

**UNIVERSIDADE FEDERAL DE ALAGOAS
UNIDADE ACADÊMICA CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA QUÍMICA**

FILIFE BRITO PEREZ

**AVALIAÇÃO DA APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NA DETECÇÃO
DO USO DE EQUIPAMENTOS DE PROTEÇÃO INDIVIDUAL EM AMBIENTES DE
TRABALHO**

Maceió
2022

FILIPE BRITO PEREZ

**AVALIAÇÃO DA APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NA DETECÇÃO
DO USO DE EQUIPAMENTOS DE PROTEÇÃO INDIVIDUAL EM AMBIENTES DE
TRABALHO**

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Química da Universidade Federal de Alagoas como requisito para obtenção do título de Bacharel em Engenharia Química.

Orientador: Prof^a. Dr^a. Livia Maria de Oliveira Ribeiro

Maceió
2022

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

P438a Perez, Filipe Brito.
Avaliação da aplicação da inteligência artificial na detecção do uso de Equipamentos de Proteção Individual em ambientes de trabalho / Filipe Brito Perez. – 2022.
48 f. il. : figs. ; tabs. color.

Orientadora: Livia Maria de Oliveira Ribeiro.
Monografia (Trabalho de Conclusão de Curso em Engenharia Química).
Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2022.

Bibliografia: f. 45-48.

1. Inteligência artificial. 2. Equipamentos de Proteção Individual - Detecção. 3. Segurança do trabalho. I. Título.

CDU: 331.45:004.383.8

RESUMO

O Equipamento de Proteção Individual (EPI) providencia uma segurança maior ao serviço de um trabalhador, minimizando os riscos de acontecer acidentes, proteger de doenças relacionadas ao ambiente de trabalho, e até prevenir fatalidades. Porém, diversos trabalhadores não utilizam os EPIs obrigatórios, e, conseqüentemente, os dados relacionados à esta situação são alarmantes. A Organização Internacional de Trabalho (OIT) relatou que a cada 15 segundos, uma morte é provocada por conta de acidentes ou doenças provenientes do ambiente de trabalho. Além disto, 153 trabalhadores estão envolvidos em acidentes trabalhistas, acidentes que ocorrem pelo exercício do trabalho a serviço da empresa. Ademais, 6300 mortes diárias são registradas decorrentes de acidentes ou doenças ligadas ao trabalho, totalizando uma quantidade por volta de 2,3 milhões de mortes anuais (Silva, 2018). Para combater isto, este trabalho será direcionado à detecção de EPIs utilizando técnicas de Deep Learning com o algoritmo de rede neural convolucional destinada à detecção de objetos em tempo real, como a *You only look once* (YOLO), em que na sua quarta versão escalada (*Scaled-YOLOv4*), seu tempo de inferência (tempo de previsão) chega até 12 milissegundos (Wang, 2021). Sendo assim, essa técnica será implantada nos laboratórios químicos situados na Universidade Federal de Alagoas.

Palavras-Chave: EPI, acidente de trabalho, redes neurais, YOLO

ABSTRACT

Personal Protective Equipment (PPE) provides greater safety at the service of a worker, minimizing the risks of accidents, protecting against diseases related to the work environment, and even preventing fatalities. However, several workers do not use the mandatory PPE, and, consequently, the data related to this situation are alarming. The International Labor Organization (ILO) reported that every 15 seconds, a death is caused due to accidents or diseases arising from the work environment. In addition, 153 workers are involved in labor accidents, accidents that occur due to the exercise of work at the company's service. In addition, 6300 daily deaths are recorded due to accidents or illnesses related to work, totaling around 2.3 million annual deaths (Silva, 2018). To combat this, this work will be directed to the detection of PPE using Deep Learning techniques with the convolutional neural network algorithm for the detection of objects in real time, such as You only look once (YOLO), in which in its fourth scaled version (Scaled-YOLOv4), its inference time (prediction time) reaches up to 12 milliseconds (Wang, 2021). Therefore, this technique will be implemented in the chemical laboratories located at the Federal University of Alagoas.

Key words: PPE, accident at work, neural networks, YOLO.

LISTA DE FIGURAS

Figura 1 - Principais equipamentos de proteção individual.	15
Figura 2 - Tipos de luvas para proteção individual.	17
Figura 3 - Representação visual de um Perceptron.	20
Figura 4 - Gráfico da função de ativação ReLU.....	21
Figura 5 - Representação visual de uma RNA.	21
Figura 6 - Etapas do ciclo de treinamento de uma rede neural.....	23
Figura 7 – Exemplo de curva de uma <i>loss function</i>	24
Figura 8 - Representação visual da arquitetura de uma CNN	25
Figura 9 - Exemplo de um <i>kernel</i> em uma fonte de pixel (imagem ou feature map).....	26
Figura 10 - Exemplo de detecção de objetos da arquitetura YOLO.....	29
Figura 11 - Exemplo de imagem coletada.	31
Figura 12 - Interface da plataforma <i>Roboflow</i>	32
Figura 13 - <i>Bounding boxes</i> produzidas com a ferramenta <i>labelImg</i>	33
Figura 14 - Dados gerados pela ferramenta <i>labelImg</i>	33
Figura 15 - Demonstração do fenômeno <i>overfitting</i>	35
Figura 16 - Representação gráfica do cálculo do IoU.	37
Figura 17 - Exemplos de TN, FP e FN.	37
Figura 18 - Exemplo de uma curva de Precisão-Recall.....	39
Figura 19 - Blocos de códigos do <i>Google Colab</i>	40
Figura 20 - Valores de erros de treino e validação ao longo das épocas da <i>YOLOv5</i>	43
Figura 21 - Valores de erros de treino e validação ao longo das épocas da <i>YOLOR</i>	44
Figura 22 - mAP@0.5 ao longo das épocas para <i>YOLOv5</i>	44
Figura 23 - mAP@0.5 ao longo das épocas para <i>YOLOR</i>	45
Figura 24 - mAP@0.5 ao longo das épocas para <i>YOLOR</i>	45
Figura 25 - mAP@0.5:0.95 ao longo das épocas para <i>YOLOv5</i>	46
Figura 26 - mAP@0.5:0.95 ao longo das épocas para <i>YOLOR</i>	46
Figura 27 - mAP@0.5:0.95 ao longo das épocas para <i>YOLOX</i>	46

LISTA DE TABELAS

Tabela 1 - Informações contidas nos arquivos de etiquetação.	33
Tabela 2 - Representação das classes nos arquivos de etiquetação.	34
Tabela 3 - Informações de treinamento de cada modelo.	42
Tabela 4 - Métricas de treino dos modelos.	47
Tabela 5 - mAP das classes da <i>YOLOv5</i>	48
Tabela 6 - mAP das classes da <i>YOLOR</i>	48
Tabela 7 - mAP das classes da <i>YOLOX</i>	48

LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Average Precision</i>
API	<i>Application Programming Interface</i>
CNN	<i>Convolutional Neural Network</i>
COLAB	<i>Colaboratory Jupyter network environment</i>
CSS	<i>Cascading Style Sheets</i>
EPI	Equipamentos de Proteção Individual
FN	<i>False Negative</i>
FP	<i>False Positive</i>
FPS	<i>Frames Per Second</i>
GPU	<i>Graphics Processing Unit</i>
HTML	<i>HyperText Markup Language</i>
IA	Inteligência Artificial
IoU	<i>Intersection over Union</i>
LASSOP	Laboratório de Sistema de Separação e Otimização de Processos
LER	Lesões por Esforços Repetitivos
mAP	<i>Mean Average Precision</i>
MS COCO	<i>Microsoft Common Objects in Context</i>
NR	Norma Reguladora
PMC	Perceptron Multicamadas
R-CNN	<i>Region-based Convolutional Neural Network</i>
RAM	<i>Random Access Memory</i>
ReLU	<i>Rectified Linear Unit</i>
RNA	Redes Neurais Artificiais
RNC	Rede Neural Convolutacional
TP	<i>True Positive</i>
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1.	INTRODUÇÃO.....	11
2.	OBJETIVOS.....	13
2.1	Geral.....	13
2.2	Específicos.....	13
3.	REVISÃO BIBLIOGRÁFICA.....	14
3.1	Segurança do Trabalho.....	14
3.1.1	Acidentes e Doenças de Trabalho.....	14
3.1.2	Equipamento de Proteção Individual (EPI).....	15
3.2	Computação Visual.....	18
3.3	Inteligência Artificial.....	18
3.4	Redes Neurais.....	19
3.4.1	Perceptron.....	20
3.4.2	Rede Neural Artificial.....	21
3.4.3	Redes Neurais Convolutivas.....	24
4.	METODOLOGIA.....	30
4.1	Coleta de imagens.....	30
4.2	Aumentação dos dados.....	31
4.3	Etiquetagem de imagens.....	32
4.4	Treinamento dos modelos.....	34
4.4.1	Época.....	34
4.4.2	<i>Batch</i>	35
4.5	Métricas de performance.....	36
4.5.1	<i>Intersection over Union (IoU)</i>	36
4.5.2	<i>Mean Average Precision (mAP)</i>	37
4.5.3	<i>Frames Per Second (FPS)</i>	39
4.6	Ambiente de Treinamento.....	39

	10
4.7 Modelos de Detecção	40
4.7.1 YOLOv5	41
4.7.2 YOLOR	41
4.7.3 YOLOX	41
4.8 Tempo de treinamento	41
5. RESULTADOS	43
5.1 Desempenho de treinamento.....	43
5.2 Desempenho de teste	44
5.2.1 mAP das Classes.....	47
6. CONCLUSÃO.....	49
REFERÊNCIAS	50

1. INTRODUÇÃO

A saúde dos trabalhadores é essencial para manter um ambiente de trabalho saudável e produtivo (ANTONIAZZI et al., 2009). A segurança do trabalho enfatiza a prevenção dos acidentes de trabalhos decorrentes dos inúmeros riscos existentes nos ambientes em que este é executado, tais como: riscos ergonômicos, químicos, físicos, biológicos e de acidentes (JÚNIOR et al., 2012).

O uso de Equipamento de Proteção Individual (EPI) está diretamente relacionado com a segurança individual do trabalhador. Porém, diversos trabalhadores relatam desconforto com a utilização de EPI, deixando de cumprir com o dever dos trabalhadores, que é seguir a Norma Regulamentadora (NR) 06 (SILVA et al., 2018).

De acordo com o Observatório de Segurança e Saúde no Trabalho, foram registradas mais de 5,6 milhões de acidentes de trabalho entre 2012 e 2020, gerando gastos previdenciários, desde 2012, de mais de R\$ 100 bilhões de reais, apenas com despesas acidentárias (OIT, 2021). Dentre as principais causas, estão: não utilizar o EPI adequadamente, quedas, uso de ferramentas inadequadas e estresse.

A tecnologia da informação desempenha um papel fundamental na esfera mundial, estando presente em diferentes aspectos da sociedade. Atualmente, observa-se uma grande dependência dessa tecnologia por parte das indústrias. O conceito de combinar os principais avanços tecnológicos com os processos industriais está sendo considerado como a Quarta Revolução Industrial, mais conhecida como Indústria 4.0. (STHEL e LOUREIRO, 2017).

Com a tecnologia presente em todos os âmbitos que envolvem a sociedade, e a crescente da Indústria 4.0., a Inteligência Artificial (IA) vem ganhando destaque por avanços recentes na área, e sua grande capacidade de resolver problemas complexos, otimizando resultados sem precedentes.

Inteligência Artificial possui um grande potencial para aumentar produtividade, auxiliando empresas e pessoas a usarem os recursos de maneira mais eficiente e otimizando a maneira de como interagir com grandes conjuntos de dados (HALL e PESENTI, 2017). Empresas da indústria química de grande porte, como a Bayer, utilizam a Inteligência Artificial para alcançar melhores resultados, como a predição de doenças cardiovasculares e doenças crônicas (BAYER, 2020).

Uma das causas dos avanços recentes da Inteligência Artificial se dá pelo crescimento vertiginoso do poder computacional das máquinas (THOMPSON et al., 2020), possibilitando cálculos computacionais com matrizes em grande escala, como é o caso das Redes Neurais Convolucionais (RNC), uma arquitetura de rede neural voltada à execução de processos matemáticos e estatísticos em imagens. Há arquiteturas que permitem a detecção de objetos em tempo real, como a *You Only Look Once* (YOLO) (REDMON e FARHADI, 2018).

No desenvolvimento deste trabalho será realizado a avaliação de modelos para detecção de Equipamentos de Proteção Individual (EPI) em laboratórios químicos na Universidade Federal de Alagoas, utilizando Redes Neurais Artificiais focadas em detecção de objetos em tempo real.

2. OBJETIVOS

2.1 Geral

Este trabalho tem como objetivo geral a avaliação de técnicas computacionais aplicadas à prática de segurança do trabalho e utilização de Equipamentos de Proteção Individual (EPI).

2.2 Específicos

- Investigar os principais equipamentos utilizados nos laboratórios de ensino e pesquisa de Química e Engenharia Química, bem como seu uso;
- Avaliar a aplicabilidade de técnicas computacionais, Redes Neurais Artificiais, na detecção de objetos em tempo real;
- Avaliar a incidência de acidentes no Laboratório de Sistema de Separação e Otimização de Processos (LASSOP).
- Construir um modelo de detecção em tempo real de Equipamentos de Proteção Individual (EPI) utilizando a técnica de Redes Neurais Artificiais Convolucionais.

3. REVISÃO BIBLIOGRÁFICA

3.1 Segurança do Trabalho

Segundo Barsano e Barbosa (2018), a segurança do trabalho é a ciência que estuda as possíveis causas dos acidentes e incidentes durante a atividade laboral do trabalhador. O principal propósito desta área é prevenir acidentes, doenças ocupacionais e outras formas que seja danoso à saúde do trabalhador. Quando sua finalidade é atingida, proporciona um ambiente de trabalho seguro e saudável, tanto para o empregado, quanto para o empregador.

