Technical Execution Roadmap: High-Precision Feature Extraction from SVAMITVA Imagery

A Long-Term Strategy for Achieving 95% Accuracy by March

Our Team is Structured for Heterogeneous Specialisation

To conquer the diverse feature types in SVAMITVA imagery, we have organised our 8-member team into three distinct functional squads. This division of labour allows for deep focus on specific AI models and GIS workflows, which is critical for meeting the deadline and accuracy targets.

Squad	Members	Role & Focus	Core Stack
A: GIS & Data Ops	2 Members	Pre-processing (Coordinate-aware Tiling) & Post-processing (Geospatial Stitching, Vector Regularization).	`Rasterio`, `GDAL`, `GeoPandas`, `QGIS`.
B: Building Specialists	3 Members	Fine-tuning **Mask R-CNN** for Footprints and N-Class Roof Classification (RCC, Tile, Tin).	`Detectron2`, `mmdetection`, `PyTorch`.
C: Infrastructure Specialists	3 Members	Semantic Segmentation (**U-Net** / DeepLab) for Roads/Water; Object Detection (**YOLOv8**) for Assets (Wells/Tanks).	`segmentation_models.pytorch`, `ultralytics`.

Specialisation is the Key to Conquering Complexity

A single person cannot master three different AI model types and GIS software simultaneously within the deadline. By dropping the 3D problem, we deployed all 8 **members** to divide the complex task into manageable, specialised roles.

Squad A (GIS)

They are the **map handlers.** They ensure images enter the AI pipeline without losing crucial geographical data and that the final results are delivered as perfect, legally compliant maps (Shapefiles).

Squad B (Buildings)

This is the **hardest classification** task. AI models easily confuse grey concrete (RCC) with grey metal (Tin). This squad focuses 100% on solving that **specific classification** puzzle.

Squad C (Infrastructure)

Roads are thin lines, buildings are solid shapes, and wells are tiny dots. These features require different neural network architectures (Segmentation vs. Detection). This squad handles all non-building features.

Phase 1 Blueprint: Pipeline Prototyping on Open Datasets (Now – 27 Jan)

To complete **the development** of End-to-End (E2E) pipelines using proxy data, ensuring we are ready for the **SVAMITVA** imagery upon arrival.

Squad A (GIS & Data Ops)

Develop **`tile.py`** utility for coordinate-aware image tiling.

Method: Sliding Window with 20% overlap stride on a 400px dimension.

Source: Proxy orthophotos.

Squad B (Buildings)

Train a baseline Mask R-CNN model (e.g., with a ResNet-50 FPN backbone).

Objective: Validate the instance segmentation output pipeline.

Source: SpaceNet 7 or CrowdAI datasets.

Squad C (Infrastructure)

Train a **U-Net** model on the DeepGlobe dataset for road extraction.

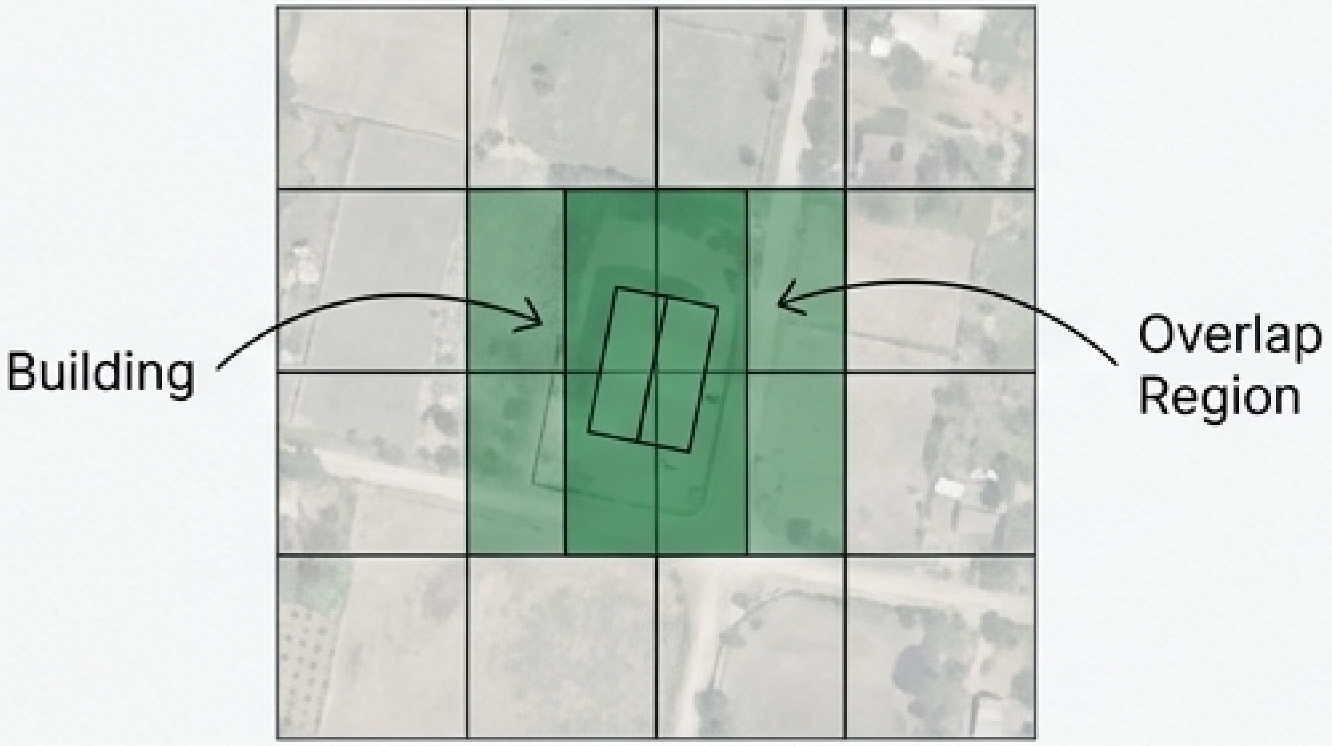
Set up the **YOLOv8** workflow using its CLI on a standard detection dataset to establish the workflow.

Phase 1 Strategy: Building the Factory Before the Raw Materials Arrive

We do not have the SVAMITVA data yet. Waiting until 27 January to begin coding would guarantee failure. We are using open-source ‘practice’ datasets that resemble drone maps to build and debug our entire workflow now.

