Open Suturing Skills Challenge Submission

David Teixeira[1 [PG55929]], Eduardo Cunha[1 [PG55939]], Jorge Rodrigues[1 [PG55966]], and Tiago Rodrigues[1 [PG56013]]

Universidade do Minho
{pg55929,pg55939,pg55966,pg56013}@uminho.pt

Abstract. This paper presents a deep learning-based system for automated assessment of open suturing skills using video analysis. Leveraging CLIP, R3D, and YOLO architectures, the model integrates 2D, 3D, and object-level features through a temporal and cross-attention transformer to classify surgical performance. We address three tasks: overall skill classification, OSATS dimension scoring, and object tracking. Despite technical innovation, model performance in Task 1 plateaued due to class imbalance and domain mismatch in pretrained encoders. Task 2 was conceptualized as a variation of the Task 1 architecture, but without proper assessment. Task 3 achieved high accuracy in object and hand detection. The results highlight the need for domain-specific training and balanced datasets for surgical skill evaluation.

Keywords: Surgical Skill Assessment · Video Analysis · Multi-Modal Fusion · Deep Learning

1 Introduction

The advancement of artificial intelligence (AI) has increasingly permeated skill-based education, offering new pathways for personalized feedback and performance assessment. In domains requiring nuanced motor skills, such as surgical suturing, traditional training methods often fall short due to constraints in time, expert availability, and subjectivity in evaluation. The integration of automatic assessment and feedback systems represents a promising shift towards scalable, objective, and reflective skill development.

In physical education, as Hsia et al. (2024)[1] demonstrate, AI-based systems employing pose estimation technologies have successfully provided learners with immediate, personalized feedback, enhancing not only skill accuracy but also learner motivation and reflective practice. Their study on a yoga assessment model using OpenPose and expert rule-based feedback revealed that automated systems could match expert evaluations while significantly improving student outcomes.

This paper presents a deep learning-based suturing skill assessment model, designed to evaluate performance through video analysis and student classification. Our model aims to deliberate precise classification, leveraging state-of-the-art vision models.

2 Methodology

To develop a robust model, we adopted a five-stage AI development pipeline, commonly known as the SEMMA. This methodology guided our data handling, model construction, and evaluation strategy to ensure both technical rigor and educational relevance in skill classification.

We began by selecting a representative subset of the available suturing performance data, later scaled, considering the available data (OSS 2024) and disk space. During exploration, we performed an analysis of the dataset to identify relevant information relative to the data. Preprocessing and augmentation were considered to normalize and enrich the dataset. For classification, we developed and fine-tuned a deep learning pipeline based on state-of-the-art vision models. Model performance was evaluated using MICCAI metrics when possible.

3 Data and Preparations

The dataset consisted of suturing performance videos captured under controlled conditions, ensuring consistency in studio setup, camera angle, and lighting across all recordings. This uniformity minimized variability in the visual data, allowing us to focus on skill-based differences rather than environmental factors.

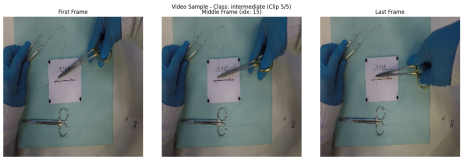

Fig. 1. Sample taken from a video

The dataset exhibited an uneven distribution of samples across skill levels or categories, which required careful consideration during model training and evaluation. Also, not all videos spanned the full 5 minute duration, which might need adjustments in preprocessing to handle varying sequence lengths.

Given the consistency in recording conditions, minimal preprocessing was applied. The primary step involved resizing videos where necessary to ensure uniform input dimensions for the model. No additional processing was performed, as the raw data quality was deemed sufficient for analysis.

While data augmentation techniques were considered to enhance dataset diversity, their application was limited due to the controlled nature of the original recordings. The decision prioritized preserving the authenticity of the skill demonstrations over artificially expanding the dataset.

This streamlined approach to data preparation ensured that the model's performance reflected true skill assessment, rather than artifacts of preprocessing or augmentation.

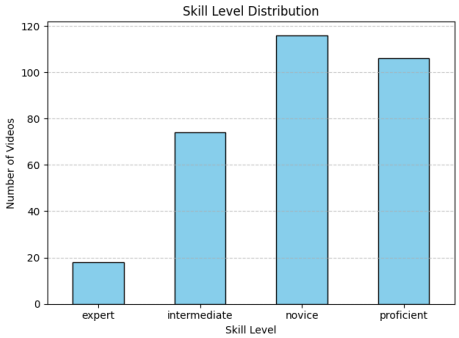

Fig. 2. Class Distribution

4 Model Development

The conceived architecture, as illustrated in figure 3, is based on N. A. Tu et al.'s (2025) [2] work on aerial classification, which has similar aspects to the video material we want to classify. Given a video X and its label Y, we store the label for training purposes and only feed the video to the model. The model has, passed on the conception, the prompts representing each label, which are encoded through a text encoder. As for the video, we have 3 separate visual models (2D, 3D, and object detection) encoding the input on 3 separate vectors. Respectively, we obtain the text features $\{S_k\}$, from the text encoder, we obtain 2D features $\{U_t\}$, from the 2D encoder, we obtain 3D features $\{W_t\}$, from the 3D encoder, and we obtain object-level features $\{Y_t\}$, from the object detection model. We use a temporal transformer to capture correlations across frame-level features and then integrate these with 3D and object-level features via a cross-transformer that operates on their concatenation. Finally, we normalize the resulting vector coming from the cross transformer, as well as the text features, so we can measure the cosine similarity between the two and obtain a vector with the likelihood of the video being from each class. The maximum value is the most likely class. Another option is utilizing an MLP head as a last step for classification.

4.1 Clip

CLIP [3] is used in this architecture for two roles: encoding the label prompts and encoding individual video frames. For the text side, we tokenize and encode each prompt using the CLIP text encoder, and project them into a common embedding space via a learned linear projection. This process only needs to be done once. For the video side, each frame is passed independently through CLIP's vision encoder. The resulting embeddings are then linearly projected and temporally summarized using a temporal transformer. This allows the model to capture time-aware 2D visual semantics of the video, aligned with the language space. The frames must be resized to a size of 244 x 244 *px*.

