2022

Image Geo-localization with Cross-Attention

Connor Greenwell

University of Kentucky, cgree3@gmail.com

Digital Object Identifier: https://doi.org/10.13023/etd.2022.341

Recommended Citation

Greenwell, Connor, "Image Geo-localization with Cross-Attention" (2022). *Theses and Dissertations--Computer Science*. 120.
https://uknowledge.uky.edu/cs_etds/120

Connor Greenwell, Student

Dr. Nathan Jacobs, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

Image Geo-localization with Cross-Attention

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy in the
College of Arts and Sciences at the
University of Kentucky

By
Connor Greenwell
Lexington, Kentucky

Director: Nathan Jacobs, Professor of Computer Science
Lexington, Kentucky
2022

ABSTRACT OF DISSERTATION

Image Geo-localization with Cross-Attention

The problem of estimating the location from which un-geotagged photographs were captured has been well studied by the computer vision community in recent years. The central proposal of this thesis is to define a common framework within which existing approaches can be constructed and evaluated, and to introduce a new method under this framework which uses cross-attention between the query image and a database of satellite imagery with known geotags. Our experiments fit within three broad categories: 1) evaluating the ability of image localization approaches to generalize to unseen regions; 2) examining performance changes under various reference database resolutions, scales, and densities; and 3) exploring localization with multi-modal reference imagery. Our key contribution is the notion of attending between query and reference imagery throughout inference, compared with the existing practices of attending late or not at all.

KEYWORDS: computer vision, machine learning, image localization, remote sensing, geospatial analysis

Connor Greenwell

August 15, 2022

Image Geo-localization with Cross-Attention

By
Connor Greenwell

Nathan Jacobs
Director of Dissertation

Simone Silvestri
Director of Graduate Studies

August 15, 2022
Date

ACKNOWLEDGMENTS

First to my advisor, Nathan Jacobs. A simple "thank you" cannot convey my gratitude. Eight years ago you answered an email from an undergrad looking to learn more about machine learning, and the conversations that followed kicked off the career-path I am on today. Sometimes I wonder where I'd be had I not had the courage to start that discussion. Thank you for everything you've done: taking that first chance on me, encouraging me to go to grad school, teaching me everything I know. I am forever in your debt.

Thank you to the other sources of advice and support I have had over the years, Matt Leotta, David Page, and especially Richard Souvenir. And thank you to my committee, Ramakanth Kavuluru, Brent Harrison, and Mike Sama. Your advice and feedback has been invaluable.

Thank you to my countless collaborators: Scott Workman, Muhammad Usman Rafique, Hunter Blanton, Zach Bessinger, Tawfiq Salem, Ryan Baltenberger, Menghua Zhai, Jon Crall, Benjamin Brodie, Mohammad Tariq Islam, and Gongbo Liang. We've accomplished so much together, and there's so much left to do.

And a special thank you to my family and friends, without whom I would not have had the strength to do any of this. Thank you Mom, Dad, and Maddie for shaping me into the person I am today. And finally, thank you to my darling wife Abbie for lifting me up and being there for me every single day, I love you with all my heart.

Financial Support Portions of this work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via contract 2017-16110300001. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily

representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

To Abbie,
You make all
things possible.

And to Tesla,
Oh, how I wish
you were here.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Motivation

Estimating where exactly on the Earth a picture was taken using only its image content is a challenging task that has received significant research attention in recent years. At this scale, context clues such as architecture [7, 11], plants and signs, geography, and even the clothing styles of people in photographs [9, 17] can all be used to refine the range of possible places an image might have been captured at. In robotics applications, the fine-grained image localization methods that are used often feature localization error that is measured on the order of meters or centimeters [19–21]. In contrast, for global image localization it may only be possible to estimate where an image was captured to within a threshold of 10s or 100s of kilometers and only occasionally are precise locations able to be found.

Despite the extreme drop in spatial precision, these methods are still broadly applicable in a number of settings. For example in art, for retrieving images that look like they come from the same place as a reference photo. In crime and policing, identifying the sites of crimes posted to the internet, including identifying where sex trafficking and abuse imagery such as those involving minors (CSAM) was captured [1]. There also exist similar but fully manual efforts by independent investigation groups [2] which reconstruct crimes and disasters [3] and document war zones [4]. And in business intelligence, identifying and localizing trends in images to specific places without the need for geotags. Also in anthropology, localizing old photographs in both space and time [44]. And finally, there are those who localize images for the fun and challenge of it, such as the *View From Your Window* project [5] and the players of GeoGuessr [6].

Further, we anticipate the future use of image localization in photograph collection managers, *a la* the heat-map functionality in Google Photos (Fig. 1.1). Between the time of the invention of photography and the common adoption of consumer-grade global positioning system (GPS) modules, hundreds of millions or even billions of photographs

[1] http://www.traffickcam.com/about

[2] Forensic Architecture: https://forensic-architecture.org/, Bellingcat: https://www.bellingcat.com/

[3] https://forensic-architecture.org/investigation/beirut-port-explosion

[4] https://www.washingtonpost.com/news/worldviews/wp/2014/08/26/heres-how-to-track-terrorists-on-google-earth/

[5] http://dish.andrewsullivan.com/2010/09/07/the-view-from-your-window-contest-winner-14/

[6] https://www.geoguessr.com/

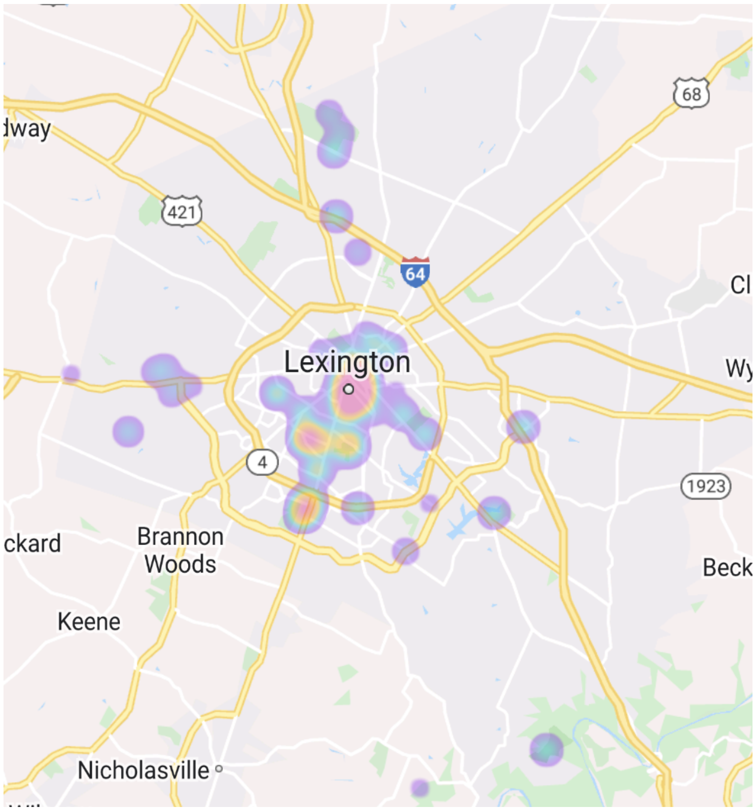


Figure 1.1: **Example of Google Photos Heat-map.** From the authors photo collection.

were captured without geotags. Even low-precision location estimates can be combined with this imagery to support the construction of family histories through space and time, or even to find that restaurant you ate at on holiday whose name you just can't quite remember.

1.2 A Review of Image Localization Methods

Recent image localization approaches fall into two broad categories: 1) those which are classification-based [35, 39, 53] that divide the Earth into distinct regions *a priori* and identify within which of those regions an image was captured; and 2) those which are retrieval-based [15, 27, 29, 41, 52, 55, 66] that do similarly *ad hoc* by learning a common feature space within which query images can be matched against massive databases of geo-tagged imagery, composed of either ground-based or satellite-view imagery with known locations.

Classification-based Localization

Classification-based approaches first divide a given region, typically the entire Earth, into smaller regions within which a query image will be classified as being captured, or not [35, 39, 53]. Most often, the parent-region is recursively divided into localizable sub-regions or cells following a strategy that is based on the density of images located in each sub-region. If a sub-region contains a number of images larger than a set upper-bound, it is divided into a new collection of sub-regions. Many of the resulting sub-divisions will contain too few or possibly zero images; those which contain less than a lower-bound are completely discarded to ensure that each sub-region contains sufficient imagery to train a classification model.

The earliest work in this space, PlaNet [53], recursively divides the world into a quad-tree where each cell is a square projected onto the Earth's sphere, and each sub-division divides these squares into quadrants. The proposed classification architecture is a convolutional neural network (CNN) whose final layer estimates the probability that the query image was captured in any given cell, directly from the pixels of the query image. This model is also applied to localizing photo albums by extending it with a Long Short-Term Memory (LSTM) model which considers images from an album in sequence, iteratively refining the global localization prediction.

A variant of this approach, called C-PlaNet [39], extends PlaNet with combinatorial partitioning, wherein several coarse divisions of the parent region are computed independently, then are combined to create a set of fine-grained regions. The coarse divisions of the parent region are decided by merging neighboring quad-tree cells of a specific depth, rather than recursive splitting. Each of the fine-grained locations is uniquely identified by the set of coarse cells that overlap it. Following this fact, multi-head classifier optimizes multiple classification heads and losses, one for each coarse division, and fine location estimates are made by combining the coarse estimates.

In practice, images are commonly collected from a number of settings that each feature unique patterns and limitations in their localization cues. For example, images captured indoors often feature smaller objects that may be culturally relevant and indicate a specific part of the world, but are not precisely localizable without additional clues such as a skyline visible through a window. In rural outdoor images, the unique combination flora and fauna may serve a similar function to the culturally relevant objects in indoor images. An approach called GeoEstimation [35] accounts for these different cases by turning image localization into a multi-stage process which features a classification of query into one of a number of scene categories, scene category specific localization models, and predictions at multiple partitioning scales.

Retrieval-based Localization

Retrieval-based methods approach localization from a different direction than classification-based methods. The primary difference being that the set of locations that a query image can be assigned to is not defined *a priori*, but instead at runtime. The main method for defining these locations is by incorporating an outside data source that is closely tied to the idea of location, typically satellite imagery, and estimating the likelihood that the query image is associated with the reference image. While it is possible to use other data sources, satellite images feature a number of traits that are beneficial to localization: coverage of the planet that is (mostly) uniform, and frequent updates as collection platforms continue to orbit the Earth taking new photographs.

Early approaches in this space [27, 52, 55] imported techniques from metric learning and image-retrieval to image localization by training CNNs to match query photographs against large databases of geotagged satellite imagery. They did so by training encoder models which regress a feature vector from an input image such that the feature is suitable for retrieval. Two encoders are trained, one each for photographs and satellite imagery. These models are trained in tandem by presenting pairs of known pairs of images, regressing feature vectors, and optimizing a metric learning loss which maximizes the similarity between the vectors of known matches and while minimizing said similarity for known negative examples.

Improved methods of regressing the feature vector encoders began by replacing the simple global average pooling (GAP) operator which typically terminate each encoder. One such work replaced the pooling step with a continuous variant of a histogram of gradients operation [15], via which the main improvement over GAP is that visual concepts that are useful for localization can be represented and also quantified in the overall image representation. When localizing full panoramic images with known orientation, the spatial orientation of those visual words can be incorporated into the images overall representation as shown by methods [41] which learn spatially-aware aggregation methods as part of their image encoder.

Much of the work in this space has assumed that camera orientation was known while location was unknown, aligning all images, panorama and satellite, to face north during both training and inference. In practice this assumption rarely holds when attempting to localize images found in the wild, which are typically not panoramas nor packaged with compass metadata. To overcome this, later methods drop the assumption of north-facing

Table 1.1: **Overview of image geo-localization datasets and their properties.**

Dataset	Scale	Size	Pano.	Sat zoom	Sat res.
CVUSA (CVPR) [62]	USA	~50K	✓	Bing 19	750x750
CVUSA-500k [55]	USA	~500K	✓	Bing 14,16,18	750x750
VIGOR [66]	4 cities	~100K	✓	Google 20	640x640
CVACT [29]	1 city	~128K	✓	Google 20	1200x1200
im2gps [11]	global	~6.5M	X	—	—
YFCC100M (w/GPS) [46]	global	~5M	X	—	—

imagery by applying random rotations to the panorama and train the localization models for the additional task of recovering the orientation offset [29, 41], seeking to improve both feature learning and localization accuracy.

Each of these methods has been focused on the problem of retrieving localizing images against databases of satellite imagery which cover large spatial areas, such as the contiguous United States. Recently, meter-level urban localization [66] has been explored in select urban areas with the added challenge of including *confusors*, or reference images without paired query examples in either the training or testing set.

1.3 Image Localization Datasets

The fundamental components of an image-localization dataset are images, each with some form of geotag. Many such datasets have been published over the years and each has their own specific qualities. The images themselves can be sourced in a couple of ways. The least common approach is for them to be specially collected by hand using GPS-enabled cameras for the purpose of the dataset, however this approach is much more common for other types of datasets. Much more often, images are scraped from social media sites such as Twitter, Facebook, or Flickr [46]. Wherever they come from, images posted to the internet often contain metadata in the form of image metadata (Exif) tags, and one of the encoded features is the GPS signal from the camera or phone which captured or posted the photograph. The other common approach is to scrape the 360°panoramas served by the Google StreetView service [54]. These are indexed geographically in Google Maps and the public application programming interface (API) enables pairing these images with accurate geotags.

Often paired with these images is a satellite image or other overhead view. These are often sourced from the basemaps used by mapping services such as Google, Bing, or Apple Maps. These basemaps are composite images composed of a number of captured by high-resolution satellite platforms on cloudless days. These images feature a large

Table 1.2: **Overview of differences between CVUSA and CVUSA-500k.** CVUSA-500K is more than 10x larger, features higher resolution imagery with known location, and multiple spatial scales of overhead imagery. We add additional imagery in the form of cutouts from a Sentinel-2 basemap.

Dataset	GPS	Ground Res.	m/pix	Aerial Res.	Aligned?
CVUSA (CVPR) [62]	X	1232x224	0.30	750x750	✓
CVUSA-500k [55]	✓	3072x1536	9.55, 2.39, 0.60	800x800	✓
+ Low-res Sentinel-2	✓	—	225.4	256x256	X

amount of detail, typically pixels can be as small as 0.5 meters per side, but can be quite old as the satellites that produce them may only view the entire Earth a few time per year. Alternatively, coarser basemaps computed from low-resolution satellites that have a more frequent revisit rate are used. One such platform is Sentinel-2, which produces a new image of the Earth twice a week at a resolution of around 10 meters per pixel.

1.3.1 CVUSA-500k

Much of the work in the retrieval-based image localization space is based on solving the panorama→satellite problem. This is different enough from most classification-style localization, which usually operates on non-panoramic photographs, as to make their methods and results difficult to compare. Most of this work is based on a what is frequently referred to as the CVUSA dataset [62]. It consists of approximately 35K Google StreetView panoramas paired with co-located satellite imagery. However, the absolute location of each pair is unknown.

To support our aim of bridging classification and retrieval, we focus on non-panoramic images. For our experiments, we adapt the full version of the CVUSA dataset [54] (which we refer to as CVUSA-500k) in two ways. First, we simulate ground-level views by sampling cutouts from the panoramas in CVUSA-500K. Second, we collect a new source of satellite imagery composed of red-green-blue (RGB) cutouts from a Sentinel-2-based basemap. The bounds of these cutouts are defined by the H3[7] cell within which the panorama is located. This has two side effects: 1) panoramas are no longer guaranteed to be located at the exact center of the paired overhead image; and 2) multiple panoramas can be associated with a single overhead image. The latter side effect makes this data much more like the classification problem which is inherently many-to-one (many photographs to a single location bin).

