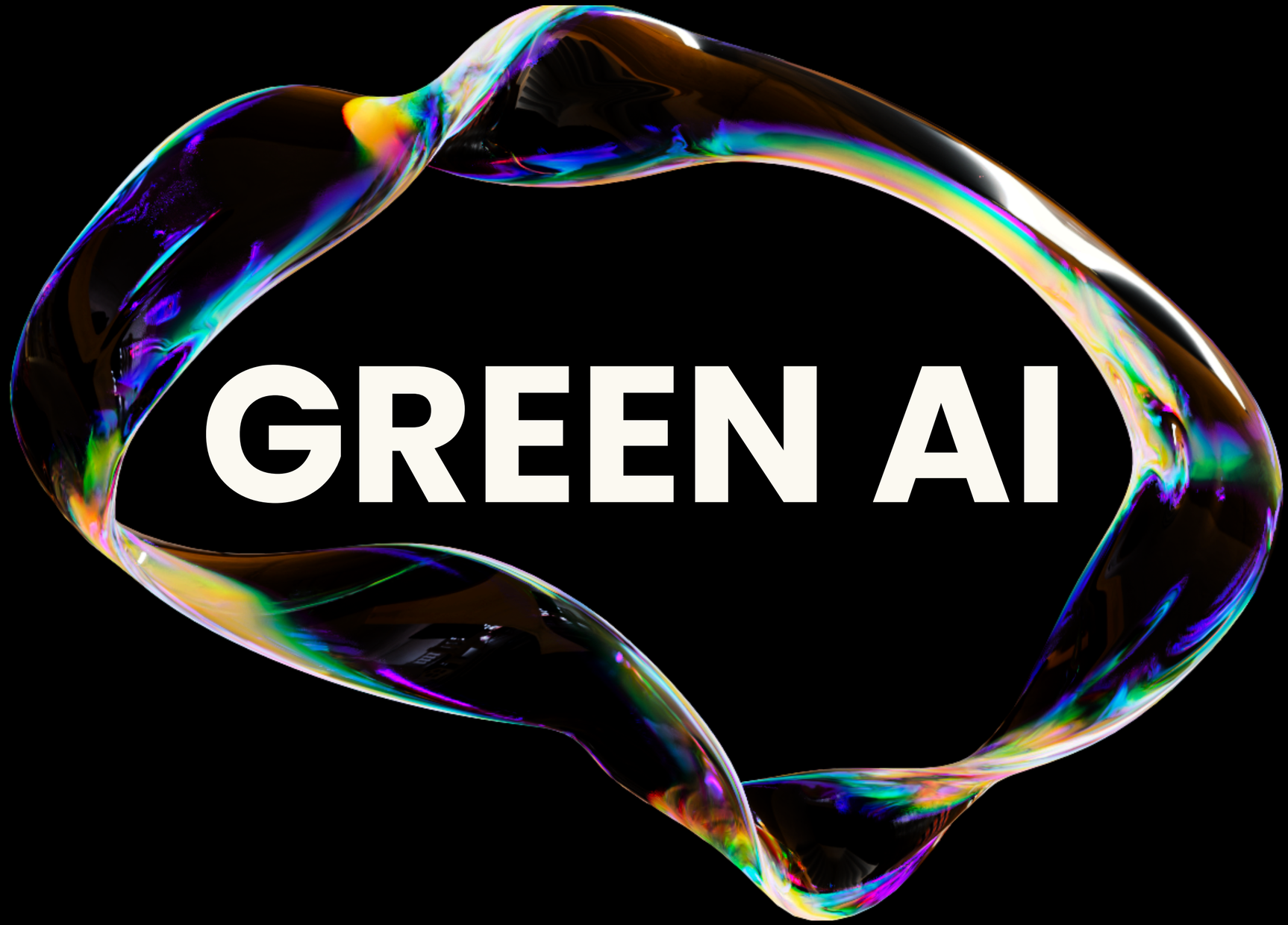


RIFFI Naïl SAGROUN Léa PAHLAWAN Léonard SAINT JEAN François

AI SATELLITE DETECTION OF ILLEGAL MINING

The Problem

Illegal mining is accelerating across the Amazon,
driving deforestation, and encroaching into protected zones.

