

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA  
CURSO DE ENSINO SUPERIOR DE ENGENHARIA MECATRÔNICA**

**Victor Ribeiro Domke**

**ESTUDO DE REDES NEURAIS CONVOLUCIONAIS E SUA APLICAÇÃO NA  
CLASSIFICAÇÃO DE DEFEITOS EM EMBALAGENS DE LENÇOS UMEDECIDOS**

**FLORIANÓPOLIS, 2022.**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA  
CURSO DE ENSINO SUPERIOR DE ENGENHARIA MECATRÔNICA**

**Victor Ribeiro Domke**

**ESTUDO DE REDES NEURAIS CONVOLUCIONAIS E SUA APLICAÇÃO NA  
CLASSIFICAÇÃO DE DEFEITOS EM EMBALAGENS DE LENÇOS UMEDECIDOS**

Trabalho de Conclusão de Curso submetido  
ao Instituto Federal de Educação, Ciência  
e Tecnologia de Santa Catarina como parte  
dos requisitos para obtenção do título de  
Bacharel em Engenharia Mecatrônica.

Orientador:  
Prof. Mauricio Edgar Stivanello, Dr. Eng.

**FLORIANÓPOLIS, 2022.**

Ficha de identificação da obra elaborada pelo autor.

Domke, Victor

**ESTUDO DE REDES NEURAIS CONVOLUCIONAIS E SUA APLICAÇÃO  
NA CLASSIFICAÇÃO DE DEFEITOS EM EMBALAGENS DE LENÇOS UMEDECIDOS**  
/ Victor Domke; orientação de Maurício Stivanello.  
- Florianópolis, SC, 2022.  
46 p.

**Trabalho de Conclusão de Curso (TCC) - Instituto Federal  
de Santa Catarina, Câmpus Florianópolis. Bacharelado  
em Engenharia Mecatrônica. Departamento  
Acadêmico de Metal Mecânica.  
Inclui Referências.**

1. Aprendizado profundo. 2. Visão computacional.  
3. Indústria de higiene pessoal. 4. Lenço umedecido. 5.  
Redes neurais convolucionais. I. Stivanello, Maurício.  
II. Instituto Federal de Santa Catarina. III. ESTUDO  
DE REDES NEURAIS CONVOLUCIONAIS E SUA APLICAÇÃO NA  
CLASSIFICAÇÃO DE DEFEITOS EM EMBALAGENS DE LENÇOS UMEDECIDOS.

**ESTUDO DE REDES NEURAIS CONVOLUCIONAIS E SUA APLICAÇÃO NA  
CLASSIFICAÇÃO DE DEFEITOS EM EMBALAGENS DE LENÇOS  
UMEDECIDOS**

**VICTOR RIBEIRO DOMKE**

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em 2022 e aprovado na sua forma final pela banca examinadora do Curso Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 10 de Março, 2022.

Banca Examinadora:

---

Maurício Edgar Stivanello, Dr. Prof , Orientador  
Instituto Federal de Santa Catarina

---

Delcino Picinin Junior, Dr. Prof.  
Instituto Federal de Santa Catarina

---

Francisco Rafael Moreira da Mota, Dr. Prof.  
Instituto Federal de Santa Catarina

## **AGRADECIMENTOS**

Meus agradecimentos iniciais dedico à todos meus familiares, em especial, minha mãe Christine Ribeiro e meu irmão Arthur Ribeiro Domke pelo amor e compreensão que tiveram comigo no decorrer dessa jornada acadêmica. Segundamente, mas não menos importante, devo agradecer a minha companheira Manuella, que está presente em momentos difíceis e também muito bons da minha vida. E hoje podendo celebrar mais uma conquista ao seu lado, me enche de felicidade.

Aos meus grandes amigos, Gabriel Garcia, Gustavo Marcon e João Victor Cunha gostaria de deixar um grande agradecimento por me apoiarem durante todos estes anos e de certa forma permitirem estar ao lado de vocês.

Por fim, agradeço a todos professores, servidores e técnicos do Instituto Federal de Santa Catarina. Por de alguma forma terem contribuído com meu caminho na graduação. Em especial, sou grato ao professor Maurício Stivanello por me instruir no decorrer deste trabalho. Ao realizar minha transferência para a Engenharia Mecatrônica, foi através de suas aulas que eu pude ter certeza de que havia feito uma boa escolha.

## RESUMO

Este trabalho propõe a utilização de técnicas de aprendizado de máquina com foco em Redes Neurais Convolucionais, para a elaboração de sistemas de inspeção de qualidade. Os sistemas possuem o intuito de classificar imagens de embalagens de lenços umedecidos de uma linha de produção, em embalagem ok, embalagem com falha na tampa ou embalagem com lenço pra fora. Para isso, foram selecionados 3 produtos distintos fabricados em uma máquina presente em uma indústria de higiene pessoal. Cada produto possui sua rede neural individual. Para o desenvolvimento dos classificadores foi utilizada principalmente a biblioteca *Fastai*, com algoritmos escritos em Python, e a plataforma utilizada para os testes e treinamentos foi o Google Collaboratory. Todos os requisitos propostos no trabalho foram atendidos. Os classificadores dos produtos, demaquilante, higiene íntima e wipe germ home obtiveram acurácias de 91,8%, 96,0% e 96% respectivamente. Todas as CNNs usaram a arquitetura ResNet101 e o tempo médio de inferência para apenas uma imagem de 0,3s.

**Palavras-chave:** Aprendizado Profundo; Indústria de Higiene Pessoal; Lenço umedecido; Redes Neurais Convolucionais; Sistema de Inspeção de Qualidade; Visão computacional.

## ABSTRACT

This work proposes the use of machine learning techniques focused on Convolutional Neural Networks to develop quality inspection systems. The systems have the intention of classifying images of wet wipes packages from a production line, in package ok, package with a flaw on the lid or package with the wipe out. For this, three different products were selected, manufactured in a machine present in a personal hygiene industry. Each product has its individual neural network. To develop the classifiers it was used mainly the library *Fastai*, with algorithms written in Python, and the platform used for testing and training was the Google Collaboratory. All the requirements proposed in the work were met. The classifiers for the products, demaquilant, intimate hygiene and wipe germ home obtained accuracies of 91.8%, 96.0% and 96% respectively. All CNNs used the ResNet101 architecture and the average inference time for only one image of 0.3s.

Keywords: Computer Vision; Convolutional Neural Networks; Deep Learning; Neural Networks; Personal Care Industry; Quality Inspection System; Wet Wipes.

## LISTA DE FIGURAS

Figura 1 – Produtos utilizados no trabalho. . . . .	15
Figura 2 – Máquina <i>Flow Pack</i> . . . . .	15
Figura 3 – Exemplos de falhas. . . . .	16
Figura 4 – Perceptron simplificado . . . . .	18
Figura 5 – Estrutura de um Perceptron . . . . .	19
Figura 6 – Função Sigmoides . . . . .	19
Figura 7 – Multilayer Perceptron . . . . .	20
Figura 8 – Representação básica de uma CNN . . . . .	21
Figura 9 – Processo de convolução para os dois primeiros passos. . . . .	22
Figura 10 – Exemplo de <i>stride</i> . . . . .	22
Figura 11 – Exemplo de padding. . . . .	23
Figura 12 – Max pooling e Average pooling. . . . .	24
Figura 13 – Operação de Flattening. . . . .	25
Figura 14 – Exemplos de curvas sub-ajustada, adequada e sobre-ajustada, respectivamente. . . . .	25
Figura 15 – Bloco residual: realimentação da entrada X. . . . .	26
Figura 16 – Bloco residual: realimentação da entrada X. . . . .	27
Figura 17 – Níveis de organização da <i>fastai</i> . . . . .	28
Figura 18 – Exemplo de matriz de confusão. . . . .	30
Figura 19 – Categorias que o sistema deve classificar . . . . .	33
Figura 20 – Câmera utilizada para a obtenção das imagens . . . . .	34
Figura 21 – Imagens capturadas. . . . .	35
Figura 22 – Organização das pastas das imagens adquiridas . . . . .	36
Figura 23 – Perda x Taxa de aprendizado (Wipe) . . . . .	38
Figura 24 – Matriz de confusão CNN Higiene Íntima . . . . .	40
Figura 25 – Matriz de confusão CNN Demaquilante . . . . .	40
Figura 26 – Matriz de confusão CNN Wipe Germ . . . . .	41



## LISTA DE TABELAS

Tabela 1 – Categorização dos resultados de predição para um modelo de aprendizagem de máquina. . . . .	29
Tabela 2 – Quantidade de imagens por classe e produto . . . . .	35
Tabela 3 – Resultados finais das três CNNs utilizando ResNet101 . . . . .	39
Tabela 4 – Tempo de treinamento gasto por cada CNN . . . . .	42
Tabela 5 – Tempo de inferência para uma imagem dos classificadores . . . . .	42

## **LISTA DE CÓDIGOS**

Código 1 – Código da estrutura do DataBlock (Íntimo) . . . . .	36
--	----

## LISTA DE ABREVIATURAS E SIGLAS

ABIHPEC	Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos
CNN	Convolutional Neural Network (Redes Neurais Convolucionais)
CTC	Camada Totalmente Conectada
CV	Computer Vision (Visão Computacional)
DL	Deep Learning (Aprendizado Profundo)
FP	Flow Pack
HPPC	Setor de Higiene Pessoal, Perfumaria e Cosméticos
IA	Inteligência Artificial
ML	Machine Learning (Aprendizado de Máquina)
MLP	<i>Multilayer Perceptron</i>
RNA	Redes Neurais Artificiais ( <i>Artificial Neural Networks</i> )
SIQ	Sistema de Inspeção de Qualidade
TL	<i>Transfer Learning</i> (Aprendizado por transferência)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos</b>	<b>13</b>
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>14</b>
<b>2.1</b>	<b>A Indústria HPPC e mudanças tecnológicas</b>	<b>14</b>
2.1.1	Lenços umedecidos	14
2.1.2	Máquina <i>Flow Pack</i>	15
2.1.3	Defeitos analisados	15
<b>2.2</b>	<b>Visão Computacional e RNAs</b>	<b>16</b>
2.2.1	Visão Computacional (CV)	16
2.2.2	Redes Neurais Artificiais (RNAs)	17
<b>2.3</b>	<b>Descrição geral das CNNs</b>	<b>20</b>
2.3.1	Camada de Convolução	21
2.3.2	Camada de Pooling	23
2.3.3	Camada totalmente conectada	24
<b>2.4</b>	<b>Redes Neurais Residuais (ResNet)</b>	<b>25</b>
2.4.1	Transfer Learning	26
2.4.2	Biblioteca Fastai	27
2.4.3	Aumento de dados (Data Augmentation)	28
2.4.4	Métricas	29
2.4.4.1	<i>Acurácia</i>	29
2.4.4.2	<i>Precision (Precisão)</i>	29
2.4.4.3	<i>Recall</i>	30
2.4.4.4	<i>F1-Score</i>	30
2.4.4.5	<i>Matriz de confusão</i>	30
<b>2.5</b>	<b>Trabalhos Correlatos</b>	<b>31</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>32</b>
<b>3.1</b>	<b>Discussão sobre os requisitos do sistema</b>	<b>32</b>
3.1.1	Requisitos funcionais e não funcionais	32
<b>3.2</b>	<b>Descrição da solução</b>	<b>33</b>
3.2.1	Preparação e obtenção das bases de dados	34
3.2.2	Treinamentos dos modelos	36
<b>4</b>	<b>APRESENTAÇÃO DOS RESULTADOS</b>	<b>39</b>
<b>4.1</b>	<b>Resultados de classificação</b>	<b>39</b>
<b>4.2</b>	<b>Custos computacionais</b>	<b>41</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>43</b>
<b>5.1</b>	<b>Sugestões para trabalhos futuros</b>	<b>43</b>
	<b>REFERÊNCIAS</b>	<b>44</b>

## 1 INTRODUÇÃO

Este trabalho trata de uma aplicação de Redes Neurais Convolucionais (CNN do inglês *Convolutional Neural Networks*) com foco na indústria de higiene pessoal. O estudo foi realizado em uma empresa localizada na cidade de São José no estado de Santa Catarina, atuando no mercado de HPPC há 30 anos e busca ser a maior referência neste setor.

