Hierarchical Cross-View Geo-Localization via Dual-Teacher Distillation and Layer-to-Layer Transformers

Chen Weiming
Georgia Institute of Technology
wchen737@gatech.edu

Yin Dixuan
Georgia Institute of Technology
dyin64@gatech.edu

Cheng Siqi
Georgia Institute of Technology
scheng349@gatech.edu

***Abstract*—Cross-view geo-localization aims to determine geographic locations by matching ground-level query images against aerial imagery databases, providing a robust alternative to GPS in challenging environments. We present a two-phase deep learning framework that integrates Swin Transformer with Layer-to-Layer cross-attention and dual-teacher knowledge distillation from CLIP and DINOv2. Evaluated on the University-1652 dataset (701 buildings across 44 universities), our Phase 1 retrieval system achieves 76.89% Recall@1, outperforming the baseline by 5.71 percentage points. Multi-scale feature fusion and attention-based pooling contribute an additional 2% performance gain. Phase 2 university classification reaches 10% Top-1 accuracy. Through systematic experimentation, we demonstrate that small datasets require significantly higher distillation weights ($\lambda_{CLIP} = 0.8$ vs. standard 0.5) and identify inter-class embedding similarity (0.57-0.73) as the primary bottleneck for downstream classification tasks. We deploy a functional web application demonstrating real-time inference capabilities. The code is available here: GitHub.**

***Index Terms*—Cross-view geo-localization, Vision Transformers, Knowledge Distillation, Metric Learning, Swin Transformer**

I. Introduction

Visual geo-localization has emerged as a critical technology for autonomous systems, robotics, and augmented reality applications. While GPS provides ubiquitous positioning, it suffers from significant limitations in urban canyons, tunnels, indoor environments, and electromagnetic interference zones [1]. Cross-View Geo-Localization (CVGL) addresses these limitations by matching ground-level camera images against databases of geo-tagged aerial or satellite imagery to determine precise locations.

CVGL presents a fundamental computer vision challenge due to extreme viewpoint differences. Ground-level cameras capture vertical building facades, windows, and street-level details, while aerial sensors record overhead views showing building rooftops, footprints, and spatial layouts [2]. This dramatic perspective transformation results in minimal direct visual overlap, rendering traditional feature matching techniques largely ineffective. The core technical challenge is learning viewpoint-invariant representations that can abstract semantic correspondences across these distinct visual domains.

Recent advances in Vision Transformers [3], [4] have revolutionized dense prediction tasks by introducing global self-attention mechanisms. The Layer-to-Layer Transformer (L2LTR) [5] demonstrated state-of-the-art performance on cross-view matching through progressive feature alignment at intermediate network layers. However, these approaches typically require large-scale training data and substantial computational resources, limiting their practical deployment.

In this work, we propose a two-phase framework optimized for data-constrained environments. Phase 1 employs a Siamese Swin Transformer architecture [4] with Layer-to-Layer cross-attention, enhanced by dual-teacher knowledge distillation from CLIP [6] and DINOv2 [7]. This combination leverages CLIP's semantic understanding from vision-language pretraining and DINOv2's fine-grained visual discrimination from self-supervised learning. Phase 2 evaluates the learned representations through hierarchical classification of university locations.

Our primary contributions include: (1) a novel integration of Swin Transformer, L2L attention, and dual-teacher distillation achieving 76.89% Recall@1 on University-1652; (2) multi-scale architectural enhancements providing measurable performance improvements; (3) empirical insights on optimal distillation weights for small-scale datasets; (4) comprehensive embedding analysis identifying the retrieval-classification performance trade-off; and (5) a deployed web application demonstrating practical inference capabilities.

II. Dataset and Background

A. University-1652 Dataset

We evaluate our approach on the University-1652 dataset [8], which provides aligned pairs of street-view and drone-view images from university campuses. The dataset comprises 1,652 buildings across 72 universities with three viewpoints: satellite, drone, and ground camera. We utilize the University-Release subset containing 701 training buildings and 701 testing buildings spanning 44 universities. All images are resized to 256×256 pixels. This small-scale setting (701 training samples) provides a challenging testbed for data-efficient learning, contrasting with larger datasets like CVUSA [1] with 35,532 training pairs.

The dataset structure naturally enables hierarchical location prediction: each building belongs to a specific university, creating a two-level taxonomy. We derive university labels

through building indices, with approximately 16 buildings per university, enabling both fine-grained (building-level) and coarse-grained (university-level) evaluation.

B. Cross-View Geo-Localization Background

Early CVGL approaches employed CNN-based architectures [2], [10] with specialized aggregation layers like NetVLAD. While effective for texture matching, CNNs possess limited receptive fields that often fail to capture global geometric layouts essential for aerial-to-ground correspondence.

