



Motivation & Problem Statement

Real-world gap:	Temporal shift risk	Cross-dataset risk	Current fix	Goal
ML cloud-detection models work on training-time data, fail on new scenes	Even weeks/months apart → big accuracy drop	Curated → raw Sentinel-2, or to other satellites → big accuracy drop	Fine-tuning = slow and expensive	Create robust, transferable models without retraining

Can we make a cloud segmentation model generalize across time and datasets —without re-training every time?

Our hypothesis: using a lightweight percentile-based normalization step can reduce spectral inconsistencies enough to make models transferable across real-world conditions

Research Objective

- 1 Improve generalization of cloud detection models.
- 2 Reduce the need for frequent model retraining.
- 3 Evaluate effectiveness of normalization for multi-temporal and cross-dataset satellite imagery.



Key Research Design Choices

Satellite Imagery

- Sentinel-2, Level-1C (13 bands, 10 m resolution)

Training Dataset

- CloudSEN12+
- High-quality manual labels
- Focused on temperate zones
- June imagery

Testing Dataset

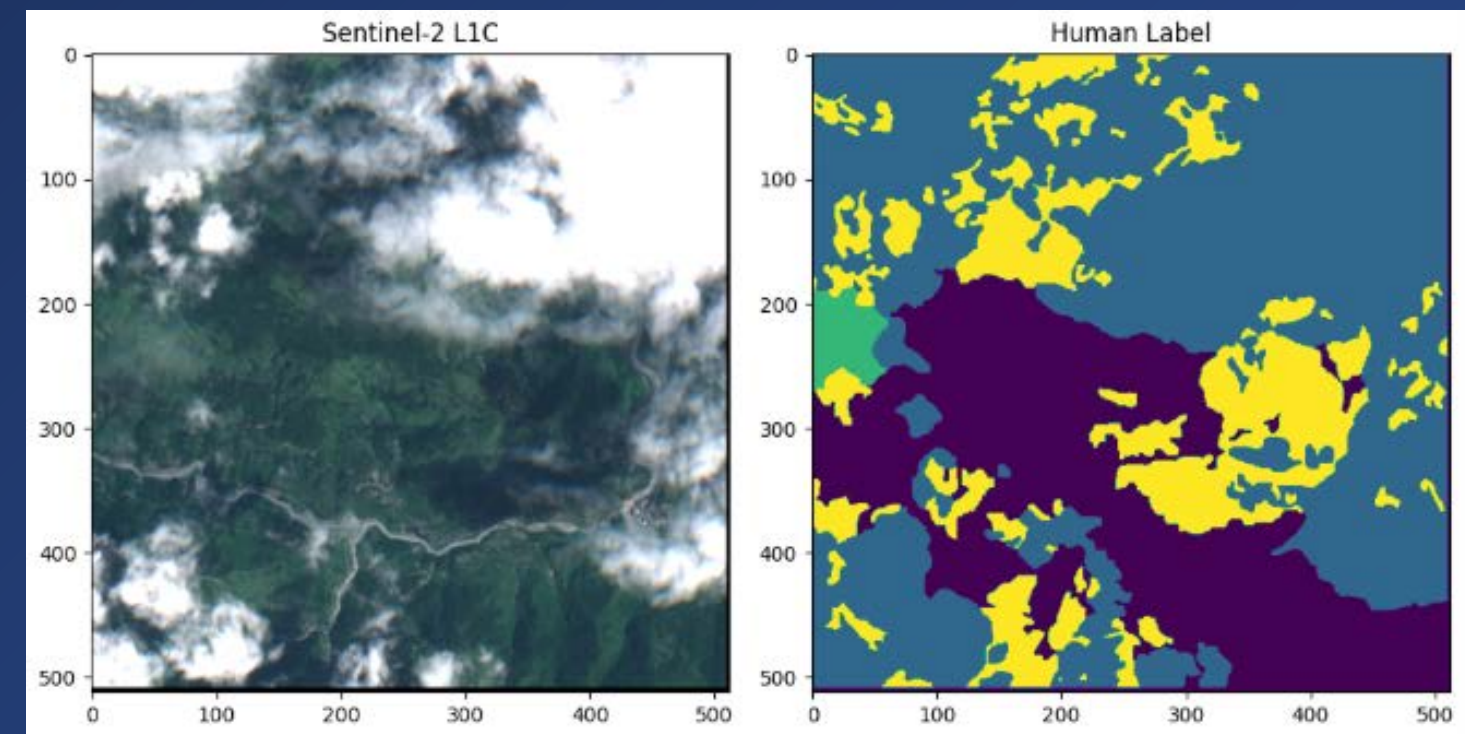
- SentinelHub Image
- August imagery
- Real-world temporal & dataset shift

