c), pode ser reconhecida com probabilidade razoável. Isto serve como parâmetro para a equipe de manutenção definir qual estrutura deve ser limpa.

Com os 30 vídeos de validação, foram observados os rolos nos quais a gravação estava mais estável, sem tremulação. Com as observações e análises das classificações a matriz de confusão foi gerada conforme Figura 5.8. Com as equações de revocação, precisão e F1-Score tem-se resultados para a classe *Sujo* de 0,77, 0,74 e 0,75, respectivamente. Comparando com as variações na Figura 5.5, é um resultado de acordo com as observações de teste.



Figura 5.7: Quadros de validação de campo com probabilidades de classificação em tempo real.

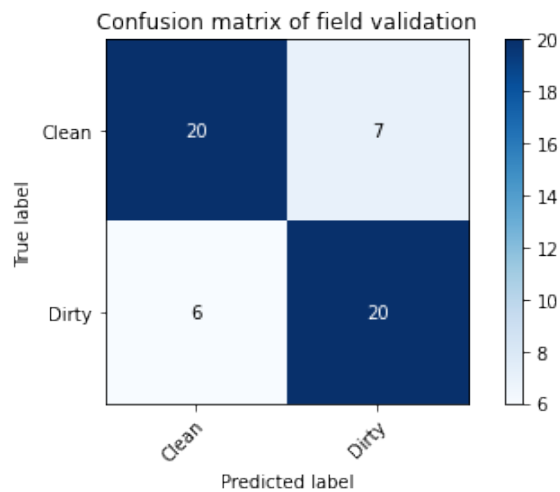


Figura 5.8: Matriz de Confusão da Validação em Campo.

Após a validação em campo, foram realizadas 30 rodadas com 100 quadros de um vídeo para analisar o tempo médio de processamento dos quadros durante a classificação. O experimento foi realizado em uma CPU Intel® Core™ i5-7200U a 2,50 GHz × 4 com memória de 7,6 GiB e Intel® HD Graphics 620. A média calculada de Quadros por Segundo (FPS) foi de 3,1789, ou seja, 3 FPS.

A técnica de localização discriminativa ajudou a trazer mais valor para os resultados

encontrados. Esta opção foi considerada ao final das avaliações como forma de avaliar o que a rede está olhando e mapear os resultados com os dados de validação em campo. Com as imagens na Figura 5.9 pode-se ver que o modelo está olhando para a área de interesse. A cor vermelha indica a área que foi mais ativada pela rede. Em azul fraco, tem-se a região que tem menos importância na classificação.

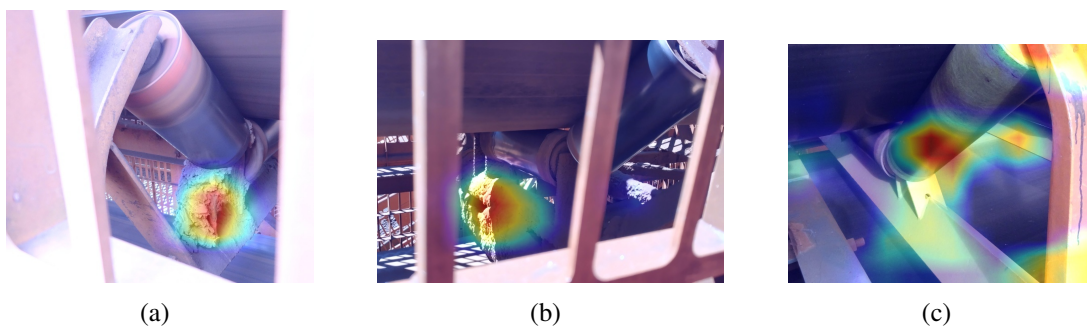


Figura 5.9: Localização discriminativa aplicada ao modelo.

## 6. Conclusão

Encontrar soluções que visam a melhora das inspeções, aumento da segurança e redução de custos é essencial para toda empresa no setor mineral. A detecção automática da aglomeração de resíduos usando IA pode auxiliar no processo de inspeção dos TC, ajudando na tomada de decisão dos setores de manutenção e aumentando a segurança nos processos.

O uso de ferramentas de IA se mostrou adequado para o problema apresentado, viabilizando um treinamento eficiente no reconhecimento das características desejadas: aglomeração da sujeira nas estruturas de TC. Os experimentos realizados demonstraram que o sistema é capaz de reconhecer as aglomerações de sujeira, mas, em função da quantidade de dados disponíveis para treino, pode ter sua detecção prejudicada em função de características específicas como imagens à noite.