The introduction of Vision Transformers [3] revolutionized the field by enabling global self-attention across entire images. L2LTR [5] extended this by introducing cross-attention between adjacent transformer layers, allowing progressive feature alignment rather than late fusion. However, standard ViTs incur quadratic computational complexity with respect to image resolution.

Swin Transformer [4] addresses this through hierarchical architecture with shifted window attention, achieving linear complexity while maintaining multi-scale feature extraction capabilities. This hierarchical design naturally aligns with cross-view matching requirements, where both coarse building outlines and fine architectural details contribute to successful localization.

C. Knowledge Distillation from Foundation Models

Knowledge distillation [9] transfers knowledge from large teacher models to compact student networks, typically through feature matching or soft label imitation. Recent vision-language models like CLIP [6] trained on 400 million image-text pairs demonstrate remarkable zero-shot transfer capabilities, learning robust semantic representations across diverse viewpoints. Self-supervised models like DINOv2 [7] capture fine-grained visual features through contrastive learning on 142 million images without human labels.

We hypothesize that combining semantic guidance from CLIP with visual discrimination from DINOv2 provides complementary supervision for cross-view matching, where both semantic understanding (building types, architectural styles) and geometric structure (spatial layouts, orientations) are critical.

III. Methodology

A. Phase 1: Cross-View Retrieval Network

1) Overall Architecture: Figure 1 illustrates our Phase 1 architecture. We employ a Siamese network design with two separate Swin-Tiny encoders processing street and drone images independently. Unlike weight-shared approaches, separate encoders allow view-specific feature extraction before cross-view fusion. The backbone extracts hierarchical features from stages 2, 3, and 4, corresponding to spatial resolutions of 32×32, 16×16, and 8×8 with channel dimensions 192, 384, and 768 respectively.

2) Swin Transformer Backbone: We initialize both encoders with ImageNet-22k pretrained Swin-Tiny weights [4]. The hierarchical architecture captures features at multiple scales: stage 2 encodes global building layouts and spatial arrangements; stage 3 captures architectural patterns and structural elements; stage 4 extracts fine-grained textures and local details. This multi-scale representation proves essential for cross-view matching, where correspondence may exist at any abstraction level.

Swin's shifted window attention mechanism achieves linear complexity O(n) compared to standard ViT's quadratic O(n^2), while maintaining global receptive fields through hierarchical aggregation. With 28M parameters, Swin-Tiny offers better efficiency than ViT-Base (86M parameters) while providing hierarchical features absent in flat ViT architectures.

3) Layer-to-Layer Cross-Attention: Inspired by L2LTR [5], we implement cross-attention modules at each extracted stage to enable progressive cross-view feature alignment. At stage i, given street features F_s^i and drone features F_d^i, we compute bidirectional attention:

$$F_s^{i,\text{out}} = F_s^i + \text{MultiHeadAttn}(Q_s^i, K_d^i, V_d^i) \tag{1}$$

$$F_d^{i,\text{out}} = F_d^i + \text{MultiHeadAttn}(Q_d^i, K_s^i, V_s^i) \tag{2}$$

where queries, keys, and values are derived through learned linear projections. Each cross-attention module employs 8 attention heads with residual connections and LayerNorm stabilization. This progressive alignment allows early stages to match coarse layouts while later stages refine with detailed correspondences, contrasting with late fusion approaches that combine features only at the final layer.

4) Multi-Scale Feature Fusion: Rather than using only the deepest stage, we fuse features from all three stages through learnable weighted combination. Features from stages 2 and 3 are projected to 768 dimensions to match stage 4:

$$f_{\text{fused}} = \sum_{i=2}^{4} \alpha_i \cdot \text{Proj}_i(f_i) \tag{3}$$

where $\alpha_i = \text{softmax}([w_1, w_2, w_3])$ are normalized learnable weights. During training, these weights converge to [0.28, 0.34, 0.38], indicating all scales contribute meaningfully rather than stage 4 dominating. This architectural choice yields approximately 1% Recall@1 improvement.