Model Architecture

- U-Net with MobileNetV2 encoder

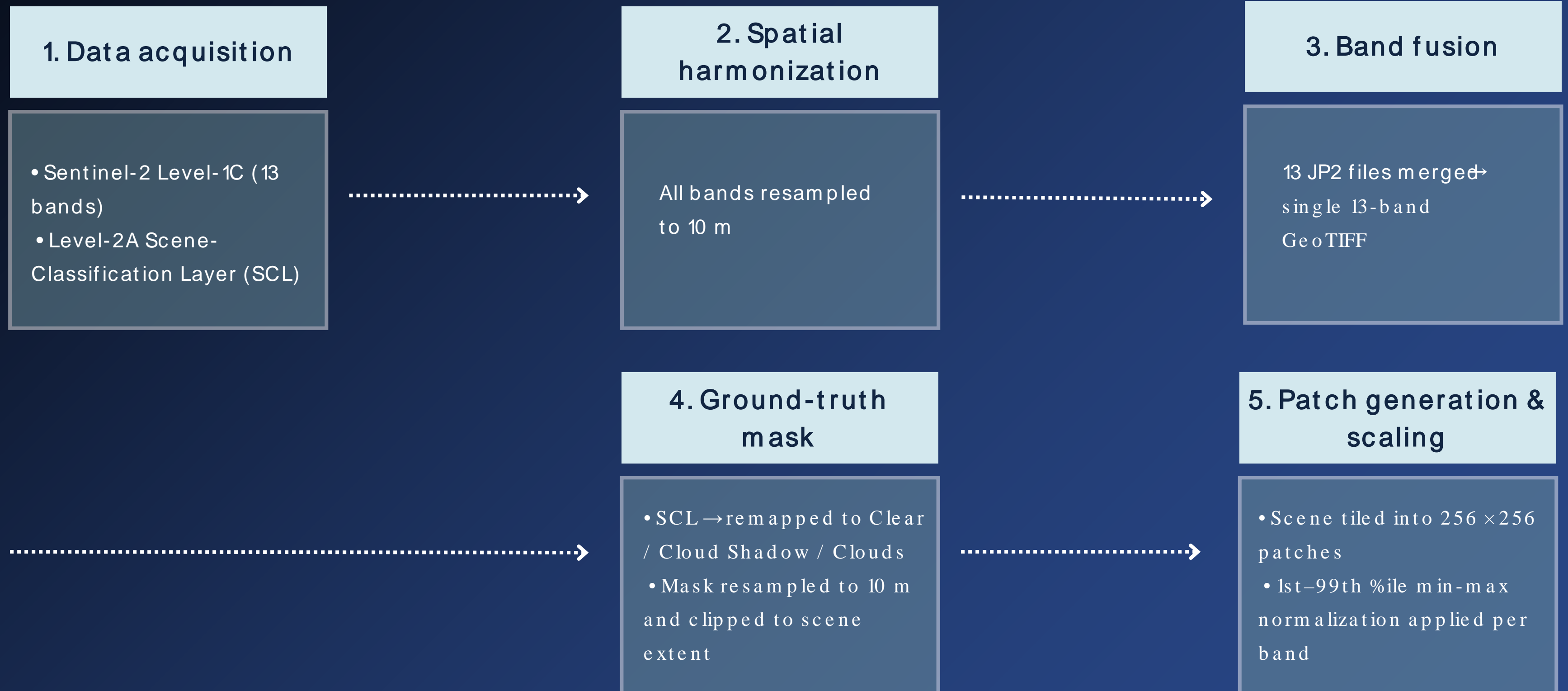
Normalization Method

