GeoVision: Fine-Grained Urban Geolocation in San Francisco via Distribution-Aware Visual Models

Mathijs Ammerlaan mathijs@stanford.edu
Nils Kuhn nfkuhn@stanford.edu
Raúl Molina Gómez rmolinag@stanford.edu

Abstract

We address the task of fine-grained image-based geolocation within a single city. Focusing on San Francisco, we evaluate two approaches: (1) a probabilistic model that outputs a mixture of spatial Gaussian distributions and (2) a grid-based classifier that discretizes the city into a fine spatial mesh. Both are built on top of vision transformers: either the pre-trained StreetCLIP model or a custom ViT trained from scratch. Trained on over 70,000 street-level images, our models achieve localization accuracy within 600 meters on average. While the grid-based classifier offers higher top-1 accuracy, the Gaussian model provides richer uncertainty estimates. We further analyze model behavior using attention rollout techniques, showing that pre-training enables the network to focus on meaningful geographic cues such as building façades and road features. Our results highlight the feasibility of lightweight, image-only localization in dense urban environments and the complementary strengths of classification and distribution-based approaches.

1. Introduction

Accurately geo-localizing street-level images is a long-standing challenge in computer vision, traditionally tackled at global scales. However, fine-grained localization within a single city remains underexplored and presents unique difficulties. Urban environments often contain highly repetitive visual elements, such as similar-looking buildings, tree-lined streets, or traffic signs, making precise geo-localization within a dense cityscape like San Francisco a non-trivial task. Solving this problem requires models capable of identifying subtle, location-specific visual details that distinguish one neighborhood from another.

In this project, we tackle the task of high-resolution image-based localization within the city of San Francisco. Using a custom dataset of over 70,000 street-level images collected via the Mapillary API [8] (each annotated with precise GPS coordinates) we train and evaluate two types of neural network models for local-scale image geo-localization:

1. A Gaussian regression model, which predicts k 2D Gaussian distributions over latitude and longitude coordinates, each one weighted with the probability of the mean to be the actual location of the image. We experiment with data augmentation and compare performance using both a custom Vision Transformer (ViT) and a pre-trained StreetCLIP model.
2. A grid-based classification model, which discretizes the San Francisco area into a 31×31 spatial grid (961 cells), and classifies each input image into one of these fine-grained spatial tiles. This model also uses StreetCLIP and benefits from data augmentation to improve generalization.

Beyond model training and evaluation, we conduct detailed prediction analysis and explore the internal decision-making process of our models. Using attention rollout techniques, we visualize the attention maps of our transformer-based architectures to better understand which image regions are driving the localization predictions. This analysis reveals that models often focus on distinctive elements such as unique building facades, signage, and street layouts, validating their potential to learn localized geographic priors.

This work contributes a lightweight, image-only solution for high-resolution urban localization, with potential applications in:

- Reconstructing the location of untagged or historical images (e.g., vacation photos).
- Forensic analysis of crime scene imagery.
- Enhancing autonomous driving systems in environments with poor or unavailable GPS signals.

By comparing probabilistic regression and classification-based approaches, and analyzing model attention patterns, we demonstrate the feasibility and complementary strengths of different techniques for city image localization. Our results suggest that while classification offers speed and simplicity, Gaussian models provide richer spatial uncertainty,

offering a valuable tradeoff depending on the application's needs.

2. Related Work

2.1. Global Image Geolocation

Early approaches to image geolocation, such as Im2GPS [7], framed the problem as an image retrieval task: a query image was matched to a large database of geotagged images using hand-crafted visual features like GIST or SIFT. Although effective at the time, these methods struggled with generalization and scalability. The introduction of deep learning brought significant advances, PlaNet [14] proposed treating geolocation as a classification problem over thousands of discrete geographic cells, using convolutional neural networks (CNNs) to directly predict location classes. This significantly improved robustness and allowed end-to-end learning. Later, CPlaNet [13] extended this idea by incorporating combinatorial partitioning, refining the spatial resolution of predictions without exponentially increasing the number of output classes. Despite these advances, most global models still localize at the city or country level, lacking precision for urban-scale applications.

