

USO DE APRENDIZADO DE MÁQUINA PARA IDENTIFICAÇÃO DE CAPACETES DE PROTEÇÃO INDIVIDUAL

Deivid Wesley Pereira da Silva¹, Luiz Fernando Segato dos Santos¹, Luan Matheus Moreira¹

Instituto Federal de Mato Grosso do Sul – Aquidauana-MS

deividweslps@gmail.com, luiz.santos@ifms.edu.br, luan.moreira@ifms.edu.br

Resumo

Um fator que permite minimizar acidentes com trabalhadores da construção civil é o uso de Equipamentos de Proteção Individual (EPIs). Independentemente de as Normas Regulamentadoras estabelecerem requisitos para o uso, armazenamento e conservação, existe um problema cultural de uso que principalmente em pequenas edificações é difícil de resolver. Assim, o objetivo deste artigo foi propor o uso de CNNs para o reconhecimento de equipamentos de proteção individual. Como resultado preliminar foram implementadas duas redes neurais, Yolov5 e Resnet50 que nos permitiu o reconhecimento de capacetes EPI, utilizando o *dataset* disponibilizado pela plataforma *roboflow* que facilitou a obtenção das imagens testadas.

Palavras-chave: Capacete de proteção individual, Aprendizado de máquina, Extração de características.

Introdução

A preocupação com a segurança dos trabalhadores da construção civil ainda é extremamente relevante, pois ainda observamos canteiros de obras pouco industrializados. No Brasil, a utilização da “moldagem in loco” de materiais (e. g., concreto simples, concreto armado) é amplamente utilizada, principalmente em pequenas edificações. Assim, enquanto novos processos de canteiro de obras são estudados e desenvolvidos, tendo em vista a racionalização do segmento, os agentes sociais devem pensar em mecanismos que minimizem os acidentes no trabalho da construção civil.

Diante destas dificuldades, a utilização de métodos computacionais capazes de auxiliar na prevenção de acidentes se torna uma alternativa viável. Diversas redes neurais são utilizadas no reconhecimento de padrões e extração de características em diversas áreas do conhecimento, tais como na agricultura, na medicina, na geologia e no trânsito.

Desta forma, uma motivação para esta pesquisa foi de investigar métodos computacionais que são capazes de extrair as melhores características a serem utilizadas no reconhecimento e fiscalização do uso de capacetes de proteção individual em canteiros de obras de construção civil por meio de visão computacional com a implementação da técnica *Transfer Learning*.

Transfer Learning é a reutilização de um modelo pré-treinado em um novo problema. Isto é, vou usar uma rede neural treinada em outro conjunto de dados, geralmente maior, para resolver um novo problema. (MEAZZINI, 2019).

As redes neurais convolucionais (CNN) se utilizam de uma arquitetura especial que é particularmente bem adequada para classificar imagens. O uso dessa arquitetura torna as redes convolucionais rápidas de treinar, o que é vantajoso para trabalhar com redes profundas. (PERES, 2021).

Portanto este trabalho tem como objetivo o desenvolvimento de um método capaz de auxiliar na identificação de capacetes de proteção individual utilizando CNNs como a Resnet50 e yolov5 com *transfer learning*. Foram utilizadas duas redes neurais para fins de comparação e teste, bem como foram feitos testes utilizando uma base de imagens disponibilizada gratuitamente pela plataforma *Roboflow*.

Metodologia

Para o desenvolvimento deste projeto optamos por utilizar duas redes neurais para sabermos qual rede obteria os melhores resultados, estas redes neurais foram: Yolov5 e Resnet50, ambas redes convolucionais, pois estávamos trabalhando com imagens e redes convolucionais são mais adequadas para reconhecimento de objetos que é o nosso caso. O *dataset* foi obtido através de uma ferramenta chamada *Roboflow* que disponibiliza imagens de capacetes EPI gratuitamente para o treinamento de redes neurais. Esta ferramenta trabalha com anotações que é uma demarcação na imagem do objeto desejado, as demarcações utilizam de elementos gráficos e depois são transformadas em um arquivo no formato xml, contendo as informações de onde o objeto desejado está na imagem. Este procedimento foi adotado para melhor desempenho da rede neural.

