

1      **Computer Vision for Automatic Classification of Wounds with YOLOv8 Neural**  
2      **Network**  
3

4      JOHNNY CLEITON WILLY DA SILVA\*, Federal University of Pernambuco, Brazil  
5

6      ADRIEN DURAND-PETITEVILLE, Federal University of Pernambuco, Brazil  
7

8      JONAS FERREIRA SILVA, Federal University of Pernambuco, Brazil  
9

10     This work employs the YOLOv8 algorithm for the detection and classification of wounds in digital images, with the primary objective  
11    of showcasing the potential of Computer Vision in the healthcare domain, particularly in the identification and classification of injuries.  
12    Using a dataset, the study aims to demonstrate the efficacy of the YOLOv8 neural network in lesion identification. By integrating  
13    Artificial Intelligence (AI) into medical applications, this work proposes an alternative for wound prognosis, serving as a potential  
14    application to assist in self-treatment of injuries.  
15

16    Key Words: Classification, Wounds, Computer Vision, YOLOv8  
17

18  
19    **1 INTRODUCTION**  
20

21     Computer vision has emerged as a powerful tool in the healthcare domain, presenting significant potential to enhance  
22    the detection and classification of medical conditions. In this context, this work focuses on the application of the  
23    YOLOv8 algorithm for the detection and classification of wounds in digital images, exploring the vast possibilities  
24    offered by artificial intelligence (AI) in this scenario. Wounds, whether acute or chronic, pose a significant challenge in  
25    the healthcare field, requiring innovative approaches for diagnosis and treatment [13]. With the advancement of digital  
26    technologies and the growing availability of medical data, the field of artificial intelligence emerges as a promising tool  
27    to improve the identification and effective classification of different types of wounds. In this context, neural networks  
28    stand out as powerful instruments capable of learning complex patterns from extensive datasets.  
29

30     This work explores the possibility of using the YOLOv8 neural network for the classification and localization of  
31    wounds in images and videos. Leveraging the YOLOv8 architecture, renowned for its efficiency and speed in object  
32    detection, the study aims to present a new alternative to enhance precision and effectiveness in the specific context of  
33    wound identification. The neural network is trained on datasets of images and tested on images and videos containing  
34    various instances of wounds, allowing YOLOv8 to develop robust discrimination and localization capabilities. This  
35    project aims to make a significant contribution to the automation of wound diagnosis, potentially streamlining and  
36    improving the accuracy of the clinical process, resulting in substantial benefits for the healthcare field.  
37

38     This article is structured as follows: Section 2 provides the definition of the technologies and tools used, which were  
39    necessary for the project's implementation. In Section 3, other studies that have also employed the YOLO algorithm for  
40    wound detection are described. In Section 4, the methodology for wound localization and classification is presented,  
41    including the assembly of the dataset and the training of the YOLO neural network. Section 5 contains the results of the  
42    work, along with discussions about the findings. Finally, Section 6 presents the conclusion, listing possible future work.  
43

44  
45    Authors' addresses: Johnny Cleiton Willy da Silva, jcws@softex.cin.ufpe.br, Federal University of Pernambuco, Recife, Pernambuco, Brazil; Adrien  
46    Durand-Petiteville, Federal University of Pernambuco, Recife, Brazil, ajsd@cin.ufpe.br; Jonas Ferreira Silva, Federal University of Pernambuco, Recife,  
47    Brazil, jfs6@cin.ufpe.br.  
48

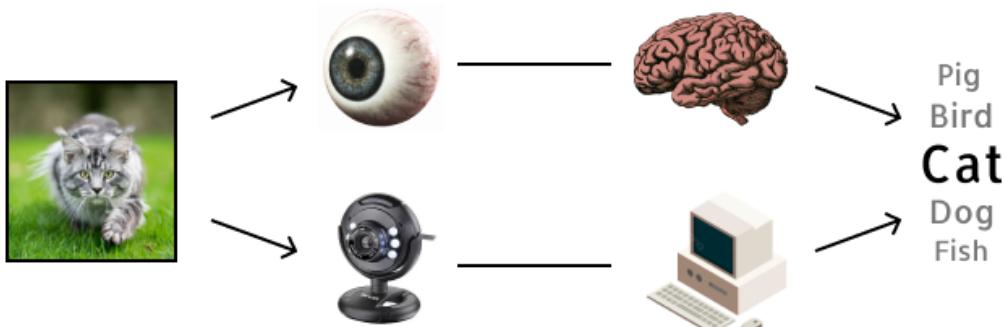
## 53      2 BACKGROUND

54  
55 In this section, a definition is provided for the main areas and technologies utilized during the execution of this work.  
56 The solution proposed in this article is the automatic detection and classification of wounds, grounded in the metrics  
57 for lesion classification in the healthcare field and the processing of digital images through computer vision algorithms.  
58

