Environmental change detection WITH DINOv3
(Using OSCD Dataset)

Karimov Temurbek
Kadirov Bekzod

Project Proposition Value

- Our project proposes an automated change detection system that compares satellite images from different years to identify how landscapes, infrastructure, and urban areas have evolved over time.
- This solution enables analysts, researchers, and local authorities to efficiently monitor long-term changes without manual inspection.

Value Provided

- Automated detection of year-to-year changes in buildings, vegetation, land use, and infrastructure
- Improved accuracy using transformer-based features, including DINOv3
- Faster analysis compared to manual comparison of satellite images
- Scalable monitoring for large geographic regions
- Supports data-driven decision-making for urban planning, environmental studies, and disaster prevention

Why This Problem Matters ?

Primary Goal: Develop an automated end-to-end pipeline for multi-temporal satellite image change detection using the state-of-the-art DINOv3 vision transformer.

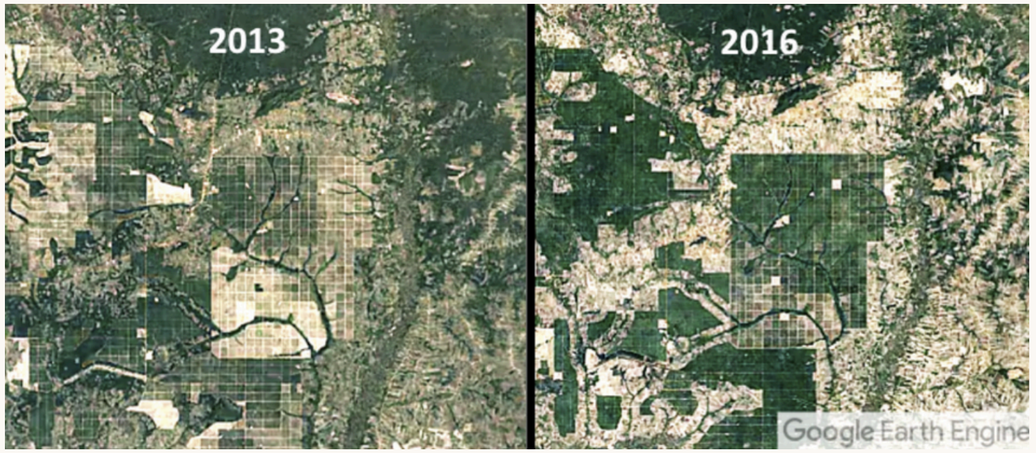
Technical Objectives:

- Feature Extraction: Leverage self-supervised pre-trained DINOv3 features to capture subtle environmental and structural transformations.
- Accuracy Enhancement: Achieve higher precision in identifying urban expansion and deforestation compared to traditional CNN-based methods.
- Scalability: Ensure the model can process large-scale datasets (like OSCD) efficiently.

Impact Objective: Provide a robust tool for urban planners and environmental researchers to monitor land-use changes with minimal manual intervention.

Urban planning: Cities expand, infrastructure evolves, new buildings appear.

Environmental monitoring: Forest loss, agricultural expansion, water-level changes.