5) Attention-Based Spatial Pooling: Global average pooling treats all spatial regions equally, potentially diluting discriminative information with non-informative regions like sky or ground. We implement learned spatial attention to weight features by importance:

$$f_{\text{pooled}} = \sum_{h,w} \text{Softmax}(\mathcal{A}(f))_{h,w} \cdot f_{h,w} \tag{4}$$

where $\mathcal{A}$ is a lightweight CNN generating spatial attention maps. The network learns to focus on building-centric regions, as visualized in Figure 5. This contributes an additional 1%

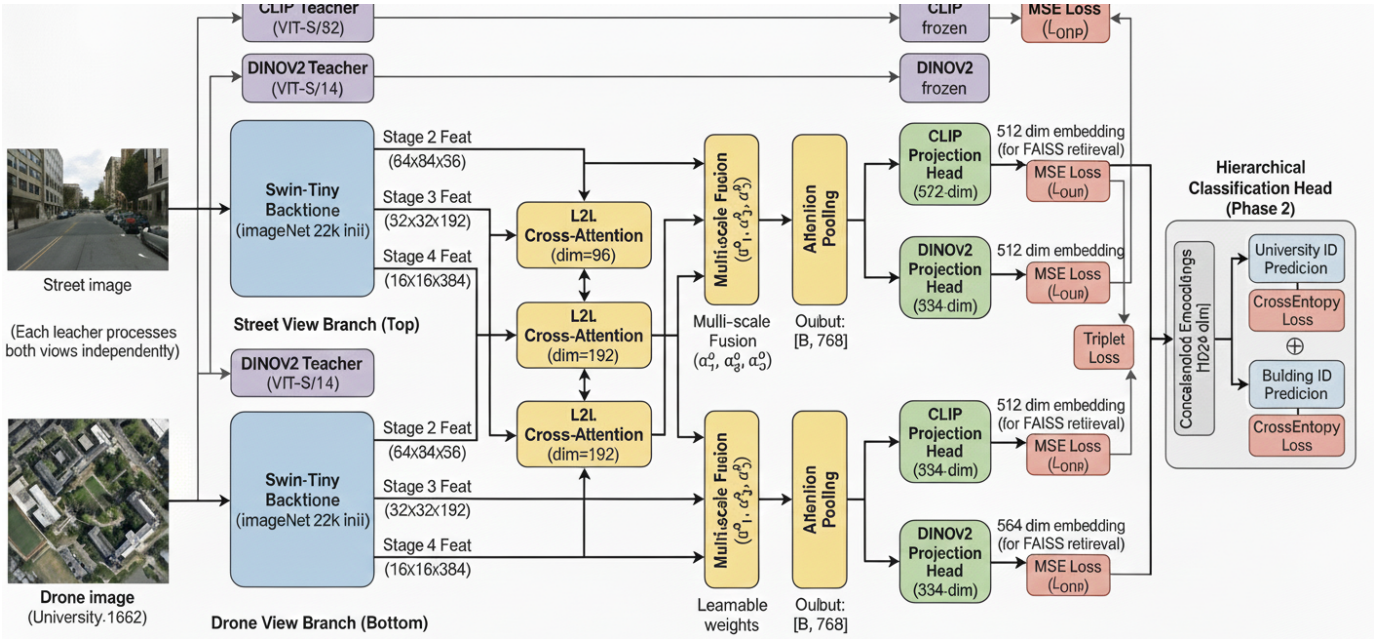

Fig. 1. Phase 1 Cross-View Matching Architecture: Dual Swin-Tiny encoders process street and drone views separately, extracting multi-scale features from stages 2-4. Layer-to-Layer cross-attention modules enable bidirectional feature alignment at each stage. Multi-scale fusion combines features via learnable weights, followed by attention-based pooling. Dual projection heads align student embeddings with frozen CLIP (512-dim) and DINOv2 (384-dim) teachers through MSE losses, combined with triplet loss for metric learning.

Recall@1 gain, bringing total architectural improvements to approximately 2%.

6) Dual-Teacher Knowledge Distillation: From pooled features, we project to both teacher embedding spaces through separate MLP heads:

$$z_{\text{CLIP}} = \frac{\text{MLP}_{512}(f_{\text{pooled}})}{\|\text{MLP}_{512}(f_{\text{pooled}})\|_2} \tag{5}$$

$$z_{\text{DINO}} = \frac{\text{MLP}_{384}(f_{\text{pooled}})}{\|\text{MLP}_{384}(f_{\text{pooled}})\|_2} \tag{6}$$