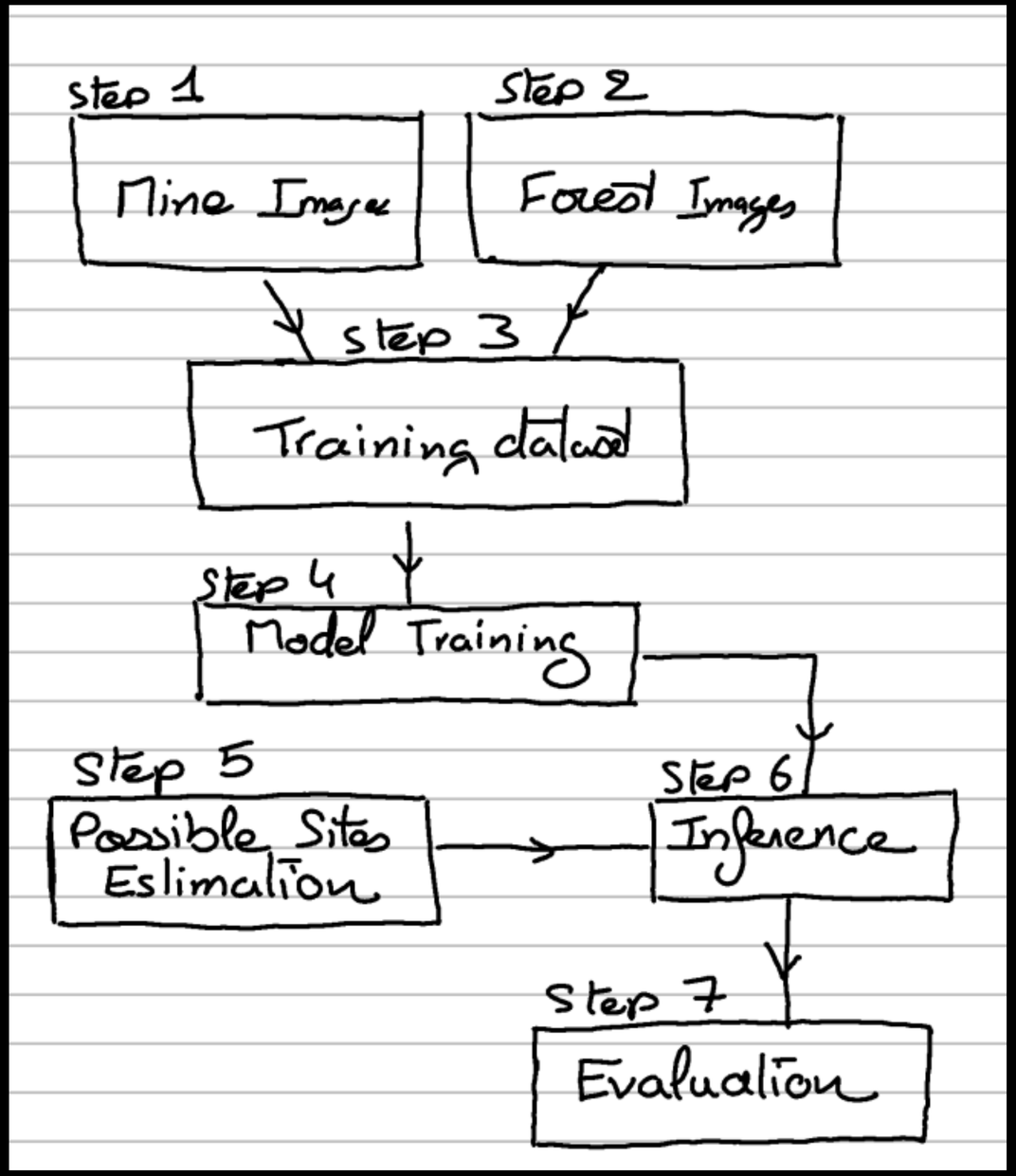
Impact

- Rapid canopy and forest loss
- Encroachment into protected zones
- Pollution of waterways

AI Satellite Detection

- Early detection of illegal mining activity
- Tracking canopy and land degradation
- Scalable, automated surveillance of protected areas

Illegal Mining Expansion → Illicit Revenues → Reinvestment → Accelerated Degradation →