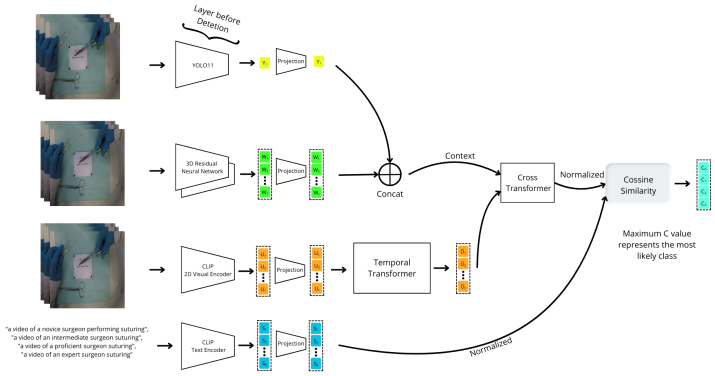

Fig. 3. Overall system pipeline of the designed Model

Mathematically, given a set of T video frames ${x_t}_{t=1}^{T}$, the CLIP image encoder produces a sequence of features U_t, which is passed through a temporal transformer to produce a summary representation.

$$\bar{U} = TemporalTransformer(CLIP(x_t)_{t=1}^{T}) \tag{1}$$

The class text prompts ${p_k}_{k=1}^{4}$ are tokenized and encoded as:

$$s = TextProj(CLIPTextEncoder(s_k)_{k=1}) \tag{2}$$

4.2 R3D

The R3D (ResNet3D-18) network processes video clips in a spatiotemporal fashion. Each video is divided into S subclips, and each subclip is a tensor of shape (C, T, H, W). The R3D model captures motion and temporal dynamics over short-term windows. After feature extraction, the final fully connected layer is removed and replaced with an identity layer that allows a projection layer to map the 3D features into the shared embedding space. The clips must be resized to a size of 244 x 244 *px*.

Let w_i be the 3D feature of the i^{th} subclip:

$$W = R3DProj(R3D(z_i)) \tag{3}$$

where z_i is the i^{th} video subclip. The collection of S projected vectors ${W_i}_{i=1}^{S}$ is used as part of the context in the fusion stage.

4.3 Yolo

YOLO is used in this model not for detection outputs, but to extract semantically rich visual features from object-centric backbones. Instead of relying on detection

heads, we attach a forward hook to a stable intermediate layer of the YOLO backbone. The input is a resized video frame, 640 x 640 *px*, per sample, and the extracted features undergo global average pooling and are linearly projected into the shared embedding space.

Formally, let f be the YOLO frame, and let $\phi(f)$ be the hooked feature map:

$$Y = YOLOProj(GAP(\phi(f))) \tag{4}$$

This yields a vector encoding coarse object-level scene understanding complementary to the motion and frame-level appearance features.

For Task 3, which focuses on tracking surgical objects and hands, our approach was to leverage the YOLO architecture due to its proven efficiency in real-time object detection. The main consideration was whether to use YOLOv8 or YOLOv11. Based on official benchmarks [4], we selected YOLOv11, as it offers higher theoretical accuracy and speed compared to YOLOv8. Given that YOLO speed was not the most relevant part, prioritizing accuracy was more suitable for our application.

To enable robust tracking of hands and instruments, we initially considered using MediaPipe for hand annotation. However, MediaPipe struggled with accurate detection due to the presence of surgical gloves. Consequently, we opted for two specialized datasets: *Labeled-surgical-tools* [5] for instruments and *Surgical-hands* [6] for gloved hands. We merged these annotations into a unified YOLO-format dataset comprising six classes: needle, clamp, curved scissor, straight scissor, left hand, and right hand.

To enhance tracking performance, we incorporated segmentation masks alongside bounding box annotations. Unlike bounding boxes, which only provide coarse localization by enclosing objects within rectangular regions, segmentation offers pixel-level delineation of object shapes. This allows the model to better distinguish between similarly shaped tools and to handle occlusions more effectively, as it can focus on the precise contours of each object. In cluttered surgical scenes, where tools and hands often overlap, segmentation significantly improves object association across frames, enabling more accurate and robust multi-object tracking.

Phase 1: COCO Pre-training with Frozen Backbone. The model was initialized with COCO-pretrained weights, freezing the backbone layers (0–9). This retained general visual features learned from large-scale data, while adapting the detection head to the surgical domain.

Phase 2: Full Fine-tuning on Surgical Data. All layers were unfrozen, allowing the entire network to adapt to the specific visual characteristics of surgical scenes.

Phase 3: Final Adaptation with Strong Regularization. The final phase involved strong regularization and advanced data augmentations (color jitter, geometric transformations, mosaic, mixup, copy-paste) to improve generalization and robustness. All layers remained unfrozen, and hyperparameters were adjusted for stability (lower learning rate, higher momentum, increased weight decay).

This multi-phase strategy ensures the model leverages generic features, progressively adapts to the surgical context, and is regularized to prevent overfit-

ting. Only the first phase uses a frozen backbone; all subsequent phases allow full domain adaptation, following best practices in transfer learning for medical computer vision.

Initially, to evaluate object detection alone, we used the MAP metric. However, since the official challenge metric was not yet released, we adopted HOTA (Higher Order Tracking Accuracy) for a comprehensive evaluation. HOTA balances detection and association accuracy, overcoming limitations of traditional metrics like MOTA and MOTP. It is well-suited for multi-object tracking in complex surgical scenes, ensuring consistent identity preservation of hands and tools.

4.4 Cross Transformer

The CrossTransformer integrates the temporally summarized 2D features from CLIP with the 3D motion and YOLO object-centric context. Specifically, the CLIP-based temporal summary acts as the query, and the R3D and YOLO projections act as the context. A single cross-attention transformer layer updates the query based on the context, allowing the model to attend to object and motion patterns relevant to the high-level frame semantics.

