Instance Segmentation of Human Body Parts Using YOLOv8: Design and Implementation of a Web-Based System

Maya Silvi Lydia[1*], Anandhini Medianty Nababan[1], Elviawaty Muisa Zamzami[1], Amru Khair Al Hakim[1], Desilia Selvida[2], Pauzi Ibrahim Nainggolan[2], Dhani Syahputra Bukit[3], Lanova Dwi Arde M[3] and Rahmita Wirza O.K. Rahmat[4]
[1] Department of Computer Science, Universitas Sumatera Utara, Medan, 20155, Indonesia
[2] Computer Vision and Multimedia Laboratory, Universitas Sumatera Utara, Medan, 20155, Indonesia
[3] Department of Public Health, Universitas Sumatera Utara, Medan, 20155, Indonesia
[4] Department of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia
E-mail: maya.silvi@usu.ac.id
[*]Corresponding author

Keywords: instance segmentation, YOLOv8, human body, deep learning, detection

Received: June 23, 2025

Image segmentation extracts meaningful structure from images. This study presents a YOLOv8-based instance segmentation approach and a web-based system that partitions human images into the head, body, right arm, and left arm. We curated 107 manually annotated images of female university students aged 19–22, captured under controlled poses; the dataset was split into 92/10/5 images for training/validation/testing. To improve robustness, we applied augmentation (rotation, shear, brightness and contrast adjustment, darkening, and noise). The model was trained and evaluated, yielding its best performance at 200 epochs with mAP@0.50 of 0.979, precision 0.914, recall 0.995, and F1-score 0.95. We implemented the system as a web app with a Flask backend and HTML/CSS/JavaScript frontend that accepts uploads, runs segmentation, displays masks and confidence scores, and enables downloads. The proposed design supports downstream tasks requiring fine-grained human-part analysis. While results are strong within this limited and homogeneous cohort, we note reduced reliability for overlapping limbs and emphasize the need for broader data—diverse demographics, clothing, and poses—to assess generalizability. Training was conducted on Google Colab for reproducibility.

Povzetek: Predstavljen je YOLOv8-temeljen sistem za segmentacijo delov človeškega telesa s spletno aplikacijo, ki dosega visoko natančnost, a zahteva širše podatke za boljšo posplošljivost.

1 Introduction

Health technology innovations that utilize artificial intelligence (AI) are growing as multidisciplinary research increasingly explores its applications in predictive medicine, health services management, and clinical decision-making [1, 2], with a growing focus on data-driven healthcare solutions [1, 3]. Previous studies in machine learning models are used to classify stunting status and predict children's growth, enabling early detection and intervention through algorithms such as Random Forest and Extra Trees [4]. More broadly, the Rapid development of digital technology has enabled computers to perform increasingly sophisticated tasks, accelerating progress in AI, particularly in computer vision.

Computer vision is a part of AI that focuses on interpreting images or video streams to support decision-making and perform specific tasks. It encompasses several subfields, image segmentation, including image classification, and object detection [5]. Image segmentation represents a fundamental approach within the field of image analysis, wherein an image is systematically divided into distinct regions based on shared visual characteristics. This process allows researchers to extract structured, meaningful, and interpretable information from each segmented area, thereby supporting deeper analytical insights and informed decision-making in various applications. [6]. When applied to the segmentation of human body parts, this process becomes essential for facilitating more detailed recognition and examination of specific anatomical regions, such as the head, body, right arm, and left arm.

Human body parts possess distinct geometric characteristics that enable precise identification. The distinctions can be utilized in computer vision and related computational methods to identify and analyze various anatomical structures effectively [7]. Segmentation of the human body has its own challenges due to the complexity of the varied shapes and bodies. Therefore, a method that is able to segment accurately and efficiently is needed.

Image segmentation is classified into two principal categories: semantic segmentation, which assigns a class label to every pixel within an image, and instance segmentation, which further distinguishes between individual objects belonging to the same class, thereby enabling more granular analysis [8]. As a part of computer vision, Instance segmentation work integrates object detection with segmentation techniques. It not only identifies and locates objects within an image but also generates a pixel-wise segmentation mask for each detected entity [8, 9].

Instance segmentation not only focus identifies the boundaries of objects but also distinguishes between instances of the same object. This is very important in the analysis of the human body, as it allows the identification and analysis of each individual body part in more detail. Instance segmentation is useful for measuring the size of detected objects, cropping them from their background, and more accurately detecting objects. In this study, the decomposition of human body parts intends to break down the human image into more detailed segments, such as the head, body, right arm, and left arm. Human parsing is a Fundamental visual comprehension task that requires the segmentation of human images into clear body parts [14]. Research related to human parsing is often utilized in the fashion industry for style analysis [10].

The most significant and popular advancement in human body segmentation is the You Only Look Once (YOLO) framework, a groundbreaking single-stage object detection algorithm recognized for its real-time efficiency and high accuracy [11]. YOLO establishes a single, unified architecture for dividing images into bounding boxes and calculating class probabilities for each box when compared to previous object identification approaches, such as R-CNN [28]. Using YOLO shows that execution is much faster and more precise. This algorithm can also accurately predict images or illustrations [12]. The YOLOv8 method will be implemented in this study to divide the human image into more detailed parts. YOLOv8 offers object detection capabilities with a higher level of speed, accuracy, and efficiency compared to previous versions [13, 15, 16].

Our study, which utilizes the YOLOv8 algorithm, aims to detect objects due to its high speed and accuracy and then employs the instance segmentation method to separate individual objects into distinct segments. The first step involves detecting objects using object detection algorithms, followed by refining segmentation through pixel-level classification; this method is expected to achieve accurate and detailed segmentation. This leverages the power of both tasks, improving overall performance and resulting in high-quality object masks. From the above explanation, this research will produce a web-based system to decompose parts of the human body into clear segments, such as the head, body, right arm, and left arm.

While significant progress has been achieved in instance segmentation for human body parts, the effectiveness and robustness of such systems depend greatly on the diversity and representativeness of the datasets used for training and evaluation. In this study, the dataset consists of manually annotated images of female students aged 19–22 in controlled poses and clothing. This limited scope provides a useful foundation for methodological exploration but also restricts the generalizability of the findings. The reported results, therefore, should be interpreted within the context of this specific and homogeneous dataset.

Consequently, the objective of this research is to develop and evaluate a web-based human body segmentation system using the YOLOv8 instance segmentation model, focusing on the head, body, right arm, and left arm. Although promising results are obtained within this dataset, future research with more diverse and representative datasets is necessary to assess the broader applicability and robustness of the proposed framework.

2 Related work

2.1 Related work summary table

We summarize key human body-part parsing and instance-level human analysis methods in Table 1 to position our contribution among prior work. The comparison spans task formulation (semantic vs. instance-level), canonical datasets (e.g., LIP [14], PASCAL-Person-Part, CIHP [22], MHP-v2.0), reported metrics (mIoU/AP), and deployment notes. This overview clarifies where prior art excels and where gaps remain for body-part instance segmentation suitable for lightweight, web-served applications.

Overall, CE2P, SCHP, PCNet/ACENet deliver strong semantic human parsing on LIP/PASCAL-Person-Part/CIHP, whereas PGN and Parsing R-CNN [22] target instance-level human parsing with heavier, often two-stage pipelines. Prior work seldom reports lightweight, web-deployed systems for per-instance body-part masks (including explicit left/right hands under occlusion). In contrast, our YOLOv8-based system achieves high accuracy on a constrained dataset and is implemented as a browser-facing application (Flask + HTML/CSS/JS), thereby addressing both the instance-level requirement and practical deployment considerations.

Table 1: Comparative summary of related work on human body-part parsing and instance-level analysis.