Image annotation é o processo de rotular imagens de um conjunto de dados para treinar um modelo de aprendizado de máquina. Portanto, a anotação de imagem é usada para rotular os recursos que você precisa que seu sistema reconheça. (BOESCH, 2022). Essa estratégia foi incorporada no projeto a fim de que o treinamento fosse direcionado aos objetos de interesse.

Como tinham muitas imagens foi preciso reduzir a quantidade de imagens e balancear o *dataset*, por isso foi feito um *dataset* menor, o dataset original tinha 8 mil imagens já o segundo *dataset* (reduzido) tinha 200 imagens, os dois *datasets* foram

utilizados na Yolo. Mais detalhes serão vistos na seção de resultados. Na ferramenta *Roboflow*, os *datasets* foram particionados em 80% das imagens para treino e 20% para validação, também foram usadas imagens de fora do *dataset* para fins de teste, essas imagens foram pegadas aleatoriamente da internet, um exemplo pode ser visto na Figura 1.



Figura 1. imagem da internet.

Após ter feito o balanceamento do *dataset* partimos para os treinamentos, a primeira rede neural utilizada foi a Yolov5 que foi escolhida pois ela é fácil de utilizar e o código é disponibilizado gratuitamente no site [Github](https://github.com). O tamanho das imagens dos *datasets* é variado, porém é aplicado um algoritmo do próprio Roboflow que redimensiona as imagens para 416 x 416 pixels deixando todas com o mesmo tamanho. A seguir temos duas tabelas mostrando a quantidade de imagens obtidas de cada classe utilizada para o primeiro *dataset* e o segundo respectivamente.

Tabela 1. Quantidade de amostras dos objetos de interesse obtidos via *Roboflow* a partir do *dataset* 1.

Classe	Amostras
head	4.874
helmet	14.884

Tabela 2. Quantidade de amostras dos objetos de interesse obtidos via *Roboflow* a partir do *dataset* 2.

Classe	Amostras
head	483
helmet	334

Os primeiros treinos foram feitos utilizando o *dataset* com 8 mil imagens, então foram feitos 5 treinamentos variando os hiperparâmetros *Batch* e *Epoch* conforme os resultados dos treinamentos, depois foram feitos outros 5 treinos com o *dataset* de 200 imagens, para ver o desempenho que será mostrado em detalhes na seção de resultados.

Para a Resnet50 o mesmo *dataset* com apenas 200 amostras que foi utilizado na Yolov5 foi utilizado aqui pois foi visto

nos resultados que a quantidade de imagens menor foi mais vantajosa, porém foram feitas algumas mudanças, tivemos que recortar o objeto de interesse de cada imagem com a técnica *Image Annotation*, podemos ver um exemplo na Figura 2.

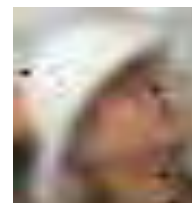


Figura 2. imagem recortada com *Image Annotation*

Na sequência dois códigos foram desenvolvidos, um para treinamento e outro para validação da rede neural.

No código de treinamento foi utilizado *Transfer Learning* que consiste em congelar as primeiras camadas da rede e adicionar novas camadas densas no final. Foi utilizada uma técnica para aumentar a quantidade de imagens utilizando um algoritmo do keras chamado *ImageDataGenerator* que faz variações nas imagens originais, neste caso a utilização desta técnica nas 200 imagens proporcionou diversas variações que resultaram em novas 1471 imagens divididas nas duas classes (cabeça e capacete) conforme consta na Tabela 3.

Tabela 3. Quantidade de imagens de capacetes e cabeças partir do *dataset* 2 aumentado para a Resnet50.