2.2. City-Scale and Local Place Recognition

While global-scale geolocation has seen extensive research, urban or city-scale localization remains more challenging and less explored. The high visual similarity between different locations within a city (e.g., repeating architecture or vegetation) makes precise localization harder. NetVLAD [2] proposed a CNN-based architecture for place recognition using weak supervision, relying on triplet loss and a trainable VLAD layer to create compact and discriminative global descriptors. This model became a foundational method for urban localization tasks and inspired many retrieval-based approaches. Meanwhile, datasets such as StreetLearn [9] have enabled research into realistic, city-scale navigation, offering panoramic views and GPS data across urban areas. These benchmarks promote tasks like loop closure detection, route planning, and vision-based geolocation within a constrained map.

2.3. Vision-Language Models for Geolocation

Recent work has explored the use of vision-language models for geolocation. CLIP [11] learns joint representations of images and text through contrastive pretraining on 400 million image-text pairs, providing strong zero-shot generalization. Building on this, StreetCLIP [5] fine-tunes CLIP for geo-grounded tasks by training on 1.1 million street-level images annotated with GPS coordinates. By synthesizing text prompts that describe geographic context (e.g., "a street in downtown Tokyo"), StreetCLIP aligns images with location-aware textual embeddings, enabling fine-grained localization across diverse environments. These models benefit from broad pretraining and offer semantic reasoning that complements traditional geometric approaches, making them highly promising for urban-scale localization tasks like ours.

3. Methodology

3.1. CLIP-Based Visual Encoder

To address the geolocalization task, we employed StreetCLIP, a robust foundation model tailored for open-domain image geolocalization and other geography-related tasks. StreetCLIP is built upon OpenAI's CLIP (Contrastive Language–Image Pre-training) model, specifically the ViT-L/14 architecture, which utilizes Vision Transformers with 14x14 pixel patches and processes images resized to 224×224 pixels. The original CLIP model was trained on a large dataset of 400 million image-text pairs, enabling it to learn a wide range of visual concepts from natural language supervision [10].

StreetCLIP adapts this architecture by fine-tuning it on a dataset of 1.1 million geo-tagged street-level images from 101 countries, encompassing both urban and rural scenes. To align the model with geolocalization tasks, synthetic captions were generated from image class labels using a domain-specific caption template. This approach allows StreetCLIP to transfer its generalized zero-shot learning capabilities to the specific domain of image geolocalization [6].

In our implementation, we utilize the vision encoder component of StreetCLIP, which processes input images and returns a pooled visual embedding via its `pooler_output`. This embedding serves as the input to our custom classification or density-estimation heads. We do not modify the architecture of the encoder itself, allowing us to benefit directly from the robust geographical priors learned during pretraining. In subsequent sections, we describe how we build on top of this backbone for both our grid-based classifier and our probabilistic Gaussian output model.

3.2. Gaussian-Based Classification

The first prediction method that we proposed to be useful for geolocalization, was a Mixture Density Network (MDN) [3]. The idea behind this is that the model should predict a probability distribution over San Francisco. Specifically in cases where more images could come from more then one distant positions, we wanted the model to be able to give a high probability for all of the locations. A simple example for such an image would be a picture of a forest without any signs or other specific features. This image could be shot in multiple forests possibly far away from each other. Predicting the mean location of the forests would not be

meaningful. Multiple Mixture Density Networks solve this issue by using multiple Gaussian curves, where the model can predict each mean and standard deviation of the Gaussian curves, together with a weight which is used to weight the individual Gaussian. In theory, this could lead to a much more understandable and intuitive prediction. For the training, we used the Mixture Density Network loss, which is the negative logarithmic likelihood of the true coordinates. This Gaussian-based classification was used in three different model and each model was predicting and weighting six Gaussians. One without data augmentation, one with data augmentation and a third Custom ViT which didn't use the pretrained StreetCLIP model, but also used data augmentation. We will explain the architecture in 3.4. Custom ViT.

3.3. Grid-Based Classification

The second proposed method is a classification-based approach that divides the San Francisco area into a 31×31 spatial grid, resulting in 961 distinct tiles (classes). Each image is classified into one of these tiles. This method uses the StreetCLIP transformer model, replacing its original projection head with a custom multilayer perceptron (MLP). The new head consists of a linear layer with 512 hidden units, followed by a normalization layer, a ReLU activation function, and a final output layer that produces a score vector over the 961 classes. In this case, we focus exclusively on one prediction (the highest probability), although for validation purposes we can check the highest k probabilities to increase accuracy of the model. The loss function used for this model is the Cross Entropy Loss, which performs well for classification tasks and consistently delivers strong performance in similar settings. The model was also trained using data augmentation techniques to avoid overfitting and improve generalization.