Method (Year)	**Task / Output**	**Dataset(s)**	**Metric & Score (type)**	**Application / Deployment**	**Limitations / Notes**
CE2P (AAAI 2019) [23]	Single & multi-person human parsing (semantic parts)	LIP, CIHP, MHP-v2.0	56.50% mIoU (LIP); 45.31% mean APr (CIHP); 33.34% APp@0.5 (MHP-v2.0)	Academic SOTA at the time	Strong parsing; instance-aware via challenge tracks; no web deployment discussed.
Parsing R-CNN (CVPR 2019) [22]	Instance-level human parsing (two-stage, region-based)	CIHP, MHP-v2.0, DensePose-COCO	SOTA on CIHP/MHP; 1st place DensePose-COCO (2018)	High-accuracy, region-based pipeline	Heavier two-stage design; not targeted for real-time web serving.
SCHP (Self-Correction Human Parsing, 2019) [24]	Single-person human parsing (semantic)	LIP, PASCAL-Person-Part	Best on LIP/PPP in paper; 1st in CVPR2019 LIP Challenge	Robust to noisy labels	Semantic (not per-instance masks); no web system reported.
PCNet (Part-Aware Context Network, CVPR 2020) [25]	Context Network, CVPR 2020) Human parsing (semantic, context-aware)	PASCAL-Person-Part, LIP, CIHP	Reports SOTA gains on these datasets	Context modeling for parts	Semantic parsing (not instance separation); offline evaluation.
ACENet (2020) [26]	Human parsing (semantic; affinity-aware)	LIP, PASCAL-Person-Part	58.1% mIoU (LIP)	Accuracy-oriented	Semantic parsing; not focused on web deployment or per-instance limbs.
PGN / Part Grouping Network (ECCV 2018) [27]	Instance-level human parsing (detection-free; parts + instance edges)	CIHP (introduced), PASCAL-Person-Part	Outperforms prior methods on PPP; strong results on CIHP	Joint parts & instance edges	Earlier instance-level approach; heavier than YOLO-style one-stage; no web deployment.
This Work — YOLOv8-seg (Web)	Instance-level human body-part segmentation (Head, Body, Right Hand, Left Hand)	107 images, female students 19–22; split 92/10/5 (train/val/test); controlled poses & clothing	mAP@0.50 = 0.979 (all classes) at 200 epochs	Web app (Flask backend; HTML/CSS/JS frontend); ~9–13 FPS on tested setup	Dataset is homogeneous; generalization limited; remaining errors under occlusion/overlap (hands vs torso)

2.1. Human body

The human body can be classified into two primaries sections: the trunk (truncus) and the limbs, which are further divided into upper and lower extremities. The part of trunk includes head, neck, and torso, with the torso itself comprising the chest, abdomen, and waist. Upper limbs are connected to the torso through shoulder cuffs, which consist of the clavicula (collarbone) and scapula (shoulder blades), which move with the torso. Two main part of the pelvic girdle composed by pelvic bones and a sacrum (lower spine), connecting the lower limbs with the torso [17].

As a complete biological structure, the human body is composed of multiple interconnected parts, including the head, neck, torso, arms, and legs. The head is the upper part of the body that contains the brain and features such as eyes, ears, nose, and mouth. Part of the neck connects the head with the torso. The whole trunk includes the chest and abdomen, where various vital organs such as the heart, lungs, and stomach are located. Two arms are located at the sides of the body and serve to perform various movements and tasks, while the two legs provide support and allow movement. The structure of the human body is very complex and diverse, with each part having an important role in maintaining overall health and balance.

2.2 Image segmentation

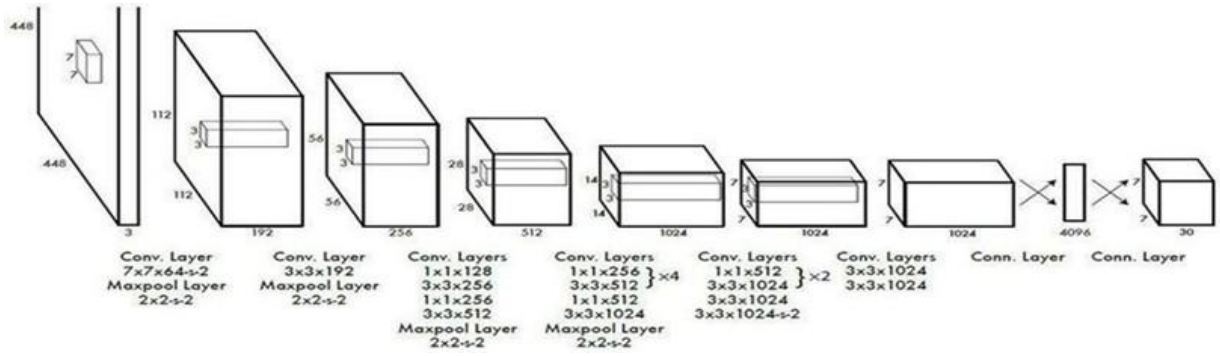


Figure 1: Proposed method using YOLOv8

Image segmentation is a core task in the field of computer vision, wherein a digital image is divided into multiple regions or segments by grouping together areas that share similar visual characteristics, such as color, texture, shape, or intensity This segmentation process enables the extraction of meaningful regions of interest (RoI's), facilitating a deeper analysis of an image's structure and the information it contains [6, 18].

Segmentation plays a critical role in various digital image processing tasks, including object identification, feature extraction, and visual content analysis. It serves as a key preprocessing step in domains such as autonomous driving, medical imaging, industrial inspection, and augmented reality [18]. The result of segments generally consists of non-overlapping pixel groups that are homogeneous and represent significant regions based on human visual perception.

Despite its importance, image segmentation presents challenges, particularly in defining what constitutes a "meaningful region" due to subjective visual interpretation, as well as in representing complex objects based solely on low-level features [18]. Generally, segmentation techniques are categorized into two types: (i) semantic segmentation, which labels each pixel according to its corresponding class, and (ii) instance segmentation, which differentiates between distinct objects belonging to the same class.

2.3 Instance segmentation

This segmentation is related to the precise identification of all objects present in a single image. Therefore, combining object detection, object location, and object classification is key elements in instance segmentation. In other words, this segmentation approach is focused on the goal of clearly distinguishing between each object that is categorized as a similar instance.

The main objective of instance segmentation is to recognize and delineate individual objects within an image [19]. Implementing this method is expected to enhance accuracy and efficiency while minimizing potential errors arising from the intricate nature of human body structures.

Instance segmentation has gained considerable attention in computer vision research, particularly for complex applications such as robotics, autonomous vehicles, and surveillance. Several instance segmentation frameworks have been proposed, and most of them leverage deep learning to improve segmentation accuracy exponentially. Generally, instance segmentation techniques can be classified into three broad categories: multi-stage approaches, single-stage approaches, and methods utilizing semi-supervised or weakly supervised learning [20].

2.4 You only look once (YOLO)

YOLO was initially proposed by Joseph Redmon et al in 2015 as an integrated, real-time object detection framework that treats detection as a single-stage regression task, directly mapping image pixels to bounding box coordinates and class probabilities. The original model demonstrated impressive speed, processing images at up to 155 fps, albeit with relatively lower localization accuracy compared to its contemporaries [21]. Since its first introduction, YOLO has undergone several iterations, with improvements in accuracy, speed, and features. Popular versions include YOLOv1, YOLOv2 (also known as YOLO9000), YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7 and the latest variants such as YOLOv8, as well as community-created implementations [29, 30, 31].

