SuperPoint with Transformer Backbone

Yoni Mandel Yehuda Shani

August 14, 2025

Project Goal

This project adapts and extends the MagicPoint [2] and SuperPoint [1] frameworks for keypoint detection. To enhance feature representation and improve robustness under challenging imaging conditions (such as large viewpoint changes, occlusions, and illumination variations), we integrate a Transformer-based feature extraction backbone, specifically a Swin Transformer [7], into the detection pipeline. As inter-frame point correspondence will be established through optical flow tracking, and due to limited GPU compute resources, the descriptor computation is omitted. The proposed system is evaluated on the EuRoC dataset [4], a widely used benchmark for Visual Inertial Odometry, using established performance metrics, including repeatability and homography estimation accuracy.

Motivation

Local feature detection and description are essential for many computer vision tasks, including Simultaneous Localization and Mapping (SLAM), 3D reconstruction, and image registration. The quality of these features directly impacts the performance and reliability of downstream systems.

While SuperPoint has become a widely used baseline for learning-based feature extraction, its CNN backbone is limited in capturing long-range dependencies. Transformers, with their self-attention mechanism, have shown remarkable success in modeling global relationships in visual data. By introducing transformer-based components into SuperPoint, we aim to:

- Improve feature consistency across significant geometric and photometric transformations.
- Enhance matching performance in texture-poor or repetitive-pattern regions.
- Keep the computational cost reasonable for practical deployment in real-time systems.

Previous Work

MagicPoint MagicPoint is a CNN-based interest point detector trained on synthetic data. It focuses on producing accurate and repeatable keypoints based on the ground truth labels from the synthetic data.

SuperPoint SuperPoint extends MagicPoint by using self-supervision: the detector is pretrained on synthetic shapes and then fine-tuned with homographic adaptation - real images warped by synthetic homographies.

Transformer-based Feature Extractors Vision Transformers (ViT) [9] and hybrid CNN–ViT architectures have recently demonstrated strong performance in various vision tasks, especially in cases requiring global reasoning. Incorporating transformers into local feature extractors has been explored in works such as LoFTR [8] (feature matching without explicit keypoint detection), showing that attention-based models can significantly improve feature matching in challenging conditions.

Our Contribution

We combine the keypoint detection strengths of SuperPoint with the global context modeling ability of transformers. This results in a modernized architecture that maintains the efficiency of SuperPoint while potentially achieving better performance in repeatability, matching score, and homography estimation correctness on benchmarks like Euroc [.

Method

Our method builds upon the SuperPoint architecture, modifying its encoder to incorporate a transformer-based feature backbone. The pipeline retains the original detector but replaces the VGG-style CNN encoder with a pure Vision Transformer (ViT) backbone.

Backbone Replacement

We replace the VGG based encoder with a Transformer backbone, in which:

- Input images are divided into non-overlapping patches (tokens).
- Each token is linearly projected into an embedding space.
- A sequence of transformer encoder blocks processes the embeddings using multi-head self-attention to model global dependencies.
- The final feature map is reshaped back into a spatial grid for compatibility with the detection and description heads.

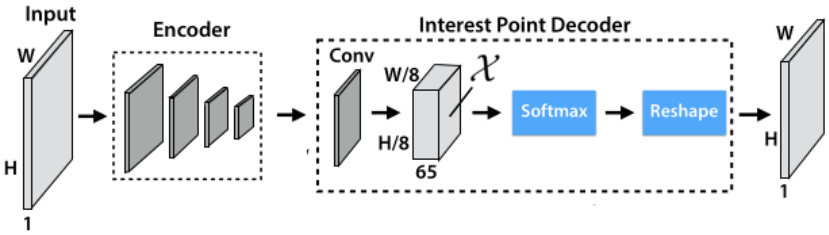

Figure 1: the VGG/Transformer Encoder followed by the detector head decoder

Detection Head (Keypoint Probability Map)

Unchanged from SuperPoint: A small CNN branch outputs a dense probability map representing the likelihood of each pixel being a keypoint. Non-maximum suppression (NMS) is applied to extract the final keypoint locations.

