AddressCLIP: Empowering Vision-Language Models for City-wide Image Address Localization – Supplementary Material –

Shixiong Xu[1,3*†], Chenghao Zhang[2*], Lubin Fan[2‡], Gaofeng Meng[1,3,4‡], Shiming Xiang[1,3], and Jieping Ye[2]

[1] MAIS, Institute of Automation, Chinese Academy of Sciences
[2] Alibaba Cloud
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences
[4] CAIR, HK Institute of Science & Innovation, Chinese Academy of Sciences

1 Dataset Details

We provide more details of the dataset construction and statistical results.

1.1 Return Information of Google Maps API

For a querying pair of GPS coordinates (*latitude, longitude*), the reverse Geocoding API of Google Maps will return a list of address results, which are sorted by the distance between the address and query GPS coordinates. The information contained in each returned result is shown in Fig. 1. We also provide some examples of formatted addresses in returned results of the same query in Fig. 2.

Apart from the formatted address information, for each result, location type information is also provided to mark what kind of address is returned. Specifically, "ROOFTOP" means the address is an accurate location (usually a building). "RANGE_INTERPOLATED" means the result is an approximate position (usually on the road). "GEOMETRIC_CENTER" means the result is the geometric center of a multi-segment line (such as a street) or a polygon (such as an area). "APPROXIMATE" indicates that the returned result is an approximate location. We only use the formatted address and location type for address annotation. More details can be found in the official documentation of the Geocoding API of Google Maps.

1.2 Details of Address Annotation

We annotate administrative address information from coarse to fine for images with GPS coordinates in image geo-localization datasets by reverse geocoding, address extraction, and semantic address partition. An detailed illustration is shown in Fig. 3.

† This work was done when Shixiong Xu was an intern at Alibaba Cloud.
* Equal contributions
‡ Corresponding authors

```
results[]: {types[]: string, formatted_address: string,
 address_components[]: {short_name: string,
   long_name: string,
   postcode_localities[]: string, types[]: string},
 partial_match: boolean, place_id: string,
 postcode_localities[]: string,
 geometry: {location: LatLng,
   location_type: GeocoderLocationType
   viewport: LatLngBounds, bounds: LatLngBounds}}
```

Fig. 1: The specific information and their types in each returned result.

```
results[0].formatted_address: "277 Bedford Ave, Brooklyn, NY, USA"
results[1].formatted_address: "Grand St/Bedford Av, Brooklyn, NY, USA"
results[2].formatted_address: "Williamsburg, Brooklyn, NY, USA"
results[3].formatted_address: "Brooklyn, NY, USA"
results[4].formatted_address: "New York, NY, USA"
results[5].formatted_address: "Brooklyn, NY, USA"
results[6].formatted_address: "Kings County, NY, USA"
results[7].formatted_address: "New York Metropolitan Area, USA"
results[8].formatted_address: "New York, USA"
```

Fig. 2: Examples of formatted addresses in the returned results of the same query.

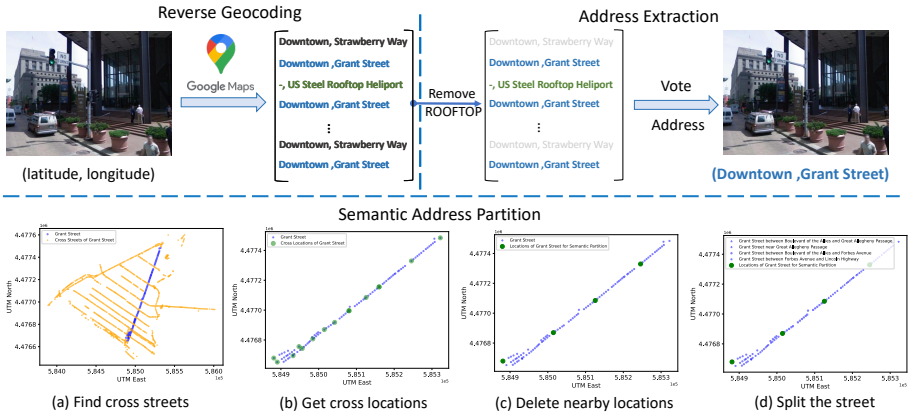


Fig. 3: The pipeline of address annotation, including the reverse Geocoding from GPS to addresses, extraction of address information, and semantic address partition.