3.1.1 Acidentes e Doenças de Trabalho

Acidente de trabalho é definido pelo Ministério da Previdência Social como aquele que ocorre com um empregado, ou trabalhador avulso, no exercício de suas atividades, acarretando em lesão corporal ou alguma perturbação funcional que sua consequência seja o óbito, a perda ou redução, temporária ou permanente, da capacidade para o trabalho (BRASIL, 1999).

O Brasil registrou dados alarmantes relacionados aos registros de acidentes de trabalho e doenças profissionais. De acordo com o Sistema de Informação de Agravos de Notificação (SINAN) do Ministério de Saúde, a quantidade de acidentes de trabalhos graves notificados registrou um aumento de 40% no ano de 2020, em que o número de acidentes saltou de 94.353 em 2019 para 132.623 no ano posterior (OIT, 2021).

De acordo com a Organização Mundial de Saúde, as doenças relacionadas ao trabalho são divididas em duas categorias, que são as doenças profissionais ou do trabalho (BRASIL, 2001b).

De acordo com esta classificação, para que seja classificada como doença profissional é preciso que esta doença seja inerente às atividades executadas em ambiente de trabalho, pois haveria exposição a esses agentes. Portanto, a ligação entre atividades e patologia seria de fácil assimilação. Exemplos de doenças profissional seriam silicose e pneumoconiose (inalação de poeiras inorgânicas).

Já na categoria de doenças relacionadas ao trabalho, as doenças de trabalho, “são enquadradas as afecções nas quais não se identifica apenas um agente causal, mas vários, entre os quais os laborais” (BRASIL, 2001b). Em outras palavras, não são doenças que foram necessariamente causadas no trabalho, mas as atividades laborais podem contribuir a ocorrência da doença. Pode-se citar as Lesões por Esforços Repetitivos (LER), como exemplo.

Para que a transmissão de doenças e riscos de acidentes provenientes de um ambiente laboral, é necessário medida diversificadas para reduzir o risco ocupacional. O apropriado uso dos Equipamentos de Proteção Individual (EPI) diminui consideravelmente esses riscos, e é considerada uma das principais medidas preventivas.

3.1.2 Equipamento de Proteção Individual (EPI)

De acordo com o Ministério de Trabalho e Emprego, é considerado como Equipamento de Proteção Individual “todo dispositivo ou produto, de uso individual utilizado pelo trabalhador, destinado à proteção de riscos suscetíveis de ameaçar a segurança e a saúde no trabalho” (BRASIL, 2018). Também é possível ter equipamentos compostos por vários dispositivos. É possível observar os principais equipamentos na Figura 1.

Figura 1 - Principais equipamentos de proteção individual.



Fonte: Disponível em: < <https://blog.rhmateriaiseletricos.com.br/porque-devemos-usar-epi>>. Acesso em: 29 de abr. de 2021

Os EPIs exigem responsabilidades tanto do empregador, como do empregado. Pelo lado do empregador, suas funções são: fornecer aos empregados, gratuitamente, EPI adequado ao risco, que esteja em perfeitas condições; exigir seu uso; orientar e treinar o trabalhador sobre como usá-lo adequadamente, guardá-lo e conservá-lo; substituir de forma imediata quando o EPI for danificado ou extraviado, entre outros (BRASIL, 2018).

Já as responsabilidades do empregado em relação ao EPI são: utilizá-los, obrigatoriamente, apenas quando for para a atividade necessária; responsabilizar-se pela guarda e conservação; notificar o empregador quando o equipamento for danificado e cumprir as determinações do empregador sobre o uso adequado (BRASIL, 2018).

O EPI é a ferramenta fundamental para a prevenção de acidentes, entretanto, a resistência do profissional em utilizá-lo e o seu uso incorreto são as principais barreiras para prevenir a exposição de acidentes e doenças (MALAGUTI, 2008).

A baixa adesão ao uso dos equipamentos de proteção individual e o seu manuseio incorreto são decorrentes de fatores como desconforto, incômodo, descuido, esquecimento, falta de hábito, inadequação dos equipamentos, quantidade insuficiente e a descrença quanto ao seu uso (TIPPLE et al., 2007; SOUZA et al., 2008). Pode ser somado outros fatores como estresse, cansaço físico e infraestrutura precária

Há alguns EPIs que são essenciais quando se trata de laboratórios químicos devido aos riscos presentes nestes locais, principalmente os citados na Norma Regulamentadora (NR) 09: “compostos ou produtos que possam penetrar no organismo pela via respiratória, nas formas de poeiras, fumos, névoas, neblinas, gases ou vapores, ou que, pela natureza da atividade de exposição, possam ter contato ou ser absorvidos pelo organismo através da pele ou por ingestão.” (BRASIL, 2020).

3.1.2.1 Calça e calçados

De acordo com a NR 6, é preciso ter calças e calçados para proteção das pernas e dos pés, respectivamente, contra respingos de produtos químicos (BRASIL, 2018). Com isto, é necessário utilizar calçados que proteja todas as partes dos pés, sem nenhuma abertura, assim como a calça não deve ter nenhum rasgão.

3.1.2.2 Jaleco

De acordo com o Manual de Biossegurança do Estado do Espírito Santo (2017), o jaleco fornece uma proteção, reduzindo a oportunidade de transmissão de microrganismos e contaminação química, além de proteger a pele da exposição de sangue e fluidos corpóreos, salpicos e derramamentos de material infectado.

É preferível que o jaleco possua mangas longas, sendo confeccionado em algodão ou fibra sintética. O equipamento deve ser resistente e impermeável, e não deve ser utilizado fora de um laboratório.

3.1.2.3 Luvas

Existem luvas para situações específicas que podem ocorrer no cotidiano de um laboratório. A mais comum é a luva de látex, em que é utilizada para procedimentos do modo geral, oferecendo proteção contra agentes biológicos, ácidos e bases diluídos, exceto para solventes orgânicos (HOSPITAL DAS CLÍNICAS, 2015).

Como pode ser visto na Figura 2, há diversos tipos de luvas, cada uma especializada em um tipo de proteção, em que são:

- a) de fibra de vidro com polietileno reversível: para proteção contra materiais cortantes;
- b) de PVC e látex nitrílico: para produtos químicos;
- c) térmicas de nylon: específica para o manuseio de materiais em temperaturas extremamente baixas;
- d) de fio de kevlar tricotado: para o manuseio de materiais com temperaturas até 250°C.

Figura 2 - Tipos de luvas para proteção individual.



Fonte: Disponível em: <<https://www.ktequipamentos.com.br/luvas-protecao-individual>>. Acesso em: 06 de set. de 2021.

3.1.2.4 Máscaras

O uso da máscara é destinado para proteção da face contra respingos químicos, e também névoas e poeiras, além de minimizar a inalação de fases e produtos voláteis (HOSPITAL DAS CLÍNICAS, 2015). Há dois tipos de máscaras, as semi-faciais e as de proteção total.

A primeira é prática, leve, e de fácil manuseio. Sua principal finalidade é reter poeira e partículas, além de vapores tóxicos que sua concentração não ultrapasse dez vezes o limite de exposição. Já as máscaras de proteção total suportam concentrações até cinquenta vezes o limite (FILHO, 2008).

As máscaras semi-faciais se tornaram comuns nos laboratórios do Brasil para também combater a proliferação do vírus SARS-CoV-2 no período de março de 2020 até o momento atual (setembro de 2021).

3.1.2.5 Óculos protetor

O tipo de óculos que mais se adequa ao contexto do laboratório é aquele que oferece proteção contra respingos de agentes corrosivos, irritações e entre outros ferimentos causados pela ação de produtos químicos, radiações e partículas sólidas. É importante que os óculos proporcionem uma visão transparente, sem distorção (ESPÍRITO SANTO, 2017).

3.2 Computação Visual

Este segmento da computação tenta simular a capacidade de ver e interpretar objetos que um olho humano possui. A compreensão do órgão humano acontece no córtex visual do cérebro, uma das partes mais complexas no sistema de processamento do cérebro, de uma forma muito rápida (MILANO e HONORATO, 2010).

Segundo Crowley e Christensen (1995), visão computacional é a área da ciência que estuda e procura desenvolver teorias e métodos voltados à extração automática de informações úteis contidas em imagens. Tais imagens são capturadas por dispositivos como câmera de vídeo ou scanner.

Segundo Forsyth e Ponce (2011), a definição de computação visual é dita como um mecanismo que usa métodos estatísticos para desembaralhar os dados usando modelos construídos com ajuda da geometria, física e teoria do aprendizado, precisando de um entendimento razoável sobre os valores individuais dos pixels.

Pixel é a unidade básica constituinte de uma imagem, em que pode ter apenas um canal de cor em escala de cinza, ou três canais separados de cor, sendo estes canais representando a quantidade de vermelho, de verde e de azul (RGB, em inglês), em que cada canal varia de 0 a 255. Conseqüentemente, o pixel também é a unidade básica de um vídeo, pois este é o conjunto de imagens capturadas ao longo do tempo.

3.3 Inteligência Artificial

Inteligência Artificial é amplamente definida com o que se preocupa com comportamentos inteligentes, estes envolvem percepção, raciocínio, aprendizagem, comunicação e ação em ambientes complexos. A IA tem como objeto fazer com que máquinas realize ações tão bem, ou até melhor do que humanos (NILSSON, 1998).

Ao decorrer dos últimos avanços, IA se tornou parte intrínseca da evolução da sociedade atual. O desenvolvimento desta área veio a partir do rápido surgimento de sistemas de computadores que pudessem executar processos em uma grande quantidade de dados de maneira rápida e de modo mais efetivo e preciso que um humano (IBM, 2020).

Na contemporaneidade, as aplicações da IA são diversas, algumas delas são: detecção de células cancerígenas através de imagens, produção de música e arte similares a de um humano, possibilidade da comunicação entre surdos e ouvintes através de tradução entre língua oralizada e língua de sinais, entre outras. A resolução desses problemas complexos são possíveis graças as Redes Neurais.

3.4 Redes Neurais

Uma grande dificuldade de uma máquina compreender determinada situação por completo, ou pelo menos em níveis humanos, é a quantidade de fatores que podem influenciar no dado, como por exemplo, identificar um rosto humano em uma imagem, sendo que a representação, em pixel, de cada rosto humano chega a se tornar única, além de que o rosto pode aparecer em qualquer posição da imagem, e em qualquer ângulo.

As Redes Neurais Artificiais (RNAs) resolvem este tipo de problema, pois consegue abstrair representações de menor complexidade, que em conjunto conseguem representar dados mais complexos (GOODFELLOW, BENGIO e COURVILLE, 2015).

A área que estuda as RNAs é o *Deep Learning*, segmento da Inteligência Artificial que investiga diversas áreas, como: computação visual, processamento natural de linguagem, tradução de máquina, reconhecimento de voz, entre outros.

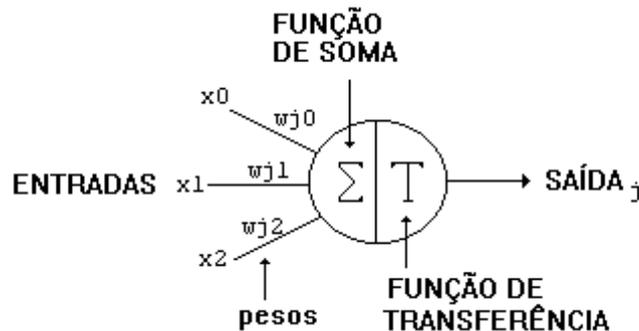
Os algoritmos de *Deep Learning*, assim como *Machine Learning*, podem ser classificados pelo tipo de dados utilizados, afetando o modo em que irão aprender. As classificações podem ser: aprendizado por supervisão, em que os dados são etiquetados, ou seja, o dado estará acompanhado com a classe a que pertence, e seu objetivo é prever valores de duas maneiras: por classificação (as previsões são finitas e discretas) e por regressão (as previsões são infinitas e contínuas) (HARRINTON, 2012).

Há também o aprendizado não-supervisionado, em que os dados não vêm com a classe que pertencem. Os algoritmos deste tipo tentam extrair informações importantes a partir dos dados, como agrupá-los em diferentes grupos a partir de semelhanças entre as características (conhecido como clusterização), por exemplo (HARRINTON, 2012).

3.4.1 Perceptron

Para poder entender uma RNA, é necessário compreender sobre a sua unidade primária, o *Perceptron*, que é nada mais do que uma arquitetura de rede neural de uma camada por si só, atuando como um classificador binário. Sua representação visual pode ser vista na Figura 3.

Figura 3 - Representação visual de um Perceptron.