[7] Uber H3: `https://h3geo.org/`

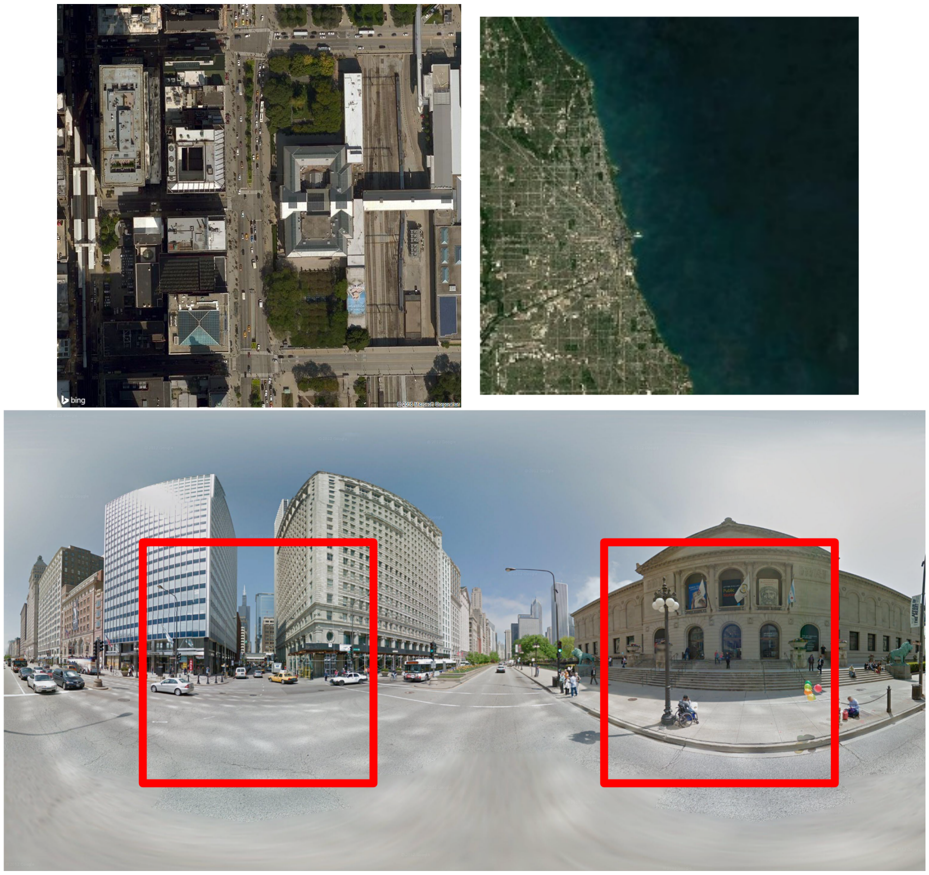


Figure 1.2: **Example images from the CVUSA-500k dataset.** (Top Left): An overhead image of Chicago, IL with resolution 800x800 at 0.6 meters/pix. (Top Right): An overhead image of Chicago, IL with resolution 256x256 at 225 meters/pix. (Bottom): A panoramic street-view image from the dataset with lateral cutouts are highlighted. Orientation with respect to the overhead image is unknown.

1.4 Challenges

Global image localization is a challenging task for a number of reasons. One of the more difficult is the issue of data density. Localization datasets are typically collected either from geotagged social media images on the internet, or by collecting 360°panoramas such as those available from Google Street View. Both of these approaches are biased towards urban areas and away from rural areas, albeit for different reasons. The panorama based datasets contain this bias because the imagery is almost always captured on roadways, and there are simply more of those in urban centers. There are also issues of completeness and remaining up to date. Social media data's biases are more subtle, and can be grouped into two main factors: 1) tourist bias, i.e. the tendency for people to take and post more photographs when on vacation (often in large cities) or when notable events occur; and 2) availability bias, where it is necessary to own a camera to take and post photos, while also

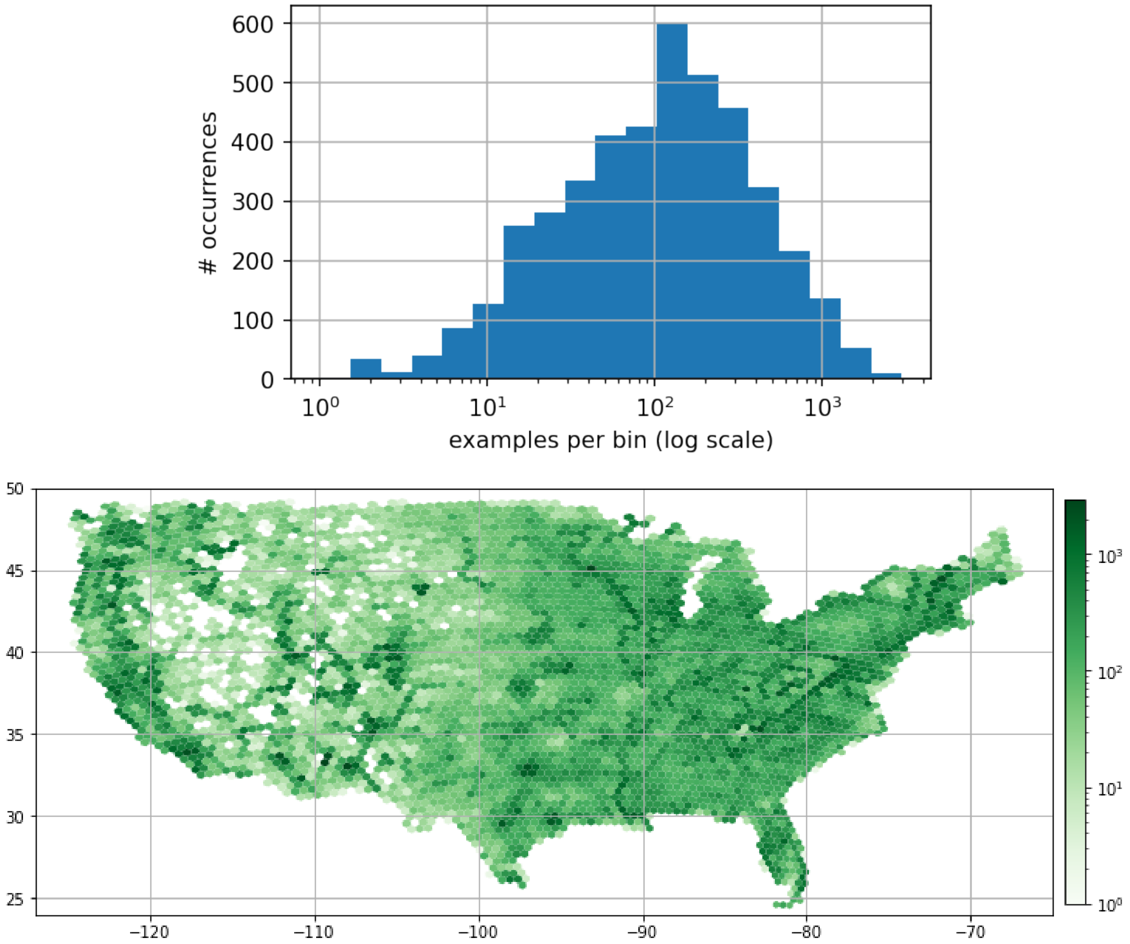
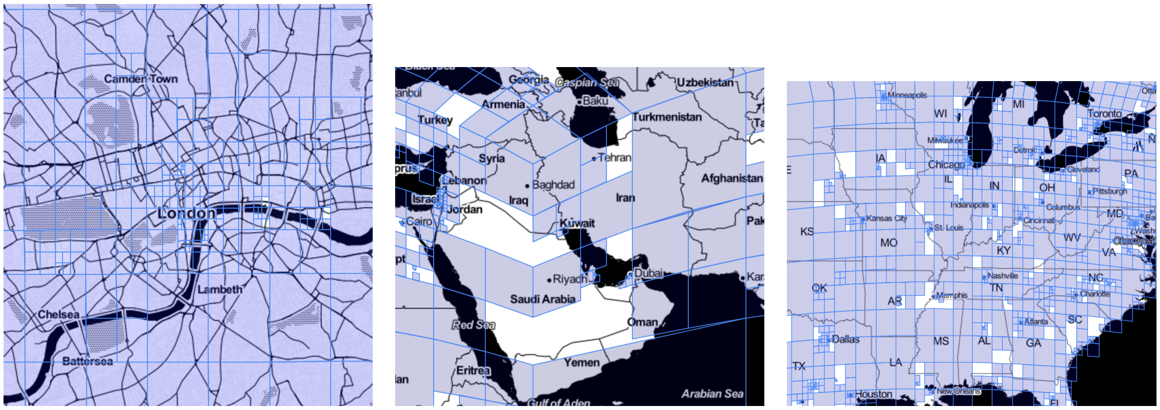


Figure 1.3: **Density of images in geospatial bins.** The distribution of the number of CVUSA examples that exist within each bin is non-uniform and varies geographically. (Top) A histogram of how many occurrences of each quantity of examples per bin as they occur in the dataset. (Bottom) Bin density roughly correlates with population density.

posting them to a social media site that the data is being collected from. Some sites are more suited to this than others, namely Flickr, while others are more difficult to collect from for reasons relating to their Terms of Service or API access, such as Facebook or Twitter.

Ambiguity present in the content of images is an additional challenge that localization approaches need to overcome. For example, individual locations within a given forest, desert, corn field, etc. , might not be visually distinct from any other and thus unable to be precisely located by an algorithm. Similarly, indoor areas provide few to none geographic clues about the rooms specific location on the Earth, and many are only loosely localizable to a particular region based on context clues such as language on signage and regionally-identifiable objects. Data from social media comes with additional ambiguities, for many



(a) High density areas.

(b) Low density areas.

(c) Incomplete coverage.

Figure 1.4: **Undesirable properties of binning strategies based on image density.** (a) Some areas feature high-density, low-area cells, potentially as small as a few city blocks in London, UK. (b) Other regions of the world feature cells that cover very large areas, such as Iran which only has 6 cells. (c) Finally, there is incomplete coverage even in areas such as the American Midwest.

kinds of photographs such as selfies or portraits the main subject of the image is not directly relevant to the location of the image and a localization algorithm will need to identify features on the periphery on the image. Finally, images sourced from the public often come with erroneous geotags, either as a result of simple GPS sensor noise, user error when uploading to social media, or intentionally obfuscated to protect sensitive info such as the authors home or work address.

Additionally, there are inherent issues with classification based approaches to overcome, such as the high variability in the size of the resulting cells required for classification approaches, which can range in size from as small as a single city block (Fig. 1.4(a)) to the size of a small country (Fig. 1.4(b)). Similarly, the density of the images does not always correlate with the importance or difficulty of localizing an image from that location. Classification approaches are also limited to making predictions within the regions from which they are provided images with known locations. Typically this results in incomplete coverage (i.e. areas interest around the world that are not contained within any cells, as shown in Fig. 1.4(c)). Further, as they cannot make predictions in areas they were not trained they are unable to generalize.

1.5 Research Objectives

In this work, we seek to unify the two classes of localization approaches and show that such a hybrid approach can overcome issues that are present in both. To do so, we first need to establish a framework from which both the existing approaches and our proposed hybrid method can be constructed and evaluated. Our framework starts from the existing classification-based localization model where a query photograph is passed-through a machine learning model, and then a set of logits corresponding to some set of location labels or classes is predicted, where the computation of those logits and how they relate to location is an "implementation detail". For classification, this is simply a linear projection of the outputs of the model to a vector whose size is the same as the number of pre-computed categories. For retrieval, that linear projection is defined by a database of reference imagery and another learned model.

Our proposed approach can be considered a hyper-network for classification, with classifiable locations dynamically defined by overhead imagery with known locations, and further conditioned on the specific information available in those images. Specifically, we propose using a cross-attention operation to make conditional estimates of geo-location, where the query sequence is a single feature token representing the ground level image, and the key/value sequence is a collection of satellite images the query could possibly be located within. In this document we will explore and evaluate a number of specific configurations of this architecture.

Introducing a transformer decoder to the localization model enables attaching additional information about the query image and the reference database by encoding and including such information on the positional encoding added to each token passed to the transformer. This information can include specific location details for each of the reference images or additional metadata known to belong to the query image. In this work we focus on the location of the reference imagery and what the optimal way of encoding that might be.

Both the retrieval approach and our proposed hybrid models are capable of making predictions in regions on which they were not trained, i.e. the model is never exposed to specific details of architecture and other localizable features from such regions. We will evaluate and compare these models in these settings to measure each models ability to generalize beyond the area in which it was trained. We are also interested in the situation where there is a limited amount of data available from held out regions, and will evaluate all models under varying amounts of extra data.

Finally, all methods in the proposed framework can be trained for various scales of target locations, from very coarse (1,000's of km in area) to very fine (10's of km in area). This variability also opens up the possibility of hierarchical localization, where one or more models are used to first localize an image to a large region, then increasingly specific sub-regions in turn. We will evaluate each model for these different settings and identify which are best suited for each scale and which combinations produce the best hierarchical localizer.

1.6 Summary and Thesis Outline

Thesis Statement Applying cross-attention to the features generated during retrieval-based image-localization enables a new class of localization approach based on conditionally refined image representations.

In Chapter 3, we address the problem of image geo-localization, of which there are two primary classes of approaches: 1) classification-based, where images are predicted as being within one of a set of pre-defined geographic regions; and 2) retrieval-based, where images are queried against a database of images with known locations. We seek to bridge these approaches, by modifying the classification approach to no longer assign images to a fixed set of geographic locations, instead conditioning those locations on overhead imagery. Our proposed approach uses cross-attention as a general image localizer, predicting among which provided satellite images a query photograph was captured within. Existing approaches are tailored to one of these settings or the other. In contrast, our approach attempts to solve both problems with a single model. We present the existing methods, and detail our proposed hybrid approach. We also conduct a detailed ablation study of the proposed approach.

In Chapter 4 we explore and evaluate the generalizability of each method of interest. To do so we divide a major image-localization dataset into two geographically distinct parts, training on one and testing on the other. We follow evaluations approaches similar to several existing works which have measured localization accuracy in unknown areas [29, 66], except at much larger spatial scales and our evaluation is not limited to urban areas. Further, we explore the impact of varying the location encoding for the hybrid methods reference image database. We also evaluate the change in generalization accuracy as various amounts of data from the held out regions are introduced during training, a proxy task for low-data situations in areas of interest.

In Chapter 5 we evaluate the impact of varying the spatial resolution of reference data and the spatial extent of location bin sizes on each of the localization methods presented in previous chapters. Localizing to areas of varying size can change the accuracy of localization at the detriment of precision, and each model has their own trade-off profile. We also evaluate a number of configurations for hierarchical localization.

In Chapter 6 we introduce a multi-modal variant of the the proposed image-localization approach. As a proxy for situations where the reference image database is composed of satellite images sourced from multiple distinct platforms, we train localization models on reference imagery that has been broken into their constituent channels. Further, we evaluate this approach on a satellite-view semantic segmentation task.

2.1 Transformers in Computer Vision

Transformers are a class of deep learning model which were first introduced [50] in 2017 in the context of natural language processing (NLP) where they quickly became one of the dominant approaches to sequence-to-sequence language tasks. Following their rise in popularity in the NLP space, they were then applied to computer vision problems by converting images to sequences of tokens, much like sequences of word tokens in NLP [8]. Transformers similarly became one of the dominant approaches to a variety of computer vision problems, especially in settings where billion-scale datasets are available. Since then, Transformers have been applied to video understanding [2], semantic segmentation [43, 56, 60, 64], 3D point cloud understanding [63], and much more.