Reverse Geocoding. Reverse geocoding is also known as address lookup, which converts a location into an administrative address that is easy to understand. We use the reverse Geocoding API of Google Maps to obtain the address information for each location. Specifically, given the GPS coordinates $(latitude, longitude)$ of a location, the reverse Geocoding API returns a list of addresses that are ordered by their match degree with the coordinate, *e.g.* $[A^{(1)}, A^{(2)}, \cdots, A^{(R)}]$, together with their location types. However, simply selecting $A^{(1)}$ as the ground truth address is often imprecise since the API tends to match the GPS coordinates to a place's or building's geometry center and return its address. For instance, when a large building is situated at an intersection of *Street A* and *Street B* but faces *Street A*, the coordinates on *Street B* near this building might always be labeled as *Street A* with $A^{(1)}$. Note that accessing the Google Maps API incurs certain costs, and for large and dense datasets like SF-IAL, which covers hundreds of thousands of locations, retrieving the address for every single location is prohibitively expensive and unaffordable for us. Therefore, for the SF-IAL dataset, we randomly sampled a subset of locations to conduct reverse geocoding, while the information for the remaining locations was filled in using the nearest neighbor's information. Therefore, we require additional post-processing steps to ensure the accuracy of the street-level address annotation.

Table 1: More statistics of the proposed Image Address Localization datasets.

Dataset	Covered Area	# Neighborhood train / test	# Street train / test	# Sub-Street train / test	# Locations Returned by API / Total
Pitts-IAL	20 km^2	19 / 19	194 / 165	428 / 327	10,586 / 10,586
SF-IAL-Base	6 km^2	15 / 15	121 / 110	400 / 369	8,371 / 17,067
SF-IAL-Large	170 km^2	124 / 124	332 / 327	3,616 / 3,406	17,686 / 233,820

Address Extraction. To alleviate the above issue, we adopt the following three steps to extract accurate street-level address information from the returned list for each location. Firstly, we remove the returned addresses that are matched to buildings along the street, whose location type is marked as "ROOFTOP" in the results. Secondly, among the remaining addresses, we choose the most frequently occurring address as the definitive address through a voting mechanism. Thirdly, we correct inaccurately labeled addresses through manual random verification. Building upon the aforementioned efforts, we have thus obtained accurate street-level annotations for each location. This serves as a vital foundation for the subsequent sub-street segmentation.

Semantic Address Partition. As mentioned in the main paper, to balance the length of streets while eliminating the naming ambiguity at intersections, we adopt the semantic address partition strategy for a more granular segmentation of streets. Specifically, the process is as follows: First, identify all the intersecting streets (orange) for each main street (blue). Second, obtain the intersection points (green points) on the main street that intersect with other streets. Third, remove closely spaced intersection points based on a certain threshold distance to avoid redundancy and too short sub-streets. Finally, split the main street into sub-streets and assign their names based on the remaining intersection points. In this way, the textual representation of addresses consists of the main street name and the name of one or two streets that intersect it. Moreover, to mitigate the long-tail issue inherent in street distributions, we employ a method of clustering adjacent addresses, merging shorter sub-streets (< 5 locations) into longer ones or broader areas. This process ensures that the description of addresses is both general and precise. It is important for the datasets that have irregular streets and sparse locations, *e.g.*, Pitts-IAL.