Classe	Amostras
head	814
helmet	657

Ao realizar este procedimento o *dataset* foi dividido em conjuntos de treino e validação, com 80% das imagens para treino e 20% para validação, algumas quantidades de imagens extras pegadas aleatoriamente da internet com rostos e capacetes foram utilizadas para fazer testes depois que as etapas de treino e validação estivessem concluído. Então foram feitos treinos mudando os hiperparâmetros como *Batch*, *Epoch* e as últimas camadas convolucionais da rede neural, mais detalhes destas configurações serão vistos na seção de resultados.

Resultados e Discussão

Yolov5

O primeiro treinamento (Tabela 4 e Figura 3) foi realizado utilizando um *dataset* com 8 mil imagens disponibilizadas pela ferramenta *Roboflow* e os resultados foram bons, porém apenas com as imagens do *dataset*, a rede neural não se deu bem com imagens da internet, de fora do *dataset*. O que nos interessa nestes gráficos são as métricas de validação como *val/obj_loss*, *val/box_loss*, *precision* e *mAP_0.5*, onde *mAP_0.5* é como se fosse a acurácia.

Tabela 4. Médias do treino 1

Média	Validação
box_loss	0.03598755
obj_loss	0.01434285
mAP_0.5	0.6018345
precision	0.8713465

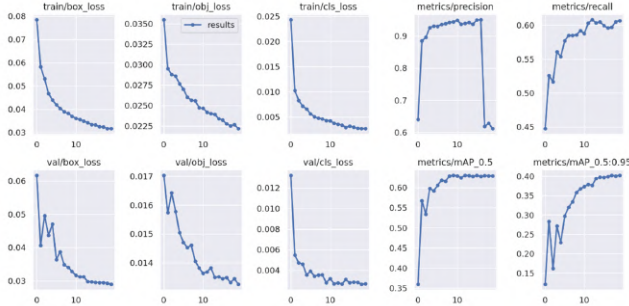


Figura 3. Gráfico do primeiro treino.

Tendo como base os gráficos resultantes do primeiro treinamento foram feitas mudanças no parâmetro *batch* que foi mudado para 10 que piorou os resultados, podendo ver que a precisão diminuiu e mAP_0.5 caiu também conforme Tabela 5 e Figura 4.

Tabela 5. Médias do segundo treino.

Média	Validação
box_loss	0.03521415
obj_loss	0.0141386
mAP_0.5	0.3371789
precision	0.8667895

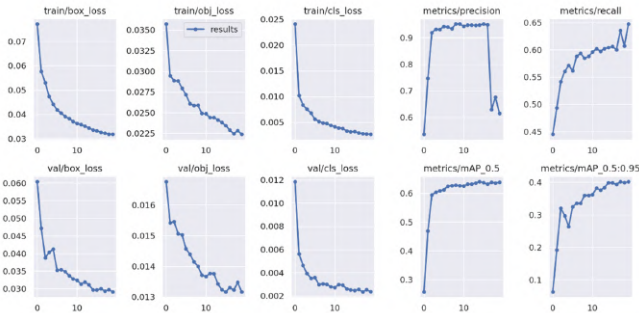


Figura 4. Gráfico do segundo treino.

E por fim no treinamento 3 foi mudado o parâmetro *epoch* para 25 e podemos perceber que no gráfico (Figura5) houve uma queda da precisão no final do treinamento, porém podemos ver na Tabela 6 que a média mAP_0.5 subiu novamente.

Tabela 6. Médias do terceiro treino.

Média	Validação
box_loss	0.0342896
obj_loss	0.01407392
mAP_0.5	0.6067144

precision	0.8049192
-----------	-----------

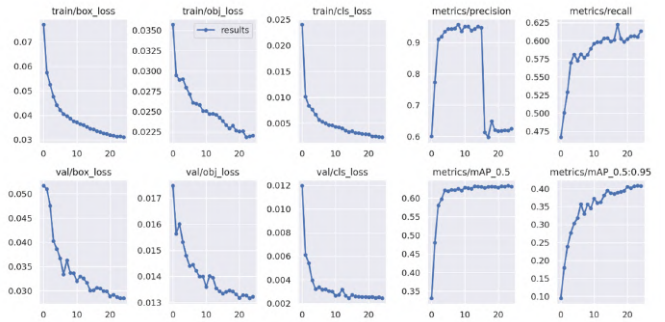


Figura 5. Gráfico terceiro treino.