### 59      2.1 Computer Vision: classification and localization tasks

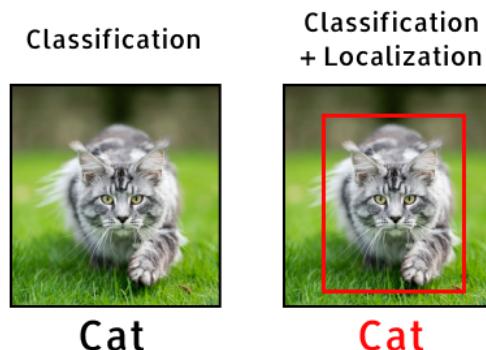
60 Computer Vision (CV) is an interdisciplinary field that aims to empower computers to interpret and comprehend visual  
61 information, simulating the human capacity to process images [8]. This domain of artificial intelligence involves the  
62 development of algorithms and models capable of analyzing visual data, triggering a variety of practical applications  
63 in sectors such as medicine, industrial automation, autonomous vehicles, security, and entertainment. Enhancing  
64 computers' ability to visually understand the world is crucial since human vision plays a crucial role in decision-making.  
65 By endowing machines with a more sophisticated visual understanding, they become better equipped to make informed  
66 and contextually relevant choices, thus approaching the human capacity to process visual information to improve the  
67 quality and accuracy of decisions made [12]. The visual understanding of the world, for both humans and machines,  
68 involves the ability to detect, recognize, and interpret objects in images. Among various essential tasks in the field of  
69 CV, image classification stands out, where systems are trained to assign labels or categories to objects present in a scene  
70 [26].  
71

72      2.1.1 *Classification*: The computational classification task, in theory, bears similarities to human vision. As illustrated  
73 in Figure 1, the human system uses eyes for visualization and the brain for interpretation, while the computational  
74 system employs a camera for input capture and a processing unit for interpretation. Both systems can thus recognize  
75 the input, enabling the identification of the class to which the object belongs. Taking objects that can belong to various  
76 classes as an example, such as dogs, cats, birds, among others, the computer vision model assigns a class based on  
77 probabilities. In the context of an image, the system determines the probability that the detected animal belongs to the  
78 cat class, for instance, establishing a classification with the highest possible confidence. This approach highlights the  
79 ability of computer vision models to perform complex classification tasks, reflecting a notable similarity to the human  
80 ability to interpret and categorize visual objects.  
81



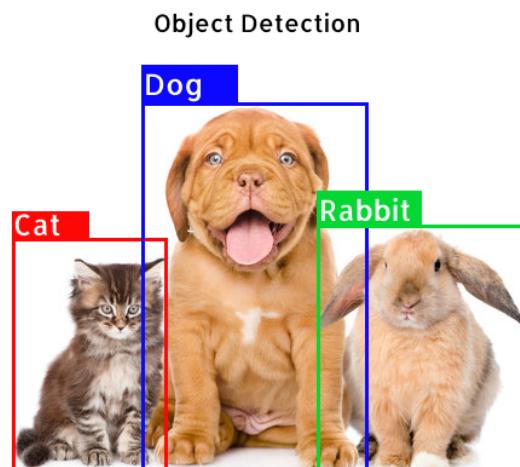
102      Fig. 1. Classification of images in human vision and in computer vision.  
103  
104

105    2.1.2 *Localization*: The task of classification and localization in computer vision refers to the ability of a computational  
 106 system to identify and classify objects in an image while simultaneously determining the precise location of these  
 107 objects [12]. As illustrated in Figure 2, the localization component involves determining the spatial coordinates of  
 108 the identified object in the image. This localization assignment is often achieved through bounding boxes, which are  
 109 rectangles that enclose the region where the object is situated. The coordinates of these bounding boxes represent  
 110 the relative position of the object in relation to the image, thus providing a detailed understanding of the spatial  
 111 arrangement of the identified elements.  
 112  
 113



124 Fig. 2. Classification and localization of objects in a scene.  
 125  
 126  
 127  
 128

129    2.1.3 *Detection*: While the task of classification and localization focuses on analyzing the image as a whole and  
 130 providing information about the location of a specific object within that image, the task of detection goes further by  
 131 identifying the presence of multiple objects and providing data about their classes and individual locations. This is  
 132 illustrated in Figure 3, where the detection of various classes, such as cat, dog, and rabbit, takes place.  
 133  
 134



135 Fig. 3. Object detection in a scene.  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152  
 153

Both tasks, detection and classification followed by localization, play crucial roles in various computer vision applications, with the choice between them guided by the specific requirements of the problem at hand. In the scope of this work, the focus will be on the use of the classification and localization task, emphasizing its relevance in contexts where the goal is to understand and identify specific elements within an image.

## 2.2 YOLO algorithm

Among the various object detection algorithms, the YOLO (You Only Look Once) framework has stood out for its remarkable balance between speed and accuracy, enabling quick and reliable identification of objects in images and videos [23]. It addresses the object detection problem differently from traditional approaches, such as Region-based Convolutional Neural Network (R-CNN) and its variants. According to its creators (Redmon *et al.* [17]), the main distinctive feature of YOLO is that it performs object detection in a single pass through the neural network, rather than dividing the task into multiple steps, such as region proposals, feature extraction, and classification. This design makes YOLO faster and more computationally efficient.

Currently, YOLO has 8 different versions, with each one bringing improvements in terms of accuracy and speed. Figure 4 presents the timeline of the algorithm, showcasing each year of evolution.