1.3 More Statistics and Visualizations

We have presented the basic information of the three image address localization datasets in Sec. 5.2 of the main paper. In Tab. 1, we provide more statistical details of the introduced three datasets, including the geographic area coverage of the locations, the number of neighborhoods, streets, and sub-streets in the training and test sets, as well as the number of addresses captured by the Google Maps API. Additionally, in Fig. 4, we illustrate the distribution of the number of locations associated with each address across the three datasets. The phenomenon of long-tail distribution is significantly mitigated after applying the semantic address partition strategy, resulting in a more balanced distribution of the number

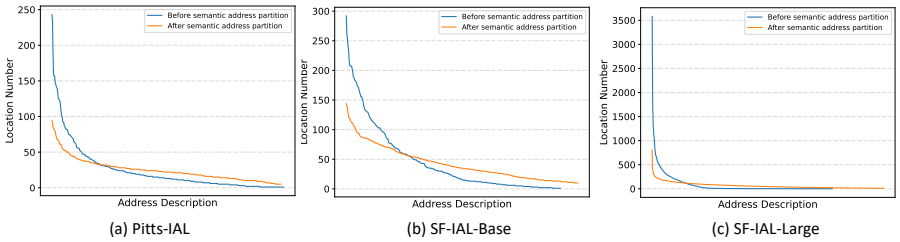


Fig. 4: The address distribution in the three Image Address Localization datasets. Both the distributions before and after semantic address partition are demonstrated.

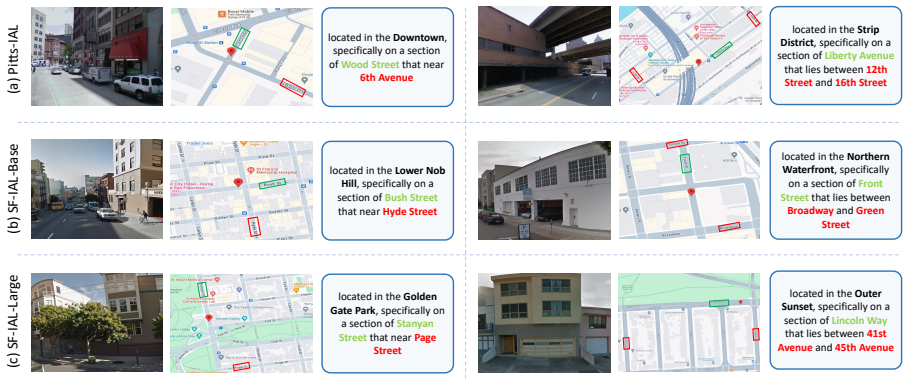


Fig. 5: Examples of address annotation for the three Image Address Localization datasets and their locations on the map.

of locations per address. Fig. 5 showcases examples of address annotations along with their corresponding positions on the map from three introduced datasets.

2 More Implementation Details

In training, the Adam ($\beta_1 = 0.9$ $\beta_2 = 0.98$) is adopted as the optimizer with the cosine learning rate from 2.4e-5 to 2.4e-8. We set the loss weights α, β and γ of the final objective to 1, 0.2, and 0.8, respectively. The batch size is set to 32 for each GPU and all the model is trained for 100 epochs on 8 Tesla V100.

3 Additional Ablations about Scene Caption

3.1 Scene Caption with Different Models

In this section, we present and discuss the impact of scene captions generated by different models. We utilize different versions of the vision-language model

Table 2: Comparison examples between the generated captions by BLIP-Caption-Base and BLIP-Caption-Large.

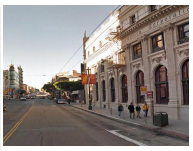
Image Captioning Examples with BLIP [7]:				
				
Base	a street view of a large building with cars parked on the side	a street view of a river and a city	a street view of a city with cars and buildings	a street view of a city with buildings and people
Large	a street view of a city with a lot of cars parked on the side of the road and tall buildings	a street view of a river and a city with a bridge in the background and a car driving on the road	a street view of a city street with cars and buildings on both sides of the street and a tram crossing	a street view of a city street with a few people walking on the sidewalk and a building in the background

Table 3: Examples of scene captions of BLIP-Caption-Base and BLIP-Caption-Large models and their performance comparisons on the Pitts-IAL and SF-IAL-Base datasets.