Fonte: Disponível em: <<http://redesneuraisartificiais.blogspot.com/2010/10/o-primeiro-modelo-de-um-neuronio-criado.html>>. Acessado em: 18/08/2021.

x_i : Entradas de dados

w_j : Pesos associados às entradas

Σ : Soma ponderada das entradas

T: Função de ativação (função de transferência)

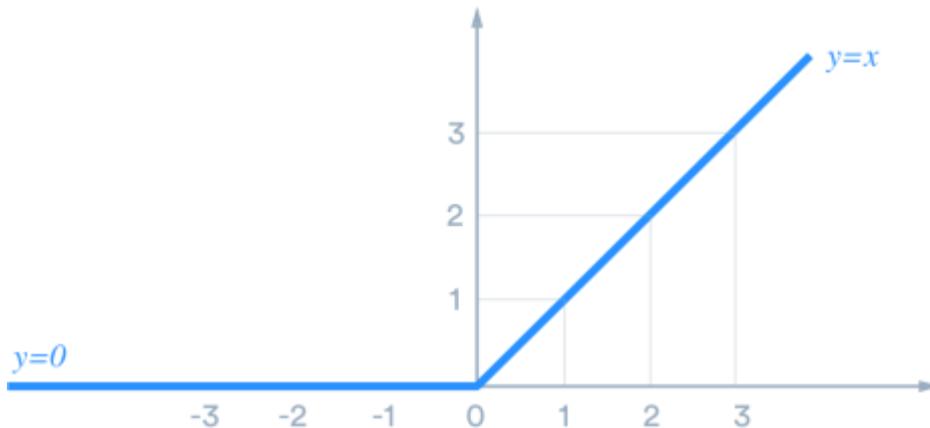
O funcionamento de um *Perceptron* se dá por um processo iterativo, a partir da entrada de dados, x_i , em que pode ser considerada uma camada de entrada, e cada um dessas entradas terão um peso, w_j , associado. Estes pesos são iniciados com valores aleatórios, e vão mudando ao longo das iterações, para que consiga produzir previsões de boa qualidade.

São os pesos que possibilitam a capacidade de aprender de um *Perceptron*, e conseqüentemente, de uma rede neural. Após isto, as entradas são somadas ponderadamente em relação aos seus pesos associados dentro do nó (este está sendo representado como um círculo na Figura 3). Então é aplicada uma função de ativação para retirar a linearidade do algoritmo, viabilizando a resolução de problemas não-lineares.

Existem diversas funções de ativação, como por exemplo o *Rectified Linear Unit* (ReLU), em que transforma todos valores negativos em zero, e os valores positivos permanecem com o mesmo valor. Em outras palavras, a função irá produzir o máximo entre 0 e o resultado da soma ponderada, como é mostrado na equação 1. Esta é uma das funções mais utilizada atualmente (XU e ZHANG, 2021), e seu gráfico pode ser visto Figura 4.

$$f(x) = \max(0, x) \quad (1)$$

Figura 4 - Gráfico da função de ativação ReLU.



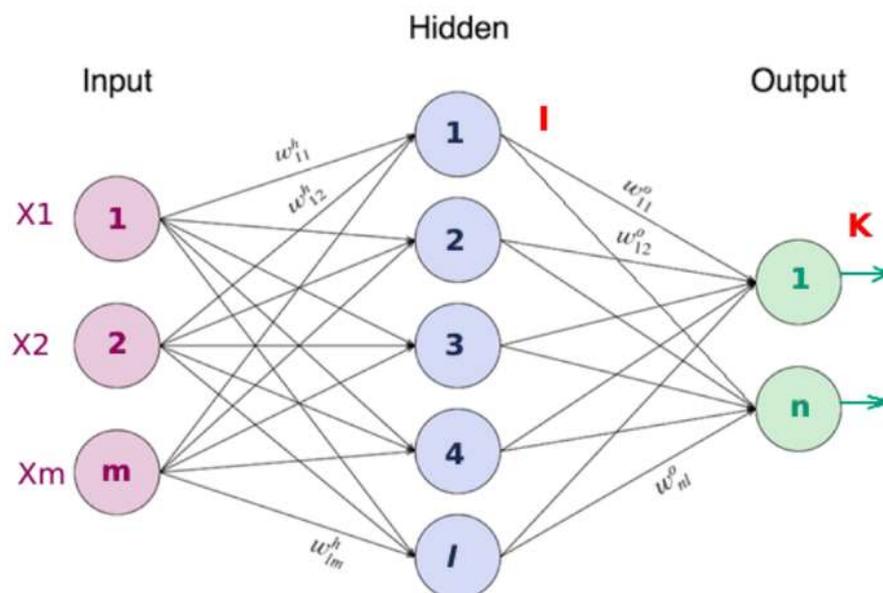
Fonte: Disponível em: <<https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>>. Acessado em: 18/08/2021.

A concepção do *perceptron* teve influência no funcionamento de um neurônio, em que este recebe sinais elétricos dos dendritos, modula estes sinais e entrega diversos outros sinais através das sinapses, sendo estas ativadas quando ultrapassam um limiar. Este fenômeno se propaga através dos outros neurônios.

3.4.2 Rede Neural Artificial

O conjunto de diversos *perceptrons* organizados em camadas é chamado de Rede Neural Artificial (RNA) ou Perceptron Multicamadas (PMC), sendo esta é a arquitetura mais simples das redes neurais. Sua representação visual pode ser vista na Figura 5. Aqui, cada *perceptron* é chamado de nó ou neurônio.

Figura 5 - Representação visual de uma RNA.



Fonte: Disponível em: <<https://medium.com/swlh/neural-networks-4b6f719f9d75>>. Acesso em: 29 de abr. de 2021

x_i : Dados de entrada

w_{ij}^k : Pesos associados à camada k aos nós i e j das camadas interligadas

l, k, \dots : Camadas da rede neural

O nome *Deep Learning* vem justamente das camadas presentes nas redes neurais. Além da camada de entrada, em que os dados são introduzidos na rede neural, e a camada de saída, sendo essa a que produz a resposta prevista pelo algoritmo, há camadas escondidas, ou profundas, que em inglês é justamente *deep*, que são camadas que realizam transformações nos dados para que a previsão da rede seja a mais fiel à realidade possível.

A organização de uma rede neural (entrada, camadas escondidas e saída) é o que compõe a sua arquitetura. Além da organização, há tipos de camadas que podem servir para diferentes propósitos. A camada escondida presente na Figura 5 é uma camada densa, em que todos os neurônios estão conectados com os neurônios da camada posterior. É importante ressaltar que a quantidade de neurônios presentes em cada camada é arbitrária, fazendo parte do design da arquitetura.

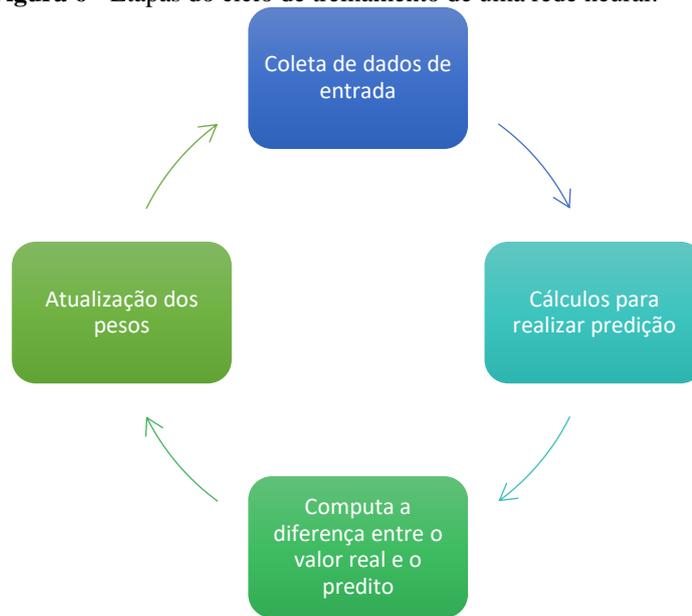
Sendo assim, existem infinitas possibilidades de arquiteturas para as redes, sendo o tipo mais básico a da rede que se encontra na Figura 5, uma arquitetura em que cada camada alimenta a camada seguinte, e não há retroalimentação (exceto para realizar cálculos de treinamento), tornando-se assim uma rede neural *feedforward* (outro nome para uma RNA) (DATA SCIENCE ACADEMY, 2021).

3.4.2.1 Treinamento de uma rede

Os responsáveis pelo aprendizado do algoritmo são os pesos presentes nas camadas. Inicialmente, estes pesos serão iniciados com valores aleatórios, até ser refinado para que a rede possa produzir valores próximos ao real.

Para que uma rede seja capaz de “aprender”, o algoritmo irá realizar uma série de execuções e cálculos matemáticos consecutivamente. As execuções podem ser vistas na Figura 6. O ciclo a seguir tem como referência Chollet (2017), e será executado quantas vezes for necessário:

Figura 6 - Etapas do ciclo de treinamento de uma rede neural.



Fonte: Autor, 2021.

- a) Coleta de dados de entrada: A etapa inicial é coletar os dados para servir como fonte do aprendizado da rede;
- b) Cálculos para realizar previsão: O cálculo presente nesta etapa pode ser visto na equação 2, em que z_i é a saída da camada i , f é a função de ativação, W_i é a matriz de pesos, X_i é o vetor de entrada da camada i , e b é o termo que permite a translação para a equação;

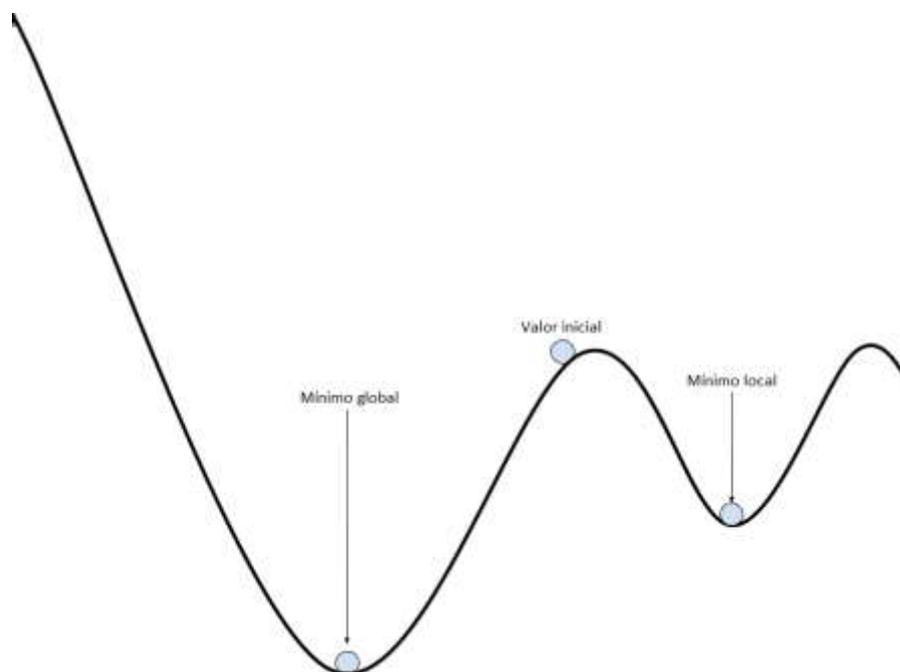
$$z_i = f(W_i \cdot X_i + b) \quad (2)$$
- c) Diferença entre o valor real e o predito: Nesta etapa será computado o quão distante a previsão da rede em relação ao valor real utilizando uma função de custo, também chamada como *loss function*. Existem diversas funções de custos específicas para cada situação;
- d) Atualização dos pesos: A última etapa é a mais complexa entre todas do ciclo, e será explicada com mais detalhes a seguir.

A atualização dos pesos tem finalidade de minimizar os erros produzidos na etapa anterior, e para realizar isto de maneira eficiente existem diversas técnicas envolvidas. A primeira técnica é a Descida do Gradiente (*Gradient Descent*), em que utiliza derivação para encontrar o erro mínimo possível. Esta técnica calcula o valor do gradiente (visto que os valores envolvidos são matrizes) do erro em função dos pesos.

Como um dos objetivos do treinamento de uma rede envolve reduzir o erro ao menor valor possível, e sendo a *loss function* uma curva dos parâmetros da rede (pesos) em relação

ao valor do erro, isto se torna um problema de otimização, em que o objetivo é encontrar o mínimo global. Um exemplo dessa problemática pode ser vista na Figura 7.

Figura 7 – Exemplo de curva de uma *loss function*.



Fonte: Autor, 2022.

Porém a curva não terá apenas dois eixos, como mostrado na Figura 7, será uma curva de cada parâmetro envolvido no treinamento, ou seja, milhares, e muitas vezes milhões, de parâmetros para analisar em relação ao erro.

Para calcular o erro usa-se um algoritmo denominado *backpropagation*, ou retropropagação, em que calcula a variação de cada peso em relação ao erro, tomando como partida a camada que produz as saídas do modelo, ou seja, a última camada, e indo em direção até a camada que o peso se encontra. Com isto, é necessário utilizar a regra da cadeia para atualizar cada peso (ALBON, 2018).

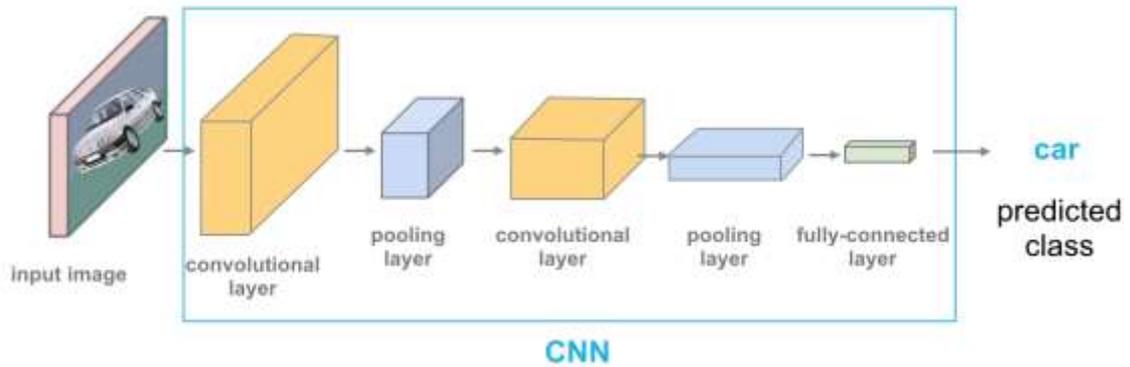
3.4.3 Redes Neurais Convolutivas

Apesar dos *perceptrons* multicamadas realizarem a operação de classificação (determinar a classe de um objeto dentre um conjunto de classes) com excelência, este tipo de arquitetura não trabalha bem quando os dados a serem manipulados são imagens, e um dos principais motivos disto ocorrer é que a dimensão dos dados de entrada necessitam ser vetores, perdendo as propriedades espaciais que uma matriz possui (ELGENDY, 2020).