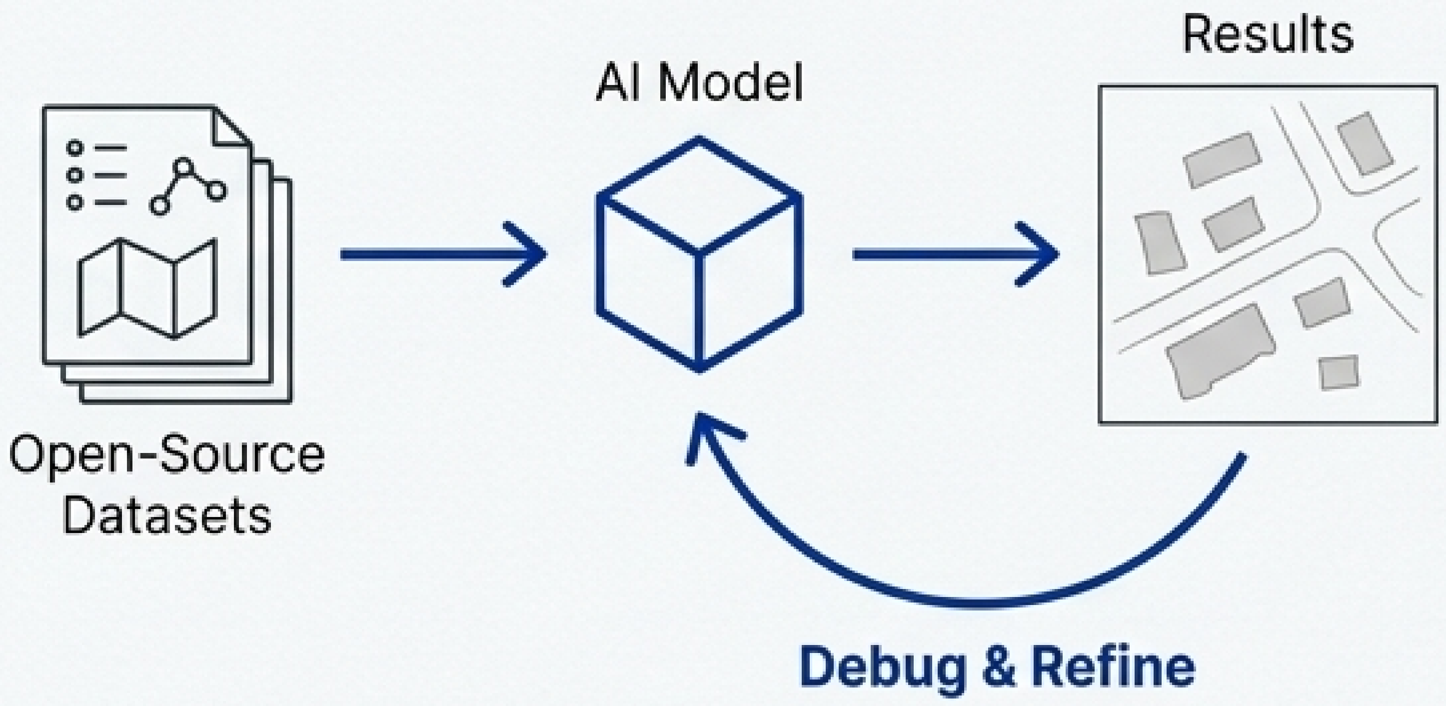
Explaining the ‘Overlap Tiling’ Rule

Drone maps are too large for AI models to process at once. Squad A is writing a tool to cut them into small, overlapping squares. The overlap is crucial: if we cut straight through a house, the AI will fail to recognise the “half-house”. Overlapping ensures every feature is seen “whole” in at least one square.



Explaining ‘Model Practice’

Squads B and C are setting up their respective AI models now. This allows us to resolve installation errors, debug code, and master the tools *before* the project clock officially starts. We will be ready on Day 1.



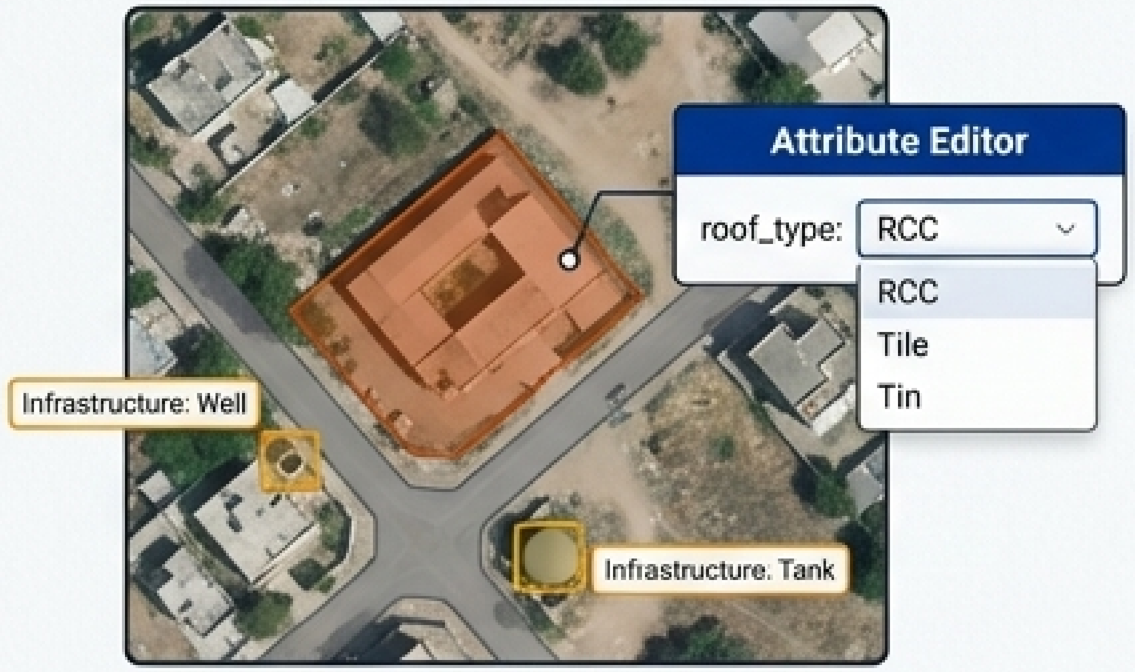
Phase 2 Blueprint: Data Audit and Ground Truth Generation

Step 1: Data Audit (Days 1–5)

- **GSD Validation:** Validate Ground Sampling Distance to understand image resolution.
- **Quality Analysis:** Conduct visual contrast analysis to determine the necessity of pre-processing.
- **Potential Enhancement:** Apply **CLAHE** (Contrast Limited Adaptive Histogram Equalization) if needed to distinguish low-contrast features (e.g., a well vs. a shadow).

Step 2: Annotation Sprint (Days 6–15)

- **Tool: CVAT** for distributed annotation across all 8 team members.
- **Annotation Protocol:**
  - **Buildings:** Polygons + N-Class attribute (`roof_type`).
  - **Roads:** Polygons.
  - **Infrastructure (Wells, Tanks):** Bounding Boxes (for detection head).