Caption by Models	Pitts-IAL				SF-IAL-Base			
	SSA-1	SSA-5	SA-1	SA-5	SSA-1	SSA-5	SA-1	SA-5
BLIP-Caption-Base	80.28	95.99	82.48	96.43	86.25	99.00	87.41	99.19
BLIP-Caption-Large	80.39	96.27	82.62	96.74	86.32	99.09	87.44	99.23

BLIP [7] for image captioning, namely BLIP-Caption-Base and BLIP-Caption-Large, to generate additional scene descriptions. Both of the two models are prompted with "A street view of". The minimum and maximum length of the output captions are set to 10 and 30, respectively. Tab. 2 presents examples of scene captions generated by the two models, where BLIP-Caption-Base produces naive descriptions while BLIP-Caption-Large can generate more elaborate captions. Tab. 3 shows the results of AddressCLIP on the Pitts-IAL and SF-IAL-Base datasets using different captions generated by the above two models. It can be observed that the resulted two AddressCLIP models achieve comparable performance, with that trained with richer scene captions yielding a slightly higher performance. This suggests that using a more powerful model to generate richer descriptive information can further enhance performance, but the gains might be limited. In practice, one needs to balance the cost of generating scene captions with the performance benefits.

3.2 Scene Caption Formats

Tab. 4 presents the performance comparison results of different scene caption forms of whether the textual address is incorporated or not. It is clear that the addition of geographical address information improves the accuracy compared to using only scene caption, suggesting that the combination of textual address and

Table 4: Performance of different scene caption formats on the proposed datasets.

Scene Caption Format	Pitts-IAL				SF-IAL-Base			
	SSA-1	SSA-5	SA-1	SA-5	SSA-1	SSA-5	SA-1	SA-5
scene caption w/o address	79.32	95.51	81.44	95.87	84.79	97.99	86.04	98.19
scene caption w/ address	**80.39**	**96.27**	**82.62**	**96.74**	**86.32**	**99.09**	**87.44**	**99.23**

Table 5: Performance of AddressCLIP on the Pitts-IAL dataset when the granularity of search space is varied. $\mathcal{W}$ is the number of prior streets.

Settings	None	Neighborhood	$\mathcal{W}$=20	$\mathcal{W}$=10	$\mathcal{W}$=5	$\mathcal{W}$=2
SSA-1	80.39	82.18	80.83	82.20	85.17	89.57

Table 6: Performance of different geographic coverage on the Pitts-IAL dataset.

# Images	3	6	12	24
SSA-1/SSA-5	56.86/84.58	69.08/92.06	76.80/95.45	80.39/96.27
SA-1/SA-5	61.12/86.50	72.48/93.09	79.45/96.01	82.62/96.74

scene captions is beneficial in enhancing AddressCLIP's capacity to accurately align images with their corresponding locations.

4 Discussion about the Characteristics of AddressCLIP

4.1 AddressCLIP with Prior Knowledge

In practical applications of address localization, users often possess some level of prior geographical context. For instance, while the address of an image may be unknown, the neighborhood or several candidate streets may be known. This additional information can effectively narrow the search space, thereby improving the model's accuracy due to the reduced number of potential addresses for consideration. To assess AddressCLIP's adaptability in these situations, we perform experiments where the candidate address is limited within a predefined neighborhood or several streets. Results across different settings are shown in Tab. 5. The performance of AddressCLIP improves as we search from coarse to finer granularity. The model's capacity to adapt to restricted search spaces affirms its applicability in real-world scenarios where partial geographic context is commonly available.

4.2 Different Geographic Coverage

The IAL task assumes that the addresses during testing are covered during training, thus the training sets and testing sets are geographically overlapped. In the

Table 7: Effect of mixed training on both Pitts-IAL and SF-IAL-Base datasets.