Given $\bar{U}$ and context W, Y the transformer fuses these as:

$$v = CrossTransformer(query = \bar{U}, context = [W, Y]) \tag{5}$$

The output v is a single fused video embedding.

4.5 Cosine Similarity

Both the fused video representation v and the class prompt embeddings s_k are L2-normalized before comparison. Classification is performed using cosine similarity, following the approach of CLIP. The predicted class is the one with the highest similarity score:

$$y = \arg\max_k \left(\exp(\text{logit}) \cdot \cos(v_k, s_k)\right) \tag{6}$$

This enables prompt-based classification in a unified embedding space, allowing the model to flexibly adapt to new prompts or reworded class descriptions.

4.6 MLP Head

While the cosine similarity approach enables prompt-based classification in a zero-shot or prompt-adaptable manner, the second task posed a different requirement: to classify each of the eight OSATS categories into one of five discrete skill levels (ranging from 0 to 4). Since each OSATS dimension is treated as an independent classification problem, the prompt-based approach is less applicable.

To address this, we replaced the cosine similarity layer with a dedicated Multi-Layer Perceptron (MLP) head. This head is composed of a single linear

projection that maps the fused video embedding into a flattened output space of shape (8×5), corresponding to the eight OSATS tasks and five discrete classes per task:

$$\hat{y} = \text{MLP}(v) \in \mathbb{R}^{8\times 5} \tag{7}$$

The output is reshaped to separate the predictions per task, and a softmax is implicitly applied via the cross-entropy loss function for each task. Each OSATS task contributes its own independent classification loss, and the total loss is computed as the mean across all tasks:

$$\mathcal{L} = \frac{1}{8}\sum_{i=1}^{8} \text{CE}(\hat{y}_i, y_i) \tag{8}$$

This MLP-based formulation is better aligned with the multi-label nature of Task 2, where multiple categorical predictions are made for each video sample. It enables the network to learn OSATS-specific representations while maintaining the shared backbone and temporal integration pipeline. Later on, we used a similar head for Task 1, due to a more trainable approach than the cosine similarity. Naturally, the number of categories was 1 instead of 8.

4.7 Experimental Settings

To evaluate our model, we utilized the dataset provided for the MICCAI 2024 Suturing Skill Assessment Challenge. This dataset, curated under controlled conditions, includes labeled video recordings representing various levels of suturing proficiency. The 2025 challenge dataset was not publicly available at the time of this study, and therefore was not used.

We adhered to the official evaluation protocols and metrics defined by the MICCAI challenge. Specifically, the F1-score and the Expected Cost were used as the primary performance indicators. The F1-score reflects the harmonic mean of precision and recall across class predictions, providing a balanced assessment of classification accuracy. The Expected Cost metric incorporates the cost-sensitive nature of misclassification, which is particularly relevant in educational contexts where certain errors may have more significant implications than others.

All experiments were implemented in Python using PyTorch, with support from the Ultralytics YOLOv11 framework and OpenCLIP for vision-language modeling. Pretrained weights were used for all encoders to leverage transfer learning. The system was developed and tested on a 2020 MacBook with an Apple M1 chip for initial prototyping and debugging. Final training and evaluation were conducted on a workstation equipped with an NVIDIA RTX A6000 GPU, enabling efficient processing of the video-based inputs. The training set comprised 80% of the available data, while the remaining 20% was reserved for validation, with all random splits performed using a fixed seed for reproducibility. The core hyperparameters used throughout training included a batch size of 8, later down scaled to 4, 8 sampled 2D frames per video, 4 temporal 3D clips

per video, and 16 frames per 3D clip. All intermediate visual representations were projected into a common embedding space of 512 dimensions. For the large model, the parameters were scaled so that a later scaled to 1024. For the small model, it was scaled to 128. All model variants were trained using the same core hyperparameters. The training process used the Adam optimizer with a learning rate of 0.001 and L2 weight decay of 1×10^{-4}. Each model was trained with at least 5 epochs, depending on the learning curve or resource availability. The number of output classes for Task 1 was set to 4. Model checkpoints were saved automatically during training. Cross-entropy was used as a loss function.

4.8 Models

For Task 1, we experimented with several model variants to explore classification performance under different supervision constraints. These included: a frozen model using cosine similarity, an unfrozen variant with an MLP head, a model using the full pipeline without the YOLO component, and a version using only CLIP features with an MLP head. These variants are detailed in the results section.

5 Results

In this section, we present the outcomes of our experiments across the three tasks defined in the challenge. Multiple model variants were tested, each differing in architecture and training strategies. The results were influenced by practical constraints, including limited computational resources and time. These limitations impacted both the depth of experimentation and the extent of model optimization. The following subsections detail the performance metrics and observations for each task.

5.1 Task 1 Performance

The results for Task 1 (see Table 1) did not meet our expectations with regard to the evaluation metrics. Across all model variants, we observed a consistent ceiling in performance, with the maximum achievable F1-score plateauing at 0.161. Despite architectural differences, all models eventually converged to this upper bound. Notably, larger models or those with more expert-level components reached this performance ceiling more quickly during training, suggesting that increased model capacity accelerated convergence but did not lead to higher final performance.

A detailed analysis of the classification outputs revealed that all models tended to over-predict the labels "0" and "2", which correspond to "novice" and "proficient" respectively. These categories were also the most represented in the training dataset, as seen in the Figure 1, suggesting that class imbalance significantly influenced the learning dynamics. This overuse of dominant

classes indicates that the models may have defaulted to frequency-based heuristics rather than learning discriminative skill features from the videos.

These observations lead us to conclude that the current models exhibited limited capacity to extract meaningful representations for Task 1. This limitation could be attributed to two main factors: the pretrained visual encoders may not have been adequately tuned for surgical video semantics, and the unbalanced class distribution likely skewed the optimization process.