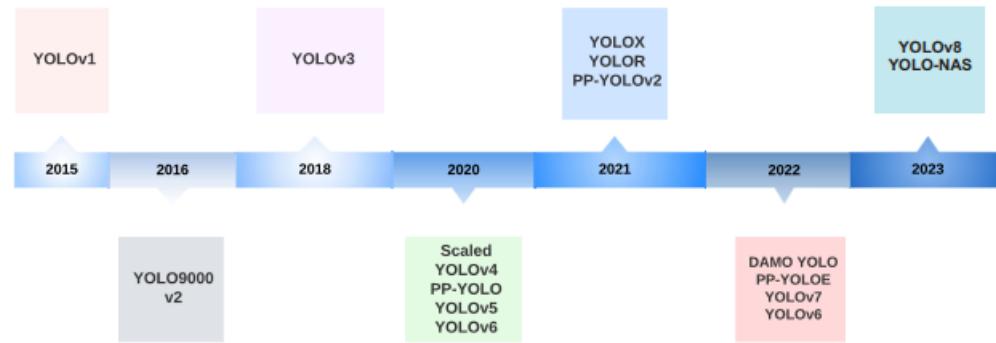


Fig. 4. A timeline of YOLO versions. Source: Juan and Diana [23]

The original version of YOLO, proposed by Joseph Redmon [17], was released in 2015. Innovatively, this algorithm introduced the approach of object detection in a single pass through the neural network, providing notable efficiency. This approach involves dividing the image into a grid, where each cell is responsible for predicting bounding boxes and the associated probabilities for specific classes. Although it marked a significant advancement, the original YOLO had limitations in the accuracy of detecting small objects and in situations of overlap.

The second version of YOLO, also known as YOLO9000, was released in 2016 and brought significant improvements. It introduced anchors to aid in predicting bounding boxes and had the capability to detect a large number of classes (over 9000) using a diverse dataset. It improved accuracy and dealt with smaller objects more effectively. Version 3 was released in 2018 [18] and brought further improvements in accuracy and speed. It adopted a deeper network architecture with three scales of detection. It implemented object detection at multiple scales, allowing efficient detection of objects of different sizes, and introduced the option to use different image sizes during training and testing.

209 YOLOv4 was released in 2020, led by Alexey Bochkovskiy [5], and introduced significant improvements in accuracy  
 210 and efficiency compared to YOLOv3. Focused on increasing accuracy, efficiency, and object detection capability in  
 211 images and videos, YOLOv4 adopted the Darknet architecture to enhance feature representation and, consequently,  
 212 model accuracy. A few months later, version 5 of YOLO was released by Glen Jocher, founder and CEO of Ultralytics  
 213 [24]. In this version, one of the novelties was the change in architecture, using PyTorch instead of Darknet.  
 214

215 A sixth enhanced version of YOLO [11] was released in September 2022, introducing innovations such as task learning  
 216 strategies, improved classification/regression losses, and a quantization scheme. These improvements result in faster  
 217 and more accurate detectors, surpassing previous models. The seventh version of YOLO was published in July 2022 by  
 218 the same authors as YOLOv4, surpassing all known object detectors in terms of speed and accuracy [23].  
 219

220 2.2.1 *YOLOv8*: The version 8 of YOLO was released in January 2023 by the same company that developed YOLOv5,  
 221 Ultralytics [25]. It came with five scaled versions, where each version varies to meet specific applications and hardware  
 222 requirements. In Table 1, the nomenclature "YOLOv8n," "YOLOv8s," "YOLOv8m," "YOLOv8l," and "YOLOv8x" denotes  
 223 different versions or scales of the model, each optimized for different use-case scenarios:  
 224

- 225 (1) YOLOv8n (Nano): This version is designed to be lightweight and suitable for devices with limited resources,  
 226 such as embedded devices or systems with hardware constraints. It can be considered a "nano" model due to its  
 227 compact nature and resource efficiency.
- 228 (2) YOLOv8s (Small): This version is a small-sized option, designed to provide a balance between performance and  
 229 efficiency. It is more powerful than Nano but still maintains a relatively small footprint, making it suitable for a  
 230 variety of applications.
- 231 (3) YOLOv8m (Medium): This version is medium-sized, offering a balance between performance and size. It can  
 232 be an intermediate choice between lighter models and more robust models, suitable for a variety of computer  
 233 vision tasks.
- 234 (4) YOLOv8l (Large): YOLOv8l is a large version, optimized to provide more robust performance. Larger models  
 235 often have more parameters and can capture more complex relationships, but they may also be more demanding  
 236 in terms of computational resources.
- 237 (5) YOLOv8x (Extra Large): This is the extra-large version, optimized to deliver maximum performance, even if it  
 238 means a more substantial computational footprint. It is ideal for cases where performance is crucial, and robust  
 239 hardware resources are available.

