Journal of Pain Research Dovepress

Open Access Full Text Article ORIGINAL RESEARCH

Employing the Artificial Intelligence Object Detection Tool YOLOv8 for Real-Time Pain Detection: A Feasibility Study

Marco Cascella[1], Mohammed Naveed Shariff[2], Giuliano Lo Bianco[3], Federica Monaco[4], Francesca Gargano[5], Alessandro Simonini[6], Alfonso Maria Ponsiglione[7], Ornella Piazza[1]

[1]Anesthesia and Pain Medicine, Department of Medicine, Surgery and Dentistry "scuola Medica Salernitana", University of Salerno, Baronissi, 84081, Italy; [2]Department of AI&DS, Rajalakshmi Institute of Technology, Chennai, TN, India; [3]Anesthesiology and Pain Department, Fondazione Istituto G. Giglio Cefalù, Palermo, Italy; [4]Anesthesia and Pain Medicine, ASL NA1, Napoli, Italy; [5]Anesthesia and Intensive Care, U.O.C. Fondazione Policlinico Campus Bio-Medico, Roma, Italy; [6]Pediatric Anesthesia and Intensive Care Unit, Salesi Children's Hospital, Ancona, Italy; [7]Department of Electrical Engineering and Information Technology, University of Naples "federico II", Naples, 0125, Italy

Correspondence: Giuliano Lo Bianco, Anesthesiology and Pain Department, Fondazione Istituto G. Giglio Cefalù, Palermo, Italy, Email giulianolobianco@gmail.com

Introduction: Effective pain management is crucial for patient care, impacting comfort, recovery, and overall well-being. Traditional subjective pain assessment methods can be challenging, particularly in specific patient populations. This research explores an alternative approach using computer vision (CV) to detect pain through facial expressions.

Methods: The study implements the YOLOv8 real-time object detection model to analyze facial expressions indicative of pain. Given four pain datasets, a dataset of pain-expressing faces was compiled, and each image was carefully labeled based on the presence of pain-associated Action Units (AUs). The labeling distinguished between two classes: pain and no pain. The pain category included specific AUs (AU4, AU6, AU7, AU9, AU10, and AU43) following the Prkachin and Solomon Pain Intensity (PSPI) scoring method. Images showing these AUs with a PSPI score above 2 were labeled as expressing pain. The manual labeling process utilized an open-source tool, makesense.ai, to ensure precise annotation. The dataset was then split into training and testing subsets, each containing a mix of pain and no-pain images. The YOLOv8 model underwent iterative training over 10 epochs. The model's performance was validated using precision, recall, and mean Average Precision (mAP) metrics, and F1 score.

Results: When considering all classes collectively, our model attained a mAP of 0.893 at a threshold of 0.5. The precision for "pain" and "nopain" detection was 0.868 and 0.919, respectively. F1 scores for the classes "pain", "nopain", and "all classes" reached a peak value of 0.80. Finally, the model was tested on the Delaware dataset and in a real-world scenario.

Discussion: Despite limitations, this study highlights the promise of using real-time computer vision models for pain detection, with potential applications in clinical settings. Future research will focus on evaluating the model's generalizability across diverse clinical scenarios and its integration into clinical workflows to improve patient care.

Keywords: pain, artificial intelligence, automatic pain assessment, computer vision, action units

Introduction

Pain assessment is a critical component of patient care.[1] Since subjective methods for pain evaluation rely on self-reporting by patients, these approaches can be prone to inaccuracies, especially in cases where patients are unable to communicate effectively, such as infants,[2] individuals with cognitive impairments,[3] or those under sedation.[4]

Automatic pain assessment (APA) refers to a set of research and clinical methods used to provide an objective and quantifiable measure of pain, reducing reliance on subjective self-reports.[5] APA usually implements artificial intelligence (AI) strategies.[6] AI is a multidisciplinary field of computer science focused on designing and developing intelligent systems that mimic human cognitive abilities. Computer vision (CV) is a subfield of AI and computer science that can enable computers to interpret and understand the visual world.[7] It encompasses many tasks, including image and video

Received: 16 August 2024
Accepted: 6 November 2024
Published: 9 November 2024

recognition, object detection and tracking, image segmentation, scene reconstruction, and more.[8] Central to many of these CV tasks are Convolutional Neural Networks (CNNs). They are a class of deep learning models implemented for processing and analyzing visual data due to their ability to automatically and adaptively learn spatial hierarchies of features through backpropagation.[9]

In recent years, advancements in CV technologies have paved the way for innovative approaches to pain detection.[10] Since facial expressions of pain demonstrate consistency across various factors, including age, gender, cognitive states, and different types of pain,[11] AI and CV strategies can be implemented to objectively detect and quantify pain based on observable facial cues.[5,6]

Facial expression pain behavior is primarily assessed through the study of Action Units (AUs). These AUs reflect involuntary muscle activity that occurs in response to pain stimuli. They are not consciously controlled by the individual, but rather triggered by the body's natural response to discomfort or distress.[5] The Facial Action Coding System (FACS) is a set of numerical codes developed by Ekman and Friesen to identify and describe the different muscular movements of the human face.[12] This system divides the human face into anatomical regions and identifies specific muscle actions associated with different emotions and facial expressions. For example, certain AUs may indicate a genuine smile, while others may suggest sadness, anger, or surprise. The Prkachin and Solomon Pain Intensity (PSPI) is a pain expression score based on FACS.[13]

Notably, automated facial recognition systems have been developed using AI methods, thus reducing the necessity for the high level of skills and training traditionally required for manual assessment of AU expression.[14,15] In this scenario, by implementing AI technologies, AUs can be used for objective pain studying.[16,17] This approach can offer the potential to augment conventional pain assessment methods, providing healthcare providers with valuable insights into patient well-being and enabling more timely and targeted interventions.[18]

In this research, we aim to demonstrate that You Only Look Once (YOLO), a real-time CV object detection model, is applicable for pain detection through facial expressions. While previous approaches have demonstrated acceptable performance in pain detection using APA models,[5,10,15,17] the use of YOLO offers several advantages, particularly when combined with AUs. Moreover, since YOLO is specifically designed to handle real-time detection and localization tasks, it seems well-suited for dynamic clinical environments where pain needs to be detected in real time by capturing the spatial relationships between multiple facial features in pain expressions.

Methods

The study's methodology involves the development of a data-driven AI model. The CV algorithm YOLOv8 was implemented. The focus of the study, facial expressions of pain, was analyzed through a feature engineering process. Finally, the resulting model was trained and tested (Figure 1).

Deep Learning Architecture

Due to its remarkable efficiency in identifying multiple objects within an image, YOLOv8 is one of the most utilized real-time object detection algorithms.[19,20] Its architecture's backbone, *Darknet-53*, is a robust CNN used for high-complex tasks in the field of feature extraction. For example, it is implemented as multiclass object detection in research on intelligent vehicles.[21] Additionally, YOLOv8 employs a feature pyramid network (FPN) paradigm, which enhances its ability to perceive objects across varying sizes and spatial resolutions.[19] This property can be particularly advantageous for detecting pain expressions across diverse facial morphologies and image contexts. In this study, we employed a transfer learning approach, starting with pre-trained YOLOv8 weights from the common object in context (COCO) dataset, used for different CV tasks.[6,7] We then fine-tuned these weights using our custom pain/no-pain dataset, allowing the model to retain the general object detection capabilities while specializing in detecting pain expressions. This process was optimized with 10 epochs, a learning rate of 0.01, and a batch size of 2, using Adam as the optimizer.