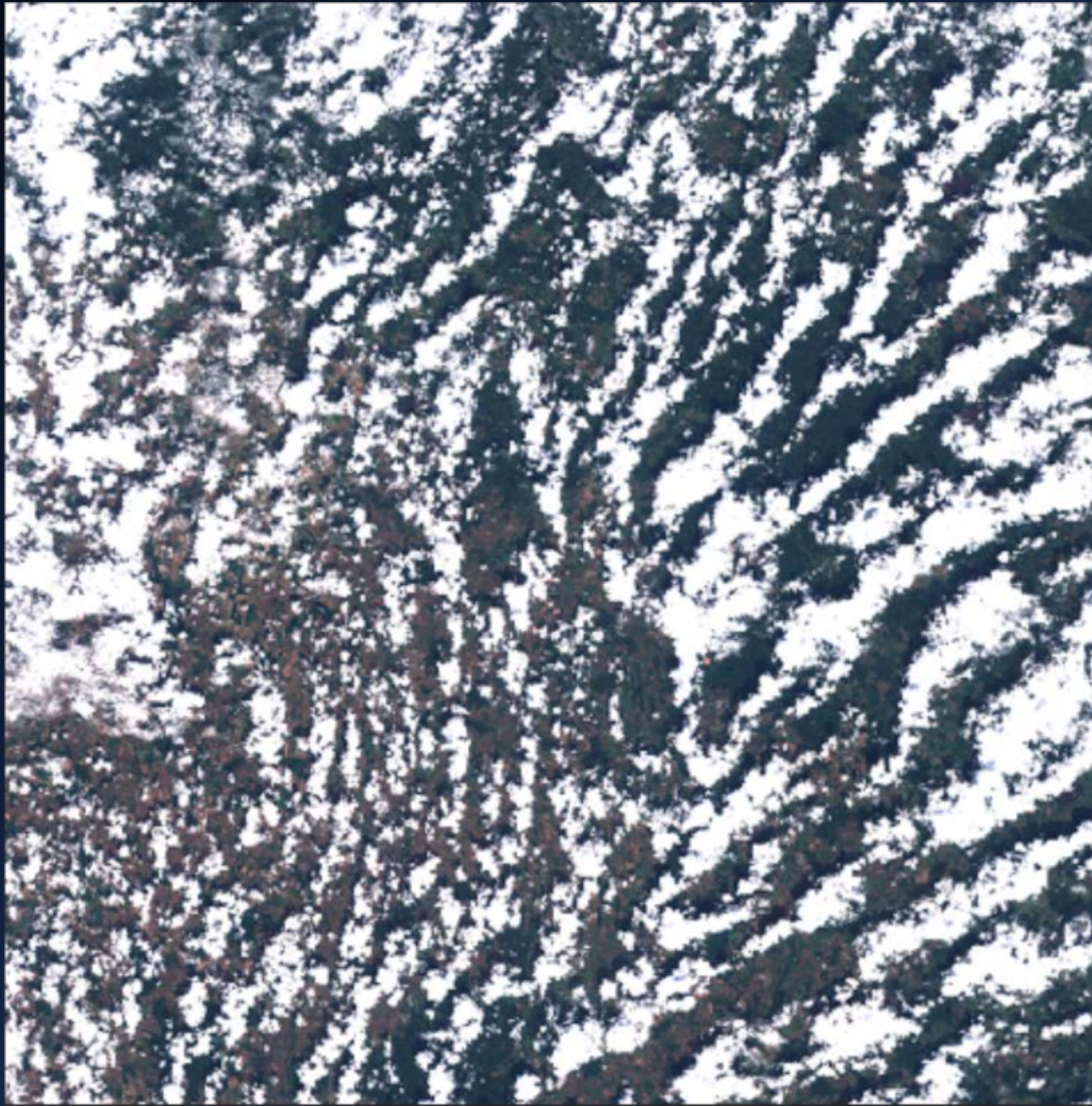
- 1st-99th Percentile Min-Max per band



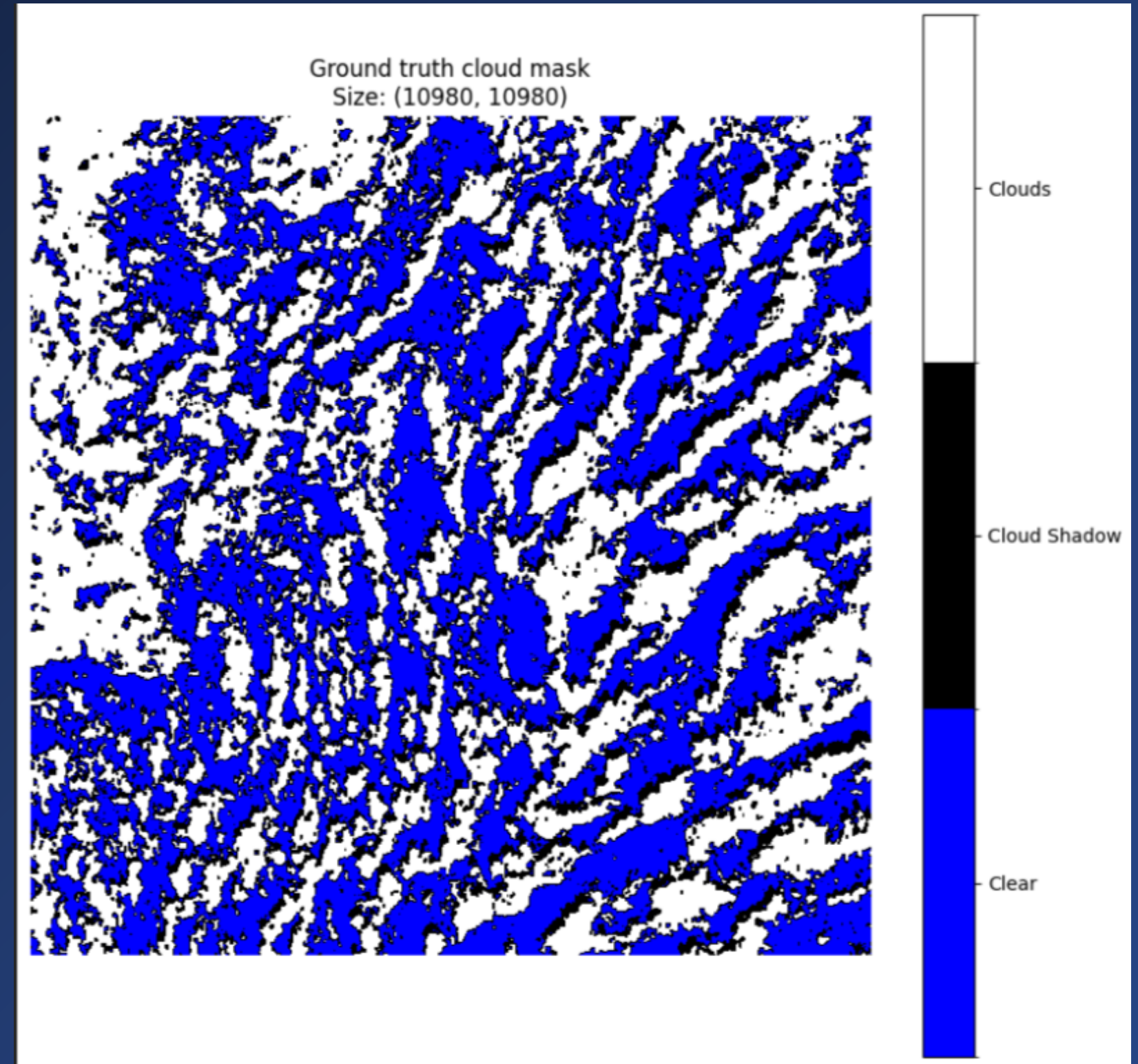
Picture 1 Example of Sentinel-2 L1C Image and Corresponding CloudSEN12+ Human-Labeled Mask