Model Designation	F1-score	Expected cost	Epoch
Frozen CLIP + R3D + YOLO	0.161	0.386	6
CLIP + R3D + YOLO + MLP	0.161	0.302	2
CLIP + R3D + MLP	0.161	0.302	3
CLIP + MLP	0.161	0.302	4
CLIP + R3D + YOLO + MLP Large	0.161	0.302	1

Table 1. Task 1 Results

5.2 Task 2 Performance

Despite incorporating 3D semantics through the R3D module, the performance on Task 2 did not meet our expectations (see Table 2). The addition of temporal features was anticipated to enhance classification accuracy, yet the observed F1-score remained modest. Unfortunately, we lack comparable benchmarks from other submissions for this task, making relative performance assessment difficult. However, in contrast with models from last year's challenge, our results fall noticeably short. The loss curve behavior mirrored that of Task 1, suggesting persistent issues with data imbalance.

Model Designation	F1-score	Expected cost	Epoch
Frozen CLIP + R3D + YOLO	0.118	-	9
Frozen CLIP + R3D + YOLO Small	0.118	-	0

Table 2. Task 2 Results

5.3 Task 3 Performance

As shown in the table below, the models achieve strong results when detecting tools on images from the test dataset subset. However, this high performance does not fully translate to video frames, where the models still face challenges. In particular, detection accuracy drops for objects in uncommon positions, such as open scissors or tools with non-standard colors. Similarly, hand detection remains difficult in surgical contexts, with the models struggling to reliably identify hands, particularly when occlusions or unusual hand poses are present.

Model Designation	mAP@0.5 Overall	Inference Speed
YoloFineTune	0.9788	0.9 ms
YoloTrackingWithoutHands	0.8642	2.1 ms
yoloTrackingWithHands	0.9897	0.8 ms

Table 3. Performance results of different YOLO models for Task 3.

We also managed to develop a segmentation-based model and tested its tracking, which achieved a HOTA score of 0.6312. However, these results are not as promising as they might initially seem. Upon closer inspection, it appears that the model effectively learned to distinguish only one of the objects, while achieving a very low precision (0.013). This extremely low precision indicates a large number of false positives in the model's detections.

6 Future Work

Despite the potential of our proposed models, the observed limitations in performance, particularly in Task 1, highlight several promising directions for future research and development.

One immediate step is to incorporate the MICCAI 2025 dataset once it becomes available. This dataset may exhibit a more balanced class distribution and broader sample diversity, potentially addressing the overfitting and class bias issues identified in our current models. Re-training and evaluation on this newer dataset could offer a more accurate benchmark for our approach. If the new dataset continues to show imbalance, future work should explore alternative datasets from public sources or other surgical training platforms to supplement training.

Our current implementation relies on pretrained encoders not specifically tuned for surgical video semantics. Future iterations could benefit from domain-specific visual backbones or models pretrained on medical or procedural datasets.

Another direction involves scaling the model architecture to increase or decrease its learning capacity. Deeper or wider transformer blocks, enhanced fusion layers, and larger embedding spaces could allow the system to better capture nuanced skill expressions across multiple visual modalities. On the other hand, due to the use of already powerful and computationally intensive models, down scaling could offer a practical alternative. A more lightweight architecture might reduce overfitting, especially given the limited dataset size, and allow for faster experimentation and deployment without significant loss in accuracy.

In our current setup, YOLO operates on individual frames, limiting its temporal understanding. Future iterations should investigate feeding multiple frames into the YOLO encoder to capture temporal object patterns, such as coordinated tool usage or hand transitions, which are critical for accurate skill assessment.