245  
 246 Table 1. YOLOv8 scaled versions. Source: Ultralytics [25]  
 247

Model	Size (pixels)	mAPval (50-95)	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

256  
 257 Each column represents a specific feature of the YOLOv8 models:  
 258

- 259 • Model - name of the model;  
 260

- Size - size in pixels used for training and testing the model;
- mAPval - the mean average precision on validation data, a common metric used to evaluate the quality of the object detection model. "50-95" refers to the overlap range used to calculate the average;
- Speed (CPU ONNX) - average time in milliseconds the model takes to process an image when run on a CPU using the ONNX format;
- Speed (A100 TensorRT) - average time in milliseconds the model takes to process an image when run on an A100 GPU using TensorRT, a deep learning inference library optimized for NVIDIA GPUs;
- Params - number of model parameters, measured in millions (M) and generally related to the model's complexity;
- FLOPs (B) - the number of floating-point operations per second, measured in billions (B), a measure of the model's processing workload.

As shown in the graphs in Figure 5 taken from the company's Github page Ultralytics, the authors claim that YOLOv8 outperforms previous YOLO releases such as YOLOv7, YOLOv6 and YOLOv5. The graph shows the COCO (Common Objects in Context) dataset, a challenging dataset widely used for training and evaluating computer vision algorithms, especially in object detection tasks.

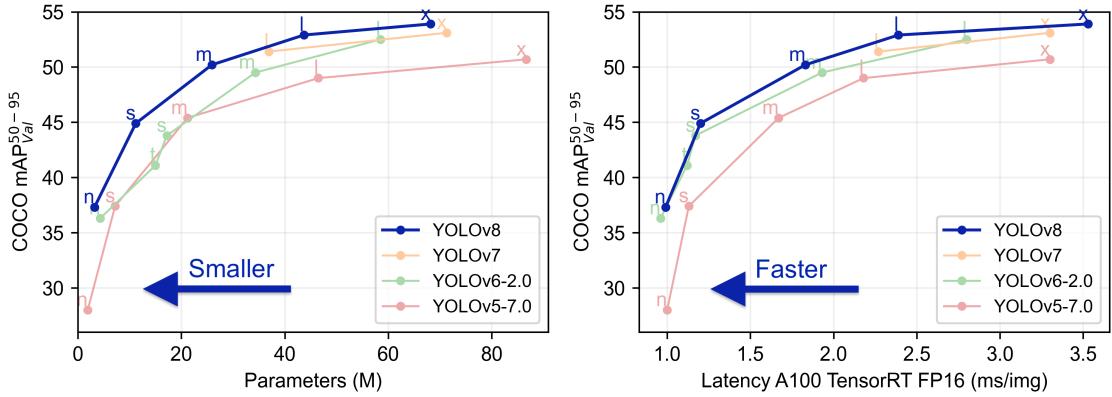


Fig. 5. YOLOv8 compared to previous versions. Source: Ultralytics Github Page [25]

The upward curve in both graphs demonstrates the efficiency of YOLO algorithms. In the left graph, varying Parameters (M) from 0 to 80 explores different model configurations. The consistent increase in the COCO mAPval (50-95) metric as parameters are adjusted indicates an improvement in model performance with higher settings. In the right graph, latency (inference time) in milliseconds per image (ms/img) is considered in relation to YOLO model performance on the COCO dataset. The graph varies latency in relation to the COCO mAPval (50-95) metric, indicating that as latency increases, the mAP metric also increases, finding a suitable balance between latency and accuracy (mAP).

### 2.3 Biological Classification of Wounds

A wound is a disruption in the integrity of the skin or mucous membrane, often associated with damage to underlying tissues. This injury can be triggered by a variety of factors, and the healing process is typically initiated to restore tissue integrity [10]. Various types of wounds exist, and their classification can vary depending on various elements such as

313 the origin of the injury, its depth, extent, and the presence of infection. Wound identification can be done visually and  
 314 through biological indicators. Figure 6 visually presents various wound shapes.  
 315



339 Fig. 6. Visual variations of wounds: (a) cut, (b) burn, (c) ulcer, (d) blister, (e) bite, (f) necrosis, (g) hematoma, (h) open cut, (i) weapons  
 340 of fire, (j) scratch, (k) stitched, (l) infection.

341 However, as highlighted by Santos *et al.* (2011), although dermal wounds encompass a diversity of skin injuries, they  
 342 can be broadly categorized into three main types: surgical, pathological and traumatic. Figure 7 visually illustrates this  
 343 division.



359 Fig. 7. classification of wounds by cause.

361 **2.3.1 Surgical Wounds:** they result from planned procedures, generally carried out in controlled and sterile envi-  
 362 ronments, such as operating rooms. They are characterized by a deliberate incision in the skin to carry out medical  
 363

364

365 interventions. These wounds are classified as aseptic clean due to the highly controlled environment where strict asepsis  
366 measures are applied. The objective is to minimize the presence of microorganisms and reduce the risk of infection.  
367 Special attention to surgical technique, hygiene and the use of sterile materials contributes to this classification [4].  
368

369     2.3.2 *Pathological or Ulcerative Wounds*: are associated with underlying medical conditions, such as pressure ulcers in  
370 bedridden patients, diabetic ulcers, or wounds resulting from specific dermatological conditions. The classification of  
371 these wounds depends on the presence or absence of infection and the degree of contamination [20].  
372

373     2.3.3 *Traumatic Wounds*: result from unscheduled events, such as accidents, falls or impacts. They can be classified  
374 according to contamination and degree of injury. Contaminated wounds result from traumatic events in dirty environ-  
375 ments or exposure to contaminated materials, and infected wounds develop when a secondary infection occurs after  
376 the injury. Management of these wounds requires careful assessment and specific measures to prevent infections [7].  
377