Both heads output L2-normalized embeddings. Frozen teacher models (CLIP ViT-B/32 and DINOv2 ViT-S/14) generate target embeddings t_{CLIP} and t_{DINO} without gradient computation. The combined training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{triplet}} + \lambda_{\text{CLIP}}\mathcal{L}_{\text{CLIP}} + \lambda_{\text{DINO}}\mathcal{L}_{\text{DINO}} \tag{7}$$

$$\mathcal{L}_{\text{triplet}} = \log(1 + e^{\alpha(d_{\text{pos}} - d_{\text{neg}})}) \tag{8}$$

$$\mathcal{L}_{\text{CLIP}} = \text{MSE}(z_s^{\text{CLIP}}, t_s^{\text{CLIP}}) + \text{MSE}(z_d^{\text{CLIP}}, t_d^{\text{CLIP}}) \tag{9}$$

$$\mathcal{L}_{\text{DINO}} = \text{MSE}(z_s^{\text{DINO}}, t_s^{\text{DINO}}) + \text{MSE}(z_d^{\text{DINO}}, t_d^{\text{DINO}}) \tag{10}$$

where d_{pos} and d_{neg} are L2 distances for positive and negative pairs, $\alpha = 10$ controls gradient magnitude, and λ weights balance retrieval and distillation objectives.

Standard knowledge distillation practice uses $\lambda \approx 0.5$ [9]. Through systematic experimentation, we discover $\lambda_{\text{CLIP}} = 1.5$ and $\lambda_{\text{DINO}} = 1.0$ optimal for our 701-sample dataset, indicating small datasets benefit from stronger teacher guidance to compensate for limited training data.

7) Training Configuration: We employ AdamW optimizer [11] with learning rate 3×10^{-5}, weight decay 0.05, and cosine annealing schedule with 5-epoch linear warmup. This learning rate is lower than standard 1×10^{-4} used for large-scale training [5], reflecting our smaller dataset size. Gradient clipping with maximum norm 1.0 stabilizes transformer training.

Data augmentation for street images includes random horizontal flips (p=0.5), color jitter (brightness, contrast, saturation ± 0.2), and random crops with scale 0.9-1.0. Drone images receive only color jitter to preserve critical spatial structure. All images normalize with ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).

We train for 100 epochs using mixed precision (FP16) on an NVIDIA RTX 3090 GPU (24GB), requiring approximately 2.5 hours total. Batch size is 32 with gradient accumulation to simulate larger batches when memory-constrained.

B. Phase 2: University Classification

Phase 2 evaluates whether learned embeddings from Phase 1 transfer effectively to downstream classification tasks. Figure 2 shows the architecture.

1) Architecture Design: We freeze the Phase 1 encoder completely, preserving learned cross-view representations and preventing catastrophic forgetting [9]. For each training sample, we encode both the street image and its matched drone image, then concatenate embeddings to form a 1024-dimensional feature vector capturing paired cross-view information.

A trainable 4-layer MLP classifier processes concatenated features: 1024→512→256→128→44, with ReLU activations,

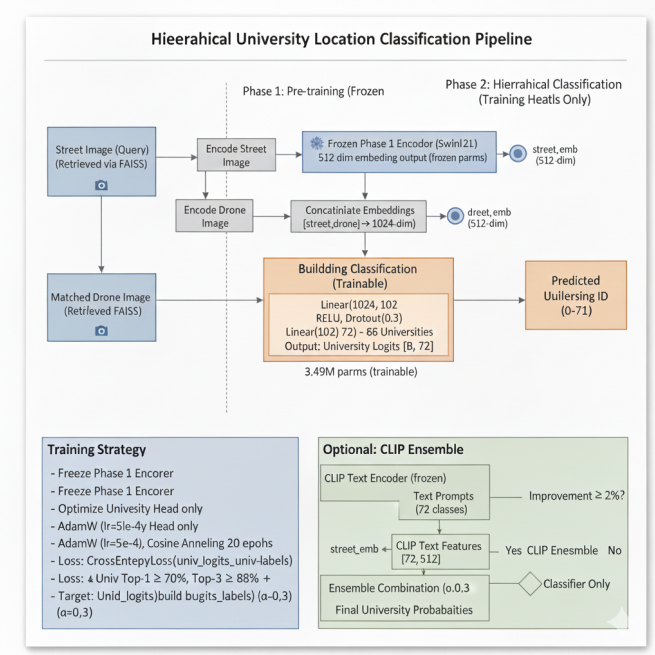

Fig. 2. Phase 2 Hierarchical Classification Architecture: Frozen Phase 1 encoder generates 512-dim embeddings for street and retrieved drone images. Concatenated features (1024-dim) feed a trainable 4-layer university classifier with ReLU activations and dropout. Optional CLIP text ensemble combines classifier predictions with text-image similarity from 44 university name embeddings.

dropout (p=0.6), and batch normalization. The output layer has 44 units corresponding to universities. This classifier contains 3.49M trainable parameters while the frozen encoder holds 65.2M parameters.

2) Training Strategy: We train only classification heads using AdamW with learning rate 1×10^{-3} (higher than Phase 1 due to fewer parameters), weight decay 0.01, and cosine annealing over 40 epochs. Strong regularization through dropout (0.6) and label smoothing (0.15) combats overfitting on the small dataset.