Overall, the YOLO framework functions by partitioning an image into a grid-like structure, where each grid cell is responsible for making several bounding box predictions along with a confidence score. This score represents the model's certainty regarding the presence of an object in the respective cell while also estimating the precision of the generated bounding box. In addition to generating bounding boxes and associated confidence scores, each grid cell in the YOLO framework is also tasked with predicting the class of the object it contains. However, one inherent limitation of this approach is that each cell is restricted to predicting only a single object class, which may reduce accuracy in scenarios involving

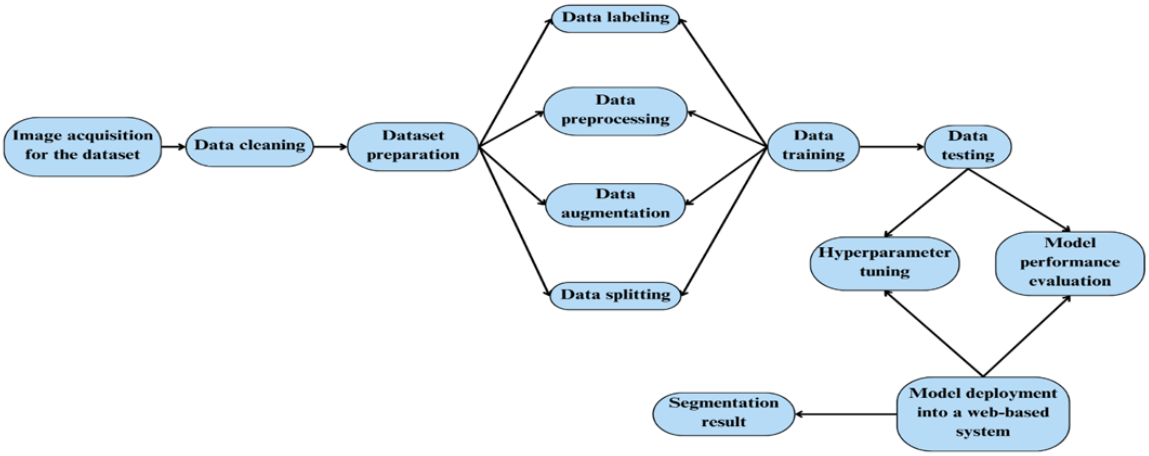


Figure 2: Research methodology

overlapping or closely positioned objects of different types. The model produces its output in the form of a multidimensional tensor that represents the entire grid structure superimposed on the input image. This tensor encapsulates the predicted class labels, bounding box coordinates, and confidence scores for each individual grid cell [11, 21]. A visual representation of the YOLO architecture, encompassing these components, is provided in Figure 1.

To assess whether an object is present in a predicted bounding box, the YOLO algorithm calculates a confidence score by multiplying the likelihood that an object exists in the box by the Intersection over Union (IoU) value. The IoU measures the extent of overlap between the predicted bounding box and the corresponding ground truth annotation used during training. As a standard evaluation metric in object detection tasks, IoU provides a quantitative measure of how accurately the model localizes objects within the image.

Moreover, the class confidence score obtained by combining the conditional probability of the object class with the bounding box's confidence score—plays a pivotal role in the model's final prediction. This score reflects both the probability of a specific class being present in the box and the degree of alignment between the predicted box and the actual object, offering a comprehensive indication of prediction reliability. This score reflects the level of confidence in a particular class for each bounding box, indicates the likelihood of a specific class in that box, and how well the prediction box matches the actual object.

3 Method

This study's methodological approach focuses on developing a human body segmentation system through YOLO-based instance segmentation. This methodology includes the stages of problem analysis, needs analysis, system architecture, and user interface design. Each stage is explained systematically to ensure seamless integration between the concepts used and the system's implementation.

3.1 Problem analysis

In this study, we used Problem analysis to identify the primary source of the problem; then, an in-depth examination was carried out on the issue that needs to be solved so that an efficient system can be developed. In this context, body recognition and analysis through images by means of segmentation is the primary focus, allowing the decomposition of human body parts for more accurate detection and recognition. The analysis of this problem helps to identify the main obstacles in the recognition and analysis of the human body through imagery, where one of the proposed approaches is to utilize Instance Segmentation using YOLO with the aim that each segment of the human body can be identified individually, allowing separation between parts such as the head, body, right arm, and left arm. Consequently, this technique offers valuable applications across various domains, including medical image processing and anthropometric assessments, as it enhances structural detail and contextual understanding of the human body.

3.2 Need analysis

A needs analysis is carried out to ensure that the system can meet user expectations. This analysis includes the functional and non-functional needs that must be fulfilled to ensure the system's development meets its intended objectives.

Functional needs analysis involves an explanation of the procedures that must be carried out by the system to meet the requirements. The functional requirements required in this system include: (i)The system can separate objects of the human body. (ii)Detects and highlights the head, body, right arm, and left arm. (iii) Generate segmentation results that can be used for further calculations.

Table 2: Annotated instances per split. Each image contains one instance of each part, yielding the following instance counts:

Split	Images	Head	Body	Right Hand	Left Hand	Total instances
Train	92	92	92	92	92	**368**
Val	10	10	10	10	10	**40**
Test	5	5	5	5	5	**20**
Total	107	107	107	107	107	**428**

Non-functional needs refer to features, characteristics, or limitations related to the functions or services provided by the system. Below are the essential functional requirements for this system: (i)The interface of this system is designed to be easy to understand so it can be used easily. (ii)To operate the system, the device must be connected to the internet. (iii)The limitations of the parts of the human body that can be detected are the head, body, right arm, and left arm.

3.3 Dataset & annotations

Dataset splits. We curated 107 manually annotated images and split them into 92 / 10 / 5 images for train / validation / test, respectively (Table 2). Each image contains exactly one instance of each annotated part (Head, Body, Right Hand, Left Hand), so the per-split instance counts equal the number of images per part, yielding a total of 428 instances (368/40/20 for train/val/test).

Primary evaluation scope. Because the test split is small (5 images; 20 instances), our primary reported metrics are computed on the aggregated validation+test split (15 images; 60 instances), unless otherwise specified. Test-only results are provided in Supplementary Table 2.

3.4 Research method

The methodology of this research begins by taking human images, which are then used as a dataset. After being collected, the images go through a sorting process (data cleaning) to determine the images that are suitable for use in research. Before the dataset is used for model training, a preparation process is carried out that includes manual labeling using the Roboflow platform. In addition to manual labeling, there is also a preprocessing stage, which includes resizing the image to 640×640 pixels. At this stage, various data augmentation techniques are applied, such as rotation, shear, adjustments in brightness, darkening, and the addition of noise. To ensure effective learning despite dataset limitations, the images are divided into three subsets: 92 images for training, 10 images for validation, and 5 images for testing. The dataset used in this study is limited in both size and diversity, consisting of 107 images of female students aged 19–22, photographed under controlled conditions with uniform poses and clothing. These constraints limit the model's ability to generalize beyond the specific context of this dataset.

Upon completion of the preparation process is completed, the dataset is trained using Google Colab. During training, hyperparameter tuning and model performance evaluation are conducted to determine the optimal configuration. Following the model achieves optimal results, further testing is carried out to ensure the performance of the resulting segmentation. The model that has been obtained is then implemented into a web-based system. The output of this system is in the form of segmented images that can be used for further analysis. The overall research process, covering data acquisition, preprocessing, training, evaluation, and deployment, is illustrated in Figure 2.

3.5 Ethics & consent

Ethics approval. This study involved prospectively collected photographs of adult volunteers (female students aged 19–22) at the Faculty of Public Health, Universitas Sumatera Utara. The protocol was reviewed and approved by the [Name of Ethics Committee/IRB] (Approval No.: [XXXX], Date: [DD Mon YYYY]). All procedures complied with the Declaration of Helsinki and relevant institutional guidelines.

Participant consent. Prior to image capture, participants were informed about the study aims, procedures, risks, and data handling. Written informed consent was obtained for image collection and for the use of de-identified images and derived annotations for research and publication purposes, including illustrative figures in this article. Participants were informed that participation was voluntary and could be withdrawn at any time without penalty.

Privacy and anonymization. To minimize the risk of re-identification, all images used outside the core research team are de-identified. When faces or potentially identifying features are visible, they are obfuscated (e.g., face blurring) before any sharing beyond the research team or inclusion in figures.

Data-sharing policy. Due to privacy and consent restrictions, raw images will not be made publicly available. We will share derived, non-identifiable artifacts—including pixel-wise masks, bounding boxes, class labels, and per-image metadata (age range, pose category)—under a simple Data Use Agreement (DUA) for bona fide research. Requests should be sent to [nainggolan@usu.ac.id]. The complete training/evaluation code and the dataset configuration (YAML) with class definitions and split indices (92/10/5) will be released in a public repository; instructions will allow qualified researchers to reproduce our results using

their own data or upon DUA-approved access to the derived artifacts.

Consent for publication. Participants provided consent for the publication of de-identified sample outputs and figures illustrating the segmentation results.

3.6 Training configuration and hyperparameters

We trained the model using the Ultralytics YOLOv8 instance-segmentation implementation. The main settings were: input size = 640×640 (imgsz=640), batch size = 4 (batch=4), number of epochs = 200 (epochs=200), and early stopping disabled (patience=0). We used the Ultralytics default optimizer, i.e., SGD with momentum 0.937 and weight decay 5×10^{-4} (no manual override). Unless otherwise noted, all other optimization and augmentation knobs followed the Ultralytics defaults (including the library's learning-rate schedule). The dataset was provided via a standard Ultralytics YAML configuration file.