De acordo com o Painel de Dados de Mercado da ABIHPEC (Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos), o setor de HPPC obteve em 2020 um crescimento de 5,8% em vendas *ex-factory*, quando comparado com o mesmo período de 2019 (ABIHPEC, 2021). Apesar de todas as incertezas relacionadas ao mercado interno e externo devido a pandemia do Covid-19, pode-se observar que esse setor obteve bons resultados.

Dito isto, as empresas relacionadas a este setor e inclusive de outros procuram introduzir em suas linhas de produção, sistemas de inspeção por meio de visão computacional, visando realizar controle de qualidade e automação em seus processos de produção. "O controle da qualidade é um aspecto de grande importância da indústria, pois afeta diretamente a forma como os clientes enxergam a marca e a empresa, e pode atrair ou afastar novos consumidores."(KUGLER, 2019).

Com isso, vê-se a necessidade da elaboração de um sistema de inspeção em uma máquina chamada de *Flow Pack*, por meio da utilização de visão computacional com o objetivo otimizar atividades que atualmente são desempenhadas por funcionários e substituí-los por um Sistema de Inspeção de Qualidade (SIQ) que realize a mesma tarefa de forma rápida e confiável, podendo assim realocar essa mão de obra para outras funções.

O processo em questão é a classificação e identificação de defeitos embalagens de toalhas umedecidas produzidas. A indústria onde o estudo será realizado possui uma linha de 9 produtos distintos, entretanto, para este projeto será utilizado somente três produtos desta linha, tais como: lenço umedecido *wipe germ home* álcool 70° inpm 50 un, lenço umedecido *cottonbaby lady* demaquilante 20un e lenço umedecido *cottonbaby lady* higiene íntima 20un (figura 1).

A classificação de objetos é uma atividade que nós, seres humanos, não encontramos dificuldades para realizar, e atualmente, tão pouco para os computadores. A tecnologia mais recente que possibilitou os computadores a interpretar imagens são as CNNs. "A CNN é um tipo de rede neural artificial (RNA) com aprendizado supervisionado que processa cada uma de suas camadas imitando o córtex visual do olho humano."(MAJIN, 2021).

Em meados dos anos 2000, Geoffrey Hinton e Ruslan Salakhutdinov, publi-

caram um artigo trazendo o termo de aprendizado profundo. As CNNs são uma das subáreas do *deep learning* ou aprendizado profundo. O aprendizado profundo é uma técnica de aprendizagem de máquina que consiste em ensinar os computadores a realizar ou interpretar tarefas que nós seres humanos estamos acostumados a fazer, como por exemplo aprender.

Sendo assim, o estudo e implementação das redes neurais convolucionais para a detecção de falhas na linha de produção dos produtos citados anteriormente será o foco principal deste presente trabalho. Para isso será utilizado de técnicas de *deep learning* com a finalidade de criar algoritmos que classifiquem e identifiquem os defeitos nas embalagens dos dois produtos citados anteriormente.

## 1.1 Objetivos

Nas seções a seguir será apresentado o objetivo geral e os objetivos específicos deste trabalho.

### 1.1.1 Objetivo Geral

O objetivo geral deste projeto é desenvolver um sistema de visão computacional que classifique imagens de produtos produzidos em uma linha de produção em uma indústria de higiene pessoal.

### 1.1.2 Objetivos Específicos

Abaixo estão listados os objetivos específicos referente a este trabalho:

- a) Aprofundar os estudos de redes neurais convolucionais, tais como, arquitetura neural e métodos de implementação;
- b) Estruturar uma base de imagens e pré processá-las;
- c) Definir, após obtida a base de dados, definir qual parcela irá ser utilizada para etapa de treinamento, e qual será para a etapa de validação do modelo neural;
- d) Implementar os algoritmos;
- e) Validar o modelo e analisar os resultados de desempenho;
- f) Documentar o desenvolvimento do projeto.

## 2 DESENVOLVIMENTO

No presente capítulo será abordado os conceitos básicos referente à visão computacional, redes neurais, arquiteturas, métricas e aumento de dados.

### 2.1 A Indústria HPPC e mudanças tecnológicas

A indústria de HPPC (Higiene Pessoal, Perfumaria e Cosméticos) vem passando por um crescimento ao longo de alguns anos, isso se dá devido à algumas mudanças no perfil de seus consumidores, no Brasil não seria diferente. Segundo a Euromonitor International (2016), o Brasil é o terceiro maior mercado consumidor do setor, ficando atrás apenas dos Estados Unidos e da China, com 16,5% e 10,3% do consumo mundial respectivamente.(ECONOMIA DE SERV., 2016)

Com esse avanço econômico neste setor, é visto que uma indústria atuante neste mercado, necessita de avanços tecnológicos em seus processos produtivos, para que assim obtenham melhores resultados e a um custo de produção mais reduzido. Atualmente, a empresa em questão possui uma linha de manufatura de toalhas umedecidas que depende de funcionários para realizar a inspeção visual dos produtos saídos da máquina. Sendo assim, o controle de qualidade fica na responsabilidade desses trabalhadores, gerando mais custo, encargos empregatícios e não conformidades com relação a exatidão da inspeção. Tendo isso em vista, nota-se a necessidade de utilizar de novas abordagens para este problema, para que haja a substituição deste trabalho que é dependente de um operário para tecnologias que farão a mesma função, porém sendo mais preciso e ágil.

O domínio de aplicação deste trabalho é a implementação de técnicas de aprendizado profundo que realizem a detecção e classificação de imagens de produtos com defeitos na máquina do *Flow Pack*, que fábrica lenços umedecidos.

#### 2.1.1 Lenços umedecidos

A matéria-prima utilizada para a fabricação de lenços umedecidos são tecidos fibrosos de alta qualidade e dermatologicamente testados. O hábito de uso dos lenços umedecidos é complementar ao uso de papel-higiênico, os lenços trazem mais praticidade e de certa forma maior higienização. O uso tem se tornado mais comum para mães de recém-nascidos ao longo das últimas décadas. Entretanto, o produto não se limita apenas à este nicho de mercado. "Porém, com a diversidade de opções no mercado, muitas pessoas ficam em dúvida sobre qual a versão ideal para cada caso. É bastante comum que adultos comprem toalhas infantis sem saber que existem opções para cada idade, gênero e objetivo."(COTTONBABY, 2020). Portanto os produtos que

analisaremos neste trabalho, possuem distintas funções, porém todos são produzidos na mesma máquina. Conforme apresentados na figura 1.

**Figura 1 – Produtos utilizados no trabalho.**



a) Wipe Germ Home

b) Demaquilante

c) Higiene íntima

Fonte: Elaborado pelo autor (2022).

### 2.1.2 Máquina *Flow Pack*

A máquina usada para estudo será o *Flow Pack*. Máquina que realiza a fabricação dos produtos que serão inspecionados pelo sistema de visão conforme mostrado na figura 2.

**Figura 2 – Máquina *Flow Pack*.**



Fonte: Elaborado pelo autor (2022).

### 2.1.3 Defeitos analisados

Os defeitos são bem distintos um dos outros facilitando a classificação e detecção. Tais não conformidades são possíveis de encontrarmos na fabricação dos 3 produtos que serão analisados e elas são: falha na tampa adesiva (ou selo) e lenço para fora da embalagem (figura 3).



**Figura 3 – Exemplos de falhas.**

Fonte: Elaborado pelo autor (2022).

As falhas na tampa podem ocorrer por dois motivos. O primeiro é falha ou rompimento do rolo no desbobinador de selos e o segundo acontece por falta de sincronismo do desbobinador. Já o lenço estar fora da embalagem se dá por conta da dessincronização da esteira de onde os lenços saem para a embaladora; ou o fechamento da embalagem está fora do ponto.

## 2.2 Visão Computacional e RNAs

Nas subseções seguintes serão apresentados os conceitos e métodos a respeito dos temas: visão computacional (*Computer Vision* ou CV), redes neurais artificiais (RNAs) respectivamente.

### 2.2.1 Visão Computacional (CV)

A visão computacional (CV) consiste em um conjunto de métodos e técnicas que possibilitam máquinas e equipamentos a analisarem determinados objetos a partir de imagens adquiridas por meio de sensores matriciais (câmeras). O objetivo da *computer vision* é realizar tarefas analisando objetos físicos reais e cenários baseado na sua programação e na análise de imagens. (SHAPIRO; STOCKMAN, 2000).

Portanto, aplicando estas técnicas de visão computacional será possível que o *Flow Pack* possua uma SIQ e, por consequência, obtendo melhores resultados na produção. Sendo um sistema automático, proporcionará uma maior velocidade de processamento de imagens, consequentemente, gerando um maior volume de produção e menos não conformidades. (PROBST; MELO; ROLOFF, 2010).

As primeiras experiências em visão computacional aconteceram nos anos 1950, com o uso de algumas das primeiras redes neurais para detectar os limites de um objeto e para classificar objetos simples em categorias como círculos e quadrados. Nos anos 1970, o primeiro uso comercial de visão computacional interpretou textos manuscritos e digitados usando reconhecimento ótico de caracteres. Esse avanço tinha como objetivo interpretar textos escritos para deficientes visuais.

Com o amadurecimento da internet nos anos 1990, grandes volumes de imagens foram disponibilizados online para análises, e o desenvolvimento de programas de reconhecimento facial explodiu.

Esses crescentes conjuntos de dados ajudaram a possibilitar que máquinas identifiquem pessoas específicas em fotos e vídeos. (SAS, 2020).

"A aplicação de IA para solucionar grandes problemas da área de ciência da computação tem sido possível a partir da produção de *hardwares* mais potentes"(ALVES, 2020). Porém sabe-se que esses *hardwares* só vieram ser capazes de solucionar essas problemáticas recentemente. Nos anos 50 e 70 o acesso à esta tecnologia era limitado a pesquisadores em grandes empresas e universidades.

A visão computacional com a utilização do *Deep Learning* possui uma grande diversidade de aplicações, sendo algumas delas as seguintes:

- a) Reconhecimento Facial: é um tipo avançado de detecção de objetos que não apenas reconhece as faces de pessoas, como pode identificar o indivíduo;
- b) Detecção de objetos: é a aplicação que realiza a detecção de um ou mais objetos;
- c) Segmentação de imagens: é a separação de varias regiões ou partes em uma imagem que serão analisadas separadamente;
- d) Classificação de objetos: é um tipo de detecção de padrão que identifica padrões em imagens para auxiliar nas suas classificações.

### 2.2.2 Redes Neurais Artificiais (RNAs)

Com algumas definições sobre o que é visão computacional, podemos seguir adiante nos estudos sobre as IAs e seus subconjuntos ou subáreas.