The motivation behind this approach is to simplify the learning of a continuous output space by framing the task as a classification problem. This discrete formulation offers a more straightforward training procedure and may outperform the Gaussian-based model in terms of prediction accuracy and speed. However, the Gaussian approach provides a richer representation of uncertainty, which may be advantageous in scenarios where multiple regions in the city present high likelihoods (e.g., parks, the wharf or other ambiguous areas).

3.4. Custom ViT

To check how important the pretraining is for the geolocalization task, we trained our own custom ViT from scratch. The ViT takes a similar 224x224 image input and does not utilize the preprocessor of the StreetCLIP model. The architecture is a reduced version of the CLIP model. It has 12 transformer layers and 16 heads in each of them. (Clip uses 24 layers and also 16 heads.) The dimension of all tokens is 1024 (similar to CLIP) and the hidden dimension of the FFN is 2048. (CLIP uses 2048.) The positional embedding as well as the cls token are initialized randomly and are both learnable by the ViT. We propose that the size of our dataset with over 70.000 images and about 50.000 training images should be enough to train a model without pretraining. This Custom ViT, therefore, can be used to compare the result of pretrained and not pretrained models.

3.5. Training Setup

All models are initialized from pretrained StreetCLIP weights and trained using the AdamW optimizer, since it provides a good balance between regularization and consistent stable training, while also adding momentum to avoid local minima. We use a base learning rate of $1 \cdot 10^{-4}$, with a sinusoidal profile: the learning rate increases rapidly during the initial epochs and gradually decreases in later stages of training following a cosine curve. This strategy enables faster convergence early on while allowing fine-tuning during the final epochs, where smaller learning rates are crucial for minimizing the loss and avoiding getting stuck without going deeper into the loss function. To stabilize optimization, we apply dropout (0.5), layer normalization, and OneCycleLR learning rate scheduling. Training is performed in two stages: first training the output head on frozen encoder features, then fine-tuning the full model. We train using mixed-precision on a single GPU with batch sizes ranging from 16 to 64, depending on the model and memory constraints.

4. Dataset

Our dataset consists of 70,000 street-level images from San Francisco, collected via the Mapillary API. Each image is associated with precise GPS coordinates (latitude and longitude). We define geographic bounds as $(37.6, 37.9)$ for latitude and $(-123.0, -122.3)$ for longitude.

Figure 1: Visual Coverage of the total San Francisco Image Dataset [4]

For the classification model, coordinates are mapped to a 31×31 spatial grid as explained before.

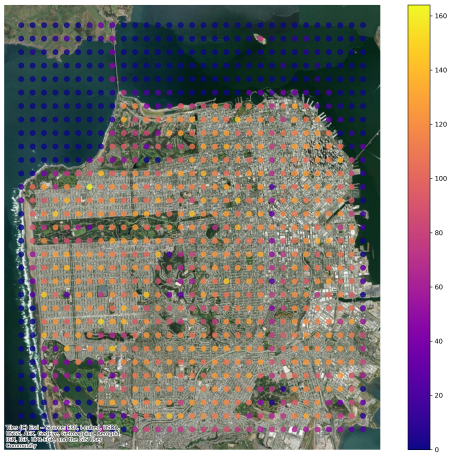


Figure 2: Visual Coverage of the classification grid [4]

5. Experiments and Results

5.1. Data Augmentation

Data augmentation can be very important for ViTs to learn robust predictions and to increase the performance with noisy input data. It can reduce overfitting to the training data since the training data is varied each epoch. Additionally, data augmentation can be used to synthetically train the ViT to be robust against spatial, rotational changes as well as mirrored images. We wanted to test the importance of data augmentation on our task of geolocation. Therefore, we implemented a stochastic image transformation which emphasizes changes in color, changes in the rotation and changes within the perspective of the image. Additionally, we used gaussian blur to synthetically train the model on images with worse quality than our model. All of these transformations were used in a stochastic manner, meaning that the intensity of the transformations varied for each image and each epoch. Specifically, we used the following values:

1. RandomAffine: 10 degrees, translation of (0.05, 0.05), a scaling of (0.95, 1.05) and a probability of 0.8
2. GaussianBlur: kernel size of 3, sigma of (0.1, 0.5) and a probability of 0.3
3. ColorJitter: brightness of 0.2, contrast of 0.3 and saturation of 0.2
4. RandomPerspective: distortion scale of 0.2 and a probability of 0.3

