Scene-Aware Mobile Visual Localization Without GPS

Cristian McGee

December 2025

Abstract

Accurate, precise localization in dense environments remains challenging for GPS-based systems, which often provide only broad position estimates and degrade indoor and elevated structures. In this work, we explore a purely vision-based alternative tailored for quick mobile deployment and custom environments such as university campuses, theme parks, and large attractions. We focus on building-level localization on the University of Central Florida (UCF) campus using only mobile video content, a compact ViT-Tiny backbone, and a lightweight 3D map. We collected over 2M frames for our datasets by recording more than ten hours of video around 41 labeled campus buildings and introduce a k-sampling technique that combines multiple frames into single training elements. We further introduce a distance-aware loss that encodes the campus geometry, narrowing the performance gap between small and large models. We then fine-tune the ViT-Tiny model on these k-sampled elements and deploy it in a prototype mobile framework that runs entirely on-device without network connectivity, mapping visual predictions into a low-poly 3D campus environment. This enables GPS-free localization with building-to-building navigation using only local computation and publicly available GIS data. Our work suggests that cheap, scene-aware visual models are a promising foundation for affordable, GPS-independent navigation in complex real-world environments. [1]

1 Introduction

Accurate localization is a core requirement for modern navigation, robotics, and augmented reality. While the Global Positioning System (GPS) provides

[1] GitHub: `https://github.com/nizswan/UCF_VISION`

reliable global coordinates in many outdoor settings, it may be ineffective in dense urban environments, indoors, elevated surfaces, and areas with minimal connection. In places such as university campuses, theme parks, stadiums, or shopping centers, GPS can indicate which block a user is on but cannot reliably distinguish which building, entrance, or floor they are currently facing. Visual Positioning Systems and visual place recognition methods address this limitation by using camera images to infer location from the surrounding scene, typically by matching visual features against a database of known views [2, 6, 3].

Recent advances in Vision Transformers (ViTs) have reshaped the landscape of image recognition. Recent work shows that transformers applied directly to image patches can achieve state-of-the-art performance on standard benchmarks, rivaling and even surpassing convolutional networks [1]. Building on this, light-weight variants such as MobileViT demonstrate that transformer-based architectures can be adapted to run efficiently on mobile devices, achieving competitive accuracy with low latency and minimal parameter counts [4]. These developments make it increasingly feasible to deploy purely vision-based localization models directly on smartphones, without relying on continuous network connectivity or large cloud backends.

At the same time, most commercial visual localization systems leverage large-scale imagery and server-side infrastructure, limiting their applicability to custom environments such as university campuses, private industrial sites, or amusement parks [3, 6]. In these settings, organizations often have detailed maps or building information but lack a simple pipeline to connect those assets with an on-device, scene-aware localization model. Thus, there exists a need for methods that can be trained on locally collected images, run entirely on commodity mobile hardware, and integrated with 3D maps to provide intuitive navigation.

In this work, we explore a compact, vision-only localization system as a step toward GPS-free navigation. We focus on building-level localization on the University of Central Florida (UCF) campus using only mobile video, a ViT-Tiny backbone with fewer than 6 million parameters, and a low-poly 3D map rendered in Unity. Our k-sampling strategy converts video into sequentially embedded training samples, enabling the model to leverage sequential views of each building while remaining small enough for real-time, on-device inference. We further show how the resulting classifier can be integrated into a simple 3D campus environment to both localize the user and provide building-to-building navigation without GPS.

Our main contributions are threefold:

- We introduce a k-sampling based dataset construction procedure that concatenates multiple frames into single elements, allowing us to control the degree of temporal context while keeping the overall frame budget fixed.
- We demonstrate that a sub-6M-parameter ViT-Tiny model can achieve robust, building-level scene recognition similar to that of ViT-Large using a distance-aware loss function. We showcase that this model is suitable for mobile deployment on modern smartphones.
- We present a framework that connects visual predictions to a low-poly 3D campus map, enabling GPS-free localization and route planning between buildings using only local computation and publicly available GIS data.

2 Methodology

We segregate our methodology into a data collection process, training process, and deployment process in which we develop a cheap and effective scene-aware visual model that is capable of running locally.