The key insight of Transformer is the introduction of an attention mechanism. Given three sets of tokens, a query-set $Q \in \mathbb{R}^{N \times F}$, a key-set $K \in \mathbb{R}^{M \times F}$, and a value-set $V \in \mathbb{R}^{M \times G}$, we define attention as the following function:

$$\text{Attention}(Q, K, V) = \text{Softmax}(Q \cdot K^T) \cdot V, \tag{2.1}$$

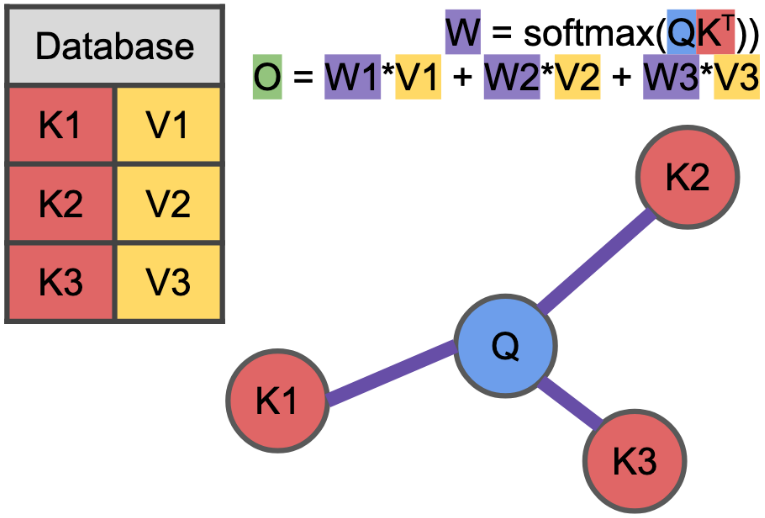


Figure 2.1: **Attention is the soft approximation of a key-value lookup in a database.** The key-indexing step is replaced with a soft-assignment from the query feature to each of the key features. The soft-assignment weights are then used to compute a weighted sum of the corresponding value features.

where N and M are quantities of tokens and F and G are the size of the input and output token vectors, and the output of the function is a new set of tokens in $\mathbb{R}^{N\times G}$. The attention function can be thought of as soft dictionary lookup, where the key-indexing is relaxed from the identity function to a similarity-based weighting between Q and K, and the returned value is a weighted sum over all values V.

A very common extension called *multi-head attention* first projects the input tokens down to n lower-dimensional sub-spaces $\mathbb{R}^F \mapsto \mathbb{R}^{F/n}$, performs attention on each of these "heads" independently, then concatenates the outputs. Generally, this is done in order to reduce the spatial and computational complexity of attention, speed up learning, and learn more robust features as a form of in-network ensembling.

In practice, there are two specific incantations of attention which are commonly used: 1) *self-attention* where $Q = K = V$, and 2) *cross-attention* where $K = V$. In their simplest forms, Transformers are residual-connected stacks of these basic attention-variants and shallow multi-layer perceptrons (MLPs). Specifically, an encoder layer is defined as:

$$Q_{i+1} = \text{Norm}(Q_i + \text{SelfAttention}(Q_i)) \tag{2.2}$$

$$Q_{i+1} = \text{Norm}(Q_{i+1} + \text{MLP}(Q_{i+1})), \tag{2.3}$$

and a decoder layer is defined as:

$$Q_{i+1} = \text{Norm}(Q_i + \text{SelfAttention}(Q_i)) \tag{2.4}$$

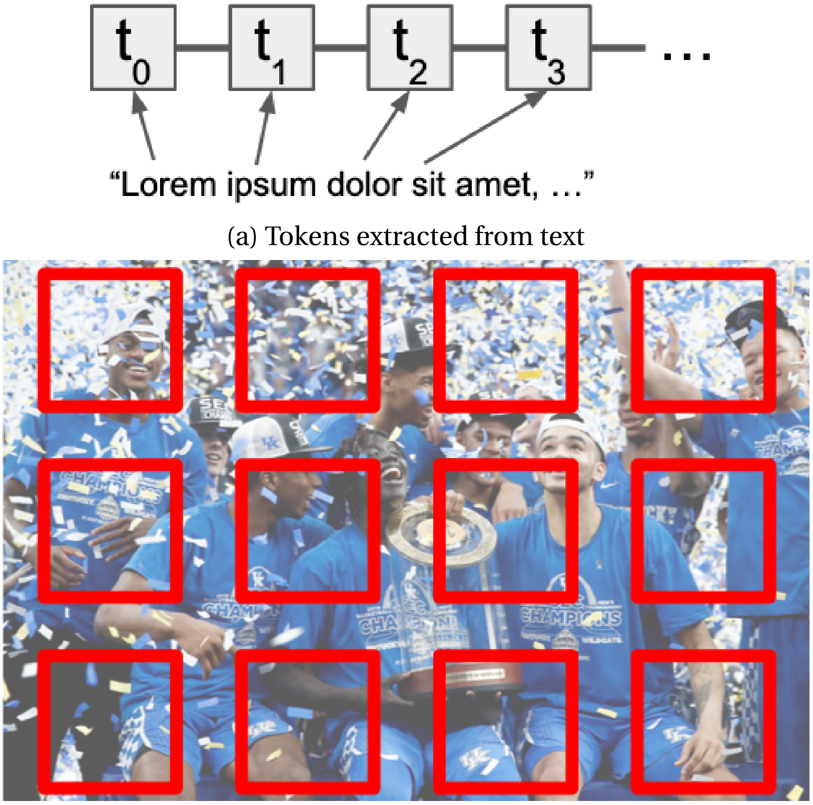
$$Q_{i+1} = \text{Norm}(Q_{i+1} + \text{CrossAttention}(Q_{i+1}, V_i)) \tag{2.5}$$

$$Q_{i+1} = \text{Norm}(Q_{i+1} + \text{MLP}(Q_{i+1})), \tag{2.6}$$

where $\text{Norm}(\cdot)$ is some normalization function, such as BatchNorm [16], InstanceNorm [49], or LayerNorm [1].

The input to Transformer encoder- and decoder-networks is a sequence of tokens where each token represents *something*. In NLP settings, these are typically words or n-grams of words, and the tokens themselves are learned embedding vectors that encode the semantic meaning of the corresponding word or n-gram. A wider variety of solutions exist in the vision space. In the simplest case, $n \times n$ sub-images are extracted from an input image on a grid, then each is flattened and linearly projected onto $\mathbb{R}^F$. In more complex systems, this step may be replaced with a CNN, useful for encoding the content of a patch and its surrounding context, or even reducing an entire image to a single token in multi-image tasks, such as image-localization.

In order to incorporate the relative positioning of each token into the decision-making process, the position of each token must be encoded and combined with the tokens



(a) Tokens extracted from text

(b) Tokens extracted from an image

Figure 2.2: **Examples of tokenizing data for processing by a transformer.** (a): Text is typically broken down by word, and each is converted to a token by a learned embedding model which associates each word with a vector. (b): Images are converted to tokens by first extracting windowed sub-images, then a separate model predicts a token vector from that sub-image.

representation. This positional encoding is responsible for representing the positions of each token in the input sequence as a signal that can be used by the self-attention modules to reason about the relative and absolute positions of each token. In the original NLP context this was the indices denoting where each tokenized word was located in the sentence. In the computer vision setting, this is typically the X, Y position of each tokenized sub-image. Typically these coordinates are converted to a high-dimensional continuous representation by projecting them to a higher dimension $p' \in \mathbb{R}^{n/2}$ with a random projection matrix whose weights are kept fixed throughout training. The final positional encoding is $p = [sin(p'), cos(p')]$, where $p \in \mathbb{R}^n$ and this is added directly to the tokens representation vector.

In settings such as image-level classification or regression, the tokens produced by the Transformer model must be reduced to a single summary token before being passed to downstream prediction models. One simple approach to this is to take the mean of the entire sequence of tokens, however in practice this leads to less desirable representations. The typical approach is to append an additional token to the sequence, referred to as the CLS or class token. This token has a fixed value throughout training and the corresponding output is passed directly to any sequence-level downstream tasks.

2.2 Metric Learning and Image Retrieval

The goal of representation learning is to learn models which take high-dimensional and un-evenly sized inputs from a given domain and compactly embed them into a low-dimension latent space. One common approach to learning to represent a data domain is the auto-encoder [6,33,51], where the task being optimized is embedding, and then reconstructing, datapoints from the low dimensional latent code. Similarly, generative models [23,37] learn to generate plausible examples from a dataset given a sample from a pre-defined representation space, where "plausible" is defined by a critic network which is simultaneously trained to differentiate between real examples from the dataset and the results of the generator model.

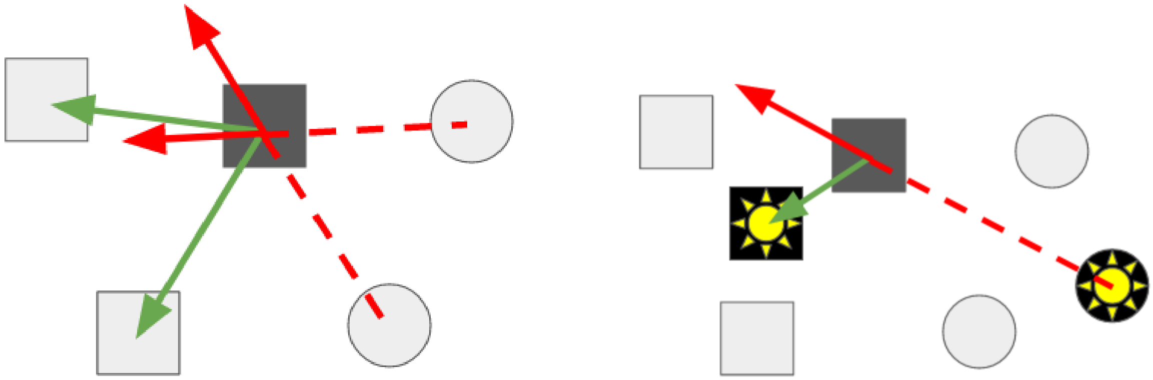
Metric-learning approaches [13, 14, 24, 36, 45] instead learn such representations based solely on relationships between datapoints by directly optimizing for a latent space where similar examples have embeddings that are close to each other, and dissimilar examples are embedded far from each other. These latent spaces are shaped by two main factors: 1) the distance function, or metric, which is used to compare the latent representations predicted by the trained model; and 2) the loss function, which for a given batch of examples and the associated pairwise distance matrix will drive the shape and arrangement of the latent space by directing the training of the representation model.

While one of the natural choices for a distance function would be the well known L_n distances, much more common in the metric learning literature is the cosine distance function [5,30,55]:

$$D_{\cos}(X,Y) = -\frac{X \cdot Y}{||X||\,||Y||}, \tag{2.7}$$

where X and Y are representations returned by the metric learning model, and $D_{\cos} \in [-1,1]$ where $D = -1$ indicates identical inputs and $D = 1$ maximally different inputs. There has also been some recent work which explored the use of the signal-to-noise ratio as a distance for metric learning [59].

A number of loss functions have been proposed for this task; most fall into one of two categories: 1) tuple-based methods, and 2) proxy-based methods. Tuple-based methods [36,52] are those where the computed distances for sets of known matching



(a) Tuple-based

(b) Proxy-based

Figure 2.3: **Overview of basic metric learning paradigms.**

and non-matching examples are used directly to push match embeddings towards each other and mis-match embeddings away from each other. For example the contrastive and triplet losses:

$$L_{\text{contrastive}} = \max(0, M - D(x_i, x_j)) \tag{2.8}$$

$$L_{\text{triplet}} = \max(0, D(x_i, y_i) + M - D(x_i, x_j)) \tag{2.9}$$

$$L_{\text{soft-triplet}} = \text{sigmoid}(D(x_i, y_i)) - D(x_i, x_j))) \tag{2.10}$$

where x_i and x_j are known to match, and y_i is known to be a mis-match to both. Many such methods also introduce a margin term M which defines the minimum desired distance between matched examples.

Recently, proxy-based methods [4, 22, 34] have introduced the use of artificial anchor points in the latent space as a tool for reducing the computational complexity required when computing the pairwise distance matrix, especially compared to contrastive and triplet based losses. This is achieved by only considering distance computations between proxies and sampled datapoints, rather than all pairs of data points, i.e. the complexity of the distance computation step is reduced from $O(|X|^2)$ to $O(|X| \times |P|)$ where $|X|$ is the size of each sampled batch and $|P|$ is the number of proxies. These methods learn a set of proxy points within the latent space as an extra set of weights to be optimized during training. In settings where there are labels available to be leveraged, each category is assigned a proxy representation, and during training the known categories are used to assign matches/non-matches. In settings where there are no labels to leverage, the closest proxy in the latest space as defined by the chosen distance function $D(\cdot, \cdot)$ is used instead.

Chapter 3 Methodology

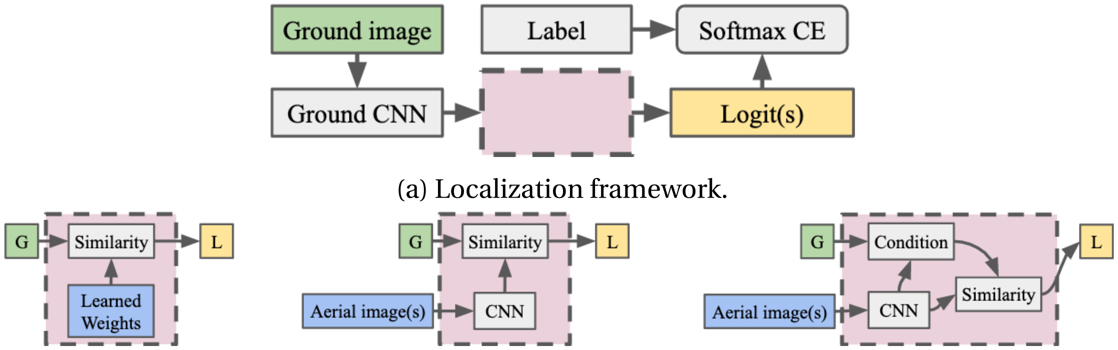
In this chapter we describe our proposed approach to unifying the classification and retrieval settings of image geo-localization. First we present a framework within which image localization approaches can be constructed under. Next, we introduce our proposed approach which we call MetaLoc, which borrows important elements from each of the existing categories of approaches. From retrieval, we take the ability to condition localization estimates on a database of imagery with known location; from classification, we take the training strategy and loss function. Specifically, MetaLoc takes as input a query photograph and a set of aerial images with paired geographic information. All images are tokenized into feature vectors, geographic information is encoded into an identically shaped vector and combined with the aerial image vectors. Next, a new ground image feature is computed using a transformer decoder where the existing ground feature is the query and the database of aerial features is the key/value. The resulting aerial-conditioned ground feature is then compared with the aerial features using the dot product to estimate the final similarity score. The remainder of this section discusses these steps in greater detail.

3.1 A Framework for Image Localization

In order to facilitate both the fair comparison of localization methods and to highlight their main similarities and differences, we construct a framework within which localization methods can be constructed. Localization methods share a number of common components, specifically their inputs, outputs, and losses which drive the training process. By fixing as many of these components to be the same in all settings we can more rigorously explore which changes have the most impact on localization performance.

The input to our framework is a single query image whose location is to be estimated. A feature encoder model transforms that image into a compact representation, often a vector in $\mathbb{R}^n$. That representation is then passed to some method-specific model which makes the actual location estimation. Finally, that estimate is compared with the ground truth to compute a loss value which is used to train all of the constituent models and parameters. We diagram this framework in Fig. 3.1(a).

The main source of variation is in the method specific model. For a simple classification model (Sec. 3.2 and Fig. 3.1(b)), this can be as simple as a learned set of weights in $\mathbb{R}^{n\times L}$, where L is the number of locatable regions. Other methods may include additional



(a) Localization framework.

(b) Classification approach.

(c) Retrieval approach.

(d) Proposed MetaLoc approach.

Figure 3.1: **Overview of image localization framework and approaches.** Image geolocalization can be decomposed into a simple framework, as shown in (a), where query a image is reduced to a feature vector which is passed as input to some method which predicts among which possible locations the image is from.

sources of input and new models and weights (Figs. 3.1(c) and 3.1(d)). We present two such methods in Secs. 3.3 and 3.4

While the specifics of the image encoder and the loss function are flexible, for most of this document we fix them to be a ResNet18 model [12] and the softmax cross entropy loss, respectively. Similarly, we compare all models using the same set of metrics, evaluating for recall at various thresholds both in terms of metric similarity and geographic proximity. We describe these metrics in detail in Sec. 3.5.

3.2 Classification

Classification-based approaches to image localization take a single query image as input and directly predicts where the image was captured. Predictions take the form of estimated parameters to a categorical distribution, where each sub-region that the image could possibly have been captured in is considered a category.