A IA pode ser dividida em subáreas de acordo com suas aplicações. A Association for the Advancement of Artificial Intelligence (AAAI) é considerada uma associação de referência e na sua última chamada de trabalhos dividiu as aplicações em nove subáreas: Pesquisa; Machine Learning, Data Mining e Big Data; Planejamento Automatizado; Representação de Conhecimento; Raciocínio (Probabilístico ou não); Processamento de Linguagem Natural; Robótica; Sistema de Agente e Multi-Agente e Aplicações. (ALVES, 2020).

Como o aprendizado profundo e o aprendizado de máquina tendem a ser usados de forma permutáveis, vale a pena observar as particularidades entre os dois. Aprendizado de máquina, aprendizado profundo e redes neurais são todos subconjuntos da IA. No entanto, o *deep learning* é na verdade um subconjunto do aprendizado de máquina, e as redes neurais são uma subclasse do aprendizado profundo. (IBM, 2020)

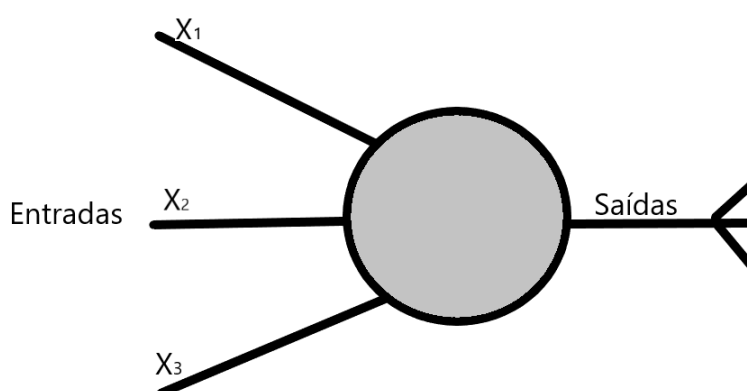
As RNAs são algoritmos que constroem modelos matemáticos que realizam interpretações de dados, portanto quando definido que as redes neurais artificiais fazem

parte do campo do aprendizado de máquinas, determina-se que estes parâmetros se auto-ajustarão para que de fato haja o aprendizado.

Fazendo uma simplificação do processo de uma rede neural biológica, um neurônio se conecta aos outros utilizando uma estrutura chamada de sinapses, recebe impulsos através de uma das suas extremidades conhecida como dendritos, processa estas informações em seu núcleo celular a partir de reações químicas e elétricas, e transmite novos impulsos resultantes pelo axônio para outros neurônios. (BRIGNOLI, 2021)

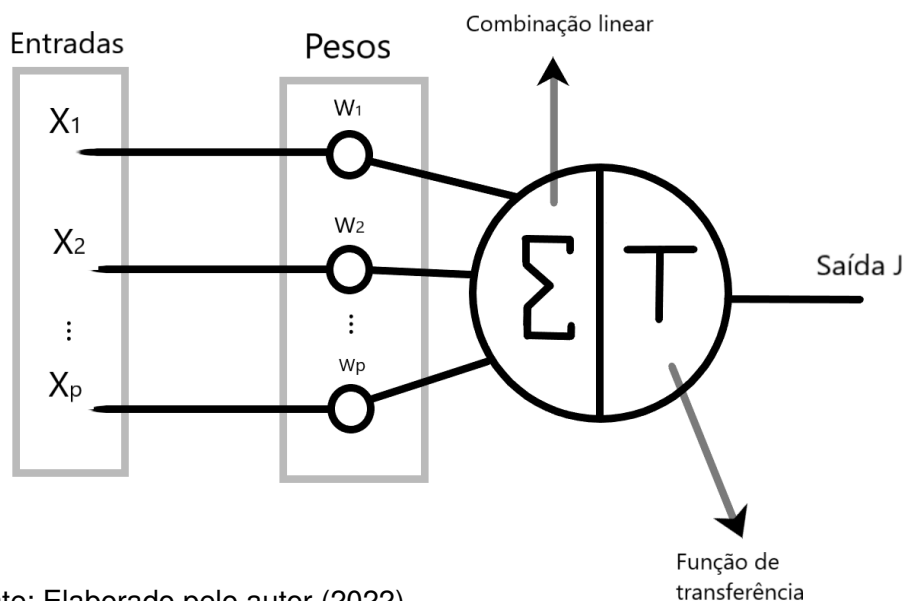
Uma RNA é composta por neurônios artificiais que se organizam em camadas, podendo também serem chamados de *perceptrons*. Com isso, elas buscam simular o funcionamento de um cérebro humano, sendo assim, estas camadas de *perceptrons* se conectam entre si e com as demais camadas seguintes para que assim haja a multiplicação das entradas por pesos e assim resultando em uma classificação binária. (HAYKIN, 2007).

**Figura 4 – Perceptron simplificado**



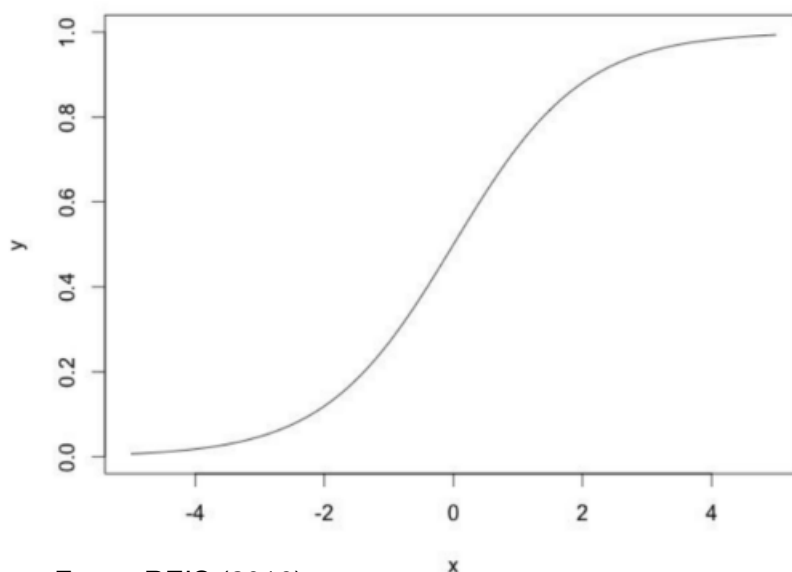
Fonte: Elaborado pelo autor (2022).

O primeiro modelo de um *perceptron* conhecido na literatura é o modelo de McCulloch e Pitts datado no ano de 1943. As entradas ( $X_1, X_2, \dots, X_p$ ) são multiplicadas por uma matriz de pesos ( $W_1, W_2, \dots, W_p$ ), e então é realizada uma combinação linear desses valores calculados (conforme apresentado na figura 5).

**Figura 5 – Estrutura de um Perceptron**

Fonte: Elaborado pelo autor (2022).

Após feita a combinação linear é aplicado uma função de transferência. Podemos comparar essa função de transferência ou função de ativação ao processo eletro-químico que um neurônio biológico realiza. A escolha de qual tipo de função de ativação é a mais adequada para determinada aplicação, difere do tipo de dado que será utilizado. Na figura 6 é apresentada uma das funções de ativação mais utilizadas.

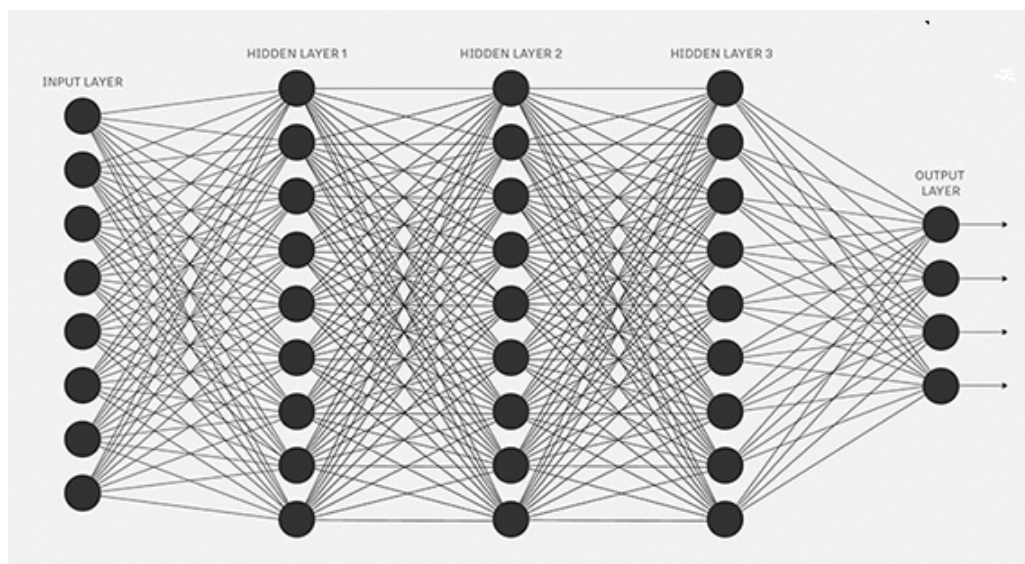
**Figura 6 – Função Sigmoide**

Fonte: REIS (2016).

Seguindo na área das RNAs, é importante lembrar da arquitetura que é mais utilizada para o reconhecimento de padrões, o *Multilayer Perceptron* ou MLP. Tal

arquitetura possui uma estrutura bem definida: uma camada de entrada (*Input Layer*), uma ou mais camadas ocultas (*Hidden Layers*) e uma camada de saída (*Output Layer*), como apresentado na figura 7.

**Figura 7 – Multilayer Perceptron**



Fonte: DATA SCIENCE ACADEMY

Antes de aprofundar os conhecimentos em CNNs é preciso definir como os computadores e as redes neurais interpretam as imagens. Uma imagem é formada por pixels, eles são a menor parte dela, a junção de vários deles de forma matricial criando a imagem completa. Um pixel pode ter uma variação de intensidade entre 0 e 255, isso sendo em uma dimensão quando a imagem é em tons de cinza. Já para imagens coloridas ou RGB (vermelho, verde e azul) continuamos com a mesma variação de intensidade, porém temos três dimensões de matrizes de pixels que agrupadas formam uma imagem colorida (BRIGNOLI, 2021).

Então, nas seções seguintes faremos um estudo mais aprofundado sobre as CNNs.