Depois destes treinamentos foi feito o balanceamento do *dataset* e foram feitos os treinamentos novamente, porém dessa vez com a mesma quantidade de imagens de capacetes e sem capacetes e com apenas 200 imagens. Neste novo treino os parâmetros *Epoch:50* e *Batch:10* foram alterados. Podemos observar na Tabela 7, bem como na Figura 6, que a precisão foi baixa em relação aos treinos com um *dataset* maior.

Tabela 7. Médias do primeiro treino.

Média	Validação
box_loss	0.05751948
obj_loss	0.0387524
mAP_0.5	0.539739274
precision	0.55142415

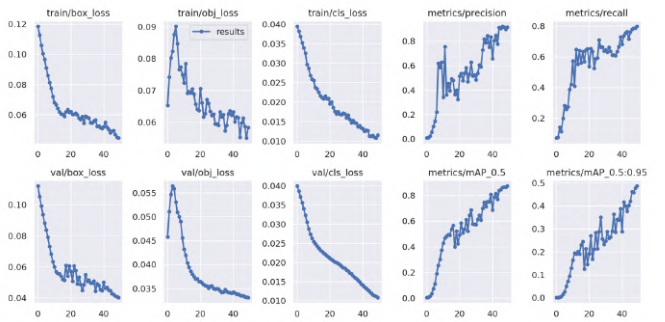


Figura 6. Gráfico do primeiro treino.

No próximo treino o parâmetro *batch* foi alterado para 7 para ver se teria alguma mudança perceptível, porém podemos observar que a mudança foi mínima (Tabela 8), mas podemos ver que os gráficos (Figura 7) mudaram, mas nada muito grande.

Tabela 8. Média do segundo treino.

Média	Validação
box_loss	0.05724582
obj_loss	0.03763886
mAP_0.5	0.553368636
precision	0.537609728

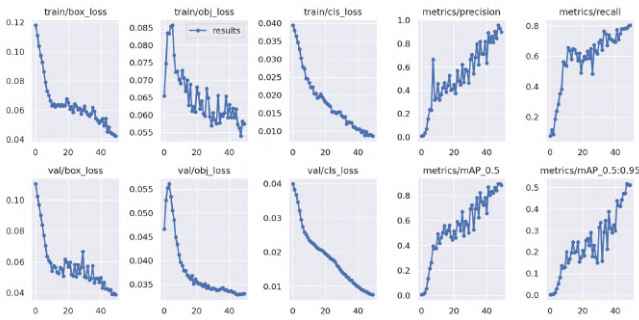


Figura 7. Gráfico do segundo treino.

Em outro treino foi mantido o parâmetro *Bach* utilizado no último treino e mudado o parâmetro *Epoch*:100. aqui podemos observar que no gráfico (Figura 8) a precisão continuou subindo e os parâmetros *loss* caíram, como pode ser visto na Tabela 9 que as médias precisão e *mAP_0.5* subiram o que é ótimo, e as médias *box_loss* e *obj_loss* caíram significativamente.

Tabela 9. Média do terceiro treino.

Média	Validação
<i>box_loss</i>	0.04903325
<i>obj_loss</i>	0.03584054
<i>mAP_0.5</i>	0.719177198
<i>precision</i>	0.721360466

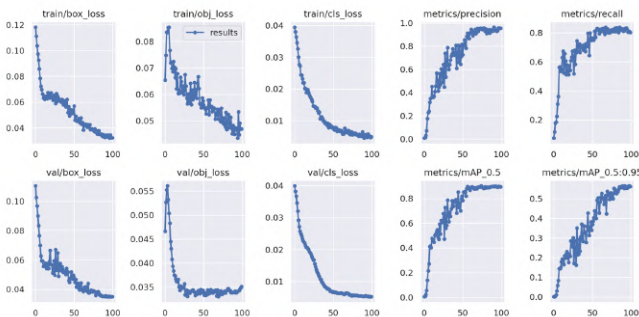


Figura 8. Gráfico do terceiro treino.