Step 1

Creating the dataset for the positive class

Step 2

Creating the dataset for the negative class

Step 3

Pre-processing of the generated images

Step 4

Fine tuning the model resnet34

Step 5

Using segformer to determine suspicious mining zones

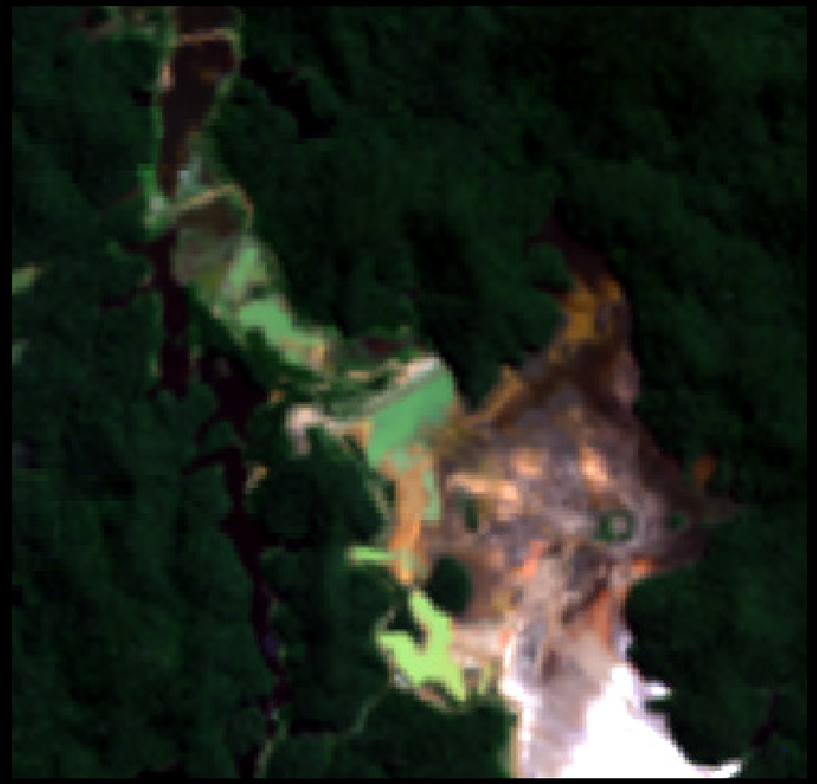
Step 6

Transmit the candidates from the segformer to the model

Step 7

Predicting the output for these candidates

STEP 1 – POSITIVE CLASS



```
35.099643538899585,30.8579059054111
19.137595445428687,51.23954076430057
20.22026405094736,44.48164661457545
20.335805632054797,44.42736954023631
21.245062854087898,44.72797344416286
21.515819204965407,41.048632813715074
21.64826312343959,40.604705608067015
```

Problem:

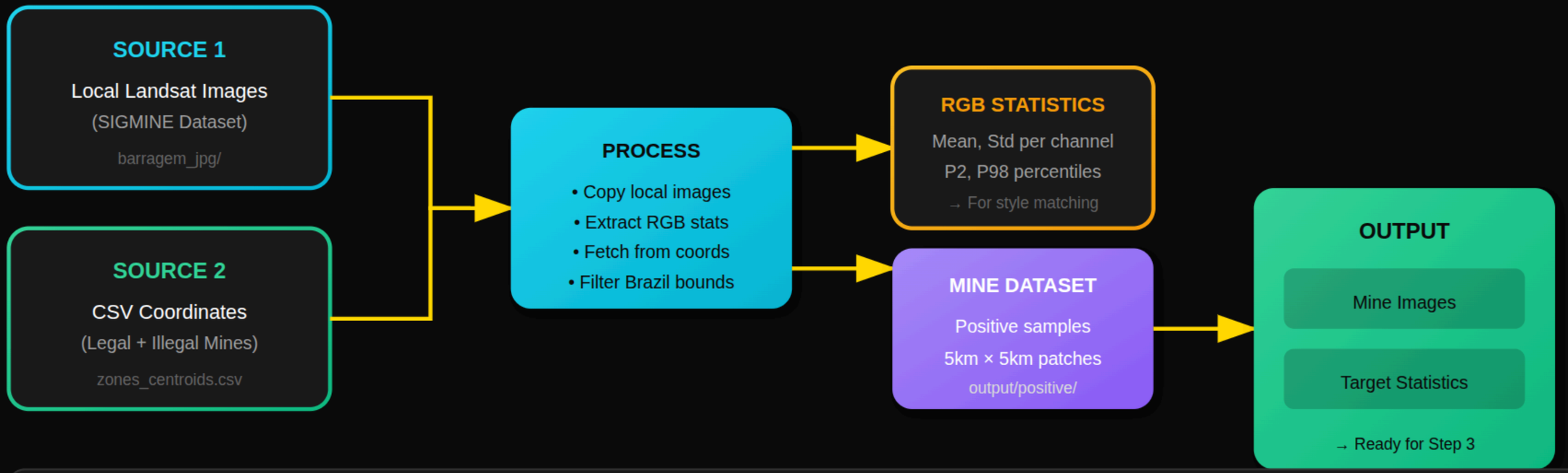
There is no extensive free to use data concerning legal mining sites in brazil