The YOLO algorithm operates by dividing the input image into a grid of $S \times S$ cells. Each grid cell is responsible for detecting objects whose centers fall within it. The model predicts a fixed number of bounding boxes per grid cell, along with confidence scores indicating the presence of an object and the accuracy of the predicted bounding box. Additionally, each cell predicts class probabilities for the object. Therefore, YOLO divides images into grids, and each grid cell

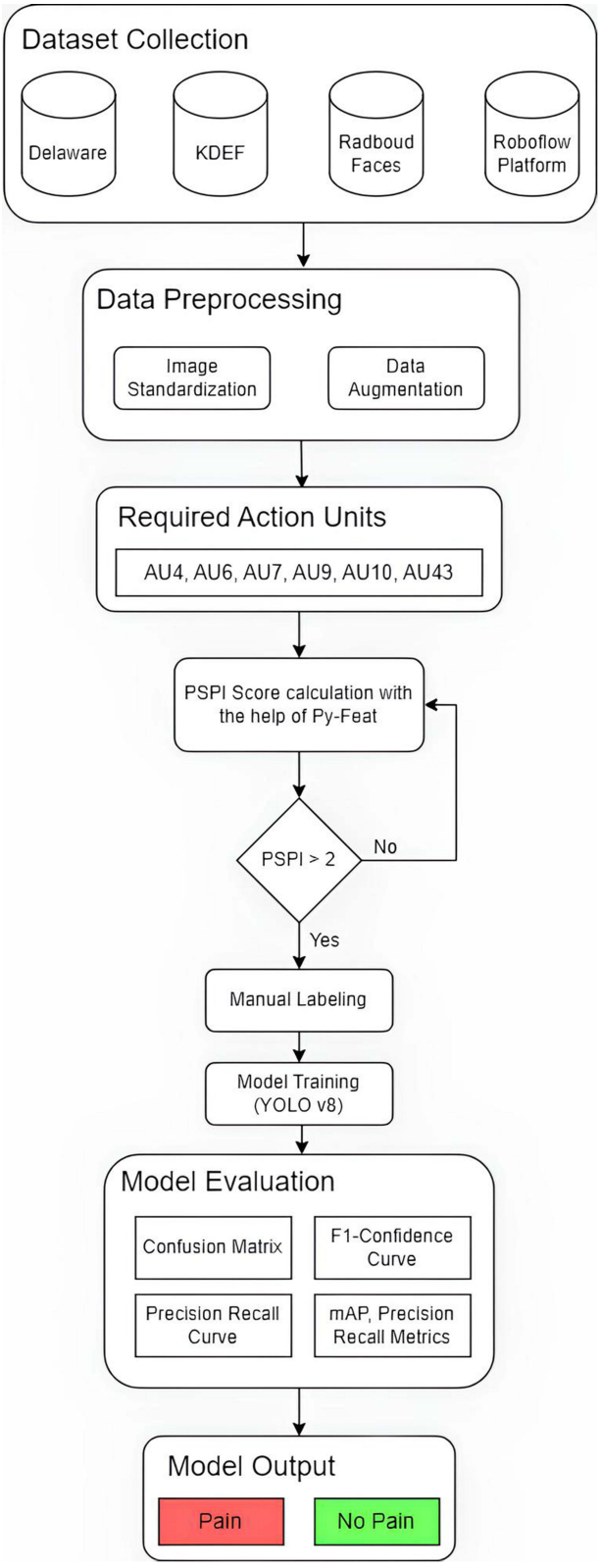

Figure 1 Study Flowchart.
Abbreviations: AU, action unit; PSPI, Prkachin and Solomon Pain Intensity.

predicts bounding boxes and class probabilities for objects. The class probabilities represent the likelihood of a specific object class being present in the predicted box.

For image division, given an input image of size $\boldsymbol{m}\times\boldsymbol{n}$, YOLO divides it into an $\boldsymbol{S}\times\boldsymbol{S}$ grid. For instance, with $\boldsymbol{S}=7$, the image is divided into 49 (7x7) cells. Concerning bounding box prediction, each grid cell predicts $\boldsymbol{B}$ bounding boxes. Therefore, each bounding box prediction includes:

- *(x,y)*: Coordinates of the bounding box center relative to the grid cell.
- *(w,h)*: Width and height of the bounding box, normalized by the width and height of the image.

The confidence (C) score is calculated as follow:[19,20]

$$C = Pr\,Pr\,(\text{Object}) \times IOU\frac{truth}{pred}$$

Where $IOU\frac{truth}{pred}$ is the Intersection over Union between the predicted box and the ground truth box.

Each grid cell also predicts class probabilities for $\boldsymbol{C}$ classes (class prediction):

$$P(C_i|Object)$$

For each bounding box, the final score (ie, final detections core) for each class-specific prediction is given by:

$$Score(C_i) = P\,(Object) \times C$$

Additionally, YOLO employs a multi-part loss function to optimize the model during training. This loss function comprises:

- Localization Loss: Measures errors in the predicted bounding box coordinates.
- Confidence Loss: Measures errors in the confidence score.
- Class Probability Loss: Measures errors in the predicted class probabilities.

This loss function ensures that the model is effectively trained to predict accurate bounding boxes, confidence scores, and class probabilities.

Action Units Selection

A "core" set of AUs was selected based on Prkachin's framework, identifying them as key indicators of pain expression.[13] The contraction of the muscles that bring the eyebrows together and downward is reflected in AU4 (Brow Lowering), which frequently denotes discomfort. AU6 (Cheek rising) is the rising of the cheeks in response to discomfort; this usually results in the crow's feet around the eyes. AU7 (Lid Tightening) and AU9 (Nose Wrinkling) are frequently linked to pain reactions. These AUs are characterized by the tightening of the muscles surrounding the eyes and the nose. The expressions AU10 (Upper Lip Raising) and AU43 (Eye Closure) denote lifting the upper lip and closing the eyes, respectively (Table 1).

Table 1 Action Units (AUs) Implemented

Action Unit (AU)	Definition	Description
AU4	Brow Lowerer	Distress or discomfort
AU6	Cheek Raiser	Mild pain feel
AU7	Lid Tightener	Uncomfortable sense
AU9	Nose Wrinkler	Heavy pain feel
AU10	Upper Lip Raiser	Raise in pain
AU43	Eyes Closed	Unwilling or unusual feeling

Datasets Implemented

To train and validate our model, we implemented four open-source pain datasets. This array of datasets was chosen to encompass diverse demographic profiles and clinical contexts. The Delaware Pain Dataset served as the cornerstone of our model development. It offers a collection of images portraying expressions of pain, captured in both controlled laboratory settings and real-world environments. The images were taken from 240 individuals, with a nearly equal representation of pain and no-pain expressions. The pain expressions were identified based on specific AUs related to discomfort, such as AU4 (Brow Lowering), AU6 (Cheek Raising), AU7 (Lid Tightening), and AU43 (Eye Closure). Most of our data came from this dataset due to its alignment with our study's focus on pain detection.[22]

Additionally, to further enrich our dataset and ensure its diversity, we incorporated the Karolinska Directed Emotional Faces (KDEF),[23] Radboud Faces Database (RaFD),[24] and Roboflow Platform (Open-Source Pain Images). The KDEF dataset consists of images representing seven basic emotions including happiness, sadness, anger, surprise, disgust, fear, and neutral expressions.[23] For our study, we focused on neutral expressions, which were labeled as "no-pain". These 50 neutral images were selected to supplement the no-pain category, ensuring a balanced dataset. The RaFD contains facial expressions showing various emotions and head poses.[24] We selected 20 images that exhibited neutral facial expressions and classified them under the "no-pain" category. Finally, we included images from the Roboflow open-source platform, specifically focusing on images with facial expressions indicative of pain. These 30 images were categorized as "pain". Given this strategy, we ensured that the dataset contained an equal number of pain (250 images) and no-pain (250 images) instances, creating a balanced distribution. The characteristics of the implemented datasets are listed in Table 2. Datasets overview and segregation are shown in Table 3.