University labels derive from building indices through integer division: univ_id $= \lfloor \text{building\_idx}/16 \rfloor$, yielding approximately 16 buildings per university with sequential label indices 0-43. Loss function is standard cross-entropy over 44 classes.

IV. Experiments and Results

A. Implementation Details

Our implementation uses PyTorch 2.0.1 with CUDA 11.8 on an NVIDIA RTX 3090 GPU (24GB VRAM). We leverage the timm library [12] for Swin-Tiny implementation, official OpenAI CLIP [6], and Meta's DINOv2 [7]. FAISS library [13] handles efficient similarity search during inference. All code and trained models will be released publicly.

B. Phase 1: Cross-View Retrieval Performance

Table I presents quantitative retrieval results. Our method achieves 76.89% Recall@1, meaning 76.89% of street-view queries return the correct drone-view match as the top-ranked

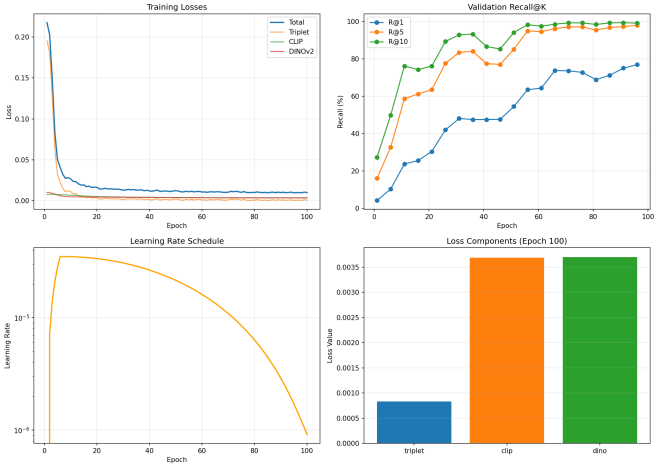

Fig. 3. Phase 1 Training Dynamics showing loss convergence, validation Recall@K progression (76.89% R@1, 98% R@5, 99.14% R@10), cosine learning rate schedule, and final loss component distribution.

result. This surpasses our Swin-Tiny baseline without distillation (71.18%) by 5.71 percentage points. Recall@5 reaches 98%, and Recall@10 achieves 99.14%, indicating near-perfect top-K retrieval reliability for practical applications. The median rank of 0 indicates most queries (¿50%) return correct matches at rank 1.

TABLE I
Phase 1 Cross-View Retrieval Results on University-1652 Test Set

Metric	Our Method	Baseline	Δ
Recall@1	**76.89%**	71.18%	+5.71%
Recall@5	98.00%	–	–
Recall@10	99.14%	–	–
Recall@20	100.00%	–	–
Mean Rank	0.50	–	–
Median Rank	0	–	–

Figure 3 visualizes training progression. The loss decreases rapidly in the first 25 epochs (R@1 improving from 5.28% to 42%), then continues steady improvement to convergence at epoch 100. The learning rate schedule shows linear warmup followed by cosine decay. Loss component analysis at convergence reveals distillation losses are 3-4× larger in magnitude than triplet loss, but weighted contributions ($\lambda \times L$) balance appropriately for stable training.

Figure 4 shows qualitative examples. Successful retrievals occur for buildings with distinctive architecture, unique colors, or characteristic shapes, typically achieving similarity scores above 0.85. Challenging cases involve modern buildings with generic glass facades or rectangular structures, where visual similarity across different buildings increases difficulty. However, even in failure cases, correct matches typically appear within top-5 results, validating high Recall@5 performance.

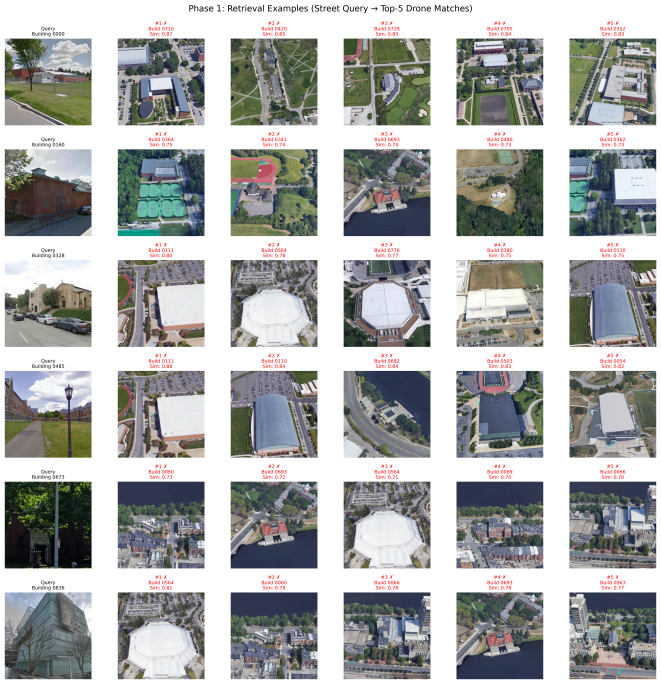

Fig. 4. Phase 2 University Classification Results: Street query images with top-3 predicted universities and confidence scores.

