Computer Vision
Surgical Applications

00970222

Final Project Report

Noam Carmon 207234170
Shahar Hillel 207530551
Raz Biton 315507780

1 Phase 1: Synthetic Data Generation

1.1 Data Generation Pipeline

The synthetic pipeline generates labeled surgical tool images through scene setup, randomized lighting, tool placement, keypoint extraction, 2D projection with post-projection corrections, and annotation. The pipeline is:

Scene Setup and Object Loading

Scenes begin in a clean Blender environment, where surgical tools are imported as 3D mesh objects. A point light is randomized in position and intensity, and HDRI environment lighting is applied with random rotation. Material properties (e.g., roughness, metallic, IOR) are varied to simulate realistic reflections and appearances.

3D Keypoint Generation (Ground Truth)

Keypoints are computed per tool using tailored logic based on 3D mesh geometry; the method varies by tool category. **Despite minor keypoint drift, auto-labeling preserves tool structure and pose well**, so we prefer it over manual labeling.

Keypoint Computation (All Tools)

Tool	**Keypoint(s)**	**Computation (summary)**
Needle holder	top_left, top_right	Top 15% by Z; choose min X (left) and max X (right).
Needle holder	bottom_left, bottom_right	Bottom 15% by Z; choose min/max X.
Needle holder	middle_left, middle_right	Keep Z in 30–70%; pick near centroid of left/right X-percentiles.
Needle holder	joint_center	Avg of (top tips, mids); move 30% toward middle-center; nearest vertex.
Tweezers	bottom_tip	Vertex with lowest Z.
Tweezers	top_left, top_right	Split arms by median X; in each arm take top 20% by Z, pick extreme X.
Tweezers	mid_left, mid_right	Per arm, choose vertex closest to median Z.
Fallback (generic)	top_left, top_right	Top 20% by Z; pick min/max X.
Fallback (generic)	bottom_left, bottom_right	Bottom 20% by Z; pick min/max X.

2D Projection

We project the 3D keypoints to 2D using the camera intrinsics and camera poses sampled around the tools. We fix left/right labels right after projection and again at the end.

View-Dependent Labeling

To maintain consistent left/right annotation across views, projected X-values are compared, and labels swapped if needed.

Partial Visibility Strategy & Post-Projection Rules

We induce partial visibility by sampling closer, lower camera shells and/or offsetting the look-at so tools are cropped—mimicking OR framing. After projection, we drop out-of-frame keypoints and derive a bbox from the

instance mask; then we clamp bbox-out points to the edge, gently nudge any off-mask points onto the tool (skip snapping for tweezers `bottom_tip`), and enforce left/right by X-order.

Rendering and Annotation

We compute each object's bounding box from instance segmentation, composite the RGBA render over a photo background with a soft surgical spotlight, save the background-composited RGB image and a visualization image, and write one JSON record per image with the saved image name and, for each object, its id/category, 2D keypoints, and bounding box.

1.2 Implemented Variations

Instrument positioning and orientation

We place the camera around the tools but prefer a top-down view.

Lighting conditions (intensity, color, direction)

Each render randomizes a point light's position (direction) and intensity. We also use an HDRI environment with random rotation and vary the world/background brightness each try.

Background variation

Rendered frames are composited over randomly chosen photographic backgrounds.

Synthetic Images — Creative Approach 1: Photo Backgrounds, Spotlight & Pairing

It consists of **two creative variations**:

- **Surgical-room spotlight:** Add a soft radial spotlight centered on the tools to mimic OR lighting and improve visibility.
- **Multi-tool images:** Render needle holder–tweezers pairs in joint top-down scenes to reflect common two-instrument frames and create realistic occlusions.