Com os cenários apresentados, foi observado que o ajuste total do modelo de uma RNC é o mais adequado, alcançando resultados de revocação nos dados de teste na faixa de 0,70 até 0,97. Com as comparações realizadas com as curvas ROC verificou-se que os modelos utilizados são eficientes na classificação e conseguem separar bem uma estrutura da classe *Limpo* de uma estrutura da classe *Sujo*.

Com o sistema de inspeção funcional, as validações de campo foram realizadas sendo possível observar a aplicabilidade do sistema, assim como a sua performance na generalização de características importantes. Esta etapa foi crucial para observação do sistema em tempo real e como ele pode ser afetado em situações não controladas. Os resultados de campo demonstraram estar de acordo com os testes realizados, e os dados de revocação apresentaram um resultado dentro da variação máxima nos dados de teste para o cenário 2.

## **7. Trabalhos Futuros**

Como sequência para novos trabalhos na busca de implementar o sistema no ROSI, pode-se levantar os seguintes pontos:

- Coleta de nova base de dados, complementando a atual;
- Treinamento de sistema para reconhecimento do objeto rolo, sendo possível a classificação da imagem apenas quando existir um rolo do TC na imagem;
- Acoplar o sistema final na plataforma móvel do ROSI e realizar testes de desempenho.

# Referências Bibliográficas

- AGHDAM, H. H., HERAVI, E. J. *Guide to convolutional neural networks: a practical application to traffic-sign detection and classification*. Springer, 2017.
- ALPORT, M., GOVINDER, P., PLUM, S., et al.. “Identification of conveyor belt splices and damages using neural networks.” *Bulk Solids Handling*, v. 21, n. 6, pp. 622–627, 2001.
- ARBIB, M. A. *Brains, machines, and mathematics*. Springer-Verlag New York, 1987.
- BAZZANI, L., BERGAMO, A., ANGUELOV, D., et al.. “Self-taught object localization with deep networks”. Em: *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–9. IEEE, 2016.
- BHATIA, Y., RAI, R., GUPTA, V., et al.. “Convolutional neural networks based potholes detection using thermal imaging”, *Journal of King Saud University-Computer and Information Sciences*, 2019.
- BISHOP, C. *Pattern Recognition and Machine Learning*, v. 1. Springer-Verlag New York, 2006.
- BRADSKI, G. “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.
- BROWNLEE, J. “A Gentle Introduction to Transfer Learning for Deep Learning”. 2017. Disponível em: <<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>>. Acessado em: 08/06/2019.
- BUDUMA, N., LOCASCIO, N. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. "O’Reilly Media, Inc.", 2017.
- CARVALHO JÚNIOR, J. R. D. *Processamento digital de imagens para a identificação automática de falhas em rolos dos transportadores de correias*. Tese de Mestrado, Universidade Federal de Ouro Preto (UFOP), 2018.
- CONVEYOR EQUIPMENT MANUFACTURES ASSOCIATION. *Belt Conveyors for Bulk Materials*. K-Kom, 2007. ISBN: 1-891171-59-3.



- DA SILVA, I. N., SPATTI, D. H., FLAUZINO, R. A., et al.. “Artificial neural networks”, *Cham: Springer International Publishing*, 2017.
- DATA SCIENCE ACADEMY. “Deep Learning Book”. 2019. Disponível em: <<http://www.deeplearningbook.com.br>>. Acessado em: 07/06/2019.
- FACELI, K., LORENA, A. C., GAMA, J., et al.. *Inteligência Artificial: Uma abordagem de aprendizado de máquina*. 2011.
- FERNÁNDEZ, A., GARCÍA, S., GALAR, M., et al.. *Learning from imbalanced data sets*. Springer, 2018.
- GARCIA, G., ROCHA, F., TORRE, M., et al.. “ROSI: A Novel Robotic Method for Belt Conveyor Structures Inspection”. Em: *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 326–331. IEEE, 2019.
- GONZALEZ, R., WOODS, R. “Digital image processing, 4th edn. ISBN: 9780133356724”. 2017.
- GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep learning*. MIT press, 2016.
- GUASTALA, F. A. “Estudo do efeito da degradação térmica nas propriedades mecânicas de correias transportadoras operando em alta temperatura.” 2013.
- HAMZELOO, E., MASSINAEI, M., MEHRSHAD, N. “Estimation of particle size distribution on an industrial conveyor belt using image analysis and neural networks”, *Powder technology*, v. 261, pp. 185–190, 2014.
- HAYKIN, S. S., HAYKIN, S. S., HAYKIN, S. S., et al.. *Neural networks and learning machines*, v. 3. Pearson Upper Saddle River, 2009.
- HE, K., ZHANG, X., REN, S., et al.. “Deep residual learning for image recognition”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- KARIM, R. “Illustrated: 10 CNN Architectures”. July 2019. Disponível em: <<https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#c5a6>>.
- KARPATHY, A., OTHERS. “Cs231n convolutional neural networks for visual recognition”, *Neural networks*, v. 1, 2016.
- MADHAVAN, S. *Mastering Python for Data Science*. Packt Publishing Ltd, 2015.