2.1 Data Collection

We assigned a unique label to 41 significant buildings on the UCF campus. To optimize data collection, 1-2 operators walked around each landmark along circular routes, recording videos that captured a wide range of depths and viewing angles. Each video was recorded in portrait orientation with a 9:16 aspect ratio at 60 frames per second. In total, we collected over 10 hours of footage, corresponding to more than 2 million frames.

After video collection, we performed feature extraction by labeling each video according to its building class, since each recording contained only a single landmark. From these labeled sequences, we constructed our dataset using a k-sampling strategy that concatenates multiple frames into a single sample to encode sequential data, as illustrated in Figure 1. Since each frame has aspect ratio 9:16, a k-sampled element formed by horizontal concatenation has aspect ratio $9k$:16.

Given a sampling interval ρ, we traverse each video and, every ρ frames, generate a k-sampled element with $k \in \{1, 2, 3, 5\}$. The particular frame indices associated with each anchor frame are determined by Algorithm 1, where we fix $\xi = 2$ seconds and choose $\rho \in 23, 43$ depending on video length.

Figure 1: Varying k's in k-sampling

To make comparisons across different values of k fair, we maintain approximately the same total number of frames in each dataset while allowing the number of elements to vary. For example, the $k = 1$ dataset contains 10,000 training elements and 2,500 test elements, the $k = 2$ dataset contains 5,000 training and 1,250 test elements, the $k = 3$ dataset contains 3,300 training and 830 test elements, and the $k = 5$ dataset contains 2,000 training and 500 test elements. This construction yields multiple datasets with comparable frame budgets but different degrees of sequential data aggregation. Thus, allowing us to isolate the effect of k-sampling on downstream localization performance.

2.2 Training

For powerful classification on a small model, we fine-tune a ViT-Tiny backbone with fewer than 6 million parameters. Modern smartphones can already execute on-device neural networks with hundreds of millions to over one billion parameters, so our sub-6M model comfortably fits within typical mobile budgets [4, 5].

We consider two preprocessing pipelines for our experiments – *base* and *distort*. In the *base* setting, we downscale each k-sampled image to 224×224 to match the input requirements of ViT-Tiny. This resizing is applied to all values of k, so larger k elements undergo a stronger spatial downsampling. In the *distort* setting, we apply the same resizing practice, but additionally introduce random augmentations, (camera tilt and perturbations of brightness, hue, and saturation), to study how well the model generalizes from distorted training images to undistorted test images.

Let X denote the set of input elements and Y the set of building labels, with $(x_i, y_i) \in X \times Y$ and f the model. We denote $p(x_i)$ as the softmax output of f on x_i. For each class $y \in Y$, we associate a 2D position $\text{pos}(y) = (\text{pos}(y)_x, \text{pos}(y)_y)$ on a bird's-eye-view campus map, with the student union set as the origin (see § 2.3 and Figure 2). Using these coordinates, we define

Algorithm 1 k-sampling over video frames

Require: Indexed sequence of frames $I = (I_0, I_1, \ldots, I_{N-1})$, frame rate q (frames per second), time span ξ (seconds), sampling interval ρ (in frames), bundle size $k \in \mathbb{Z}^+$

$\Delta \leftarrow \lfloor \xi \cdot q \rfloor$ ▷ Span in terms of frames
Initialize dataset $\mathcal{D} \leftarrow [\,]$
for $a = 0$ to $N - 1$ **step** ρ **do** ▷ k-sample every ρ frames
  if $k = 1$ **then**
    $S \leftarrow \{a\}$
  else
    $S \leftarrow [\,]$ ▷ List of frame indices for this sample
    for $j = 0$ to $k - 1$ **do**
      $i_j \leftarrow \left(a + \left\lfloor \frac{j}{k-1} \cdot \Delta \right\rfloor\right) \bmod N$
      Append i_j to S
    end for
  end if
  $X_a \leftarrow \mathrm{Stack}\big(I_i \mid i \in S\big)$ ▷ Concatenate k frames into one element
  Append X_a to $\mathcal{D}$
end for
return $\mathcal{D}$

the Euclidean distance

$$\phi : Y \times Y \to \mathbb{R}^+; \quad \phi(y_1, y_2) = \sqrt{\big(\mathrm{pos}(y_1)_x - \mathrm{pos}(y_2)_x\big)^2 + \big(\mathrm{pos}(y_1)_y - \mathrm{pos}(y_2)_y\big)^2}.$$