Train / Test	SSA-1	SSA-5	SA-1	SA-5
Pitts / Pitts	80.39	96.27	82.62	96.74
Pitts + SF / Pitts	80.46$_{+0.07}$	95.95$_{-0.32}$	82.62$_{-0.00}$	96.51$_{-0.23}$
SF / SF	86.32	99.09	87.44	99.23
Pitts + SF / SF	85.51$_{-0.81}$	98.31$_{-0.78}$	86.82$_{-0.62}$	98.72$_{-0.51}$
Pitts + SF / Pitts+SF	83.07	97.17	84.79	97.65

main experiments on the Pitts-IAL dataset, 24 images were taken from different perspectives at each location. To explore the potential of the proposed AddressCLIP under conditions with less geographical coverage, we randomly reduce the number of images per location to 12, 6, and 3. Tab. 6 shows the results of different geographic coverage on Pitts-IAL. One can observe that AddressCLIP can preserve **75%** of original performance with only **12.5%** location coverage, demonstrating its efficiency.

4.3 Mixed Training on Multi-city Datasets

In the main experiments, the proposed AddressCLIP is trained and evaluated on the Pitts-IAL and SF-IAL-Base datasets respectively, but this does not mean that our method can only work on a single city. To explore the potential of AddressCLIP on multiple city datasets, we combine the Pitts-IAL and SF-IAL-Base datasets as a mixed dataset for training. Tab. 7 shows the performance comparisons with single-city dataset training. As can be seen, without increasing the model size, the performance of mixed training achieves comparable performance to that trained on each dataset ($<$ **0.8**% degradation), which shows the scalability and potential of AddressCLIP to work across multiple cities.

5 Implements of "Image-GPS-Address" Pipeline

The "Image-GPS-Address" pipeline involves utilizing the image geo-localization technology to predict the GPS coordinates of a given image query and then translating them into textual addresses by the reverse Geocoding API of Google Maps. To compare our proposed end-to-end method with existing solutions, we have implemented this two-stage pipeline for these methods and provided evaluation results on the Pitts-IAL dataset in the Sec. 6.5 of the main paper. The experiment demonstrates that our approach achieves superior street-level address localization capabilities, presenting a promising method. We acknowledge that the proposed method is not yet as precise as GPS localization (within 25 meters), but its advantage lies in being an end-to-end solution. Moreover, the predicted textual addresses are more semantically meaningful and align with human description habits.

Specifically, we adopt state-of-the-art geo-localization approaches (i.e., CosPlace [3], MixVPR [1], EigenPlaces [4], AnyLoc [6], SALAD [5]) and use their publicly available model weights for feature extraction, all of which are claimed to have robust generalization capabilities. The database and query sets of the introduced Pitts-IAL dataset are utilized for the retrieval-based methods. We use ResNet50 with a feature dimension of 512 for image retrieval, except for AnyLoc [6] and SALAD [5], which use DINOv2 for feature extraction. For each query image, we first calculate its Euclidean distance with all the database images. Then we select the location of the image that has a minimum distance with the query as the predicted location. This location is used for reverse Geocoding to obtain the textual addresses.

Formally, given a query image Q, we define D as the database containing all reference images, where each reference image is denoted as D_i, $\forall i \in [1, 2, ..., N]$, and N is the total number of images in the database. We compute the Euclidean distance in the feature space between the query image and each reference image in the database as follows:

$$Dist(Q, D_i) = \sqrt{\sum_{j=1}^{M} (Q(j) - D_i(j))^2}, \tag{1}$$

where M represents the dimension of the feature. $Q(j)$ and $D_i(j)$ are the j-th feature of the query and database image, respectively. Then the predicted location L_Q for the query image is determined by assigning the GPS of the reference image with the minimum Euclidean distance in feature space, i.e.,

$$L_Q = GPS(\underset{D_i \in D}{\operatorname{argmin}} \operatorname{Dist}(Q, D_i)), \tag{2}$$

where $GPS(\cdot)$ indicates the lookup table of the Image-GPS pairs in the database.

In addition, when using the reverse Geocoding API, for fair comparison, we exclude returned address information where the location type is "ROOFTOP", and choose the most frequently occurring address from the remaining addresses as the final predicted address.

