

Claude Code Skills Report — Scaffluent FYP Context

Context: Scaffluent is a Final Year Project building an **automated structural inspection system for scaffolding and falsework** using computer vision. The team (Evan, Chloe, Saanvi, Shalini) has four technical pillars: occlusion handling, YOLO-based defect detection, macroscopic structural analysis (potentially VLM-powered), and native iOS/LiDAR system integration. See [Scaffluent_meeting_notes.md](#) for full project details.

The tasks in this session — background removal, YOLO26 detection, Gemini 3 Flash bounding boxes, structured output, and a comparative report — map directly to Scaffluent's technical stack and decision points.

Demonstrated Hard Skills

1. Computer Vision Pipeline Engineering

The session walked through three distinct CV techniques that mirror Scaffluent's architecture:

- **Background removal** ([remove_bg.py](#)): Implemented flood-fill from image edges + connected component analysis to isolate the subject from a complex background. This is directly analogous to **Evan's occlusion handling role** — removing safety netting to reveal underlying structure. The technique progressed through iterations: color thresholding → flood-fill → connected component filtering, demonstrating the kind of levelled progress (L1 → L2 → L3) the meeting notes prescribe.
- **YOLO26 object detection** ([run_yolo_bus.py](#)): Set up and ran the latest YOLO26 model locally on CPU. This is the exact technology stack for **Chloe's microscopic defect detection layer** — flagging missing braces, loose fasteners, etc. with bounding boxes. The session revealed YOLO's strength (fast, precise on photographic scenes) and its limitation (zero detections on out-of-distribution input like pixel art), which is critical knowledge for a team that will encounter unusual visual conditions on Hong Kong construction sites with netting.
- **Gemini 3 Flash bounding boxes** ([gemini_bbox.py](#)): Integrated a VLM for zero-shot detection with structured JSON output. This validates **Saanvi's macroscopic layer approach** — using a VLM like Gemini for high-level structural understanding where YOLO's fixed 80-class vocabulary falls short (e.g., identifying scaffolding configurations, assessing overall structural layout).

2. Multi-Model API Orchestration & Structured Output

- Configured OpenRouter with the OpenAI SDK to call Gemini 3 Flash ([gemini_structured.py](#)), including **enforced JSON schema** (`response_format` with `json_schema` and `strict: True`) — not just “ask nicely in the prompt” but API-level guarantee of output structure. This is essential for **Shalini’s system integration role**: the iOS app must parse detection results reliably from both YOLO (tensor format) and VLM (JSON) pipelines.
 - Handled the full image-to-insight pipeline: base64 encoding → multimodal API call → structured JSON parsing → PIL-based visualization. This end-to-end orchestration is what Scaffluent’s production system will need when stitching together occlusion removal → YOLO detection → VLM analysis → user-facing display.
-

Demonstrated Soft Skills

1. Technical Communication & Comparative Analysis

The [YOLO vs Gemini report](#) demonstrates the ability to synthesize research from 5 parallel sub-agents into a single, decision-oriented document with:

- Executive summary table for quick scanning
- Metric-by-metric deep dives with hard numbers
- A “When to Use Which” decision guide
- A “use both together” hybrid recommendation

This directly serves Scaffluent’s decision-making: Chloe (YOLO) and Saanvi (VLM) need to understand each tool’s tradeoffs. The report provides the basis for the team’s technical architecture decisions, particularly around when YOLO handles real-time detection and when Gemini provides semantic understanding — exactly the layered approach the meeting notes describe (microscopic vs macroscopic).

2. Adaptive Problem-Solving Under Constraints

Multiple obstacles arose during the session, each resolved without user intervention:

Problem	Resolution
<code>python not found</code>	Switched to <code>python3</code>
Externally-managed Python environment (PEP 668)	Created a <code>.venv</code> virtual environment
Star artifacts remaining after background removal	Pivoted from dilation-based approach to connected component analysis — kept only the largest component
<code>pip install ultralytics</code> appeared stuck	Diagnosed as slow PyTorch download (~240 KB/s for 80MB), waited for completion

Problem	Resolution
Need for latest YOLO version	Dispatched web-search sub-agent, discovered YOLO26 (January 2026) rather than defaulting to outdated YOLOv8

This mirrors the meeting notes' emphasis on the **30-minute rule** ("if stuck, try something else or ask for help") and **partial progress = good progress**. Each setback was treated as a solvable constraint, not a blocker.