All distances are normalized by $d_{\max} = \max_{y_1, y_2} \phi(y_1, y_2)$, so $\tilde{\phi}(y_1, y_2) = \phi(y_1, y_2)/d_{\max} \in [0, 1]$. When we want to emphasize avoiding nearby buildings as opposed to distant ones, we also use an inverse-normalized distance. For $\phi(y_1, y_2) > 0$ we set $\psi(y_1, y_2) = (1/\phi(y_1, y_2))/\max_{y_1', y_2'}(1/\phi(y_1', y_2'))$ and define $\psi(y_1, y_2) = 0$ when $\phi(y_1, y_2) = 0$. We then introduce a regularizing weight term

$$\lambda_\phi(\lambda; y_1, y_2) = \begin{cases} \lambda\, \tilde{\phi}(y_1, y_2), & \lambda \geq 0, \\ |\lambda|\, \psi(y_1, y_2), & \lambda < 0, \end{cases}$$

so positive λ scales the loss by the normalized distance, while negative λ scales it by the inverse-normalized distance, and $\lambda = 0$ removes the spatial contribution. Our direction-aware loss function then combines cross-entropy with this spatial term:

$$\ell_\theta(x_i, y_i) := \ell_{\mathrm{CE}}(f(x_i), y_i) + p(x_i)\, \lambda_\phi\big(\lambda; f(x_i), y_i\big),$$

and the overall objective is $\mathcal{L}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \ell_\theta(x_i, y_i)$.

From a learning standpoint, λ_ϕ turns our objective into an instance of cost-sensitive and distance-aware classification, where the penalty for predicting y_1 instead of the true label y_2 depends on a task-specific cost rather than a standard loss. With normal distance scaling ($\lambda > 0$), distant buildings receive larger penalties, encouraging the model to respect the global campus layout and avoid significant spatial errors, prioritizing being in a local neighborhood as opposed to exact classification. With inverse distance scaling ($\lambda < 0$), the loss instead emphasizes errors between nearby buildings, sharpening local decision boundaries in dense regions. This error-sensitive loss is closely related to standard cost-sensitive losses, particularly used when different mistakes have different severities (e.g., [7]); here the misclassification costs are explicitly induced by the campus geometry.

We report a few ablation studies in § 3, but ultimately adopt the following configuration as our default. We use a batch size of 64, train for 30 epochs, and set the weight decay to 1×10^{-4}. The learning rate is selected via a small grid search over $\{1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}\}$, with ViT-Large performing best at 1×10^{-4}, ViT-Tiny at 3×10^{-4}, and the ViT-Base and ViT-Small models using values in the range $[1 \times 10^{-4}, 3 \times 10^{-4}]$.

2.3 Deployment

For deployment, our goal is to run scene-aware localization on a mobile device, rendering the user's position inside a 3D campus environment without relying on GPS or network connectivity. We use publicly available GIS data to extract the approximate positions of the 41 target buildings. This information is then imported into Unity to construct a low-poly 3D map of the UCF campus, providing a lightweight but representational visual of the surroundings.

At runtime, the mobile client periodically captures a video frame (or a k-sampled bundle of frames) and feeds it through the fine-tuned ViT-Tiny model on the device. The predicted building label is mapped to its corresponding 3D location in the Unity scene, allowing us to place the user at the correct building on the virtual campus map. Because the model contains fewer than 6M parameters, inference can be performed locally with modest latency on modern smartphones. The process occurs as a periodic sanity check that the predicted location remains consistent.

Beyond localization, we use the same Unity representation and GIS geometry to support navigation between buildings. Each building is represented as a node in a 2D ground-plane graph, with edges induced by walkable

Figure 2: Low-poly campus model compared with real footage.

paths extracted from the turf provided from the data. Given a source and target building, we can compute routes either by minimizing Euclidean (ℓ_2) distances on the map or by approximating Manhattan distances along the campus walkways. The resulting path is displayed in the Unity scene, enabling users to both determine their current location and receive a suggested route to their destination without GPS.