6 Details of Instruct Tuning with LLaVA

Multimodal large language models (MLLMs) are key building blocks for general-purpose visual assistants, and they have become increasingly popular in the research community. To apply MLLMs to the task of image address localization, we construct a multimodal dataset that pairs visual images with textual addresses in a question-and-answer format from Pitts-IAL. Fig. 6 shows an example. Here, we adopt LLaVA-1.5 [8] as the MLLM since it demonstrates impressive results on instruction-following and visual reasoning capabilities with open-source code and models. The instruct tuning process involves adjusting the LLaVA model's parameters with LoRA. During training, The AdamW is used as the optimizer and the cosine annealing scheduler is used to adjust the learning rate. We set

```
[
  {
    "id": "<path>",
    "image": "<path>",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nWhere might this photo have been taken? \
                  Tell me its street level adress."
      },
      {
        "from": "gpt",
        "value": " The address of this photo might be Grant Street, Pittsburgh, PA."
      },
    ]
  },
  ...
]
```

Fig. 6: An example of the constructed conversation data from the Pitts-IAL dataset.

the batch size to 16 and the learning rate to 1e-4. All training is conducted on 8 GeForce RTX 3090 GPUs with 24GB memory. The training of three epochs costs 20 hours. The input image size is set to 224×224. The fine-tuned LLaVA model, LLaVA-IAL, shows a significant improvement in the ability to predict textual addresses from images, which is indicative of its enhanced understanding of the visual and textual cues pertinent to the task of image address localization. This advancement holds promise for applications that require intelligent navigation and seamless interaction between the digital and physical realms.

7 Qualitative Demonstration

In this section, we qualitatively demonstrate the effectiveness of our method. We first show the results of AddressCLIP with the image query. Then, we provide more visualizations of the similarity map between the image embedding and the address text query in Pittsburgh and San Francisco.

7.1 AddressCLIP with Image Query

Fig. 7 shows the Top-5 textual address predictions generated by the proposed AddressCLIP, based on given image queries, along with their locations on the map. The examples provided come from the Pitts-IAL and SF-IAL-Base datasets. In the majority of cases, the correct prediction is identified within the first address (Top-1), demonstrating AddressCLIP's precise address localization capability. Subsequent predicted addresses are also close to the correct location. Additionally, we showcase some failure examples where the Top-1 prediction is not correct. Even so, the correct address can still be predicted within the Top-5 addresses, and the Top-1 predicted address is typically close to the actual location.

7.2 AddressCLIP with Address Text Query

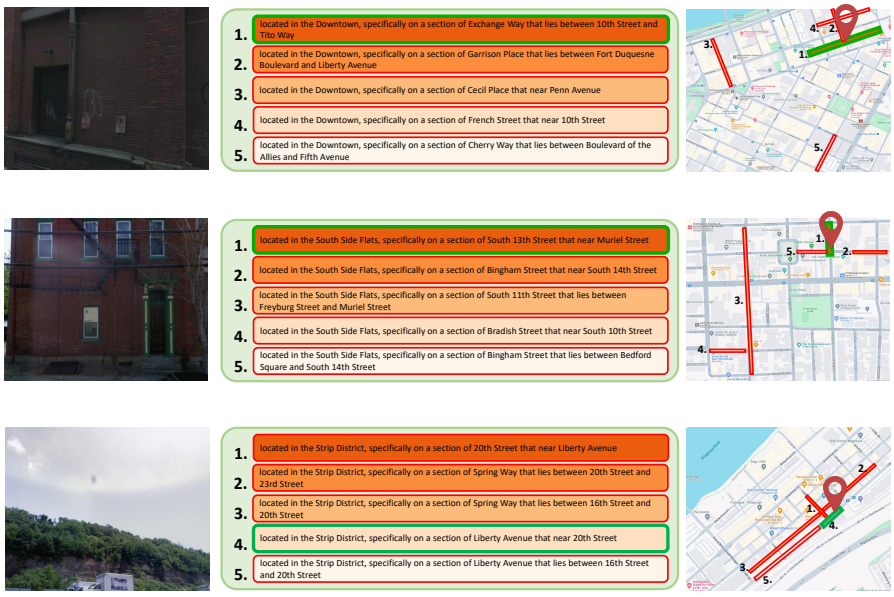
In Fig. 8, we display more visualizations of the embedding similarity distribution between images and given address queries on the map of Pittsburgh and San Francisco. It is observed that on both Pitts-IAL and SF-IAL-Base datasets when provided with a text query, our AddressCLIP is capable of effectively pinpointing the approximate area corresponding to the text based on the features of the street view images. The results presented are divided into three levels: neighborhood, street, and sub-street from the top row to the bottom.