Solution:

Dataset of legal mine dams

Csv of coordinates of known mines in the world transformed to images using planetary computer library

STEP 1: POSITIVE CLASS DATA COLLECTION



KEY FUNCTIONS:

`extract_statistics()` → Reservoir sampling for memory efficiency

`build_positive_class()` → Copies local + fetches from CSV coords

Brazil Bounds: lon(-75,-35) lat(-35,5)

STEP 2: NEGATIVE CLASS DATA COLLECTION

MOTIVATION

Prevent model from always predicting "mine"

PROTECTED AREAS

Amazon rainforest
Conservation zones
N = 800 samples

→

RANDOM SAMPLING

Pick N random coords
Filter Brazil bounds
n_mines ≈ n_forest

→

SATELLITE FETCH

Coords → Images
5km × 5km tiles
Cloud filter <20%
2023-2025 imagery

→

STYLE MATCHING

Match to mine stats
P2/P98 percentile
normalization
→ Homogeneous set

→

OUTPUT

Forest Images

CHALLENGES

- Exclude town/urban areas (mislabeled data)
- Possibility of illegal mines in "forest" samples
- Tiling issues → Need to stitch tiles together

KEY FUNCTIONS:

`generate_forest_samples(n=800)`	→ Random protected area sampling
`match_image_to_stats(img, target)`	→ Distribution matching
`SatelliteFetcher.fetch_image()`	→ API call + normalization

STEP 3: BUILDING THE DATASET

PROBLEMS

- Cannot merge datasets directly
- Not enough training data
- Style/distribution mismatch

SOLUTION 1: DISTRIBUTION MATCHING

Forest Img → P2/P98 → Matched

Match RGB percentiles to mine stats

SOLUTION 2: DATA AUGMENTATION

5× augmented copies per image

Satellite-specific transforms

SatelliteAugmentation class

SATELLITE AUGMENTATION TRANSFORMS

90° Rotations Cardinal angles only	**Flips** Horizontal + Vertical	**Brightness** ±15% (sun angle sim)
Contrast ±15% variation	**Atmospheric Haze** 30% probability	**Sensor Noise** Gaussian σ=0.02

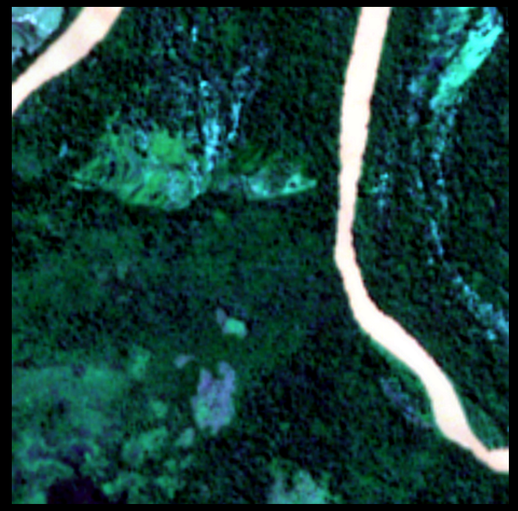
FINAL DATASET

positive/ Original mines	negative/ Matched forest
pos_augmented/ 5× augmented	neg_augmented/ 5× augmented