3 Experimental Results

All experiments are conducted on NVIDIA H100 GPUs. We first evaluate the performance drop-off using the base preprocessing pipeline with $\lambda = 0$, to obtain a clean comparison between the ViT-Tiny, ViT-Small, ViT-Base, and ViT-Large models. Figure 3 and Table 1 show that accuracy is mostly stable between models; the accuracy drop-off is not significant for all k's except k=5. This suggests that the intrinsic dimension for constructing the geographical differences may be small enough to be captured by smaller models. This in turn makes mobile deployment promising, potentially promoting the use of even smaller models. The per-epoch training time is also similar across models, with ViT-Large being the only clear outlier.

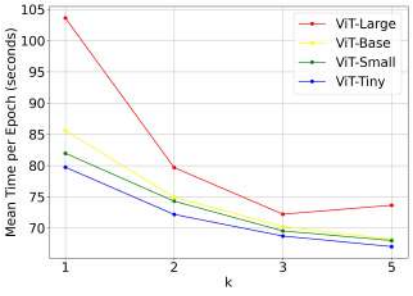

Figure 3: Mean training time per epoch on NVIDIA H100 GPU

Table 1: Accuracy varying model & k

Model	**k=1**	**k=2**	**k=3**	**k=5**
ViT-Large	98.60	98.64	98.18	97.19
ViT-Base	97.76	98.16	97.22	94.39
ViT-Small	97.32	98.00	97.10	95.59
ViT-Tiny	97.20	96.16	95.89	91.38

We next observe the effects on the models while varying the distance-penalty weight λ. We vary $\lambda \in \{-30, -10, -3, -1, -0.3, -0.1, 0.1, 0.3, 1, 3, 10, 30\}$ to study how changing λ may alter performance with regards to accuracy and the mean distance errors. Figure 4 reports test accuracy as a function of λ, while Figure 5 shows the mean geographical distance between incorrectly labeled predictions and their true labels.

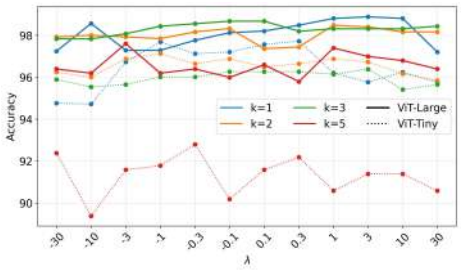

Figure 4: Test accuracy varying λ

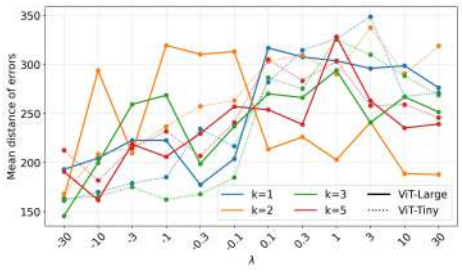

Figure 5: Average distance of errors

We highlight two main observations. First, we initially hypothesized that $\lambda \in \mathbb{R}^+$ would reduce the mean distance of misclassified examples, however in practice, it turns out that $\lambda \in \mathbb{R}^-$ caused the most significant drop. A plausible explanation is that the largest errors arise from buildings that are relatively close but not nearest neighbors in the 3D layout, so negative values of λ penalize these medium-range confusions more aggressively. Second, varying λ narrows the performance gap between ViT-Tiny and ViT-Large, underscoring that the distance-aware loss is well tailored to this geographical classification configuration. For deployment, we select ViT-Tiny with $k = 1$ and $\lambda = 0.3$, which achieves a test accuracy of 97.77%, representing a meaningful gain in a high-accuracy training setting.

In practice, users will capture buildings under diverse conditions, including changes in camera angle, lighting, and image resolution. To approximate this variability, we train on distorted images while varying both λ and k; the resulting accuracies are reported for ViT-Tiny in Table 2. Even under moderate distortions, the ViT-Tiny model maintains strong performance and continues to form reliable decision boundaries between many visually similar buildings.