Labeling Using Py-Feat for Action Unit and Emotion Detection

We manually analyzed and identified the AUs critical for recognizing pain in facial expressions. Each image in the dataset was meticulously labeled according to these AUs, ensuring that the model learned to recognize the specific

Table 2 Datasets Implemented

Dataset	Data Processed and Details	Ref [ ]
Delaware Pain Dataset	Images: 229 Details: Painful and neutral expressions, ratings on various pain dimensions. This dataset was chosen due to its alignment with the units of pain measurement utilized in our study.	[22]
Karolinska Directed Emotional Faces (KDEF)	Images: 490 Details: 7 emotions (happiness, sadness, anger, surprise, disgust, fear, neutral)	[23]
Radboud Faces Database	Images: 1783 Details: Facial expressions, emotions, facial actions, head poses.	[24]
Roboflow Platform (String: Facial Expressions)	Images: 772 Details: Angry, disgust, fear, happy, neutral, sad expressions.	[31]

Table 3 Dataset Overview and Splitting

Dataset	Total Images Used in Research	% of Dataset	Pain Images	No-Pain Images	Ref [ ]
Delaware Pain Dataset	400	80%	220	180	[22]
Karolinska Directed Emotional Faces (KDEF)	50	10%	0	50	[23]
Radboud Faces Database	20	4%	0	20	[24]
Roboflow Platform (String: Facial Expressions)	30	6%	30	0	[31]

features indicative of pain. For this aim, we utilized Py-Feat, an open-source toolkit designed for detecting facial AUs emotions, and facial landmarks.[25] Py-Feat integrates various models to recognize and quantify facial movements, emotions, and head poses. It was implemented to identify and chart 20 significant AUs, providing a comprehensive visual representation of these units based on their presence in the analyzed image. The tool also plots various emotions, including anger, disgust, surprise, happiness, sadness, fear (collectively indicated as the big six emotions), and neutrality, derived from the detected AUs (Figure 2).

However, a critical emotion, "Pain", is not directly identified by Py-Feat's standard emotion recognition process.[25] To address this limitation, we processed the output CSV file produced by Py-Feat (containing the detected AU values) for further analysis. Therefore, the entry data was used for calculating the pain intensity. For this aim, the PSPI score was calculated based on the intensities of specific AUs (AU04, AU06, AU07, AU09, AU10, AU43),[12] providing a quantitative measure of pain-related facial expressions.

In particular, to quantify the pain intensity, we extracted the values of the specific AUs from the output CSV file. The pain intensity was then computed using the PSPI score. The formula for this calculation is:

This formula incorporates the intensities of AU4, AU43, and the maximum values of AU6/AU7 and AU9/AU10, reflecting the additive nature of these AUs in pain expressions. AU6 and AU7 (cheek rising and lid tightening), as well as AU9 and AU10 (nose wrinkling and upper lip raising), were grouped due to their frequent co-occurrence during pain expressions. AU4 (brow lowering) and AU43 (eye closure) are key pain indicators that contribute separately to the pain score.[5,11–13]

The PSPI score was used as a threshold to determine which images exhibit significant pain expressions. Specifically, images with a PSPI intensity score greater than 2 were selected for training our model. These images were then labeled accordingly, confirming and marking those with higher pain expressions. This threshold ensures that only images with noticeable pain indicators are used in the training dataset, thereby enhancing the model's ability to accurately recognize and label pain in facial expressions (Figure 3).

After calculating the PSPI score for each image and identifying those with a score greater than 2, the images were reviewed to confirm whether they met the criteria for expressing pain. If an image satisfied the intensity of pain, it was labeled as "pain" Conversely, if the image did not meet the pain intensity criteria, was labeled as "no_pain". The open-source program Makesense.ai was implemented to manually label images based on the PSPI intensity score.[26] This

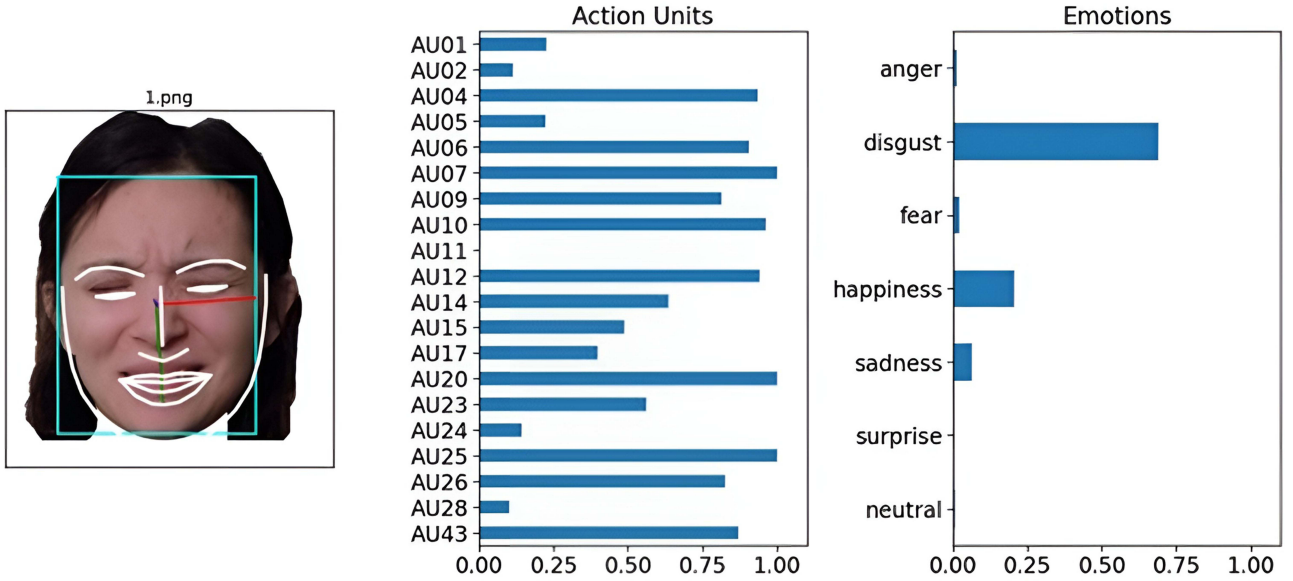

Figure 2 The output generated by Py-Feat after detecting the action units (AUs) and emotions. The X-axis represents the normalized value of the detected Aus and emotions, ranging from 0 to 1. When the emotion is very intense, the value will be 1; otherwise, it will fall somewhere within this range.
Notes: Images adapted from Mende-Siedlecki P, Qu-Lee J, Lin J, Drain A, Goharzad A. The Delaware Pain Database: a set of painful expressions and corresponding norming data. *Pain Rep*. 2020;5:e853.[22]

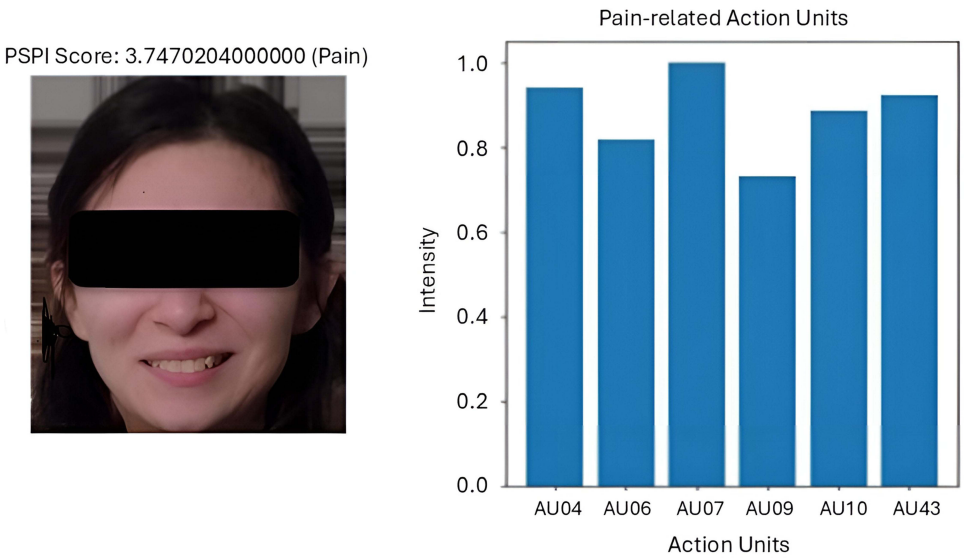

Figure 3 Prkachin and Solomon Pain Intensity (PSPI) score (in the frame, 3.747) and pain-related Action Units intensities (bar plot). The results are used as a threshold to determine which images exhibit significant pain expressions.
Notes: Images adapted from Mende-Siedlecki P, Qu-Lee J, Lin J, Drain A, Goharzad A. The Delaware Pain Database: a set of painful expressions and corresponding norming data. *Pain Rep*. 2020;5:e853.[22]

Figure 4 Makesense.ai implementation to manually label images based on the Prkachin and Solomon Pain Intensity (PSPI) score. The four images in the figure show different facial expressions of a subject, with bounding boxes drawn around the face. In each image, a bounding box is drawn to highlight the face, and different colors of boxes indicate different labels or levels of pain intensity assigned during the manual annotation process. There is a progression from a neutral expression (top row) to more intense facial expressions associated with pain (bottom row).
Notes: Images adapted from Mende-Siedlecki P, Qu-Lee J, Lin J, Drain A, Goharzad A. The Delaware Pain Database: a set of painful expressions and corresponding norming data. *Pain Rep*. 2020;5:e853.[22]

manual labeling process ensures accuracy in the dataset by verifying and categorizing each image based on the established pain threshold, thus contributing to the effectiveness of the model training (Figure 4).