Pre-processing the Sentinel Hub Test Scene





Picture 2 Sentinel Hub image for testing



Picture 3 Generated mask for Sentinel Hub image

Model Architecture & Training Setup

Architecture

- U-Net encoder-decoder with skip connections
- Encoder: MobileNet V2 (ImageNet-pretrained)
- Input: 13 Sentinel-2 bands (256 × 256 px patches)
- Output: 4 classes – Clear, Thick Cloud, Thin Cloud, Cloud Shadow

Training Parameters

- Optimizer Adam, LR 1×10^{-3}
- Loss Multiclass Dice Loss (handles class imbalance)
- 30 epochs, batch size 16, early-stop on val IoU

Frameworks

- PyTorch + Segmentation Models PyTorch
- Orchestrated with PyTorch Lightning

Evaluation Metrics

- Class-wise & mean IoU
- Overall pixel Accuracy

```
class CloudSegmentationModel(pl.LightningModule):  
    def __init__(self, lr=1e-3):  
        super().__init__()  
        self.model = smp.Unet(  
            encoder_name="mobilenet_v2",  
            encoder_weights="imagenet",  
            in_channels=13,  
            classes=4  
        )  
  
        self.criterion = smp.losses.DiceLoss(mode="multiclass")
```

Picture 4 U-Net with MobileNetV2 encoder

Experimental Setup

No normalization

Training Dataset:

- CloudSEN12+ June patches (high-quality, temperate zone)
- No normalization applied initially



Baseline Model:

- U-Net + MobileNetV2 encoder trained for 30 epochs
- Metrics: IoU (per class & mean), accuracy



Testing Scenarios:

- Intra-dataset: CloudSEN12+ test set (same distribution)
- Cross-dataset: Sentinel Hub (August, Latvia)

With normalization

Normalization Experiment:

- Applied 1st–99th %ile min-max normalization per band
- Same model architecture retrained with normalized data
- Evaluated with same metrics & test scenarios