No penúltimo treino os dois parâmetros *batch* e *epoch* foram alterados respectivamente para 5 e 50, e foi observado que o desempenho caiu como apresentado na Tabela 10 e Figura 9. Podemos ver que a precisão caiu juntamente com *mAP_0.5*.

Tabela 10. Média do quarto treino.

Média	Validação
<i>box_loss</i>	0.056
<i>obj_loss</i>	0.037
<i>mAP_0.5</i>	0.6
<i>precision</i>	0.582

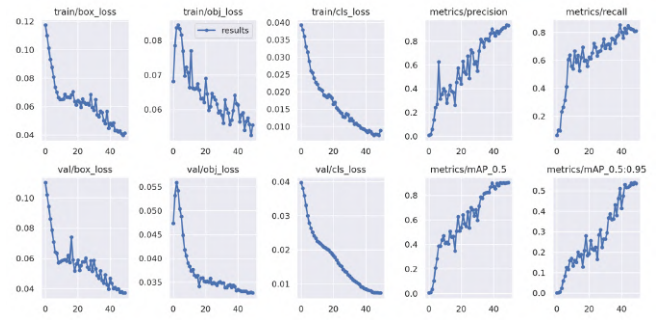


Figura 9. Gráfico do quarto treino.

No último treino (Tabela 11) o parâmetro *Epoch* foi alterado para 100 resultando em uma melhora da rede neural, podemos ver que as médias todas melhoraram, podemos observar nos gráficos essa mudança conforme a Figura 10.

Tabela 11. Média do quinto treino.

Média	Validação
<i>box_loss</i>	0.048
<i>obj_loss</i>	0.035
<i>mAP_0.5</i>	0.74
<i>precision</i>	0.736

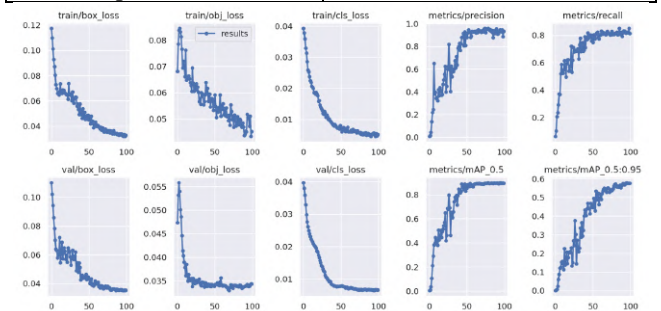


Figura 10. Gráfico do quinto treino.

Na Figura 11 abaixo é possível observar um exemplo do resultado do reconhecimento das cabeças pelo modelo do último treino a partir da imagem contida na Figura 1.



Figura 11. melhor resultado com imagem da internet.

Resnet50

Para o primeiro treino (Tabela 12 e Figura 12) foi adicionado três camadas densa, a primeira com 2048 neurônios a segunda com 1024 e a terceira com 512, o *batch* utilizado foi 20 e o *Epoch* 50.

Tabela 12. Média do primeiro treino.

Média	Validação
Accuracy	0.978
Loss	0.128
F1_score	0.974
Precision	0.990

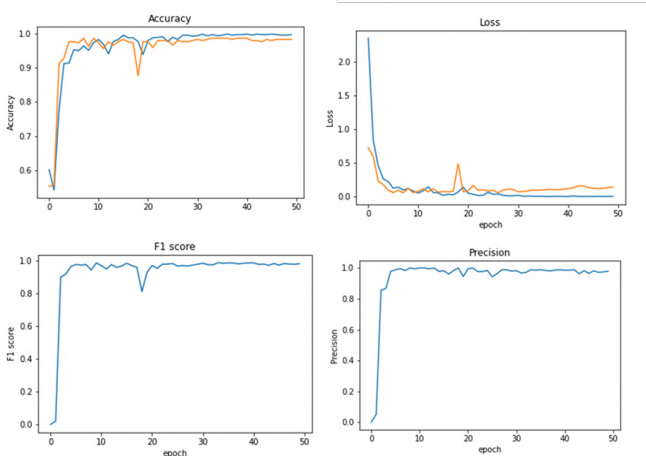


Figura 12. Gráfico do primeiro treino.

