Video Analysis of Vamana Procedures in Panchkarma Using YOLO Segmentation for Ayurvedic Research

---

Supervised by:
Dr. Sumit Kalra
Assistant Professor
Dept. of CSE, IIT Jodhpur

Co-Supervised by:
Dr. Pooja Pathak
All India Institute of Ayurveda

Submitted on: May 03, 2025

Submitted by:

Ishan Shah (B23CM1050)
Vyom Shah (B23CM1058)

Indian Institute of Technology Jodhpur

Abstract

This project investigates a novel approach for video analysis in Ayurveda, focusing on the Vamana procedure in Panchkarma, which involves therapeutic vomiting after consuming large quantities of ghee. Working with videos provided by the All India Institute of Ayurveda (AIIA), we developed a model to automatically identify and analyze specific events and things in this procedure.

Our approach leverages **YOLO** (You Only Look Once) segmentation with a custom-built dataset, marking a pioneering effort in using Computer Vision for Ayurvedic research. The model was trained on annotated data using Google Colab, with an emphasis on detecting vomital extracts, estimating their color, and quantifying volume based on pixel area.

This report covers the end-to-end pipeline including dataset annotation, model architecture design, training methodology, and experimental results. Our findings demonstrate the feasibility of applying modern video analysis techniques to traditional Ayurvedic practices, opening new avenues for automated medical research.

Keywords: Ayurveda, Panchkarma, Vamana, Computer Vision, YOLO, Video Analysis

Contents

1. Introduction

1.1 Background

In Ayurveda, Panchkarma is a centuries-old therapeutic practice aimed at purifying the body through five major procedures, one of which is Vamana. During Vamana, patients are administered medicated ghee and herbal formulations, inducing controlled purgation to eliminate accumulated toxins. Despite its widespread usage, the evaluation of its efficacy remains largely subjective. This project proposes a computer vision-based solution for real-time quantitative and qualitative assessment of Vamana processes.

1.2 Objectives

The primary objectives are:

- To develop a model for real-time video analysis capable of segmenting and identifying key Vamana events.
- To estimate the volume of vomital extracts through segmentation and color-based surface area analysis.
- To construct a custom annotated dataset for the specific domain of Vamana procedures.

1.3 Contributions

- **Custom Dataset:** Developed using Roboflow with manual intervention to ensure quality.
- **YOLO-S Application:** Implemented the lightweight YOLO v8-S segmentation model, optimized for real-time use.
- **Volume Estimation:** Devised an area-based pixel regression system combined with color filtering for vomital volume estimation.

2. Methodology

2.1 Dataset Creation and Annotation

As there were no existing datasets suitable for Vamana analysis, a custom dataset was developed:

- Frames were extracted from recorded videos.
- Roboflow was utilized for annotation.
- Manual segmentation marked vomital regions frame-by-frame to generate training masks.

2.2 Model Selection: YOLO v8-Segmentation

The chosen model is YOLOv8-Segmentation (YOLO-S) by Ultralytics, suitable for real-time segmentation tasks. Key highlights:

- **Model Type:** YOLOv8-Segmentation
- **Parameters:** 2.4 million
- **Input Image Size:** 640x640
- **Training Epochs:** 60
- **Confidence Threshold:** 0.25
- **Platform:** Google Colab with T4 GPU

2.3 Algorithm for Volume Estimation and Color Detection

Color Detection: RGB pixel values within segmented areas were matched to CSS3 color labels (e.g., Light Yellow, Dark Green). This made the visualization and classification of vomital content more interpretable.
Volume Estimation: A simple linear regression model was trained on manually labeled data to map segmented pixel area to actual volume in milliliters. The area was calculated using the binary segmentation mask.
Thresholding: A minimum area threshold was set to ignore small, noisy detections.

3. Implementation Steps and Code Workflow

3.1 Dependencies Installation

Before running the model, ensure the following libraries are installed:

```
!pip install ultralytics
!pip install opencv-python
!pip install matplotlib
!pip install pillow==9.4.0
```

Other optional tools: Roboflow SDK for dataset integration, NumPy, Pandas for data processing.

3.2 Model Training Code (Google Colab)

Key code components:

```
from ultralytics import YOLO
model = YOLO("yolov8s-seg.pt")
model.train(data="path/to/dataset.yaml", epochs=60, imgsz=640)
```

Inference:

```
results = model.predict(source="video.mp4", conf=0.25, save=True)
```

Color Analysis:

```
from PIL import Image
image = Image.open("segmented_output.jpg")
pixels = list(image.getdata())
Perform RGB-based color classification
```

Volume Estimation:

```
seg_area = np.count_nonzero(mask)
predicted_volume = regression_model.predict([[seg_area]])
```

4. Accuracy Metrics and Analysis

The model performance was assessed using both qualitative observation and a limited quantitative analysis due to the small size of the dataset.

- **Intersection over Union (IoU):** The model was trained on 24 manually annotated images focused primarily on the sink region, where vomital fluid was expected to collect. The segmentation mask for the sink area achieved a visually satisfactory overlap with the ground truth in most frames, although a precise IoU value was not calculated due to the absence of class-specific evaluation.
- **Class Limitations:** Only the 'sink' area (channel flow region) was included in the training data. Class 0 (the patient's current vomit flow) was intentionally omitted from the YOLOv8 segmentation model. Therefore, the model does not detect or analyze patient's current vomit beam, but solely focuses on vomital fluid detected in the sink region.
- **Conditional Detection:** Importantly, the model detects the sink area only if the patient is in a head-down position, indicating that fluid presence and flow typically coincide with that posture.
- **Visual Evaluation:** Frame-by-frame visual inspection showed that the model reliably detected the presence of fluid within the sink when lighting and camera angles were favorable. Detection quality deteriorated in cases of poor lighting, occlusion, or obstructions in the video frame.
- **Precision and Recall:** Given the small dataset and absence of a standard validation split, formal metrics like precision, recall, and mAP could not be computed. Hence, only qualitative validation was used in this study.