Para isto, existe uma arquitetura chamada Rede Neural Convolutiva (RNC), CNN em inglês, em que o seu fundamento é o mesmo de uma PMC, porém, as *features* - padrões

presentes nos dados, no caso, imagens - são extraídas através de um conjunto de pesos compartilhados que realizam cálculos matemáticos em cada valor de um pixel. Uma representação da arquitetura de uma CNN pode ser vista na Figura 8.

Figura 8 - Representação visual da arquitetura de uma CNN



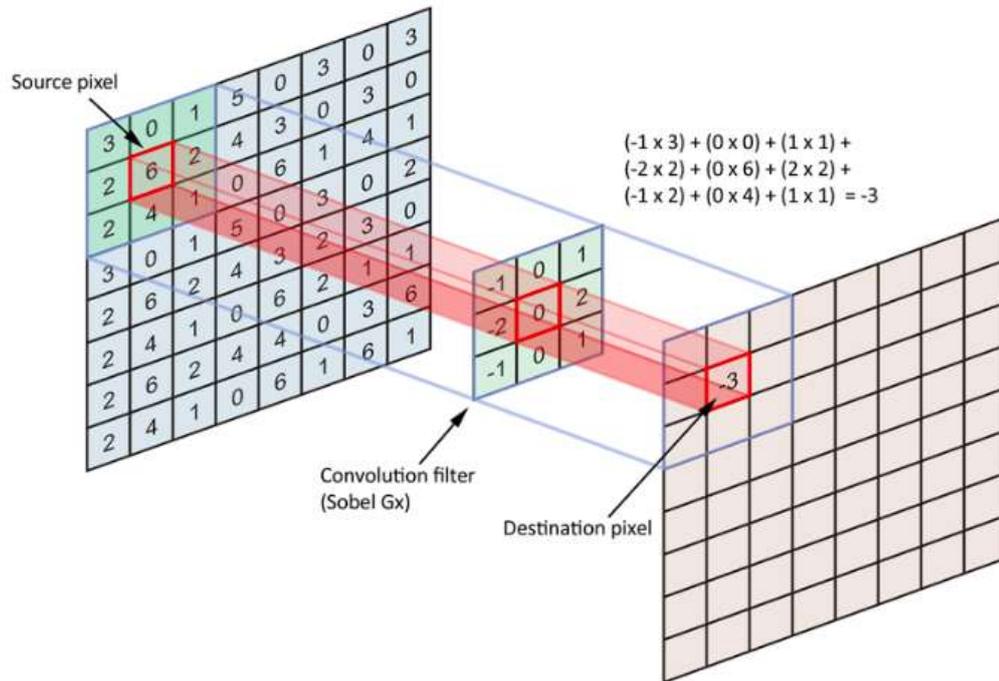
Fonte: Disponível em <https://cezannec.github.io/Convolutional_Neural_Networks/>. Acesso em: 07 de set. de 2021.

3.4.3.1 Camada convolucional

Esta é a camada mais fundamental de uma CNN, pois é nesta camada em que irá acontecer a extração de *features* e gerar *features maps*, que é nada menos que transformações matemáticas nas imagens (ou em *features maps* anteriores) apontando quais foram os padrões obtidos.

Os pesos desta camada são compartilhados, e são chamados de *kernels*, ou filtros, que são no formato de uma matriz que percorrem todos os valores de uma imagem (ou *feature map*), ou seja, em todos os pixels. Diversos *kernels* podem percorrer o mesmo *feature map*, produzindo diversos mapas de padrões (ELGENDY, 2020). A visualização de um *kernel* sobre uma imagem pode ser visto na Figura 9.

Figura 9 - Exemplo de um *kernel* em uma fonte de pixel (imagem ou feature map).



Fonte: Disponível em <<https://medium.com/ai-salon/understanding-deep-self-attention-mechanism-in-convolution-neural-networks-e8f9c01cb251>>. Acesso em: 07 de set. de 2021.

A utilização destes pesos compartilhados gera algumas vantagens em relação à camada densa: a quantidade de pesos a serem refinados caem drasticamente, pois em cada camada haverá $m \times m \times n$ pesos a serem modificados - sendo m a dimensão do *kernel* (que geralmente varia entre 2x2 e 5x5), e n , a quantidade de *kernels* por camada -, e não milhões, que geralmente estão presentes em uma única camada densa (GÉRON, 2019). Uma outra vantagem é a capacidade de extrair *features* espaciais que uma matriz possui.

3.4.3.2 Camada de *pooling* ou *subsampling*

Esta camada realiza a redução de dimensionalidade nos mapas de pixels, gerando diversos benefícios, como diminuir a quantidade de esforço computacional, e seu propósito principal, que é capturar padrões mais complexos. As primeiras camadas convolucionais irão registrar padrões mais simples, como curvas, linhas e bordas das imagens (GÉRON, 2019).

Ao reduzir as dimensões dos dados envolvidos na rede, será possível buscar padrões mais complexos e com mais detalhes, além de identificar objetos maiores presentes nos dados. Essa redução ocorre ao aplicar uma operação estatística nos mapas de pixels, sendo soma e média as operações mais comuns (GÉRON, 2019).

3.4.3.3 Camada conectada completamente ou densa

Por fim, acopla-se uma rede densa, PMC, ao fim das camadas extratoras de *features* (camadas convolucionais e de *pooling*) para poder realizar a previsão de classificação. É importante salientar que as redes neurais artificiais realizam a ação de classificação com bastante precisão, o que se torna ineficiente neste tipo de arquitetura é a extração de características e padrões em dados com dimensões de segunda ordem ou maior, como é o caso de uma imagem.

3.5 Transfer Learning

Um fenômeno que ocorre nos treinamentos de redes neurais é a similaridade dos *features maps* entre as camadas iniciais das diferentes redes, independente do objetivo de treino e o conjunto de dados utilizados (YOSINSKI et al., 2014). Isto é consequência do nível de complexidade das *features* destes níveis de camadas, em que se pode observar que são padrões mais simples, relativamente comum entre diversos domínios de dados.

Transfer Learning é o ato de empregar uma rede neural treinada em um conjunto de dados para realizar previsões em um outro domínio de dados. Como o próprio nome relata, é a transferência de conhecimento entre aplicações (problemáticas) de domínios distintos. Em termos básicos, a técnica consiste em utilizar os pesos de camadas já treinadas para aplicações diferentes. É comum denominar as redes empregadas para este propósito de redes pré-treinadas.

A aplicação desta técnica contribuiu substancialmente para o avanço desta área da inteligência artificial, visto que redes neurais de arquiteturas complexas, com conjunto de dados de grandes dimensões cujo treinamento foi realizado em máquinas de grande poder computacional estão sendo disponibilizadas gratuitamente para a comunidade conseguir utilizar estes modelos para aplicações próprias (ELGENDY, 2020).

Segundo Elgendy (2020), há alguns tipos de abordagens quando se trata de *Transfer Learning*, em que irá depender da similaridade entre as aplicações, a do autor do modelo (fonte) e a do usuário que deseja utilizar este modelo para sua aplicação própria (alvo). A similaridade será observada entre os domínios das aplicações e os conjuntos de dados utilizados. As abordagens são as seguintes:

- Rede pré-treinada como classificador: esta abordagem utilizada todas as camadas da rede, e não realizar nenhum tipo de treinamento extra. Geralmente este tipo de situação ocorre para o mesmo tipo de problemática entre as aplicações;

- Rede pré-treinada como extrator de *features*: opera-se a rede como extrator de *features*, apenas realizando o treinamento das camadas densas, camadas que realizam a classificação;
- Ajuste fino da rede pré-treinada: esta é a abordagem mais comum, em que se congela algumas camadas de extração de *features* – os pesos permanecem com os mesmos valores -, e treina-se o restante das camadas com o conjunto de dados desejado. A quantidade de camadas que irão ser congeladas dependerá da similaridade das aplicações e da quantidade de dados do *dataset* (conjunto de dados).

3.6 YOLO

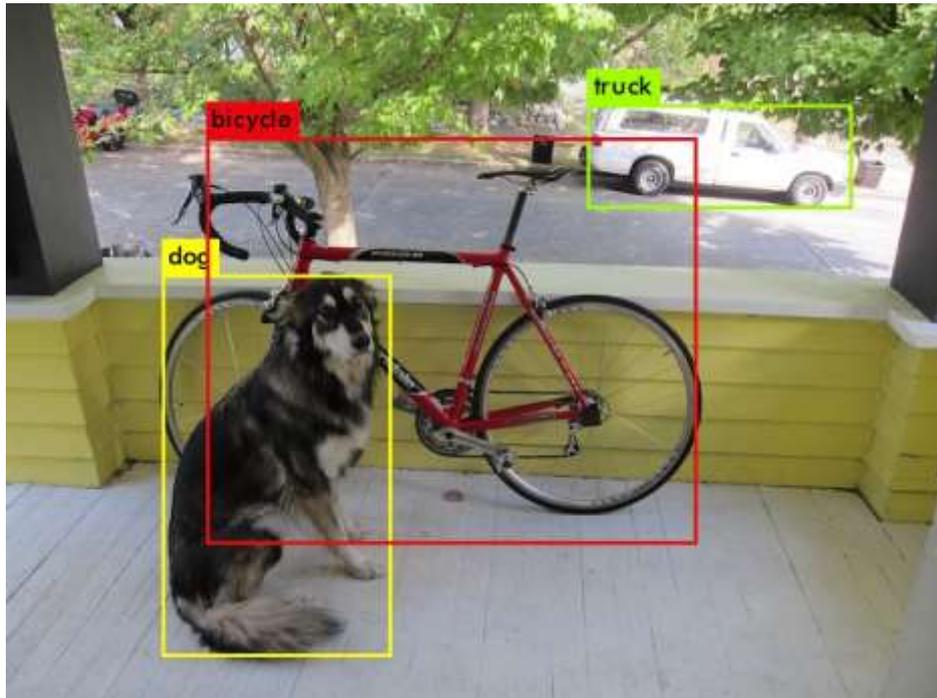
Entre os diversos propósitos que as redes neurais convolucionais proporcionaram, surgiu a necessidade de realizar classificações de objetos em tempo real. Isto surgiu com a arquitetura *R-CNN*, em que tinha o propósito de, além de classificar múltiplos objetos em uma imagem, localizá-los em caixas delimitadoras (*bounding box*) (GIRSHICK et al., 2014).

Porém não é possível operar esta predição em tempo real com essa arquitetura devido à lentidão para realizar o treinamento em apenas uma imagem, durando em torno de 47s, por conta que a arquitetura é formada por três redes diferentes.

Alicerçado em realizar uma previsão em tempo real, surgiram diversas arquiteturas inspiradas na *R-CNN*, a *Fast R-CNN* (GIRSHICK, 2015) e a *Faster R-CNN* (REN, 2016), com a última da família realizando predições em 0.2 segundo (200 milissegundos). Apesar do ótimo valor alcançado, estas versões de redes possuem a grande desvantagem de serem constituídas por duas ou mais redes, limitando o desempenho das predições.

Então, em 2016 surgiu a *You Only Look Once* (YOLO), uma arquitetura capaz de realizar predições em tempo real sendo constituída por apenas uma rede neural (REDMON et al., 2016). A partir daí, outras arquiteturas inspiradas na YOLO surgiram, formando uma família desta arquitetura. Um exemplo de aplicação da YOLO pode ser visto na Figura 10.

Figura 10 - Exemplo de detecção de objetos da arquitetura YOLO.



Fonte: Disponível em <<https://pjreddie.com/darknet/yolo/>>. Acesso em: 22 de set. de 2021.

4. METODOLOGIA

O presente trabalho se propõe em realizar detecção e classificação de EPIs em tempo real, tomando em conta os EPIs considerados como os mais relevantes para se utilizar em laboratório químicos.

Para que isto seja possível, é necessário a coleta de imagens pertinentes ao contexto do trabalho, utilização de modelos com capacidade de detectar objetos em uma grandeza de tempo consideravelmente pequena e, então, realizar testes e inferências com os modelos para se observar o que tem a melhor performance.

4.1 Coleta de imagens

A etapa de coleta de dados é extremamente crítica em qualquer processo relacionado à utilização da inteligência artificial, pois é a partir dos dados que um modelo irá ter a capacidade de efetuar predições significativas. Há algumas maneiras de se obter o dado, seja a partir de um banco de dados ou através de APIs.

Para o trabalho em questão, foi utilizada a técnica de *web scraping* e *web crawling*. A primeira consiste no processo de extrair dados estruturados provenientes da web automaticamente (ZYTE, 2021). Ou seja, a extração do código fonte de uma página, e todo esqueleto composto por uma página web tradicional, HTML, CSS e *Javascript* (linguagens que compõem uma página web), possibilitando coletar informação da página em questão.

A segunda técnica consiste em transitar por páginas webs através de uma *engine* especializada, simulando o comportamento de um humano que acessa a internet. A biblioteca *selenium* em Python disponibiliza a execução de ambas as técnicas, e com esta biblioteca foi possível coletar 3195 imagens distintas do website *iStock*, em que disponibiliza uma versão gratuita de cada imagem com a resolução de 612x408.

A escolha do site se deu por apresentar um conjunto imenso de imagens gratuitas com uma qualidade razoável, suficiente para produzir bons resultados nos classificadores de imagens. Na Figura 11 pode ser vista uma imagem de exemplo coletada via *selenium*.