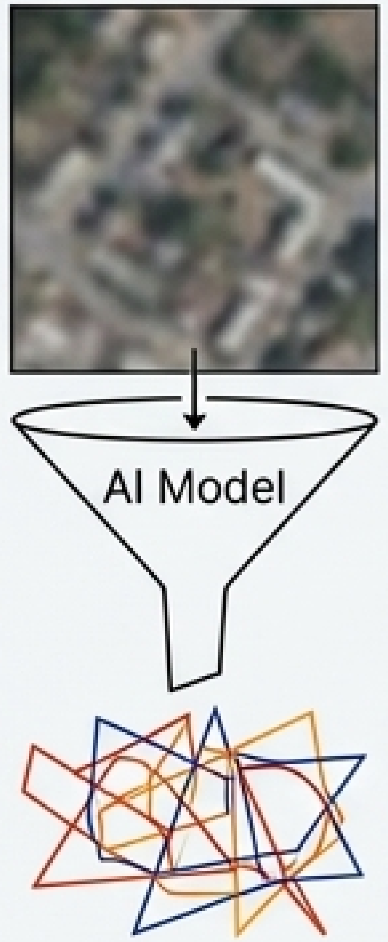
Phase 2 Strategy: Inspecting the Materials and Writing the Textbook

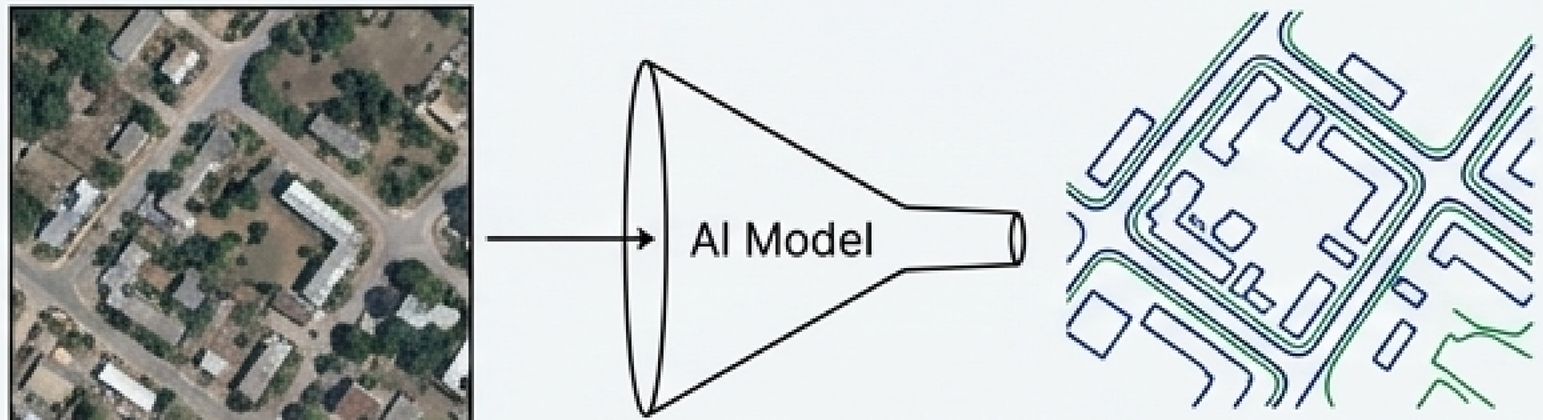
Garbage In, Garbage Out. An AI cannot learn from bad data.

The Audit Explained

First, we inspect the maps.
We calculate GSD (how many centimetres one pixel represents).

We check if images are too dark or blurry.
If a human analyst cannot distinguish a well from a shadow, the AI certainly cannot. We will mathematically improve image image quality first if necessary.

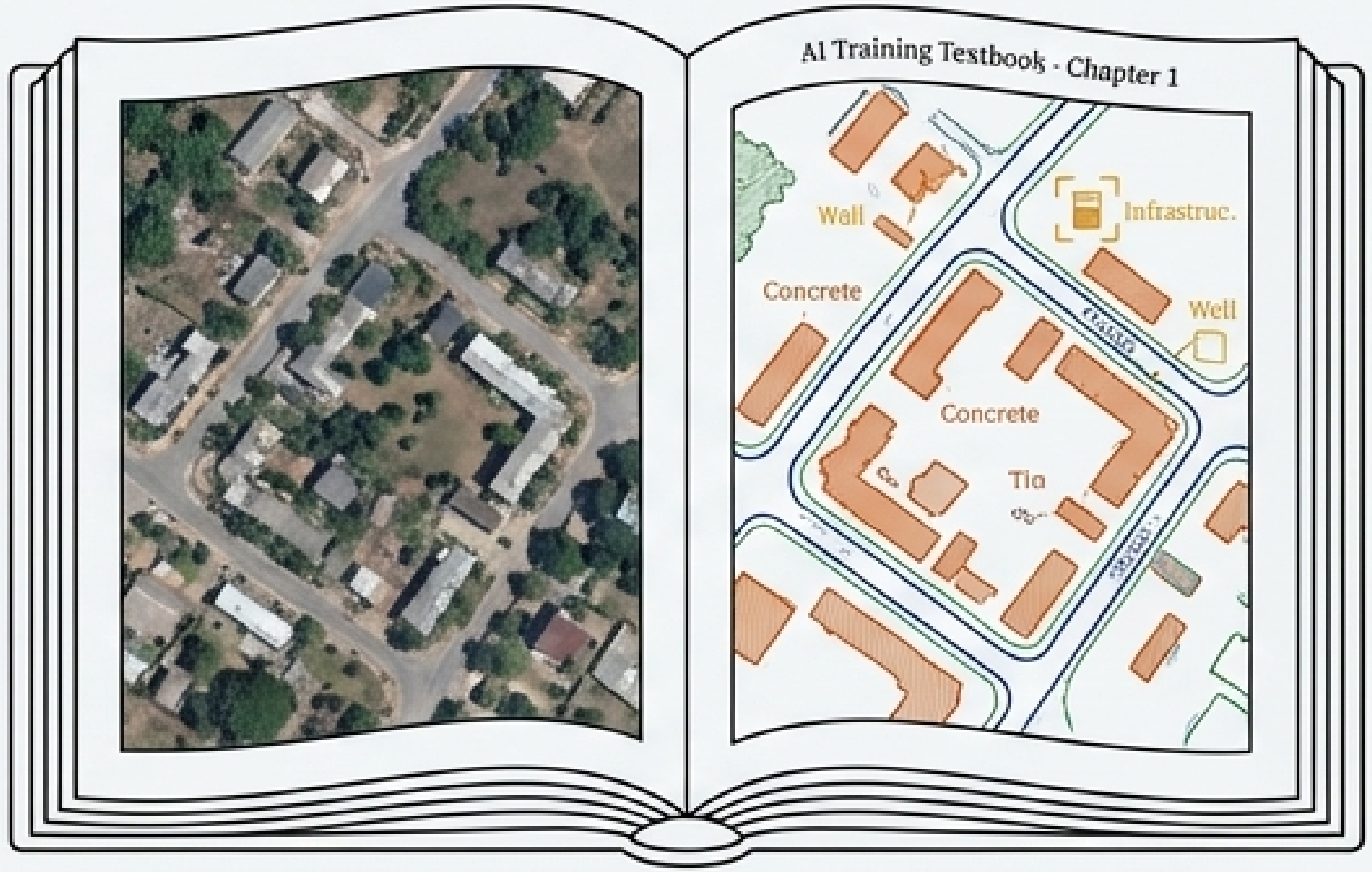




Perfect Output

The Annotation Explained

This is the single most important step to achieving 95% accuracy.
All 8 members will spend a week manually drawing the features.
We will draw perfect outlines around buildings and label them (Concrete, Tin, etc.).
We are essentially writing the high-quality “textbook” from which the AI will study.