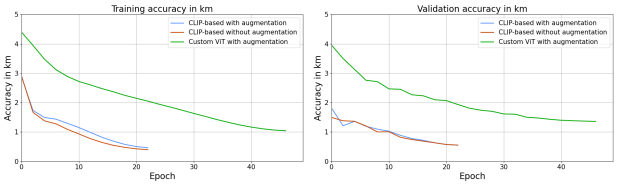
A common transformation that we did not use is "HorizontalFlip". Streets, facades and other important image features are most often not symmetric and flipped images are unlikely in normal usage of photos. Therefore, the "HorizontalFlip" does not provide more information about the images and instead flipped images could generate confusion, which is why we did not use that transformation.

5.2. Training Results

The training and validation curves of our models show that all of them were able to learn a good prediction of the geolocalisation of the street images. We used for all of our models a "One Cycle LR scheduler" [12] with a maximum learning rate of $5 * 10^{-5}$. The scheduler starts with a low learning rate and increases its value over the first 30% of the total batches. Afterwards it decreases the the value again until the end of training. This can increase the stability and learning speed of the model. A first trial with a maximum learning rate of $1 * 10^{-4}$ showed an instable learning process and reduced performance, which is why we chose the final maximum learning rate.

5.2.1 StreetCLIP with Gaussian Head

The learning curves of the three models with the Mixture Density Network head show that they were able to learn to predict reasonable coordinates. The Accuracy of the predictions were calculated by the distance between the gaussian mean with the highest value and the true position of the image. Both pretrained CLIP models were able they were able to predict the actual position with an average error of 600 meters. The Custom ViT was trained for additional 25 Epochs to compensate the advantage of pretrained parameters. The learning curves show that it took much longer for the Custom ViT to learn good location predictions. Additionally, the custom ViT is struggling with overfitting to the training data since the average validation accuracy 300 meters higher than the training accuracy. All validation accuracies started off below the training accuracies due to dropout within the training, underlining the overfitting. That behavior can not be seen for the CLIP-based models. Both the model with and without augmentation show a very similar performance in training and validation. This indicates that the pretraining supports generalization even in the case of finetuning without augmentation and shows the robust performance of the CLIP model.

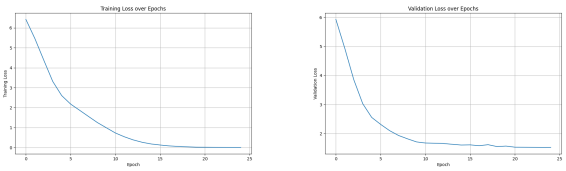


(a) Training loss curve (b) Validation loss curve

Figure 3: Gaussian-based models training curves

5.2.2 Grid-Based Classification Model

Regarding the learning curves for the grid-based classification model, we can see in Figure 4a that, after 25 epochs, the loss curve on the training dataset goes almost to 0 (with the real value being 0.00022). This indicates that the model was able to almost perfectly predict the classes of the pictures in the training dataset, as we are using the cross entropy loss for this model, as explained above. It also could indicate that the model has reached the global minima and not just local minima. Additionally, during the fine-tune of the model, the validation loss kept going down, as Figure 4b shows. In fact, the validation accuracy presented a logarithmic profile, with a final accuracy of a 66.82% on the validation dataset, staying almost stationary but always increasing after 18-20 epochs. With these results, we can assert that the agent was able to learn the important features on the dataset to classify the images while not overfitting the training data.



(a) Training loss curve (b) Validation loss curve

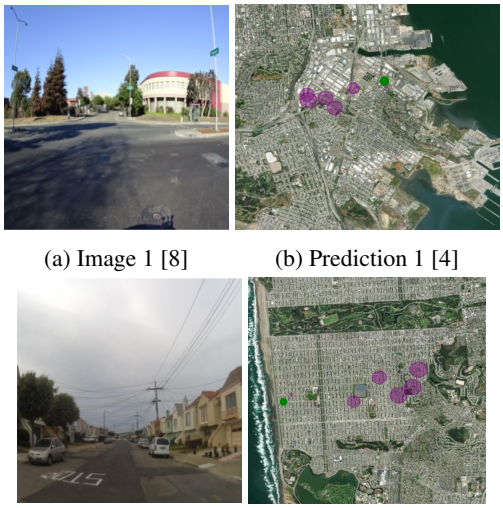
Figure 4: Grid-based classification model training curves

5.3. Qualitative Analysis of the Prediction Performance

To validate the trained models, we performed some prediction tasks with each of the presented models. For all predicted images, the purple circles show the predictions and the green circles show the actual location of the image. For additional prediction results, please refer to the ZIP file attached with this report.