Distribution Matching explanation is next slide

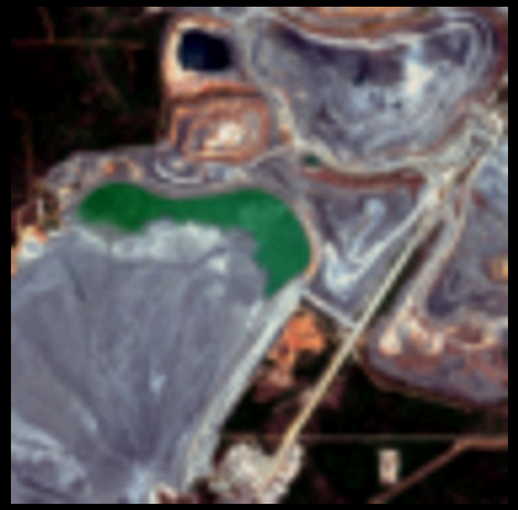
PRE-PROCESSING DATA

Downloaded Images



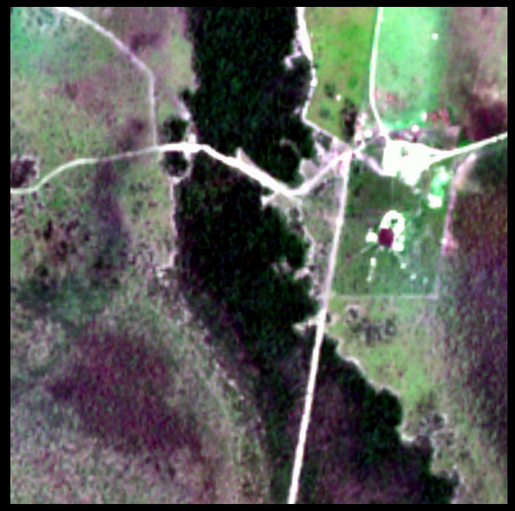
Original Fetched data

Transformation 1



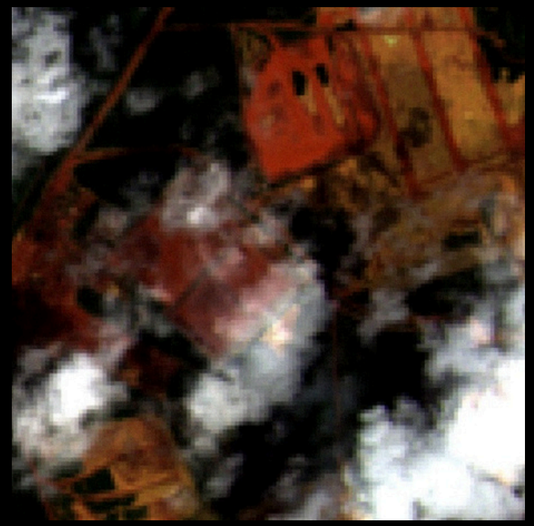
Final Applied transformation:

- Extracted statistics from 2/98% dam data
- Normalized stats to match local data
- Chose a distance of 5*5 km
- Downscaled to 95 pixels back to 512