Software and Libraries Implemented

The software and libraries used for this project were primarily written in Python 3.6 (Python Software Foundation, United States). Firstly, the process requires the Python libraries `py-feat` and `seaborn`. Next, the `Detector` class from Py-Feat is initialized with models for face detection (`retinaface`), landmark detection (`mobilefacenet`), AU detection (`xgb`), emotion recognition (`resmasknet`), and face pose estimation (`img2pose`). The target image is then specified and displayed using Py-Feat's `imshow` function. The `detect_image` method analyzes the image to detect facial features, AUs, emotions, and poses, with results printed to the console. These results are saved to a CSV file and read into a Pandas DataFrame for further analysis.

Performance Assessment

To mitigate potential biases and enhance model generalization, we standardized the dataset (n=3274) by selecting 500 images, evenly distributed between pain and no-pain instances, for training and validation purposes.[27] Specifically, the dataset was divided into 68% training data (340 images) and 32% validation data (160 images). The training set consisted of 170 pain and 170 no-pain images, while the validation set included 80 pain and 80 no-pain images.

In this study, we opted to train the YOLOv8 model for 10 epochs to achieve a balance between computational efficiency and model performance. This choice was guided by the observation that training over a smaller number of epochs allows for quicker iterations and timely adjustments, which is particularly beneficial when working with a diverse dataset like ours. Additionally, by selecting 10 epochs, we aimed to minimize the risk of overfitting, ensuring that the model learns the salient features necessary for pain detection without becoming overly tailored to the training data. The validation set represents unseen data by the model, and the performance metrics provided in this paper—such as precision, recall, and mAP—are derived from the validation data, not from the training set. Therefore, validation data serves as a proxy for how well the model can generalize to unseen pain expressions. These validation metrics are critical for assessing the model's ability to detect pain in real-world scenarios, which aligns with the objectives of our research. Furthermore, the model was tested using the BioVid Pain Expression Dataset, where subjects experienced real-time pain. This evaluation process was essential for validating the model's practical application in real-world scenarios.

Results

During the training process (epochs), the model exhibited consistent improvements in performance, demonstrating significant enhancements in accuracy and efficiency in pain detection tasks. In epoch 1, the model demonstrated a precision of 0.496, recall of 0.871, mAP50 of 0.519, and mAP50-95 of 0.355. Subsequent epochs witnessed incremental enhancements in precision, recall, and mAP scores, culminating in epoch 10, where the model achieved a precision of 0.818 and recall of 0.807, with a mAP50 of 0.874 (Figure 5).

Details on metrics of the training phase are reported in Table 4.

The model achieved impressive precision and recall scores for both pain and no-pain classes. Specifically, the precision for pain detection was 0.868. Similarly, the precision for no-pain instances was 0.919, signifying a high level of accuracy in identifying non-painful expressions (Figure 6). When considering all classes collectively, our model attained a mean Average Precision (mAP) of 0.893 at a threshold of 0.50.

The F1-confidence curve provides valuable insights into the performance of the pain detection model across different confidence thresholds. As illustrated in Figure 7, the F1 scores for the classes "pain", "nopain", and "all classes" reached a peak value of 0.80 at a confidence threshold of 0.285.

The confusion matrix describes the performance of the pain detection model in classifying instances as either pain or no-pain. The rows represent the actual values, while the columns represent the predicted values. Among the results, 65% of actual pain instances were correctly classified as pain (True Positive, Pain). False Positive (No Pain) was 11%; False Negative (Pain) 35%; and True Negative (No Pain) 89% (Figure 8).

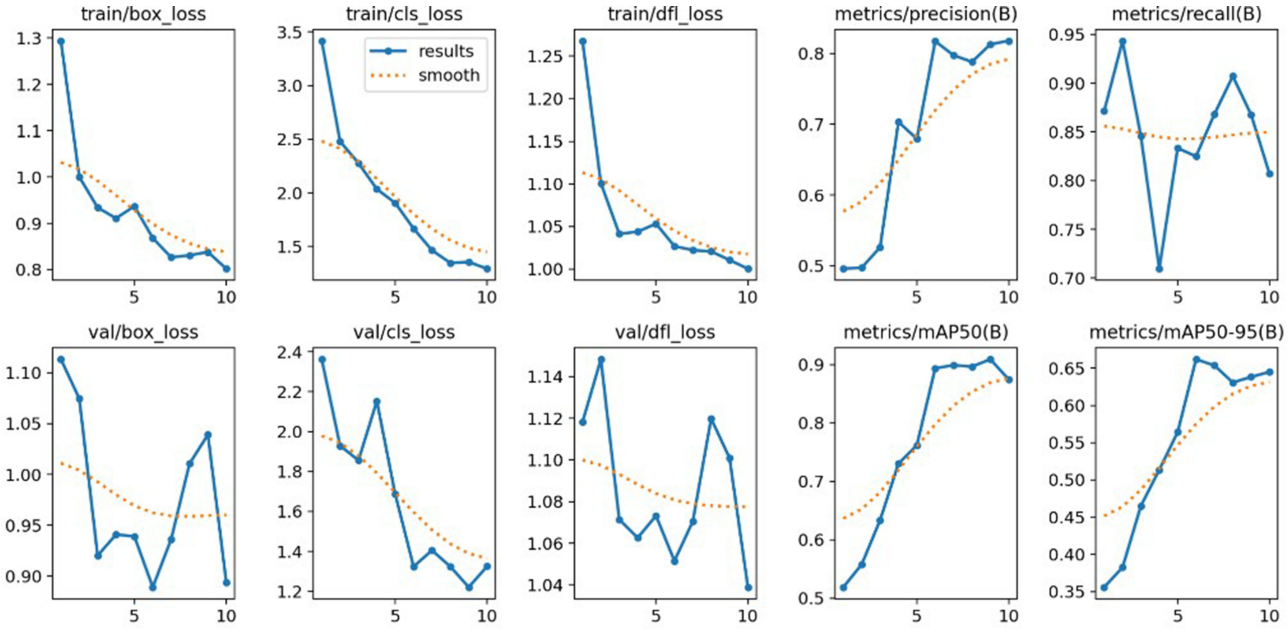

Figure 5 Training process. The solid blue lines in each graph represent the actual results observed during training and validation, while the dotted Orange lines denote smoothed trends, providing a clearer view of the overall performance trajectory across epochs. The training box loss graph shows a decreasing trend, starting from approximately 1.3 and dropping to around 0.8 over the epochs. This indicates that the model is improving in terms of predicting bounding boxes with greater accuracy as training progresses. The training classification loss starts high at around 3.5 and steadily decreases to about 2.0. This reduction signifies that the model is becoming more proficient at correctly classifying the data as training continues. Concerning the distribution Focal Loss (dfl), this loss metric decreases from 1.25 to about 0.85, showing that the model is increasingly accurate in focusing on difficult-to-classify examples. Moreover, the precision metric shows an upward trend, improving from 0.6 to 0.8 whereas recall improves from 0.85 to 0.95, reflecting the model's increasing ability to correctly identify true positives. The validation box loss fluctuates but generally remains between 0.90 and 1.05. Despite some variability, the overall trend suggests stabilization in the model's performance on unseen data. The classification (cls) loss varies significantly between 2.3 and 1.4, indicating some inconsistency in the model's classification performance on the validation set. Like other losses, distribution focal loss (dfl) fluctuates around 1.0 to 1.15, showing variability but an overall trend that suggests the model is still learning. Furthermore, the mean Average Precision at 50% (mAP50) Intersection over Union (IoU) threshold steadily increases from 0.6 to 0.85, indicating improving performance in detecting objects with a reasonable level of overlap whereas mAP50-95, shows an upward trend from 0.35 to 0.65, demonstrating that the model's performance is improving across a range of overlap levels. These metrics collectively illustrate that the model is progressively learning and improving in both the training and validation phases, albeit with some variability in the validation losses.