Figura 11 - Exemplo de imagem coletada.



Fonte: iStock, 2021.

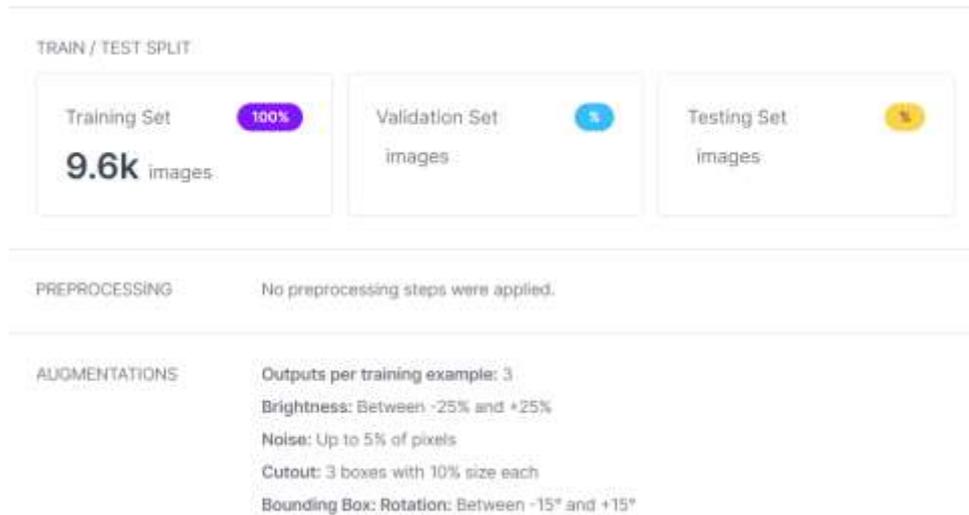
4.2 Aumentação dos dados

Para que uma rede possa generalizar bem, é necessário que a variação das imagens seja diversificada, em que possa haver conteúdos com ambientes de diferentes luminosidades, ângulos, direção dos objetos, entre outros tipos de variações (NOVAK et. al., 2018).

Entretanto, a diversificação das imagens nem sempre é algo tangível e acessível de se atingir, podendo prejudicar a predição da rede para situações em que não foram apresentadas durante o treinamento. Para isto, existe uma técnica denominada de aumento dos dados, em que esta variação é realizada artificialmente.

Existem diversas ferramentas que realizam esta técnica, sendo uma delas disponibilizada pela plataforma *Roboflow*, a qual foi utilizada neste trabalho. É possível ver na Figura 12 as técnicas utilizadas: variação de luminosidade de $\pm 25\%$, diminuição de 5% dos pixels da imagem, retirar 10% do conteúdo das *bounding boxes*, além de rotação de $\pm 15\%$ de seu conteúdo.

Vale salientar que a geração das imagens é realizada de maneira aleatória, não é necessário que uma imagem contenha todas essas variações. Após a aumento de dados, a quantidade de imagens triplicou, indo para 9585.

Figura 12 - Interface da plataforma *Roboflow*.

Fonte: Autor, 2022.

4.3 Etiquetação de imagens

A etapa de coleta de imagens, em si, não traz nenhum efeito para um modelo de classificação e localização de imagem, pois é necessário realizar um pré-tratamento que faça identificar onde cada objeto está localizado na imagem, e qual objeto é este. Esta localização se dá ao desenhar um retângulo ao redor do objeto com uma ferramenta apropriada para isto. Este retângulo é comumente denominado de *bounding box*, que quando traduzido significa caixa delimitadora.

A ferramenta escolhida para realizar esta atividade foi a *labelImg*, que possibilita o formato de entrada para a arquitetura YOLO. A ferramenta utilizada gera um arquivo contendo a posição de cada objeto na imagem acompanhado de sua classe, com o mesmo nome da imagem associado e no formato .txt.

É possível observar o resultado do processo de etiquetação de uma imagem utilizando o *labelImg* na Figura 13 e o arquivo gerado na Figura 14, em que contém informações de todos os *bounding boxes*.

Figura 13 - *Bounding boxes* produzidas com a ferramenta *labelImg*



Fonte: Autor, 2021.

Figura 14 - Dados gerados pela ferramenta *labelImg*.

```
3 0.368464 0.248775 0.089869 0.120098
0 0.362745 0.164216 0.114379 0.053922
1 0.435458 0.337010 0.129085 0.203431
1 0.519608 0.270833 0.104575 0.188725
2 0.414216 0.639706 0.419935 0.647059
4 0.714869 0.746324 0.181373 0.507353
2 0.196895 0.329657 0.106209 0.316176
2 0.719771 0.319853 0.243464 0.443627
```

Fonte: Autor, 2021.

Pode ser visto na Figura 14 diversas linhas contendo 5 colunas separadas por um caractere de espaço. A Tabela 1 contém todos os dados do arquivo gerado com suas respectivas colunas. Os valores são proporcionais às dimensões da imagem associada, consequentemente, as informações geométricas das *bounding boxes* estão entre 0 e 1.

Tabela 1 - Informações contidas nos arquivos de etiquetação.

Classe	X Central	Y Central	Largura	Altura
3	0.368464	0.248775	0.08987	0.1201
0	0.362745	0.164216	0.11438	0.05392
1	0.435458	0.33701	0.12909	0.20343
1	0.519608	0.270833	0.10458	0.18873
2	0.414216	0.639706	0.41994	0.64706
4	0.714869	0.746324	0.18137	0.50735
2	0.196895	0.329657	0.10621	0.31618
2	0.719771	0.319853	0.24346	0.44363

A coluna de classe está associada a um arquivo de configuração gerais que o modelo YOLO utiliza, em que seu formato muda para cada arquitetura, porém a representação das classes permanece constante. A representação de cada classe pode ser vista na Tabela 2.

É importante salientar que a utilização das classes no modelo foi no idioma inglês, pois é uma boa prática adotar esta linguagem para nomear as classes. A tradução das classes estará em parênteses.

Tabela 2 - Representação das classes nos arquivos de etiquetação.

Classe	Representação
Óculos (glasses)	0
Luva (glove)	1
Jaleco (lab coat)	2
Máscara (mask)	3
Calça (pants)	4
Calçado (shoe)	5

4.4 Treinamento dos modelos

Para o treinamento de modelos de redes neurais, é bastante favorável uma máquina poderosa com uma boa GPU (placa gráfica), para executar cálculos de alta complexidade envolvendo matrizes, e uma alta memória RAM, para que seja possível treinar com maior a maior quantidade de dados possíveis. Estas configurações alavancam a eficiência máxima que uma rede neural pode obter. Antes de descrever o processo de treinamento em si, é necessário descrever alguns conceitos envolvidos.

4.4.1 Época

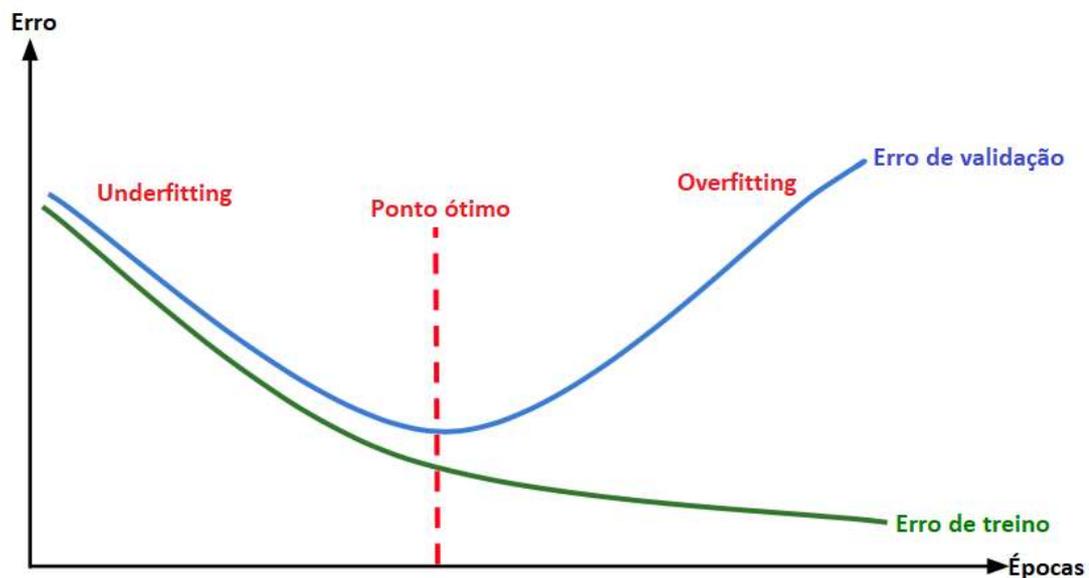
Época se trata da quantidade de etapas que um modelo de inteligência artificial realizar o processo de treinamento. Uma etapa, como já mencionada, se trata desde a entrada de dados, cálculo da predição, cálculo do erro e ajuste dos parâmetros utilizando a *backpropagation* (CHOLLET, 2017).

Quanto mais épocas, maior a capacidade do modelo melhorar sua eficiência e performance. Porém é preciso muito cuidado à quantidade de etapas, pois se o treinamento for muito longo, pode não ocorrer o aprendizado sobre os dados, e sim um fenômeno que se assemelha à memorização.

O nome deste fenômeno é *overfitting*, em que o modelo não generaliza bem para dados que não foram utilizados no treinamento. Esta ocorrência pode ser visualizada em um gráfico, em que se analisa o valor do erro ao decorrer das épocas nos dados de treinamento, e em dados de validação, que é um conjunto de dados não utilizados no treinamento (ELGENDY, 2020).

Se a curva de erro para os dados de treinamento continuar decrescendo, e a mesma curva para os dados de validação começar a subir, significa que está ocorrendo *overfitting*. Um exemplo hipotético desta ocorrência pode ser visto na Figura 15.

Figura 15 - Demonstração do fenômeno *overfitting*.



Fonte: Autor, 2022.

O fenômeno contrário também está acontecendo, o *underfitting*. Porém, é nada menos que tempo insuficiente de treinamento, gerando uma baixa capacidade de generalização do modelo. Vale ressaltar que o gráfico da figura acima é uma demonstração hipotética, nos casos reais, o erro fica estacionário durante um tempo, e o erro de validação aumenta em uma baixa taxa. Isto será visto mais adiante.

4.4.2 *Batch*

Treinar com todos os dados do conjunto demanda um alto poder computacional, como alta memória RAM, e o treinamento pode durar longos períodos de tempo, pois utiliza todos os dados nos cálculos matriciais envolvidos (GÉRON, 2019). Para resolver isto, existe a técnica de treinamento em *batch*.

Batch é uma parcela do conjunto total de dado, geralmente com o tamanho de 2^n , em que os dados de entrada são escolhidos de maneira estocástica, ou seja, de maneira aleatória. Os dados escolhidos mudam a cada época (GÉRON, 2019).

A introdução de uma quantidade menor de dados envolvido em cada etapa auxilia o modelo encontrar o mínimo global da *loss function*, pois adiciona uma aleatoriedade que evita que o valor do erro estacione em um mínimo local, o que ocorre com frequência quando se usa o conjunto de dados inteiro. Porém, uma desvantagem dessa aleatoriedade é que o valor do erro dificilmente será exatamente o mínimo global, e sim um valor próximo (GÉRON, 2019).

4.5 Métricas de performance

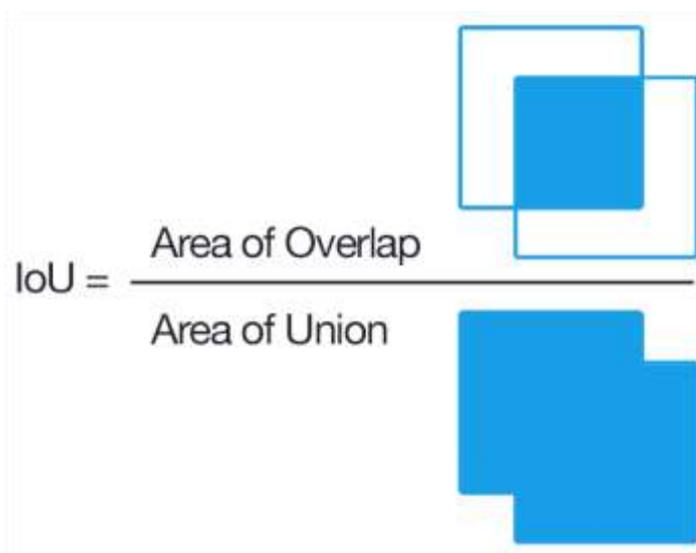
Usualmente, a acurácia de um modelo de inteligência artificial é uma boa métrica para indicar sua performance. Porém nem sempre o caso é este, como por exemplo para detectores em tempo real. Algumas métricas analisam o quão é a detecção de um modelo, sendo elas: *Mean Average Precision* (mAP) e *Frames Per Second* (FPS).

4.5.1 *Intersection over Union* (IoU)

Para entender o mAP, é necessário entender o *Intersection over Union* (IoU) – sua tradução é Interseção sob União – que nada mais mede a área de interseção entre a *bounding box* (B_{real}) verdadeira e a *bounding box* predita ($B_{predita}$) pelo modelo, sobre a união das duas *bounding boxes* (ELGENDY, 2019). É possível observar este cálculo na equação (3), e ver uma representação visual na Figura 16.

$$IoU = \frac{B_{real} \cap B_{predita}}{B_{real} \cup B_{predita}} \quad (3)$$

Figura 16 - Representação gráfica do cálculo do IoU.