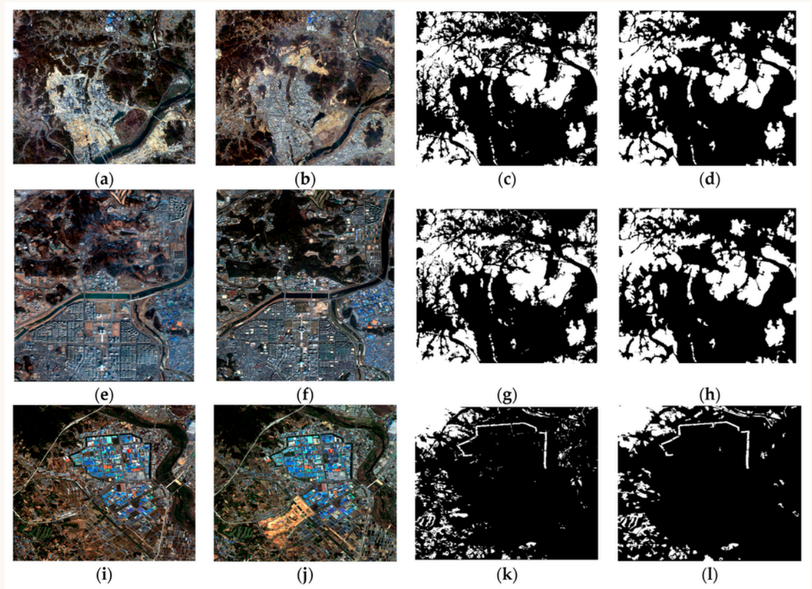


Infrastructure or landscape transformation over years

Research Questions & Hypotheses

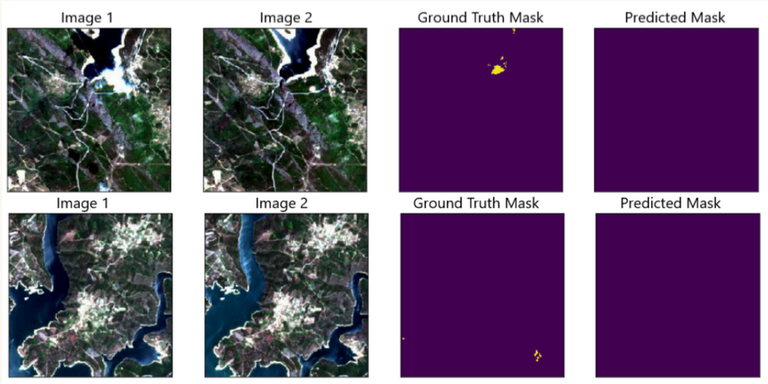
Research Questions

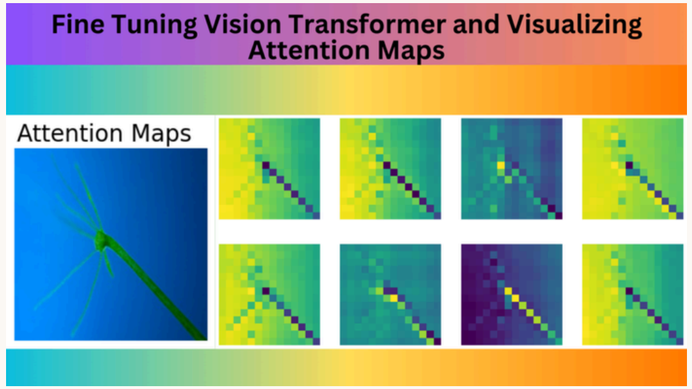
- RQ1 — Can DINOv3 detect subtle changes?
- RQ2 — Can we detect different types of changes?
- RQ3 — Does multi-year comparison improve consistency?
- RQ4 — Which fusion method works best?
- RQ5 — Can the model generalize to different regions?