Rationale for the 640×640 input resolution. We adopted 640×640 for three reasons. (i) Architectural efficiency. 640 is a multiple of the model's stride, yielding clean feature-map sizes and fast convolutional blocks. (ii) Compute constraints. With our available GPU memory, 640×640 allowed stable training at batch size 4 without out-of-memory errors; higher resolutions substantially increase memory and latency roughly with pixel count. (iii) Task fit. In our images, the person occupies a relatively large portion of the frame; 640×640 preserves sufficient detail for fine structures (e.g., hands) while maintaining throughput. During preprocessing we use the framework's standard letterbox resizing, which preserves aspect ratio by padding as needed.

3.7 Hardware and software

Hardware. All experiments were run on Google Colab using a single NVIDIA T4 (16 GB VRAM) GPU. The host machine was a Google Compute Engine VM on Linux (Colab environment) with Intel Xeon-family vCPUs and system RAM provisioned by Colab. We archived the GPU and CPU specs (driver, memory, core count) in the run logs to support reproducibility.

Table 3: Summary training (100 epochs, learning rate = 0.01)

Class	Box			Mask		
	Precision	Recall	mAP@0.50	Precision	Recall	mAP@0.50
All	0.913	0.913	0.913	0.913	0.913	0.913
Body	0.996	0.996	0.996	0.996	0.996	0.996
Head	0.972	0.972	0.972	0.972	0.972	0.972
Right Hand	0.913	0.913	0.913	0.913	0.913	0.913
Left Hand	0.996	0.996	0.996	0.996	0.996	0.996

Table 4: Summary training (150 epochs, learning rate = 0.01)

Class	Box			Mask		
	Precision	Recall	mAP@0.50	Precision	Recall	mAP@0.50
All	0.946	0.952	0.977	0.946	0.952	0.977
Body	0.993	1	0.995	0.993	1	0.995
Head	0.992	1	0.995	0.992	1	0.995
Right Hand	0.897	0.9	0.957	0.897	0.9	0.957
Left Hand	0.901	0.907	0.959	0.901	0.907	0.959

Table 5: Summary training (200 epochs, learning rate = 0.01)

Class	Box			Mask		
	Precision	Recall	mAP@0.50	Precision	Recall	mAP@0.50
All	0.914	0.995	0.979	0.914	0.995	0.979
Body	0.92	1	0.995	0.92	1	0.995
Head	0.921	1	0.995	0.921	1	0.995
Right Hand	0.908	0.986	0.977	0.908	0.986	0.977
Left Hand	0.909	0.994	0.95	0.909	0.994	0.95

Table 6: Summary training (250 epochs, learning rate = 0.01)

Class	Box			Mask		
	Precision	Recall	mAP@0.50	Precision	Recall	mAP@0.50
All	0.931	0.974	0.973	0.931	0.974	0.973
Body	0.916	1	0.995	0.916	1	0.995
Head	1	1	0.995	1	1	0.995
Right Hand	0.9	0.9	0.934	0.9	0.9	0.934
Left Hand	0.909	0.994	0.968	0.909	0.994	0.968

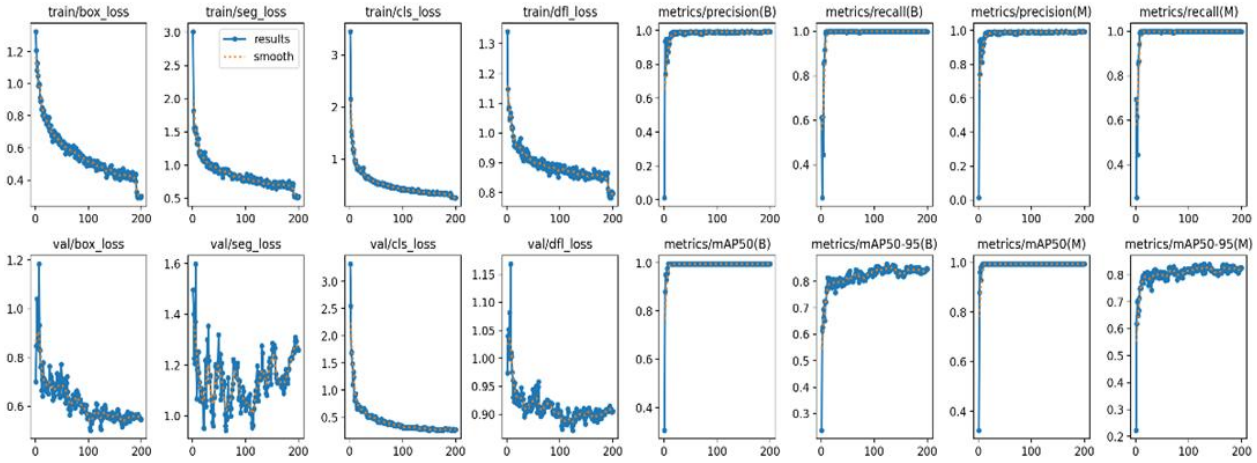


Figure 4: Model performance graph

Table 7: Summary training (200 epochs, learning rate = 0.01) — no augmentation

Class	Box			Mask		
	Precision	Recall	mAP@0.50	Precision	Recall	mAP
All	0.943	0.955	0.974	0.943	0.955	0.974
Body	0.986	1	0.995	0.986	1	0.995
Head	0.99	1	0.995	0.99	1	0.995
Right Hand	0.894	0.9	0.928	0.894	0.9	0.928
Left Hand	0.902	0.919	0.977	0.902	0.919	0.977

Software; The software stack consisted of Python 3.10.12, PyTorch 2.3.0+cu121 with CUDA 12.1, and Ultralytics YOLOv8 8.2.25, running on Linux (Colab).

3.8 Research questions, hypotheses, and intended outcomes

Research Questions (RQs).

- **RQ1**. Can a YOLOv8-based instance segmentation model trained on a small, controlled dataset accurately segment four human body parts (head, body, right arm, left arm)?
- **RQ2**. How consistent is the model's performance across classes and under challenging poses (e.g., partial occlusions or overlap between limbs)?
- **RQ3**. What training configuration (e.g., number of epochs around 100–250 at LR = 0.01) yields the best trade-off among mAP@0.50, precision, recall, and F1 on this dataset?
- **RQ4**. Does a lightweight web application (Flask + HTML/CSS/JS) that wraps the trained model deliver outputs (segmentation masks and confidence scores) suitable for downstream use in constrained scenarios?

Hypotheses.

- **H1 (Accuracy target).** On the held-out test set, the model will achieve mAP@0.50 ≥ 0.95, precision ≥ 0.90, and recall ≥ 0.95 averaged over the four classes (consistent with the performance envelope we aim for on this dataset).
- **H2 (Classwise robustness).** Body and head will reach per-class AP@0.50 ≈ 0.99, while the hands may exhibit slightly lower AP@0.50 due to occlusion/overlap; nevertheless, overall recall will remain ≥ 0.95.
- **H3 (Training efficiency).** Within the explored schedule (100–250 epochs at LR = 0.01), performance will saturate by ~200 epochs, yielding the best balance among mAP@0.50 precision, recall, and F1.
- **H4 (System deliverable).** A browser-based interface that supports upload → segmentation → mask visualization/download will enable practical adoption and reproducible end-to-end evaluation in controlled settings.

Intended outcomes and success criteria.

- **Primary outcome (accuracy).** Achieve mAP@0.50 ≥ 0.95, precision ≥ 0.90, recall ≥ 0.95, and report F1; provide per-class and overall metrics on the held-out test set, with the IoU threshold specification stated explicitly.
- **Secondary outcomes (robustness & analysis).** (a) Document performance under occlusion/overlap cases and quantify any class-specific degradation; (b) summarize the effect of training epochs on mAP@0.50/precision/recall/F1 to justify the chosen configuration.