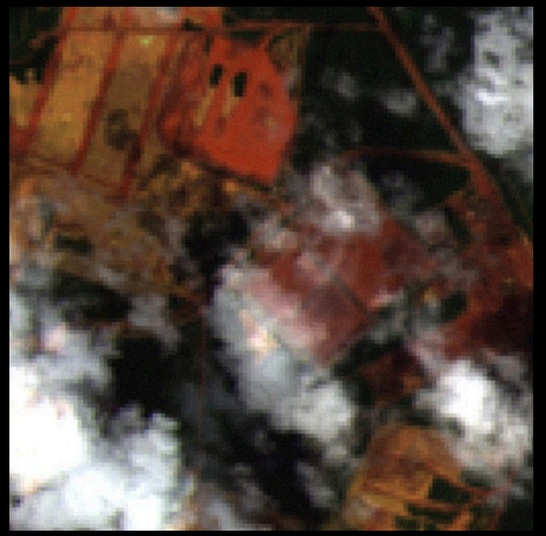


DATA AUGMENTATION

Brightness + Rotation



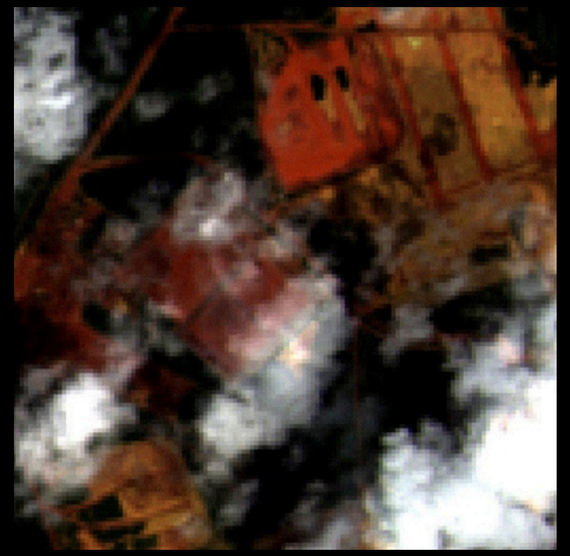
Original



Flip

← Higher contrast

Higher Brightness →



STEP 4: MODEL TRAINING

MODEL ARCHITECTURE

ResNet-34
Pretrained ImageNet

+ Custom Classification Head
+ Dropout (0.5)
+ Single Output (Binary)

PREPROCESSING PIPELINE

Resize 224×224 → **ToTensor** [0,1] → **Norm** ImageNet

Mean: [0.485, 0.456, 0.406]
Std: [0.229, 0.224, 0.225]

HYPERPARAMETERS

Batch Size:	**16**	Learning Rate:	**1e-4**
Max Epochs:	**300**	Val Split:	**15%**
Early Stop:	**10 epochs**	Loss:	**BCE**
Optimizer:	**Adam**	Scheduler:	**RedPlateau**