3.1 On-Device Performance

We also analyze the cost of loading the model and performing inference on commodity smartphones. We observe highly efficient behavior where the

$k\backslash\lambda$	-30	-10	-3	-1	-0.3	-0.1	0	0.1	0.3	1	3	10	30
1	95.48	95.44	96.20	96.04	95.96	95.96	95.96	96.40	96.08	96.56	95.96	96.08	94.40
2	96.16	96.40	96.40	96.08	95.92	97.12	96.16	96.88	96.88	96.80	96.56	96.32	96.56
3	96.14	95.65	96.86	96.01	96.14	96.38	96.38	95.29	97.10	96.62	96.26	96.01	95.65
5	91.58	91.58	92.18	93.39	94.39	93.39	93.79	93.39	94.19	92.18	93.39	93.19	94.59

Table 2: Performance on distorted images for ViT-Tiny

model loading takes roughly 10 seconds, and a single forward pass requires about 0.1 seconds on average for an iPhone XIV. This enables practical periodic loops for continuous position confirmation and updates.

Taken together, these results show that a sub-6M-parameter model can match or closely approach the accuracy of much larger ViT variants while keeping spatial error low, even under image distortions. Combined with $\sim$0.1 second inference latency and $\sim$10 second model load time, this supports our choice of ViT-Tiny with $k = 1$ and $\lambda = 0.3$ as a practical configuration for real-time deployment.

4 Discussion and Future Work

4.1 Limitations

Our study has several limitations. First, we focus on a single campus and building-level labels, so it is unclear how well our k-sampling strategy and distance-aware loss transfer to other environments. Second, our dataset is predominantly recorded under favorable weather and lighting conditions. Third, we treat building coordinates as fixed 2D points and ignore elevation and detailed 3D structure. These constraints suggest caution when generalizing beyond our current deployment setting, motivating the extensions for future work discussed below.

4.2 Future Directions

We leave it up to future researchers to expand in an outdoor environment by collecting data through varying conditions such as weather fluctuations, day & night time changes, and seasonal variations. Additionally, we hope future researchers develop theoretical advancements in regards to our strategy.

We also believe that this approach has its greatest potential in settings where GPS is uninformative, particularly in inherently three-dimensional environments. Examples include different floors within large buildings, indoor corridors and atria, multi-level parking structures, and high-elevation

regions where precise vertical localization matters. Extending our method to incorporate floor-aware or height-aware labels, as well as richer 3D scene structure, is a promising direction for future deployment-oriented research.

Additionally, because this work is ultimately aimed at improving user experience, a natural next step is to conduct user oriented case studies that directly compare conventional GPS-based navigation with our 3D computer-vision system in terms of usability, perceived reliability, and overall satisfaction.

5 Conclusion

We have presented a lightweight, vision-based alternative to GPS for campus-scale localization that relies only on simple video input, a compact ViT-Tiny backbone, and a low-poly 3D map. Our k-sampling Algorithm 1 converts raw video into sequential training samples, allowing robust building-level classification with fewer than six million model parameters. Because inference can be performed entirely on-device, our approach is well-suited for densely built environments where GPS is noisy or unavailable, such as large university campuses, theme parks, stadiums, and compact city streets. With these developments, businesses and organizations could deploy inexpensive, higher-quality navigational tools using simple computer vision pipelines and inexpensive 3D assets.

Our prototype demonstrates that accurate, scene-aware localization is feasible on modern smartphones without network connectivity or specialized hardware, highlighting the practicality and affordability of this approach for users. At the same time, there remains substantial room for improvement such as extending beyond building-level recognition to finer-grained, location-aware models that distinguish floors, corridors, and elevation changes. We view this work as an initial step towards visual, GPS-free navigation systems that can operate across a wide range of real-world environments.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[3] Google LLC. Build global-scale, immersive, location-based ar experiences with the arcore geospatial API. `https://developers.google.com/ar/develop/geospatial`, 2023. Accessed 2025.

[4] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. arXiv:2110.02178.

[5] Qualcomm Technologies, Inc. The future of AI is hybrid: Unlocking the generative AI future with on-device and hybrid AI. `https://www.qualcomm.com/`, 2022. Whitepaper.

[6] Reply S.p.A. Visual positioning systems: Revolutionizing navigation and ar. `https://www.reply.com/en/3d-and-mixed-reality/visual-positioning-systems`, 2023. Accessed 2025.

[7] Ohad Volk and Gonen Singer. Adaptive cost-sensitive learning in neural networks for misclassification cost problems. *arXiv preprint arXiv:2111.07382*, 2021.