Endoscopic Vision Challenge 2025: Open Suturing Skills

Guilherme Barbosa[1], Rodrigo Ralha[1], Rafael Peixoto[1], João Carvalho[1], and Diogo Ferreira[1]

Universidade do Minho, Rua da Universidade, 4710-057 Braga, Portugal
https://www.uminho.pt

Abstract. This report presents the methodology, preprocessing steps, and models used to address the three proposed tasks related to surgical skill assessment from video data. We explore a variety of architectures, including different ResNet variants, convolutional neural networks (CNNs), and object detection models such as YOLO.
YOLO was applied to detect surgical instruments and regions of interest within video frames, providing spatial context for subsequent classification tasks. Models such as ResNet34, ResNet50, and Inception were tested, some combined with LSTMs to capture temporal information. The report concludes with a discussion of the results, limitations, and possible future improvements, including the use of domain-specific pretrained models and enhanced temporal modeling.

Keywords: Classification · Second keyword · Another keyword.

1 Introduction

The assessment of surgical skills from video data represents a critical challenge in medical education, with significant implications for training and certification processes.

The ability to objectively evaluate technical proficiency, such as suturing techniques, can enhance the standardization and reproducibility of skill assessments, addressing limitations in traditional subjective evaluations.

Recent advancements in deep learning and computer vision offer promising avenues for automating such assessments, leveraging models capable of extracting spatial and temporal features from complex video sequences. This report addresses the MICCAI 2025 challenge, focusing on three distinct tasks: classification of the Global Rating Score (GRS), prediction of Objective Structured Assessment of Technical Skill (OSATS) categories, and tracking of hands and tools via keypoint detection.

Our approach integrates a range of deep learning architectures, including convolutional neural networks (CNNs) such as ResNet34, ResNet50, and InceptionV3, alongside object detection models like YOLO and temporal modeling through Long Short-Term Memory (LSTM) networks. These models were selected to capture both spatial details within individual frames and temporal dynamics across video sequences, crucial for evaluating surgical performance. Due to computational constraints, our experiments prioritized functional validation over exhaustive hyperparameter optimization, limiting training to a single epoch and a subset of the dataset. This report details the methodology, preprocessing steps, model architectures, and results, while also discussing limitations and potential avenues for future improvement, such as the incorporation of domain-specific pretrained models and enhanced temporal modeling strategies.

The remainder of this paper is structured as follows. Section 2 details the methodology that guided the development of this work. Section 3 presents information about the dataset provided by the official challenged and how we preprocessed it. Section 4 presents detailed information regarding the models developed or theorized by the group to solve each of the proposed tasks. Section 5 presents the results attained by each model. Finally, Section 6 concludes this work and states possible actions to improve the conceived models.

2 Methodology

The codebase was developed within a Jupyter Notebook environment and executed using Google Colab. Due to inherent limitations in computational credits provided by this platform, model training was restricted to a single epoch. This decision prioritized the validation of model functionality over hyperparameter tuning aimed at achieving peak results. The general structure of each script closely mirrored the practices established during the second phase of this course.

For the development of each model, our initial step involved a thorough data analysis, which included a review of the original published paper by Hoffmann et al. [1]. Subsequently, a preprocessing pipeline was applied to the data, incorporating filters. The processed data, either in its original form or after specific filtering, was then provided to individual models based on their distinct requirements.

3 Dataset

The dataset employed in this study comprises video data capturing surgical procedures, with a specific focus on suturing tasks. Each video is recorded at an original resolution of 1920×1080 pixels and a frame rate of 29.97 frames per second (fps). To facilitate analysis and model training, an exploratory data analysis was conducted to assess the dataset's characteristics, followed by targeted preprocessing steps to optimize the data for the proposed tasks.

3.1 Pre-processing

To prepare the dataset for model training, we focused on enhancing the visibility of the suture thread within the video frames through a key preprocessing step. Initially, we experimented with various filters, as outlined in 1, and ultimately selected the Prewitt filter for its effectiveness. This technique was adopted experimentally to emphasize critical features necessary for accurate classification and detection tasks. While it currently lacks formal scientific validation, this approach proved valuable in highlighting essential details for our model's performance.

Subsequently, the resolution of the frames was downscaled from 1920×1080 pixels to 960×960 pixels, reducing computational demands while retaining essential visual information. Additionally, the frame rate of the original videos was reduced from 29.97 fps to 5 fps, further decreasing the dataset size and enabling more efficient processing within the constraints of our computational resources. These preprocessing measures ensured the dataset was appropriately formatted for subsequent model training and evaluation phases, with each model capable of adapting the video data if further optimizations are required.