In other work [39, 53] the parent region that fully encompasses all examples in the dataset, typically the entire Earth, is divided into smaller regions. Several approaches to this partitioning have been proposed, including a Quad-trees[1]. To decide whether or not to subdivide a particular region into a set of sub-regions, the number of images from a reference dataset that lie within the query region is counted and then if the count is above some threshold the region is split, discarding sub-regions that contain fewer images than a second threshold. This splitting criteria is applied recursively until there

[1] Google S2: `https://s2geometry.io/`

are no remaining sub-regions that contain enough images to warrant a split. Each of these resulting sub-regions is considered a category to which all of its contained images are assigned.

To ensure compatibility with retrieval-based methods (Sec. 3.3), we simplify this process significantly and instead divide the parent region into uniformly sized sub-regions without recursive subdivision. Because quad-tree cells are often non-uniform polygons, especially towards the magnetic poles and quad-tree decision boundaries, we instead use a hexagonal tiling of the Earth [2], assigning each image to its corresponding cell at a particular zoom-level.

Within our image localization framework, these first compute a compact feature representation, $v \in \mathbb{R}^n$, which encodes the location relevant information in the query image and compares it against a learned set of location representations, one per locatable region, $W \in \mathbb{R}^{n \times L}$, typically via a dot product between the representation vector v and the learned weights, $v \cdot W \in \mathbb{R}^L$. each element of $v \cdot W$ is a similarity score related to the likelihood that the query image was captured at that particular location.

3.3 Retrieval

In contrast to classification-based image localizers, retrieval-based methods do not learn to localize images to a fixed set of regions. Instead they learn to score the similarity between a query image and a set of satellite images representing a number of regions such that the matching satellite image has a high score and all others are low. Then, when deploying such models, the similarity between a query image and each of the satellite images for potential locations is computed and used to estimate the query's location, those with the highest score are considered the most likely. This last step is conceptually similar to classification except with a potentially varying number of categories.

To compute the similarity between query and satellite images, a second model is trained which acts as a feature extractor for satellite images. While this satellite feature encoder could be any CNN, typically it is identical to the query feature extractor with an independently updated set of weights. The similarity scoring function itself can take many forms, most often a dot product, $G \cdot A = S$ where $G \in \mathbb{R}^{|G| \times n}$, $A \in \mathbb{R}^{|A| \times n}$, and $S \in \mathbb{R}^{|G| \times |A|}$, or the cosine similarity, which is the dot product between unit-normalized G and A, $s = \text{cossim}(c * F, c * A)$, where c is a scaling factor.

Typically the loss function for retrieval-based methods that is used is the triplet margin loss [52] or some variant. These optimize representations such that the similarity scores

[2] Uber H3: `https://h3geo.org/`

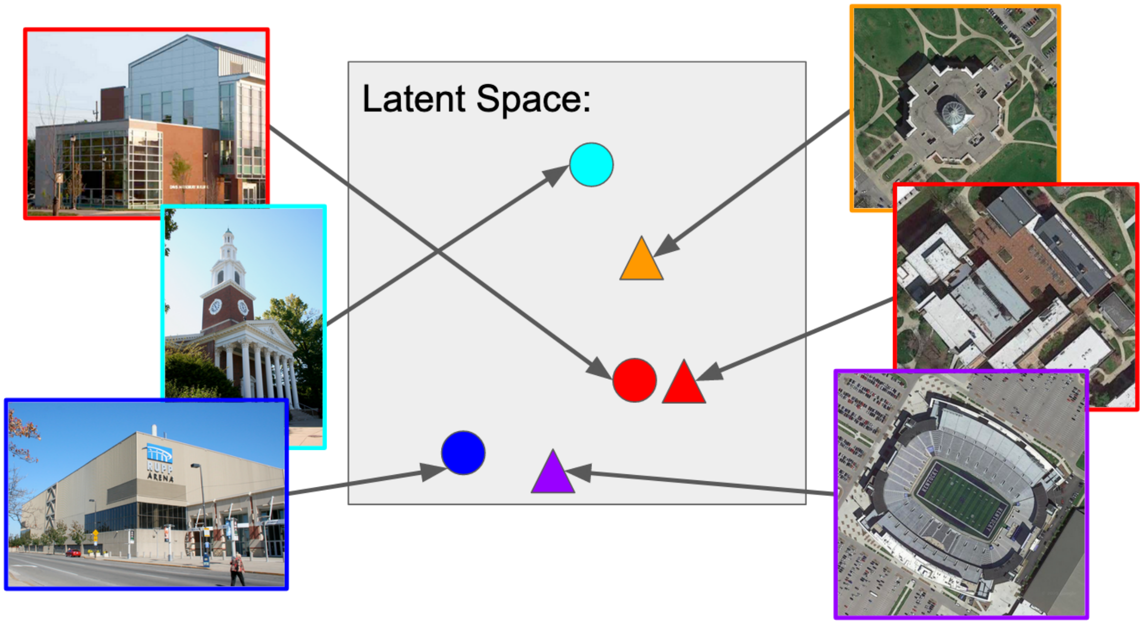


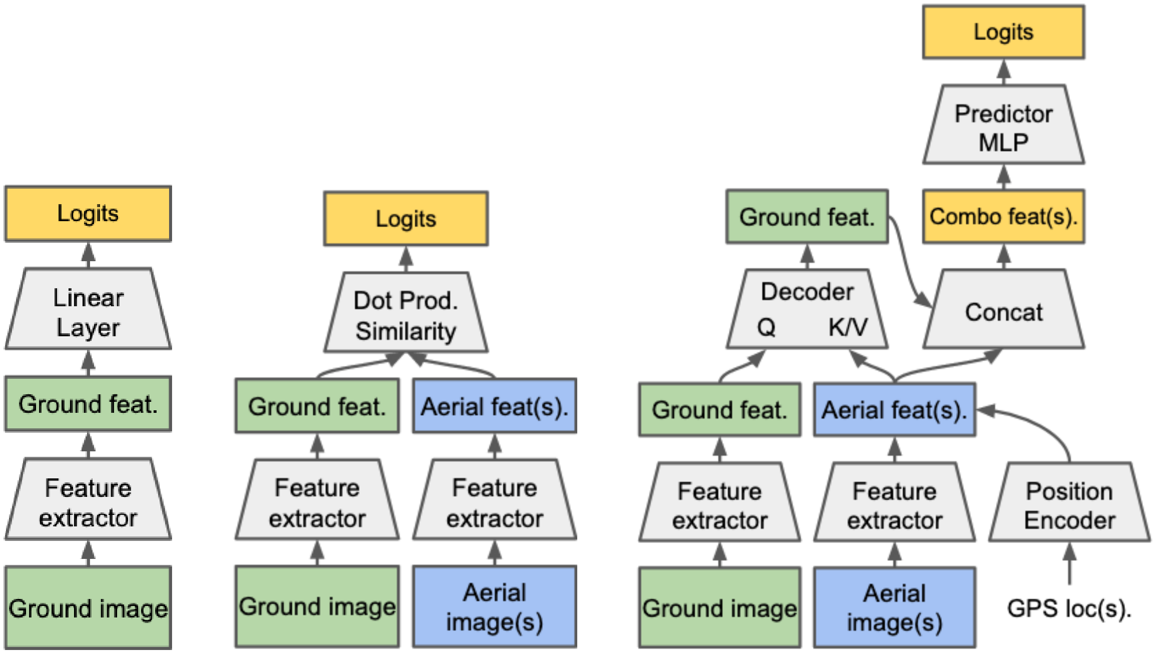
Figure 3.2: **Overview of the latent feature space in retrieval-based image localization.** Images from two different modalities, ground-level photographs and overhead satellite images, are mapped into a semantic latent space where buildings with similar purposes are grouped together.

for positively matched pairs and negative pairs have at least some margin (or difference) and that the similarity between negative pairs is also minimized. In the simplest case negative pairs are selected randomly from within each training batch; more complex sampling schemes are common. For a more detailed explanation of retrieval and metric learning methods please refer to Sec. 2.2.

A subset of metric learning methods concern themselves with the situation where the dataset is not composed of one-to-one matches but instead of grouped (but not necessarily categorized) examples [22]. Following the example of these works and also taking into account the similarity with training models for classification, we proceed by training our retrieval models with the softmax cross entropy loss.

3.4 MetaLoc: Attention-based Image Localization

The core proposal of this work is to condition the ground-level image features by the aerial reference features using a cross-attention operation in the form of a transformer decoder. This approach shares much in common with retrieval-based approaches, with the additional difference of enabling the inclusion of geographic information in the aerial features which is not possible in retrieval-based methods. We term this model *MetaLoc*.



(a) Classif. (b) Retrieval (c) MetaLoc

Figure 3.3: **Detailed overview of the proposed image-localization architectures.**

3.4.1 Computing Image-level Tokens

The inputs of MetaLoc are a query photograph with an unknown location and a database of satellite images with known location, bounding boxes, etc. We first reduce the images to individual tokens for the downstream transformer decoder, $f_g \in \mathbb{R}^n$, and $\bar{f}_a \in \mathbb{R}^n$ respectively. To do so, we use an off-the-shelf CNN terminated with a global average pooling step, estimating a feature vector for each image, $f_g = F(I_g), f_a = F(I_a)$ In practice, we use separate tokenizing networks, F_g and F_a, for each of the two input modalities, ground and aerial imagery.

3.4.2 Location-based Positional Encoding

We convert the known location of the center point for each image in the satellite database into a positional encoding vector. We adapt the strategy described in the original ViT [8] paper, which operates by projecting the coordinates of a fixed grid into onto a random high-dimensional space and then computing the sine and cosine of each value. The result, shown in Fig. 3.4, resembles a smooth, warped grid. Every position has a unique embedding and neighboring positions are smoothly translatable from one another.

In our specific use-case of image localization, we use the GPS coordinates associated with the center point of each reference satellite image. First, these GPS coordi-

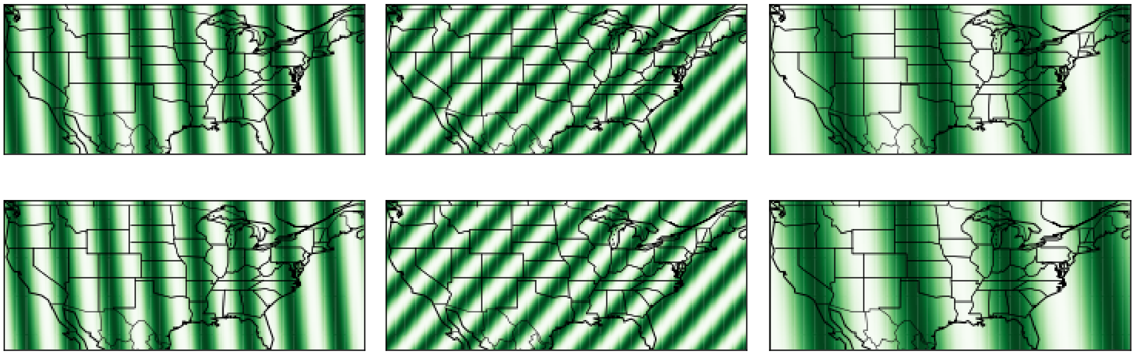


Figure 3.4: **Positional encoding with GPS coordinates.** The top and bottom rows are the sin(.) and cos(.) components of the encoding, respectively. We show each location projected onto a random 3D subspace. In practice the dimensionality is much higher, half the size of the input token, resulting in a total positional encoding size equal to the token size.

nates are projected to a higher dimension, $p' \in \mathbb{R}^{n/2}$, with a random projection matrix whose weights are kept fixed throughout training. The final positional encoding is $p = [sin(p'), cos(p')]$, where $p \in \mathbb{R}^n$. This encoding is added directly to each aerial token, $f'_a = f_a + p$.

3.4.3 Conditional Representation with Transformer Decoders

Our central proposal with *MetaLoc* is to condition aerial image representations on the provided representations of all images involved in the localization of a particular query image. In other words, we seek to refine each representation based on all available context (all satellite images, where they were captured, etc.). This is in contrast with existing retrieval-based work where the similarity between the query image and each satellite image is independent.

To estimate this conditional feature for each satellite image, we propose using a transformer decoder [50]. First, multi-head cross-attention (MHCA) is computed between the query images feature and the collection of satellite image features, $g' = MHCA(g, a)$, which is then followed by a stage of multi-head self-attention and a shallow MLP. The result is a collection of features that have been conditioned on both the query image and the full context provided by the satellite image database.

3.4.4 Predictor Head

We explore two different approaches to making the final prediction of similarity between the query and reference imagery. The first is identical to the retrieval-based approaches which compute similarity via a dot product or the cosine similarity function.

The second approach we explore is to directly estimate the similarity score from the conditioned aerial representation using an MLP. Conceptually this is very similar to how predictions are made following a transformer decoder in other settings where each token is passed independently through a MLP to make some final prediction for each token. In this case, each token is a reference image in our satellite image database which has been conditioned on the query image and the task is to estimate how likely the query was to be captured within a given images region.

3.5 Metrics

We compute a number of recall-based metrics for evaluating the proposed set of image localization models. An example is considered to have been localized correctly "within-k" if the true locations similarity score is among the k largest values, $\textit{within-k}(i) = l_i \in \textit{argsort}(S_i)[0:k]$, and recall is defined as:

$$Recall = \frac{TP}{TP + FN}, \tag{3.1}$$

where TP is the quantity of examples where the true label was *within-k* and FN are those examples that were missed at the same threshold k. Each metric primarily differs in how this score is aggregated and the resulting biases.

Micro Recall @ k Computes the recall score globally. This is biased towards categories (locations) with large numbers of examples, especially if the model already performs well in those areas.

Macro Recall @ k Computes the recall score first for each label, then take a simple average. This metric is biased towards performance in low-population categories due in part them being much more common in our datasets.

Distance Thresholded Micro Recall @ 1 This metric instead considers the minimum distance between the query's true location and the $(k = 1)$ predicted location. If the distance is below some threshold T then it is called a match. Micro Recall is computed as normal.

Table 3.1: **Localization Results.** In the High-res setting, all three methods perform similarly with MetaLoc performing slightly better, especially the top-1 within threshold metrics. Due to the fact that each high-res location has an identical number of examples, the micro- and macro-recall rates for these models are the same, we denote this by *italicizing* the macro-recall scores.

		Micro-recall			Macro-recall			Top-1 < Thresh.		
	Method	1	10	1%	1	10	1%	10km	50	100
Low-res	Classif.	**44.45**	63.31	90.25	**30.33**	50.05	85.09	**49.95**	66.53	73.32
	Retrieval	37.99	**63.74**	90.58	28.51	55.77	88.45	45.65	67.13	75.15
	MetaLoc (Ret)	38.21	63.50	**91.09**	29.13	**56.65**	**89.79**	45.90	**67.86**	**75.87**
	MetaLoc	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
High-	Retrieval	14.04	38.24	90.85	*14.04*	*38.24*	*90.85*	34.62	52.92	63.64
	MetaLoc (Ret)	14.39	38.89	91.05	*14.39*	*38.89*	*91.05*	**36.37**	**56.04**	**66.33**
	MetaLoc	**14.79**	**39.48**	**91.13**	*14.79*	*39.48*	*91.13*	35.64	53.68	63.78

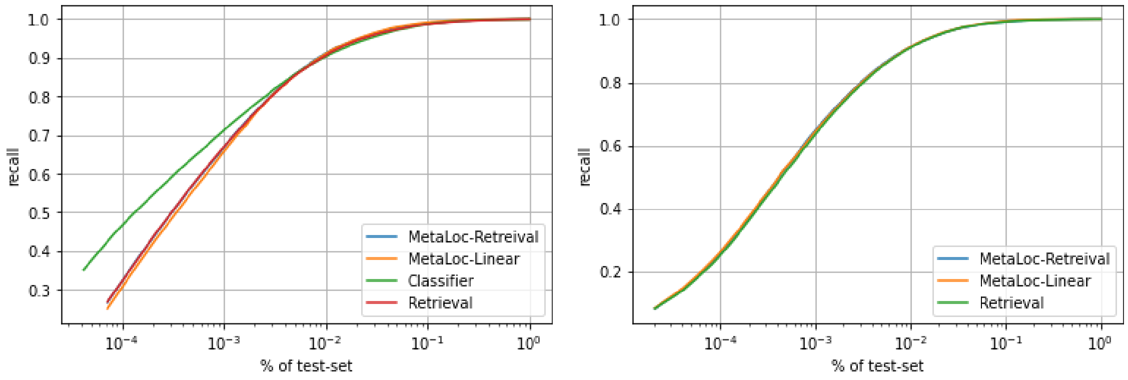
3.6 Experiments

3.6.1 Implementation Details

For our the tokenizing network in our experiments we use a ResNet-18 [12] network pre-initialized with ImageNet weights, with the final linear layer re-initialized to produce 512-d tokens. The transformer decoder used in the MetaLoc models is composed of 4 layers with 4 heads each, internal size 1024, and GeLU activation. The scaling factor c of the Retrieval and MetaLoc (Ret) approaches is empirically set to 3. The classifier model is trained with approx. 23k categories.