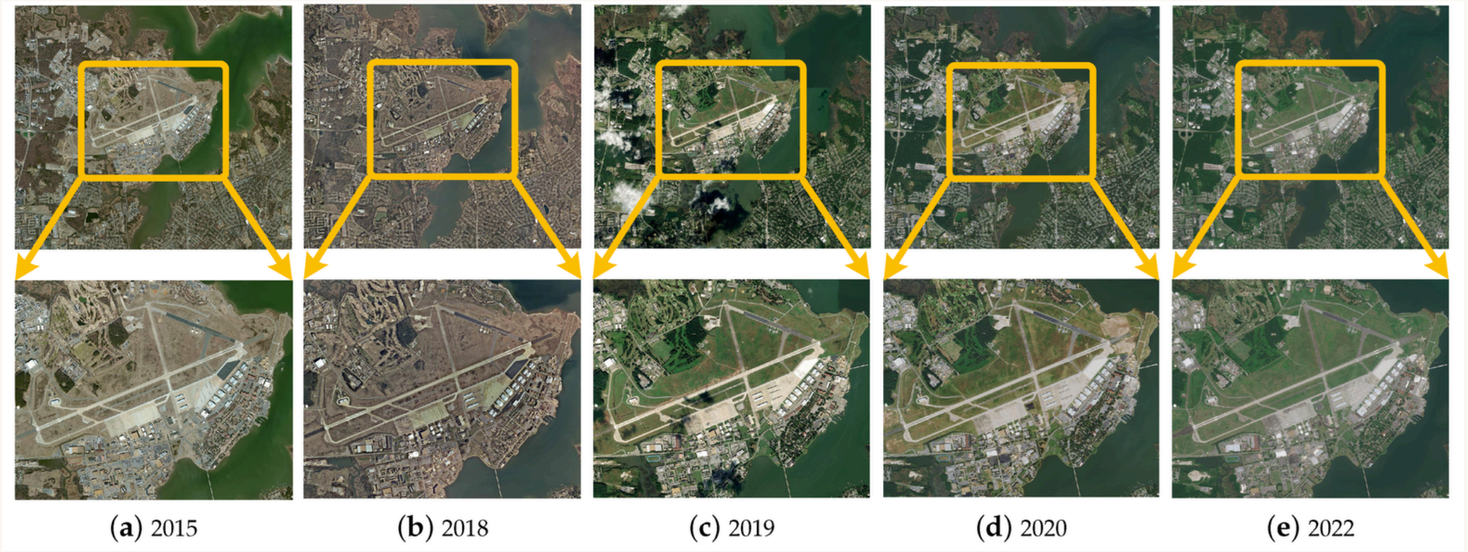


Hypotheses

- H1: DINOv3 features improve sensitivity to fine-grained structural and land-cover changes.
- H2: A dual-branch or multi-temporal architecture outperforms single-image models for identifying long-term transformations.
- H3: Combining multiple years of imagery results in smoother and more reliable change maps.
- H4: Spatial augmentations and patch-based processing increase robustness to noise, lighting differences, and seasonal effects.







Methodology & Workflow

Data Preparation:
Utilizing the OSCD dataset and Google Earth Engine to obtain multi-temporal satellite imagery

Feature Extraction:
Implementing a dual-branch hierarchical transformer encoder based on DINOv3 to extract deep spatial features from pre-change and post-change images.

Preprocessing:
High-resolution satellite imagery is spatially aligned, normalized for lighting, and processed using 256x256 pixel patches with a 128-pixel overlapping stride to ensure precise feature extraction

Feature Fusion:
Using a difference module to compare features from different years and highlight structural transformations.

Classification:
A lightweight MLP decoder processes the fused features to generate a final Binary Change Map.

OSCD Dataset samples used for training and testing (Ref. Brasilia city)

Pre-image

Post-image

Ground Truth

Development Choices & Techniques

Initial Design Choices

- We started with a Transformer-based encoder to capture global spatial context.
- A Mask2Former-style decoder was initially used for segmentation.
- Training used Binary Cross-Entropy loss and full backbone fine-tuning.

Issues Observed in Early Experiments

- Training was unstable when fully fine-tuning a large Vision Transformer.
- Binary Cross-Entropy loss was dominated by background pixels.
- Overlap-based metrics (IoU, F1) remained low despite high overall accuracy.

Backbone: Vision Transformer with DINOv3

- We use a large Vision Transformer pretrained using self-supervised DINOv3.
- Exact model:
- vit_large_patch16_dinov3.sat493m
- Self-supervised pretraining provides strong representations without dense labels.

Siamese Encoder Design

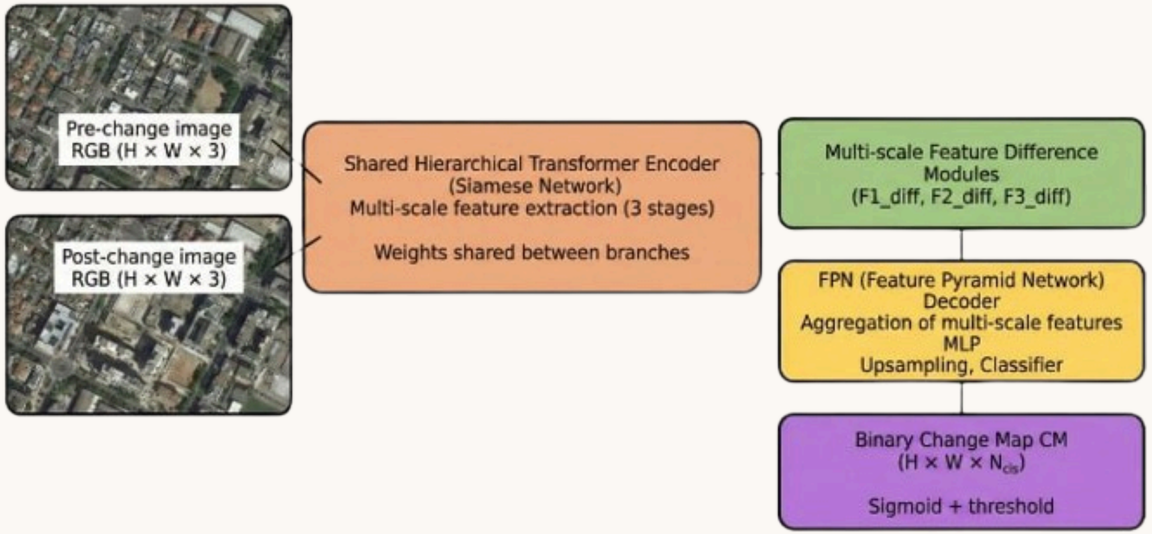
- Pre-change and post-change images are processed by the same backbone.
- Weights are shared to enforce temporal consistency.
- The backbone outputs multi-scale feature maps at different resolutions.