For Task 1, the videos were divided into 16 equal segments, and the first frame from each segment was extracted for analysis. This method, inspired by the official dataset analysis document [1], provides a representative sample of the video content while minimizing computational overhead. The extracted frames underwent additional preprocessing, as detailed in this subsection, to enhance their compatibility with deep learning models.

4 Models

In this section, we delineate the specific tasks established by the official challenge. **Task 1** focuses on classifying the **Global Rating Score (GRS)** into one of four integer classes, ranging from 0 to 3. **Task 2** aims to categorize five distinct integer scores (0-4) across eight **Objective Structured Assessment of Technical Skill (OSATS)** categories. These categories include: `OSATS_RESPECT`, `OSATS_MOTION`, `OSATS_INSTRUMENT`, `OSATS_SUTURE`, `OSATS_FLOW`, `OSATS_KNOWLEDGE`, `OSATS_PERFORMANCE`, and `OSATSFINALQUALITY`. Finally, **Task 3** involves the tracking of hands and tools using keypoint detection. The subsequent subsections will detail the models developed for each of these respective tasks.

4.1 Task 1

The fundamental approach for capturing temporal relationships between features is supported by the work of [2]. As noted by the authors, "LSTMs are a type of RNN that allows long-term dependencies to be recognized and are therefore well suited for the classification of sequences such as videos." This principle formed the foundation for the methodologies developed across all tasks.

After establishing the strategy for temporal feature extraction, the subsequent phase focused on deriving spatial features from individual video frames. For this purpose, several alternative models were explored. Initial investigations involved evaluating the performance of established architectures such as ResNet34, ResNet50, MobileNetV2, InceptionV3, and Faster R-CNN. These models were selected based on insights gained from [2], [3], and [4], with the latter also suggesting the potential benefits of combining multiple feature extractors. The characteristics of these initial models are summarized below:

- **ResNet34**: Receives image of size (960, 960), returns 512 features.
- **ResNet50**: Receives image of size (960, 960), returns 2048 features.
- **MobileNetV2**: Receives image of size (960, 960), returns 1280 features.
- **InceptionV3**: Receives image of size (299, 299), returns 2048 features.
- **Faster R-CNN**: Receives image of size (256, 256), returns 256 features.

All the models adhere to a consistent architectural framework, as illustrated in Figure 2.

Subsequently, informed by external knowledge, the group integrated Google's InceptionV3 [2] and Meta's Segment Anything Model (SAM) into the experimental pipeline. This combined approach, denoted as **SAM + InceptionV3**, proved to be computationally intensive, exhibiting peak RAM consumption of up to 65 GB during training. Consequently, this model was trained on a substantially smaller dataset (comprising only 10 entries, stratified into training, testing, and validation subsets). Despite the reduced dataset size, the performance achieved during training was comparable to that of the previously tested models. However, it presented overfit and due to the prohibitive resource consumption, this specific approach was ultimately deemed impractical for the full-scale task. The architecture can be visualized in Figure 3 specifics are:

- **InceptionV3**: Receives image of size (299, 299), returns 2048 features.
- **SAM**: Receives image of size (1024, 1024), returns 256 features.

An alternative to SAM's resource demands was sought through the application of a pretrained U-Net model in conjunction with InceptionV3, denoted as **U-Net + InceptionV3**. However, even when utilizing the entirety of the available dataset, the performance outcomes were suboptimal, registering among the lowest results obtained throughout this task. The specifications are:

- **U-Net**: Receives image of size (224, 224), returns 512 features.
- **InceptionV3**: Receives image of size (299, 299), returns 2048 features.

Further exploratory alternatives included a Vision Transformer (ViT), and a combination of a ViT and a VGG-19 based autoencoder. While these approaches did not present significant resource consumption challenges, their performance metrics did not meet the desired expectations. The configurations explored were:

- **ViT + Feed Forward Network (FFN)**:
  - **ViT**: Receives image of size (224, 224), returns 768 features.
- **ViT + VGG19 + FFN**:
  - **ViT**: Receives image of size (224, 224), returns 768 features.
  - **VGG19**: Receives image of size (224, 224), returns 512 features.