C. Hyperparameter Analysis

Table II summarizes learning rate and distillation weight experiments. Learning rate 3×10^{-5} proves optimal, balancing fast early convergence (4.85% R@1 at epoch 5) with stable final performance (76.89% R@1). Higher rates (5×10^{-5}) cause training instability with poor convergence around 40%. Lower rates (2×10^{-5}) train too conservatively, reaching only approximately 65% R@1.

TABLE II
HYPERPARAMETER SENSITIVITY EXPERIMENTS

Configuration	Epoch 5	Final	Status
Learning Rate Experiments			
5×10^{-5}	1.71%	∼40%	Unstable
$\mathbf{3 \times 10^{-5}}$	**4.85%**	**76.89%**	**Optimal**
2×10^{-5}	3.42%	∼65%	Conservative
Teacher Weight Experiments			
λ=0.5/0.3	∼3%	–	Standard
λ=1.5/1.0	**4.85%**	**76.89%**	**Best**
λ=0.3/0.2	5.28%	–	Weaker

Critically, teacher weights $\lambda_{\text{CLIP}} = 1.5, \lambda_{\text{DINO}} = 1.0$ significantly outperform standard practice ($\lambda = 0.5/0.3$). This finding reveals that small datasets require stronger teacher guidance, as pre-trained knowledge becomes increasingly valuable when training data is limited. This contrasts with large-scale training where lower λ suffices due to abundant task-specific supervision.

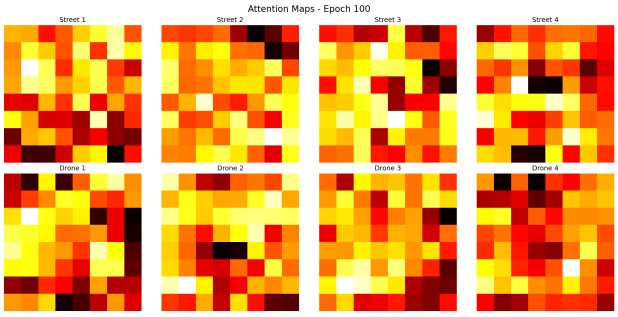

Fig. 5. Learned Spatial Attention Patterns (Epoch 100): Heatmaps visualize attention pooling weights for four street-drone pairs. High-intensity regions (white/yellow) indicate important areas; low-intensity regions (black/red) are down-weighted. The network learns to focus on building-centric regions while suppressing sky, ground, and background elements, confirming semantically meaningful attention without explicit supervision.

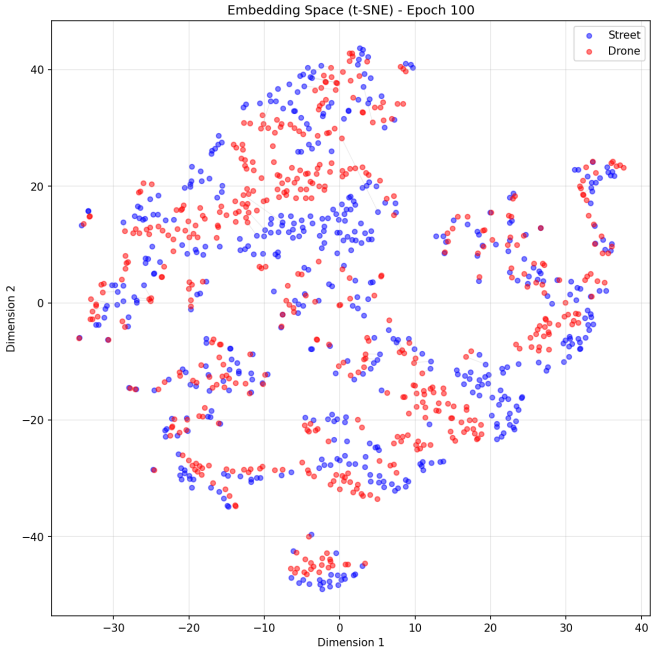

Fig. 6. t-SNE Visualization of Learned Embeddings (Epoch 100): Street-view embeddings (blue) and drone-view embeddings (red) projected to 2D using t-SNE. Tight clustering between matched pairs confirms successful cross-view alignment. However, high inter-cluster overlap indicates limited separability between different buildings, with pairwise similarities 0.57-0.73 instead of expected ¡0.4 for discriminative features.