A Appendix

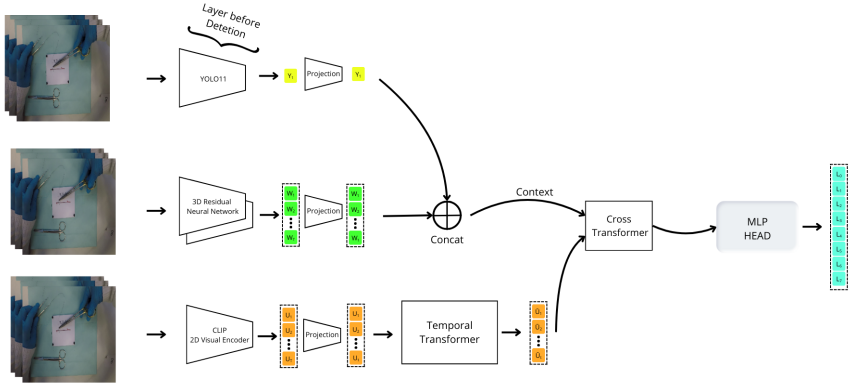

Fig. 4. Architeture with MLP Head

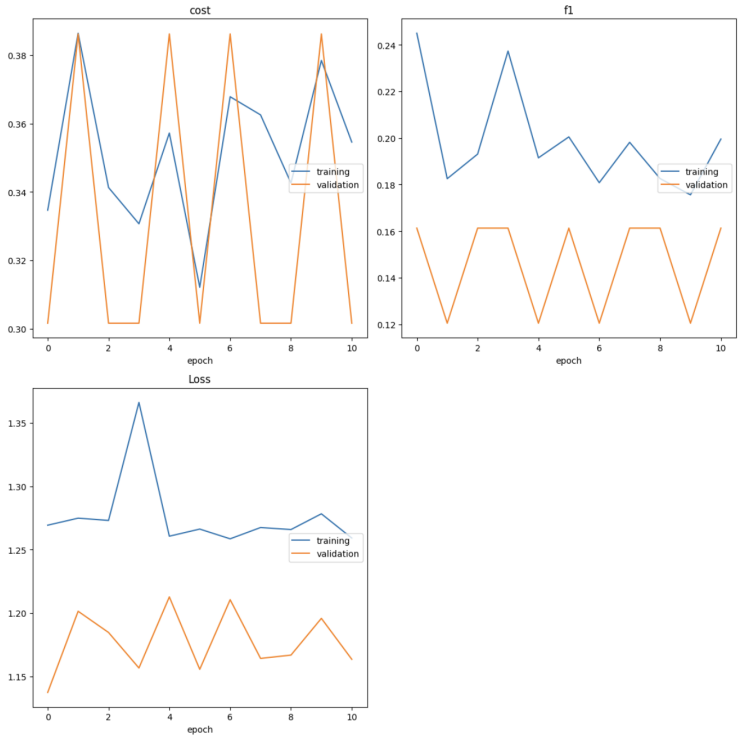

Fig. 5. Results for Frozen with Cosine Model

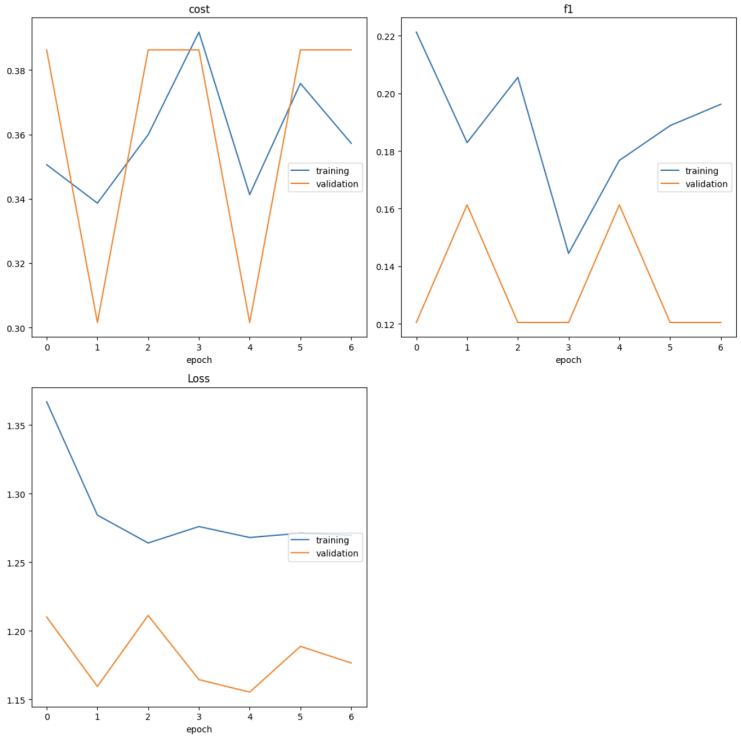

Fig. 6. Results for Unfrozen with MLP Head Model

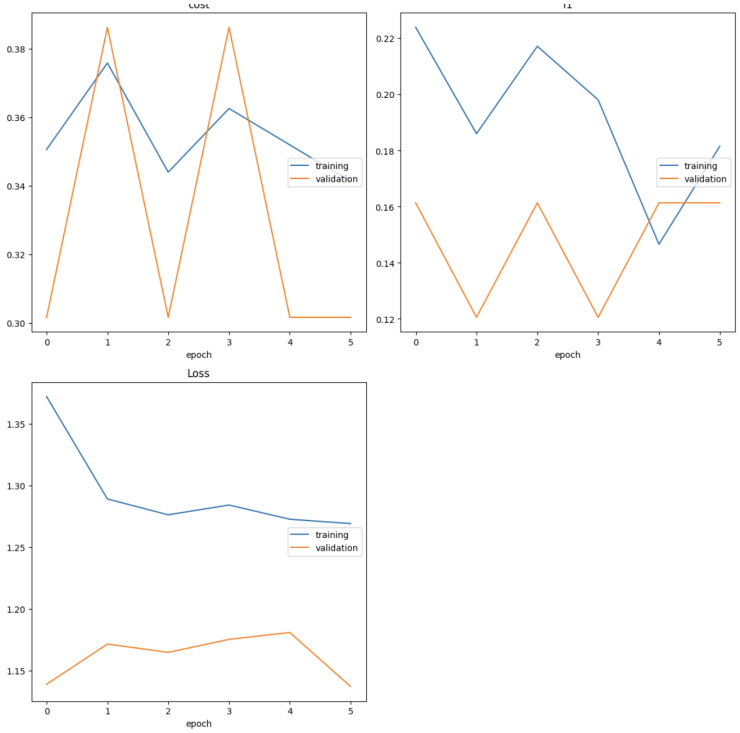

Fig. 7. Results for Unfrozen with No YOLO MLP Head:Model

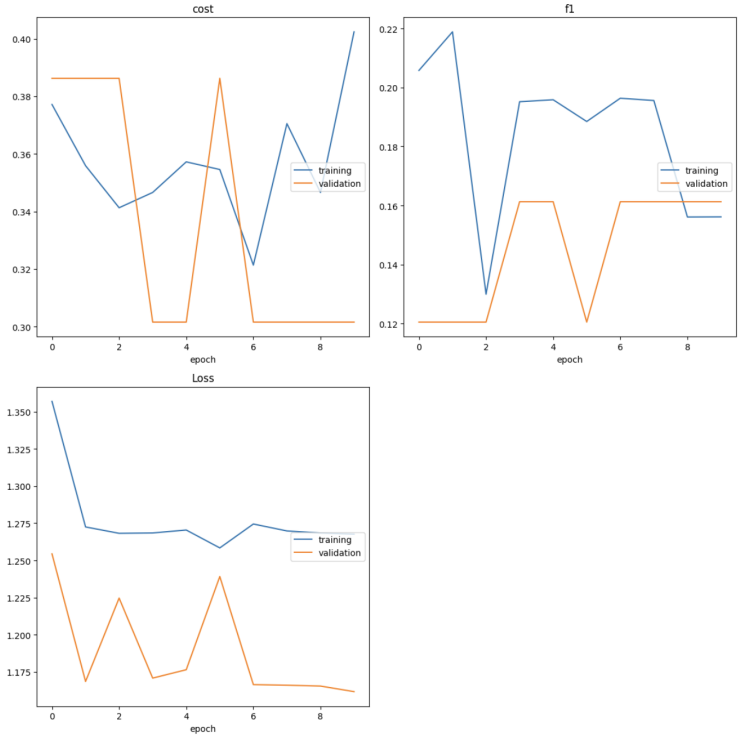

Fig. 8. Results for Just CLIP with MLP Head Model

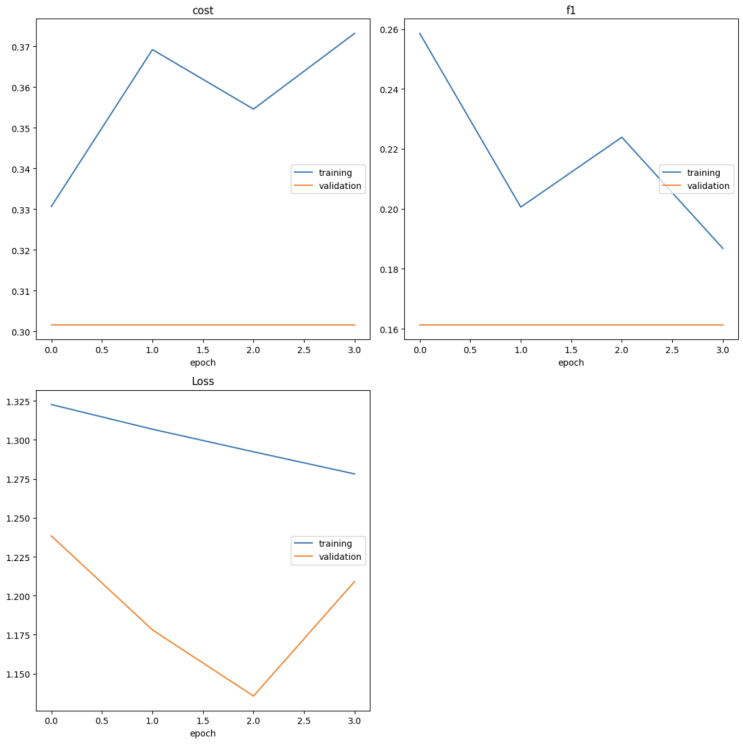

Fig. 9. Results for Large Unfrozen with MLP Head Model

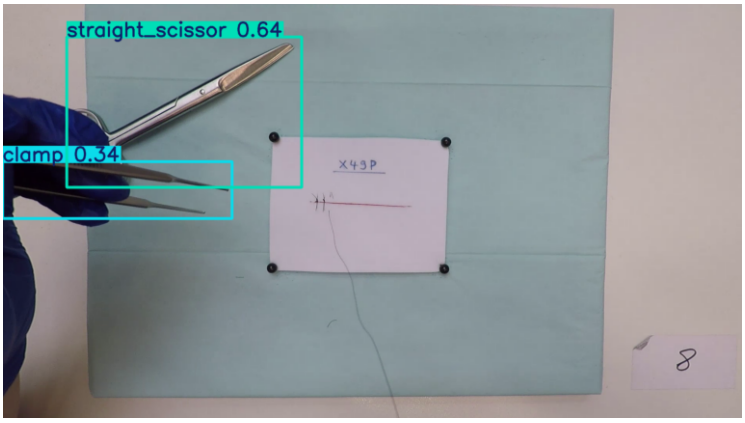

Fig. 10. Task 3: Video Frame Detection

B Research

For this appendix section, we showcase some of the research conducted for the paper.

B.1 Keywords

As for the keywords used for research, we used: Open Suturing, Skill Assessment, Video, Deep learning, Vision-language Models

B.2 TSN for Classification

In the study "AIxSuture: Vision-Based Assessment of Open Suturing Skills," [7] the authors evaluate the use of Temporal Segment Networks (TSNs) for classifying surgical skill levels in open suturing tasks based solely on video data. Two different backbone architectures were tested: I3D (Inflated 3D ConvNet) and the Video Swin Transformer. Both approaches yielded strong classification results across the three defined skill levels: novice, intermediate, and proficient, with the Video Swin Transformer slightly outperforming I3D in terms of macro-averaged F1 score. However, the Swin model is considerably more computationally demanding and, due to hardware limitations, was only partially fine-tuned. This partial training likely affected its ability to generalize fully from the data.