We train and test with lateral cutouts from streetview panoramas; during training we sample a random 60° window whose center-point is facing laterally left or right away from the road with a random offset of ±45°, during testing no random offset is used. When training on the high resolution satellite images from CVUSA-500k, a random 512x512 cutout is used; and during testing the central 512x512 pixel region. At all phases of training and testing, images and cutouts are downsampled to 256x256 using bilinear interpolation. When training with low-resolution satellite imagery, we use H3 zoom-level 5 sized patches from a Sentinel-2 basemap resized to 256x256 pixels.

All models are trained for 100,000 iterations with batch size 512, using the Lamb [58] optimizer and 1-Cycle [42] learning rate schedule with a maximum learning rate of 0.005. Each model is trained on a single NVIDIA V100 at half-precision for around 2 days.



(a) Low-res data

(b) High-res data

Figure 3.5: **Same-area Localization Recall Curves.**

Table 3.2: **MetaLoc Results with Various Input Field of View Settings.**

	Micro-recall			Macro-recall			Top-1 < Thresh.		
FoV	1	10	1%	1	10	1%	10km	50	100
30°	26.43	50.27	82.93	20.59	43.67	79.63	32.07	50.75	59.69
45°	33.05	58.97	88.71	25.44	51.83	86.32	39.58	60.60	69.29
60°*	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37

3.6.2 Results

First, we evaluate the performance of each of the proposed image-localization approaches. We focus on two settings: 1) Low-res, which uses coarse-scale reference imagery sourced from Sentinel-2, and 2) High-res, which uses fine-scale reference imagery sourced from Bing Maps.

In Tab. 3.1 and Fig. 3.5, we find that in the low-res setting, the classification and retrieval methods perform best in terms of micro- and macro-recall scores, especially for lower values of k. Where MetaLoc performs best appears to be at higher values of k. This may indicate that the model is more able to localize the query image to regions that are close to, if not exactly, where the image is from. This insight is supported by the fact that MetaLoc is able to localize the top-1 prediction to within 10km approx. 45% of the time, and within 100km approx. 75% of the time.

In the high-res setting, all three methods considered perform similarly, with MetaLoc outperforming retrieval by a small margin in every metric.

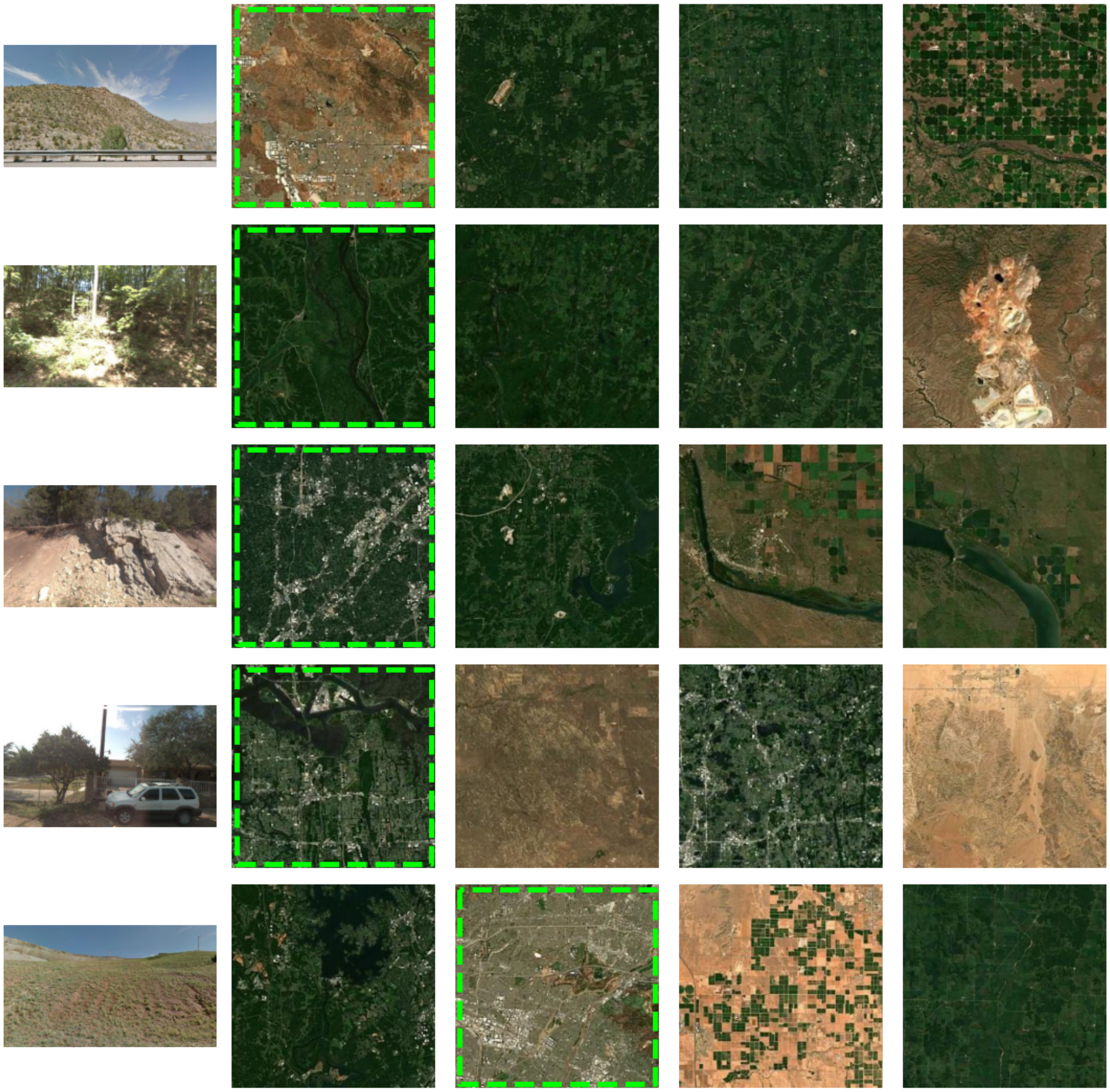


Figure 3.6: **Sample predictions ordered by estimated similarity between query and reference imagery.**

3.6.3 Ablation Study

We compare our proposed approach against a number of baselines. For the coarse setting, we compare against a simple classifier (referred to as PlaNet-style [53]), a simple retrieval model, a retrieval model with the addition of a transformer decoder, and finally with the addition of a positional encoding scheme. We compare the same model settings for the fine setting, sans the classification baseline.

A number of specific choices regarding the architecture were made empirically. Here we explore additional settings to those and evaluate the impact those choices have on *MetaLoc's* performance. In each of these tables, the setting marked with an asterisk (*) is

Table 3.3: **MetaLoc Results from Various Tokenizer Networks.**

	Micro-recall			Macro-recall			Top-1 < Thresh.		
Tokenizer	1	10	1%	1	10	1%	10km	50	100
ResNet-18 *	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
ResNet-34	37.47	63.17	91.36	27.98	55.20	89.35	44.50	66.25	74.62
ResNet-50	34.12	60.82	90.89	26.61	53.97	89.10	42.06	65.42	74.19

Table 3.4: **MetaLoc Results from Various Decoder Parameters.**

		Micro-recall			Macro-recall			Top-1 < Thresh.		
Decoder		1	10	1%	1	10	1%	10km	50	100
Token-D	128	26.24	49.98	83.02	17.60	40.37	79.51	32.79	55.07	65.40
	256	31.03	56.81	88.66	22.62	49.25	86.44	38.24	60.81	70.70
	512 *	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
Depth	1	34.71	60.24	89.95	26.48	52.88	87.61	41.51	63.02	71.55
	2	34.72	60.83	90.24	26.28	53.36	88.08	41.56	63.79	72.56
	4*	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
	8	35.75	62.43	91.42	26.98	55.05	89.77	43.18	66.31	74.93
Heads	1	36.16	62.22	90.85	27.65	54.56	88.88	43.38	65.80	74.14
	2	35.37	61.27	90.49	26.47	53.05	88.46	42.51	64.42	73.14
	4*	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
	8	35.85	62.49	91.04	27.48	54.73	89.00	43.04	65.55	74.21

the default used in all other tables.

Field of View The default field of view (FoV) value of 90°was chosen to maximize the visual keypoints visible in each cutout. However in the wild, images can feature a wide variety of FoV values. In Tab. 3.2, we evaluate 4 such settings: 30, 60, 90, and 120 degrees.

Representation model parameters The tokenizer network can take a number of forms, most often a CNN. For its simplicity and flexibility, we selected a ResNet-18 model pretrained on the ImageNet image classification task to be the representation network for both ground and satellite imagery. In Tab. 3.3 we evaluate other choices in the ResNet family as well as a Transformer model.

Decoder parameters Transformer decoders feature a large number of individually tuneable parameters, the most significant of which control the models size. In Tab. 3.4, we evaluate the effect that model depth, model width, and number of transformer heads each has on localization performance.

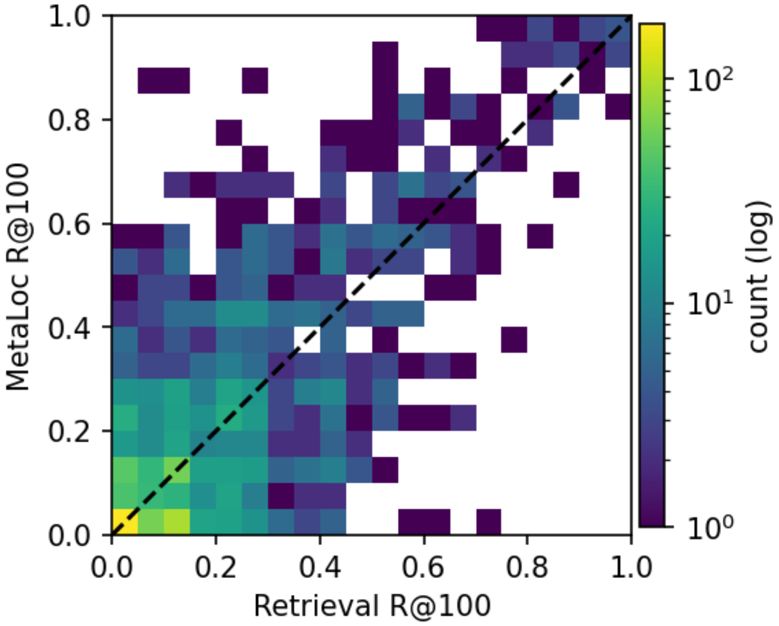


Figure 3.7: **Relative recall rates of the MetaLoc and Retrieval approaches.** The Recall@100 rates for each geographic bin are compared against each other in a 2D histogram. Each histogram cell represents the occurrence rate of a particular pair of recall scores. Much of the density lies on or above the diagonal, implying that in many of the geographic cells the MetaLoc model is a direct improvement over Retrieval.

3.6.4 Comparing Recall Rates

In this section we visualize the relative distribution of recall scores for our proposed MetaLoc model and the baseline Retrival approach. For each H3 cell in the Cross-area test set, we compute the Recall@K scores for both models. In Fig. 3.7, we show the 2D histogram comparing the relative distribution of these scores. We observe that much of the density in this plot is above the diagonal, which directly shows that for more cells than not, the MetaLoc model improves on the Retrieval baselines metric.

3.7 Discussion

In this chapter we present and evaluate a unified approach to image geo-localization. We extend the most common retrieval-based localization dataset to include satellite imagery intended to support classification-based methods. Further, we propose an approach which uses a transformer decoder to estimate the likelihood that an image was captured within a given satellite image. This work establishes a number of baseline results that we hope inspire the community to compete against.

Chapter 4 Generalizing to Unseen Areas

Humans are excellent at localizing images from places they have never been using contextual clues and other hints present in images. There's even a game where users test their global image localization skill[1]. Ideally, machine learning models would be able to perform similarly well, even in areas and situations where they lack training data.

Many parts of the world are over- or under-imaged. For example, New York City and Paris have been extensively well documented by historians and tourists while many of the un-inhabited parts of the world may have only ever been imaged by satellites. More difficult still, places change over time. New buildings are constructed, the addition of a dam will dramatically alter a waterline, and forest fires can leave communities and their surrounding biomes disrupted for decades. Localizing accurately in all of these settings is arguably more important than localizing the images in the well-managed datasets on which these models are trained because these are real situations where an image localization model could be deployed and required to be accurate.

Existing approaches vary in how or whether they even generalize to unseen areas. For example, classification models are unable to localize images to unknown areas by default; since each location is treated as a separate category and as such the model will have never been presented an example of that category. In contrast, retrieval methods keep a reference database to localize against which can potentially be exchanged for a new database covering a different area or upgraded to an extended one covering both. This practice has not been well studied however, only recently [66] has a dataset been presented to specifically evaluate this, and even then only at the extremely fine scale of meter-level accuracy within urban environments.

Given that the proposed hybrid approach is based on a transformer decoder, we can control the amount of geographic context included in the database by changing aspects of the positional encoding. In retrieval methods, nothing about the satellite images location or its position relative to other satellite images is included. We propose introducing that kind of information and in this chapter will explore a number of possible approaches.

4.1 Same- vs. Cross-Area

We seek to evaluate the ability of our proposed approach to generalize to unseen areas. Inspired by VIGOR [66], we split CVUSA-500k into two the eastern and western United

[1] `https://www.geoguessr.com/`

Table 4.1: **Same- vs. Cross-Area Splits.** The aerial images are H3 resolution-4 cutouts from a Sentinel-2 basemap covering the same area as the CVUSA dataset (CONUS + Alaska). East-West split determined by median longitude in CVUSA (-91.21).

(a) Matching

		Same-Area		Cross-Area	
		Region	Quantity	Region	Quantity
Train	Ground	All	979K	West	514K
	Aerial	All	489K		257K
Test	Ground	All	50K	East	514K
	Aerial	All	25K		257K

(b) Classification

		Same-Area		Cross-Area	
		Region	Quantity	Region	Quantity
Train	Ground	All	979K	West	514K
	Aerial	All	4.3K		2.8K
Test	Ground	All	50K	East	514K
	Aerial	All	4.3K		1.4K

States by the median longitude value in the CVUSA dataset, -91.21°, producing an even split between training and testing. In Tab. 4.1 we present the specifics of these splits.

We also seek to evaluate the situation where there is a limited amount of data available in the held out region. To evaluate MetaLoc and the baseline Classification and Retrieval methods for this low data setting, we further augment the Cross-area split by transferring small amounts of data from the test split to the train split. Specifically, for 4 different thresholds (1, 2, 5, and 10) we identify bins in the test split with *more examples* than the threshold, and from each bin we sample *threshold* examples to transfer to the training set. Bins with fewer examples than the threshold are dropped from the test set entirely. We do so because we require unseen examples from each bin in the test set, while also having at *threshold* examples per bin in the training set. Note, the *threshold = 0* setting is equivalent to the base Cross-area split. We summarize the final train/test dataset sizes for each threshold in Tab. 4.2.

4.2 Positional Encoding Considerations

The positional encoding component of a transformer model is responsible for representing the positions of each token in the input sequence as a signal that can be used by the self-attention modules to reason about the relative and absolute positions of each token.