4.1 Validation with Manual Records

We validated our estimations using manually recorded data from Ayurvedic practitioners during the Vamana process. The fluid color and area detection correlated closely with ground truth observations. However, challenges in precise volume estimation (as detailed in Section 5.1) led to a gap in predicted volume compared to actual measurements.

4.2 Limitations and Justification

- **Color Detection Robustness:** The RGB-based color classification using CSS3 color names showed consistent results. Minor hue variations did not drastically affect outcome classification.
- **Volume Estimation Uncertainty:** The linear regression approach used for area-to-volume conversion was constrained by the lack of real depth information and a standardized camera angle. Hence, the model could capture relative trends (e.g., increasing volume) but not absolute accuracy.
- **Frame Quality Impact:** Blurred or low-resolution frames, especially those affected by motion, degraded segmentation performance. Frame interpolation and denoising could improve this in future implementations.

5. Discussion and Future Work

5.1 Challenges

- **Device Heterogeneity:** The source videos were captured using different mobile devices and cameras, leading to variations in frame resolution, aspect ratios, compression artifacts, and frame rates. This non-uniformity introduced inconsistencies in model performance, particularly during segmentation.
- **False Positives in Head-Down Detection:** One of the most significant challenges was misclassification of a patient's posture. Due to camera angles, lighting, or people occluding the subject in the frame, the model occasionally misinterpreted other postures as the "head-down" position—triggering premature or incorrect volume analysis.
- **Frame-by-Frame Computation Overhead:** Real-time analysis requires processing every frame (typically 25–30 fps), leading to high computational demand and longer inference times, especially on resource-constrained systems. Optimizing the runtime while maintaining accuracy is non-trivial.
- **Manual Annotation Bottleneck:** Since there is no public dataset for Vamana procedures, manual frame-by-frame annotation was required. This was labor-intensive and required domain expertise to accurately identify vomital regions and postures.
- **Variability in Vomital Appearance:** Vomital substances had different viscosities, transparencies, and colors based on the individual's physiology, making segmentation and pixel-based color analysis more complex. These inconsistencies could result in missed or inaccurate detections.

5.2 Future Directions

- **Temporal Sequence Modeling:** Future versions of this model can leverage time-series techniques (like LSTM or 3D CNNs) to analyze the vomiting event over multiple frames. This could help better identify start-end durations of vomiting and reduce false triggers from single-frame misclassifications.
- **Depth Estimation for Volume Accuracy:** Integrating monocular depth estimation models or stereo vision setups can significantly improve vomital volume estimation by factoring in real-world depth, which is not captured in simple pixel area regression.

- **Multi-Class Labeling:** Introducing class labels for different types of vomital substances (e.g., bile, mucous, ghee, etc.) may allow a more nuanced clinical understanding and a richer training set for multi-class segmentation.
- **Hardware Integration:** Future systems could integrate with hospital record systems or wearable devices to automatically log vomiting events along with patient metadata like time, pulse, and post-vomiting status.
- **Domain Transfer Learning:** With limited Vamana-specific data, domain adaptation from other medical segmentation datasets could be investigated to enhance generalization in unseen scenarios.
- **Cloud-based Video Analysis Dashboard:** A web-based portal where Ayurvedic clinicians can upload videos and receive automatic analysis reports would scale the model's usability beyond research prototypes.

6. Conclusion

This research successfully bridges ancient Ayurvedic practice with modern deep learning tools. The segmentation and analysis model built using YOLOv8-S demonstrates promising potential in real-time clinical video assessment, offering new avenues for evidence-based practices in Ayurveda. We have built an AI-based system that performs the following tasks:

1. Detect when the patient's face turns downward.
2. From that point, analyze video frames for expelled fluid.
3. Segment the fluid using a YOLOv8-Segmentation model.
4. Measure the area (in pixels) and color of the detected fluid.
5. Record timestamped fluid data in an Excel file and map the detected color with items like phanta, milk, etc.
6. Estimate the volume of expelled fluid (in ml) using linear regression trained on manual measurements.

Appendices

Appendix A: Dataset Samples

Appendix B: System Specifications

- Python Version: 3.9
- Pillow Version: 9.4.0
- YOLOv8 Version: YOLOv8-Segmentation
- GPU: Google Colab T4

Addon Content and Code

Sample YOLO-S Training Command

```
yolo task=segment mode=train model=yolov8s-seg.pt
data=data.yaml epochs=60 imgsz=640 conf=0.25
```

Volume Estimation Algorithm (Simplified)

```
def estimate_volume(segmented_area_pixels, pixel_to_ml_ratio=0.05):
    return segmented_area_pixels * pixel_to_ml_ratio
```

Color Detection and Mapping (Sample)

```
from webcolors import rgb_to_name

def get_color_name(rgb_value):
    try:
        return rgb_to_name(rgb_value)
    except ValueError:
        return "Unknown"
```