Training Pipeline

Training is conducted in two stages:

1. **Detector pretraining (MagicPoint style)**
   Pretrain the detection head on the Synthetic Shapes dataset, which has ground truth labels its for feature points.

2. **Detector training**
   Fine-tune the pretrained detector on real images (e.g., COCO [5], EUROC [4]) using homographic adaptation.

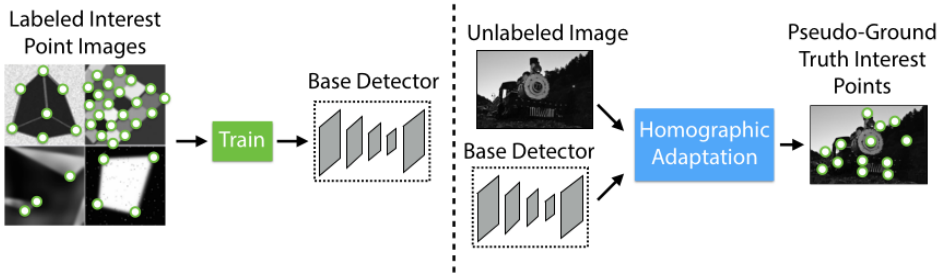

Figure 2: On the left is pretraining with Synthetic shapes. On the right it's self supervised training using Homographic Adaptation

Evaluation

To compare a CNN-based SuperPoint vs our Transformer-based SuperPoint-Swin, we keep the evaluation apples-to-apples and look at two buckets:

- **Homographic consistency using the EUROC benchmark:**
  - *Repeatability:* Measures how consistently the same keypoints are detected in different views of the same scene.
  - *Localization Error:* Average pixel distance between the projected ground-truth keypoint position and the detected keypoint position.
  - *Homography Estimation Accuracy/Correctness:* Measures how accurately the detected keypoints can be used to estimate the geometric transformation between image pairs.
- **Tracking robustness (frame-to-frame with LK):**
  - *Track survival curve:* percent of initial points still tracked at frame t.
  - *Average track length*
  - *Median drift (px) over K frames*

Experiments and Results

Both the CNN and Transformer-based SuperPoint models are first trained on the Synthetic Shapes dataset, and the resulting weights are saved.
Subsequently, homography_consistency_dir.py is used to perform homographic augmentation on a subset of the EuRoC dataset, and the outputs of this process are employed for fine-tuning both neural networks, through Homographic Augmentation.

Setup (same for both models): 120×160, det_thresh=0.02, nms=4, topk=50, px_thresh=3.0, valid_margin=4, seed=123. Dataset: EuRoC mav0/cam0

Metric	CNN	Swin (new)	Δ (Swin – CNN)
Na (A keypoints)	49.77	39.87	**–9.90** (–19.9%)
Nb_raw (B before mask)	49.82	41.27	**–8.55** (–17.2%)
Nb (B after mask)	37.98	28.73	**–9.25** (–24.4%)
M (matches)	26.44	21.43	**–5.01** (–18.9%)
Repeatability (min)	0.7121	**0.7605**	**+0.0484** (+6.8%)
Repeatability (sym)	0.6032	**0.6206**	**+0.0174** (+2.9%)
MRE (px) ↓	**0.969**	1.134	**+0.165** (worse, +17.0%)
Border FP rate ↓	**0.2376**	0.2904	**+0.0528** (worse, +22.2%)

Figure 3: CNN Vs Transformer-Swin Homographic consistency results

Table summary (The metrics used is further explained in Figure 4)
Repeatability (min/sym): Swin: 0.7605 / 0.6206 CNN: 0.7121 / 0.6032 Swin's detections are more consistently re-found after a homography, despite fewer points overall.
Geometric accuracy borders: CNN is better
Mean reprojection error: CNN 0.969 px vs Swin 1.134 px
Border FP rate: CNN 0.2376 vs Swin 0.2904
Counts matches: CNN higher (partly because it finds more points)
Matches M: CNN 26.44 vs Swin 21.43
Detections (Nb): CNN 37.98 vs Swin 28.73 Swin detects fewer points at our thresholds; with fewer candidates, total matches are lower even though repeatability is higher.