5.3.1 Gaussian-Based Classification Models

As discussed before, the Mixture Density Model outputs six Gaussian probability density functions and weights them. This can be visualized as a probability density function over the landscape of San Francisco. A Qualitative analysis showed that both the model with and without augmentation do not use more than one Gaussian to predict the location of an image. A typical prediction is shown in 5. In this case, the purple circles show the area within each of the gaussians standard deviation. It can be seen that the predictions of five of the gaussians isn't close to the target at all. Only one prediction lays on the edge of actual location in green. It's standard deviation is much smaller. The other 5 predicted gaussians are usually between to the mid of the city and the actual coordinate. Only the last gaussian gives a valuable prediction with a much smaller standard deviation. The weight is only focused on this one last prediction often with a weight bigger than 99%. The indice of the main gaussian also did not change, which clearly shows that the model learned only one good prediction. That behavior reduces the geolocalization to a regression problem. The unintended outcome does not keep the model from making good prediction like the validation accuracy shows.



(a) Image 1 [8] (b) Prediction 1 [4]

(c) Image 2 [8] (d) Prediction 2 [4]

Figure 5: Typical prediction of a gaussian-based mode, where only one gaussian was used with a weight larger than 0.99 and a very small standard deviation

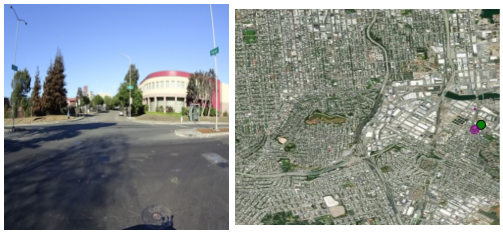
The qualitative analysis of the prediction of our custom ViT shows a similar result. The prediction of all six gaussian densities is closer to each other, but the weights are still focused on just one gaussian. This underlines that the behavior is likely due to other factors besides the model.

The most likely reason for the behavior is that the model largely reduced the weights for some of the gaussians very early, so that they were not able to learn better predictions due to vanishing gradients. Their worse performance would cause their weights to decrease further over time. To prevent this, another training could force a minimum weight value for all of the gaussians within training time. This would force the model to learn meaningful predictions with all gaussians. Good predictions could then lead to higher weights and might result in the intended behavior. Due to the generally good performance, it is also reasonable to treat the problem again as regression model. This indicates that the task of geolocalization within one city is not complex enough to benefit from the Mixture Density Model. This might change for the task of geolocalization in a bigger area.

5.3.2 Grid-Based Classification Model

Regarding the performance of the grid-based classification model, Figures 6b and 6d show the predictions and corresponding ground truth for the images in Figures 6a and 6c, respectively. Based on these examples and further analysis of predictions on the test dataset, the model demonstrates strong performance, particularly when considering the top 6 predicted scores (visualized as smaller purple circles).

As this is a classification task, all images assigned to the same cell result in identical predicted coordinates. However, because the grid tiles are sufficiently small, the predicted locations align closely with the actual coordinates. In cases of misclassification, many errors involve neighboring tiles that are adjacent to the true cell. This behavior suggests that even when the prediction is not exact, the model is still able to estimate the location with high spatial accuracy, often within a few hundred meters of the ground truth.



(a) Image 1 [8] (b) Prediction 1 [4]

(c) Image 2 [8] (d) Prediction 2 [4]

Figure 6: Examples of predictions of the grid-based classification model

When comparing the grid-based model with the Gaussian-based models, we observe that the former achieves higher accuracy, improved performance and faster training times. However, it provides only a single prediction (or a top-k set of most probable cells), which can be limiting in cases where the image is difficult to localize, particularly in urban areas where many locations share similar visual features, such as architecture, vegetation or traffic elements.

In contrast, the Gaussian model offers a more informative representation of uncertainty, providing not only a predicted location but also additional details such as the standard deviation and secondary peaks in the probability distribution. This richer output is especially beneficial in ambiguous settings, such as parks or waterfronts, where visual details may not be distinctive enough for confident classification.