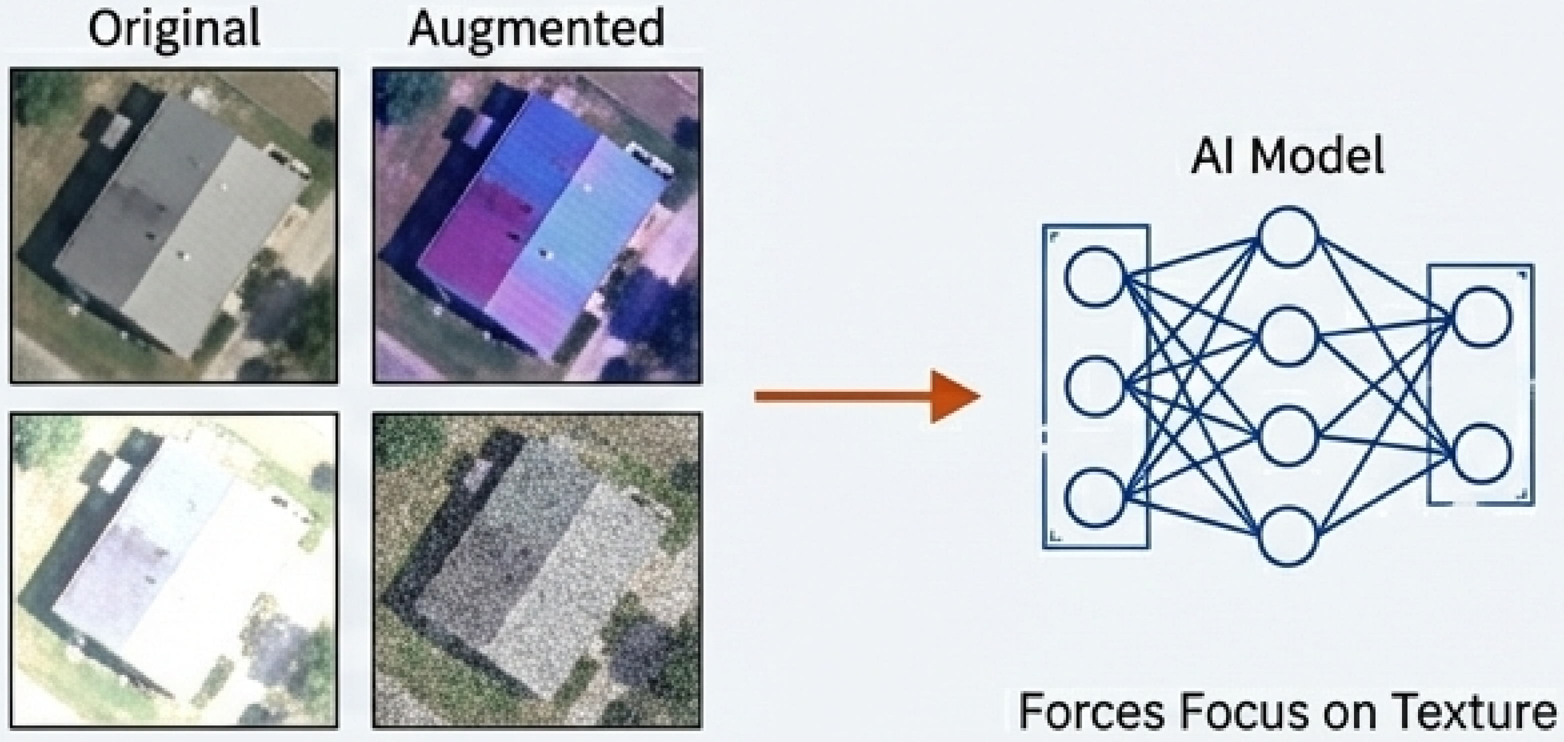


Technical Deep Dive: Iterative Training and Targeted Error Correction

Squad B (Buildings - Mask R-CNN)

Anticipated Problem: High intraclass variance and low interclass variance between RCC (concrete) and Tin roofs due to similar grey colouring, leading to classification confusion.

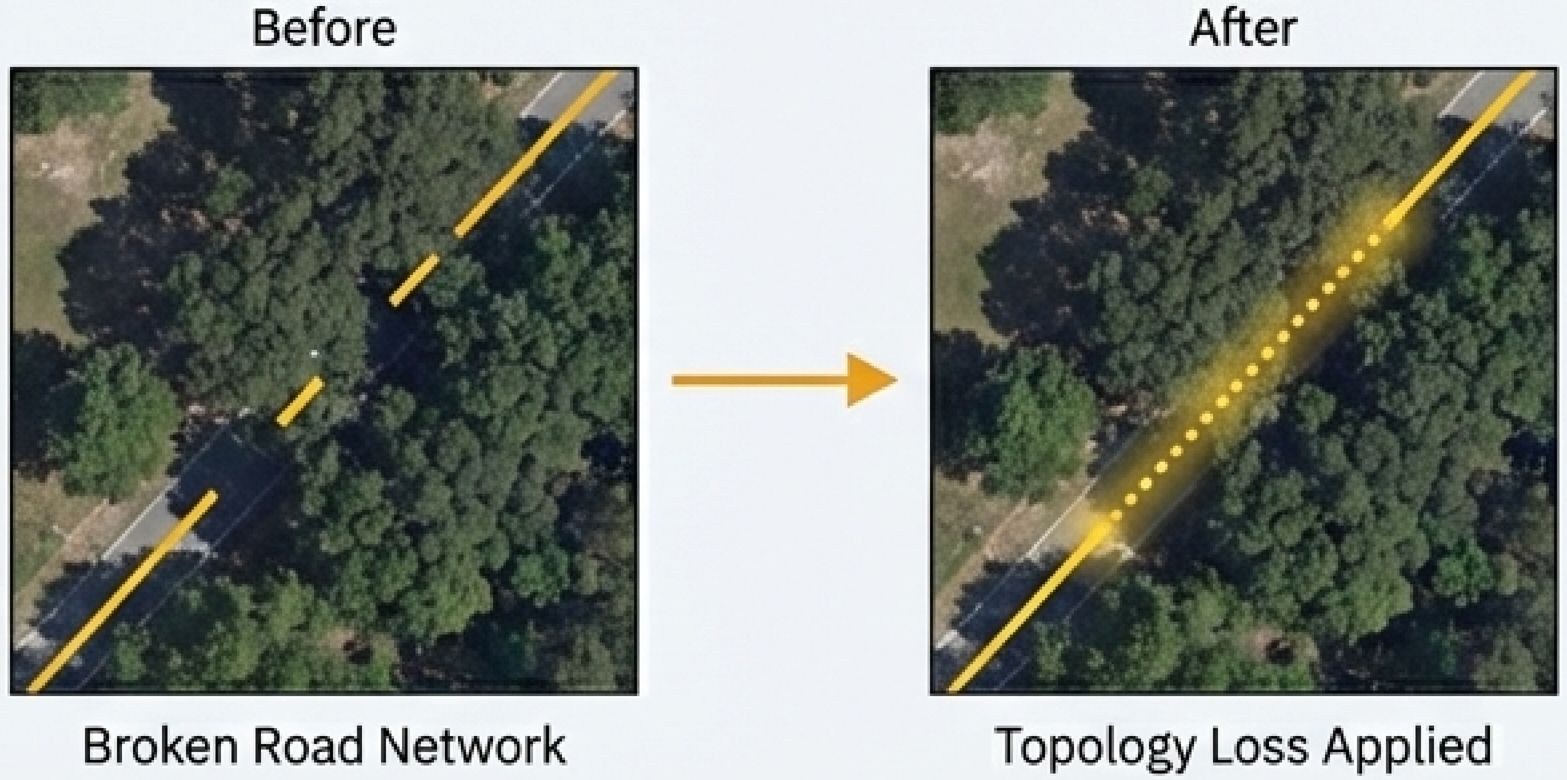
Proposed Solution: Apply aggressive Color Augmentation (Jitter) and Gaussian Noise using the `albumentations' library. This forces the model to focus on texture and geometry rather than relying on colour.