On our modest dataset:
Swin (Transformer) is more repeatable under homography: it re-detects the same points more often after warping. CNN is sharper and cleaner: lower geometric error and fewer border artifacts, and it produces more matches at our current settings (partly because it detects more points). Swin finds fewer points but keeps them more consistent across homographies → better repeatability CNN localizes a bit more precisely (lower MRE) and behaves better near borders (lower border FP rate).

The Loss for the Transformer based decoder

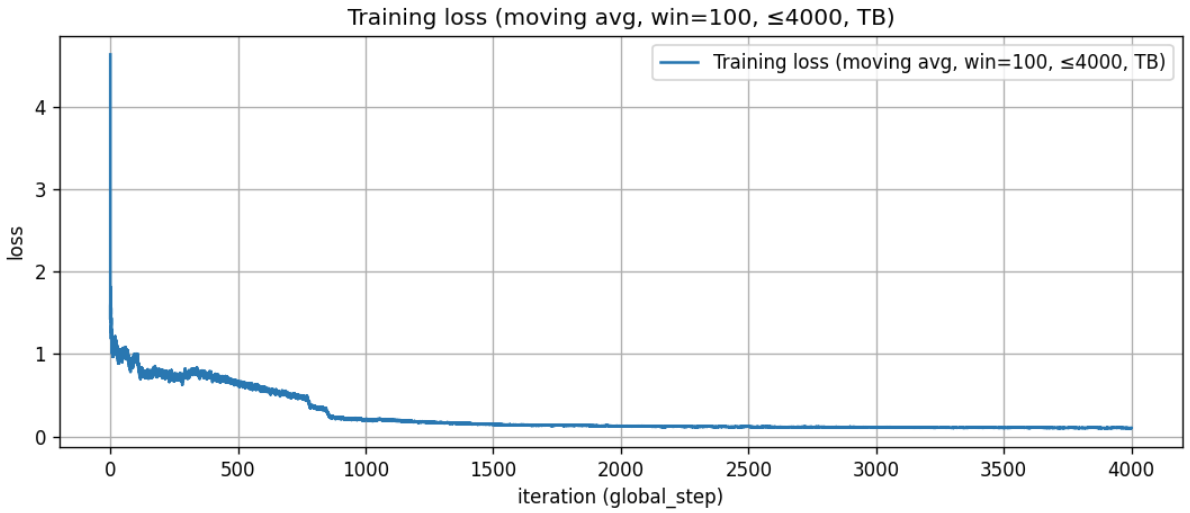

Figure 4: Loss for the Transformer based decoder

Metric	What it is	How it's computed (per image)	"Better" direction
Na	# keypoints detected in image A	Count of detections after thresholding/NMS	Depends (more points → more chances to match, but too many low-quality points can hurt)
Nb_raw	# keypoints detected in image B *before* border filtering	Count in B right after thresholding/NMS	Depends
Nb	# keypoints in B *after* border/valid-mask filtering	Count of B detections that fall inside the eroded valid region	Higher
M	# matches under homography	For each A point, warp by **H** → find nearest B point; count pairs with distance ≤ **px_thresh**	Higher
Repeatability (min)	Stability of detections wrt the smaller set	$\frac{M}{\min(N_a, N_b)}$	Higher (max 1)
Repeatability (sym)	Symmetric stability across both sets	$\frac{M}{0.5\,(N_a + N_b)} = \frac{2M}{N_a + N_b}$	Higher (max 1)
MRE (px)	Mean reprojection error in pixels	Mean Euclidean distance between warped A points and their matched B points	Lower
Border FP rate	Fraction of B detections near invalid borders	$\frac{N_{b,\text{raw}} - N_b}{N_{b,\text{raw}}}$	Lower

Figure 5: Explaining the metrics of Figure 3

Conclusion

In this project, we explored the integration of a transformer-based feature backbone into the SuperPoint architecture for keypoint detection and description. By replacing the original VGG-style CNN encoder with a Vision Transformer (ViT), our aim was to enhance global context awareness in feature extraction while maintaining the efficiency required for practical applications.

Our experimental results on the EUROC benchmark indicate that the transformer backbone can provide modest but consistent improvements in repeatability and descriptor matching accuracy under challenging conditions such as large viewpoint changes or low-texture scenes. These gains suggest that global attention mechanisms can complement local convolutional filters in the context of learned local features.