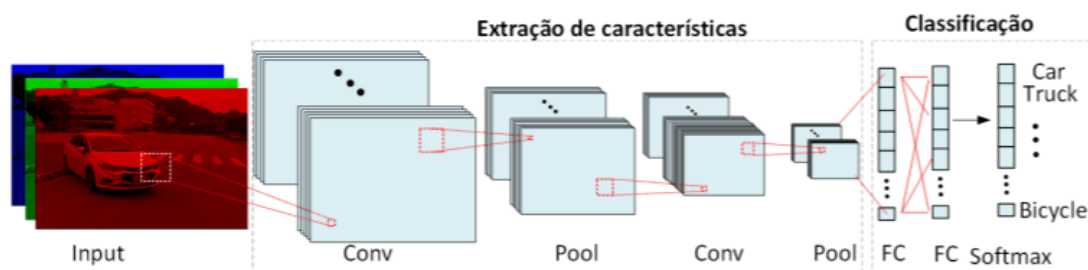
### 2.3 Descrição geral das CNNs

"Uma rede neural convolucional ou CNN (*Convolutional Neural network* ou *ConvNet*) é um tipo especial de MLP projetada para reconhecer padrões visuais diretamente dos pixels da imagem com pré-processamento mínimo." (LEMOS, 2021, p.42)

As CNNs como mencionado nas subseções anteriores são uma subárea do *deep learning*. Tal técnica foi criada para interpretar os pixels de imagens, atribuir pesos a diversas características com a finalidade de diferenciar esses aspectos. As imagens utilizadas nela precisam de um pré-processamento mínimo em comparação com as

RNAs clássicas previamente utilizadas em visão computacional, facilitando muito o seu treinamento. (HOWARD; GUGGER, 2020)

**Figura 8 – Representação básica de uma CNN**



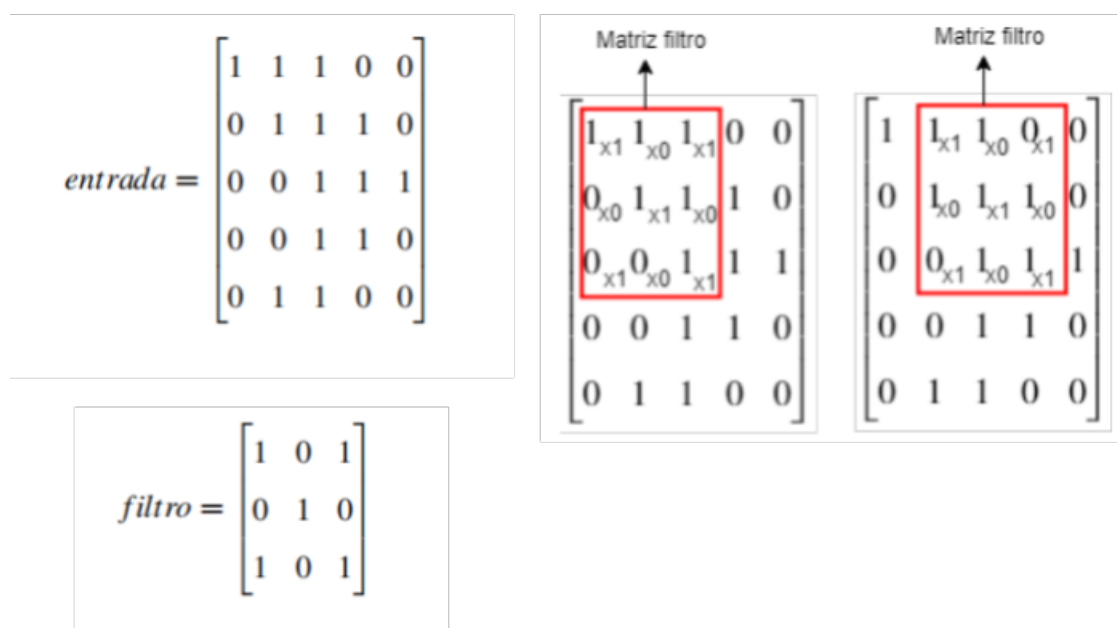
Fonte: Majin (2021).

Como mostrado na figura 8 as redes neurais convolucionais possuem geralmente 3 camadas ou blocos de construção que são as seguintes: camada de convolução, camada de *pooling* e camada totalmente conectada. "As duas primeiras, camadas de convolução e de agrupamento, executam extração de recursos, enquanto a terceira, uma camada totalmente conectada, mapeia os recursos extraídos na saída final, como classificação." (YAMASHITA *et al.*, 2018, p.612).

### 2.3.1 Camada de Convolução

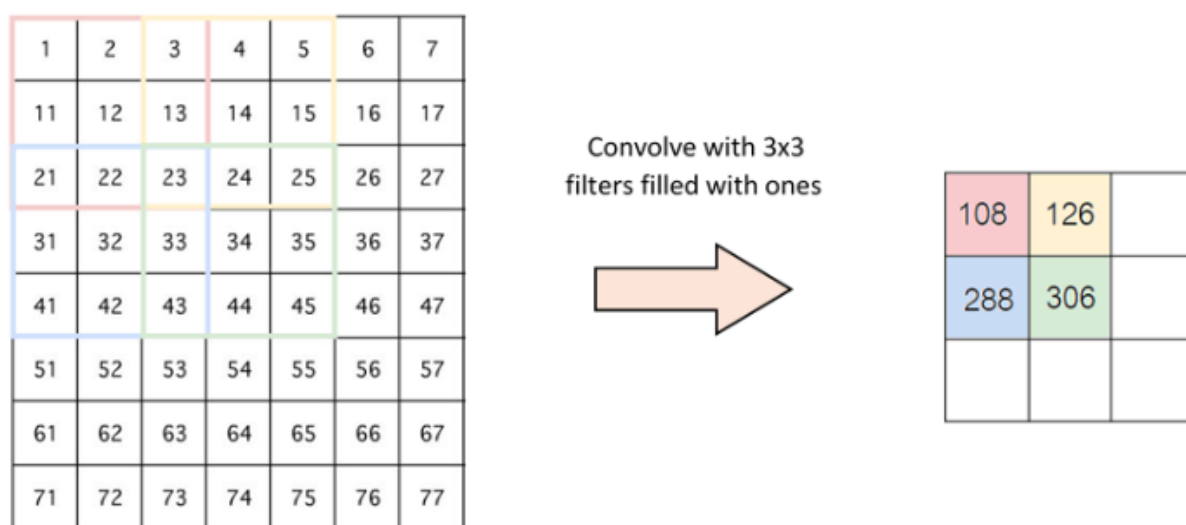
Na camada convolucional acontecem as convoluções, responsáveis pela aplicação de filtros ou *kernels* que fazem as extrações de características das imagens, tendo como resultado os mapas de características. Estas convoluções são filtros lineares comumente utilizados em processamento de imagens. Os filtros podem possuir diversos tamanhos e valores distintos que serão multiplicados pelos pixels de cada imagem. Para cada filtro aplicado, podemos obter diferentes resultados de detecções como por exemplo: detecção de bordas, cantos, círculos, quadrados, etc (MAJIN, 2021).

Um kernel é uma matriz de valores que serão multiplicados pelo volume de entrada (matriz de valores dos pixels) de uma imagem. O funcionamento de uma convolução está ilustrada na figura 9, onde podemos ver que o kernel realizando um deslocamento por todo o volume de entrada para que obtenha como resultado o mapa de características de uma dada imagem (BRIGNOLI, 2021).

**Figura 9 – Processo de convolução para os dois primeiros passos.**

Fonte: SILVA (2018).

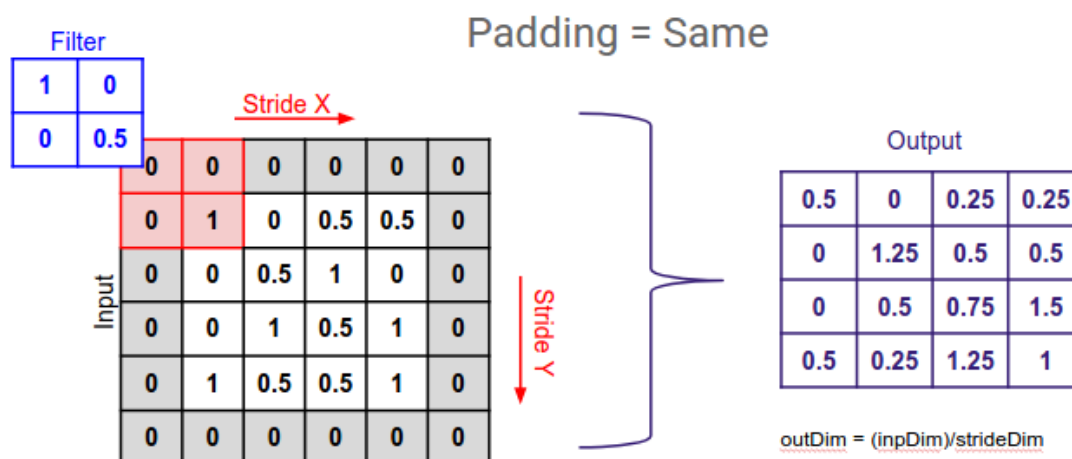
"Três parâmetros comumente utilizados para controlar o tamanho da saída das camadas convolucionais são: profundidade, stride e padding."(MAJIN, 2021, p.23). O parâmetro de profundidade irá alterar a quantidade de vezes que será utilizado o filtro em cada camada de convolução. Porém o *stride* faz o controle de movimentação deste filtro dentro do volume de entrada, de quantos em quantos pixels o kernel será deslocado dentro dessa matriz conforme na figura 10.

**Figura 10 – Exemplo de *stride*.**

Fonte: PRABHU (2019).

Por fim o *padding* é uma técnica que auxilia no problema da utilização de filtros e ou *stride* muito grande. Ela consiste na criação de um contorno na imagem, permitindo melhor controle no tamanho dos mapas de características obtido pela camada e possibilitando adquirir mais informações nas fronteiras da imagem (figura 11)

Figura 11 – Exemplo de padding.



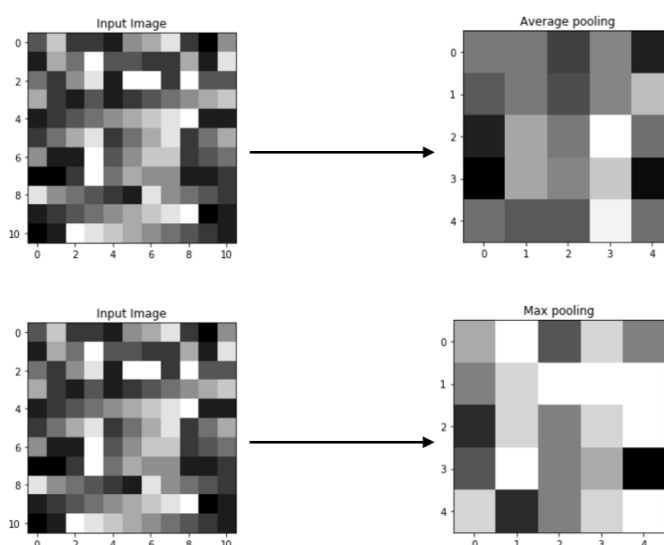
Fonte: PERERA (2019).

Ao final de todas operações lineares, como por exemplo a convolução, são associadas a uma função de ativação não linear. Alguns exemplos são: função ReLu, função sigmoid, função softmax, etc (LEMOS, 2021).

### 2.3.2 Camada de Pooling

Já por sua vez, a camada de pooling ou camada de agrupamento são normalmente utilizadas após uma camada de convolução, pois sua função é a redução progressiva do volume de saída da convolução. Para isso existem dois agrupamentos bem conhecidos, *max pooling* e *average pooling* ou agrupamento máximo e agrupamento médio respectivamente apresentados na figura 12. "Esta redução de parâmetros permite reduzir a carga computacional, limita o risco de sobreajuste (overfitting) e fornece alguma invariância a rotações e translações."(MAJIN, 2021, p.26).