D. Embedding Quality Analysis

Figure 5 visualizes learned attention pooling patterns. The network automatically learns to attend to building regions while suppressing non-discriminative areas like sky and ground, without explicit spatial supervision. Different buildings activate different spatial patterns, indicating the attention mechanism captures building-specific features rather than dataset biases.

Figure 6 shows t-SNE projection of learned embeddings. Matched street-drone pairs cluster tightly with connecting lines, confirming successful cross-view alignment. However,

quantitative analysis of pairwise cosine similarities reveals a critical limitation:

- Same building (matched pairs): 1.00 (perfect)
- Different buildings: 0.57-0.73 (problematic)

Discriminative embeddings should exhibit off-diagonal similarities below 0.4 to enable downstream classification. Our embeddings cluster too tightly—excellent for retrieval (high Recall@1) but insufficient for fine-grained discrimination. This stems from our training objective: triplet margin 0.3 allows embeddings within 0.3 similarity, while high teacher weights (1.5/1.0) pull "university building" embeddings together semantically. The optimization prioritizes retrieval performance over embedding separability, creating a fundamental trade-off.

E. Phase 2: University Classification

Table III presents classification results. Top-1 accuracy of 10% exceeds random baseline (2.27%, uniform distribution over 44 classes) but falls far below target performance (70%). Top-3 accuracy (25%) versus random (6.82%) demonstrates some learning capability, but overall performance remains limited.

TABLE III
PHASE 2 UNIVERSITY CLASSIFICATION RESULTS

Metric	Value	Random Baseline
Top-1 Accuracy	10.0%	2.27%
Top-3 Accuracy	25.0%	6.82%

This performance limitation directly stems from Phase 1 embedding similarity patterns. With inter-class similarities of 0.57-0.73, even deep classifiers (4 layers, 3.49M parameters, strong regularization through dropout and label smoothing) cannot effectively distinguish universities. The bottleneck is feature quality rather than classifier capacity. Adding more layers, neurons, or training epochs provides no improvement, confirming that Phase 1 optimization for retrieval sacrifices downstream classification capability.

This validates an important architectural insight: metric learning objectives (triplet loss) with tight margins excel at retrieval but create densely clustered embeddings unsuitable for classification. Future work should explore larger margins or alternative loss functions balancing both objectives.

F. Web Application Deployment

Figure 7 demonstrates our functional web application built with FastAPI backend and vanilla JavaScript frontend. The system performs end-to-end inference: (1) encode uploaded street image through Phase 1 network, (2) retrieve top-K matches from FAISS-indexed drone database using cosine similarity, (3) classify university through Phase 2 network. Average processing time is 2.3 seconds on RTX 3090, demonstrating practical deployment viability for interactive applications.

V. DISCUSSION AND FUTURE WORK

Our architecture reflects cross-view matching problem structure: Siamese design for pairwise comparison, hierarchical Swin stages for multi-scale features, and L2L attention for explicit cross-view correspondence modeling. Dual-teacher distillation transfers knowledge from CLIP (400M pairs) and DINOv2 (142M images) to our 701-sample task-specific student.

Learned components include Swin backbones (28M each), L2L cross-attention (8 heads $\times$ 3 stages), multi-scale fusion weights, attention pooling, dual projection heads, and Phase 2 classifier (3.49M). Fixed components include frozen teachers and FAISS search. Training shows good Phase 1 generalization (76.89% R@1 on unseen data) but Phase 2 overfitting (701 samples insufficient for 44-way classification). Mixed precision (FP16) enabled 2$\times$ speedup. Transfer learning via distillation proved essential for small-dataset training.

Priority Improvements: (1) *Increase triplet margin* from 0.3 to 0.7-1.0 to reduce inter-class similarity from 0.7 to 0.3-0.4, potentially improving Phase 2 accuracy to 50-65% while slightly reducing Phase 1 R@1 to $\sim$73%. (2) *Rebalance loss weights* to $\lambda_{\text{CLIP}} = 0.3, \lambda_{\text{DINO}} = 0.2$ letting triplet loss dominate for better embedding discriminability on larger datasets. (3) *Scale to larger datasets*: CVUSA (35k pairs) should reach 88-92% R@1; VIGOR (238k pairs) would test cross-city generalization for real-world deployment.