### 378     3 RELATED WORKS

381 In this section, studies are described that use different versions of YOLO to analyze and classify wounds, providing an  
382 innovative perspective on wound detection. The approaches presented highlight the practical application of YOLO in  
383 clinical contexts, enriching the understanding of its capabilities and contributing to the improvement of diagnoses and  
384 wound monitoring.  
385

387 In the year 2020, Patel and Yash [15] proposed a wound care recommendation system using image detection and  
388 classification techniques. Using YOLOv2, the system classifies key tissue in the wound image and suggests contextually  
389 appropriate coverage options. In 2022, Anisuzzaman et al. [3] developed an automatic wound locator using the YOLOv3  
390 model, transformed into an iOS application, capable of detecting and isolating injured regions in images. After training  
391 and testing, the YOLOv3 model achieved 93.9% average accuracy (mAP).  
392

394 In a recent study, Muhammad Adnan et al. [1] (2023) proposed an automatic wound detection technique using the  
395 YOLOv3 model, which locates and classifies wounds into four main categories: suture, cut, open, and normal skin.  
396 Experimental results demonstrate significant efficiency and robustness, with accuracy above 90%, outperforming other  
397 manual approaches. In the work of Aldughayfiq et al. [2] (2023) an approach for the detection and classification of  
398 pressure ulcers is presented, using the YOLOv5 model. With an emphasis on early detection, the method classifies  
399 ulcers into four stages and non-ulcers. The promising results show an overall average accuracy of 76.9%. Compared to  
400 previous studies, this approach offers a more efficient and accurate solution, with the potential to improve the early  
401 detection and treatment of pressure ulcers.  
402

404 Despite the wide use of the YOLO algorithm in the analysis and classification of wounds, there are still no applications  
405 in the literature with the eighth version of the algorithm, precisely the one used in this work.  
406

### 407     4 METHODOLOGY

409 In this section we describe the activities performed to obtain wound classification using the YOLOv8 neural network. In  
410 the project, we followed two important steps. First, we gather photos and videos of wounds to create a dataset. This  
411 involved taking, organizing and labeling this information. Then, we moved on to the second stage, which consisted of  
412 training the YOLOv8 neural network to recognize and locate wounds. This phase included creating the network, tuning  
413 the model for better performance, and verifying that it identifies wounds correctly. In Figure 8 it is possible to visualize  
414 these two main steps followed by their secondary steps.  
415

416

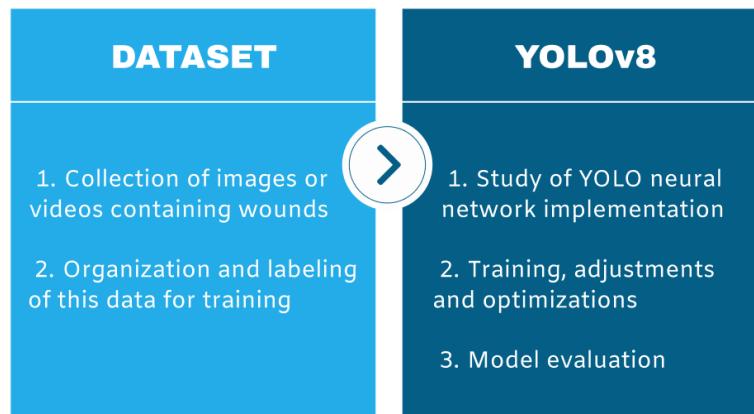


Fig. 8. Project methodology.

#### 4.1 Dataset and Labeling

To perform YOLOv8 training, three sets of data were used, the first consists of 223 images of surgical wounds (cleaned and sutured); the second with 253 images of pathological wounds (ulcers); and the third with 361 images of traumatic wounds (cuts and scratches). In total, 837 images were used in the dataset, with 585 for training, 125 for validation and 127 for testing. Each set of data, containing the markings of the locations of each wound and a text file with the same name as the image, was distributed into three folders "train", "val" and "test" according to the proportion 70-15-15 , where 70% of the data is used for training, 15% for validation and 15% for testing. The distribution between images of each class for training, validation and testing can be seen in Table 2.

Wounds	Total	Train (70%)	Val (15%)	Test (15%)
surgical	223	156	33	34
pathological	253	177	38	38
traumatic	361	252	54	55
DATASET	837	585	125	127

Table 2. Distribution of the image dataset in a 70-15-15 proportion.

The dataset used was the result of the combination of four datasets ([16], [14], [22], [21]) acquired on the Roboflow platform [19]. All images measuring 640x640 pixels were labeled using bounding boxes using the Computer Vision Annotation Tool (CVAT) [6], as shown in Figure 9. It was not possible to use the *data augmentation* technique to increase the data, as the selected datasets already had this technique applied.