The impact of dataset size on model performance is shown in Figure 9. With the inclusion of more data, the model correctly identified 81% of actual pain instances as pain, indicating a substantial improvement from previous iterations (True Positive, Pain). False Positive (No Pain) was 23%, False Negative (Pain) 19%, and True Negative (No Pain) 77%.

Table 4 Metrics of Training Phase

Epoch	Precision	Recall	mAP50	mAP50-95
1	0.496	0.87125	0.51868	0.35583
2	0.49748	0.9434	0.55783	0.38238
3	0.52638	0.84558	0.63356	0.46508
4	0.70322	0.70966	0.73051	0.51344
5	0.67949	0.83333	0.76189	0.56419
6	0.8172	0.82496	0.8936	0.6619
7	0.79739	0.86824	0.89909	0.65357
8	0.78807	0.90713	0.89648	0.63077
9	0.81296	0.86749	0.90909	0.63819
10	0.81782	0.80697	0.87451	0.64498

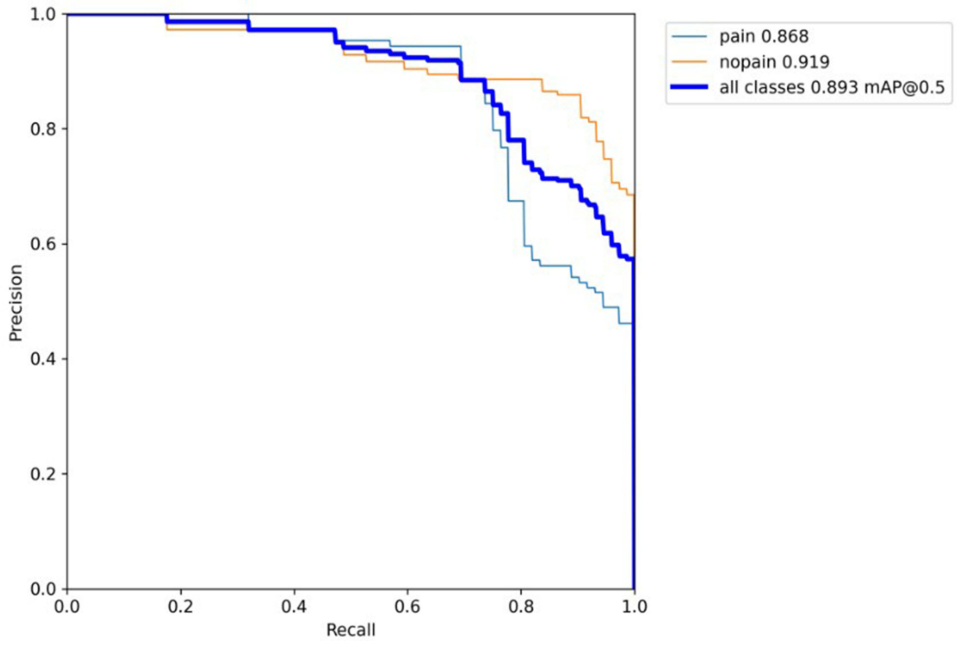

Figure 6 The Precision-Recall (PR) Curve chart illustrates the performance of our pain detection model across different thresholds. The PR curve showcases the trade-off between precision and recall at various classification thresholds. This provides insights into the model's ability to accurately classify pain and no-pain instances.

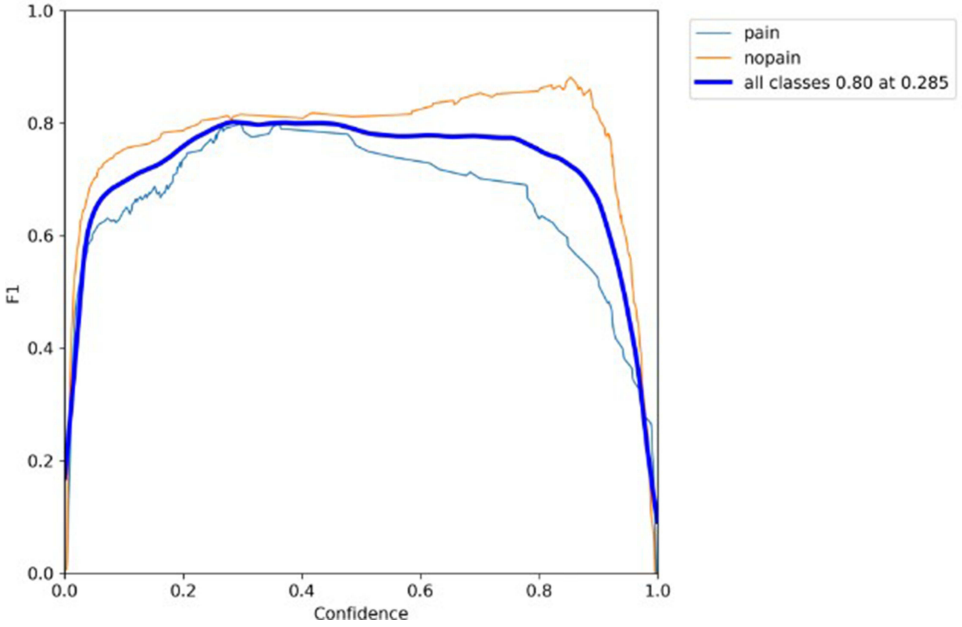

Figure 7 F1-Confidence Curve. The F1 scores (Y-axis) for the classes "pain", "nopain", and "all classes" reached a peak value of 0.80 at a confidence threshold of 0.285. The X-axis (Confidence) represents the confidence threshold at which predictions are made. It varies from 0 to 1, where 0 indicates no confidence and 1 indicates complete confidence in the prediction.

Table 5 illustrates the main performance metrics in the binary classification "pain vs no pain" after the inclusion of more data in the model.

Finally, we derived the output by testing the model with images it was not trained on. The image was taken from the Delaware Pain Database, which includes a set of painful expressions and corresponding norming data[22] (Figure 10). Not least, the model was also tested at the University of Salerno on a woman suffering from oncological pain (Figure 11). In the sequence of frames, it is evident that the model loses predictive accuracy. This result is likely attributable to facial