O segundo treino foi feito adicionando 4 camadas densas, as três primeiras camadas com 1024 neurônios e a última com 512, o *batch* utilizado foi 20 e o *Epoch* 30, conforme pode ser observado na Tabela 13 e Figura 13.

Tabela 13. Média do segundo treino.

Média	Validação
Accuracy	0.937
Loss	0.189
F1_score	0.897
Precision	0.901

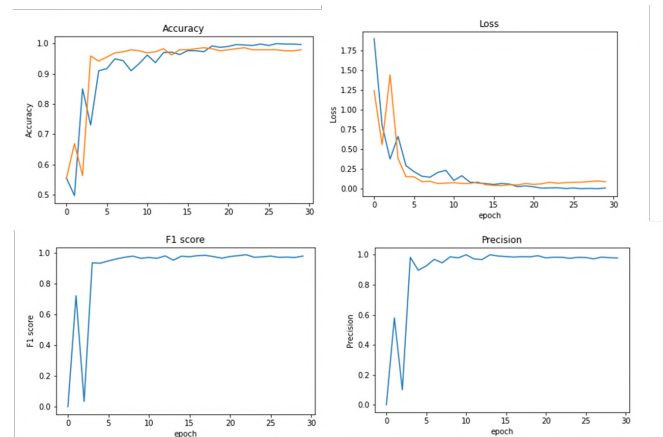


Figura 13. Gráfico do segundo treino.

Tomando de base a média dos dois primeiros treinos vimos que o primeiro treino se saiu melhor, então tomando como base os mesmos parâmetros deste treino foi feito um ajuste adicionando novas camadas. Os hiperparâmetros *Batch* e *Epoch* não foram alterados. Podemos observar na Tabela 14 e Figura 14 que as médias melhoraram com essas mudanças. Abaixo temos o exemplo das camadas que foram mudadas no código utilizando *Transfer Learning*.

```
Conv2D(32, (5,5), activation='relu', padding='same')(x)
Conv2D(64, (5,5), activation='relu', padding='same')(x)
Dropout(0.25)(x)
Dense(2048, activation='relu')(x)
Dropout(0.5)(x)
GlobalAveragePooling2D()(x)
Dense(1024, activation='relu')(x)
Dropout(0.5)(x)
Dense(512, activation='relu')(x)
```

Tabela 14. Média do quarto treino.

Média	Validação
Accuracy	0.965
Loss	0.094
F1_score	0.964
Precision	0.956

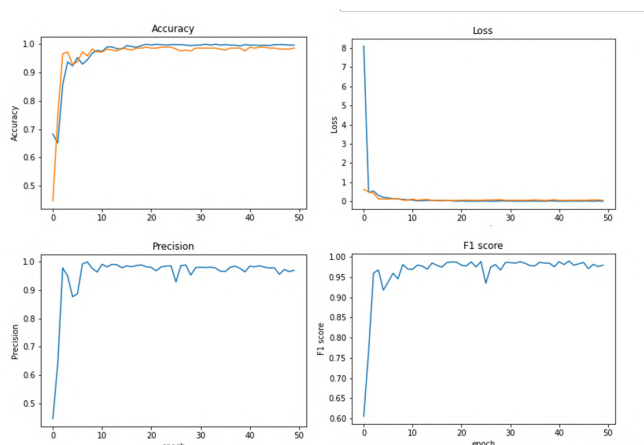


Figura 14. gráfico do quarto treino.

Neste último treino foi feita uma pequena mudança nas camadas de *dropout*, o que melhorou um pouco mais a rede neural como podemos ver na Tabela 15 e Figura 15.

Tabela 15. Média do quinto treino.