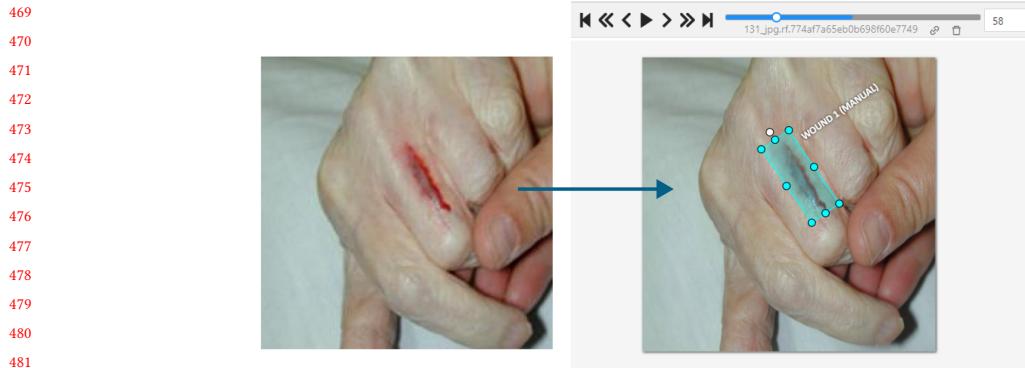


Fig. 9. Image labeling via CVAT tool.

#### 4.2 Training with YOLOv8

After assembling the dataset, the implementation of YOLOv8 began. For this, the Google Colab execution environment was used, managing the code through the use of GPU (Tesla T4), a graphics processing unit manufactured by the company NVIDIA, designed especially to accelerate artificial intelligence (AI) and learning workloads and deep learning. As an initial step, the algorithm was executed with the purpose of locating wounds in general in the images. Therefore, the images in the data set with the three classes were labeled as a single class, called "wound". After this, training was carried out to classify and locate wounds in each of the classes, labeled as "surgical", "pathological" and "traumatic". Because it is a relatively small dataset and does not require high data processing, the Nano version (YOLOv8n) was chosen to be used in training, which consisted of adjusting four different epochs: 5, 20, 50 and 80.

## 5 RESULTS

In this section, the results of the work are presented, where the data set for general wound recognition was first run and then the training for classification into three classes of wounds was run. In each subsection, the presented results are discussed.

### 5.1 Classification of Wounds with One Class

The training of the wound classification and localization network for a single class (wound) obtained a satisfactory visual result from 50 epochs, where the wound location only consists of a bounding box at the location of the wound, different from the number of smaller epochs previously used, where the result obtained several bounding boxes for just one detection. This is due to the number of epochs used, as during each epoch, samples in the training set are presented to the neural network, and the weights are adjusted based on the differences between the network's predictions and the desired outputs (labels). In this way, each epoch means a single complete pass through the entire training set during training a model. Figures 10 and 11 shows the results of inferences (or "predictions"), which are the outputs generated by the neural network during training using 5 epochs and 50 epochs. The result of wound detection using 50 epochs applied to a video can be checked [here](#).<sup>1</sup>

<sup>1</sup>The video is available at [<https://drive.google.com/file/d/15a2ynl4zlg0zOkzjLKAjqTTSABwD6gRu/view?usp=sharing>].



Fig. 10. Classification result using 5 epochs.

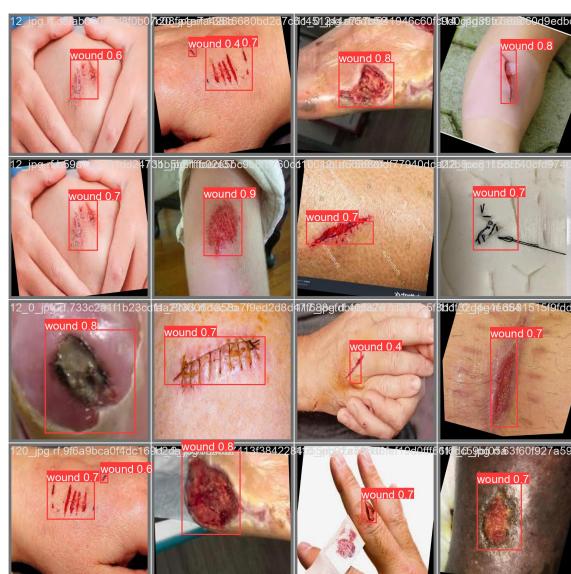


Fig. 11. Classification result using 50 epochs.

567 It is also possible to observe that there is a part of the bounding boxes in the classification with 5 epochs that  
568 recognized the "background," meaning other objects in the scene besides the wounds were erroneously classified as  
569 wounds. Additionally, there are some images where wounds were not recognized. This demonstrates that 5 epochs  
570 were not sufficient for the model to learn adequately.  
571

572

**573      5.2 Classification of Wounds with Three Classes**

**574**  
**575** When training the neural network for 3 different classes of wounds, there was a satisfactory (but not ideal) result after  
**576** 20 training epochs, where it is possible to see that the wounds were classified correctly. Figure 12 presents the results of  
**577** the neural network predictions for classifying wounds into pathological, surgical and traumatic classes with 50 epochs;  
**578** and Figure 13 demonstrates a test image comparison between the use of 5 epochs and 50 epochs, where, in training  
**579** with 5 epochs, there is an erroneous classification of a pathological wound to a traumatic wound.  
**580**