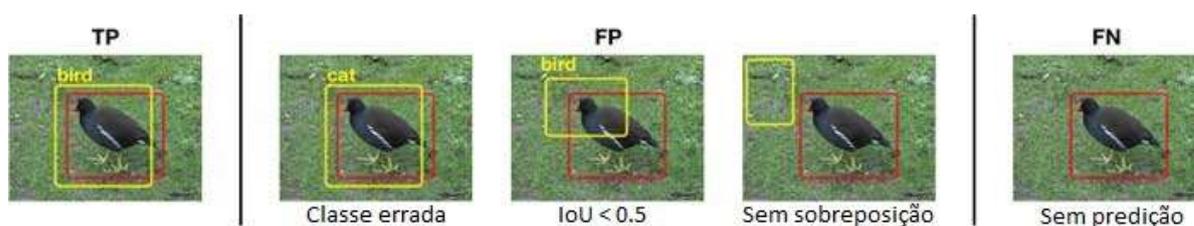


Fonte: Disponível em <<https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>>. Acesso em 18 de fev. de 2022.

Com essa métrica, é possível determinar quais dados são verdadeiros positivos (TP) – dados que a predição condiz com a classe que deveria representar –, falsos positivos (FP) – dados que a predição não condiz com o dado correto, e afirmou ser verdadeira – e falsos negativos (FN) – dados que não houve predição onde deveria ter (ELGENDY, 2019).

Exemplos de cada situação podem ser visto na Figura 17, em os quadrados vermelhos são as *bounding boxes* reais, e o quadrados amarelos, as *bounding boxes* preditas.

Figura 17 - Exemplos de TN, FP e FN.



Fonte: Disponível em <<https://machinethink.net/blog/object-detection/>>. Acesso em 19 de fev. de 2022.

O intervalo de seu valor é entre 0 e 1, quanto mais próximo de 1, mais sobrepostas as *bounding boxes* estão, assim, representando uma predição mais correta. Geralmente, o limite do IoU para determinar se a previsão foi verdadeira é de 0.5, porém há avaliações com IoU com valores maiores.

4.5.2 Mean Average Precision (mAP)

Há mais um conceito envolvido no cálculo do mAP, o cálculo da curva de Precisão-Recall. Primeiramente, deve-se entender o que significa Precisão e Recall no contexto da inteligência artificial.

Precisão procura responder a seguinte pergunta: Qual é a proporção de identificações positivas foi realmente correta (GOOGLE, 2020a)? Matematicamente, a pergunta é traduzida no cálculo mostrado na equação 4.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Sendo TP, a quantidade dos verdadeiros positivos e FP, a quantidade dos falsos positivos.

Seguindo a mesma lógica, *Recall* procura responder à pergunta a seguir: Qual proporção dos verdadeiros positivos foram identificados corretos (GOOGLE, 2020a)? O cálculo para isto pode ser visto na equação 5. FN é a quantidade de falsos negativos.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

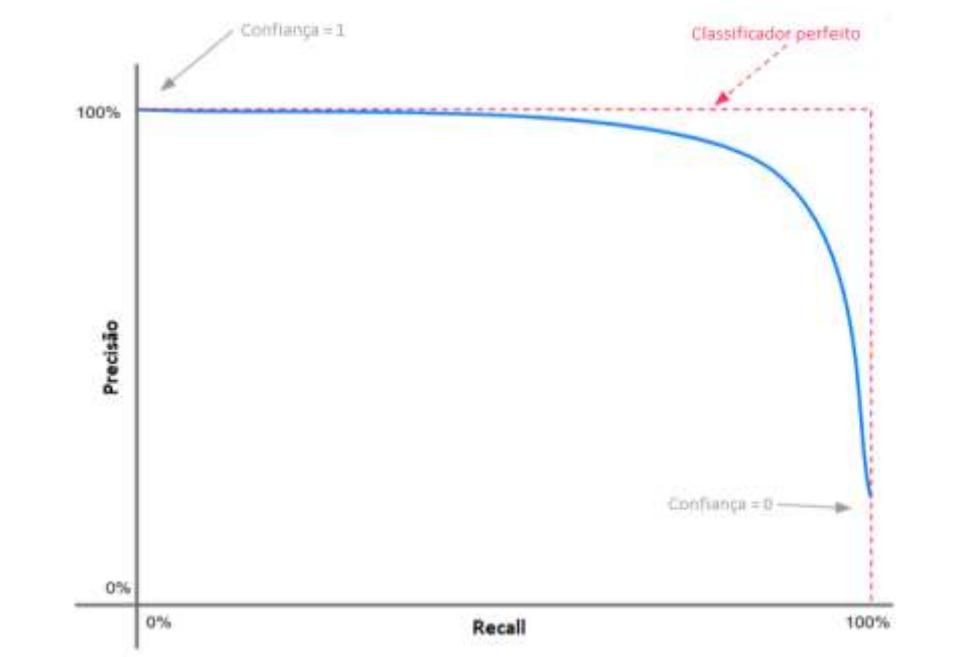
Ambas métricas devem ser analisadas para ser observar a qualidade de predição de um modelo. Porém estas métricas, por muitas vezes, são inversamente proporcionais. E isto irá depender do *confidence threshold*.

Confidence threshold (limiar de confiança) é um parâmetro ajustável do modelo, que significará o quão confiante o modelo está que a predição é real, e seu valor, geralmente, varia entre 0 e 1 (GOOGLE, 2020b). No contexto de detecção em tempo real, o parâmetro determinará a confiança de que uma *bounding box* predita é uma predição real.

Mesmo tendo um conceito similar ao do IoU, ambos diferem na prática. O *confidence threshold* determinará se uma *bounding box* deverá existir ou não, ou seja, se haverá predição ou não, e o IoU, que ocorrerá quando a *bounding box* predita já existir, determina se a predição é verdadeira ou não.

A curva de Precisão-*Recall* é o valor de cada métrica para cada valor do parâmetro de confiança. Um exemplo hipotético desta curva pode ser visto na Figura 18.

Figura 18 - Exemplo de uma curva de Precisão-Recall.



Fonte: Autor, 2022.

Com a curva de Precisão-Recall é possível calcular a precisão média (AP) de uma determinada classe, que é a área abaixo da curva.

O *Mean Average Precision* (mAP) é nada menos que a média da precisão média de cada classe, ou seja, a média da área abaixo da curva para cada classe existente no modelo (ELGENDY, 2019). Todo o processo do cálculo do mAP é realizado automaticamente pelas bibliotecas de *Machine Learning* em *Python*.

Para as avaliações dos modelos, geralmente há o mAP com o IoU de 0.5, sendo representado como $mAP@0.5$, e também uma média do mAP entre 0.5 e 0.95, com o incremento de 0.05 entre cada IoU, sendo representado como $mAP@0.5:0.95$.

4.5.3 *Frames Per Second* (FPS)

Essa métrica é utilizada para medir a velocidade de detecção do modelo. FPS é a quantidade de quadros por segundo, visto que um vídeo é composto uma quantidade x de imagens por segundo (ELGENDY, 2019).

Quando este número está associado a um detector de objetos é para demonstrar a quantos FPS o sistema conseguiu atingir $x\%$ de mAP. Quanto maior esse valor, melhor será o detector, pois significa que o modelo consegue trabalhar com mais quadros por segundos, em outras palavras, trabalhar com mais informações.

4.6 Ambiente de Treinamento

Para realizar o treinamento dos modelos de detecção de objetos em tempo real foi utilizado o ambiente do *Google Colab*, em que é disponibilizado pela *Google*, sendo um ambiente possível executar códigos de diversas linguagens, como *Python*, *R*, *Julia*, entre outros, divididos em blocos.

Além disso, apresenta uma grande gama de bibliotecas que auxiliam na execução de modelos de inteligência artificial. É possível observar alguns blocos de código do *Google Colab* na Figura 19. O primeiro bloco abaixo adiciona linhas de códigos em um arquivo do modelo *Scaled-YOLOv4*, para corrigir um erro de tipo de dado, e o segundo bloco é para copiar os resultados do treinamento para o drive pessoal.

Figura 19 - Blocos de códigos do *Google Colab*.

```

1 line1 = f'{" " * 4}if isinstance(output, list):\n'
2 line2 = f'{" " * 8}output = torch.stack(output)\n\n'
3 match_string = 'def output_to_target(output, width, height):'
4 offset = 1
5
6 with open('/content/ScaledYOLOv4/utils/general.py', 'r+') as f:
7     contents = f.readlines()
8
9     for index, line in enumerate(contents):
10        if match_string in line:
11            contents.insert(index + offset + 1, line1)
12            contents.insert(index + offset + 2, line2)
13            break
14        f.seek(0)
15        f.writelines(contents)

[3] 1 import os
2
3 def copy_info_to_drive(drive_path):
4     run_path = '/content/ScaledYOLOv4/runs'
5
6     last_run = max(next(os.walk(run_path))[1])
7     files = os.listdir(f'{run_path}/{last_run}')
8     files = [f'{run_path}/{last_run}/{f}' for f in files if f is not 'weights']
9
10    for f in files:
11        %cp {f} {drive_path}
12
13    %cp {run_path}/{last_run}/weights/best.pt {drive_path}
14    %cp {run_path}/{last_run}/weights/last.pt {drive_path}

```

Fonte: Autor, 2022.

O *Colab* também oferece placas de vídeo (GPU) de alto poder computacional através de assinaturas de planos mais avançados. Algumas GPUs disponíveis para uso são a T4, com 8.1 TFLOPS de performance, e a V100, com 10.6 TFLOPS de performance (MICROWAY, 2019). TFLOP é a capacidade de calcular um trilhão de operações de números decimais por segundo (DIGITALTRENDS, 2021).

4.7 Modelos de Detecção

Foram utilizados 3 modelos diferentes de modelos de detecção de objetos em tempo real, em que todos são baseados nas arquiteturas originais da família *YOLO*. Todas as

arquitecturas foram treinadas pelos seus autores com o dataset padrão para avaliação de modelo de detecção de tempo real, o *MS COCO*.

4.7.1 YOLOv5

Este modelo da arquitetura é bastante controverso por não conter um artigo acadêmico demonstrando os resultados, e como chegaram nos resultados. De acordo com o autor da arquitetura, a performance do modelo tem como resultado 68.9 para $mAP@0.5$ e o tempo de inferência de 12.1 milissegundos (ms), resultando em um FPS de 82.6 (JOCHER et. al., 2021).

Apesar de não haver embasamento acadêmico, seu código fonte é um dos mais utilizados, sendo colocado como destaque por 27200 usuários (JOCHER et. al., 2021). Como comparação, a arquitetura criada pelo autor original, sendo utilizada como base para outras arquiteturas, há 22800 destaques (GITHUB, 2022b).

4.7.2 YOLOR

Nesta versão da YOLO, é proposto a unificação entre o aprendizado explícito e o aprendizado implícito para que o modelo possa realizar múltiplas tarefas, além da detecção de objetos em tempo real. O aprendizado explícito é o que está direcionado diretamente ao objetivo do experimento, ou seja, ao intuito principal do modelo. O aprendizado implícito é que não está relacionado ao objetivo (WANG, YEH e LIAO, 2021).

A integração entre esses dois aprendizados está ao treinar do modo convencional, com a incorporação da modelagem do conhecimento implícito, como vetor, adicionada no cálculo do treinamento (WANG, YEH e LIAO, 2021). A performance da rede foi de 73.3 para $mAP@0.5$, com o FPS 30.

4.7.3 YOLOX

Esta arquitetura utiliza a *YOLOv3* como ponto de partida, para poder realizar melhoria em procedimentos específicos de detecção em tempo real, como retirar o gradeamento colocado nas imagens - em que cada bloco da grade possui diversas *bounding boxes* em potencial de diferentes tamanhos – entre outros processos mais onerosos ao treinamento e predição (GE et. al., 2021). A performance desta arquitetura foi 69.6 para $mAP@0.5$, com o FPS 57.8.

4.8 Tempo de treinamento

Por consequência de cada modelo possuir arquiteturas e códigos fontes diferentes, os tempos de treinamento são diferentes para cada versão, apesar dos dados serem os mesmos. Na Tabela 3, é possível observar o tempo que ocorreu o treinamento de cada modelo e a quantidade de épocas.

A diferença de épocas se dá a limitação de acesso à plataforma *Colab*, em que a assinatura adquirida permite apenas 24h de treinamento. A execução desta etapa para modelos mais demandantes, como *YOLOR* e *YOLOX*, limita-se a quantidade de épocas menores.

Tabela 3 - Informações de treinamento de cada modelo.

Modelo	Tempo de treinamento (h)	Épocas
YOLOv5	4:58	100
YOLOR	17:45	50
YOLOX	12:32	50

5. RESULTADOS

Foi avaliado a performance das arquiteturas utilizadas para o conjunto de dados alvo, a detecção de EPs. Mediu-se a precisão média (mAP) geral, em que foi levado em consideração todos os dados, obtendo-se uma visão macro do desempenho. Também houve a análise em uma visão micro, em que o mAP foi calculado para cada classe envolvida.