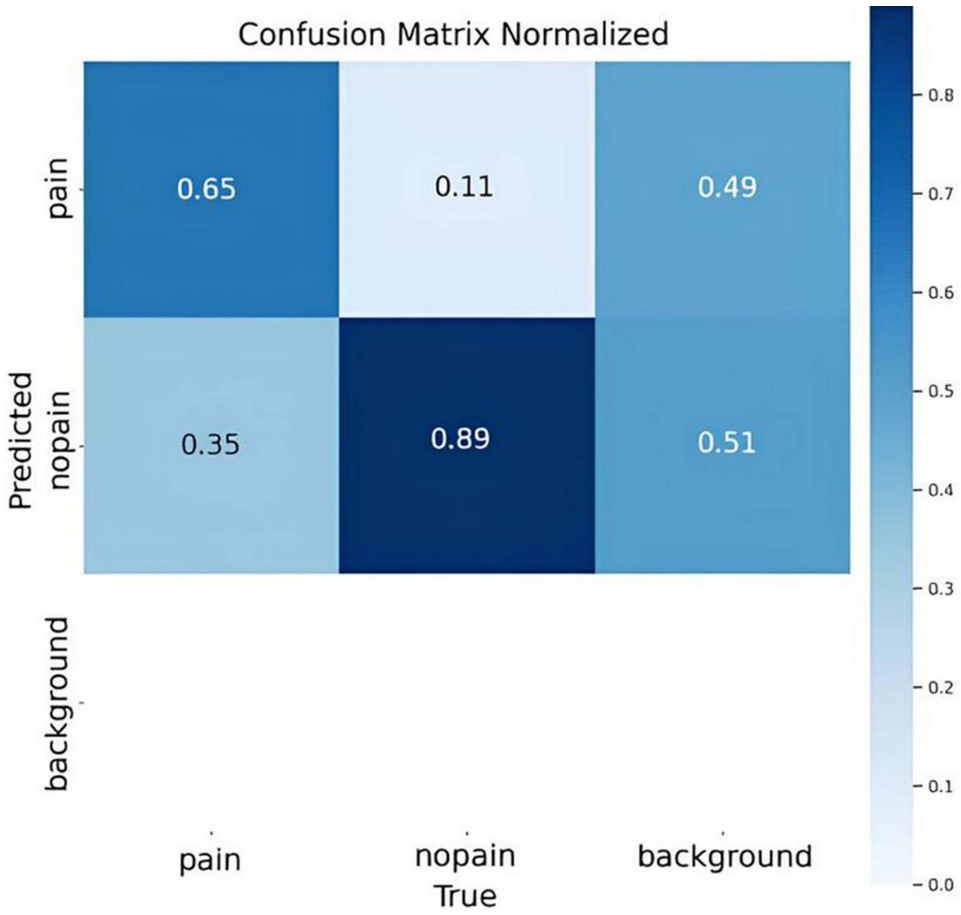

Figure 8 Confusion Matrix for 500 images. The rows in this matrix represent actual values, while the columns indicate predicted values. The model achieves a 65% accuracy in correctly identifying instances labeled as "pain" (true positive) and has a False Negative Rate of 0.35 for "pain" classifications. For "no-pain" predictions, the True Positive Rate is 0.89, and the False Positive Rate is 0.11, underscoring the model's reliability in distinguishing between pain and no-pain expressions. Inclusion of a "background" category, which, though not a target classification, is used to assess how the model handles non-relevant image regions within the binary classification framework. This category reflects areas in the input that do not correspond to either the pain or no-pain classifications. The near-equal distribution of background classifications between pain (49%) and no-pain (51%) predictions suggests balanced behavior when encountering non-facial regions, indicating that the model avoids systematic bias toward either category and is not overfitting to background features. This balanced handling of background regions reinforces the primary goal of detecting pain in facial images while maintaining neutral predictions for irrelevant content.

movements leading to the failure in detecting key AUs. This finding suggests that the experimental setting must be rigorous, with fixed camera angles and controlled conditions to minimize motion artifacts and ensure consistent detection of facial action units. Additionally, maintaining stable lighting and minimizing obstructions to the face are crucial to improving the model's reliability and accuracy.

Discussion

Our research serves as a proof-of-concept study demonstrating that object detection algorithms like YOLO can be effectively applied to pain detection through facial expressions. The balanced dataset approach we used allows us to effectively demonstrate the model's basic capability in distinguishing between pain and no-pain expressions. Our results (precision of 86.8% for pain detection and mAP of 0.893) suggest that YOLO shows promise as a viable approach for APA. Future research can build upon these findings by incorporating more complex dataset stratification strategies and population-representative test sets. In contrast, the CNN-based YOLO reframes object detection as a single regression problem, directly predicting bounding boxes and class probabilities from full images in one evaluation.[19–21] Therefore, in our investigation, by integrating the interpretative capabilities of the YOLOv8 model with the preprocessing of AUs through manual annotation, the methodology converges towards a more complete framework for APA analysis. For example, Darknet-53 offers advantages and insights into the intricate spatial relationships and hierarchical representations

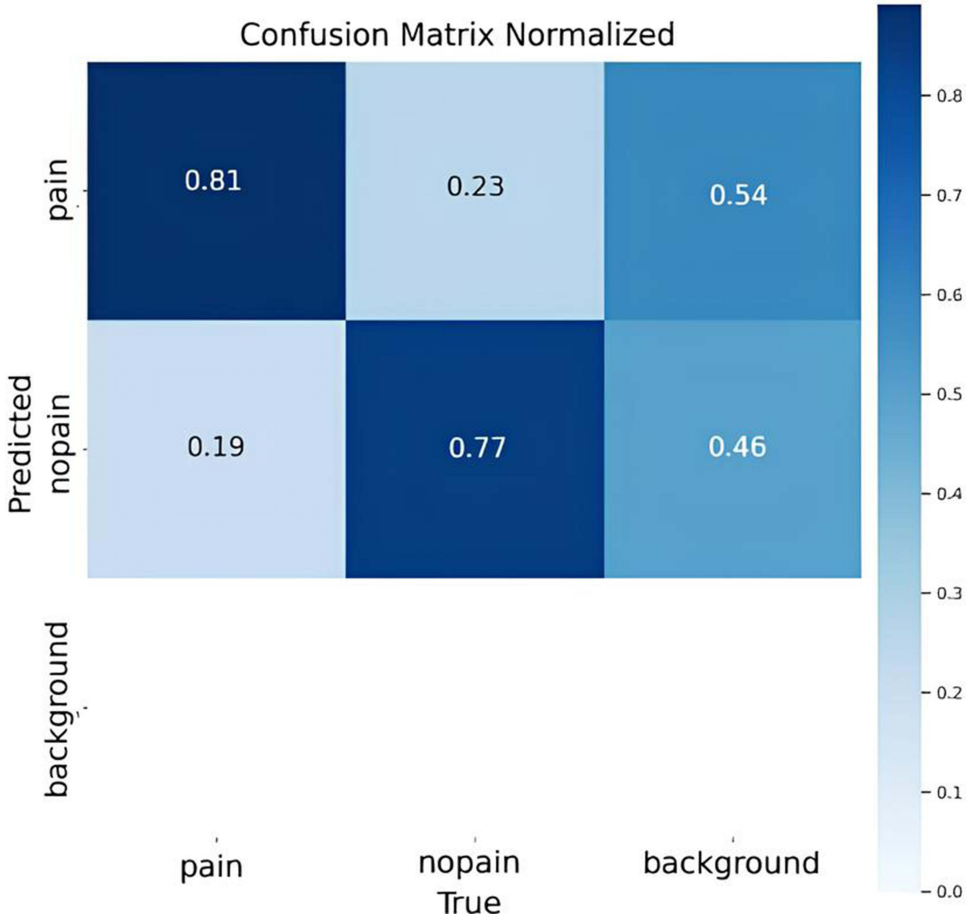

Figure 9 Change in size relates to change in matrices. The inclusion of more data has enhanced the model's accuracy, particularly in correctly identifying pain and no pain instances. True Positive (Pain) instances are 81%; moreover, the model mistakenly flags no pain as pain (False Positive) in 23%; False Negative (Pain) is relatively low (19%), meaning that the model minimizes the number of missed pain cases; and True Negative (No Pain) 77%.

inherent in facial images. This enables the model to capture subtle nuances indicative of pain expressions, facilitating the accurate detection and localization of relevant facial features associated with discomfort or distress. Additionally, YOLOv8 is recognized for its robustness in real-time object detection tasks.[28] Although several ANNs have been used for facial pain evaluation, to our knowledge, this is the first model built with YOLO. For instance, Ramis et al[29] implemented a CNN architecture encompassing 5 convolutional layers, 3 pooling layers, and two fully connected layers. Their ANN was set to receive 150×150 grayscale images as input and classify them into six classes corresponding to the big six emotions. Previously, other authors built their models on the CNN architectures VGG16 and VGG19[30] whereas a simple two-layer architecture was described in.[17] Importantly, the novelty of our approach was the training on pain datasets implemented for the process of facial expression recognition.[22–24,31] This step was of paramount importance for aligning with the study objective.