Development Choices & Techniques

•An FPN is a decoder architecture that combines features at multiple spatial scales.
•High-level semantic features are propagated to higher resolutions using a top-down pathway.
•This allows precise localization of small objects or sparse changes.

•Urban changes are often small and spatially sparse.
•Single-scale decoding leads to fragmented or coarse predictions.
•FPN integrates fine details with high-level context, improving localization.

•At each scale, pre- and post-change features are concatenated.
•A learnable projection maps them to a fixed 256-dimensional space.
•This decouples the decoder from backbone-specific feature dimensions.



Evaluation and Training Strategy & Expected Outcomes

Loss Functions: BCE vs Focal Loss

- Binary Cross-Entropy treats all pixels equally.
- In change detection, background pixels dominate the loss.
- Focal loss down-weights easy negatives and focuses learning on hard change pixels.

Training Strategy Evolution

- Stage 1: freeze the backbone to stabilize early training.
- Stage 2: unfreeze only the last two Transformer blocks.
- This prevents catastrophic forgetting while allowing task-specific adaptation.

Threshold Tuning

- The model outputs a probability map via a sigmoid activation.
- A decision threshold is tuned on the validation set.

We select the threshold that maximizes the F1-score to balance precision and recall.

Expected Outcomes

- More accurate identification of long-term changes (urban expansion, deforestation, infrastructure growth)
- Cleaner and more interpretable change maps compared to classical methods
- Higher sensitivity to subtle transformations thanks to DINOv3 features
- Robust performance across years with varying lighting and seasonal artifacts
- Practical usefulness for analysts, planners, and environmental teams

Experiments & Results

Experimental Setup & Configuration

Parameter	Value	Purpose / Details
Backbone	vit_large_patch16_dinov3.sat493m.	Self-supervised pre-trained model for robust feature extraction.
Patch Size	256x256	Optimized for Vision Transformer spatial resolution.
Stride	128	Provides 50% overlap to ensure spatial consistency.
Optimization	AdamW 8	Standard for Transformers; used with 100 epochs.
Learning Rates	Decoder: 3x10^-5; Backbone: 5 x10^-7	Differential rates to protect pre-trained backbone features.
Loss Function	Focal Loss	Specifically used to mitigate severe class imbalance.
Strategy	Two-stage Fine-tuning	20 epochs frozen, then unfreeze last 2 ViT blocks.
Augmentation	Flips, Rotations, Rescaling	Paired transforms to reduce seasonal and illumination bias.

Comparative Results

Method	Overall Accuracy (OA)	Change Accuracy
OSCD Siamese CNN (Baseline)	84.13%	78.57%
OSCD Early Fusion CNN	83.63%	82.14%
Siamese DINOv3 + FPN (Ours)	**94.84%**	**53.44%**

Examples of the results
(Ref. Las-Vegas city)

Pre-image

Post-image

Ground Truth

Prediction

Conclusions

- ***Model Effectiveness***:
  - Our Siamese DINOv3 + FPN architecture achieved a superior Overall Accuracy of 94.84%, proving highly effective for urban change detection.
- ***Technical Insight***:
  - The use of Focal Loss and a two-stage fine-tuning strategy (partial unfreezing) significantly improved training stability and handled class imbalance.
- ***Practical Impact***:
  - The system provides a reliable, automated pipeline for monitoring infrastructure development with minimal manual oversight.

Future Work

- ***Per-city Error Analysis:***
  - Conduct a deeper investigation into how different urban landscapes (e.g., Las Vegas vs. Paris) affect model precision.
- ***Multi-class Detection:***
  - Extend the model to distinguish between types of changes (e.g., new buildings vs. road construction).
- ***Alternative Fusion:***
  - Test different feature fusion operators to further improve the sensitivity to sparse changes.

Questions?