The group also attempted to use suture characteristics [5] to perform Task 1. Initially, due to the lack of a trained YOLO model to detect sutures, it was not possible to implement the pipeline. However, the proposed approach involves the following steps: Suture detection: Use a YOLO model to detect each suture in the video, generating bounding boxes with coordinates ([x_min, y_min, x_max, y_max]). Feature extraction:

1. Area: Calculate the area of each suture from the bounding boxes, computing the mean and standard deviation of areas per frame.
2. Position: Extract the center coordinates ((x_center, y_center)) of each bounding box, normalized by the frame dimensions, to analyze spacing.
3. Number of sutures: Count the number of sutures detected per frame to measure efficiency and progress.

Finally, the extracted features will be used to train a CNN-LSTM network or to fine-tune a pre-trained LLM with LoRA. The CNN-LSTM will combine spatial feature extraction (via CNN, processing the bounding boxes or derived features) with temporal progression modeling (via LSTM). Alternatively, a pre-trained LLM, fine-tuned with LoRA, can process the same numerical metrics formatted as structured sequences. This approach enables the evaluation of the student's technique quality (through suture area and position), spacing uniformity (via distance between sutures), and efficiency (number of sutures), while considering improvement over time to assign a higher grade.

4.2 Task 2

The objective of Task 2 is to predict the following eight predefined classes:

1. OSATS_RESPECT : It evaluates the delicacy and care with which the individual handles tissues. This includes avoiding unnecessary trauma, applying appropriate force, and demonstrating awareness of tissue fragility.
2. OSATS_MOTION : This class assesses the efficiency and **fluidity of hand movements** in this context. Smooth, direct, and efficient **motions** are rewarded, whereas unnecessary or hesitant movements are subject to penalties.
3. OSATS_INSTRUMENT : It assesses the individual's ability to use surgical instruments appropriately and effectively. This includes the correct selection of instruments for each task, as well as the manner in which they are **held** and **manipulated**.
4. OSATS_SUTURE : It specifically evaluates suturing technique. This includes needle handling and passage, knot formation, suture tension, wound edge approximation, and the proper placement of stitches.
5. OSATS_FLOW : It evaluates whether the procedure is carried out in a logical, organized, and efficient manner, without unnecessary **interruptions**. This includes the ability to anticipate subsequent steps.
6. OSATS_KNOWLEDGE : It measures the individual's understanding of the procedure being performed. This includes knowledge of the anatomy, the steps of the operation, potential risks, and how to manage them.
7. OSATS_PERFORMANCE : It is a comprehensive assessment of the technical skill demonstrated throughout the entire procedure. It provides an overall impression of the individual's competence, taking all aspects into consideration.

8. OSATS_FINAL_QUALITY : It evaluates the final outcome of the task or procedure. For example, if the objective was to close a wound, the quality of the closure is assessed.

As can be observed, several of these features contain a component related to motion, which the group intends to explore further.

Model 1 - Yolo + Resnet 50 + LSTM The first model started by a suggestion made by Professor Vítor Alves to one of the group members was the use of the *YOLO (You Only Look Once)* model, in which an "anchor" would be used to extract the *ROI* ("Region of interest") of each object. The group decided to build upon this idea by passing the *ROI* ("Region of interest") detected by the *YOLO* model to another model, with the aim of predicting the aforementioned categories, given their strong relationship with the movements performed by the individuals.

Next, it was necessary to choose the model to which the *ROI* would be passed. The group opted for ResNet-50. In the article proposed by [6], the authors highlight ResNet-50's capacity to handle deeper architectures and mitigate the vanishing gradient problem. ResNet-50 has been widely used and has achieved outstanding performance across a variety of computer vision applications—an essential characteristic in the context of this project.

The group also considered the integration of an LSTM in order to capture temporal sequences more effectively.

In general, the group's idea is as follows:

- *YOLO* as an Object/Instrument Detector : *YOLO* would be applied to the video frames to detect surgical instruments (e.g., needle holders or the needle and suture thread itself) or other regions of interest (such as the "surgeon's" hands or the suturing site). *YOLO* is a model capable of identifying and localizing multiple objects in real time. Its role in this context is to provide structured information about what is present in the scene and where — for instance, the position of the instruments at each moment, whether the needle is being used, whether two instruments are interacting (as in knot tying), etc. These detection outputs could serve as cues to infer which phase of the suturing process the video is currently in.
- ResNet-50 as a Feature Extractor: ResNet-50 would be used as a pre-trained convolutional neural network to extract visual features from frames or regions of interest. ResNet-50 is a powerful image recognition network capable of transforming each frame into a high-level feature vector, capturing relevant visual patterns such as texture, motion frozen by blur, hand posture, and more.
- LSTM to process the features and finalize : The LSTM serves the purpose of analyzing how the extracted features evolve over time in order to understand the actions being performed.