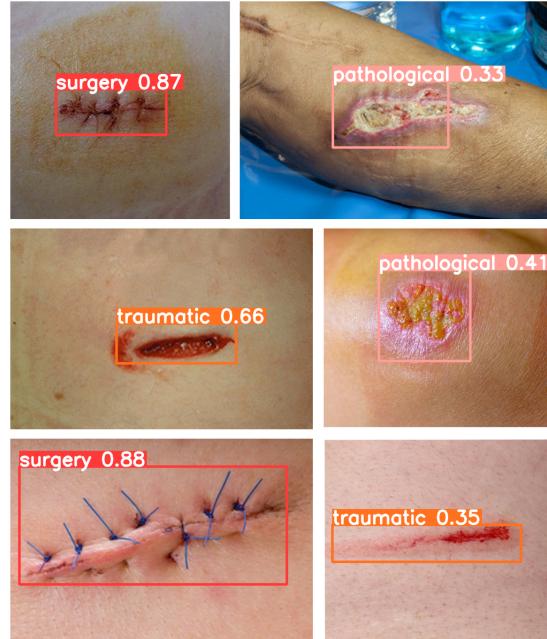


Fig. 12. Classification result using 50 epochs

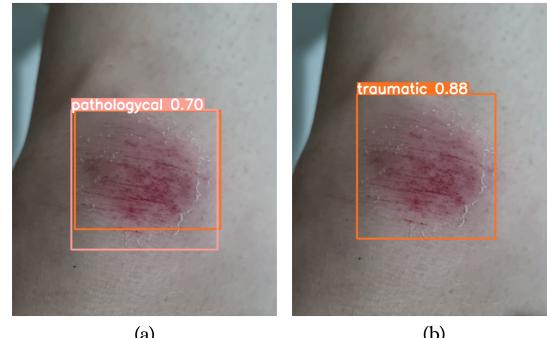
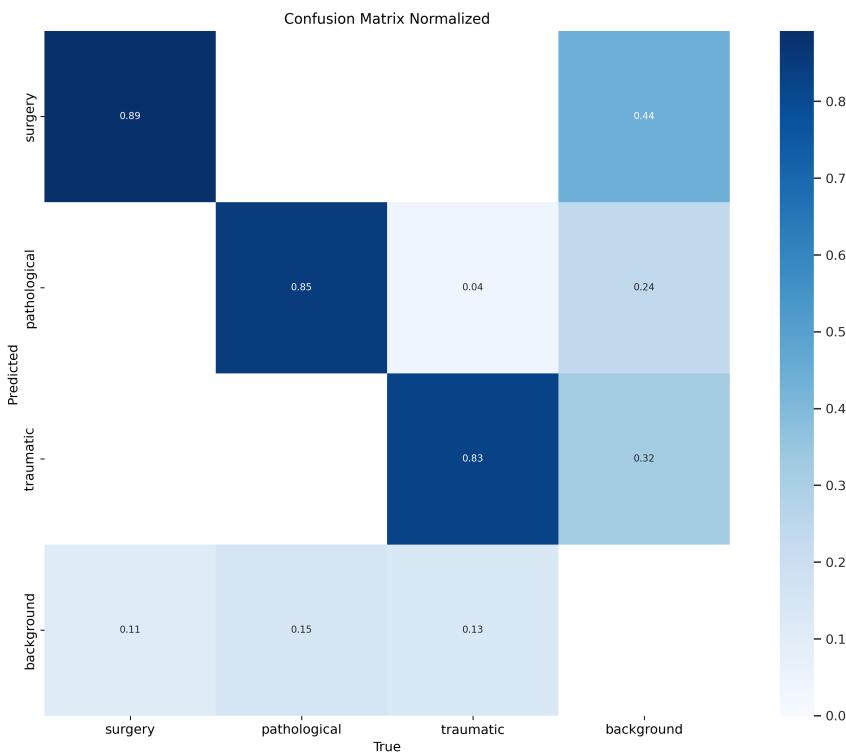


Fig. 13. Result of test image classification using two different epochs: (a) 5 epochs and (b) 50 epochs.

625      Figure 14 shows the 4x4 confusion matrix of the normalized training result with 50 epochs. The confusion matrix is  
 626      a commonly used tool to evaluate the effectiveness of the model in accurately predicting the classes of detected objects  
 627      [9]. By observing the matrix of 50 epochs, it is possible to conclude that, during training, no class achieved 100% correct  
 628      predictions.  
 629



630  
 631      Fig. 14. Normalized confusion matrix using 50 epochs for training.  
 632  
 633  
 634  
 635  
 636  
 637  
 638  
 639  
 640  
 641  
 642  
 643  
 644  
 645  
 646  
 647  
 648  
 649  
 650  
 651  
 652  
 653  
 654  
 655  
 656  
 657  
 658  
 659  
 660

661      It is possible to analyze that 44% of the images belonging to surgical wounds were incorrectly classified as "background",  
 662      as well as 24% of the images of pathological wounds and 32% of the images of traumatic wounds. Furthermore, it can  
 663      also be observed that there were an incorrect classification between the images of wound classes, with 4% of traumatic  
 664      wounds judged as pathological wounds. These erroneous results demonstrate that there was a need for a more complete  
 665      dataset, with greater variation in images, to carry out the training.  
 666