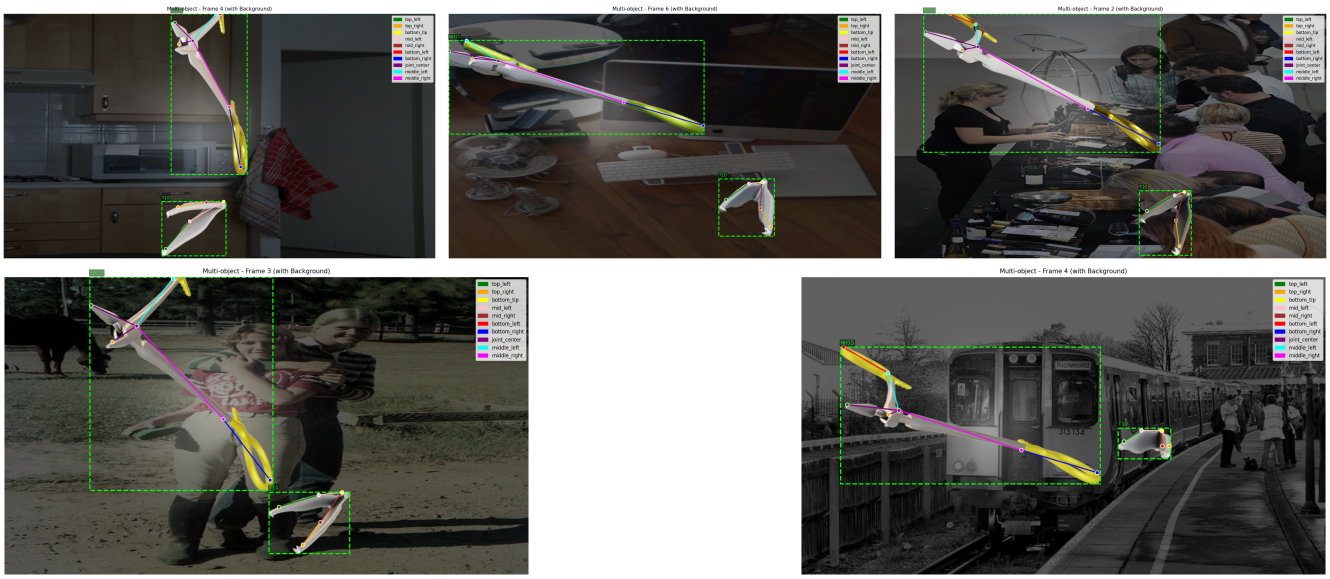

Figure 1: Approach 1 examples: spotlight and multi-tool pairing.

Synthetic Images — Creative Approach 2: Operating-Room (OR) Backgrounds

In this **creative approach** we take frames from OR videos, remove people/tools to get clean backgrounds, and composite our rendered tools on top. Then we make small changes (brightness, color, slight blur/noise, small rotations/perspective). This adds real OR context while keeping labels accurate.

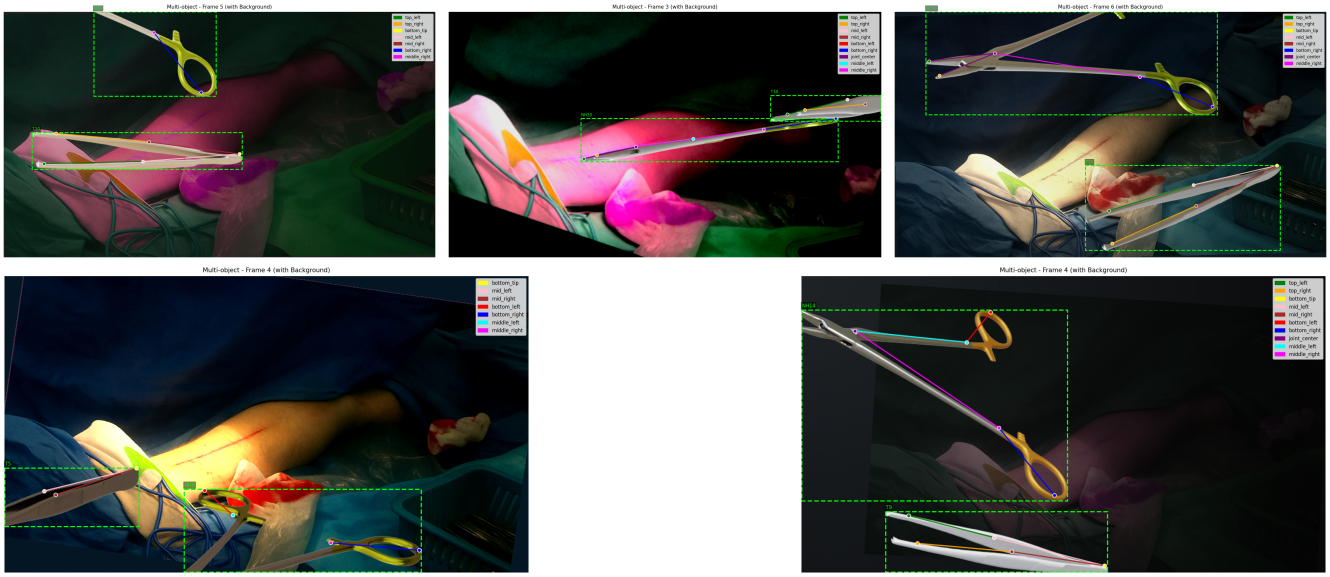

Figure 2: Approach 2 examples: tools on cleaned OR video backgrounds with light variations.

1.3 Domain Gap Analysis

Aspect	Synthetic	Real	Domain Gap
Tool Usage Context	Tools are alone and not used by hands.	Tools are held by people and touch tissue.	No hands or actions shown.
Background Semantics	Backgrounds are generic, not medical.	Surgical room environment.	Missing medical setting cues.
Texture and Surface Realism	Surfaces look clean and perfect.	There are reflections, scratches, stains, fluids.	Too perfect compared to real life.
Scene Complexity and Occlusion	Simple scenes with few overlaps.	Crowded scenes with hands and tools overlapping.	Less clutter makes training easier than it should be.

1.4 Discussion of Challenges and Key Findings

Challenges

- **Annotation Strategy:**
Manual labeling was unscalable; automation required mesh understanding and consistent landmark definitions.
- **Ground Truth Modeling:**
Bounding boxes lacked precision; keypoints added detail but required careful design.
- **Domain Alignment:**
Synthetic scenes missed surgical cues like hands, lighting, and tissue.