Parameters and arquitecture

In the case of *YOLO*, we used a *YOLOv8* and opted to freeze its parameters and only accept detections with a confidence threshold greater than 0.25. The ResNet-50 expects Regions of Interest of size 224 × 224 pixels as input and produces a 2048-dimensional feature vector, which is used as input to the LSTM. The LSTM consists of a single layer with 512 hidden units. Of course, some of these parameters / architecture could be optimized or improved (especially the LSTM architecture), but due to computational limitations, we chose to keep things simpler. Note that a potential improvement would be the use of a YOLO model specifically trained for medical or hospital environments.

Model 2 - CNN + Transformer with Temporal Attention In the second model, we developed a model that combines the spatial feature extraction capabilities of a *CNN* with the temporal modeling of a Transformer. In the preprocessing stage, each video is converted into a tensor of shape `[B, 3, T, 256, 256]`, where `T = 64` frames are uniformly sampled over its entire duration and normalized using ImageNet statistics. Shorter videos are padded by repeating the last frame to ensure a fixed input size.

The architecture begins with a pretrained *ResNet50*, from which we remove the final classification layer so that each frame is mapped to a 2048 dimensional feature vector. These vectors are then linearly projected down to 512 dimensions, forming our temporal embeddings optimized for the Transformer. Next, we add a fixed, non trainable sinusoidal positional encoding to inject frame order information into the embeddings. At the core of the model lies a Transformer Encoder consisting of four layers with eight attention heads each. The self attention mechanism learns long range dependencies across all frames, capturing the full suturing dynamics. The Transformer's output, a sequence of 64 vectors of 512 dimensions, is then aggregated via mean pooling over time, producing a single 512 dimensional summary vector for each video. Finally, this vector feeds into a fully connected layer that performs regression to predict the eight *OSATS* scores.

During training, we set `batch size = 4` videos and used the Adam optimizer with an initial learning rate of $1 * 10^{-4}$. Due to resource constraints, we also limited training to one epoch, while retaining MAE (Mean Absolute Error) as the loss function, ideal for directly quantifying average error in *OSATS* score units. Although a single epoch does not allow for validation curve analysis or early stopping, these parameters provide a lightweight baseline for rapid iteration. The data split remains 70% training, 15% validation, and 15% test to ensure comparability with future experiments that extend the number of epochs.

4.3 Task 3

For task 3 we focused on dividing this task into two smaller but more specialized models. One for hand keypoint detection and other for tools keypoint detection. We were not able to experiment in practice with this task since the dataset was only going to be available at June 13th 2025.

Task 3 is dedicated to hands and surgical tools keypoints. This task addresses the critical need for precise tracking and localization of both hands and surgical tools while suturing. Given the dataset's unavailability until June 13, 2025 beyond the current reporting date of May 29, 2025 practical experimentation was not possible. As a result, the methodology outlined here is a theoretical design supported by insights from relevant literature [7] [8]. The approach strategically divides the task into two specialized sub-models: one for hand keypoint detection and another for tool keypoint detection, optimizing performance for each domain.

Model 1 - Hand Keypoint Detection The hand keypoint detection process begins with the input of stereo frame pairs, specifically targeting hand regions. Each frame is processed using YOLOv5 to generate bounding boxes that localize the hands.

These bounding boxes are then used to crop the relevant regions, reducing computational overhead and focusing subsequent analysis on areas of interest.

The cropped frames are fed into a pre-trained EfficientNet-B3 model, enhanced with the Noisy-Student training paradigm as described by [8]. This model, known for its robustness in handling noisy data, identifies keypoint locations with high precision. This approach aligns with findings in [8], which emphasize the efficacy of EfficientNet-B3 for robust keypoint detection in dynamic hand movements.

Model 2 - Tool Keypoint Detection For tool keypoint detection, the pipeline processes stereo frame pairs targeting surgical instruments. Similar to the hand detection process, YOLOv8 is employed to detect and localize tools within each frame, producing bounding boxes that guide the cropping step.

The cropped regions are then analyzed using the SurgeoNet architecture, as outlined in [7], which is specifically designed for real-time 3D pose estimation of articulated surgical instruments and keypoint detection.

This model integrates a Transformer component to refine keypoints, enabling precise pose detection based on the synthetic dataset used. The use of a synthetically-trained network, as proposed in [7], enhances generalization to real surgical scenarios, addressing the challenge of limited real-world annotated data. The final architecture is depicted in Figure 4.