5.4. Qualitative Analysis via Attention Visualization

To better understand the spatial reasoning of our models, we apply *attention rollout visualization* [1], which aggregates attention weights across all transformer layers to reveal which regions of the input image the model attends to when making predictions. This technique helps expose implicit model biases and highlights differences in how various architectures process geographic cues.

(a) Custom ViT + Gaussian [8] (b) StreetCLIP + Gaussian [8] (c) StreetCLIP + Grid [8]

Figure 7: Attention-rollout visualizations for the same input scene under three model variants.

5.4.1 Custom ViT with Gaussian Head

In this experiment, we analyze attention rollouts from our custom Vision Transformer (ViT) trained from scratch and coupled with a six-component Gaussian mixture head. As shown in Figure 7 (left), the model consistently attends to the sky or upper regions of the image, regardless of scene type. We believe this behavior arises because, without large-scale pretraining, the model over-relies on low-information cues such as sky color and illumination, and because most of the 70,000 Mapillary images were captured on a single day, so sky features (sun angle, cloud patterns) correlate with location. Future improvements may include augmenting sky features (e.g., randomized hue or contrast), balancing temporal distributions in the training data, and introducing regularization to encourage more diverse mixture predictions.

5.4.2 StreetCLIP with Gaussian Head

Figure 7 (middle) shows attention rollouts after fine-tuning the StreetCLIP ViT-L/14 encoder with our six-component Gaussian head. Unlike the scratch-trained ViT, this model consistently highlights semantically rich cues such as façade edges, road markings, street signs, vehicles, curb cuts, and the road's vanishing point, while devoting only limited focus to the sky, primarily along the horizon where colour and illumination gradients correlate with depth. We

attribute this behaviour to two key advantages of pretraining. First, StreetCLIP begins with CLIP's large-scale pretraining on 400 M image–text pairs, which teaches it to identify objects and scene layouts across diverse contexts; it is further fine-tuned on 1.1 M geo-tagged street-level images, reinforcing its ability to extract location-specific features such as building façades, curb geometry, and signage. As a result, during fine-tuning on our San Francisco Mapillary set, the encoder naturally attends to these high-level, place-relevant elements rather than lower-information cues like sky colour alone. Second, although our 70 000 Mapillary images were collected on a single day, StreetCLIP's pretrained weights act as a strong regulariser. By adopting a low learning rate and applying early stopping, we allow only the final transformer layers to adapt to the new domain; earlier layers (encoding fundamental geometry, texture, and object semantics) remain close to their pretrained state, mitigating overfitting to transient lighting or weather conditions inherent in a single-day capture. These combined effects yield attention maps that focus on structurally informative regions of the scene, improving the model's ability to localize images based on meaningful visual landmarks rather than transient environmental factors.

5.4.3 StreetCLIP with Grid Classifier

Figure 7 (right) displays attention rollouts for the grid head (961 classes; 31×31 bins) attached to the frozen StreetCLIP encoder. Surprisingly, the model now concentrates much of its attention on large blobs in the sky, the opposite of what we observe with the Gaussian head, despite both using the same pretrained backbone. One possible explanation is that with only 31×31 discrete cells, many adjacent bins share similar ground-level content, making subtle sky hue or horizon position the easiest separable signal. Additionally, the cross-entropy loss may encourage the model to exploit global illumination cues (since a single high-confidence token can determine the class) instead of aggregating many local features. Finally, if certain grid cells correlate with specific capture times or sun angles, the classifier could be overfitting to those lighting patterns. While these factors offer a plausible explanation, other elements (such as head capacity or optimization nuances) cannot be ruled out; further analysis, such as ablating sky patches or enforcing balanced sampling across time of day, will be needed to confirm the true cause.

5.4.4 Data Augmentation vs. No Augmentation (StreetCLIP + Gaussian)

Figure 8 shows the attention rollout for the model without our augmentation pipeline. When augmentation is applied, diverse colour and geometric jitter dilute sky cues, encouraging the model to spread its attention over façades, trees, and curb geometry. In contrast, without augmentation, sky hue remains the dominant, easiest signal, causing attention to collapse around a single bright horizon patch.