Table 4.2: **CVUSA-500k train and test dataset sizes for Semi-Cross setting.** To produce each split, some number of examples from each location bin in the testing area are transferred to the training set. In the case where there are not enough examples in the bin to transfer to training while leaving at least one example for testing, the entire bin is discarded.

		Examples per bin from test area			
		0	1	5	10
East→West	Train	514K	520K	541K	562K
	Test	514K	508K	486K	460K
	Total	1.28M	1.28M	1.27M	962K
West→East	Train	514K	517K	529K	543K
	Test	514K	511K	499K	485K
	Total	1.28M	1.28M	1.28M	1.28M

In the original NLP context [50] this was the indices denoting where each tokenized word was located in the sentence. In the computer vision application of Transformers [8], this is typically the X, Y position of each tokenized sub-image in the full image. The positional encoding approach used by our *MetaLoc* model (Sec. 3.4.2) is conceptually very similar to the image transformer setting, except the resulting encoding vectors denote where on the surface of the Earth each satellite image is located. In this section we describe different approaches to representing this location information, additional position-adjacent information that can be useful to encode, and a detailed description of the positional encoding module. Other more involved approaches to this have been proposed, but are outside the scope of this work [32].

4.2.1 GPS Coordinates

Latitude-Longitude coordinates provided by the Global Positioning System are the form of location encoding that people are typically most familiar with. They are part of a spherical coordinate system with a fixed radius where Longitude, $Lon \in (-\pi, \pi)$, represents the azimuth angle around the Earths equator and Latitude, $Lat \in (-\pi/2, \pi/2)$, represents the polar angle from the equator.

4.2.2 Earth-Centered, Earth-Fixed Coordinates

GPS coordinates have a number of drawbacks including: 1) scaling issues as locations get further from the equator (one degree of Longitude represents different distances in km depending on Latitude), 2) discontinuity at the "seams" located at the poles and along the $-\pi \rightarrow \pi$ Longitudinal transition line, 3) and a fixed radius that is not able to encode elevation changes in the geography. These drawbacks introduce small hurdles to learning about geographic positions by an image localization model. An alternative encoding that overcomes these issues at the expense of being easily human interpretable is the Earth-Centered, Earth-Fixed[2] (ECEF) coordinate system. Converting from GPS results in a three-dimensional coordinate $[x, y, z] \in \mathbb{R}^3$ and is straightforward:

$$\begin{aligned} x &= r * \cos(Lon) * \sin(Lat) \\ y &= r * \sin(Lon) * \sin(Lat) \\ z &= r * \cos(Lat) \end{aligned}$$

where r is a fixed radius for the Earth, typically 1. These new coordinates are the Cartesian coordinates corresponding with the r-radius unit-sphere and they overcome all three of the stated issues with GPS coordinates.

4.2.3 Other Positional Possibilities

Positional encodings are not strictly limited to spatial and ordering related information. Other important metadata, such as spatial scale, temporal information, and details of a tokens source can all be included in the encoding vector.

Geospatial Scale The size or scale of the area described by a single token can be an important cue for a localization model, especially in settings where the localizable regions are not uniformly sized and in hierarchical localization schemes. There are two main approaches to this: 1) encode the scale factor directly, or 2) encode the bounds of the region. In this work we opt for the second approach, whenever scale is included as part of the positional encoding, it is encoded as the concatenation of the coordinates of the upper left and lower right corners of the regions bounding box: $[Lat_{ul}, Lon_{ul}, Lat_{lr}, Lon_{lr}]$, and similar for ECEF coordinates.

Data Source In settings where there are more than one source of input tokens, it may be important to encode from which source each token was sourced. Each source of tokens typically will possess their own individual details and quirks that need to be included in the localization models decision making, such as resolution, framerate, or spectral range (Chapter 6). To encode these, for each source we propose adding a separate learnable vector for each possible data source.

[2] https://en.wikipedia.org/wiki/Earth-centered,_Earth-fixed_coordinate_system

Table 4.3: **Positional Encoding Ablation Study.**

	Micro		Macro		Thresh	
	10	1%	10	1%	50km	100
None	4.80 ±9.20	7.59 ±10.58	5.10 ±6.72	7.79 ±7.38	10.71 ±5.99	26.36 ±15.81
GPS	4.89 ±9.40	8.12 ±13.37	4.59 ±7.38	7.98 ±9.28	12.87 ±7.78	30.80 ±16.62
+ BBox	2.10 ±2.98	4.27 ±4.38	3.55 ±4.68	6.16 ±6.05	12.47 ±7.44	26.47 ±16.11
ECEF	3.65 ±2.82	6.17 ±4.05	5.57 ±6.24	8.05 ±6.90	12.92 ±12.59	24.14 ±17.65
+ BBox	3.16 ±2.50	5.35 ±3.81	5.11 ±5.20	7.67 ±6.46	13.41 ±7.43	29.75 ±11.52

4.2.4 Positional Encoding Method

We convert the known location of the center point for each image in the satellite database into a positional encoding vector. We adapt the strategy described in the original ViT [8] paper, which operates by projecting the coordinates of a fixed grid into onto a random high-dimensional space and then computing the sine and cosine of each value. The result, shown in Fig. 4.1, resembles a smooth, warped grid. Every position has a unique embedding and neighboring positions are smoothly translatable from one another.

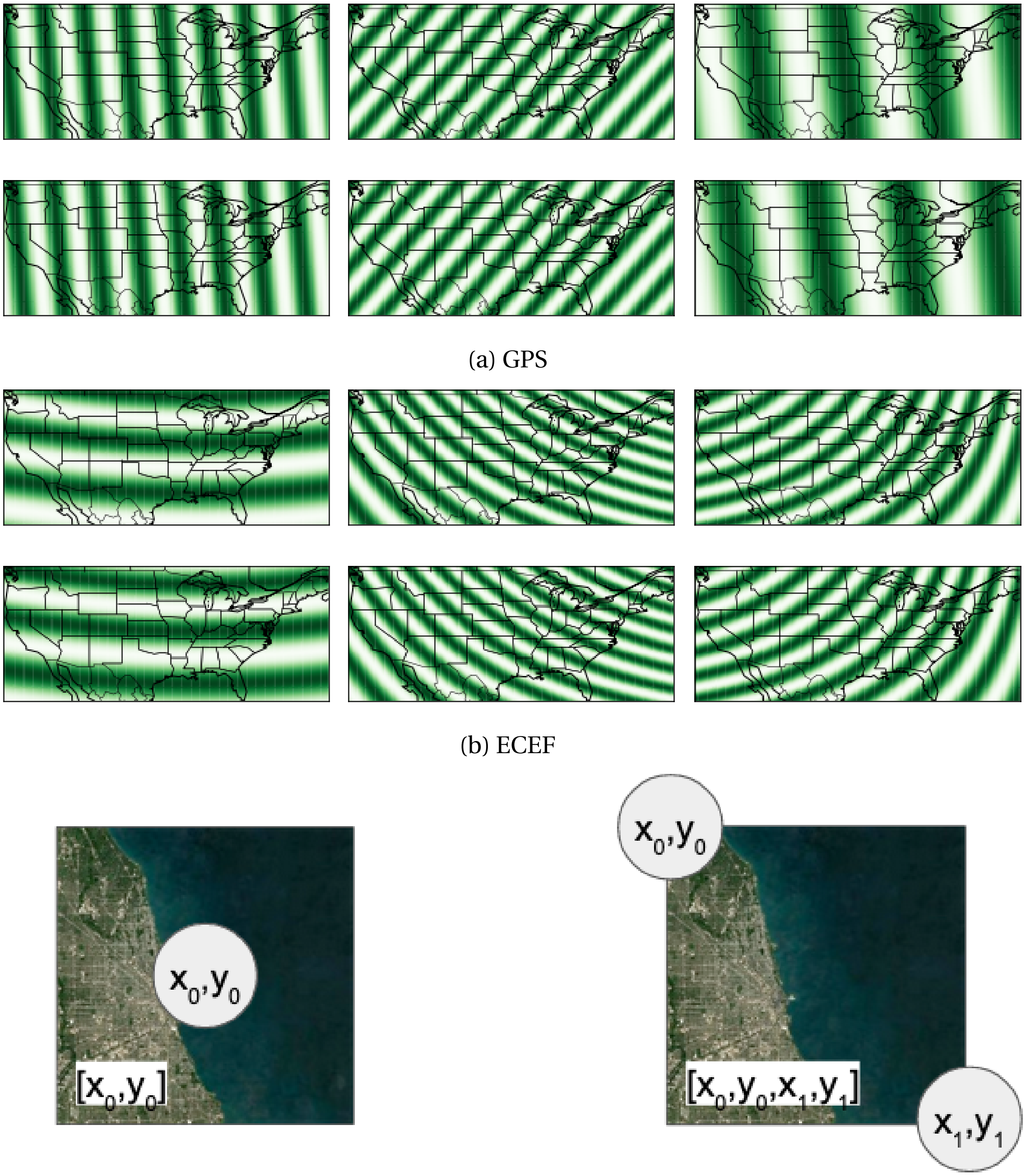
In our specific use-case of image localization, we use the GPS coordinates associated with the center point of each reference satellite image. First, these GPS coordinates are projected to a higher dimension, $p' \in \mathbb{R}^{n/2}$, with a random projection matrix whose weights are kept fixed throughout training. The final positional encoding is $p = [sin(p'), cos(p')]$, where $p \in \mathbb{R}^n$. This encoding is added directly to each aerial token, $f'_a = f_a + p$.

4.3 Experiments

In each of the following experiments we focus exclusively on the situation where the reference satellite image database is composed of Sentinel-2 basemap images, what is referred to in Tab. 3.1 as the "Low-res" setting. Unless otherwise stated, we follow the same experimental and model settings as in Sec. 3.6.

4.3.1 Ablation Study

To evaluate the effectiveness of the choice in positional encoding, we perform a simple ablation study where we train across 10 splits of the CVUSA-500k dataset. To arrive at these splits, we first divide the dataset into 10 approximately equally sized and spatially contiguous regions by performing k-means clustering (with k=10) of the ground truth spatial coordinates; the training set of each fold is 9 of these regions with the tenth held



(a) GPS

(b) ECEF

(c) Center encoding

(d) Box encoding

Figure 4.1: **Example positional encodings given different location representations.** The rows of (a) and (b) are the sin(.) and cos(.) components of the encoding, respectively. We show each location projected onto a random 3D subspace. In practice the dimensionality is much higher, half the size of the input token. (c) and (d) show where the coordinates for the center point and bounding box based positional encodings are sourced from.

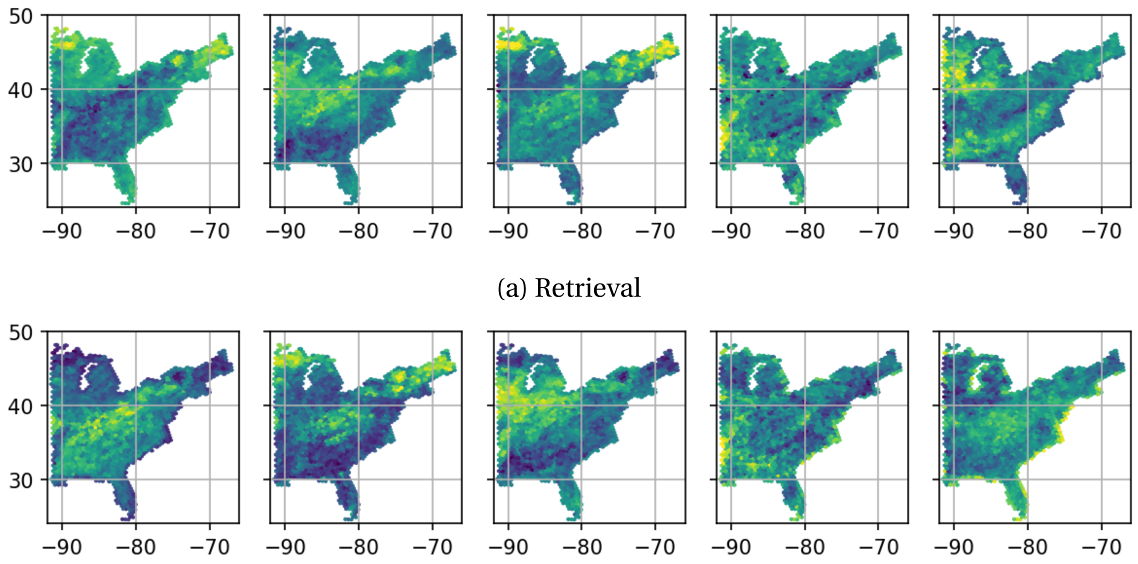
Table 4.4: **Semi-Cross Localization Results.**

ex/bin		West→East				East→West			
		Macro		Thresh		Macro		Thresh	
	Method	10	1%	50km	100	10	1%	50km	100
=0	Classification	*Not Applicable*				*Not Applicable*			
	Retrieval	**2.55**	**7.55**	12.60	25.79	0.79	4.59	6.51	13.31
	MetaLoc (Ret)	2.14	7.31	10.62	22.91	**1.21**	**6.86**	**7.98**	**15.02**
	MetaLoc	1.81	6.00	**14.01**	**30.63**	0.67	5.06	6.17	13.08
=1	Classification	2.30	7.34	4.34	7.17	3.85	7.18	5.92	8.19
	Retrieval	6.98	14.65	24.80	37.89	**4.96**	14.83	**20.65**	**31.25**
	MetaLoc (Ret)	**7.48**	**16.83**	26.33	39.73	3.23	**15.45**	18.27	30.57
	MetaLoc	6.33	14.39	**28.62**	**40.15**	3.29	14.15	12.64	22.97
=5	Classification	8.31	19.65	20.06	25.20	12.46	22.92	18.30	22.74
	Retrieval	13.64	26.51	37.28	49.53	**14.31**	33.05	31.12	44.32
	MetaLoc (Ret)	**14.13**	**28.94**	**39.85**	**51.82**	11.87	**34.21**	**32.67**	**46.65**
	MetaLoc	10.30	22.43	28.31	41.46	8.72	27.15	22.22	35.30
=10	Classification	13.68	27.78	31.24	36.28	17.57	33.03	26.46	31.95
	Retrieval	18.84	35.11	**43.47**	54.43	**20.12**	43.72	**38.12**	**51.74**
	MetaLoc (Ret)	**19.64**	**36.24**	46.82	**57.08**	18.01	**46.42**	37.87	51.20
	MetaLoc	13.60	27.98	36.93	47.81	12.41	34.79	24.00	37.04

out for testing. We compare a baseline *MetaLoc* model trained without positional encoding against four different settings covering the combinations of GPS/ECEF coordinates and scale-less/bounding-box encoded. In Tab. 4.3 we present the mean and standard deviation of the micro-, macro-recall, and top-1 within threshold metrics. We observe that in terms of micro-recall performance GPS-only performs the best at the cost of increased variance. A similar pattern arises in the macro-recall results where ECEF-only and GPS-only are best especially at the top-1% threshold. In both micro- and macro-recall, the addition of bounding box coordinates either do not help or actually hinder localization performance. This pattern does not hold for the distance-thresholded recall metric, where the bounding box based encodings perform similarly to their coordinate-only pair, and the ECEF-bounding-box encoding performing especially well. It is unclear whether there is a best approach to encoding the position of each geo-spatial token, and the choice of approach needs to be tailored to the specific problem at hand.

4.3.2 Low- and No-Data Generalization

In practice there are situations where there exists a large amount of training data for some regions, and little to none for the regions where the model is intended to be deployed for



(a) Retrieval

(b) MetaLoc

Figure 4.2: **Geographic patterns emerge in satellite-view representations.** The models trained for the West→East setting are used to predict feature representation for each Sentinel-2 basemap image. These are then decomposed into their principal components using SVD. Each column represents a different component.

localization. This can arise for any number of reasons, including the data selection bias in due to population density and the source of training data chosen. We evaluate four different methods for performance in these settings by transferring a variable amount of data (between 0 and 10 examples per sub-region) from each of the Cross-Area test sets to the corresponding training set. In Tab. 4.4, we show results for each of the described methods on each of the Semi-Cross splits.

In general, the Retrieval and MetaLoc (Ret) approaches perform similarly to each other in terms of macro recall regardless of the number of additional examples from the test region that are provided. Similarly for the MetaLoc model and top-1 retrieval at 50km.