- **System outcome (deliverable).** Provide a web application (Flask backend; HTML/CSS/JS frontend) that accepts uploads, runs segmentation, displays masks with confidence scores, and supports downloadable outputs for downstream tasks.
- **Application use cases (scope).** Target constrained scenarios such as in-clinic assessment, training/rehabilitation tracking, or structured educational/anthropometric analyses, acknowledging that generalization beyond the current cohort requires broader, more diverse data.

3.9 Data collection

The dataset was collected at the Faculty of Public Health, Universitas Sumatera Utara. To standardize conditions for segmentation, all images were captured with participants standing in an upright posture: the left arm raised to approximately 45°, and the right arm positioned downward alongside the body. Participants wore fitted clothing or—when wearing a hijab—fitted sleeves provided by the study to ensure clear upper-limb contours for segmentation. An example of the dataset used in this study is shown in Figure 3.

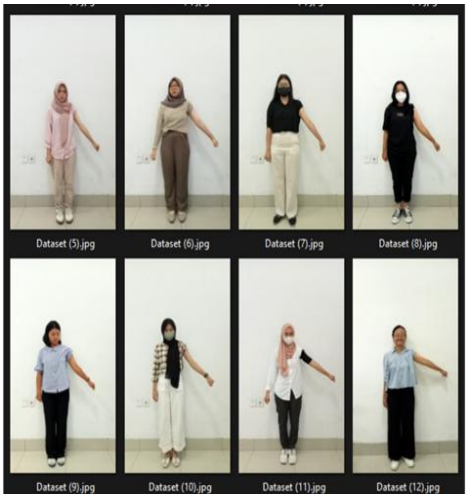


Figure 3: Sample of Images from the dataset

The dataset is relatively homogeneous, comprising predominantly young adult females in a limited set of poses, clothing styles, and viewing angles. While this standardization simplifies annotation and improves labeling consistency, it introduces sampling bias and limits the external validity of our findings. Accordingly, the reported performance should be interpreted as domain-specific rather than population-level generalization. We applied common geometric and photometric augmentations, but these cannot substitute for true diversity across demographics, poses, and environments. See Limitations and Next Steps for planned mitigation via broader data collection and cross-dataset validation.

4 Implementation and result

Hyperparameter tuning was conducted by exploring various combinations of the number of epochs, namely 100, 150, 200, and 250, with a learning rate of 0.01. The detailed results for each configuration are presented in separate tables: results for 100 epochs can be found in Table 3, 150 epochs in Table 4, 200 epochs in Table 5, and 250 epochs in Table 6.

Tuning the hyperparameters results show that the optimal combination was obtained at epoch 200 with a learning rate of 0.01. This combination results in the highest mAP@0.50 value of the other combinations, which is 0.979 for all classes, as shown in Table 5. Therefore, resulting from training with this configuration is chosen as the primary model for the system.

Table 7 summarizes the model's behavior without data augmentation. At the aggregate level, the model maintains strong precision and recall for both boxes and masks; Body and Head perform consistently well, whereas the hands remain comparatively more challenging. Relative to the augmented 200-epoch setting in Table 5, removing augmentation tends to increase precision but reduce recall and leads to a slight decrease in overall mAP. These trends highlight a precision–recall trade-off and support choosing the augmented 200-epoch configuration as the primary model, while documenting the no-augmentation variant for completeness.

Based on the model training process at epoch 200 and a learning rate of 0.01, the evaluation results of the model that was successfully built were presented. The evaluation results graph provides a visual representation of the

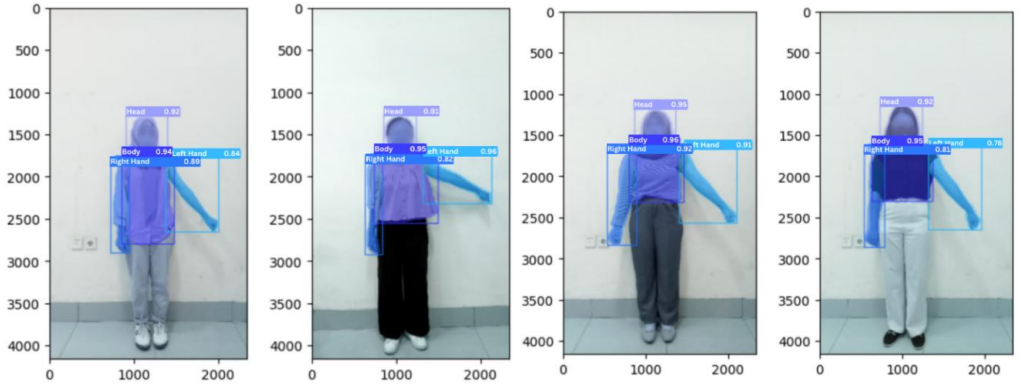


Figure 5: Model testing performance

overall performance of the model, which includes important metrics such as box loss, class loss, and dfl loss, as well as performance evaluation metrics: precision and recall in Fig 4.

The graph shows that the model has a high performance with metrics we use and mAP showing good performance. When we evaluate the results for each class specifically, variations are not much different from other classes. The results show relatively similar to the mAP values. mAP is an important metric that provides an idea of how well the model can recognize objects in all classes.

After obtaining the highest mAP value, further evaluation of the precision and recall of the model training results was carried out. Furthermore, F1 Score is used as an additional metric to evaluate the model's performance more comprehensively. The F1 Score calculated using the formula presented in (1):

$$F1 = 2\,x\,\frac{precision\ x\ recall}{precision+recall} \quad (1)$$

$$\boldsymbol{F1 = 2\,x\,\frac{0.914\,x\,0.995}{0.914 + 0.995}}$$

$$\boldsymbol{F1 = 2\,x\,\frac{0.9094}{1.909}}$$

$$\boldsymbol{F1 = 0.95}$$

Results of model testing show excellent segmentation performance. The model is able to accurately identify and separate objects in the image so that the boundaries of the object can be recognized with more precision. Precision of segmentation capabilities is a crucial aspect of computer image processing, as it ensures that the model can better understand the spatial context of objects in the image. The segmentation results obtained from the model testing are depicted Figure 5.

To complement the aggregate metrics, we report a class-wise confusion matrix computed on the unseen hold-out (validation + test, see Figure 6) because the pure test split is small, ensuring sufficient support. Values are raw instance counts (not normalized), and post-processing uses Ultralytics' default IoU and NMS. The diagonal shows perfect separation for Body and Head (15/15 each). Errors appear only as left–right hand swaps, with Right-Hand → Left-Hand = 2 and Left-Hand → Right-Hand = 2, yielding 13/15 correct for each hand. Overall, the model achieves 56/60 = 93.3% accuracy; per-class precision/recall are 100% for Body and Head, and 86.7% for both hands. These patterns are consistent with boundary ambiguity and partial occlusion around the torso, and align with the qualitative results shown next in Figs. 7–8.

As the next step, the model evaluation is completed on the Google Colab platform; the next step is to develop a web-based application as a user interface to make it easy to use this model. The main application consists of two main parts: the front end and the back end. The backend part is developed using the Flask framework, while the front end uses HTML, CSS, and JavaScript. The app runs on a local server for further testing.

Fig 7(a), 7(b), and 8 is a display of the results of the images that have been segmented. On the left side of the system page, there is an original image uploaded by the user, and on the right, there is an image that has been segmented or a segmented image, along with a button to download the segmented image. In the middle, there is the value of the prediction result. At the bottom, there is a button to return to the main page.