5 Results

The results should not be considered conclusive, as the models were trained for only one epoch and using only one-tenth of the available data. Additionally, this subset was further divided into training, validation, and test sets. This limited training setup was intended solely to verify that the implemented models function correctly. With that said, the following results were obtained:

5.1 Task 1

Table 1 summarizes the accuracy metric for the models developed during Task 1. It is important to note that, due to limited training and computational resources, most models exhibited either overfitting or underfitting behavior. From the table we can conclude that the best models appear to be the most simple ones, namely using only `ResNet34`, `ResNet50`, `MobileNetV2`, `InceptionV3` and `Faster R-CNN` followed by a simple LSTM. We can also see that the segmentation model greatly influences performance, and that the visual transformer is well complemented by the VGG encoder.

5.2 Task 2

The results shown in Table 2, as expected, are poor due to training limitations, with some indications that the model is underfitting. The Macro *F1* shows weak performance in general, while the Weighted F1 confirms that the majority of classes are poorly predicted, which is expected.

The results shown in Table 3 suggest that the architecture is more accurate in evaluating objective visual metrics, such as respect for fabric and movement efficiency, but presents greater difficulty in more subjective categories, namely final quality and task knowledge, pointing to the need for refinement in future iterations. However, more intensive training of the model was needed to draw more objective conclusions.

6 Conclusions

This report detailed our exploration of deep learning methodologies for surgical skill assessment from video data, focusing on the three tasks of the MICCAI 2025 challenge: GRS classification, OSATS category prediction, and keypoint tracking. We investigated a diverse set of architectures, including various CNNs (ResNet, InceptionV3), object detection models (YOLO), and temporal modeling techniques (LSTMs, Transformers), alongside more recent approaches like SAM and ViT, to address the multifaceted nature of these tasks.

The primary limitation of this work stems from severe computational and time constraints. Consequently, for Tasks 1 and 2, all models were trained for only a single epoch on a significantly reduced subset (approximately 1/10th) of the available dataset. This constrained training regimen means the presented results (Tables 1, 2, and 3) should be interpreted primarily as a validation of model functionality and pipeline integrity rather than a definitive measure of performance potential. Behaviors like underfitting or overfitting observed are direct consequences of this limited exposure to data and training iterations. Due to these resource limitations, we were unable to conduct more extensive training or exhaustive hyperparameter optimization, which would be crucial for achieving more robust and generalizable models.

Looking ahead, several avenues for improvement are evident. Most importantly, comprehensive training with the full dataset over multiple epochs and rigorous hyperparameter tuning is essential to unlock the true capabilities of the explored models for Tasks 1 and 2. Furthermore, a critical next step is the practical implementation and training of our proposed models for Task 3 (keypoint tracking of hands and tools). Due to the dataset for this task not being available at the time of this work, we were unable to perform even initial training epochs. Once this dataset is released, dedicated efforts will be focused on training and evaluating these specialized keypoint detection models.

Beyond these immediate steps, we also propose an enhancement to the preprocessing stage for all tasks: investigating the use of a four-channel input image. This would combine the standard RGB channels with an additional channel derived from applying filters (such as the Prewitt filter mentioned for suture visibility). This approach could potentially provide the models with richer, more discriminative features, particularly for tasks reliant on fine details like suture thread detection, without significantly increasing model complexity.

In conclusion, while this study faced substantial resource limitations, it successfully established several foundational pipelines for surgical skill assessment. The explored architectures offer promising directions, and with further development, a more robust training environment and access to all necessary datasets they hold the potential to significantly contribute to objective surgical skill evaluation.