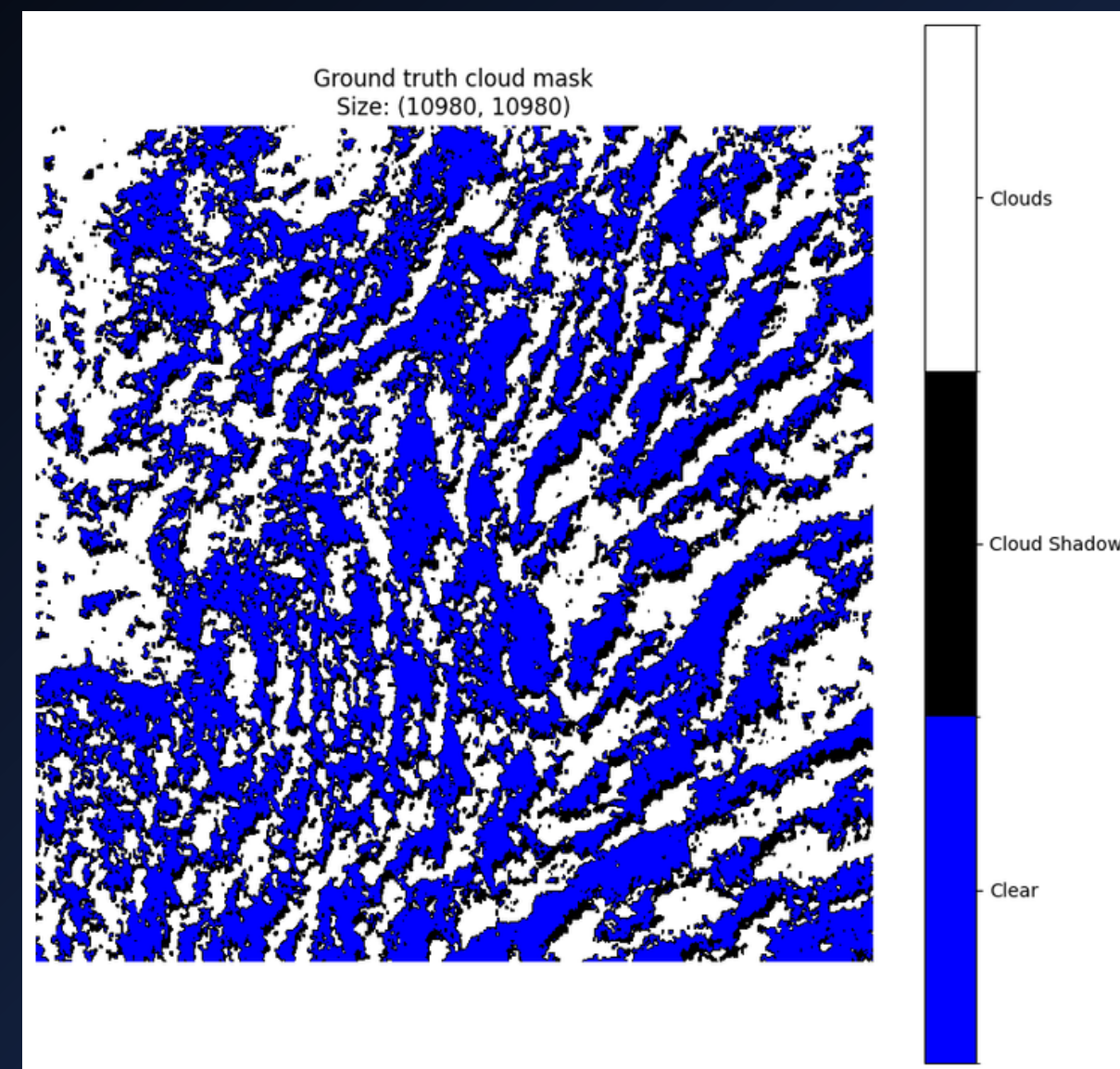
$$X_{norm} = \frac{clip(X, P_{99}, P_1)}{P_{99} - P_1}$$