8 Broader Impacts

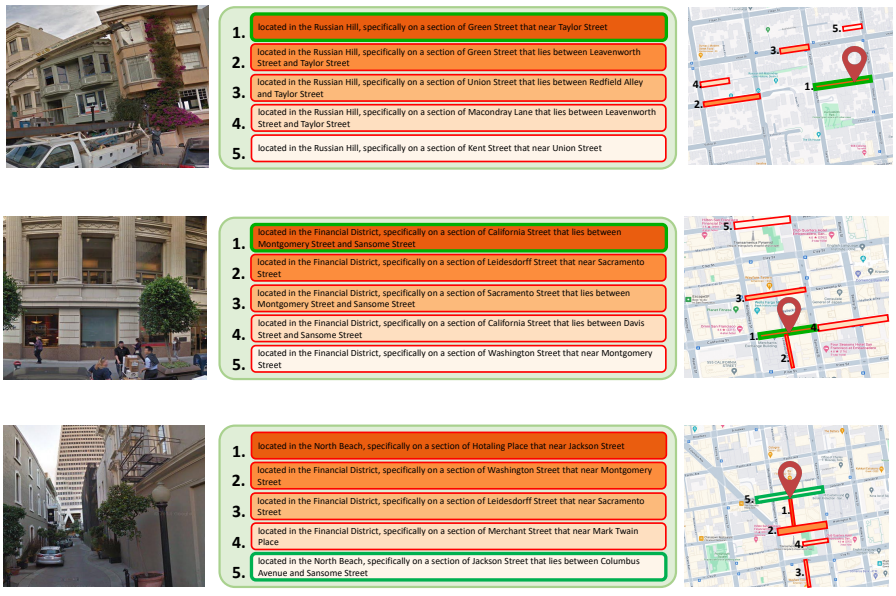
In this study, we introduce the problem of image address localization, which aims to predict the textual address where a given image was taken, consistent with how humans typically describe addresses. With the proposed AddressCLIP, we can obtain more semantic address information, which has the potential to revolutionize the way we navigate and interact with physical spaces. The introduced image address localization datasets are derived from open-source datasets Pitts-250k [2] and SF-XL [3] as well as publicly available Google Maps API, thus we do not anticipate any potential negative social impact arising from this work.

References

1. Ali-Bey, A., Chaib-Draa, B., Giguere, P.: Mixvpr: Feature mixing for visual place recognition. In: WACV. pp. 2998–3007 (2023)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: CVPR. pp. 5297–5307 (2016)
3. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: CVPR. pp. 4878–4888 (2022)
4. Berton, G., Trivigno, G., Caputo, B., Masone, C.: Eigenplaces: Training viewpoint robust models for visual place recognition. In: ICCV. pp. 11080–11090 (October 2023)
5. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition. arXiv preprint arXiv:2311.15937 (2023)
6. Keetha, N., Mishra, A., Karhade, J., Jatavallabhula, K.M., Scherer, S., Krishna, M., Garg, S.: Anyloc: Towards universal visual place recognition. IEEE Robotics and Automation Letters (2023)
7. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML. pp. 12888–12900 (2022)
8. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. ArXiv **abs/2304.08485** (2023), `https://api.semanticscholar.org/CorpusID:258179774`



(a) Image query examples of Pitts-IAL



(b) Image query examples of SF-IAL-Base

Fig. 7: The address localization results predicted by AddressCLIP and their positions on the map according to image queries. The results from Top-1 to Top-5 are displayed, with green boxes indicating correctly predicted addresses and red boxes indicating incorrectly predicted addresses.