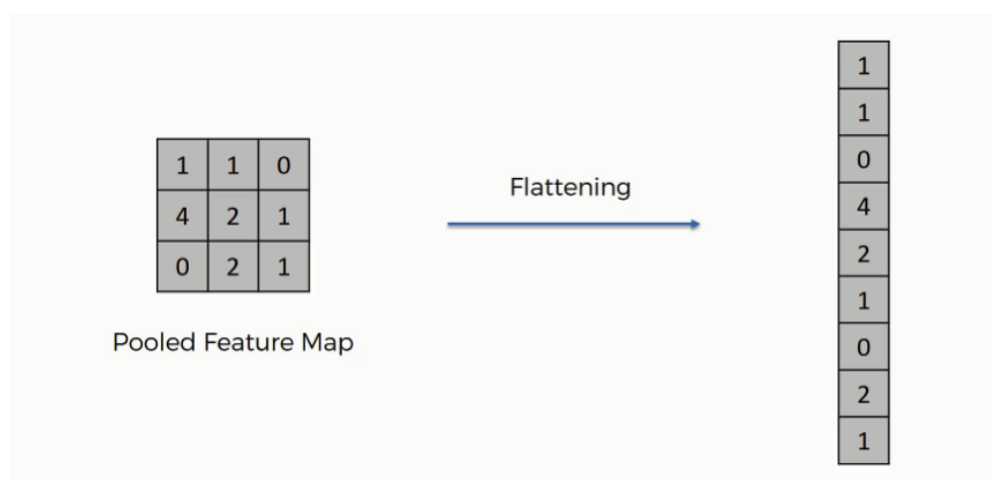
**Figura 12 – Max pooling e Average pooling.**

Fonte: SHARMA (2020).

### 2.3.3 Camada totalmente conectada

E finalmente temos a camada totalmente conectada (CTC). Em toda e qualquer CNN estão presentes as camadas totalmente conectadas, logo após as camadas de convolução ou agrupamento é necessário a utilização desta camada para que haja a classificação dos mapas de características resultantes. Para que a CTC consiga classificar os dados, é necessário que seja passada uma função de ativação nela, sendo duas delas a *Softmax* (para problemas de múltiplas classes) e a *Sigmoid* (problemas binários) (PARKHI; VEDALDI; ZISSERMAN, 2015).

As CTCs têm como entrada um vetor característica unidimensional que sofrerá um achatamento por conta de uma operação chamada *flattening*, então depois o vetor resultante da operação terá uma aplicação de uma função de ativação como por exemplo a *softmax*, sendo assim, o resultado final da CNN será um vetor unidimensional que apresenta as probabilidades de classificação.

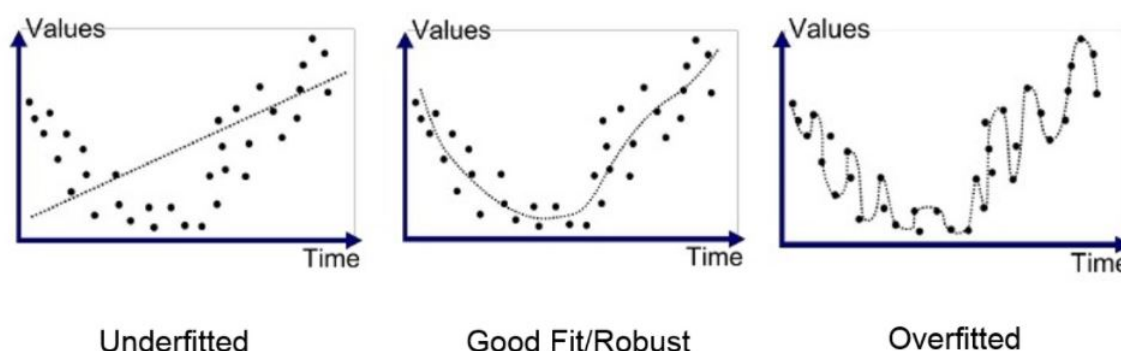
**Figura 13 – Operação de Flattening.**

Fonte: Super Data Science Team (2018).

## 2.4 Redes Neurais Residuais (ResNet)

Nessa seção será apresentado as Redes Neurais Residuais ou ResNet, um modelo de rede neural convolucional bem conhecido e muito usado. Sua utilidade se dá devido a problemas relacionados a profundidade das redes, tais como: *underfitting* (pouco treinamento), *overfitting* (treinamento em excesso) ou de degradação do gradiente de desaparecimento.

Um modelo com *underfitting* ou sub-ajustado, é um modelo que não conseguiu aprender suficiente sobre os dados. O *underfitting* leva à um erro elevado tanto nos dados de treino quando nos dados de teste. O *overfitting* ou sobre-ajustado é o oposto. É quando o modelo aprende demais sobre os dados. Neste caso é como se o modelo tivesse apenas decorado os dados de treino e não fosse capaz de generalizar para outros dados nunca vistos antes. Quando isso acontece, os dados de treino apresentam resultados excelentes, enquanto que a performance do modelo cai drasticamente com os dados de teste. Na figura 14 pode-se observar três modelos: o adequado, com *underfitting* e com *overfitting*.

**Figura 14 – Exemplos de curvas sub-ajustada, adequada e sobre-ajustada, respectivamente.**

Fonte: BRANCO (2020).

Portanto, a ResNet possibilita o treinamento de uma CNN de grande profundidade, podendo ser de 18 até 152 camadas. O que diferencia a ResNet dos outros modelos é que a cada duas camadas de pesos são realizadas realimentações do valor de entrada (conforme apresentado na figura 15) fazendo uma regularização (BRIGNOLI, 2021).

**Figura 15 – Bloco residual: realimentação da entrada X.**

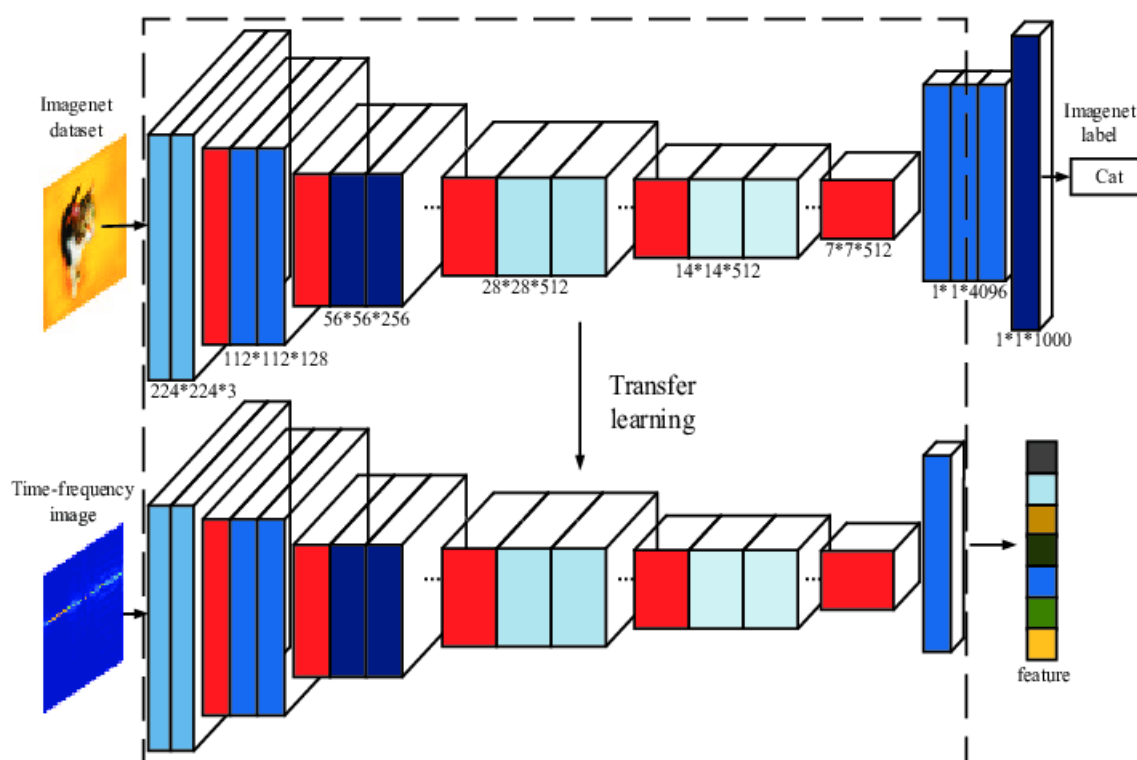


Fonte: HE *et al.* (2015).

#### 2.4.1 Transfer Learning

O projeto tem como finalidade utilizar de arquiteturas já modeladas e validadas evitando assim a criação de uma arquitetura do "zero". Para que isso aconteça será utilizado de uma técnica comumente utilizada no mundo do DL, o *transfer learning* (figura 16). Este método consiste em reutilizar modelos já treinados anteriormente, com o propósito de ser necessário apenas alguns ajustes e refinamentos no novo modelo, para que assim ele permita obter novas classificações de objetos (MAJIN, 2021).

Figura 16 – Bloco residual: realimentação da entrada X.

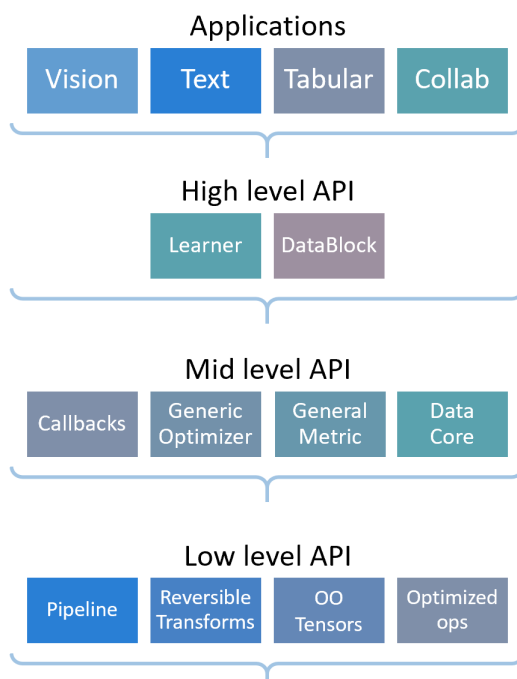


Fonte: DO S, SONG KD, CHUNG JW (2020).

#### 2.4.2 Biblioteca Fastai

Para a aplicação das técnicas de aprendizado profundo de máquina, é necessário de uma biblioteca de componentes de *deep learning* para o auxílio. Neste projeto será utilizado a *Fastai*, uma biblioteca de DL que tem como linguagem de programação *Python* e sendo considerada a de mais alto nível dentre as demais bibliotecas.

Na figura 17 pode-se observar a organização em níveis da *fastai*. Como mencionado será utilizada a sua API de alto nível, nela observa-se a classe *Learner* e *DataBlock*.

**Figura 17 – Níveis de organização da fastai.**

Fonte: HOWARD; GUGGER (2020)

O *learner* tem como função realizar o treinamento da rede, os principais parâmetros que devem ser passados para essa classe é os dados pré-processados pelo *datablock*, o modelo que será utilizado e as métricas que se desejam como resultado do treinamento.

Já o *datablock* tem como função realizar todas as tarefas relacionadas à entrada de dados, como a coleta, identificação do problema, separação de variáveis entre dependentes e independentes, aumento de dados, separação do bando de dados em uma parte para o treinamento e outra para a validação da rede, entre outras tarefas. Os principais parâmetros são: definir o caminho de onde virão as imagens para o treinamento, e como estão dispostas em relação às classes e definir qual será a porcentagem do banco de dados será reservada para o treinamento, assim como para a validação (BRIGNOLI, 2021).

#### 2.4.3 Aumento de dados (Data Augmentation)

O aumento de dados são técnicas de transformações de tamanho, posição e angulação utilizadas nos dados para que não ocorra o *underfitting* no treinamento das CNNs. Na *fastai* como mencionado anteriormente o *datablock* é a classe responsável pelas entradas de dados da nossa rede, alguns recursos presente nesta classe que é bastante utilizado é o *data augmentation*.