- MALAGONI, R. A., DOS SANTOS, D. A. “PROJETO DE CORREIAS TRANSPORTADORAS: UM ESTUDO COMPUTACIONAL DE COMPARAÇÃO DOS MÉTODOS CEMA E PRÁTICO”, *HOLOS*, v. 3, pp. 358–369, 2014.
- MASCI, J., MEIER, U., CIRESAN, D., et al.. “Steel defect classification with max-pooling convolutional neural networks”. Em: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2012.
- MATTHEE, H. “Using IR Imaging to Diagnose Problems in Conveyors and Crushers”, *Engineering and Mining Journal*, v. 210, n. 2, pp. 38, 2009.
- MINHAS, M. S., ZELEK, J. “Anomaly Detection in Images”. 2019.
- NAIXUN, H., TAO, C., RUIQING, N., et al.. “Object-Oriented Open Pit Extraction Based on Convolutional Neural Network, A Case Study in Yuzhou, China”. Em: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 9435–9438. IEEE, 2019.
- NASCIMENTO, R. S. D. *Inspeção de transportadores de correia: arquitetura integrada para uma plataforma de inspeção com uso de VANTs*. Tese de Mestrado, Universidade Federal de Ouro Preto (UFOP), 2018.
- OLIVIER, L. E., MARITZ, M. G., CRAIG, I. K. “Deep Convolutional Neural Network for Mill Feed Size Characterization”, *IFAC-PapersOnLine*, v. 52, n. 14, pp. 105–110, 2019.
- PASZKE, A., GROSS, S., CHINTALA, S., et al.. “Automatic differentiation in PyTorch”. Em: *NIPS-W*, 2017.
- QIAO, T., ZHAO, B., SHEN, R., et al.. “Infrared image detection of belt longitudinal tear based on SVM”, 2013.
- RUSSELL, S., NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd ed. USA, Prentice Hall Press, 2009. ISBN: 0136042597.
- SELVARAJU, R. R., COGSWELL, M., DAS, A., et al.. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. Em: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- SIMONYAN, K., ZISSERMAN, A. “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.
- SUNDARARAJAN, D. *Digital Image Processing: A Signal Processing and Algorithmic Approach*. Springer, 2017.

- TOFFOLO, T. A. M. “Otimização do fluxo de produtos de uma empresa mineradora”, 2009.
- TORREY, L., SHAVLIK, J. “Transfer learning”. Em: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI Global, pp. 242–264, 2010.
- VALE. “Minério de Ferro e Pelotas”. 2018a. Disponível em: <<http://www.vale.com/brasil/PT/business/mining/iron-ore-pellets/Paginas/default.aspx>>. Acessado em: 12/11/2018.
- VALE. “Carajás”. 2018b. Disponível em: <<http://www.vale.com/brasil/PT/business/mining/iron-ore-pellets/Documents/carajas/index.html>>. Acessado em: 12/11/2018.
- YANG, W., ZHANG, X., MA, H. “An inspection robot using infrared thermography for belt conveyor”. Em: *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 400–404. IEEE, 2016.
- YANG, Y., MIAO, C., LI, X., et al.. “On-line conveyor belts inspection based on machine vision”, *Optik-International Journal for Light and Electron Optics*, v. 125, n. 19, pp. 5803–5807, 2014.
- ZHOU, B., KHOSLA, A., A., L., et al.. “Learning Deep Features for Discriminative Localization.” *CVPR*, 2016.