References

1. 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
2. D. R. T. Hax, P. Penava, S. Krodel, L. Razova and R. Buettner, "A Novel Hybrid Deep Learning Architecture for Dynamic Hand Gesture Recognition," in IEEE Access, vol. 12, pp. 28761-28774, 2024, doi: 10.1109/ACCESS.2024.3365274
3. Zungu, N., Olukanmi, P., & Bokoro, P. (2025). SynthSecureNet: An improved deep learning architecture with application to intelligent violence detection. Algorithms, 18(1), 39. doi:https://doi.org/10.3390/a18010039
4. Nagendra, Savinay. "Towards Designing Deep Learning Architectures for Improving Semantic Segmentation Performance." (2025).
5. Thanapon Noraset, Prawej Mahawithitwong, Wethit Dumronggittigule, Pongthep Pisarnturakit, Cherdsak Iramaneerat, Chanean Ruansetakit, Irin Chaikangwan, Nattanit Poungjantaradej, Nutcha Yodrabum: Automated measurement extraction for assessing simple suture quality in medical education.
6. Akshay Bhuvaneswari Ramakrishnan, M. Sridevi, Shriram K. Vasudevan,R.:ManikandanAmir:H.:Gandomi Optimizing brain tumor classification with hybrid CNN architecture: Balancing accuracy and efficiency through oneAPI optimization.
7. Aboukhadra, Ahmed Tawfik, et al. "SurgeoNet: Realtime 3D Pose Estimation of Articulated Surgical Instruments from Stereo Images Using a Synthetically-Trained Network." DAGM German Conference on Pattern Recognition. Cham: Springer Nature Switzerland, 2024.
8. Müller, Lucas-Raphael, et al. "Robust hand tracking for surgical telestration." International journal of computer assisted radiology and surgery 17.8 (2022): 1477-1486.

9. Ganga Prasad BASYAL David ZENG Bhaskar P. RIMAL, Development of CNN Architectures using Transfer Learning Methods for Medical Image Classification
10. G. Divya Deepak, Subraya Krishna Bhat & Arupratan Gupta, Improved CNN architecture for automated classification of skin diseases G. Divya Deepak, Subraya Krishna Bhat & Arupratan Gupta
11. C. A. Twinanda, S. Shehata, D. Mutter, M. de Mathelin and N. Padoy, "EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos," in Proceedings of MICCAI, 2016.
12. J. Funke, J. Weitz and N. Padoy, "Video-Based Skill Assessment in Robotic Surgery Using Convolutional Neural Networks," in Proceedings of MICCAI, 2019.
13. Z. Wan and R. Sznitman, "Supervising Attention for Surgical Skill Assessment in Videos," in Proceedings of ICRA, 2024.
14. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Video Swin Transformer," in Proceedings of ICCV, 2021.

7 Appendix

7.1 State of the Art

Task 1 To address the inherent tasks of this challenge, the group conducted a review of the state-of-the-art in video processing. Literature on video classification and related computer vision tasks consistently emphasized the critical role of capturing both temporal and spatial features. Specifically, [2] highlighted the application of Temporal Segment Networks (TSN) and Temporal Shift Modules (TSM) in deep learning for video analysis. The same study also indicated that most models rely on CNNs for individual frame feature extraction, with some employing a two-stream approach where extracted features are fused and then processed by either a Recurrent Neural Network (RNN) or a Long Short-Term Memory (LSTM) network for classification. This research further underscored the importance of combining temporal and spatial features to discern movement-related relationships between frames.

The architectural decision to append a Feed-Forward Network (FFN) after the final LSTM layer was informed by work referenced in [4], which described the use of a fully connected Conditional Random Fields (CRFs) appended to a CNN output for improved object localization. The adoption of Faster R-CNN was also inspired by its recognized enhanced object detection capabilities, as cited in [4].

The integration of Vision Transformers (ViT) into our methodology was similarly influenced by the work presented in [4]. The authors' discussion of ViT's multi-head self-attention mechanism suggested its potential utility for focusing on critical regions, such as the student's hands and the suture line. As articulated by the same authors, Transformers provide a versatile backbone applicable to a broad range of computer vision tasks, including image classification and dense prediction.

Furthermore, the selection of ResNet34 and its variants, along with MobileNetV2, as feature extractors was inspired by the work of [3]. This article also illuminated the benefits of employing transfer learning to accelerate training time through the utilization of pre-learned representations.

In [5], the table detailing the characteristics of sutures and their use in quality evaluation served as the primary inspiration for the proposed idea of automating suture quality assessment using these features.

Task 2 In [9], the authors review the evolution of CNN architectures for medical image classification, focusing on transfer learning. They trace the development through key challenges like *ILSVRC* and especially the *BraTS* brain tumor segmentation challenge. Early models (2013–2014) were shallow *2D CNNs*, evolving by 2015 to more specialized *2D* architectures. A major advance came in 2016 with the introduction of *3D CNNs*, improving volumetric data segmentation. From 2017, model complexity increased significantly, with ensembles like EMMA combining multiple architectures and transfer learning, and deeper networks (up to 26 layers) becoming common. By 2018, U-Net dominated, often enhanced with techniques like autoencoder-based regularization and adaptations of ResNet and DenseNet.