After the development of the application is completed, the system is tested to assess the reliability of the model in accurately recognizing human body parts. The evaluation was performed using 30 images representing distinct sections of the human body. This evaluation aims to verify the extent to which the system can recognize and segment

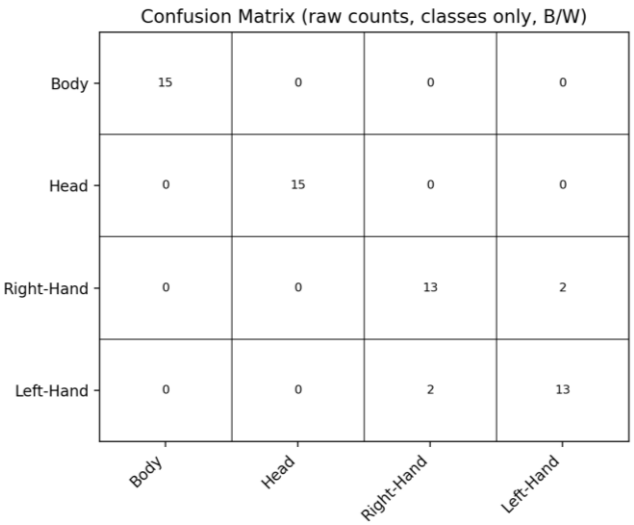
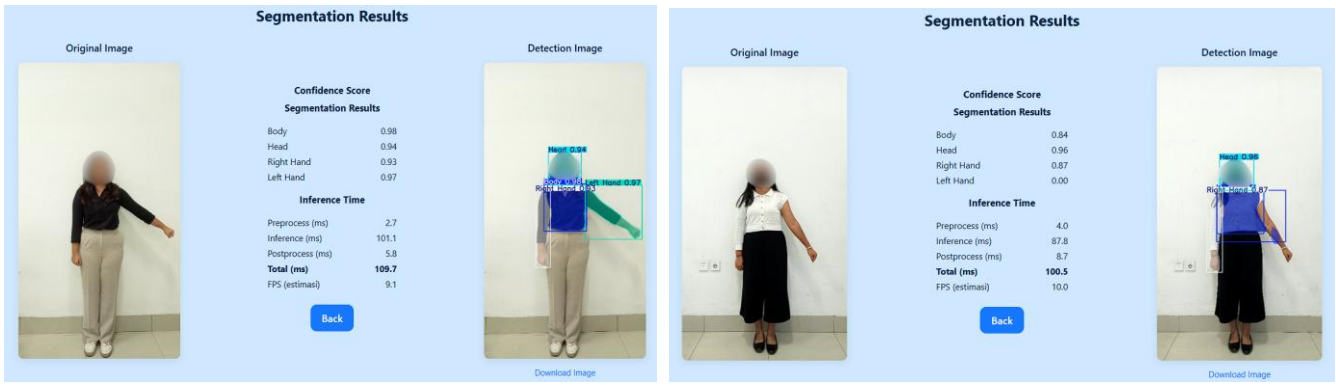


Figure 6: Confusion Matrix (raw counts, classes only) on the unseen hold-out split (validation + test). Post-processing uses Ultralytics default IoU/NMS settings. Rows denote ground-truth classes and columns denote predicted classes; values are raw instance counts (not normalized).

with high accuracy.



(a) (b)

Figure 7: Model performance on web system

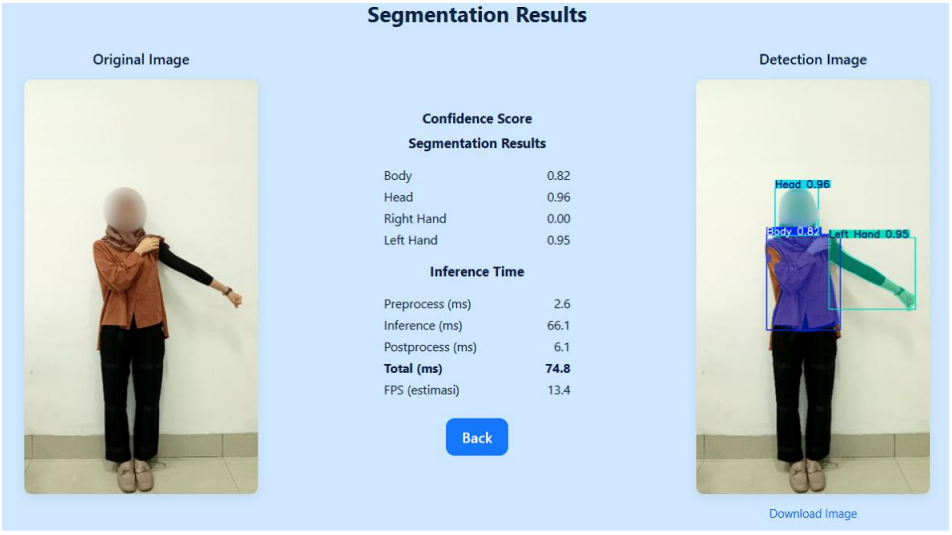


Figure 8: Model performance on web system (some part defined, not detected)

Figure 7(a). The model segments the head and left hand clearly (Head = 0.96, Left Hand = 0.95) with a reasonable body score (0.82). The right hand is missed (0.00), likely due to occlusion and overlap with the torso, so it is absorbed into the body region. Inference time: 74.8 ms total (preprocess 2.6 ms, inference 66.1 ms, postprocess 6.1 ms), ≈ 13.4 FPS. Figure 7(b). The right hand is segmented with good confidence (0.87), while the left hand is not detected (0.00) because it lies flush against the torso, reducing boundary contrast and leading the model to treat it as part of the body. Head and body remain confident (0.96 and 0.84). Inference time: 100.5 ms total (preprocess 4.0 ms, inference 87.8 ms, postprocess 8.7 ms), ≈ 10.0 FPS.

The testing results show that the model works well on most images but still faces challenges under certain conditions, such as objects having unclear boundaries. The detailed results of the segmentation for all tested images are presented in Table 8.

high confidence (Body = 0.98, Head = 0.94, Right Hand = 0.93, Left Hand = 0.97), indicating good separation of the hands from the torso across the scene. Inference time: 109.7 ms total (preprocess 2.7 ms, inference 101.1 ms, postprocess 5.8 ms), ≈ 9.1 FPS.

Note on inference time. The latency is dominated by the inference step (≈ 88–92% of total). Preprocess covers resizing/normalization, and postprocess includes decoding, thresholding, and mask/box handling. Across the three samples the system runs at ~9–13 FPS, which is near real-time for single-image processing on the tested setup.

Table 8: System test results

No	Data Test	Actual Classes	Test Results	Confidence Score
1	Data1	Body	Body	0.98
		Head	Head	0.94
		Right Hand	Right Hand	0.93
		Left Hand	Left Hand	0.97
2	Data2	Body	Body	0.99
		Head	Head	0.93
		Right Hand	Right Hand	0.92

		Left Hand	Left Hand	0.95
3	Data3	Body	Body	0.89
		Head	Head	0.95
		Right Hand	Right Hand	0.93
		Left Hand	Left Hand	0.98
4	Data4	Body	Body	0.85
		Head	Head	0.95
		Right Hand	Right Hand	0.93
		Left Hand	Left Hand	0.99
5	Data5	Body	Body	0.98
		Head	Head	0.93
		Right Hand	Right Hand	0.9
		Left Hand	Left Hand	0.98
6	Data6	Body	Body	0.96
		Head	Head	0.93
		Right Hand	Right Hand	0.97
		Left Hand	Left Hand	0.99
7	Data7	Body	Body	0.9
		Head	Head	0.94
		Right Hand	Right Hand	0.87
		Left Hand	Left Hand	0.93
8	Data8	Body	Body	0.98
		Head	Head	0.97
		Right Hand	Right Hand	0.92
		Left Hand	Left Hand	0.96
9	Data9	Body	Body	0.98
		Head	Head	0.97
		Right Hand	Right Hand	0.99
		Left Hand	Left Hand	0.89
10	Data10	Body	Body	0.84
		Head	Head	0.95
		Right Hand	Right Hand	0.83
		Left Hand	Left Hand	0.86
11	Data11	Body	Body	0.84
		Head	Head	0.96
		Right Hand	Right Hand	0.87
		Left Hand	Undetected	0
12	Data12	Body	Body	0.97
		Head	Head	0.94
		Right Hand	Right Hand	0.96
		Left Hand	Left Hand	0.93
13	Data13	Body	Body	0.95
		Head	Head	0.95
		Right Hand	Right Hand	0.92
		Left Hand	Left Hand	0.9
14	Data14	Body	Body	0.99
		Head	Head	0.94
		Right Hand	Right Hand	0.86
		Left Hand	Left Hand	0.92
15	Data15	Body	Body	0.88
		Head	Head	0.95
		Right Hand	Right Hand	0.95
		Left Hand	Left Hand	0.97
16	Data16	Body	Body	0.93
		Head	Head	0.97
		Right Hand	Right Hand	0.69
		Left Hand	Left Hand	0.91
17	Data17	Body	Body	0.87
		Head	Head	0.93
		Right Hand	Right Hand	0.9
		Left Hand	Left Hand	0.89
18	Data18	Body	Body	0.97
		Head	Head	0.98
		Right Hand	Right Hand	0.96
		Left Hand	Left Hand	0.88
19	Data19	Body	Body	0.97
		Head	Head	0.93
		Right Hand	Right Hand	0.91
		Left Hand	Left Hand	0.59
20	Data20	Body	Body	0.98
		Head	Head	0.92
		Right Hand	Right Hand	0.92
		Left Hand	Left Hand	0.94
21	Data21	Body	Body	0.85
		Head	Head	0.96
		Right Hand	Right Hand	0.9
		Left Hand	Left Hand	0.84
22	Data22	Body	Body	0.82
		Head	Head	0.96
		Right Hand	Undetected	0
		Left Hand	Left Hand	0.95
23	Data23	Body	Body	0.98
		Head	Head	0.98
		Right Hand	Right Hand	0.92
		Left Hand	Left Hand	0.77
24	Data24	Body	Body	0.99
		Head	Head	0.96
		Right Hand	Right Hand	0.9
		Left Hand	Left Hand	0.87
25	Data25	Body	Body	0.87
		Head	Head	0.95
		Right Hand	Right Hand	0.89
		Left Hand	Left Hand	0.96
26	Data26	Body	Body	0.81
		Head	Head	0.96
		Right Hand	Right Hand	0.5
		Left Hand	Left Hand	0.99
27	Data27	Body	Body	0.96
		Head	Head	0.92
		Right Hand	Right Hand	0.93
		Left Hand	Left Hand	0.97
28	Data28	Body	Body	0.93
		Head	Head	0.91
		Right Hand	Right Hand	0.89
		Left Hand	Left Hand	0.96
29	Data29	Body	Body	0.96
		Head	Head	0.95
		Right Hand	Right Hand	0.87
		Left Hand	Left Hand	0.93
30	Data30	Body	Body	0.94
		Head	Head	0.96
		Right Hand	Right Hand	0.95
		Left Hand	Left Hand	0.96