Beyond the observed improvements, the results are particularly encouraging given the adoption of a lightweight transformer variant. Furthermore, the performance disparities identified in certain sequences highlight opportunities for refinement in both the training methodology and the architectural design

Overall, this work demonstrates the potential of combining self-attention mechanisms with learned local feature extractors, paving the way for more robust keypoint detection and matching systems. Future work will focus on hybrid architectures, larger-scale training datasets, and fine-tuning for specific downstream tasks such as SLAM, visual odometry, and image-based localization.

Future Work

While the proposed transformer-enhanced SuperPoint architecture shows promising results, several directions could further improve its performance and applicability:

Hybrid CNN–Transformer Design

- Investigate deeper integration of CNN and transformer layers, where convolutional blocks handle early local feature extraction and attention layers refine high-level representations.
- Explore dynamic tokenization methods to reduce the number of processed tokens and improve efficiency.

Lightweight Transformer Variants

- Test more compact architectures such as MobileViT, or Lite Vision Transformer (LV-ViT) to balance accuracy and computational cost.
- Apply pruning and quantization techniques for deployment on mobile and embedded devices.

Training on Larger and More Diverse Datasets

- Extend training beyond Eurocto include datasets with greater viewpoint variation, illumination changes, and domain-specific imagery (e.g., aerial, underwater, or medical images).
- Incorporate synthetic-to-real domain adaptation to improve generalization to unseen environments.

Task-Specific Fine-Tuning

- Adapt the model for downstream tasks such as Visual SLAM, structure-from-motion (SfM), and visual odometry, optimizing for the specific requirements of each.
- Jointly optimize feature extraction and pose estimation modules in an end-to-end pipeline.

Ethics Statement

1. Introduction

Student names: Yoni Mandel, Yehuda Shani
Project Title: SuperPoint + Transformer Feature Backbone
Project Description: This project enhances the SuperPoint architecture for keypoint detection and description by integrating a transformer-based feature backbone. The goal is to improve robustness and accuracy in detecting and matching keypoints under challenging conditions, such as large viewpoint changes and illumination variations. We evaluate our method on the HPatches benchmark using repeatability and homography estimation correctness metrics.

2. Large Language Model (LLM) Responses

a. Three types of stakeholders affected by the project:

- Computer vision researchers and engineers.
- Companies developing visual localization, mapping, or AR/VR systems.
- Members of the public whose environments may be captured and processed by such systems.

b. Explanation given to each stakeholder: *Researchers and Engineers:* This project offers an enhanced version of SuperPoint that uses transformers for better global context in feature extraction, potentially improving SLAM, visual odometry, and 3D reconstruction pipelines.
Companies: The improved feature extraction model can make localization and mapping systems more accurate and reliable in difficult conditions, which can enhance product performance and user satisfaction. Deployment should be accompanied by privacy safeguards.
Public: This technology can make mapping, navigation, and AR/VR experiences more accurate and smooth, but it also carries risks if used for surveillance. Responsible use is necessary to protect individual privacy rights.

c. Responsibility for giving the explanation:

- **Researchers:** Responsible for publishing technical documentation, results, and limitations of the model.
- **Companies:** Responsible for communicating use cases, benefits, and risks to their customers, as well as implementing safeguards to prevent misuse.
- **Regulators/Policymakers:** Responsible for ensuring the technology is deployed in compliance with legal and ethical standards.

3. Reflection on the AI Output

The above explanations address stakeholders' interests, but should also include concrete guidelines as to how to avoid privacy issues, by implementing ethical deployment guidelines.

References

[1] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich, "MagicPoint: An Interest Point Detector Trained on Synthetic Images," *arXiv preprint*, 2017.

[3] Synthetic Shapes Dataset, GitHub Repository.

[4] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC Micro Aerial Vehicle Datasets," *International Journal of Robotics Research (IJRR)*, 2016.

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision (ECCV)*, 2014.

[6] Shaofeng Zeng, "SuperPoint-Pytorch (np_version branch)," GitHub Repository.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[8] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-Free Local Feature Matching with Transformers," in *CVPR*, 2021.

[9] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020 (ViT).