Estes recursos são diversos, porém neste projeto não é válido realizar

várias transformações sendo que a aplicação real não requer nada além de Rotações randômicas.

#### 2.4.4 Métricas

Só é possível identificar se a CNN está tendo bons resultados, devido às métricas. Elas permitem que os desenvolvedores tenham informações quantitativas, de como seu treinamento de rede está se saindo. As métricas utilizadas para classificações binárias possuem 4 possíveis resultados (conforme apresentado na tabela 1):

**Tabela 1 – Categorização dos resultados de predição para um modelo de aprendizagem de máquina.**

Classe	Valor real do exemplo	Valor retornado pelo sistema
Verdadeiro Positivo	Positivo	Positivo
Verdadeiro Negativo	Negativo	Negativo
Falso Positivo	Negativo	Positivo
Falso Negativo	Positivo	Negativo

Fonte: CUNHA, p. 53 (2017)

- a) Verdadeiro positivo (Vp): é a quantidade de dados que a rede classificou como verdadeiros e de fato a classificação esta correta;
- b) Verdadeiro negativo (Vn): número de classificações negativas que são realmente negativas;
- c) Falso positivo (Fp): é quando o sistema classifica como verdadeiro porém o certo seria falso;
- d) Falso negativo (Fn): quando o classificador interpretou o resultado como falso e o certo seria verdadeiro.

##### 2.4.4.1 Acurácia

A acurácia é uma métrica que indica o desempenho do modelo de maneira mais intuitiva, apresentando a quantidade de classificações corretas a rede realizou, podemos observar a sua equação correspondente em (1).

$$Acuracia = \frac{Vp}{Totaldeitens} \quad (1)$$

##### 2.4.4.2 Precision (Precisão)

"Outra métrica utilizada é a precisão. Esta pode ser aplicada a cada classe individualmente e mede, dos resultados positivos, quantos efetivamente eram previstos como positivos."(BRIGNOLI, 2021). Representada em (2).

$$Precision = \frac{Vp}{Vp + Fp} \quad (2)$$

#### 2.4.4.3 Recall

O recall é a métrica apresenta o calculo de quantos positivos reais o modelo previu, indicando-o como verdadeiro positivo. Quanto mais próximo de 1 melhor é o resultado apresentado. (LEMOS, 2021) (3)

$$Recall = \frac{Vp}{Vp + Fn} \quad (3)$$

#### 2.4.4.4 F1-Score

O F1-Score é uma média harmônica entre a precisão e a o recall, representada na equação (4)

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

#### 2.4.4.5 Matriz de confusão

E por fim, a matriz de confusão é uma matriz de dimensão 2x2, podendo ter dimensões maiores de acordo com a quantidade de classes que se deseja prever. Seu funcionamento tem como objetivo realizar os cálculos da quantidade de verdadeiros positivos (Vp), verdadeiros negativos (Vn), falsos positivos (Fp) e falsos negativos (Fn). O exemplo de uma matriz de confusão é apresentado na figura 18, este exemplo consiste na utilização de 100 imagens divididas entre 50 imagens de gatos e 50 imagens que não são gatos. Após o treinamento da rede é retirado a matriz de confusão, a partir dela podemos visualizar de forma mais clara os resultados das predições que a rede realizou.

**Figura 18 – Exemplo de matriz de confusão.**

		Classe esperada	
		Gato	Não é gato
Classe prevista	Gato	25 Verdadeiro Positivo	10 Falso Positivo
	Não é gato	25 Falso Negativo	40 Verdadeiro Negativo

Fonte: VASCONCELLOS (2018).

## 2.5 Trabalhos Correlatos

Devido à ampliação dos conhecimentos relacionados à visão computacional e redes neurais convolucionais, existem diversas aplicações e nas mais distintas áreas. O primeiro exemplo à ser citado é o trabalho de conclusão de curso desenvolvido por Ramon Brignoli (2021). Sua pesquisa utilizou de um sistema de classificação de imagens através de CNNs, de ambientes residenciais em 6 classes distintas, tais como: quarto, cozinha, sala, fachada, banheiro e área externa.

BRIGNOLI analisou a utilização de três modelos pré-treinados da arquitetura de ResNet, a ResNet18, ResNet34 e ResNet50. Como conclusão de suas análises, foi evidenciado que a arquitetura ResNet50 respondeu melhor para as classificações desejadas. Sua acurácia média chegou à 96% e seu tempo médio de inferência para cada imagem foi de 80ms.

O segundo exemplo é a dissertação de mestrado de Yessica Lemos (2021), cujo tema é "Inspeção automática de defeitos em ovos comerciais usando visão computacional". Seu trabalho realiza um comparativo de uma abordagem utilizando técnicas de clássicas de processamento de imagens e duas com abordagens de aprendizado profundo. As três tiveram resultados superiores à 81% de acurácia e tempo médio de processamento inferior a 1,35 segundos.

Por fim, não foi encontrado trabalho relacionados à detecção de defeitos em embalagens de lenços umedecidos. Este trabalho se propõe a contribuir com o estudo das CNNs aplicadas a processos industriais.



### 3 METODOLOGIA

Neste capítulo será apresentado de forma detalhada o desenvolvimento de um sistema de classificação de embalagens, utilizando técnicas de aprendizado de máquina em CNNs.

#### 3.1 Discussão sobre os requisitos do sistema

O primeiro passo na metodologia é determinar os requisitos funcionais, dos quais o sistema deverá atender. Partindo disso, é possível encontrar soluções que atendam os pontos levantados. Estes requisitos foram concebidos com o auxílio da equipe de projetos da empresa estudada.

##### 3.1.1 Requisitos funcionais e não funcionais

Os requisitos funcionais que orientaram o desenvolvimento do sistema de classificação foram:

- O sistema deve ser capaz de classificar cada uma das embalagens como pertencente às seguintes classes: Sem defeito, Defeito na tampa e Defeito de lenço para fora;
- O sistema deve permitir a inspeção de diferentes tipos de produtos.

Além disso, o sistema deve atender ao seguinte requisito não funcional:

- O sistema deve realizar a inspeção em, no máximo, 2 segundos.

O requisito do sistema ser capaz de classificar as três classes citadas é necessário para que seja possível a identificação de produtos defeituosos produzidos. Na figura 19 é possível observar as categorias que o sistema deve classificar.

**Figura 19 – Categorias que o sistema deve classificar**

Fonte: Elaborado pelo autor (2022).

O segundo requisito funcional trata-se do sistema ser capaz de inspecionar produtos diferentes. A justificativa para tal objetivo é devido ao fato de que a máquina usada para aplicação realiza a produção de produtos distintos.

Já o terceiro e último requisito está diretamente relacionado com o tempo de inspeção do sistema. A máquina atinge uma produção de 30 produtos a cada 60 segundos, ou seja, a cada 2 segundos o sistema deve ser capaz de predizer um único produto.

### 3.2 Descrição da solução

Com os requisitos já definidos, é possível dar início a toda a descrição de como foi realizado o projeto. Sendo assim, as técnicas escolhidas para atingir os requisitos foram de aprendizado profundo com foco nas redes neurais convolucionais ou CNNs, visto que esta abordagem tem se mostrado muito eficiente para a classificação de imagens. Para fazer a implementação destas CNNs é utilizada a biblioteca Fastai, já mencionada na subseção 2.4.2. Foi tomada a decisão de utilizar uma rede neural convolucional para cada tipo de produto, tendo em vista que os mesmos possuem características distintas. Todo o treinamento da rede foi realizado no Google Colab, pois ele permite a utilização de uma máquina virtual com GPU Tesla K80, com 24GB de memória GDDR5, capaz de realizar processamentos de uma forma muito mais ágil. O Google Drive fez o papel de armazenar e organizar a base de dados, o Drive trás mais facilidade para essa organização, pois é facilmente integrado ao algoritmo realizado no Colab.

Portanto, após todos os requisitos definidos, foi traçado um planejamento das etapas para o desenvolvimento das redes, sendo elas:

- Preparação e obtenção das bases de dados;
- Treinamentos dos modelos.

Nas subseções a seguir serão apresentadas as principais etapas de desenvolvimento das redes neurais convolucionais utilizadas neste presente trabalho.

### 3.2.1 Preparação e obtenção das bases de dados

A obtenção das imagens foi realizada de forma não integrada à linha de produção em função de restrições quanto ao acesso aos equipamentos em produção na indústria parceira. Assim, foram preparadas embalagens já produzidas de forma a apresentarem características associadas às classes a serem avaliadas (figura 3). Todas as imagens foram capturadas por uma câmera semi profissional Fujifilm Finepix S4500 com 14 mega pixels (figura 20).

**Figura 20 – Câmera utilizada para a obtenção das imagens**



Fonte: Elaborado pelo autor (2022).

As imagens foram preparadas para três produtos distintos com a finalidade de atender ao requisito do sistema inspecionar diferentes produtos e diferentes classes. Na figura 21 é possível observar os exemplos de imagens capturadas.

**Figura 21 – Imagens capturadas.**

Fonte: Elaborado pelo autor (2022).

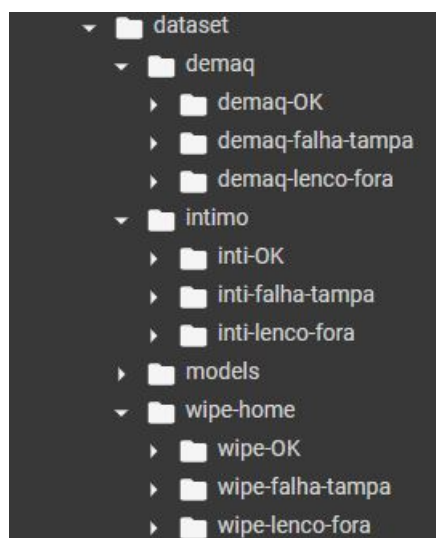
A partir dessa aquisição foi possível armazenar a base de dados no Drive para que posteriormente realizar o treinamento dos modelos. Essa organização já foi feita pensando em como os modelos irão receber essas informações. Na tabela 2 é apresentado a quantidade de imagens para cada classe.

**Tabela 2 – Quantidade de imagens por classe e produto**

Produto	Classes	Qtd. Imagens
Demaquilante	sem defeito	30
	defeito na tampa	69
	defeito lenço para fora	68
Higiene íntima	sem defeito	20
	defeito na tampa	34
	defeito lenço para fora	37
Wipe Germ Home	sem defeito	56
	defeito na tampa	62
	defeito lenço para fora	53

Fonte: Elaborado pelo autor (2022).

Para isso é necessário uma organização das pastas que contenham as imagens de cada classe. Na figura 22 podemos observar essa disposição.

**Figura 22 – Organização das pastas das imagens adquiridas**

Fonte: Elaborado pelo autor (2022).

### 3.2.2 Treinamentos dos modelos

Para os treinamentos dos modelos, como mencionado anteriormente foi utilizada a biblioteca *Fastai* e a plataforma *Google Collaboratory*. A *Fastai* possui uma função chamada *cnn\_learner*, que realiza o treinamento dos modelos desejados. Esta função requer que sejam passados alguns parâmetros para que haja o treinamento, tais parâmetros são: *DataBlock*, arquitetura que deseja-se utilizar e as métricas que serão utilizadas.

O *DataBlock* é o parâmetro que irá realizar todas tarefas relacionadas a base de dados já mencionadas anteriormente na subseção 2.4.2. No código a seguir observa-se melhor toda a estrutura do *DataBlock* de uma das aplicações.