Table 5 Performance Metrics of After Inclusion of More Data

Epoch	Precision	Recall	F1-Score	Accuracy
Pain	0.78	0.81	0.79	0.79
No pain	0.80	0.77	0.78	

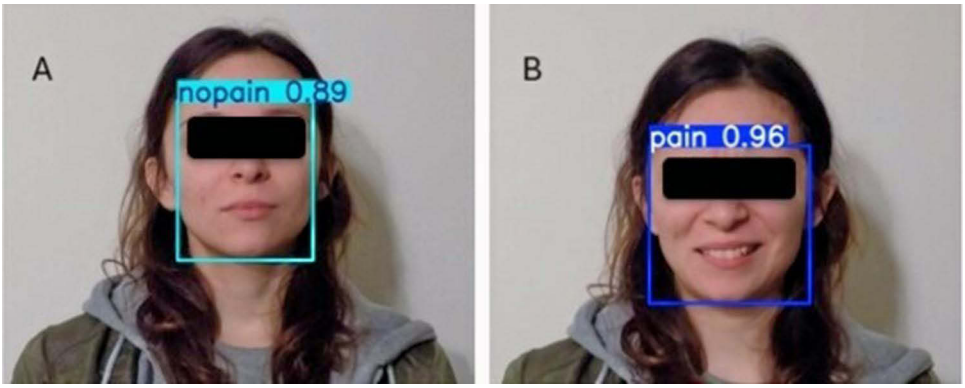

Figure 10 The images were processed using our YOLO v8 model, which correctly predicted "no pain" (**A**) and "pain" (**B**) with an accuracy of 89% and 96%, respectively. **Notes**: Images adapted from Mende-Siedlecki P, Qu-Lee J, Lin J, Drain A, Goharzad A. The Delaware Pain Database: a set of painful expressions and corresponding norming data. *Pain Rep*. 2020;5:e853.[22]

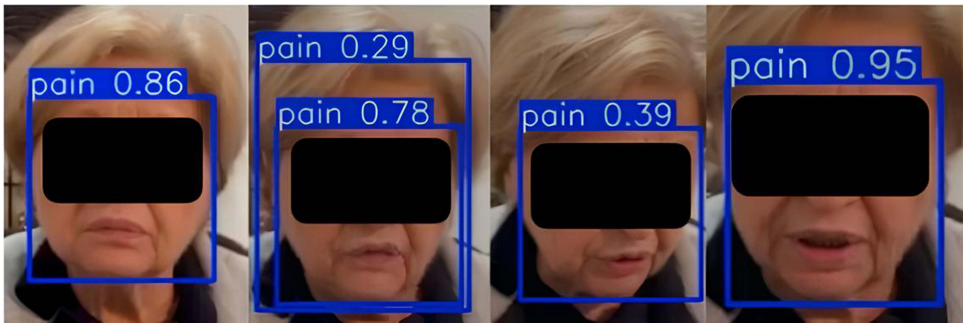

Figure 11 Model evaluation in a cancer patient (consent obtained). The sequence illustrates fluctuations in the model's detection accuracy, which could be attributed to facial movements, variations in expression, or other factors such as changes in lighting or angle. The variability in the pain scores underscores the importance of consistent experimental conditions to ensure reliable and accurate pain assessment.

Regarding feature processing, from the implemented datasets, we employed a manual labeling system on individual images. The strength of our approach lies in the use of a validated scoring method. Other researchers have explored different techniques. For example, Gomutbutra et al[32] used the open-source tool OpenFace© to generate time series data for each action unit, tracking patients' facial movements over time. This data was then transformed into two key metrics: average movement intensity and the area under the curve (AUC) surrounding the maximum peak. To calculate the AUC for each action unit, they analyzed data from 22 frames (0.03 seconds per frame) around the maximum peak.

Precision and recall are key metrics for evaluating the effectiveness of the model, as they demonstrate a balance between avoiding false positives (claiming pain when there is none) and minimizing false negatives (failing to detect pain when it exists). Specifically, precision refers to the proportion of correctly identified pain instances out of all the instances where the model predicted pain. In other words, it measures how accurate the model is when it indicates that a subject suffers from pain. Thus, a precision score of 0.81782 at epoch 10 underscores that about 81.8% of the time when the model predicted pain, it was correct. Recall, on the other hand, refers to the model's ability to detect pain when it is truly present. Specifically, it measures the actual pain instances that are correctly identified by the model. Therefore, a recall score of 0.80697 means that the model successfully identified about 80.7% of all real instances of pain. Remarkably, considering the mAP of 0.893 at a threshold of 0.5, the comprehensive metric accounts for the overall performance of the model across both pain and no-pain classes, providing a holistic assessment of its detection capabilities. Furthermore, the performance metrics across all epochs demonstrated the progressive improvement of our model's performance in pain detection tasks. Precision consistently increased with each epoch, indicating the model's ability to make accurate positive predictions. Similarly, recall remained high throughout training, suggesting the model's proficiency in capturing a large portion of positive instances (Figure 5). The mAP scores also exhibited

steady growth, reflecting the model's overall effectiveness in detecting pain expressions across different thresholds. The precision-recall curve chart and associated metrics demonstrated the effectiveness of our approach in accurately detecting pain expressions, showcasing the model's robust performance and high discriminative ability (Figure 6). Furthermore, the F1-confidence curve expressed the trade-off between precision and recall at varying confidence levels. A higher confidence threshold typically results in higher precision but lower recall, while a lower confidence threshold leads to higher recall at the expense of precision. Our model achieved a balance between precision and recall with an F1 score of 0.80 across all classes at the optimal confidence threshold of 0.285 (Figure 7). This indicates the robustness of our model in accurately detecting pain and no-pain instances, with a balanced performance in terms of both precision and recall. In other words, the high F1 scores underscore the effectiveness of our approach in pain detection tasks, providing confidence in the model's ability to generalize well to unseen data and make reliable predictions in real-world scenarios.

Training an AI-based model with additional data can significantly influence the model's performance. In our study, we incorporated a larger and more diverse dataset to enhance the model's ability to detect pain expressions accurately.[33] This augmentation aimed to provide the model with a richer understanding of various pain-related features, leading to improved performance metrics. Notably, with the inclusion of more data, the model correctly identified 81% of actual pain instances as pain, indicating a substantial improvement from previous iterations (True Positive, Pain). However, the model exhibited a slight increase in false positive predictions, with 23% of actual no-pain instances incorrectly classified as pain. This could be attributed to the increased complexity of the dataset, leading to a higher likelihood of misclassification. Conversely, the false negative rate decreased to 19%, indicating a reduction in instances where actual pain expressions were erroneously classified as no-pain. The true negative rate remained relatively stable at 77%, suggesting consistent performance in correctly identifying no-pain instances (Figures 8 and 9). These results highlight the model's enhanced sensitivity to pain-related features.

As illustrated in Figures 10 and 11, the evaluation of the test outcomes shows that the proposed approach achieved satisfactory results across multiple test cases. The consistency in performance across diverse scenarios underscores the robustness of the approach, although there were slight variations in efficiency due to the complexity of certain tasks and research settings. The results confirm that the model can be generalized effectively, but they also highlight areas where further optimization may enhance overall performance. Specifically, the observed decrease in accuracy during live detection can be attributed to factors such as uncontrolled facial movements, lighting variations, and camera angles. These factors can impair the accurate detection of Aus. Therefore, although promising, the model requires further refinements such as controlled environments, standardized lighting, and optimized camera positions. This improvement is mandatory to enhance real-world applicability.

Study Limitations

Despite the promising results, this study has several limitations that need to be addressed in future research. First, the dataset used for training and evaluation was limited in scope, which may affect the generalizability of the findings to broader contexts. Moreover, the approach's performance was tested primarily on simulated environments, which may not fully capture the complexities of real-world applications. Additionally, the computational resources required for the model can be substantial, which may limit its applicability in resource-constrained environments. Future studies should explore larger and more diverse datasets, real-world testing environments, and optimization techniques to improve the approach's scalability and efficiency. Finally, it will be essential to determine how the model truly works in terms of interpretability and explainability. This step is crucial for determining whether the model accurately detects genuine pain or can be misled by "pain-like" expressions. Incorporating biosignals could help reinforce the model by integrating physiological data with facial expression analysis to address this.[34]

Conclusion

For APA purposes, the combination of the CV YOLOv8 object detection algorithm with the insights provided by facial AUs can offer important insights. Given the focus on a subset of AUs, we developed a robust and reliable model that accurately identifies pain in facial expressions. Subsequently, through extensive experimentation and evaluation,

we demonstrated the effectiveness of our approach in accurately detecting pain expressions with high levels of accuracy. Despite limitations, the implications of this research are far-reaching, with potential applications in different scenarios.