Squad C (Roads - U-Net)

Anticipated Problem: Discontinuity in the extracted road network caused by canopy obstruction (e.g., trees over roads).

Proposed Solution: Implement a Topology Loss function. This boundary-aware loss penalises the model for breaking continuous linear structures, encouraging it to ‘guess’ the road's path through obstructions.



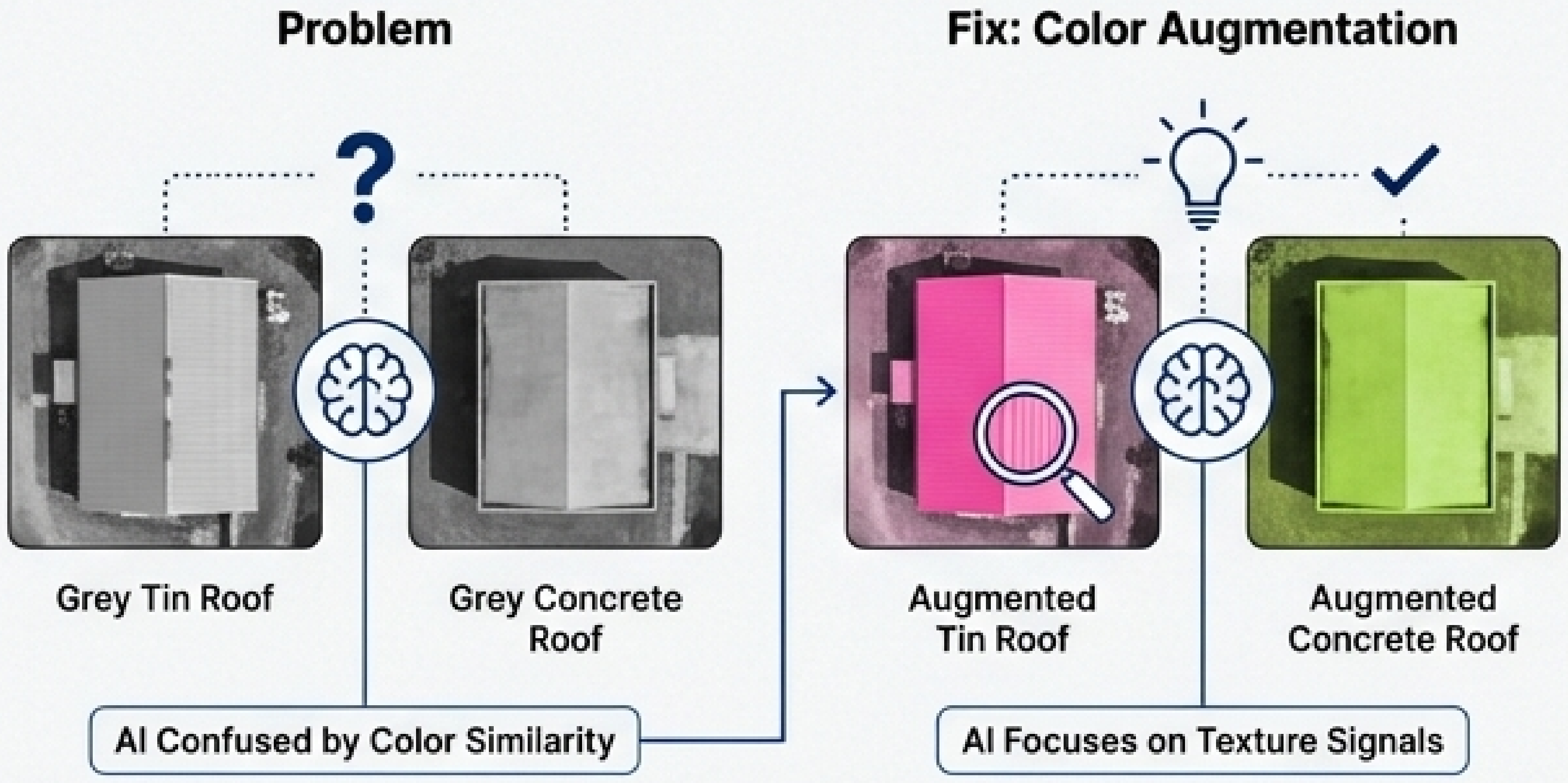
Our Strategy for Tuning: Analyse the Mistake, Then Fix the Teaching Method

The AI is now studying its textbook. When it takes a test (predicts features on unseen data), it will make specific, predictable mistakes. Our job is to analyse those mistakes and adjust our teaching method accordingly.

Fixing Building Misclassification (Squad B)

The Problem: The AI is confused because grey metal (Tin) looks just like grey concrete (RCC).