Picture 5 Percentile-Based Min-Max Normalization Formula

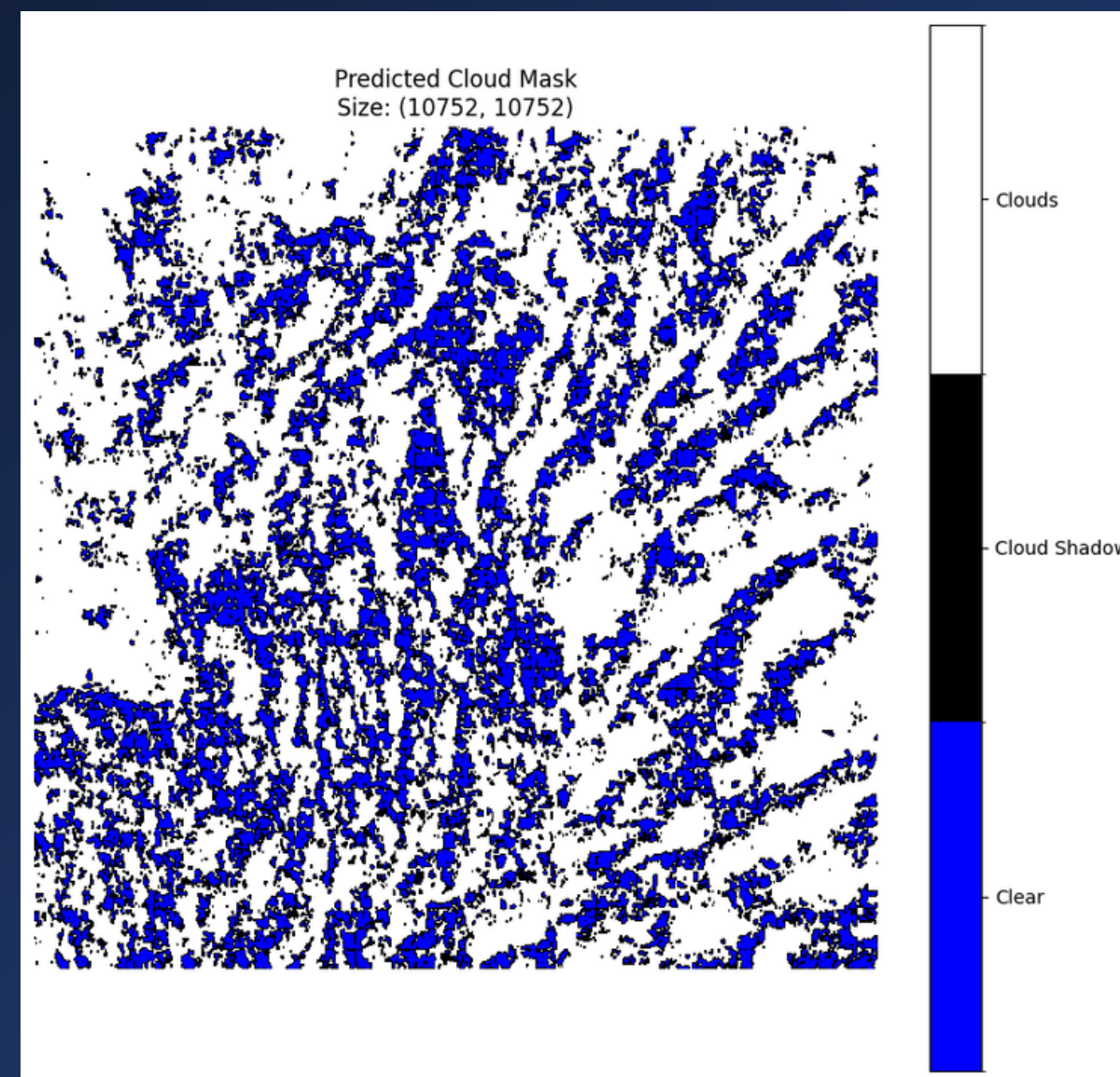
- X: pixel values of a Sentinel-2 band
- P_1 and P_{99} : the 1st and 99th percentiles of that band (calculated from the training set)
- $clip(X, P_{99}, P_1)$: clamps pixel values to within $[P_1, P_{99}]$ to remove outliers

