**StoreNow Roof Top Segmentation Project**

**Main Challenges**

* **Inaccurate Roof Border Detection on Full Images**  
  Initially, full satellite images fetched using Google Maps often contained **multiple houses**, confusing the border detection model. This led to inaccurate masks that included parts of neighbouring rooftops or background.
* **Target House Cropping Logic**  
  A key challenge was **identifying and isolating the central house.** We implemented logic to detect the **largest closed polygon** near the centre and re-ran segmentation on a cropped image with a 20px padding, which significantly improved prediction clarity.
* **Broken Roofline Predictions**  
  The roofline model (DeepLabV3+ with ResNet101) often gave **incomplete or noisy lines**, especially on complex or shadowed roofs. These masks lacked clear continuity, requiring **skeletonization and line filtering** techniques to extract usable straight roofline segments.
* **Over-Masking by SAM for Obstruction Detection**  
  The Segment Anything Model (SAM) tended to **over-segment** the roof, mistakenly classifying elevated or shaded roof surfaces as obstructions. We introduced a **dynamic colour variance thresholding** mechanism to limit this behaviour and reduce false positives.
* **Alignment Between Multiple Model Outputs**  
  Integrating different model outputs (roof border, rooflines, SAM) in a sequential pipeline posed problems due to **misaligned image sizes, mask formats, and coordinate systems.** This required strict standardization of image pre-processing and post-processing.
* **Inconsistent Roofline Partitioning**  
  Even with rooflines detected, **partitioning the full roof mask** into distinct sides based on these lines was difficult when lines were fragmented, not intersecting roof boundaries cleanly, or did not match roof angles precisely.
* **Generalization Issues Under Different Lighting & Roof Styles**  
  The models sometimes failed on **lightly coloured, dark, or shadowed roofs**, indicating limitations in generalization despite augmentations. Roofs with **complex geometry or dormers** added to prediction inconsistency.

**Roofline Specific Challenges**

**Issue 1: Noisy or Fragmented Rooflines**  
The roofline model frequently produced **broken lines**, **incomplete edges**, or **thick linear blobs** that did not align with roof borders, especially on tilted or shadowed roofs.

**What We Tried:**

* **Post-Processing with Skeletonization**: We applied morphological skeletonization to thin the roofline mask and extract 1-pixel wide line structures.
* **Contour Filtering**: We filtered out short or curved contours and retained only long, straight lines, using angle thresholds and segment length criteria.

**Issue 2: Roofline Not Intersecting Roof Borders Cleanly**  
In many cases, even when straight lines were predicted, they **didn’t connect cleanly** with the roof border polygon, making it hard to partition the roof into sides.

**What We Tried:**

* **Polygon Clipping & Intersection Heuristics**: We experimented with intersecting the border polygon with roofline segments to separate roof regions.
* **Epsilon-Based Simplification**: Applied cv2.approxPolyDP on roof polygons to simplify border edges before line intersection, but results were inconsistent for complex roofs.

**Issue 3: Roofline Leakage Outside Roof Region**  
Sometimes rooflines extended **beyond the actual rooftop boundary**, due to shadows, nearby lines, or prediction spill over.

**What We Tried:**

* **Mask Masking**: We multiplied roofline masks with the roof border mask, ensuring that any roofline outside the valid roof area was clipped out.
* **Convex Hull Experiment**: Tried convex hull wrapping of the roofline + border to enforce a boundary, but this over-smoothed angled roof and was discarded.

**Future Work and Potential Improvements**

**1. Expanding and Diversifying the Training Dataset**

The current training datasets for roof border detection (~75 images) and roofline segmentation (~100 images) are limited in both size and diversity. While augmentation (e.g., flips, rotations, lighting adjustments) helps simulate variation, it cannot fully substitute for real-world diversity.

**Recommendation:**

* Actively increase the dataset size using labelled data from multiple geographies(Germany), especially regions with varied roof materials, lighting, and architectural styles.
* You can try implementing curriculum learning, where the model is first trained on clean, ideal roof images, and then gradually exposed to complex scenes (shadows, occlusions, distortion).

**2. Handling Shadows and Tree Occlusions on Rooftops**

Google Maps imagery often includes heavy shadows from trees or building geometry, which significantly affects roof edge and roofline visibility. Even human annotators struggle under these conditions.

**Recommendation:**

* Train a shadow detection model (e.g., using deep learning techniques from shadow removal papers) to create shadow masks, which can be input alongside the RGB image to help the segmentation model ignore or de-emphasize shadowed areas.
* Alternatively, train the model in a multi-task setup, where one branch learns roof segmentation, and another learns to predict shadow masks or occlusions.
* Explore domain adaptation or contrastive learning methods to make the model invariant to lighting shifts and angles of view.

**3. Image Quality Enhancement Before Inference**

Google Maps satellite tiles are often soft, low-contrast, or blurred due to atmospheric or compression artifacts. This makes subtle edges like rooflines harder to detect.

Recommendation:  
Introduce a pre-processing stage after image retrieval and before segmentation, which applies:

* Contrast Limited Adaptive Histogram Equalization (CLAHE) for localized contrast enhancement.
* Super-resolution techniques (e.g., using SRCNN or ESRGAN) to artificially upscale and sharpen the image.
* Colour saturation boost and edge sharpening filters to highlight roof-texture features and improve model focus.

This can be a plug-in module:  
Raw Google Image → Enhanced Image → Border Model → Roofline Model → SAM Obstruction → Coordinate Extraction