The authors conclude that transfer learning and ensemble methods improve CNN performance in medical image tasks and suggest future work on advanced models like *NAS U-Net* and *EfficientNet* in multi-stage ensembles.

In [10], the authors present an approach for automated classification of skin diseases using enhanced *Convolutional Neural Network* (CNN) architectures. The study is based on four well-known pre-trained CNN models: *DarkNet-53*, *ResNet-18*, *SqueezeNet*, and *EfficientNet-b0*. The main architectural contribution was the development of three modified versions of these pre-trained models—two based on *DarkNet-53* and one on ResNet-18 by adding multiple fully connected (FC) layers at the end of the original networks, combined with batch normalization and specific activation functions.

Specifically, the *DarkNet-53-2FC3* model includes two FC layers (512 and 3 neurons) with batch normalization and Swish activation, *DarkNet-53-4FC3* includes four FC layers (2048, 1024, 512, and 3 neurons) similarly equipped, and *ResNet-18-2FC3* adds two FC layers with batch normalization, ReLU activation, and a final *SoftMax* classification layer. These modifications aimed to enhance the feature-learning capabilities of the models for the targeted skin disease classification task.

Twinanda et al. [11] presented EndoNet, showing significant improvements in recognizing surgical phases and detecting tools simultaneously, which improves the segmentation of laparoscopic procedures. Funke et al .[12] combined 2D (TSN) and 3D CNN networks to classify skills in robotic surgery, showing that the inclusion of optical flows increases accuracy in the assessment of technical skills. More recently, Wan and Sznitman [13] demonstrated that seeding attention-directed supervision on instruments accelerates generalization in skill assessment tasks in surgical videos. At the same time, the family of Transformers for video, especially the Video Swin Transformer [14], has stood out for capturing long-range dependencies more efficiently than recurrent networks, offering greater robustness to variations in camera angle and lighting. In short, the current state of the art combines visual feature extraction, tool detection, and spatiotemporal attention mechanisms—paradigms that underpin the CNN+Transformer architecture adopted in this work.

Task 3 For hand and tool detection in surgical videos, a proposed approach for hand detection involves using YOLOv5 for bounding box detection, followed by cropping the region of interest and applying EfficientNet-B3, pre-trained with Noisy Student, for keypoint detection [8]. This method leverages YOLOv5's efficiency in object localization and EfficientNet-B3's robust feature extraction for precise keypoint identification, enabling detailed tracking of hand movements during surgery.

Furthermore, the SurgeoNet framework [7] provides a robust approach for surgical tool detection and tracking by leveraging a synthetically-trained network, this enhances model generalization across diverse surgical scenarios, mitigating the limitations of sparse real-world annotated datasets, well suited for complex surgical environments.

These methods highlight the importance of combining robust object detection frameworks like YOLO with advanced feature extractors and specialized preprocessing techniques to address the challenges of detecting and tracking hands and tools in dynamic surgical environments. However, the reliance on manually annotated datasets for tools poses challenges, as the format and quality of training data significantly impact model performance, and such datasets may require extensive effort to create and validate.