Further Hard Skills

1. Occlusion Handling & Image Inpainting Research

Evan's core role — making safety nets transparent — is a research-heavy computer vision problem. Relevant techniques I can help implement and evaluate include:

- **Frequency-domain filtering:** Safety nets have regular, periodic patterns detectable via FFT. Notch filtering in the frequency domain can suppress net patterns while preserving structural detail.
- **GAN-based inpainting:** Models like LaMa (Large Mask Inpainting) or Stable Diffusion inpainting can fill in regions occluded by nets, given a mask of net locations.
- **Semantic segmentation of nets:** Train a lightweight model (or use a VLM) to generate pixel-level masks of netting, then inpaint those regions.
- **Multi-frame fusion:** If the camera moves (chest-mounted or drone), parallax can reveal structure behind partial occlusion across consecutive frames.

These map to the L1 → L2 → L3 progression the meeting notes describe: L1 = detect net presence, L2 = mask the net, L3 = reconstruct underlying structure.

2. Structural Engineering Computation

Saanvi's macroscopic layer requires 3D math, physics, and load analysis. I can assist with:

- **Point cloud processing:** Parsing LiDAR data (from Shalini's iOS app or Tristan's Intel LiDAR), SLAM-based reconstruction, scaffold member segmentation from 3D point clouds.
 - **Load-bearing calculations:** Given scaffold geometry (member lengths, connection types, material properties of bamboo or steel), compute load capacity per member using structural engineering formulas (Euler buckling, member slenderness ratios).
 - **Regulatory compliance checking:** Cross-reference detected configurations against Hong Kong's Cap. 59I scaffolding safety regulations and code requirements.
-

Further Soft Skills

1. Project Management & Deliverable Preparation

The team has an **ISD Work Plan due February 13** and needs a Gantt chart, individual scope statements, and a 3-minute pitch video. I can:

- Refine the preliminary Gantt chart from the meeting notes into a formal project plan with dependencies, milestones, and critical path analysis.
- Draft individual scope statements for each team member based on the role assignments.
- Structure the 3-minute pitch video script to address the judges' Semester 1 feedback (“code was replicable” → emphasize novel technical contributions this semester).
- Set up GitHub Projects boards with issues mapped to the Gantt chart tasks.

2. Cross-Functional Technical Translation

Scaffluent's four roles span distinct technical domains (CV/image processing, object detection/ML, structural engineering/3D, iOS/embedded systems). The meeting notes highlight past communication gaps — Evan's messages being too long, uneven workload distribution, and teammates waiting for meetings instead of acting.

I can bridge these domains by: translating Saanvi's structural analysis requirements into API contracts Shalini can integrate, helping Chloe understand how Evan's occlusion output feeds into her YOLO pipeline, and producing concise technical summaries (the “3 key points” Saanvi requested from Evan) that keep all team members aligned without information overload.

TL;DR

This session demonstrated **CV pipeline engineering** (background removal, YOLO26, Gemini bounding boxes), **API orchestration with structured output** (OpenRouter + JSON schema enforcement), **technical report writing** (5-metric YOLO vs Gemini comparison), and **adaptive problem-solving** (venv workarounds, connected component analysis, latest-model research). These map directly to Scaffluent's technical stack: YOLO for Chloe's defect detection, VLMs for Saanvi's macroscopic analysis, image processing for Evan's occlusion handling, and structured pipelines for Shalini's integration layer. Beyond what was shown, further capabilities in **occlusion handling research** (FFT filtering, GAN inpainting), **structural engineering computation** (point clouds, load analysis), **project management** (Gantt charts, scope statements for the Feb 13 deadline), and **cross-functional translation** (bridging CV, ML, structural, and iOS domains) are directly applicable to the team's Semester 2 goals.