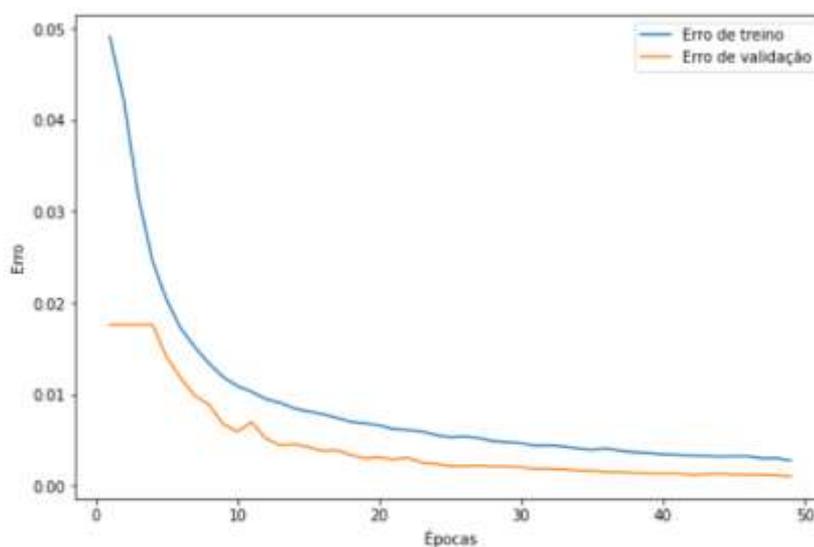
5.1 Desempenho de treinamento

Cada modelo possui uma metodologia de usabilidade, diferindo em comandos para ativar o treinamento, pois isto irá depender em como cada autor confeccionou o código para as atividades que envolvem o modelo. Isto também impacta nos arquivos gerados após o treinamento, já que estes são gerados via código.

Todos os modelos geram arquivos com a melhor versão da rede neural, além de dados contendo métricas de treinamento. Porém, estas métricas divergem entre cada versão. A *YOLOX* não disponibiliza valores do erro ao longo das épocas, apenas o mAP. As outras versões oferecem estes dados via arquivos.

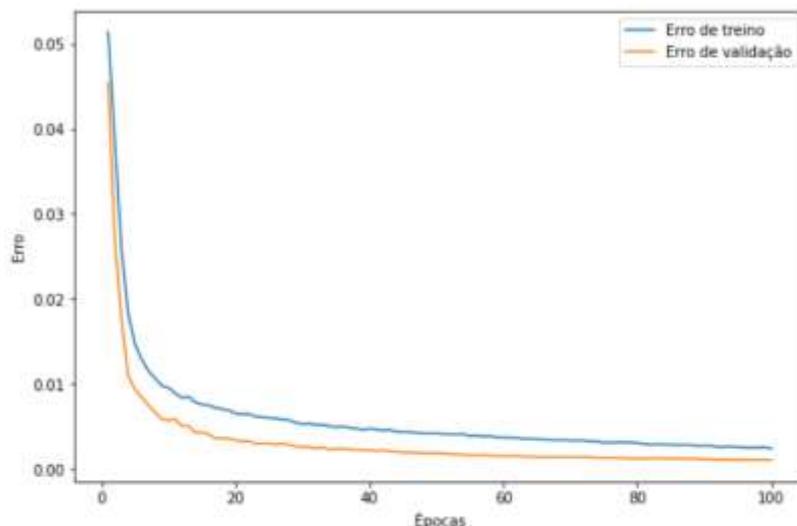
Na Figura 20 e na Figura 21 possuem os valores de erro das classes para *YOLOv5* e *YOLOv7* ao longo das épocas, respectivamente. Com estes dados pode-se observar que não houve *overfitting* de nenhum modelo, já que os valores de erro são similares tanto para os dados de treino, quanto para os dados de validação. Isso demonstra um forte indício de que os modelos são capazes de generalizar suas previsões para qualquer dado com o intuito de detectar EPs, e obtiveram bons resultados no treinamento.

Figura 20 - Valores de erros de treino e validação ao longo das épocas da *YOLOv5*.



Fonte: Autor, 2022.

Figura 21 - Valores de erros de treino e validação ao longo das épocas da *YOLOR*.



Fonte: Autor, 2022.

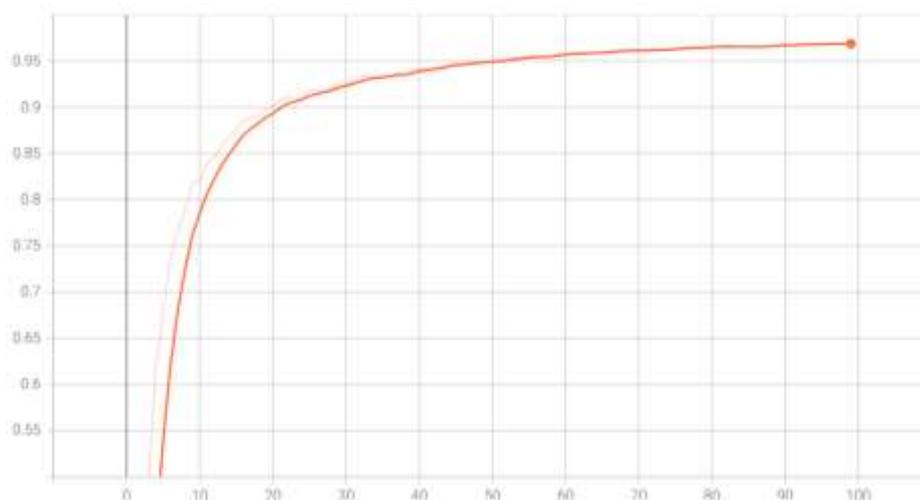
Já para a *YOLOX* será necessária uma análise mais criteriosa com seu desempenho nos dados de teste, já que não possível observar a variação dos erros.

5.2 Desempenho de teste

O desempenho para dados de teste consiste em analisar duas métricas: mAP, pois é possível entender a qualidade de detecção da rede, já que tem como objetivo metrificar a média das precisões das predições; FPS, já que a velocidade de inferência precisa ocorrer de maneira rápida para poder haver detecção em tempo real.

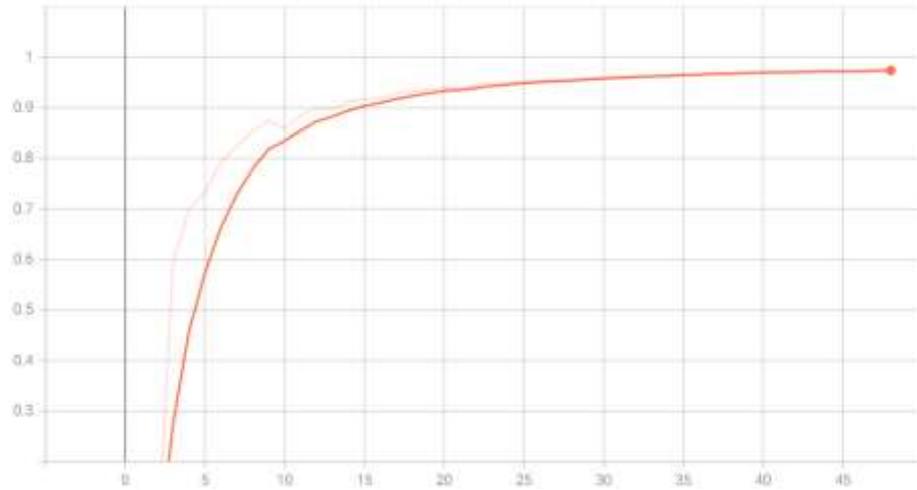
Todos os modelos obtiveram um excelente resultado, visto que todos ficaram acima de 90 para o mAP com o IoU de 0.5, como pode ser visto da Figura 22 até a Figura 24. Também se percebe que a *YOLOR* teve o melhor resultado.

Figura 22 - mAP@0.5 ao longo das épocas para YOLOv5.



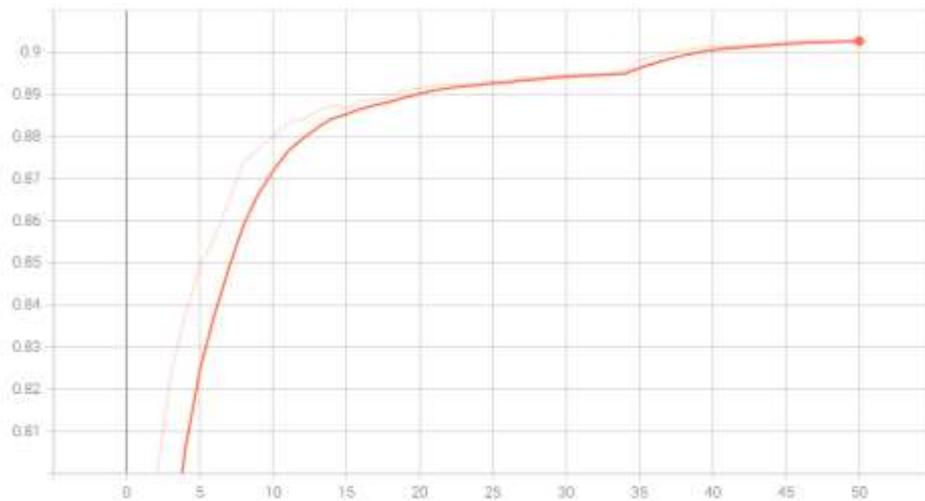
Fonte: Autor, 2022.

Figura 23 - mAP@0.5 ao longo das épocas para YOLOR.



Fonte: Autor, 2022.

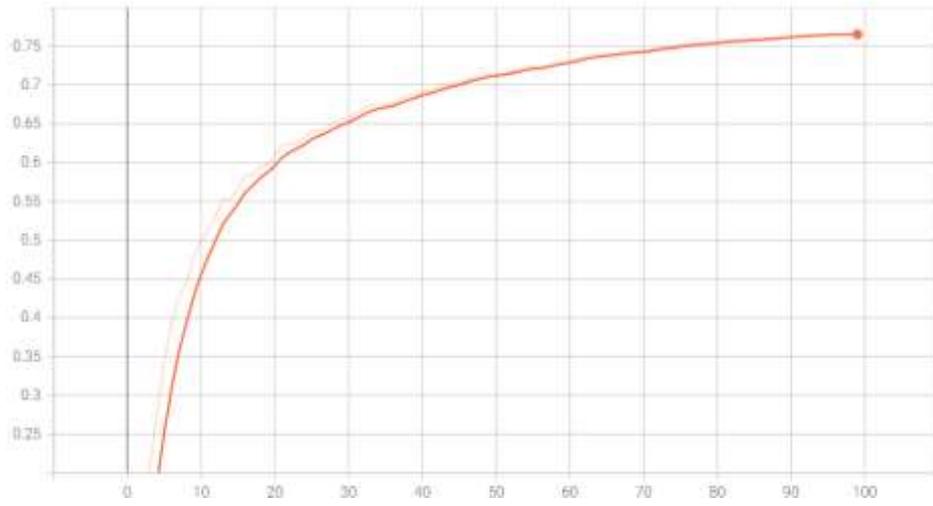
Figura 24 - mAP@0.5 ao longo das épocas para YOLOR.



Fonte: Autor, 2022.

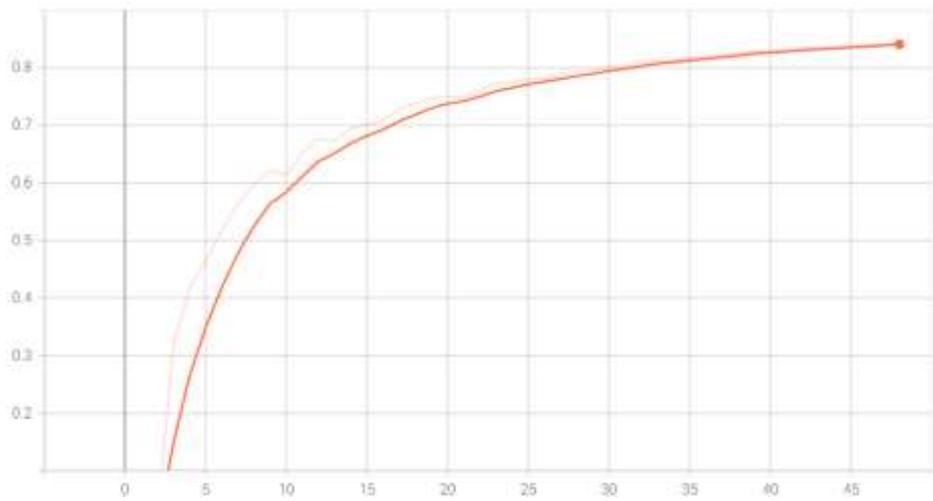
Para o mAP com o IoU variando entre 0.5 e 0.95, a melhor rede também foi a *YOLOR*, e com uma grande diferença entre a segunda melhor, como pode ser visto da Figura 25 a até a Figura 27.

Figura 25 - mAP@0.5:0.95 ao longo das épocas para YOLOv5.



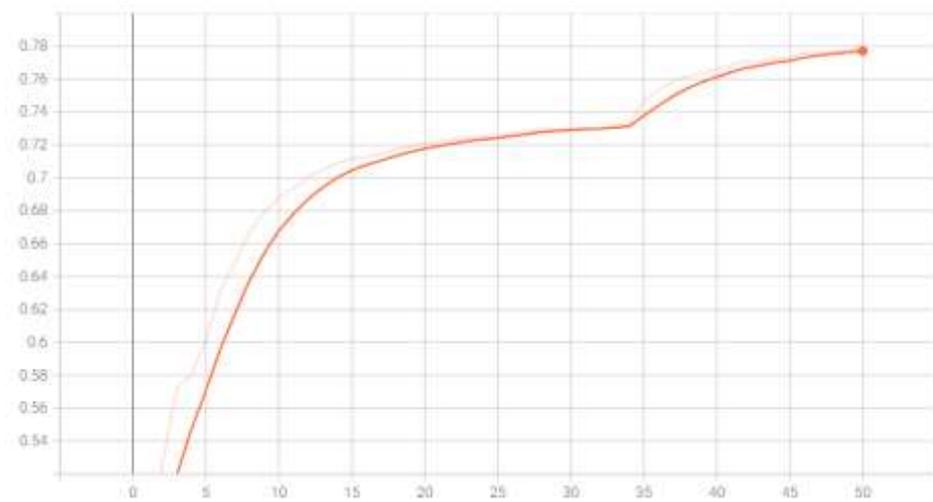
Fonte: Autor, 2022.