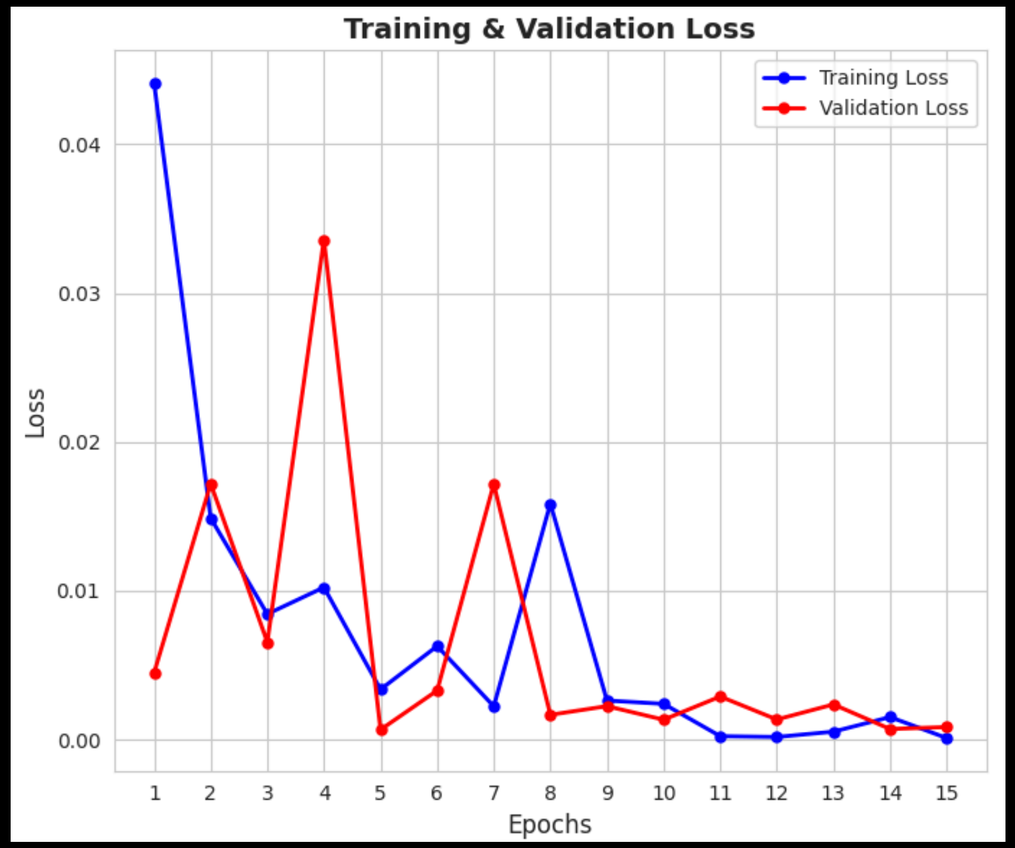
TRAINING PIPELINE

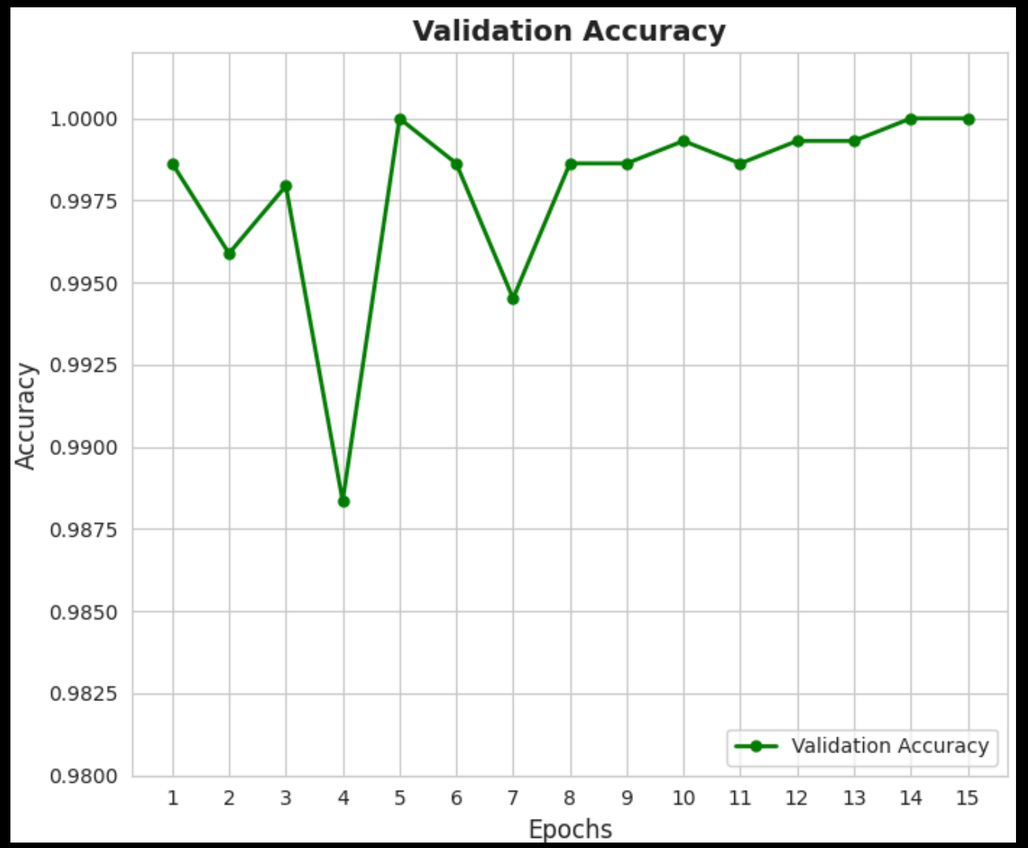
Dataset MiningDataset → **Train/Val Split** 85% / 15% → **Forward Pass** ResNet + BCE → **Backward** Adam optimize → **Early Stopping** patience=10 → **best_model.pth** Saved checkpoint

OUTPUT FILES:

- **best_model.pth** — Best val loss
- **final_model.pth** — Last epoch
- **training_history.json** — Loss/acc curves
- **config.json** — Hyperparams

RESULTS





- Rapid Convergence: By Epoch 5, the validation accuracy hits 100% (1.0) and the loss drops significantly.
- Stability: Spike in validation loss at Epoch 4, but the model recovers immediately. By the final epochs (14-15), the training loss is nearly zero, and validation accuracy is a stable 100%.
- High Performance: ResNet-34 is very effective for this specific dataset, which means the dataset is relatively easy for such an architecture.