**Código 1 – Código da estrutura do DataBlock (Íntimo)**

```
1 inti = DataBlock(  
2     blocks=(ImageBlock, CategoryBlock),  
3     get_items = get_image_files,  
4     splitter= RandomSplitter(valid_pct=0.3, seed=42),  
5     get_y= parent_label,  
6     item_tfms=Resize(128),  
7     batch_tfms=aug_transforms(max_rotate=30, pad_mode=PadMode.Zeros))  
8 dls = inti.dataloaders(path_inti)
```

Fonte: Elaborado pelo autor (2022).

Neste *DataBlock* apresentado acima, é possível observar que são criados dois blocos principais, um contendo as imagens e outro a categoria ou classes das imagens. O argumento *get\_items* é usado para obter as imagens do diretório desejado, armazenando elas nas dependências do *ImageBlock*. Já o *splitter* é a definição de

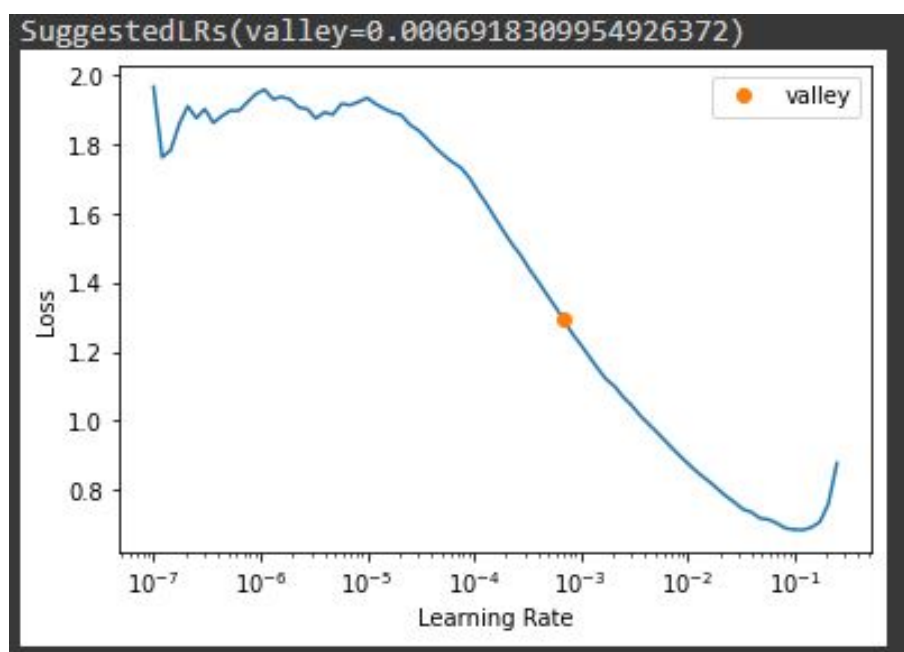
como será separado os dados de entrada. Foi aplicado um *splitter* randômico em toda base de dados, sendo 30% designado para validação e os 70% restante será para treinamento. O "get\_y" é um argumento que indica como será identificada as classes de cada imagem, armazenando essas informações no bloco *CategoryBlock*. E por fim são realizadas as transformações para aumentar a base de imagens, tais como: redimensionamento (*resize*) de 128x128 pixels, rotações de 30 graus e a aplicação de *padding* de "zeros". Após a criação do *DataBlock*, ele é atribuído o caminho das pastas das imagens que serão utilizadas no treinamento.

A arquitetura do modelo é o segundo principal parâmetro necessário para o treinamento. Neste trabalho foi escolhida a arquitetura pré-treinada ResNet101. Previamente foram realizados treinamentos utilizando outras arquiteturas ResNet com menos camadas, porém a ResNet 101 obteve resultados satisfatórios. O terceiro e último parâmetro é a definição de quais métricas serão apresentadas a cada época (*epoch*) de treinamento. Um época é quando todas as imagens do *DataBlock* são processadas pelo modelo ao menos uma vez. As métricas que foram utilizadas neste trabalho são: acurácia, precisão, *recall* e *f1-score*. Já a matriz de confusão será apresentada e analisada no próximo capítulo.

Com todos os parâmetros já esclarecidos temos a possibilidade de aplicação do *transfer learning*, cujo método já foi definido na subseção 2.4.1. Uma função dentro do TL conhecida como *fine tuning* ou ajuste fino foi utilizada para ajustar os resultados dos treinamentos. O funcionamento desta técnica consiste em atualizar os pesos dos modelos já pré-treinados com mais épocas de treinamento, obtendo as classes desejadas.

Para isso, definimos de forma experimental a quantidade de épocas usadas para este novo treinamento a partir dos resultados apresentados pela rede. As três CNNs tiveram valores de épocas diferentes uns dos outros e isso será apresentado no capítulo de resultados. Porém não basta escolher somente por quanto tempo iremos realizar o treinamento, é necessário que seja passado para o *learner* qual será a taxa de aprendizado que ele utilizará para o treinamento. Para descobrirmos esta taxa, a *Fastai* traz em seu conjunto de funções uma que encontra este valor. A função é a *lr\_find*, ela aplica diferentes taxas de aprendizado (*learning rate*) ao modelo e observa a perda (*loss*) em comparação com cada taxa, na figura 23 é apresentado o gráfico que é obtido como resultado, aplicando a função no *learner* do produto *Wipe Germ Home*. A função foi aplicada em todos os 3 modelos.

Figura 23 – Perda x Taxa de aprendizado (Wipe)



Fonte: Elaborado pelo autor (2022).

Com estes dois parâmetros já definidos, podemos usar de fato o *fine tuning* para realizar o treinamento da rede. Após o treinamento feito e as métricas devidamente analisadas, foi utilizada a função *export* existente na biblioteca da *Fastai*, para a criação dos arquivos dos classificadores. No próximo capítulo será apresentado as análises e resultados do presente trabalho.

## 4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo são apresentados os resultados dos treinamentos dos modelos separadamente, podendo assim elucidar melhor os desempenhos de cada rede para cada produto, individualmente.

### 4.1 Resultados de classificação

Com as três CNNs devidamente treinadas, é possível partir para a análise dos resultados obtidos, utilizando as métricas comumente empregadas. Tais métricas já foram apresentadas na subseção 2.4.4. Na tabela 3 pode-se observar os resultados de rodadas de treinamento (épocas), acurácia, precisão, recall e f1-score do treinamento de cada rede.

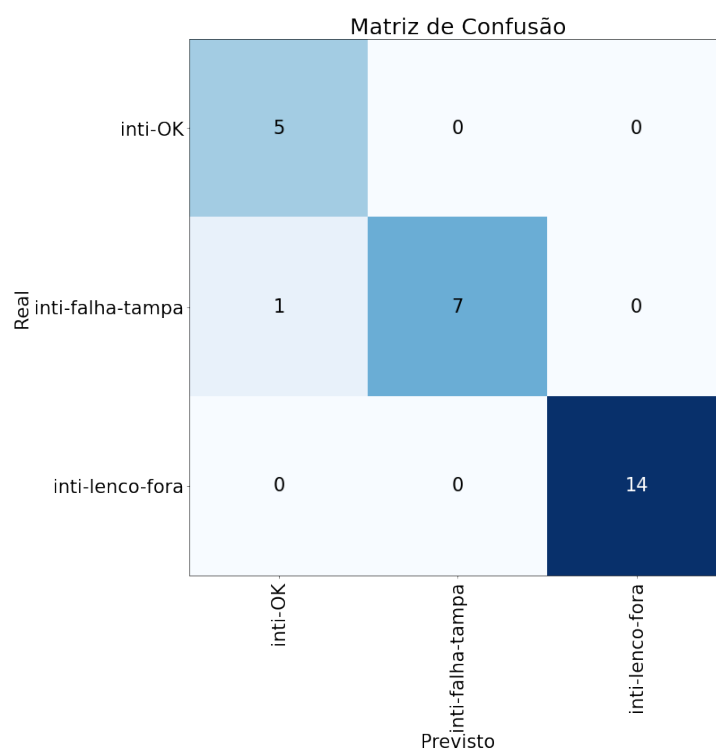
**Tabela 3 – Resultados finais das três CNNs utilizando ResNet101**

<b>CNN</b>	<b>Épocas</b>	<b>Acurácia</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-Score</b>
Demaquilante	35	91,8%	94,0%	88,9%	91,0%
Higiene Íntima	25	96,0%	94,4%	95,8%	94,7%
Wipe Germ Home	25	96,0%	96,3%	96,3%	96,2%

Fonte: Elaborado pelo autor (2022).

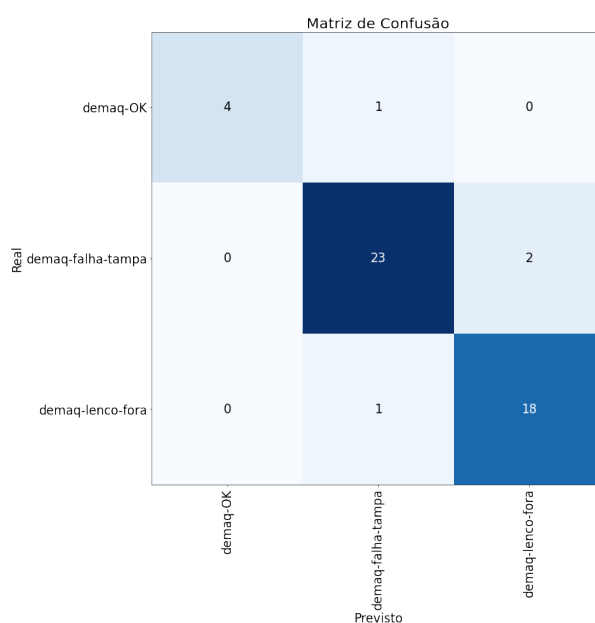
A matriz de confusão desta rede e das demais, foram obtidas com a finalidade de melhorar o entendimento do treinamento das redes, mostrando quantas imagens tiveram predições erradas. Na figura 24 observa-se a matriz de confusão correspondente à CNN Higiene Íntima.



**Figura 24 – Matriz de confusão CNN Higiene Íntima**

Fonte: Elaborado pelo autor (2022).

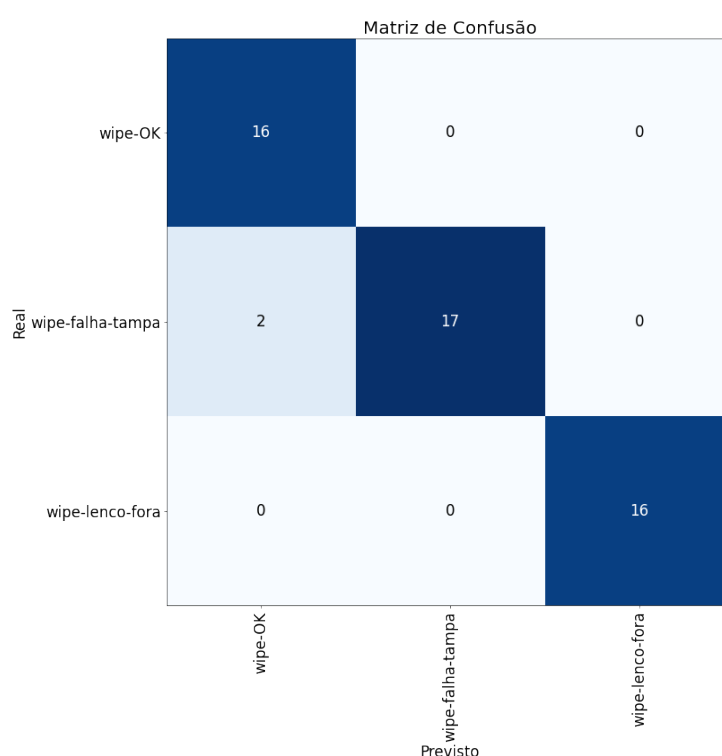
Uma justificativa pela qual uma das imagens foi classificada de forma errada, pode estar ligada pelo baixo contraste entre a tampa do produto e a embalagem em si. Já a CNN do produto demaquilante chegou em bons resultados. Apesar disso, esta rede apresentou resultados inferiores às redes criadas para os outros produtos. Na figura 25 é apresentada a matriz de confusão para a rede associada ao produto demaquilante.