Figura 26 - mAP@0.5:0.95 ao longo das épocas para YOLOR.



Fonte: Autor, 2022.

Figura 27 - mAP@0.5:0.95 ao longo das épocas para YOLOX.



Em todos os gráficos apresentados acima, nota-se uma crescente dos mAPs, conseqüentemente há como melhorar o desempenho de todas as redes ao acrescentar a quantidade de épocas no treinamento, e no caso das redes *YOLOv5* e *YOLOR*, esta melhoria com baixo risco de *overfitting*. Não é possível constatar isto para a *YOLOX* devido à falta de dados.

A Tabela 4 demonstra o desempenho macro de todos os modelos, além de seus tempos de inferência, com a *YOLOv5* tendo o menor tempo de inferência, com o valor de 3.3ms, resultando em um FPS de 303.

Tabela 4 - Métricas de treino dos modelos.

Modelo	mAP@0.5	mAP@0.5:0.95	FPS
YOLOv5	0.983	0.806	303.0
YOLOR	0.984	0.876	45.7
YOLOX	0.920	0.807	58.9

5.2.1 mAP das Classes

O desempenho de cada classe foi bastante similar entre os modelos, em que a classe Jaleco, geralmente, possui valores maiores de mAP, seguido pela classe Calça, apenas na *YOLOX* a classe Calça obteve o maior mAP. A classe Calçado obteve o menor valor de mAP, tanto para IoU 0.5 e IoU variando entre 0.5 e 0.95 em todos os modelos.

A partir da Tabela 5 até a Tabela 7, percebe-se um comportamento consistente entre as classes do dataset. Objetos menores, como calçados e luva possuem uma maior dificuldade para que a predição seja correta do que objetos maiores, como jaleco e calça. Além do tamanho dos objetos em questão, há também uma maior possibilidade de os objetos menores estarem obstruídos por outros objetos dos ambientes.

Por fim, durante a etiquetagem dos dados, foi notado nas imagens que os calçados é o objeto com maior ocorrência de falta de nitidez e clareza, outro fator que prejudica no momento da predição.

O modelo *YOLOX* não demonstrou os *mAP@0.5:0.95* de cada classe, apenas o valor geral.

Tabela 5 - mAP das classes da *YOLOv5*.

Classe	mAP@0.5	mAP@0.5:0.95
Óculos	0.992	0.801
Luva	0.983	0.796
Jaleco	0.994	0.906
Máscara	0.994	0.813
Calça	0.994	0.881
Calçado	0.939	0.638

Tabela 6 - mAP das classes da *YOLOv4*.

Classe	mAP@0.5	mAP@0.5:0.95
Óculos	0.992	0.901
Luva	0.976	0.854
Jaleco	0.994	0.960
Máscara	0.992	0.868
Calça	0.995	0.941
Calçado	0.954	0.729

Tabela 7 - mAP das classes da *YOLOv3*.

Classe	mAP@0.5	mAP@0.5:0.95
Óculos	0.907	-
Luva	0.909	-
Jaleco	0.909	-
Máscara	0.909	-
Calça	0.983	-
Calçado	0.902	-

6. CONCLUSÃO

É possível concluir que o trabalho teve êxito ao criar modelos de detecção de EPIs em tempo real, pois em todas redes treinadas tiveram uma precisão média acima de 90% (0.9), demonstrando que as redes conseguiram abstrair e generalizar os EPI mais utilizados em laboratórios químicos, no geral.

A rede com melhor desempenho de qualidade foi a YOLOR com um $mAP@0.5$ de 0.984, e um $mAP@0.5:0.95$ de 0.876, conseqüentemente suas predições geralmente possuem uma grande área de interseção com a realidade. Porém, vale notar que a rede que apresenta uma detecção mais próxima de tempo real é a YOLOv5, pois necessita apenas de 3.3ms (0.003s) para realizar predição, além de ter obtido a segunda melhor performance, com o $mAP@0.5$ de 0.983, além de precisar apenas de 4:58h de 100 épocas de treinamento, enquanto a YOLOR precisa de 17:45h de 50 épocas de treinamento.

A realização de um sistema para detectar EPIs no LASSOP não foi possível, pois o laboratório não possui câmeras para que ocorra uma detecção em tempo real, e, então, avaliar a incidência de acidentes.

REFERÊNCIAS

- ALBON, C. **Python Machine Learning Cookbook**. 1º Ed. Massachusetts: O'Reilly Media, 2018.
- ANTONIAZZI, C. T. de D. et al. Riscos à segurança do trabalhador em cooperativa agrícola. **Disciplinarum Scientia**, Santa Maria, v. 10, n. 1, p-93-103, 2009.
- BARSAÑO, P. R.; BARBOSA, R. P. **Segurança do trabalho**: Guia prático e didático. 2º Ed. Barueri: Érica, 2018.
- BAYER**, AI in Pharma, 2020. Disponível em: <<https://pharma.bayer.com/ai-pharma>>. Acesso em 27 de abr. de 2021.
- BRASIL. Ministério da Previdência Social. **Conceito, definições e caracterização do acidente do trabalho, prestações e procedimentos**. 1999. Disponível em: <http://www1.previdencia.gov.br/pg_secundarias/paginas_perfis/perfil_Empregador_10_04-A5.asp>. Acesso em: 28 abr. 2021.
- BRASIL. **Máquinas e acidentes de trabalho**. Brasília: MTE/SIT; MPAS. 2001a. (Coleção Previdência Social), v. 13.
- BRASIL. Ministério da Saúde. Departamento de Ações Programáticas e Estratégicas. Área Técnica de Saúde do Trabalhador. **Lesões por Esforços Repetitivos (LER) e Distúrbios Osteomusculares Relacionados ao Trabalho (DORT)**. Brasília, DF. 2001b.
- BRASIL. Ministério do Trabalho e Emprego. **Caminhos da análise de acidentes do trabalho**. Brasília, DF, 2003.
- BRASIL. Ministério do Trabalho e Emprego. **Normas Regulamentadoras NR 6**: Equipamento de Proteção Individual - EPI. Brasília, DF, 2018. Disponível em: <https://enit.trabalho.gov.br/portal/images/Arquivos_SST/SST_NR/NR-06.pdf>. Acesso em: 28 de abr. 2021.
- BRASIL. Ministério do Trabalho e Emprego. **Normas Regulamentadoras NR 9**: Avaliação e Controle das Exposições Ocupacionais a Agentes Físicos, Químicos e Biológicos. Brasília, DF, 2020. Disponível em: <<https://www.gov.br/trabalho-e-previdencia/pt-br/composicao/orgaos-especificos/secretaria-de-trabalho/inspecao/seguranca-e-saude-no-trabalho/normas-regulamentadoras/nr-09-atualizada-2020.pdf>>. Acesso em: 30 ago. 2021.
- CHOLLET, F. **Deep Learning with Python**. 1º Ed. Nova York: Manning Publications, 2017.
- CROWLEY, J. L.; CHRISTENSEN, H. I. **Vision as Process**. 1º Ed. Berlin: Springer Berlin, 1995.
- Data Science Academy. **Deep Learning Book**, 2021. Disponível em: <<https://www.deeplearningbook.com.br/>>. Acesso em: 05 de set. 2021.
- ELGENDY, M. **Deep Learning for Vision Systems**. Nova York: Manning Publications, 2020.

ESPÍRITO SANTO. **Manual de Biossegurança**. 2017.

FILHO, A. F. V. Conselho regional de química - IV região sp. **Manual de segurança em laboratório químico**, 2008. Disponível em:

<https://iqm.unicamp.br/sites/default/files/manual_de_seguran%C3%A7a_em_laboratorio_quimico.pdf>. Acesso em: 05 de set. 2021.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. 2º Ed. Nova Jersey: Pearson, 2011.

GE. Z. et. al. **YOLOX: Exceeding YOLO Series in 2021**. Disponível em:

<<https://arxiv.org/abs/2107.08430>>. Acesso em: 02 de fev. de 2022.

GIRSHICK et al. **Rich feature hierarchies for accurate object detection and semantic segmentation**. Disponível em: <<https://arxiv.org/abs/1311.2524>>. Acesso em: 02 de fev. de 2022.

GIRSHICK, R. **Fast R-CNN**. Disponível em: <<https://arxiv.org/abs/1504.08083>>. Acesso em: 10 de fev. de 2022.

GONÇALVES, F. R. F. **Visão Computacional Aplicada a Automação de Colhedora Multifuncional de Hortícolas-Alface**. 2019. Universidade Federal do Ceará, Fortaleza, 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Massachusetts: The MIT Press, 2015.

GOOGLEa. **Classification: Precision and Recall**. 2020. Disponível em:

<<https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>>. Acesso em: 04 de abr. de 2020.

GOOGLEb. **Classification: Thresholding**. 2020. Disponível em:

<<https://developers.google.com/machine-learning/crash-course/classification/thresholding>>. Acesso em: 04 de abr. de 2020.

HARRINGTON, P. **Machine Learning in Action**. 1º Ed. Nova York: Manning Publications, 2012.

HOSPITAL DAS CLÍNICAS. **Guias de Boas Práticas Laboratoriais**. 2015 - Faculdade de Medicina da Universidade de São Paulo, São Paulo, 2015.

IBM, What is Artificial Intelligence?, 2020. Disponível em:

<<https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>>. Acesso em: 28 de abr. de 2021.

JOCHER, G. et al. yolov5. Disponível em: <<https://github.com/ultralytics/yolov5>>, 2021. Acessado em: 8 de jun. 2022.

JUNIOR, H. R et al. Segurança do trabalho em indústria de cerâmica. **Engenharia Ambiental**. Espírito Santo do Pinhal, v. 11, n. 1, p. 147-159, jan./jun. 2014.

KENNY, C. **What Is Web Scraping?** Zyte, 2019. Disponível:

<<https://www.zyte.com/learn/what-is-web-scraping/>>. Acesso em: 03 de fev. 2022.

KENYON, C. **NVIDIA “Turing” Tesla T4 HPC Performance Benchmarks**. Microway, 2019. Disponível em: <<https://www.microway.com/hpc-tech-tips/nvidia-turing-tesla-t4-hpc-performance-benchmarks/>>. Acesso em: 27 de mai. de 2022.

MALAGUTI, S. E. et al. Enfermeiros com cargos de chefia e medidas preventivas à exposição ocupacional: facilidades e barreiras. **Rev Esc Enferm USP**. São Paulo, v. 42, n. 3, p-496-503, 2008.

MARTINDALE, J. **What is a teraflop?**. Digital Trends, 2021. Disponível em: <https://www.digitaltrends.com/computing/what-is-a-teraflop/>. Acesso em: 30 mai. de 2022.

NILSSON, N. J. **Artificial Intelligence: A New Synthesis**. Massachusetts: Morgan Kaufmann, 1997.

NOVAK et. al. **Sensitivity and Generalization in Neural Networks: An Empirical Study**. Disponível em: <<https://arxiv.org/pdf/1802.08760.pdf>>. Acesso em: 25 de mai. de 2022.

REDMON, J. et al. yolov5. Disponível em: <<https://github.com/pjreddie/darknet>>, 2015. Acessado em: 8 de jun. 2022.

REDMON, J. et al. **You Only Look Once: Unified, Real-Time Object Detection**. Disponível em: <<https://arxiv.org/pdf/1506.02640.pdf>>. Acesso em: 22 de set. 2021.

REDMON, J.; FARHADI, A. **YOLOv3: An Incremental Improvement**. Disponível em: <<https://pjreddie.com/media/files/papers/YOLOv3.pdf>>. Acesso em: 28 de abr. 2021.

REN et al. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. Disponível em: <<https://arxiv.org/abs/1506.01497>>. Acesso em: 25 de mai. de 2022.

SILVA, F. S. et al. A importância da utilização dos equipamentos de proteção individual e coletiva na prevenção de acidentes. **Revista Ambiente Acadêmico**, Cachoeiro de Itapemirim, v. 4, n. 1, jan./jun. 2018

SOUZA, A. C. S. et al. Conhecimento dos graduandos de enfermagem sobre equipamentos de proteção individual: a contribuição das instituições formadoras. **Revista Eletrônica de Enfermagem**. Goiânia, v. 10, n. 2, p. 428-437, 2008.

STHEL, J. P. V.; LOUREIRO, R. B. **A engenharia química no contexto da indústria 4.0: estudo de caso em uma usina de etanol**. Tese (Conclusão em Engenharia Química) - Escola de Engenharia, Universidade Federal Fluminense. Niterói, p. 14. 2018.

THOMPSON, N. C. et al. **The Computational Limits of Deep Learning**. Disponível em: <<https://arxiv.org/pdf/2007.05558.pdf>>. Acesso em: 28 de abr. de 2021.

TIPPLE, A. F. V. et al. Equipamentos de proteção em centros de material e esterilização: disponibilidade, uso e fatores intervenientes à adesão. **Ciência, Cuidado e Saúde**, Maringá, v. 6, n. 4, p. 441-448, out./dez. 2007.

WANG, C.; YEH, I.; LIAO, H. M. **You Only Learn One Representation: Unified Network for Multiple Tasks**. Disponível em: <<https://arxiv.org/pdf/2105.04206.pdf>>. Acesso em: 26 de mai. de 2022.

XU, Y.; ZHANG, H. **Convergence of Deep ReLU Networks**. Disponível em: <<https://arxiv.org/pdf/2107.12530.pdf>>. Acesso em: 22 de nov. de 2021.

YOSINSKI et al. **How transferable are features in deep neural networks? In: Advances in neural information processing systems**. Disponível em: <<https://arxiv.org/abs/1411.1792>> Acesso em: 20 de nov. de 2021.