STEP 5: TWO-STAGE DETECTION

STAGE 1: SEGMENTATION

STAGE 2: CLASSIFICATION

SUSPICIOUS CLASSES (LoveDA)

1	2	3	4
Building	Road	Water	Barren

Mines often classified as industrial/disturbed

→ Classifier filters false positives

WHY TWO STAGES?

- Process large areas efficiently
- SegFormer: fast candidate detection
- Classifier: high precision confirmation
- Reduces false positive rate

OUTPUT VISUALIZATIONS

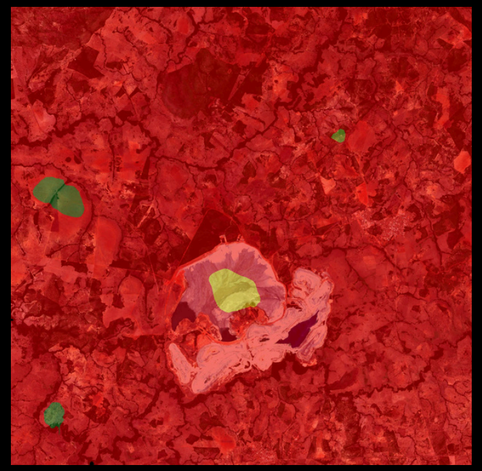
Segmentation Class overlay	**Suspicious** Red highlight
Bounding Boxes Candidates	**Confirmed** Final detections

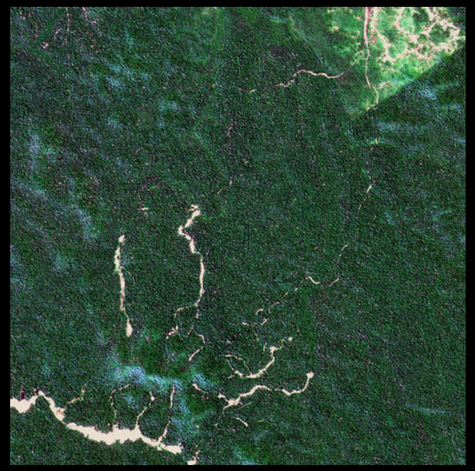
KEY CLASSES & METHODS:

Model: segformer-b2-finetuned-with-LoveDA

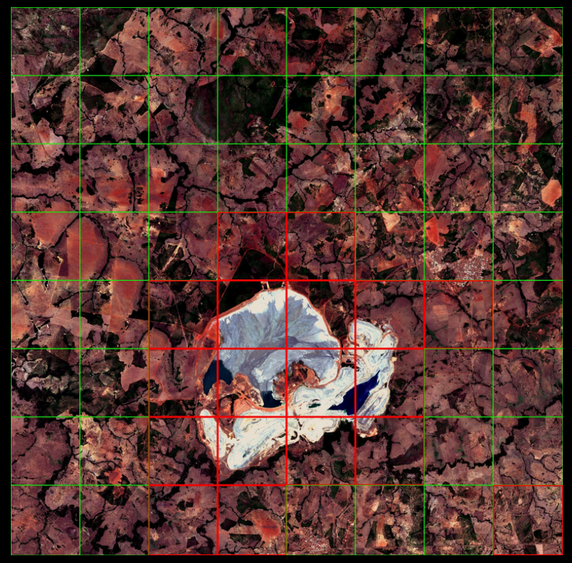
```
MiningSegmentationDetector.predict_mask(image)          → Returns raw class ID mask (H, W)
MiningSegmentationDetector.get_suspicious_mask(mask)    → Binary mask + morphological smoothing
DetectionVisualizer.draw_segmentation_overlay()         → Colored class visualization
```

RESULTS





Two Overview Examples
with highlighted candidate zones



Zones Classified as Mining/Non mining

CONCLUSIONS AND IMPROVEMENTS

Problems

- Model might only be capable of predicting mining dams
- Contrast between classes might be too big => easy classification between forest and ground but mabye harder between barren and actual mine

Possible Improvements

- Better informed sampling for the segformer
- Enhance the dataset with data other than mining dams
- a uniform approach to collecting data
- Ability to use the true validation step (needs actual data)