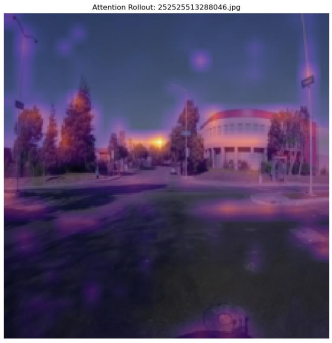


Figure 8: StreetCLIP + Gaussian *without* augmentation: attention collapses on a single horizon blob. [8]

6. Conclusion and Future Work

Our experiments show that fine-grained, image-only geolocation at city scale is already feasible with current vision-language foundations, provided that the output head is matched to the spatial structure of the task. The grid classifier built on a finetuned StreetCLIP encoder achieved the highest top-1 accuracy (66.8 % on a 31×31 mesh) and routinely placed images within a few hundred meters of their true GPS coordinates. The mixed-density (Gaussian) head reached comparable mean errors (600 m) while additionally yielding calibrated spatial uncertainty, and the scratch-trained ViT confirmed that most of this performance stems from geographic priors inherited during pre-training rather than from dataset peculiarities. Taken together, these findings demonstrate that a lightweight fine-tuning stage (often only a few epochs) is enough to repurpose StreetCLIP for dense urban localisation without any textual cues.

Equally important, our qualitative analyses highlight how model choice affects interpretability and error modes. Attention-rollout visualisations show that the Gaussian head encourages the network to attend to semantically rich, street-level landmarks (façades, curb geometry, signage) whereas the coarse grid head sometimes falls back on global illumination cues, such as sky hue, to separate adjacent cells. Although both strategies perform well numerically, the probabilistic formulation theoretically provides actionable uncertainty estimates (for example, flagging images whose highest-likelihood location is ambiguous among multiple neighbourhoods) while the classifier offers only discrete guesses. These complementary strengths suggest a two-stage pipeline in which a fast grid model prunes the search space and a density estimator refines the prediction and supplies confidence contours.

Looking ahead, three directions appear most promising.

First, extending the Gaussian head to a mixture with enforced component utilization would unlock its full expressiveness and mitigate the single-mode collapse observed in this study. Second, incorporating temporal and multimodal signals (time-of-day metadata, inertial cues, or short text snippets) could disambiguate visually similar blocks and reduce the residual kilometre-scale errors. Finally, evaluating the system on multiple cities and under varying capture conditions will clarify how well the learned spatial priors transfer and where additional domain adaptation is needed. By releasing our code, dataset splits, and trained weights, we hope to catalyse further work on reliable, uncertainty-aware urban geolocation and its downstream applications in autonomy, augmented reality, and urban analytics.

7. Contributions and Acknowledgments

- **Acknowledgements**: We thank Wenlong Huang, Fei-Fei Li, Ehsan Adeli, Justin Johnson, and Zane Durante for their guidance and valuable knowledge throughout this class; and the other students in the course for their ideas and discussions.

- **Author Contributions**: N.K. and R.M. mainly designed, implemented, and evaluated the Gaussian-based models. M.A., R.M., and N.K. developed and tested the grid-based classification models. M.A. generated and analyzed the attention roll-out visualizations. All authors contributed to writing and editing the paper and had a hand in all parts of the project.

References

[1] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, 2020.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[3] C. M. Bishop. Mixture density networks. 1994.

[4] Esri. World imagery basemap. Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community, 2024. © Esri.

[5] J. Haas, F. Xue, S. Zhou, R. Zhai, K. Ehsani, J. Deng, A. A. Efros, and T. Darrell. Learning to localize with vision-language models. *arXiv preprint arXiv:2303.08128*, 2023.

[6] L. Haas, S. Alberti, and M. Skreta. Learning generalized zero-shot learners for open-domain image geolocalization, 2023.

[7] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[8] Mapillary. Mapillary street-level imagery dataset. User-contributed street-level images licensed under CC BY-SA 4.0, 2024. Accessed via the Mapillary platform operated by Meta Platforms, Inc.

[9] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, J. W. Rae, D. Rezende, et al. Streetlearn: A multimodal environment for navigation in real cities. *arXiv preprint arXiv:1903.01292*, 2019.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[12] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[13] N. Vo and N. Jacobs. Revisiting im2gps in the deep learning era. In *ECCV*, 2018.

[14] T. Weyand, I. Kostrikov, and J. Philbin. Planet - photo geolocation with convolutional neural networks. In *ECCV*, 2016.