The authors emphasize that no preprocessing was applied to the videos beyond resizing, and no hand-crafted features were extracted. This end-to-end approach simplifies the workflow and avoids the need for additional data sources such as kinematic or force measurements. The TSN framework divides each video into segments, from which short frame snippets are sampled and passed to the backbone for feature extraction. The final prediction for each video is obtained by aggregating snippet-level predictions. Despite the relatively simple preprocessing pipeline, the models achieved an F1 scores as high as 72%, performance that is comparable to human raters.

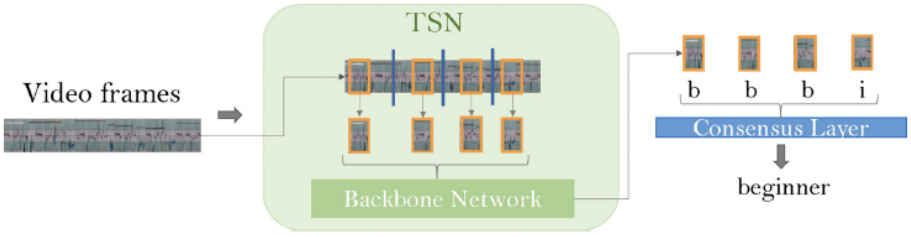

Fig. 11. Overview of the model architecture used in AIxSuture, from Hoffmann et al. (2024) [7].

B.3 Yolo based Classification

In the study "Video-based Fully Automatic Assessment of Open Surgery Suturing Skills," [8] the authors propose a video-based approach for assessing surgical skill by analyzing tool and hand motion in open suturing tasks using computer vision. They employed a modified YOLOv3 architecture, adapted into a multi-task network capable of simultaneously detecting individual tools, hands, and hand-tool interactions from webcam video data. This dual-task structure avoids the computational overhead of running two separate networks while maintaining high detection performance. The detection results form the basis for extracting kinematic features that describe motion patterns during suturing.