## 6 CONCLUSION AND FUTURE WORK

667      This project used the YOLOv8 algorithm to classify wound images into three different classes. In addition to this step,  
 668      the same dataset was used for the classification of wounds in general. The results were satisfactory considering the low  
 669      scope of the database used, however it is necessary to create a more comprehensive and better quality dataset. The  
 670      biggest obstacle encountered in this work was obtaining a dataset compatible with the proposal, which is why it was  
 671      necessary to combine several datasets and process the data for each one.  
 672  
 673  
 674  
 675  
 676

677 For future work, we can list the construction of a dataset with higher quality and more image variations. It is also  
 678 necessary to expand the number of wound classes to be classified, as in this work it was necessary to reduce the number  
 679 of classes covered due to the Inadequacy of existing datasets, it is also necessary to implement a graphical interface of  
 680 the system for interaction with the user and, finally, carry out tests and validations in real scenarios.  
 681

## 683 REFERENCES

- 684 [1] Muhammad Adnan and et al. 2023. An Automatic Wound Detection System Empowered by Deep Learning. *Journal of Physics: Conference Series*  
 685 2547, 1 (2023).
- 686 [2] Bader Aldughayfiq and et al. 2023. Yolo-based deep learning model for pressure ulcer detection and classification. *Healthcare* 11, 9 (2023).
- 687 [3] D. M. Anisuzzaman and et al. 2022. A mobile app for wound localization using deep learning. *IEEE Access* 10 (2022), 61398–61409.
- 688 [4] Leila Blanes. 2004. *Tratamento de Feridas*. Baptista-Silva JCC Editor, São Paulo. [citado 2010 out 15]. Disponível em: <http://www.baptista.com>.  
 689 Acesso em: 19 nov. 2023..
- 690 [5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
- 691 [6] CVAT. 2023. *Labeling service*. <https://www.cvcat.ai/> Acesso em: 9 mar. 2022.
- 692 [7] Carol Dealey. 2001. *Cuidando de Feridas: Um Guia para Enfermeiras* (2 ed.). Atheneu, São Paulo.
- 693 [8] David A Forsyth and Jean Ponce. 2002. *Computer vision: a modern approach*. prentice hall professional technical reference.
- 694 [9] Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*  
 695 (2020).
- 696 [10] Naomi Hargrove. 2018. *Wound Care Essentials: Practice Principles* (4th ed.). Wolters Kluwer.
- 697 [11] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. 2022. YOLOv6:  
 698 A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976* (2022).
- 699 [12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft  
 700 coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V* 13. Springer, 740–755.
- 701 [13] Laurel M. Morton and Tania J. Phillips. 2016. Wound healing and treating wounds: Differential diagnosis and evaluation of chronic wounds. *Journal  
 702 of the American Academy of Dermatology* 74, 4 (2016), 589–605.
- 703 [14] open challenge. 2023. wound detector Dataset. <https://universe.roboflow.com/open-challenge/wound-detector>. <https://universe.roboflow.com/open-challenge/wound-detector> visited on 2023-11-20.
- 704 [15] Yash Patel. 2020. *Deep Learning-Based Object Detection in Wound Images*. Ph.D. Dissertation. The University of Wisconsin-Milwaukee.
- 705 [16] project. 2023. Wound Detection Dataset. <https://universe.roboflow.com/project-tvs2o/wound-detection-u7br1>. <https://universe.roboflow.com/project-tvs2o/wound-detection-u7br1> visited on 2023-11-20.
- 706 [17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the  
 707 IEEE conference on computer vision and pattern recognition*, 779–788.
- 708 [18] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).
- 709 [19] Roboflow. 2023. *Custom Dataset*. <https://universe.roboflow.com/> Acesso em: 9 mar. 2022.
- 710 [20] B. S. Santos and outros. 2011. *Avaliação e tratamento de feridas – Orientações aos profissionais de saúde*.
- 711 [21] subu. 2022. aug deneme2 Dataset. [https://universe.roboflow.com/subu-tsw3j/aug\\_deneme2](https://universe.roboflow.com/subu-tsw3j/aug_deneme2)  
 712 visited on 2023-11-20.
- 713 [22] subu. 2022. sb Dataset. <https://universe.roboflow.com/subu-tsw3j/sb-obcga> visited on  
 714 2023-11-20.
- 715 [23] Juan Terven and Diana Margarita Córdova-Esparza. 2023. A Comprehensive Review of YOLO: From YOLOv1 and Beyond. *Comput. Surveys* (9 June  
 716 12 2023). arXiv:2304.00501v3 [cs.CV] Under Review.
- 717 [24] Ultralytics. 2022. YOLOv5. <https://github.com/ultralytics/yolov5> Acesso em: 10 mar. 2022.
- 718 [25] Ultralytics. 2023. YOLOv8. <https://github.com/ultralytics/yolov8> Acesso em: 11 mar. 2022.
- 719 [26] Yudong Zhang, Shuihua Wang, Genlin Ji, and Preetha Phillips. 2014. Fruit classification using computer vision and feedforward neural network.  
 720 *Journal of Food Engineering* 143 (2014), 167–177. <https://doi.org/10.1016/j.jfoodeng.2014.07.001>
- 721
- 722
- 723
- 724
- 725
- 726
- 727
- 728