Referring to the test outcomes above, it is evident that the model performs exceptionally well in identifying objects within images. The result is supported by high confidence scores. Confidence score is a measure that indicates the level of confidence or certainty in the results provided by a prediction system or model. This score is usually expressed in the form of numbers between 0 and 1 or in the form of percentages between 0% and 100%. The higher the confidence score, the more confident the system or model is in the results or predictions made.

In the test results with this test data, we can also see some shortcomings or challenges faced by the model. One of the main challenges is in recognizing body parts that overlap with other body parts. This is clearly seen in data numbers 11 and 22. When the body parts overlap each other, the model has difficulty recognizing and identifying each part accurately.

This difficulty may be due to the increased visual complexity when objects overlap each other, resulting in important features becoming less clear or distorted. Overall confidence score is good, but the result for the overlapping objects may be lower or even unpredictable, suggesting that the model is less confident in its predictions.

5 Discussion

This section is organized into four parts: accuracy–efficiency context (5.1), error analysis under occlusion (5.2), domain scope and application (5.3), and limitations with next steps (5.4).

5.1 Accuracy–efficiency context

Our system achieves strong within-domain performance (mAP@0.50 = 0.979, precision = 0.914, recall = 0.995 at 200 epochs) on a four-class human-part segmentation task, substantially higher than the AP reported by recent real-time or proposal-based instance segmentation methods on broad benchmarks. For example, Sem2Ins paired with fast semantic backbones yields 14.5–16.9 AP at ~20–20.8 FPS on Cityscapes, while proposal-based Mask R-CNN–style pipelines are typically <2 FPS at HD resolution [8]; Sem2Ins can reach ~54.7 FPS with ~19.1 AP under different settings. These contrasts are informative but not strictly comparable because (i) metrics are reported on different datasets and operating points and (ii) our task involves only four anatomically constrained classes in controlled scenes, whereas Cityscapes/COCO involve many object categories and diverse imagery. Still, they contextualize our accuracy–efficiency profile relative to state-of-the-art designs that trade speed for AP or vice versa.

Comparisons with instance segmentation on synthetic-to-real aerial imagery similarly illustrate task difficulty effects. A Mask R-CNN model refined on a simulator-derived dataset reports ~5 FPS on a single GPU and COCO-style AP values around 36.25 (ResNet-50-FPN) [9], which are far lower than our mAP—again reflecting the broader class set, cluttered backgrounds, and viewpoint variability in aerial scenes versus our narrow, structured domain. The one-shot human parsing literature (e.g., EOP-Net) focuses on a different objective: open-set, support-guided parsing evaluated primarily with mean IoU rather than AP/mAP, making direct numerical comparison inappropriate [10]. OSHP methods parse a query image into classes defined by a single support example and report k-way/1-way mIoU across base and novel classes; this episodic formulation targets generalization to novel labels rather than instance-level detection. Our result profile (very high recall on a fixed label set) complements rather than competes with that goal.

5.2 Error analysis under occlusion

The confidence scores reported in Table 8 reveal a consistently high certainty for most predictions (>0.9), but lower scores and misdetections occur primarily in scenarios with overlapping body parts or minimal visible separation between segments. For example, in test cases 11 and 22, the model failed to detect the left or right hand entirely when it overlapped closely with the torso, producing confidence scores as low as 0.0 for the missed parts. This suggests that the YOLOv8 segmentation head struggles when the spatial boundaries are ambiguous or when limb positions cause occlusion. The visual complexity in such conditions leads to degraded feature clarity, making it harder for the model to differentiate class-specific contours. These findings are consistent with prior observations in instance segmentation literature [9], where occlusion and inter-class overlap significantly reduce per-instance AP.

5.3 Domain scope and application

We attribute our higher mAP@0.50 and recall primarily to dataset scope and homogeneity: 107 images of female students (ages 19–22) captured under standardized poses and clothing in a controlled environment. This reduces appearance variance and occlusions, simplifies class boundaries, and limits cross-domain generalization demands, all of which are known to inflate in-domain accuracy relative to open-world settings. These choices align with our intended application—fine-grained, body-part analysis in constrained scenarios—and are embodied in a lightweight web implementation that accepts uploads, runs YOLOv8-based instance segmentation, and returns labeled masks for downstream use. Nevertheless, these findings should be interpreted within our dataset constraints; we next discuss limitations and directions for future work.

5.4 Limitations and next steps

The primary limitation of this work is dataset homogeneity: most images depict young adult females in a constrained set of poses and capture conditions. As a result, generalization to other demographics (e.g., different age groups and genders), body types, clothing styles, viewpoints, occlusions, and environments remains uncertain. To address this, we plan the following steps: (i) Data diversification — collect a larger and more diverse corpus that balances demographics, poses, camera viewpoints, backgrounds, and lighting; (ii) Cross-dataset validation — train on our dataset and evaluate on independent, external datasets to estimate out-of-distribution performance; (iii) Stratified evaluation — report subgroup-specific metrics (e.g., by demographic and pose) and conduct leave-one-group-out validation to quantify potential biases; and (iv) Transfer and adaptation — when deploying to new domains, apply fine-tuning or domain adaptation rather than assuming zero-shot transfer.

6 Conclusion

This study empirically demonstrates that YOLOv8 instance segmentation can accurately segment human body parts in a constrained setting, achieving an mAP@0.50 of 0.979 after 200 training epochs (precision 0.914, recall 0.995, F1-score 0.95). The novelty of our work is threefold: (i) curating and annotating a controlled-pose dataset of young adult females with pixel-level masks for four parts (head, torso, right hand, left hand); (ii) documenting a reproducible YOLOv8 training recipe with an epoch ablation that identifies 200 epochs as the best trade-off; and (iii) delivering a lightweight web-based inference interface that accepts image uploads and returns per-part masks with confidence scores. While

performance is strong under homogeneous conditions, generalizability remains untested due to the dataset's narrow demographics and controlled clothing/poses, and the model still struggles with overlapping parts/occlusion. Future work will broaden data diversity (subjects, clothing, poses, viewpoints, backgrounds), incorporate explicit occlusion handling and more complex spatial reasoning, and benchmark against alternative instance-segmentation methods to improve robustness for real-world applications such as body-part recognition, rehabilitation monitoring, and clinical pre-processing.