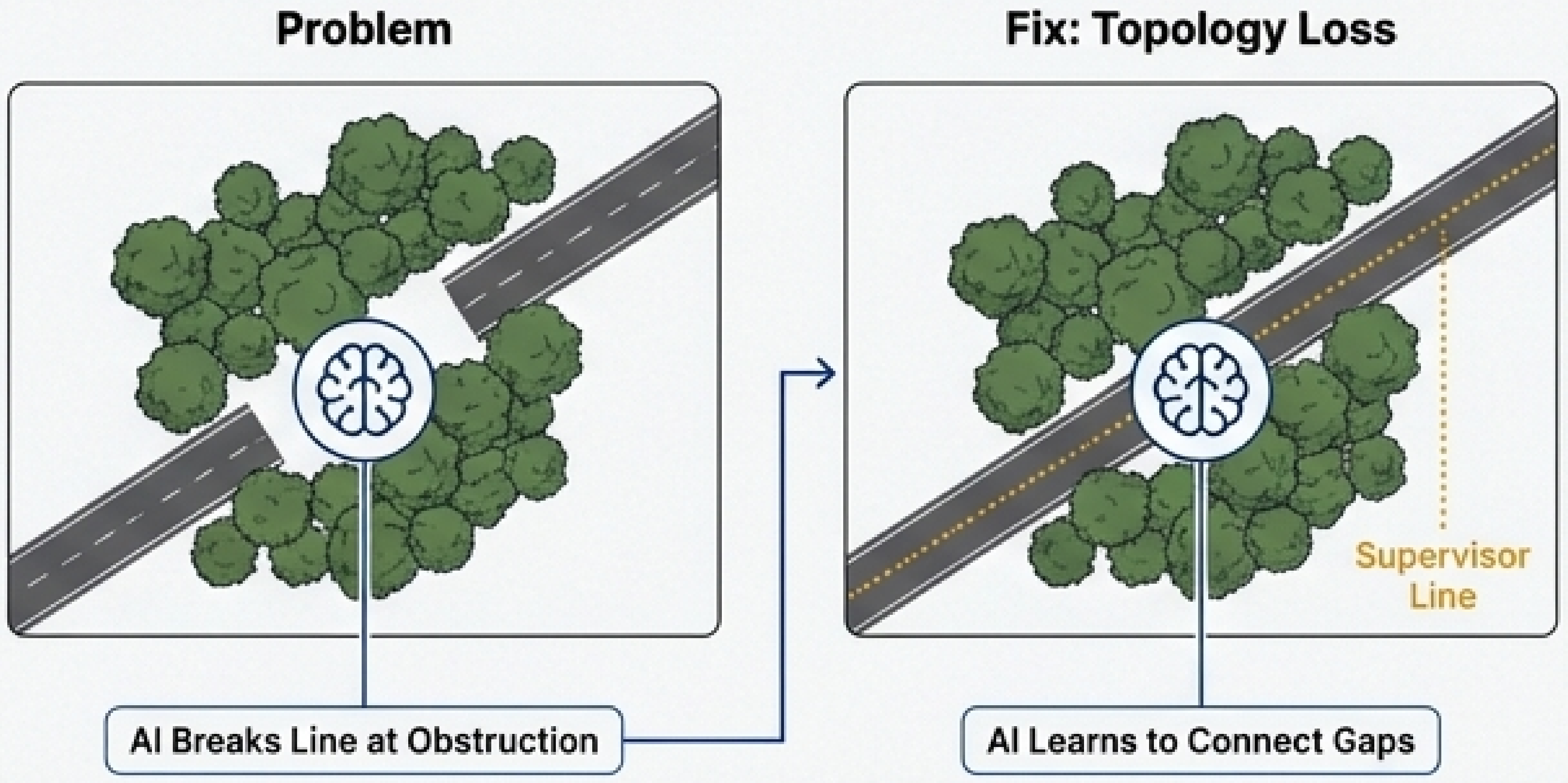
The Fix: We use a technique called Color Augmentation. By randomly and dramatically changing the colours of the training images, we force the AI to stop paying attention to colour and learn the more reliable signals of *texture* (smooth concrete vs. corrugated metal).



Fixing Broken Roads (Squad C)

The Problem: Trees hide parts of the road, causing the AI to stop drawing the line where it sees a tree.

The Fix: We use specialised mathematics (Topology Loss) that acts like a supervisor, penalising the AI whenever it breaks a continuous line. This forces the model to learn to connect road segments through gaps.



The 95% Push: Advanced Ensembling and Geometric Regularization

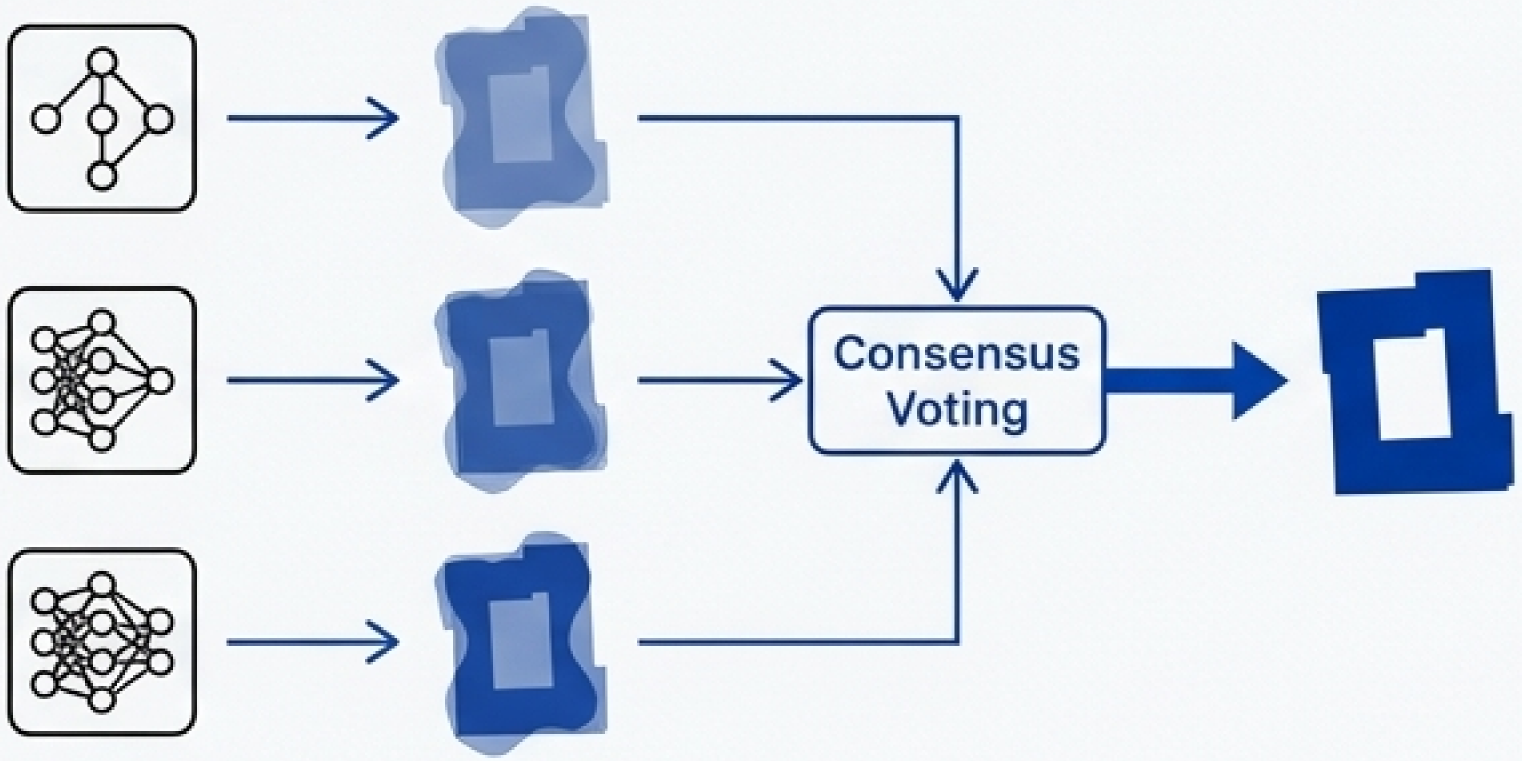
Strategy 1: Model Ensembling (Stacking)

Rationale: To increase robust generalisation and reduce variance from any single model's prediction.