Key Findings

- **Reliable Auto-Annotation:**
Ground truth labels were generated using mesh metadata—accurate and scalable.
- **Multi-Tool and Multi-Pose Scenes:**
Improved realism and contextual diversity.
- **Spotlight Lighting:**
Enhanced tool visibility and mimicked surgical illumination.
- **Pose and Camera Variation:**
Supported diverse perspectives and improved generalization.

2 Model Training (YOLOv8-Pose)

Model & Data: We fine-tuned a YOLOv8-Pose model for surgical tools. Classes included tweezers and needle holders, with 10 keypoints total. Images of size 960×544 were used for letterboxing.

Training Setup:
We used a COCO-pretrained backbone, adapting the pose head for 10 keypoints. We then used `prepare_dataset.py` script to adapt the `annotation.json` of COCO format into YOLO labels format. Augmentations included horizontal flip (with `flip_idx`), light affine transforms, brightness/contrast shifts, Gaussian noise, and motion blur.

Exact YOLO Configurations:
`epochs=1000, degrees=30, flipud=0.5, fliplr=0.5, erasing=0.4, hsv_h=0.015, hsv_s=0.5, hsv_v=0.4, mosaic=1.0 mixup=0.2, optimizer=AdamW, image_size=640`

A total of 2400 images in the train set and 600 in the validation set were used for the YOLO training.

Evaluation on Validation Images:

- Logs are included in the GitHub repository under the folder: `outputs/phase2_synthetic_logs`
- Predicted validation batch images: `val_batch{i}_pred.jpg`, i0,1,2
- Training and validation metrics over epochs: `results.png`
- Confusion matrix: `confusion_matrix.png`
- Box/pose curves are included as well
- Results were strong across all training runs: 90% precision and recall
- A few less successful attempts prompted us to increase training epochs to 1000 for better robustness

Evaluation on Test Video:
We encountered several challenges during inference:

1. **Failure to Detect Surgical Tools:**
   To address this, we increased the inference image size from 640 to 1600. This helped since the original training images had much lower resolution compared to the test video. We also tried down-sizing the video resolution and using "small" videos from the shared folder—neither helped due to reduced tool visibility in already small objects.

2. **Poor Keypoint Detection:**
   The model struggled with accurate keypoint prediction. To mitigate this, we:

- Expanded and improved the training dataset
- Used the phase 3 refinement method

3. **False Positives:**
   The model occasionally predicted surgical tools where none existed. We addressed this by:
   - Improving dataset quality
   - Using a higher confidence threshold

3 Pseudo-Labeling & Refinement on Real Videos

Strategy:
We applied self-training: a trained YOLOv8 model labeled real surgical frames, and high-confidence pseudo-labels were added back into training.

This strategy was chosen due to:

- The limited availability of labeled real data.
- Its demonstrated effectiveness in HW1 and class lectures.
- The simplicity of implementation.

Filtering:
Only detections satisfying the following criteria were kept:

- Objectness score ≥ 0.7
- At least some visible keypoints
- Confidence score ≥ 0.7

Refinement Training:
We used the weights from the best synthetic-only model and continued fine-tuning on a dataset composed of both synthetic and pseudo-labeled real data. Mild augmentations were applied during training to prevent distortion of keypoint labels.

Outcome:
The refined model showed improved robustness under realistic surgical conditions, including:

- Clinical lighting variations
- Occlusions
- Tool overlap

It also produced more stable keypoint predictions with fewer swaps.

Comparative Analysis

Quantitative:

We evaluated the refined model on labeled validation images. Logs and metrics are included in the GitHub repository under `outputs/phase3_refined_logs`.

- Precision and recall for both pose and box detection reached over 95%, exceeding the performance of the synthetic-only model. This is likely due to higher-quality pseudo-labeled training data.
- This improvement can be observed in `results.png`.
- Model confidence on correct predictions was also higher, as seen in `val_batch{i}_pred.jpg`.

Qualitative:

By comparing the refined model to the synthetic model on the output test video (included in the submission), we observed:

- Higher recall: more frames detected the presence of surgical tools.
- Better performance in occluded scenarios.
- No longer necessary to increase inference resolution; running at 640 yielded strong results, likely due to the refinement data being at that resolution.
- Increased confidence in tool and keypoint detection.
- Reduced jitter and more consistent predictions between frames.

Challenges and Key Findings

- Transferring to real-world data remains difficult—especially for pose estimation—without real labeled data.
- More data and diversity improved results, but not to the level of true labeled real data.
- Scene adaptation (cropping/resizing/focusing) helped improve detection.
- Pseudo-label refinement yielded a substantial improvement in performance.
- Temporal consistency and jitter reduction still need further work.

Conclusion & Next Steps

- Successfully generated synthetic surgical videos with keypoints.
- Achieved real-video adaptation using self-training techniques.
- Persistent challenges include specular highlights, occlusions, and rare tool orientations.
- Future work may include:
  - Temporal smoothing techniques
  - Simulating harsher lighting conditions in synthetic data
  - Incorporating limited human-labeled real data
  - Exploring alternative model architectures