Results – Transferability of Cloud Segmentation

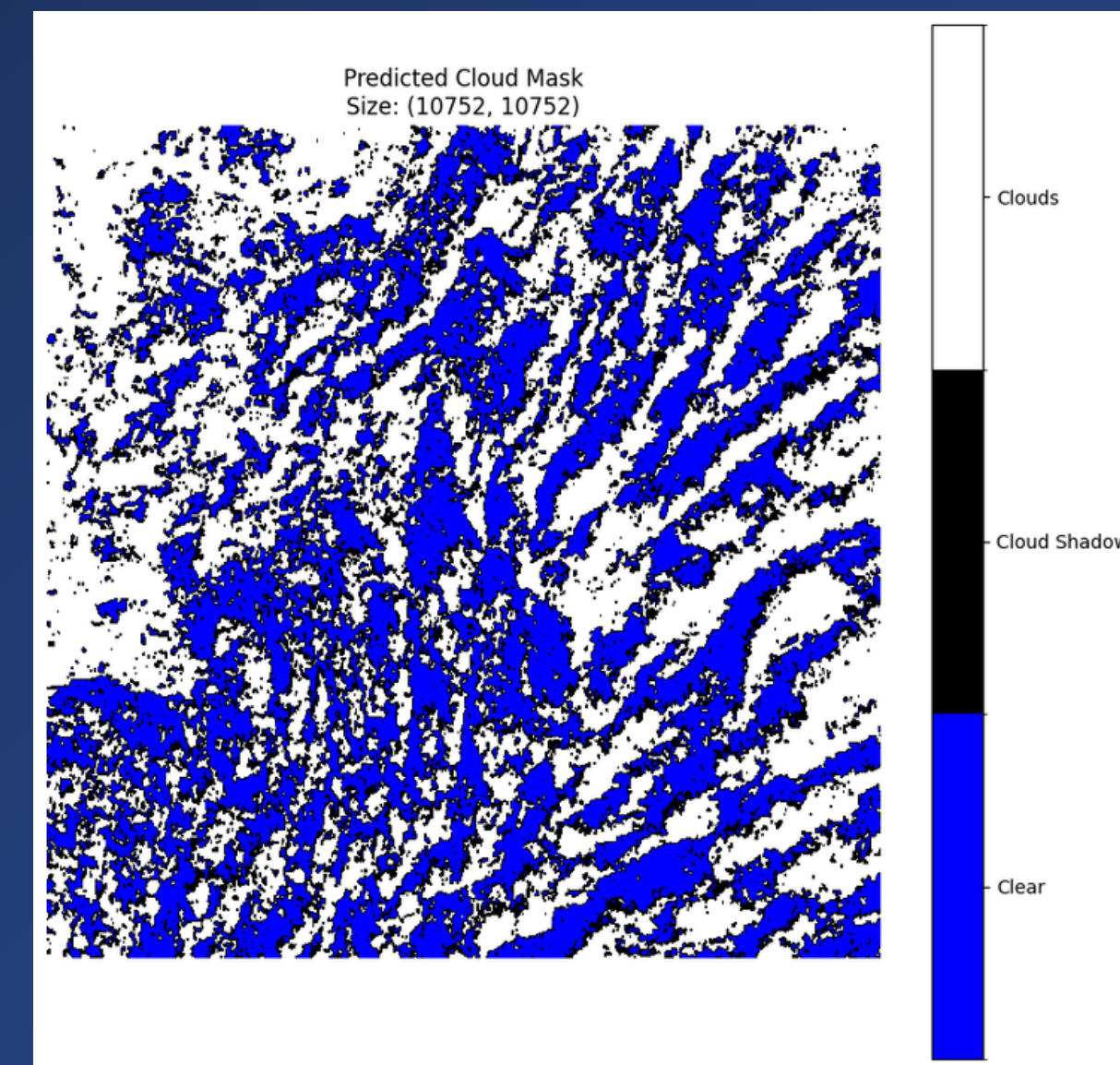
Scenario & Metric	Baseline (No Normalization)	With Percentile-Based Normalization
Intra-dataset (CloudSEN12+ June)		
Mean IoU	0.74	0.74 (no change)
Overall Accuracy	93.7 %	93 %
Cross-dataset (Sentinel Hub August)		
Clear IoU	0.43	0.61
Cloud Shadow IoU	0.21	0.45
Clouds IoU	0.47	0.53
Overall IoU	0.37	0.53
Overall Accuracy	65 %	73 %



Picture 6 Ground Truth Cloud Mask



Picture 7 Predicted Cloud Mask on Sentinel Hub image (without normalization)



Picture 8 Predicted Cloud Mask on Sentinel Hub image (with normalization)

Conclusions

Percentile-based normalization improved model transferability for cloud detection, maintaining high performance on known data and boosting accuracy on new datasets.

Normalization especially enhanced detection of cloud shadows.

However, labeling differences between datasets limit its effectiveness—additional strategies like domain adaptation are needed for full generalization.