Process:

1. Train Model 1 (Backbone: ResNet-50)
2. Train Model 2 (Backbone: ResNet-101)
3. Train Model 3 (Backbone: Swin Transformer)

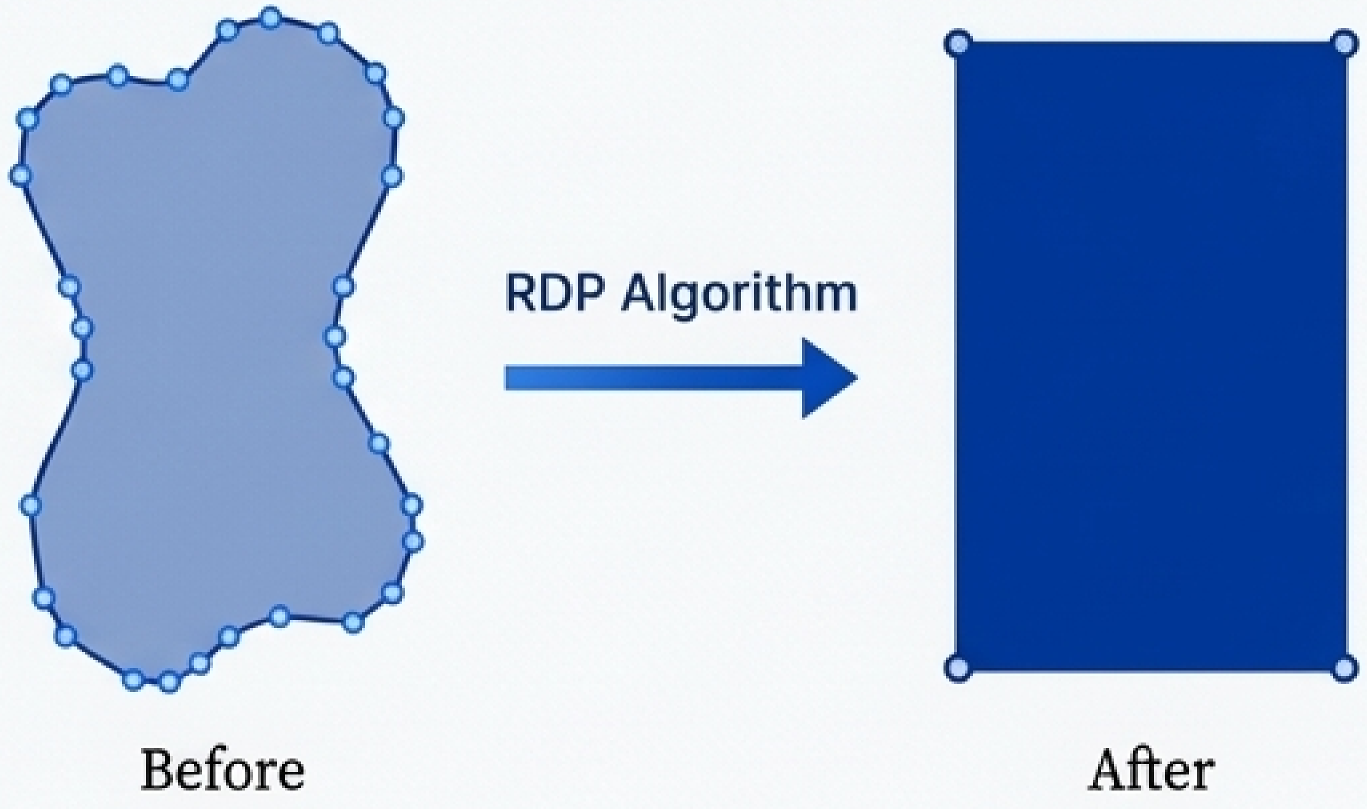
Method: Use Consensus Voting by averaging the predictions from all three models.



Strategy 2: Geometry Refinement

Problem: AI models often output 'blobby' or noisy segmentation masks for buildings.

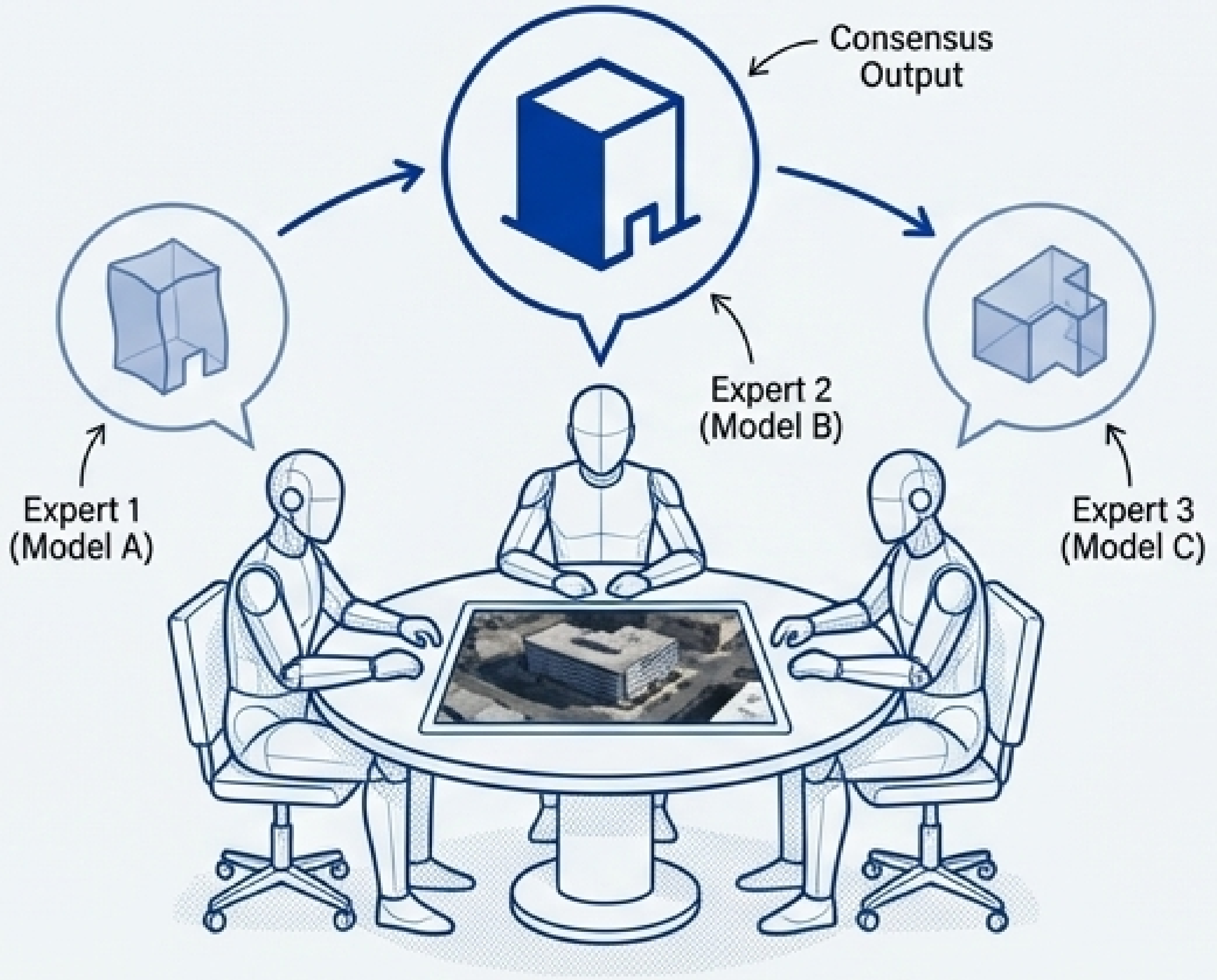
Solution: Apply the **Ramer–Douglas–Peucker (RDP)** algorithm post-prediction for contour simplification and orthogonalization, effectively 'squaring off' the corners of building footprints.