As the number of examples increases, the PlaNet-style classification models quickly catches up, but does not surpass, the other approaches. It is possible that at even higher settings the classification approach would perform more similarly to the full-data setting in Tab. 3.1.

4.3.3 Learned Satellite-View Representations

Both retrieval-based and MetaLoc image localization are based on work in metric learning which was originally conceived of to learn compact representations for imagery without the use of labels. Because there is an underlying structure (i.e. the images are geographically close) to the satellite imagery our models were trained on, we can visualize the satellite image representation each model learns as a side-effect of learning to localize.

We construct this visualization by first decomposing the learned representation using Principle Component Analysis (PCA). In Fig. 4.2, we display the first several channels resulting from this decomposition. We find that the learned representations encode distinct things as they do not immediately resemble each other, and that the learned features are geographically coherent. Regions such as the Appalachian Mountains appear as specific high-spots in some of the components, implying that geographic features such as mountains and forests play a major role in what is encoded.

4.4 Discussion

In this chapter we present a detailed discussion on positional encoding for transformers trained on spatial data, including for image localization. We also explore the impact the positional encoding has on the *MetaLoc* models ability to generalize to unseen regions, as well as an in-depth evaluation of no- and low-data situations for a the roster of models we consider in this document. Finally we qualitatively examine the learned satellite-view representations, both the unconditioned ones learned by the Retireval and *MetaLoc* models, as well as the conditional representations resulting from cross-attention with a query ground-level image.

Chapter 5 Reference Data Scale

The size of the reference areas against which image localization is performed has a direct impact on the precision and accuracy of the predictions themselves. In an extreme setting, a system which simply localized all images to a single Earth-sized bin would be accurate for 99.999... % of images[1] and have effectively 0 precision. On the other extreme, centimeter-scale location bins would be extremely precise in the rare case that they could be accurately localized. Logically, there must exist some medium point which balances the trade-off in precision and accuracy inherent in global image-localization.

The optimal size of a bin for localization is highly varied and dependent on a number of environmental factors that affect how easily recognizable the location is. Distinct geography makes regions such as the Rocky Mountains and the Sahara Desert highly identifiable. Many countries and states are recognizable by architectural details, similarly for cities and specific neighborhoods. Further, in otherwise difficult to precisely locate regions there exist clear landmarks which make localization much easier, such as specific and well-known rock formations, monuments, and buildings.

All of this motivates a multi-scale approach to localization which proceeds from coarse-to-fine scale bins. Specifically, one where we first localize an image into a coarse set of bins, then refine the localization by making predictions against only those sub-regions within the identified parent area. In this chapter, we propose training multiple models for this task, one per binning-scale and making predictions from them in sequence.

5.1 Satellite Imagery Scales and Resolutions

To support our multi-scale image localization experiments, we construct multiple scales of localization targets and associated reference satellite imagery sources. We present two strategies, one for coarse-localization and one for fine-:

Coarse We continue with the binning strategy we used in Chapter 3 where the world is divided into hexagonal cells of a given resolution using the H3 spatial library [2]. In those experiments we used the level-5 resolution, which for the CVUSA dataset produced $23.7k$ hex-cells tiling the continental United States. For this experiment, we introduce level-4 binning, resulting in $4,333$ cells.

[1] The counter examples being images captured on the Moon, on Mars, and by various probe missions.

[2] Uber H3: `https://h3geo.org`

Table 5.1: **Summary of satellite imagery sources and their spatial resolutions.** As the zoom level for each source increases, the spatial scale of the pixels (m/pix) in each corresponding image decreases as does the spatial extent of the entire image (km^2). In the coarse setting, this also corresponds with an increase in the number of distinct locations. Conversely, in the fine setting the number of distinct locations is constant.

(a) Coarse (Sentinel-2 @ 256x256 px)

Zoom	Locs	m/pix	km^2
H3-4	4,333	176.59	45.2
H3-5	23.7k	66.75	17.0

(b) Fine (Bing @ 512x512 px)

Zoom	Locs	m/pix	km^2
Bing-14	514k	9.5	4.8
Bing-16	514k	2.4	1.2
Bing-18	514k	0.6	0.3

Paired with each of these cells is an image which fully encompasses the hex-cell, sourced from the red-green-blue channels of the Sentinel-2 satellite platform. The source image features a ground-sampling-distance of 10 meters-per-pixel, and the images corresponding with each level of binning are downsampled from this to an image resolution of 256x256 pixels. For more details, see Tab. 5.1a.

Fine At fine scales, we take a different strategy and revert back to retrieval based localization where for each GPS location in the dataset there is a corresponding satellite image. This results in large areas that are left without image coverage, however this is a practical requirement for our experiments. The combined size of the fine-scale reference imagery is around 300GB gzip-compressed and full coverage of the continental United States would be many times larger.

Three zoom-levels of satellite imagery was collected, at Bing-14, -16, and -18. Each of these images was sourced from Bing maps circa 2015 [55] and is 512x512 pixels in size. For more details including ground-sampling distance see Tab. 5.1b.

5.2 Experiments

5.2.1 Scale Study

First, we consider the impact that image scale has on localization accuracy. To do so we vary reference imagery configuration following the sizes described in Tab. 5.1. For each of these we train the full set of models described in Chapter 3, all with identical hyperparameters. The results of this study can be found in Tab. 5.2.

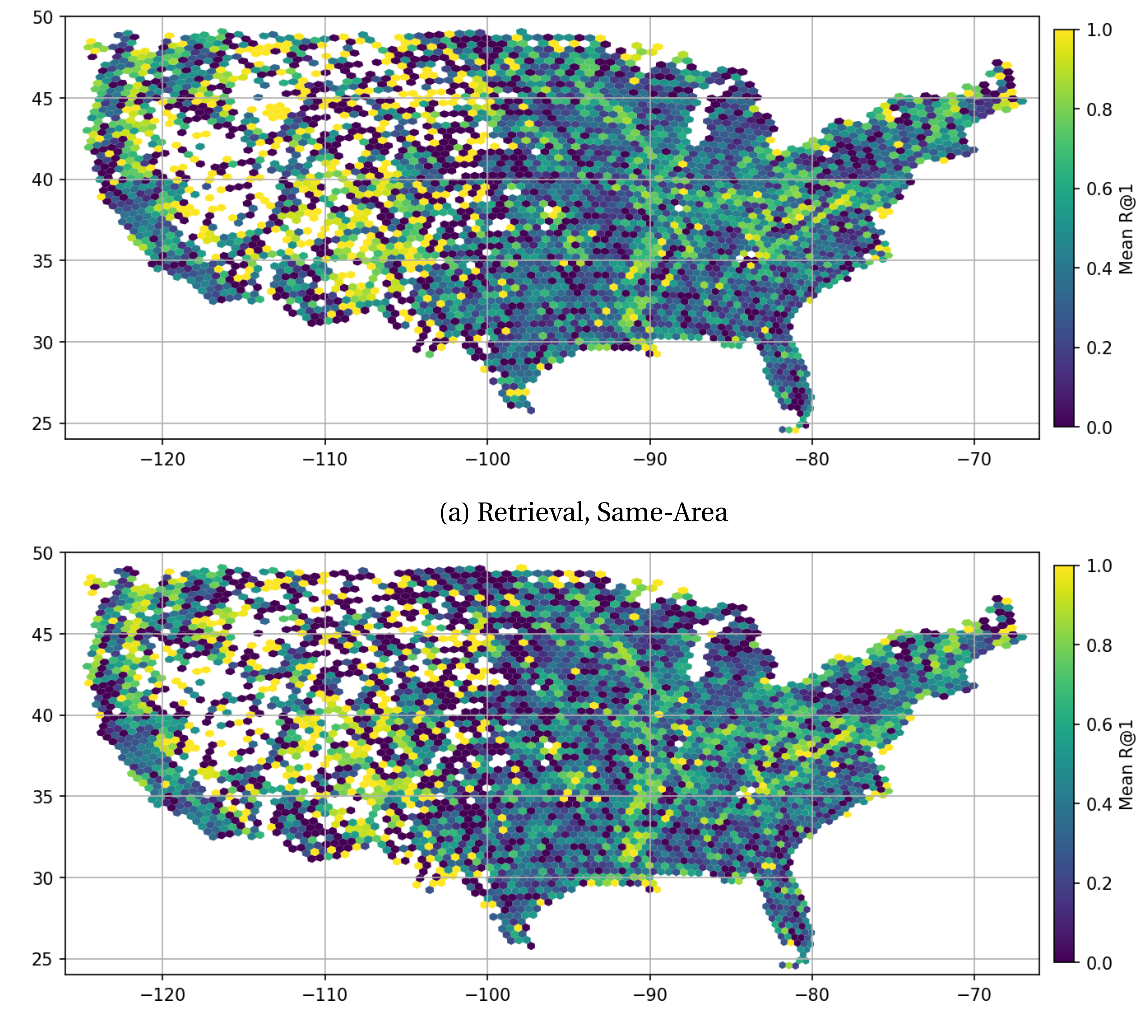
Table 5.2: **Localization Results Using Various Imagery Scales.**

			Micro-recall			Macro-recall			Top-1 < Thresh.		
		Method	1	10	1%	1	10	1%	10km	50	100
H3	=4	Classif.	61.31	83.11	93.42	51.55	77.35	90.59	61.82	74.25	81.57
		Retrieval	30.44	55.05	73.55	21.93	46.87	69.05	31.22	46.60	58.20
		MLoc (Ret)	50.11	77.88	90.28	42.26	74.51	89.92	51.36	69.88	79.47
		MetaLoc	17.06	35.60	53.78	10.85	28.19	47.71	17.60	29.22	40.41
	=5	Classif.	44.45	63.31	90.25	30.33	50.05	85.09	49.95	66.53	73.32
		Retrieval	37.99	63.74	90.58	28.51	55.77	88.45	45.65	67.13	75.15
		MLoc (Ret)	38.21	63.50	91.09	29.13	56.65	89.79	45.90	67.86	75.87
		MetaLoc	36.55	62.22	91.02	27.84	54.70	89.25	43.53	66.02	74.37
Bing	=14	Retrieval	1.36	6.16	52.67	1.37	6.17	52.66	10.41	24.46	34.90
		MLoc (Ret)	13.66	40.45	92.28	13.70	40.51	92.31	41.23	62.83	72.30
		MetaLoc	4.22	14.93	64.66	4.23	14.94	64.71	17.72	27.66	34.82
	=16	Retrieval	14.09	38.03	91.17	14.09	38.09	91.21	34.72	53.29	63.72
		MLoc (Ret)	15.00	39.63	91.30	15.04	39.66	91.31	36.64	56.12	66.63
		MetaLoc	4.53	15.20	67.36	4.54	15.22	67.39	15.62	25.56	33.26
	=18	Retrieval	16.51	40.61	92.78	16.52	40.74	92.81	33.42	48.26	59.03
		MLoc (Ret)	18.31	42.55	92.80	18.34	42.59	92.80	34.69	51.51	62.08
		MetaLoc	6.18	19.00	77.44	6.17	19.03	77.46	15.02	25.17	33.51

5.2.2 Recall Rate vs. Location as Scale Varies

Next we consider the geo-dependence of recall rate for the Retrieval and MetaLoc models. To do so, we compute Recall@K rates individually for each coarse-localization overhead image. We compute the similarity between each query image and each image in the reference database, sort by similarity, and if the true location is within the top K choices consider it a match. We then count the occurrences and divide by the total images within that cell.

In Fig. 5.1, we observe that they perform very similarly, as suggested by the results in Tab. 3.1. Interestingly, MetaLoc appears to generalize to unseen regions well, including in rural areas, the likely source of the improvement in that table. Of note, a number of bins between longitudes -120 and -100, corresponding with the Great Plains and Rocky Mountain regions of the United States, frequently only have a single test example within them and as a result are quite noisy, either with recall 1.0 or 0.0.



(a) Retrieval, Same-Area

(b) MetaLoc, Same-Area

Figure 5.1: **Mean recall rates as a function of location.**

5.3 Discussion

In this chapter we evaluated the impact of reference imagery scale on localization accuracy. To do so we introduced several additional scales of reference imagery from both the coarse and fine satellite image settings and evaluated a large collection of localization models.

There is a wide variety of future work that can proceed from these findings. For example, we proposed adding metadata describing the reference images spatial scale to the positional encoding and training a single MetaLoc model on multiple scales of satellite imagery. Doing so would enable applying a single model to the hierarchical localization approach we proposed, as well as create a natural platform from which to experiment with hierarchical localization loss functions and metrics.

Chapter 6 Multi-modal Reference Imagery

As the quantity and quality of remote sensing imagery increases, so too does the difficulty in processing it quickly and efficiently. Much of the data collected by satellites is multi-spectral imagery where each channel corresponds to a particular range of frequencies of light. Each spectrum is chosen to highlight certain details visible from space, such as cloud cover, man-made structures radiating heat from the sun, or visible spectrum light as we are used to. We propose considering each of these as separate images, or modalities, of the same subject.

Multiple views from varying input modes, such as images (greyscale, RGB, and/or multi-spectral), depth, text, etc., each provide a unique view of an example and can each be leveraged to produce more detailed and higher performance models. This can be especially useful in settings where some modes are cheap to acquire while others are expensive, such as the tension that exists in autonomous driving between the cost and quality of LiDAR (Light Detection and Ranging) sensors, and in situations where some of the collected inputs might be occluded or otherwise corrupted

Training image-localization models that are capable of performing even when some modalities are missing can be useful in settings where some modes are inexpensive to acquire while others are costly. For example, combining low-resolution imagery that is updated frequently with high-resolution imagery that is not. Similarly in situations where some of the collected inputs might be occluded or otherwise corrupted.

In this chapter we propose a multi-modal transformer approach where each available mode is converted to its own token stream and combined into a single input sequence. Predictions are then made on the entire sequence by as single transformer model and aggregated across modes into a final prediction sequence. This allows for there to be a variable number of sequences or modes at all phases of training and inference, while allowing the model to learn correspondences between features throughout. As a proof-of-concept, we evaluate our proposed approach on the Onera Sentinel-2 Change Detection (OSCD) dataset [3] by experimenting with selectively withholding subsets of the available 13 spectral bands during training and evaluation. Additionally, we apply the proposed multi-modal training approach to image localization following a similar channel-withholding scheme.

Current approaches that address the problem of missing modalities either learn to hallucinate [25, 26, 28, 40, 48, 57], such as depth from RGB imagery, and then feed those predictions to downstream models trained on a complete set of modes. This makes the assumption that the missing modality is recoverable and that artifacts introduced by the recovery process won't bias the resulting predictions. In contrast, our proposed approach makes predictions directly from the available inputs at inference time, without the need for hallucination. Other methods learn a separate model for each possible modality [10,47,61], however this comes with a sizable overhead as the number of models grows with the number of combinations of input modes and target tasks.

Our contributions include:

- Introducing a variant of the Vision Transformer that is trained for multi-modal semantic segmentation and can handle missing input modes;
- Showing that as input modes are withheld during inference, performance metrics taper off gracefully;
- Evaluating our proposed approach for remote sensing change detection;
- Evaluating our proposed approach for global image-localization.

6.1 Related Work

Hallucination Approaches and Multi-modal Learning Imputing missing modalities or data is a topic that has been explored deeply. Recent works [25,26,40,57] have employed generative adversarial networks (GANs) and other adversarial methods to recover specific modes that are either expensive or typically unavailable. [57] relies on an additional network to generate "hints" about what might be present in a missing area, which is supplied to the discriminator network. [18,25,40,65] employs a network of CycleGANs [65] to impute missing modes from examples with multiple distinct data modalities that might be present. [48] also handles multi-modal examples, except with a stacked autoencoder featuring an additional mask indicating whether data is missing or present.

Other approaches directly model the relationships between groups of related data modalities and learn to translate between them. [10] learns a model which embeds examples from one mode into the same space as a pretrained embedding model for another mode. [47] similarly uses contrastive learning to extend multi-modal learning to an arbitrary number of modes. [61] simultaneously learns all pairs of transformations between a set of related modalities, while leveraging path consistency to improve overall model performance. [26] introduces additional generators and discriminators to adapt the existing GAN structure to allow for imputation of missing data.