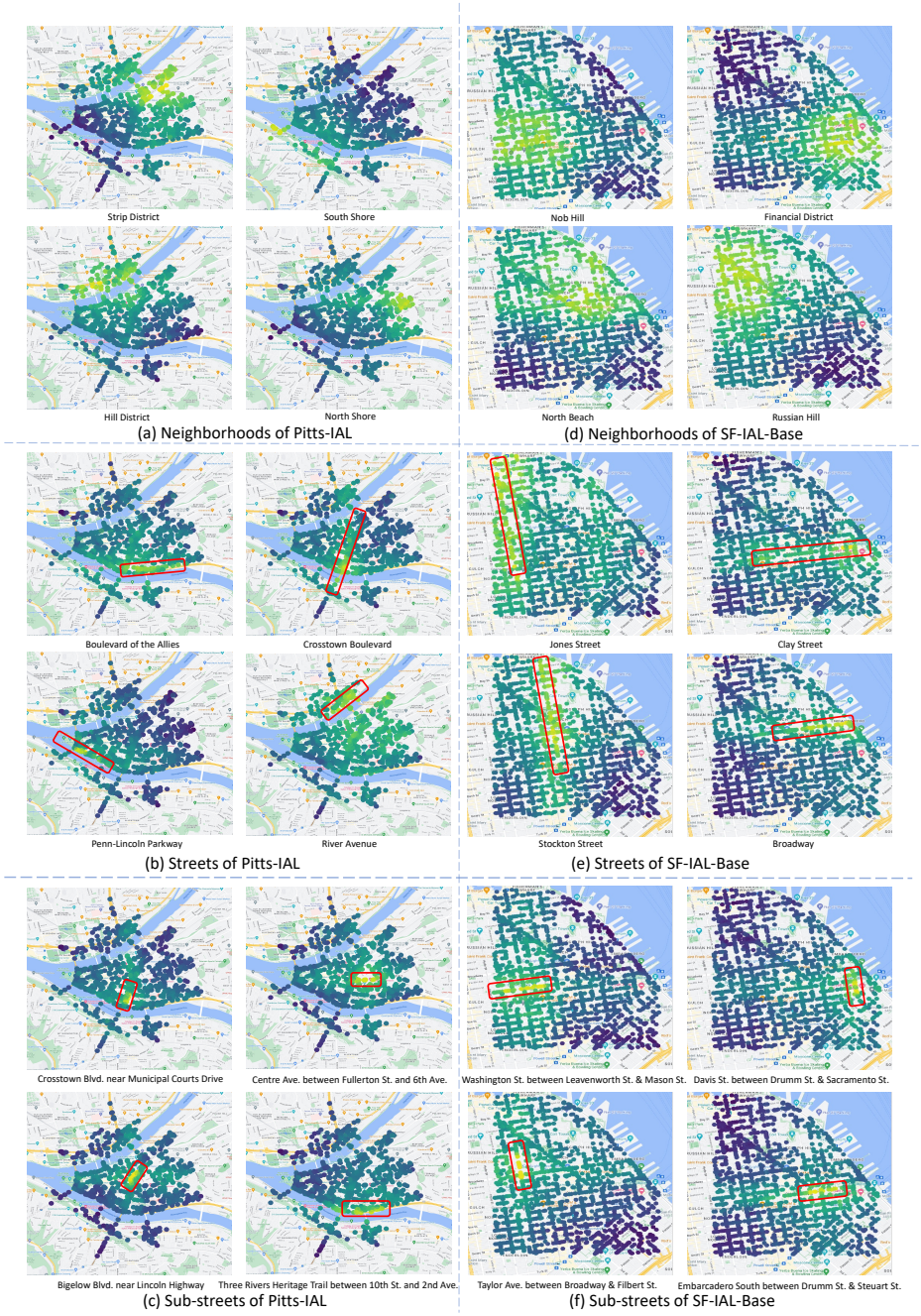


Fig. 8: More qualitative demonstrations with a given textual address query using AddressCLIP in Pittsburgh and San Francisco. The brighter the scatter point, the higher the similarity of the embedding between the image and the query address text. The red box represents the actual geographic range of the query street in the map.