The Strategy for Perfection: A Panel of Experts and a Final Polish

1. Ensembling: The Panel of Experts

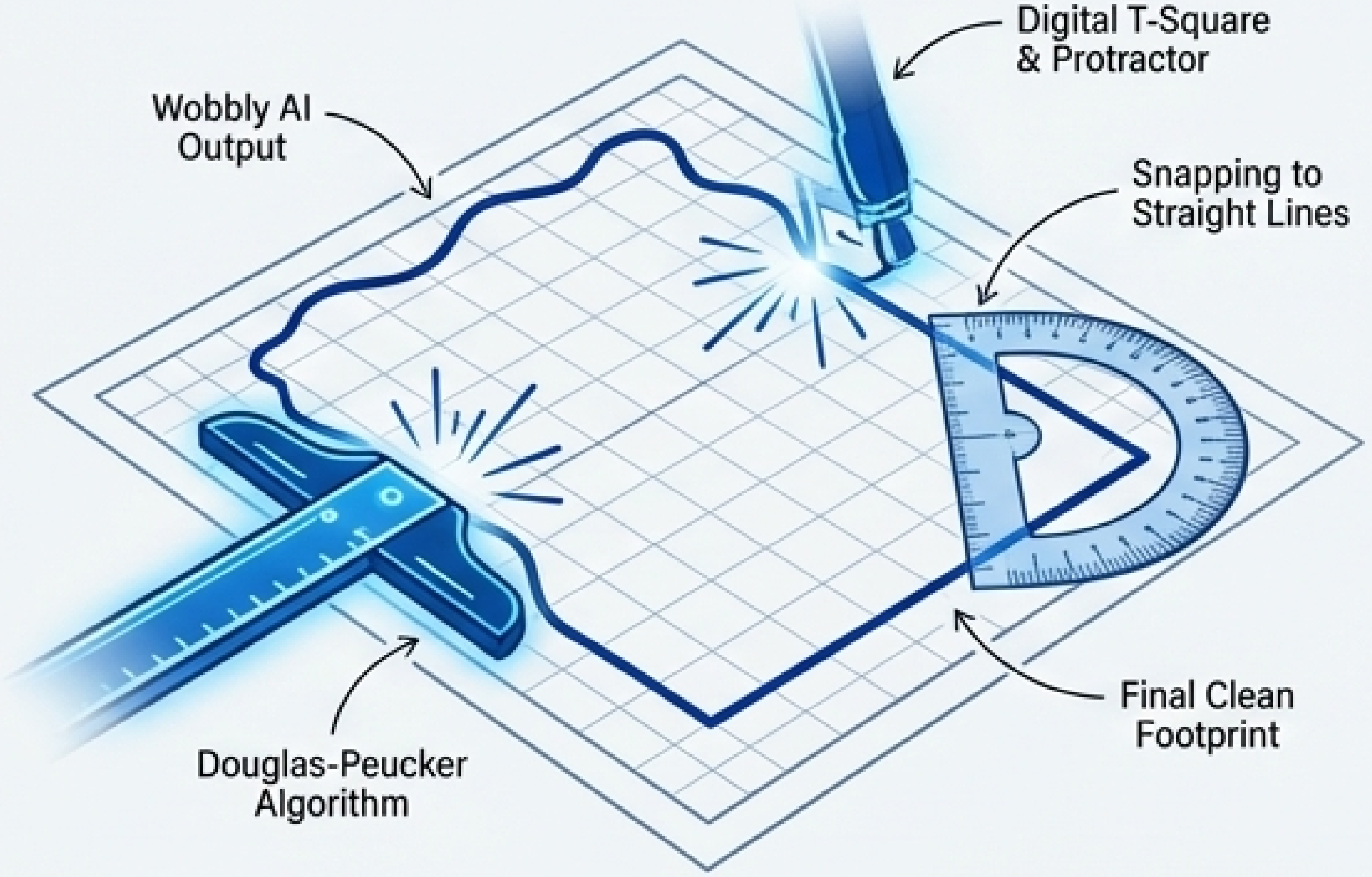
We do not rely on a single AI model. Instead, we train three different "experts" (models with distinct internal architectures). We show all three the same image and take the average of their guesses. This method consistently improves accuracy because one expert's weakness is often another's strength.



2. Regularization: Digital Squaring-Off

The Problem: AI often draws building footprints with wobbly, rounded outlines that are not geographically accurate.

The Fix: We apply a geometric formula (Douglas-Peucker) to the AI's output. This algorithm takes the wavy outline and automatically snaps it into clean, straight lines with sharp, neat corners, ensuring the buildings look like proper rectangles on the final map.



NotebookLM

Summary Execution Roadmap: Now to March

Now – 27 Jan

Pipeline Development & Prototyping

Build the end-to-end workflow using practice datasets.

27 Jan – 10 Feb

Audit & Annotation Drive

Analyse the official SVAMITVA data and create the high-quality ‘ground truth’ labels.

11 Feb – March

Training, Tuning, and Finalisation

Iterative model training, error analysis, ensembling for accuracy, and final map generation.

Achieve 95%+ Precision on Final Geospatial Outputs.

Visualising the Outcome: From Raw Imagery to Precision Geospatial Data

Input

Output

This transformation is the result of our specialised team structure, meticulous data preparation, iterative model tuning, and advanced refinement techniques.