Ethics

This manuscript can be exempted from ethics review since it qualifies as negligible risk research; it involves only existing collections of data or records that contain only non-identifiable data about human beings. For example, the Delaware Pain Dataset served as the cornerstone of our model development. It offers a collection of images portraying expressions of pain, captured in both controlled laboratory settings and real-world environments. The other datasets are listed in Table 1. Therefore, there is no need to request an opinion from our local Ethics Committee, as stated by the EU directions for data protection (Directive (EU) 2016/680 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, and on the free movement of such data, and repealing Council Framework Decision 2008/977/JHA; available at: http://data.europa.eu/eli/dir/2016/680/oj).

Disclosure

The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Meissner W, Huygen F, Neugebauer EAM. et al. Management of acute pain in the postoperative setting: the importance of quality indicators. *Curr Med Res Opin*. 2018;34(1):187–196. doi:10.1080/03007995.2017.1391081
2. Beltramini A, Milojevic K, Pateron D. Pain Assessment in Newborns, Infants, and Children. *Pediatr Ann*. 2017;46(10):e387–e395. doi:10.3928/19382359-20170921-03
3. Sabater-Gárriz Á, Molina-Mula J, Montoya P, Riquelme I. Pain assessment tools in adults with communication disorders: systematic review and meta-analysis. *BMC Neurol*. 2024;24(1):66. doi:10.1186/s12883-024-03539-w
4. Devlin JW, Skrobik Y, Gélinas C, et al. Clinical Practice Guidelines for the Prevention and Management of Pain, Agitation/Sedation, Delirium, Immobility, and Sleep Disruption in Adult Patients in the ICU. *Crit Care Med*. 2018;46(9):e825–e873. doi:10.1097/CCM.0000000000003299
5. Cascella M, Schiavo D, Cuomo A, et al. Artificial Intelligence for Automatic Pain Assessment: research Methods and Perspectives. *Pain Res Manag*. 2023;2023:6018736. doi:10.1155/2023/6018736
6. El-Tallawy SN, Pergolizzi JV, Vasiliu-Feltes I, et al. Incorporation of "Artificial Intelligence" for Objective Pain Assessment: a Comprehensive Review. *Pain Ther*. 2024;13(3):293–317. doi:10.1007/s40122-024-00584-8
7. Elyan E, Vuttipittayamongkol P, Johnston P, et al. Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward. *Art Int Surg*. 2022;2:24–45. doi:10.20517/ais.2021.15
8. Chai J, Zeng H, Li A, Ngai EWT. Deep learning in computer vision: a critical review of emerging techniques and application scenarios. *Machine Learning Appl*. 2021;6:100134. doi:10.1016/j.mlwa.2021.100134
9. Albawi S, Mohammed TA, Al-Zawi S Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.
10. Benavent-Lledo M, Mulero-Pérez D, Ortiz-Perez D, et al. A Comprehensive Study on Pain Assessment from Multimodal Sensor Data. *Sensors*. 2023;23(24):9675. doi:10.3390/s23249675
11. Chambers CT, Mogil JS. Ontogeny and phylogeny of facial expression of pain. *Pain*. 2015;156(5):798–799. doi:10.1097/j.pain.0000000000000133
12. Ekman P, Friesen WV. Measuring facial movement. *Environmental Psychology and Nonverbal Behavior*. 1976;1(1):56–75. doi:10.1007/BF01115465
13. Prkachin KM, Solomon PE. The structure, reliability and validity of pain expression: evidence from patients with shoulder pain. *Pain*. 2008;139(2):267–274. doi:10.1016/j.pain.2008.04.010
14. Castellano G, De Carolis B, Macchiarulo N. Automatic facial emotion recognition at the COVID-19 pandemic time. *Multimed Tools Appl*. 2023;82(9):12751–12769. doi:10.1007/s11042-022-14050-0
15. Samadiani N, Huang G, Cai B, et al. A Review on Automatic Facial Expression Recognition Systems Assisted by Multimodal Sensor Data. *Sensors*. 2019;19(8):1863. doi:10.3390/s19081863
16. Park I, Park JH, Yoon J, et al. Artificial intelligence model predicting postoperative pain using facial expressions: a pilot study. *J Clin Monit Comput*. 2023;38:261–270. doi:10.1007/s10877-023-01100-7
17. Cascella M, Vitale VN, Mariani F, Iuorio M, Cutugno F. Development of a binary classifier model from extended facial codes toward video-based pain recognition in cancer patients. *Scand J Pain*. 2023;23(4):638–645. doi:10.1515/sjpain-2023-0011
18. Dawes TR, Eden-Green B, Rosten C, et al. Objectively measuring pain using facial expression: is the technology finally ready? *Pain Manag*. 2018;8(2):105–113. doi:10.2217/pmt-2017-0049
19. Terven J, Córdova-Esparza D-M, Romero-González J-A. A Comprehensive Review of YOLO Architectures in Computer Vision: from YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*. 2023;5(4):1680–1716. doi:10.3390/make5040083
20. Redmon J, Farhadi A. YOLOv3: an Incremental Improvement. *arXiv*. 2018;arXiv:1804.02767.

21. Yang L, Chen G, Ci W. Multiclass objects detection algorithm using DarkNet-53 and DenseNet for intelligent vehicles. *EURASIP J Adv Signal Process*. 2023;2023(1):85. doi:10.1186/s13634-023-01045-8
22. Mende-Siedlecki P, Qu-Lee J, Lin J, Drain A, Goharzad A. The Delaware Pain Database: a set of painful expressions and corresponding norming data. *Pain Rep*. 2020;5:e853.
23. Lundqvist D, Flykt A, Öhman A. Karolinska Directed Emotional Faces. *APA PsycTests*. 1998. doi:10.1037/t27732-000
24. Verpaalen IAM, Bijsterbosch G, Mobach L, Bijlstra G, Rinck M, Klein AM. Validating the Radboud faces database from a child's perspective. *Cogn Emot*. 2019;33(8):1531–1547. doi:10.1080/02699931.2019.1577220
25. Cheong JH, Jolly E, Xie T, Byrne S, Kenney M, Chang LJ. Py-feat: python facial expression analysis toolbox. *Affect Sci*. 2023;4(4):781–796. doi:10.1007/s42761-023-00191-4
26. Makesense.ai. Available from: https://www.makesense.ai/. Accessed July 29, 2024.
27. Cascella M, Shariff MN. PAIN_CV_DATASET [Data set]. *Zenodo*. 2024. doi:10.5281/zenodo.13327991
28. Chen W, Huang H, Peng S, et al. YOLO-face: a real-time face detector. *Visual Comput*. 2021;37(4):805–813. doi:10.1007/s00371-020-01831-7
29. Ramis S, Buades JM, Perales FJ, Manresa-Yee C. A novel approach to cross dataset studies in facial expression recognition. *Multimedia Tools Appl*. 2022;81(27):39507–39544. doi:10.1007/s11042-022-13117-2
30. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv*. 2014:1409–1556. arXiv preprint.
31. Roboflow (string: facial emotions). Available from: https://universe.roboflow.com/search?q=facial%20emotions. Accessed February 18, 2024.
32. Gomutbutra P, Kittisares A, Sanguansri A, et al. Classification of elderly pain severity from automated video clip facial action unit analysis: a study from a Thai data repository. *Front Artif Intell*. 2022;5:942248. doi:10.3389/frai.2022.942248
33. Menchetti G, Chen Z, Wilkie DJ, Ansari R, Yardimci Y, Çetin AE Pain detection from facial videos using two-stage deep learning. In: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 1–5. IEEE. DOI: 10.1109/GlobalSIP45357.2019.8969274.
34. Cascella M, Di Gennaro P, Crispo A, et al. Advancing the integration of biosignal-based automated pain assessment methods into a comprehensive model for addressing cancer pain. *BMC Palliat Care*. 2024;23(1):198. doi:10.1186/s12904-024-01526-z.