Using the output of the multitask YOLO network, the authors computed several well-established motion-based metrics: procedure duration, path length, and number of hand movements. These features quantify aspects of surgical efficiency and coordination. In addition, two new metrics were introduced based on the orientation of the forceps: the mean and standard deviation of the aspect ratio of their bounding boxes throughout the task. These metrics aimed to capture qualitative differences in tool handling technique between novice and expert surgeons. Notably, the system revealed significant differences across experience levels, such as shorter execution times, more stable grip angles, and reduced path lengths among expert participants.

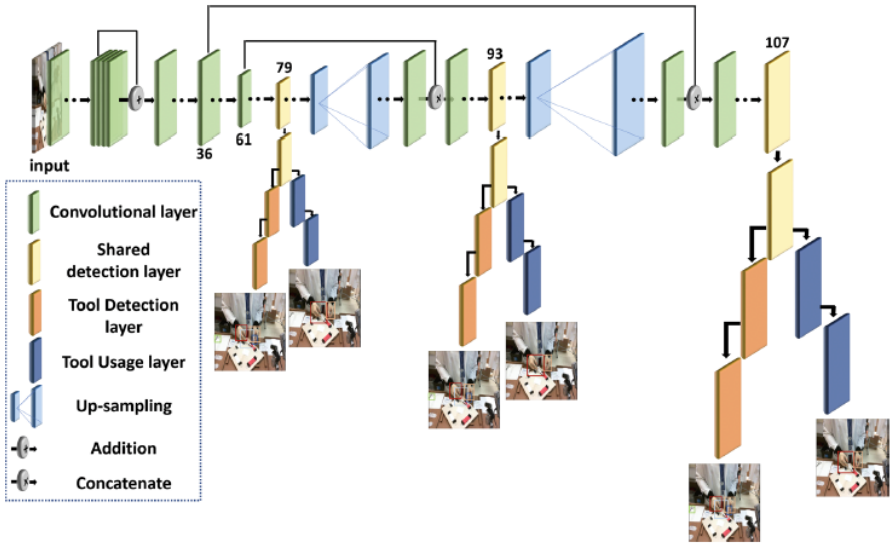

Fig. 12. Overview of the multi-task YOLO-based architecture from Goldbraikh et al. (2022) [8].

B.4 Binary Classification Using ConvLSTM and Kinetic Data

In the study "Capturing Fine-Grained Details for Video-Based Automation of Suturing Skills Assessment" [9], the authors propose a novel computer vision framework that aims to automate the evaluation of suturing technical skill using surgical video data. Although the problem tackled by the authors differs from our challenge, since the videos are manually segmented into specific sub-stitch phases and each segment is labeled as either “ideal” or “non ideal", their methodological contributions offer valuable insights for related work. Each video was manually divided into four sub-stitch actions: needle handling, targeting, driving, and withdrawal. Each corresponding phase was evaluated independently for technical quality. While this fine-grained segmentation approach diverges from more holistic classification methods, the authors argue it provides a more precise and interpretable form of assessment, particularly beneficial when addressing subtle skill deficiencies.

To support this objective, the authors integrated instrument kinematic data (recorded at 30 Hz) during training as auxiliary supervision. Though not used at inference time, this additional signal helped guide the model to focus on spatially and temporally relevant visual patterns. Furthermore, an attention mechanism was introduced to enhance the model’s ability to capture the subtle and localized variations that distinguish high and low quality performance. These strategies were incorporated into a two-stream ConvLSTM-based pipeline, which first extracted spatiotemporal features using RGB and optical flow inputs, processed them through an attention module, and finally passed them to a classification head for skill scoring.

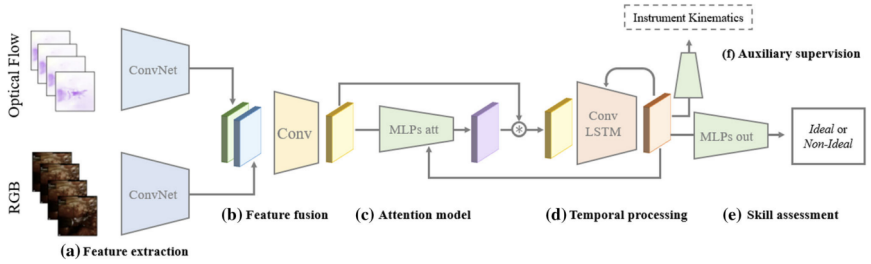

Fig. 13. Overview pipeline. From Hung et al. (2023) [9].

B.5 Classification with CNN LSTM

The study by Hashemi et al. (2025), "Video-based robotic surgical action recognition and skills assessment on porcine models using deep learning" [10], presents a deep learning framework aimed at recognizing surgical actions and assessing skill levels during robot-assisted procedures. Although the study is situated within

robotic surgery and not open surgery, its architectural design offers relevant insights. The model architecture combines a Convolutional Neural Network (CNN) for spatial feature extraction with a Long Short-Term Memory (LSTM) layer to capture temporal dynamics across video sequences. This dual-branch design allows the system to interpret surgical gestures and evaluate operator expertise over time. Key engineering strategies include input pre-processing, class balancing, and the use of regularization methods like dropout and batch normalization to combat overfitting, particularly important given the small dataset. This architecture, specifically the CNN-LSTM pairing and temporal segmentation approach, may inform similar multitask video analysis systems in open surgery contexts. Adaptations would be required to account for differences in domain semantics and visual characteristics.