**Figura 25 – Matriz de confusão CNN Demaquilante**

Fonte: Elaborado pelo autor (2022).

E por fim, tem-se os resultados do treinamento realizado na CNN que identificará as classes do produto Wipe Germ Home, apresentados na figura 26. Como o padrão da embalagem é mais simples, da mesma forma que a embalagem do produto Higiene Íntima, foi obtido uma acurácia de 96%. Porém suas demais métricas apresentaram melhores resultados dentre os três produtos estudados. Já na matriz de confusão desta CNN (figura 26), observa-se dois erros de predição que acabam sendo semelhantes aos que encontramos na matriz da CNN do higiene íntima (figura 24). O baixo contraste na tampa e a alternância nos fundos das imagens podem ser uns dos motivos pelo qual estes erros foram gerados.

**Figura 26 – Matriz de confusão CNN Wipe Germ**



Fonte: Elaborado pelo autor (2022).

## 4.2 Custos computacionais

Em seções anteriores foi mencionada a arquitetura escolhida para a execução deste trabalho, assim como a justificativa pela decisão. Com isso, serão apresentados alguns resultados, em questão de desempenho frente aos resultados desta escolha. Na tabela 4 pode-se observar os resultados finais dos treinamentos das três redes neurais. É possível analisar que por mais que a CNN do demaquilante necessitou de mais épocas para o treinamento em comparação com as outras redes, ela apresentou um tempo de treinamento mais curto. O motivo que explica essa diferença, é a simplificação da base de dados, como baixa alteração no fundo das imagens.

**Tabela 4 – Tempo de treinamento gasto por cada CNN**

CNN	Épocas	Taxa de aprendizado	Tempo de treinamento
Demaquilante	35	0.0004	5:07 min.
Higiene Íntima	25	0.0004	12:55 min.
Wipe Germ Home	25	0.0006	20:39 min.

Fonte: Elaborado pelo autor (2022).

Portanto, apesar da utilização a mesma arquitetura para todas as redes. Pode-se perceber que cada rede obteve um comportamento diferente, frente aos resultados apresentados na tabela 3. A CNN Demaquilante apresentou resultados mais baixos do que o esperado, pois a base de dados utilizada para o treinamento, não possui tanta generalização quanto as demais. Mesmo realizando a alteração para mais épocas no treinamento, causaria um *overtraining* em nosso modelo. Isso aplica-se para todas as três redes.

Na seção 3.1 são comentados os requisitos traçados para este trabalho. Com a finalidade da confirmação de que as CNNs estão atendendo ao requisito de processamento de 30 imagens a cada 60s, foi feita a inferência de 10 imagens em cada classificador, sendo assim foi possível obtermos o tempo de inferência de todos. Como conclusão, pode-se observar que os classificadores atendem ao requisito não funcional do sistema. Estes resultados estão apresentados na tabela 5.

**Tabela 5 – Tempo de inferência para uma imagem dos classificadores**

CNN	Qtd.imagens	Tempo de exec.	Tempo p/ uma imagem
Demaquilante	10	3,15s	0,315s
Higiene Íntima	10	2,34s	0,234s
Wipe Germ Home	10	3,21s	0,321s

Fonte: Elaborado pelo autor (2022).

## 5 CONSIDERAÇÕES FINAIS

Neste trabalho, foram desenvolvidos sistemas de classificação de imagens de produtos com ou sem defeitos. Para que isso acontecesse foram utilizadas técnicas de aprendizado de máquina, com foco em *deep learning*. Os resultados obtidos com os classificadores desenvolvidos demonstram que é possível utilizar redes neurais convolucionais em aplicações industriais, principalmente em controle de qualidade de processos produtivos.

Os modelos finais utilizaram-se da arquitetura ResNet101 e apresentaram resultados de acurácia superior à 91%, porém a quantidade de rodadas usadas para o treinamento foram distintos. As redes convolucionais desenvolvidas atenderam aos requisitos anteriormente citados. Todas foram capazes de classificar imagens para as classes: sem defeito, defeito na tampa e defeito lenço para fora. Com o treinamento das CNNs para 3 produtos, obteve-se a confirmação do sistema permitir a inspeção de diferentes tipos de produtos. Ambas as redes neurais apresentaram custos computacionais baixos, sendo assim, atendendo ao requisito não funcional proposto para este trabalho.

### 5.1 Sugestões para trabalhos futuros

Ao decorrer do desenvolvimento, foram encontrados alguns desafios referente à obtenção das imagens que seriam utilizadas para os treinamentos das redes neurais convolucionais. Sabe-se que para a obtenção de bons resultados em treinamento de CNNs está diretamente atrelada à boas imagens. Portanto, como sugestão para trabalhos futuros, a criação de uma boa base de dados para aplicações na manufatura de indústrias traria uma maior facilidade na execução de projetos futuros. Outra temática que seria interessante de abordar, é a aplicação das técnicas de aprendizado profundo à linhas de produção, realizando automações nelas, ou somente levantando estatísticas a respeito dos produtos produzidos.

## REFERÊNCIAS

- ABIHPEC. *Setor de Higiene Pessoal, Perfumaria e Cosméticos fecha 2020 com crescimento de 5,8%*. 2021. Disponível em: <https://abihpec.org.br/comunicado/setor-de-higiene-pessoal-perfumaria-e-cosmeticos-fecha-2020-com-crescimento-de-58/>. Acesso em: 30 jan 2022. 12
- ALVES, P. M. Inteligência artificial e redes neurais. 2020. Disponível em: <https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/106-inteligencia-artificial-e-redes-neurais>. Acesso em: 01 fev 2022. 17
- BRANCO. *Overfitting e underfitting em Machine Learning*. 2020. Disponível em: <https://abracd.org/overfitting-e-underfitting-em-machine-learning/>. Acesso em: 07 fev 2022. 25
- BRIGNOLI, R. *ESTUDO DE REDES NEURAIIS CONVOLUCIONAIS E SUA APLICAÇÃO PARA CLASSIFICAÇÃO DE AMBIENTES*. 2021. 45 p. Disponível em: <https://repositorio.ifsc.edu.br/handle/123456789/2359>. 18, 20, 21, 26, 28, 29, 31
- COTTONBABY. *Quais as diferenças entre lenços e toalhas umedecidas para a higiene diária?* 2020. Disponível em: <http://www.cottonbaby.com.br/blog/sua-familia/quais-as-diferencas-entre-lencos-e-toalhas-umedecidas-para-a-higiene-diaria/>. Acesso em: 03 fev 2022. 14
- CUNHA. *Reconhecimento e detecção de logotipos a partir de redes neurais convolucionais profundas*. 2017. 29
- DATA SCIENCE ACADEMY. *Capítulo 6 – O Perceptron – Parte 1*. 2022. Disponível em: <https://www.deeplearningbook.com.br/o-perceptron-parte-1/>. Acesso em: 03 fev 2022. 20
- DO S, SONG KD, CHUNG JW. *Basics of Deep Learning: A Radiologist's Guide to Understanding Published Radiology Articles on Deep Learning*. 2020. Acesso em: 03 fev 2022. 27
- ECONOMIA DE SERV. *A economia da beleza: o setor de higiene pessoal, perfumaria e cosméticos*. 2016. Disponível em: <https://economiadeservicos.com/2016/04/19/a-economia-da-beleza-o-setor-de-higiene-pessoal-perfumaria-e-cosmeticos/>. Acesso em: 30 jan 2022. 14
- HAYKIN, S. *Redes neurais: princípios e prática*. 2°. ed. [S.l.]: Bookman, 2007. ISBN 9788573077186. 18
- HE, K. *et al. Deep Residual Learning for Image Recognition*. 2015. 26
- HOWARD, J.; GUGGER, S. *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD*. O'Reilly Media, Incorporated, 2020. ISBN 9781492045526. Disponível em: <https://books.google.com.br/books?id=xd6LxgEACAAJ>. 21, 28
- IBM, C. E. *What is Machine Learning?* 2020. Disponível em: <https://www.ibm.com/cloud/learn/machine-learning>. Acesso em: 02 fev 2022. 17

- KUGLER, B. H. *Análise da utilização de um sistema machine vision para controle de qualidade de embalagens longa vida*. 2019. Disponível em: [https://repositorio.utfpr.edu.br/jspui/bitstream/1/24512/1/PG\\_DAMEC\\_2019\\_2\\_13.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/24512/1/PG_DAMEC_2019_2_13.pdf). 12
- LEMOS, Y. M. V. *Inspeção automática de defeitos em ovos comerciais usando visão computacional*. 2021. 170 p. Disponível em: <https://repositorio.ufsc.br/handle/123456789/227084>. Acesso em: 20 jan 2022. 20, 23, 30
- MAJIN, J. J. *Desenvolvimento de um sistema de contagem e classificação de veículos utilizando redes neurais convolucionais*. 2021. Disponível em: <https://repositorio.ufsc.br/handle/123456789/227085>. 12, 21, 22, 23, 26
- PARKHI, O. M.; VEDALDI, A.; ZISSERMAN, A. Deep face recognition. In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2015. p. 41.1–41.12. ISBN 1-901725-53-7. Disponível em: <https://dx.doi.org/10.5244/C.29.41>. 24
- PROBST, G.; MELO, D.; ROLOFF, M. Openvisionbot — pesquisa e desenvolvimento de urna plataforma robótica móvel aberta. 11 2010. 16
- REIS, B. *Redes Neurais – Funções De Ativação*. 2016. Disponível em: <http://www2.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>. Acesso em: 03 fev 2022. 19
- SAS. *Visão Computacional - O que é e qual sua importância?* 2020. Disponível em: [https://www.sas.com/pt\\_br/insights/analytics/computer-vision.html](https://www.sas.com/pt_br/insights/analytics/computer-vision.html). Acesso em: 01 fev 2022. 17
- SHAPIRO, L.; STOCKMAN, G. *Computer Vision*. [S.l.: s.n.], 2000. 16
- SILVA, R. E. V. D. *UM ESTUDO COMPARATIVO ENTRE REDES NEURAIAS CONVOLUCIONAIS PARA A CLASSIFICAÇÃO DE IMAGENS*. 2018. 52 p. 22
- VASCONCELLOS. Como saber se seu modelo de machine learning está funcionando mesmo. 2018. Disponível em: <https://paulovasconcellos.com.br/como-saber-se-seu-modelo-de-machine-learning-está-funcionando-mesmo-a5892f6468b>. Acesso em: 07 fev 2022. 30
- YAMASHITA, R. *et al.* Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, v. 9, p. 611 – 629, 2018. 21