References

[1] S. Secinaro, D. Calandra, A. Secinaro, V. Muthurangu, and P. Biancone, "The Role of Artificial Intelligence in Healthcare: A Structured Literature Review," BMC Med. Inform. Decis. Mak., vol. 21, no. 1, pp. 1–23, 2021, doi: 10.1186/s12911-021-01488-9.

[2] N. H. Oktaviani, M. N. Widyawati, and Kurnianingsih, "Development of a detection tool in pregnant women and its recommendations in utilizing artificial intelligence," J. Matern. Child Heal., vol. 09, pp. 410–420, 2024. DOI: 10.26911/thejmch.2024.09.03.11

[3] L. Rundo, R. Pirrone, S. Vitabile, E. Sala, and O. Gambino, "Recent advances of HCI in decision-making tasks for optimized clinical workflows and precision medicine," J. Biomed. Inform., vol. 108, pp. 1–13, 2020, doi: 10.1016/j.jbi.2020.103479.

[4] D. S. Bukit et al., "Leveraging Machine Learning Techniques for Stunting Detection and Height Growth Prediction in Children Aged 0-5 Years," Proc. - ELTICOM 2024 8th Int. Conf. Electr. Telecommun. Comput. Eng. Tech-Driven Innov. Glob. Organ. Resil., pp. 130–134, 2024, doi: 10.1109/ELTICOM64085.2024.10864967.

[5] M. Javaid, A. Haleem, R. P. Singh, and M. Ahmed, "Computer vision to enhance healthcare domain: An overview of features, implementation, and opportunities," Intell. Pharm., vol. 2, no. 6, pp. 792–803, 2024, doi: 10.1016/j.ipha.2024.05.007.

[6] K. K. D. Ramesh, G. Kiran Kumar, K. Swapna, D. Datta, and S. Suman Rajest, "A review of medical image segmentation algorithms," EAI Endorsed Trans. Pervasive Heal. Technol., vol. 7, no. 27, pp. 1–9, 2021, doi: 10.4108/eai.12-4-2021.169184.

[7] A. Nadeem, A. Jalal, and K. Kim, "Automatic Human Posture Estimation for Sport Activity Recognition with Robust Body Parts Detection and Entropy Markov Model," Springer, pp. 21465–21498, 2021, doi:10.1007/s11042-021-10687-5.

[8] C. Yin, S. Member, J. Tang, and T. Yuan, "Bridging the Gap Between Semantic Segmentation and Instance Segmentation," IEEE Trans. Multimed., vol. 24, pp. 4183–4196, 2022, doi: 10.1109/TMM.2021.3114541.

[9] F. X. Viana, G. M. Araujo, M. F. Pinto, J. Colares, and D. B. Haddad, "Aerial image instance segmentation through synthetic data using deep learning," J. Brazilian Soc. Comput. Intell., vol. 18, no. 1, pp. 35–46, 2020, doi: 10.21528/lnlm-vol18-no1-art3.

[10] H. He, J. Zhang, B. Zhuang, J. Cai, and D. Tao, "End-to-end one-shot human parsing," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 12, pp. 14481–14496, 2023, doi: 10.1109/TPAMI.2023.3301672.

[11] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," Computers, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[12] M. S, M. Saranya S, K. T, and S. P, "Image detection and segmentation using YOLO v5 for surveillance," in International Conference on Software Engineering and Machine Learning, 2023, pp. 142–147. doi: 10.54254/2755-2721/8/20230109.

[13] N. P. Motwani and S. S, "Human activities detection using deep learning technique- YOLOv8," in ITM Web of Conferences, 2023, pp. 1–8. doi: 10.1051/itmconf/20235603003.

[14] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, "Look into Person: Self-supervised Structure-sensitive Learning and a new benchmark for human parsing," Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 6757–6765, 2017, doi: 10.1109/CVPR.2017.715.

[15] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," Sensors, vol. 23, no. 16, pp. 1–27, 2023, doi: 10.3390/s23167190.

[16] Y. Yang, Z. Song, T. D. Palaoag, and S. Li, "An Improved Model for People Detection Based on YOLOv8," Proceeding 2024 9th Int. Conf. Inf. Technol. Digit. Appl. ICITDA 2024, pp. 1–7, 2024, doi: 10.1109/ICITDA64560.2024.10809805.

[17] A. Dr. Jaka Sunardi, M.Kes., A. dr. Prijo Sudibjo, M.Kes., Sp.S., and M. S. Dr. Endang Rini Sukamti, DIKTAT Anatomi Manusia, 1st ed., vol. 11, no. 1. Yogyakarta: UNY Press, 2020. [Online]. Available: https://books.google.co.id/books?id=8AcREAAAQBAJ

[18] Y. Yu et al., "Techniques and Challenges of Image Segmentation: A Review," Electron., vol. 12, no. 5, 2023, doi: 10.3390/electronics12051199.

[19] W. Cai, Z. Xiong, X. Sun, P. L. Rosin, L. Jin, and X. Peng, "Panoptic segmentation-based attention for

image captioning," Appl. Sci., vol. 10, no. 1, pp. 1–18, 2020, doi: 10.3390/app10010391.

[20] Y. Chuang, S. Zhang, and X. Zhao, "Deep learning-based panoptic segmentation: Recent advances and perspectives," IET Image Process., vol. 17, no. 10, pp. 2807–2828, 2023, doi: 10.1049/ipr2.12853.

[21] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," Machines, vol. 11, no. 7, 2023, doi: 10.3390/machines11070677.

[22] L. Yang, Q. Song, Z. Wang, and M. Jiang, "Parsing R-CNN for instance-level human analysis," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2019-June, no. Figure 1, pp. 364–373, 2019, doi: 10.1109/CVPR.2019.00045.

[23] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, and Y. Zhao, "Devil in the details: Towards accurate single and multiple human parsing," 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, pp. 4814–4821, 2019, doi: 10.1609/aaai.v33i01.33014814.

[24] P. Li, Y. Xu, Y. Wei, and Y. Yang, "Self-Correction for Human Parsing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 6, pp. 3260–3271, 2022, doi: 10.1109/TPAMI.2020.3048039.

[25] X. Zhang, Y. Chen, B. Zhu, J. Wang, and M. Tang, "Part-aware context network for human parsing," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 8968–8977, 2020, doi: 10.1109/CVPR42600.2020.00899.

[26] X. Zhang, Y. Wang, and P. Xiong, "Affinity-aware Compression and Expansion Network for Human Parsing," 2020, [Online]. Available: http://arxiv.org/abs/2008.10191

[27] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, "Instance-Level Human Parsing via Part Grouping Network," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11208 LNCS, pp. 805–822, 2018, doi: 10.1007/978-3-030-01225-0_47.

[28] Maya Silvi Lydia, Pauzi Ibrahim Nainggolan, Desilia Selvida, Doli Aulia Hamdalah, Dhani Syahputra Bukit, Amalia and Rahmita Wirza Binti O. K. Rahmat, "Mid-Upper Arm Circumference Measurement Using Digital Images: A Top-Down Approach with Panoptic Segmentation Using Mask R-CNN" International Journal of Advanced Computer Science and Applications (IJACSA), 16(8), 2025. doi: 10.14569/IJACSA.2025.0160839

[29] C. Pan, H. Zhao, and M. Sun, "Real-time Target Detection System in Scenic Landscape Based on Improved YOLOv4 Algorithm," Informatica (Slovenia), vol. 48, no. 8, pp. 35–48, 2024, doi: 10.31449/inf.v48i8.5700.

[30] Q. Zhang, J. Zhang, and S. Yang, "Enhancing YOLOv8 Object Detection with Shape-IoU Loss and Local Convolution for Small Target Recognition," Informatica (Slovenia), vol. 49, no. 21, pp. 105–120, 2025, doi: 10.31449/inf.v49i21.8287.

[31] M. Xu, "YOLO-Based Framework with Temporal Context and Network Analysis for Real-Time Basketball Video Understanding," Informatica, vol. 49, no. 27, pp. 173–184, 2025, doi: 10.31449/inf.v49i27.8657.