Média	Validação
Accuracy	0.976
Loss	0.064
F1_score	0.961
Precision	0.973

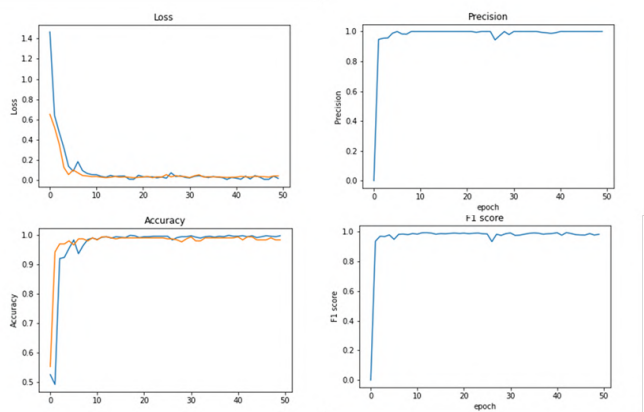


Figura 15. Gráfico do quinto treino.

Considerações Finais

Tendo em vista a evolução do projeto, pode-se dizer que obteve excelentes resultados durante o desenvolvimento do mesmo e atingiu grande parte dos objetivos propostos.

E com trabalhos futuros poderá ser implementado um método mais prático, utilizando de fato a leitura dos EPI's e onde o sistema já criado poderia ser instalado em um ambiente físico para testes e de fato visualizar tudo que foi documentado por meio de pesquisas.

Agradecimentos

Ao IFMS pela bolsa concedida para execução deste projeto.

Referências

BOESCH, Gaudenz. **Anotação de imagem: novas tecnologias e ferramentas em 2022**. [S.I] 2022. Disponível em: <<https://viso.ai/computer-vision/image-annotation/>>. Acesso em: 8 setembro. 2022.

GITHUB. YOLOV5. [S.I] [2022]. Disponível em: <<https://github.com/ultralytics/yolov5>>. Acesso em: 20 fevereiro. 2022.

MEAZZINI, Luis. **Tutorial - Transfer Learning aplicado no reconhecimento de flores. 2019**. Disponível em: <<https://medium.com/ensina-ai/tutorial-transfer-learning-3972cac5e9b5>>. Acesso em: 8 setembro. 2022.

MINISTÉRIO DO TRABALHO E EMPREGO. NR 6 - EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL - EPI. Disponível em: <https://enit.trabalho.gov.br/portal/images/Arquivos_SST/ST_NR/NR-06.pdf>. Acesso em: 14 abril. 2022.

PERES, Lucas. **Aprenda a Criar e Treinar Uma Rede Neural Convocucional (CNN)**. insight lab. 2021. Disponível em: <<https://insightlab.ufc.br/aprenda-a-criar-e-treinar-uma-rede-neural-convocucional-cnn/#:~:text=As%20redes%20neurais%20convolucionais%20%28CNN%29%20se%20utilizam%20de,que%20%C3%A9%20vantajoso%20para%20trabalhar%20com%20redes%20profundas>> Acesso em: 8 setembro. 2022.

ROBOFLOW. **EPI Image Dataset**. [S.I] [2022] Disponível em: <<https://app.roboflow.com/deivid-wesley/epi-kzk7e/2>>. Acesso em: 20 fevereiro. 2022.

TENSORFLOW. **tf.keras.preprocessing.image.ImageDataGenerator**. [S.I] [2022]. Disponível em: <<https://pt.wikihow.com/Citar-um-Site>> Acesso em: Acesso em: 8 setembro. 2022.

USE OF MACHINE LEARNING FOR IDENTIFICATION OF HARD-HAT.

Abstract: *One factor that allows to minimize accidents with construction workers is the use of Personal Protective Equipment (PPE). Regardless of the Regulatory Norms established as requirements for use, storage and conservation, there is a problem of cultural use that, especially in small buildings, is difficult to solve. Thus, the objective of this article was to propose the use of CNNs for the recognition of personal protective equipment. The result is the implementation of two neural networks, Yolov50 which was made available preliminarily the recognition of EPI helmets, the data set by the roboflow platform that facilitated the comparison of images.*

Keywords: *hard hat, machine learning, character recognition .*