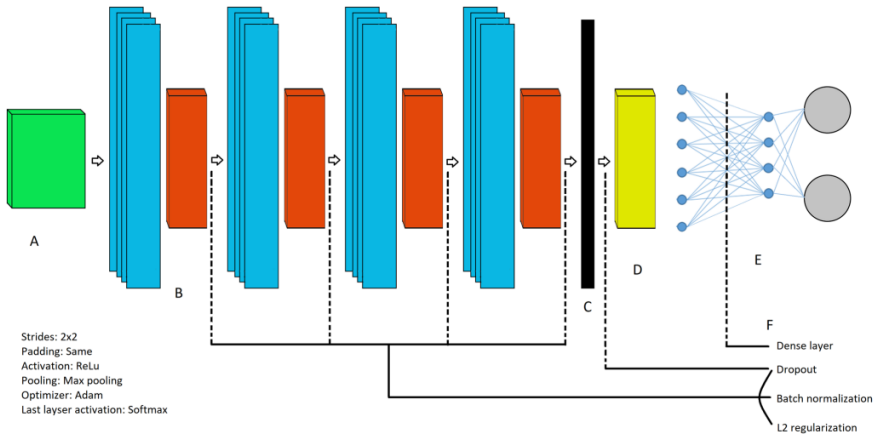

Fig. 14. Overview of the LSTM based architecture, from Hashemi et al. (2025) [10]

B.6 Experts based Classification

The paper "Efficient Mixture-of-Expert for Video-based Driver State and Physiological Multi-task Estimation in Conditional Autonomous Driving" [11] introduces VDMoE, a multitask architecture designed for driver monitoring in semi-autonomous vehicles. Although developed in a context unrelated to surgical training or skill evaluation, the architectural strategies employed, particularly for handling multi-modal and temporally structured inputs, offer transferable insights. At its core, the model integrates spatial-temporal features from facial video regions and physiological proxies into a lightweight deep learning model. A key innovation is the optimized Mixture-of-Experts framework, which separates spatial and temporal expert modules and applies a heterogeneous gating mechanism for dynamic task-specific feature routing. This is further reinforced by prior-driven regularization to encourage realistic joint predictions across mul-

tiple cognitive and physiological states. Despite its different application domain, the paper demonstrates a modular, scalable approach to multi-modal fusion and efficient inference principles that could inspire improvements in cross-task learning and real-time classification in video-based surgical training systems.

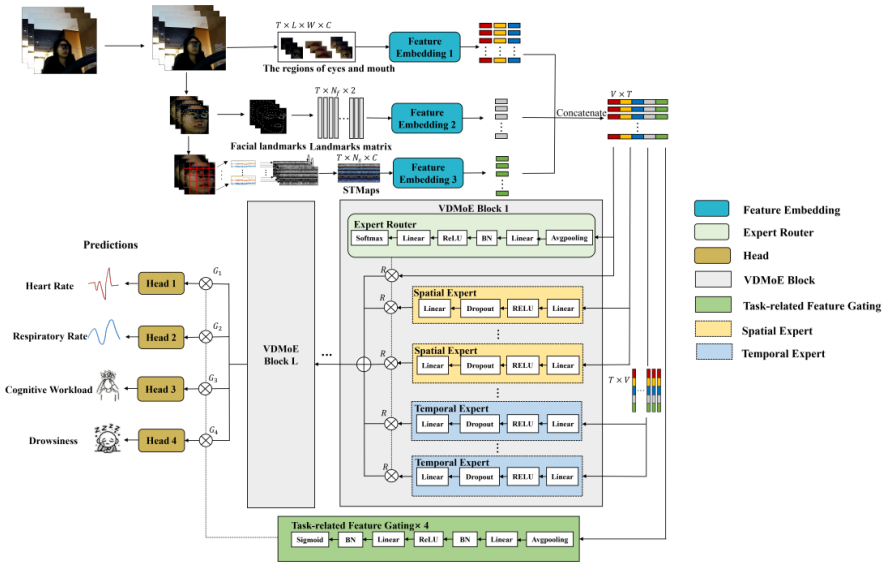

Fig. 15. Overview architecture, adaptada de Wang et al. (2024) [11]

References

1. Hsia, L. H., Hwang, G. J., & Hwang, J. P. (2023). AI-facilitated reflective practice in physical education: an auto-assessment and feedback approach. Interactive Learning Environments, 32(9), 5267–5286. https://doi.org/10.1080/10494820.2023.2212712
2. N. A. Tu and N. Aikyn, "Improving Vision-Language Models With Attention Mechanisms for Aerial Video Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 22, pp. 1-5, 2025, Art no. 8000505, doi: 10.1109/LGRS.2025.3532987. https://doi.org/10.1109/LGRS.2025.3532987.
3. Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PmLR, 2021. https://proceedings.mlr.press/v139/radford21a.html
4. Ultralytics. (n.d.). *Benchmarking*. Retrieved May 25, 2025, from https://docs.ultralytics.com/pt/modes/benchmark/
5. Labeled Surgical Tools [Dataset]. Kaggle. From https://www.kaggle.com/datasets/dilavado/labeled-surgical-tools/data
6. MichiganCOG. Surgical Hands Release [Dataset]. GitHub. From https://github.com/MichiganCOG/Surgical_Hands_RELEASE?tab=readme-ov-file

7. Hoffmann, H., Funke, I., Peters, P. et al. AIxSuture: vision-based assessment of open suturing skills. Int J CARS 19, 1045–1052 (2024). https://doi.org/10.1007/s11548-024-03093-3
8. Goldbraikh, A., D'Angelo, AL., Pugh, C.M. et al. Video-based fully automatic assessment of open surgery suturing skills. Int J CARS 17, 437–448 (2022). https://doi.org/10.1007/s11548-022-02559-6
9. Hung, A.J., Bao, R., Sunmola, I.O. et al. Capturing fine-grained details for video-based automation of suturing skills assessment. Int J CARS 18, 545–552 (2023). https://doi.org/10.1007/s11548-022-02778-x
10. Hashemi, N., Mose, M., Østergaard, L.R. et al. Video-based robotic surgical action recognition and skills assessment on porcine models using deep learning. Surg Endosc 39, 1709–1719 (2025). https://doi.org/10.1007/s00464-024-11486-3
11. Wang, Jiyao, et al. "Efficient mixture-of-expert for video-based driver state and physiological multi-task estimation in conditional autonomous driving." arXiv preprint arXiv:2410.21086 (2024).