7.2 Figures

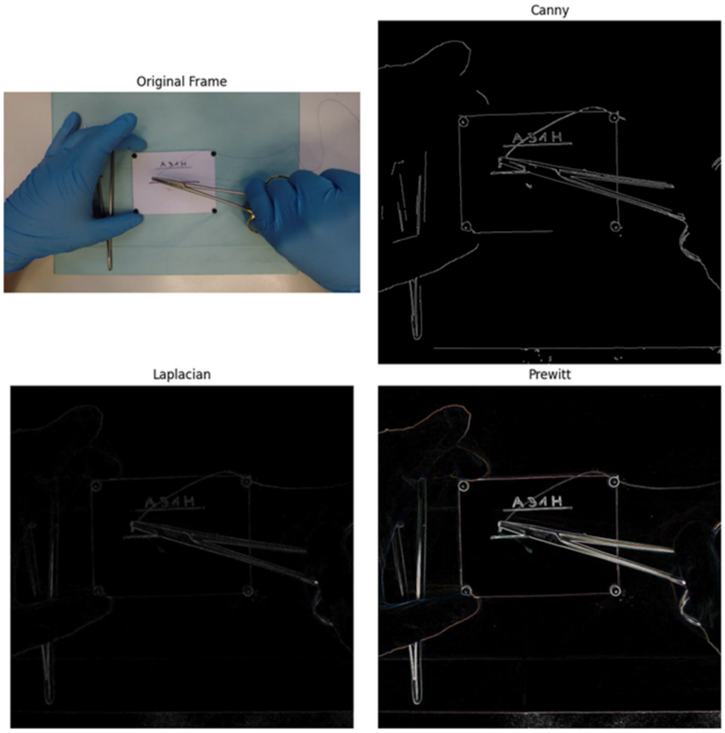

Fig. 1. Filter demonstration on a given frame

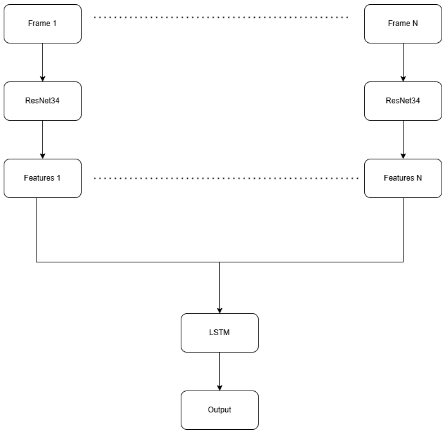

Fig. 2. ResNet34 integration architecture

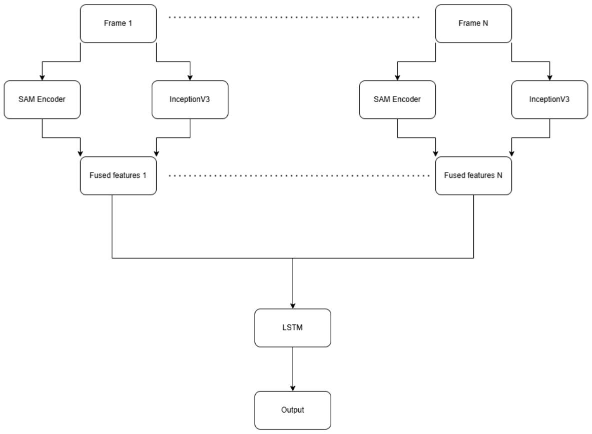

Fig. 3. SAM + InceptionV3 integration architecture

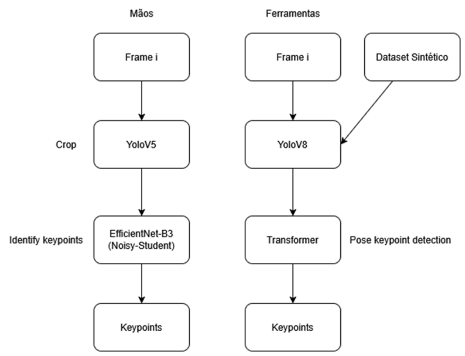

Fig. 4. Architecture Task3

7.3 Tables

Model	**Train Acc.**	**Train Loss**	**Val. Acc.**	**Val. Loss**	**Test Acc.**	**Test Loss**	**Test W. F1 Score**
ResNet34	0.46	1.34	0.54	1.35	0.50	1.29	0.3333
ResNet50	0.40	1.36	0.54	1.22	0.50	1.25	0.3333
MobileNetV2	0.44	1.37	0.54	1.25	0.50	1.27	0.3333
InceptionV3	0.43	1.62	0.00	1.14	0.50	1.20	0.3333
Faster R-CNN	0.29	1.37	0.00	1.22	0.50	1.29	0.3333
SAM + InceptionV3	0.43	1.82	0.00	2.55	0.50	0.89	0.3333
U-Net + InceptionV3	0.14	1.83	1.00	0.89	0.00	1.17	0.0000
ViT + FFN	0.14	1.31	1.00	1.22	0.00	1.31	0.0000
ViT + VGG19 + FFN	0.43	1.33	0.00	1.35	0.50	1.30	0.3333

Table 1. Performance metrics for each model (trained for 1 epoch on 1/10 of the dataset)

F1 Score	**Value (%)**
Macro Overall	14.78%
Weighted Overall	19.96%
Samples	33.04%

Table 2. Results obtained with the YOLO + ResNet50 + LSTM model (trained with 1 epoch on 1/10 of the dataset) in Task 2

OSATS Category	**F1 Score**
OSATS RESPECT	0.1270
OSATS MOTION	0.2571
OSATS INSTRUMENT	0.2571
OSATS SUTURE	0.1270
OSATS FLOW	0.0095
OSATS KNOWLEDGE	0.1429
OSATS PERFORMANCE	0.1270
OSATS FINAL QUALITY	0.0000

Table 3. Results obtained with the *CNN + Transformer with Temporal Attention* (trained with 1 epoch on 1/10 of the dataset) in Task 2