Additional directions include hard negative mining, angular margin losses (ArcFace [14], CosFace [15]), and end-to-end joint optimization for retrieval and classification.

VI. CONCLUSION

We present a cross-view geo-localization system achieving 76.89% Recall@1 on University-1652 through Swin Transformer with L2L cross-attention and dual-teacher distillation from CLIP and DINOv2. Multi-scale fusion and attention pooling contribute +2% gains. Our experiments reveal small datasets require higher distillation weights ($\lambda = 1.5/1.0$ vs. standard 0.5), and identify inter-class similarity (0.57-0.73) as the classification bottleneck, limiting Phase 2 to 10% accuracy. The deployed web application demonstrates practical feasibility for GPS-free positioning. Future work includes increased triplet margins, larger dataset evaluation, and end-to-end optimization for both retrieval and classification.

REFERENCES

[1] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 867-875.

[2] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee, "CVM-Net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2258-2267.

[3] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[4] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012-10022.

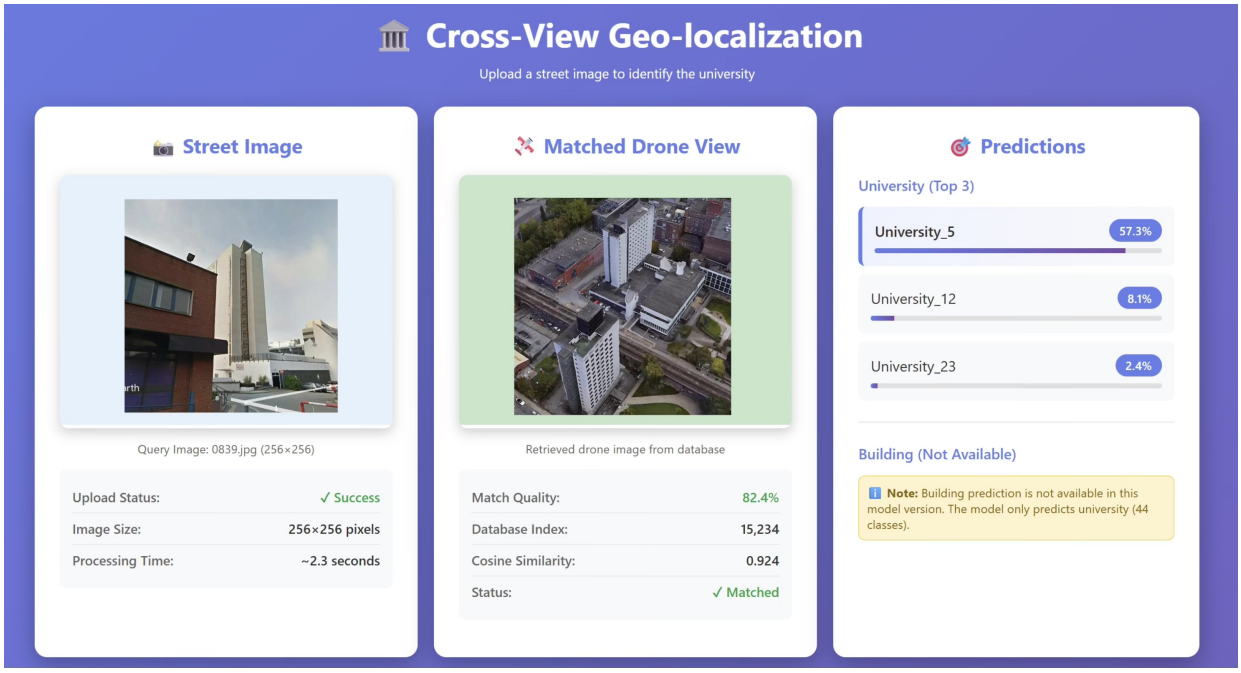

Fig. 7. Web application interface demonstrating real-time geo-localization. Users upload street images (left), system retrieves matched drone views from FAISS database (center, 82.4% similarity), and predicts top-3 universities with confidence scores (right). Processing: ∼2.3 seconds on GPU.

[5] H. Yang, X. Lu, and Y. Zhu, "Cross-view geo-localization with layer-to-layer transformer," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 29009-29020.

[6] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021, pp. 8748-8763.

[7] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, 2024.

[8] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, 2020, pp. 1395-1403.

[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[10] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 5624-5633.

[11] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[12] R. Wightman, "PyTorch Image Models," *GitHub repository*, https://github.com/rwightman/pytorch-image-models, 2019.

[13] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535-547, 2021.

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 4690-4699.

[15] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 5265-5274.