Image Segmentation Transformers Finally, there are a number of recent approaches to applying Image Transformers (ViT) to segmentation and other dense prediction tasks. [38] follows an image pyramid approach where intermediate features from a stack of Transformer encoder layers are upsampled and stacked to create the semantic features passed to a final predictor head. [43] uses a Transformer encoder model to compute dense semantic features which are then compared with a learned bank of categorical features as an approach to segmentation. [31] introduces a hierarchical shifted window approach to segmentation which increases the efficiency of the Transformer's self-attention steps. [60] also employs a hierarchical method which relates individual pixels to larger image regions using a Transformer model. [56] employs a number of improvements to the Transformer architecture that allow it to discard the positional encoding and operate on overlapping patches. Our proposed architecture is similar to [2] which separates the multi-head attention step into individual axes to combat the memory required for video segmentation, however our approach differs in that is extended to handle arbitrary additional axes (for example, modality), and is trained for setting where not all modes or time steps can be expected.

6.2 MM-ViT: Multi-modal Image Transformers

The high level architecture of our proposed approach, MM-ViT, which is visualized in Fig. 6.1, is as follows. Given a collection of views from differing modalities, we extract fixed-sized tokens following a sliding window approach, then we append a positional embedding that encodes both where in the original image the token was located spatially or temporally, but also from which modality the token was extracted. Next, the tokens from all modes are combined into a single sequence and an encoder regresses an output feature for each. Those features are aggregated across modes, by computing the mean of the mode-specific tokens at every spatial location. The resulting token sequence is passed through a shallow MLP, and reshaped and upsampled to produce the final prediction. We provide additional details in the remainder of this section.

6.2.1 Mode-Specific Positional Encoding

One of the key components of the Vision Transformer approach is the positional encoding. There are a number of accepted methods for encoding the exact position (spatial or otherwise) of a token in a sequence. We extend the positional embedding approach of ViT [8] to also encode which modality the token is from in the same fashion as its spatio-temporal location. To do so, we learn a separate positional encoding layer for each

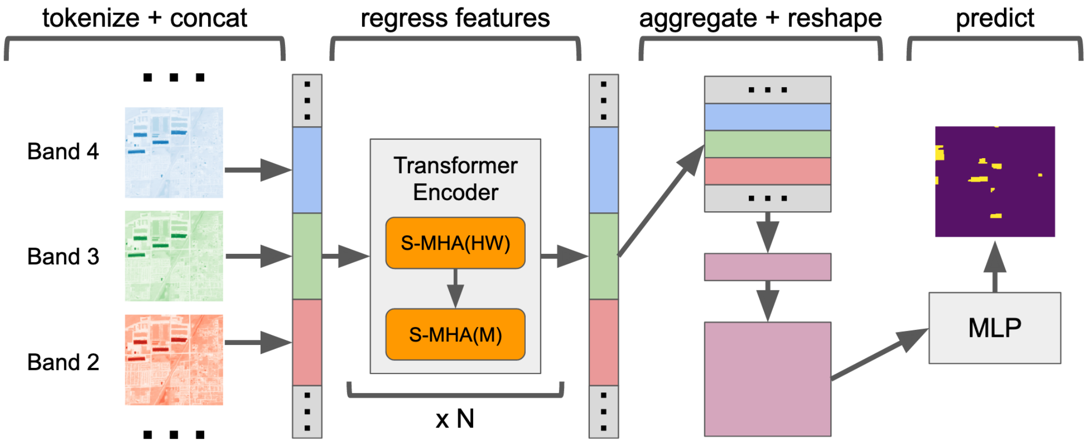


Figure 6.1: **Multi-modal transformer (MM-Vit) architecture overview.** The input modes are tokenized, have positional encodings added, and the concatenated into a single token stream. The positional encoding includes an extra feature encoding from which specific mode the token is sourced from. Next, a transformer encoder is used to predict output features for each input token. Then, the predicted tokens are aggregated across the modes to produce a single prediction for each spatial position. Finally, these are reshaped back into the input shape and a loss can be computed.

modality. The final embedding for each token is then concatenated with the flattened contents of the token and added to the token stream.

6.2.2 Encoder

Given a collection of images of from different modalities, $I_m \in \mathbb{R}^{H\times W}; \forall m \in M$, tokens are extracted in a sliding window approach, $t = [t_m^i; \forall m \in M]$. The shape of each token patch is $t_m^i \in \mathbb{R}^{P^2}$, where P is the window size for mode m. Then the sequence of tokens are each projected to a common size, $x_0 = [f_m(t_m^i); \forall m \in M]$, where $f_m(t) : \mathbb{R}^{P^2} \mapsto \mathbb{R}^F$ flattens the patch to a 1D vector, appends a positional encoding, and linearly projects each token to dimension F. Internally we represent a token sequence as a tensor, $x_0 \in \mathbb{R}^{M\times H\times W\times F}$.

The token sequence, x_0, is input to a Transformer encoder composed of L layers of multi-head attention layers followed by MLPs:

$$x_{i+1} = \mathrm{LN}(x_i) \tag{6.1}$$

$$x_{i+1} = x_i + \mathrm{MHA}(x_{i+1}, x_{i+1}, x_{i+1}) \tag{6.2}$$

$$x_{i+1} = x_{i+1} + \mathrm{MLP}(\mathrm{LN}(x_{i+1})). \tag{6.3}$$

6.2.3 Separable Multi-Head Attention

Our proposal increases the number of tokens in the sequence by the number of modes. Given that Multi-Head Attention (MHA) has $O(n^2)$ computational and space requirements, this is dramatically more expensive. To address this issue, we propose that during the attention step of each transformer encoder layer, we compute attention only on a subset of axes by reshaping the input tensor such that attention is only computed over a subset of dimensions. We call this practice *Separable Multi-Head Attention (S-MHA)*. For example, to compute attention over the $H \times W$ dimensions of x[1]:

$$x_i = \text{Reshape}(MHWF \rightarrow (M)(HW)F, x_i) \tag{6.4}$$

$$x_i = \text{MHA}(x_i, x_i, x_i) \tag{6.5}$$

$$x_i = \text{Reshape}((M)(HW)F \rightarrow MHWF, x_i). \tag{6.6}$$

We compose multiple calls over subsets of dimensions, such as over the spatial dimensions, then modality, e.g. :

$$x_i = x_i + \text{S-MHA}(HW, x_i, x_i) \tag{6.7}$$

$$x_i = x_i + \text{S-MHA}(M, x_i, x_i). \tag{6.8}$$

6.2.4 Prediction Head

The final step of our proposed approach is to aggregate features along all dimensions except height and width, then make a final prediction. We do so by a simple averaging step. Our prediction head is a shallow MLP with a single output logit which signifies the change / no-change decision. Finally, we use bilinear interpolation to upsample the low-resolution logit image to be the same size as the input image. During training, we minimize the weighted binary cross entropy loss between this resized logit map and the ground truth change map.

6.2.5 Mode Dropout

One of the stated purposes of our proposed approach is to be invariant to the availability of input modes during all stages in training and inference. To that end we propose that during training, randomly drop modes from each example independently with some probability p. We evaluate two different approaches to mode dropout: 1) zero-ing out selected modes, and 2) removing selected modes from input sequences, referred to as *zero* and *drop*, respectively.

[1] Our notation borrows heavily from the *einops* library: `https://einops.rocks/`

Table 6.1: **Performance comparison with other methods on OCSD dataset.** Both our proposed MM-ViT approach and the baseline ViT perform similarly to existing non-transformer approaches. All results are reported using all 13 bands of OSCD imagery.

Method	Prec.	Recall	Acc.	F1
Siam. [3]	24.16	85.63	85.37	37.69
EF [3]	28.35	84.69	88.15	42.48
FC-EF [3]	64.42	50.97	96.05	56.91
FC-Siam-conc [3]	42.39	65.15	93.68	51.36
FC-Siam-diff [3]	57.84	57.99	95.68	57.92
ViT	42.73	67.46	94.81	52.32
MM-ViT	46.31	61.79	95.36	52.94

6.3 Experiments

We evaluate our proposed approach for change detection in satellite imagery. Our change detection experiments focus on the Onera Sentinel-2 Change Detection Dataset (OSCD) [3], a dataset composed of pairs of Sentinel-2 images and pixel-wise change maps centered above 24 cities from around the world.

6.3.1 Implementation Details

Our change detection model for OSCD is an early fusion model. We treat each pair of input bands as a separate mode. First, we concatenate the two images together, then we compute token sequences following the method described in Sec. 6.2. The Transformer encoder used is implemented in PyTorch, has 6 layers, each are 384 neurons wide, with 6 heads, an internal width of 1024, dropout set to 0.1, and GeLU activation.

As a baseline we also compare against a standard ViT model using the same settings. The baseline differs from our proposed method in that during token creation, modes are treated like image channels, i.e. tokens are extracted only along the space and time dimensions.

Unless specified otherwise, all models are trained with zero-ing based input dropout with $p = 0.1$, are optimized with the AdamW optimizer and a learning rate of $1e-5$ following a cosine annealing learning rate schedule. We weight the change labels by 20, a value we arrived at empirically. Our models are trained on a pair of NVIDIA V100 GPUs, for 1000 epochs and an effective batch size of 64. Results are reported on model checkpoints with the best validation F1-score.

We evaluate performance using four different metrics: precision, recall, global accuracy, and F1. We find that our proposed approach and the ViT baseline both perform

Table 6.2: **Inference on OSCD subsets with our proposed model and ViT baseline.** Giga-multiply-addition operations (GMACs) are a measure of complexity during model inference.

Method	Ch.	Prec.	Recall	Acc.	F1	GMACs
ViT	13	42.73	67.46	94.81	52.32	13.41
ViT	10	41.28	68.83	94.55	51.61	13.41
ViT	4	38.42	56.99	94.32	45.90	13.41
ViT	3	29.04	48.99	92.79	36.46	13.41
MM-ViT	13	46.31	61.79	95.36	52.94	63.05
MM-ViT	10	43.52	64.51	94.96	51.98	48.50
MM-ViT	4	46.19	52.44	95.41	49.11	19.40
MM-ViT	3	45.75	40.05	95.46	42.71	14.55

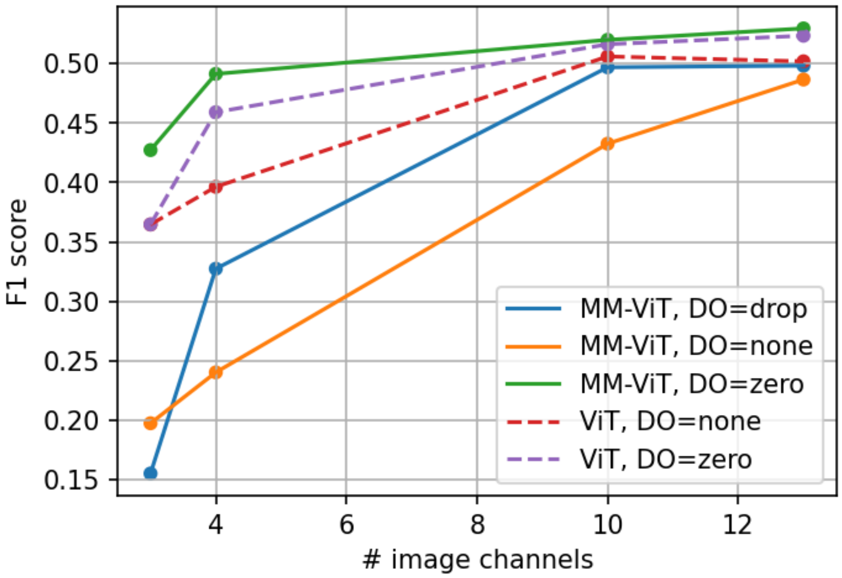


Figure 6.2: **Performance on OSCD [3] as the number of channels supplied to the model varies.** We compare our proposed model, MM-ViT, against a baseline ViT model. Both were trained under a number of mode dropout settings: *none* indicates no modal dropout, *zero* indicates dropped modes are set to be all zeros, and *drop* (MM-ViT only) dropped modes are removed from the input sequence entirely. We find that our proposed model outperforms the baseline regardless of the number of input modes when trained with *zero* dropout.

comparably with approaches that were designed specifically for the OSCD dataset (see Tab. 6.1).

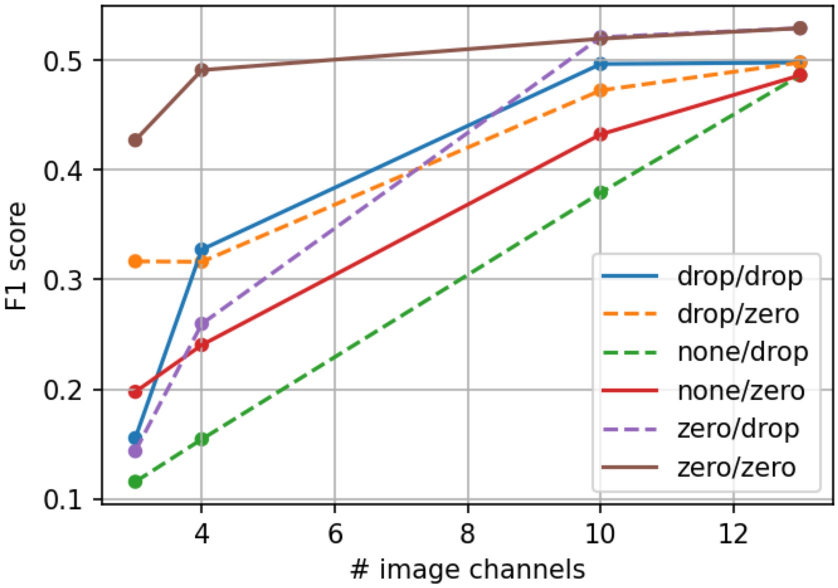


Figure 6.3: **Comparing MM-ViT performance when changing the mode dropout strategy during testing.** Lines are keyed as "train strategy / test strategy". Generally, changing the strategy from what the model was trained on negatively affects performance.

6.3.2 Inference on Subsets of Modes

We evaluate the performance of our method and the baseline when only a subset of channels is available at inference time. To do so we compare the final performance metrics between our proposed MM-ViT and baseline tested on four subsets of modes. For this experiment, we focus on the spectral band subsets from the original OSCD paper [3], specifically 13 is the full set, 10 excludes the bands with 60 meter resolution, 4 is the RGB bands plus the near infra-red band, and 3 is RGB only.

In Tab. 6.2, we show that while both models are capable of inference on subsets of the full set of OSCD channels our proposed MM-ViT approach retains more performance across subsets while actually becoming less expensive in terms of multiply-addition operations (MACs) as fewer modes are provided.

6.3.3 Comparing Modality Dropout Strategies

We evaluated three strategies for training our model: randomly zero'ing out missing modalities (*zero*), randomly removing missing modalities (*drop*), and always training with all modalities (*none*). The key difference between *zero* and *drop* is that when using *zero* we still include the tokens, they just have zero set for all features, except the positional encodings. We evaluated models trained with these strategies, across varying numbers

Table 6.3: **Ablation study on OSCD (13).**

S-MHA	Dropout	Prec.	Recall	Acc.	F1
—	—	39.05	70.16	94.11	50.17
—	zero	42.73	67.46	94.81	52.32
✓	—	40.21	61.54	94.51	48.64
✓	drop	39.35	67.83	94.22	49.81
✓	zero	46.31	61.79	95.36	52.94

of input channels, using either *zero* or *drop* at inference time. The results Fig. 6.3 show that (1) the *none* training strategy performs poorly and (2) using *zero* during training and inference performs best overall.

6.3.4 Ablation Study

We perform a simple ablation study of the extensions to the Transformer framework proposed in this paper. Starting from a basic ViT Transformer model, we sequentially add the two main proposals, Separable MHA and Mode Dropout. We investigate two different settings of Mode Dropout, zeroing-based, where selected modes are zeroed out during training, and dropping-based, where those modes are instead removed from the input sample entirely.

In Tab. 6.3, we can see that initially the baseline approach out-performs MM-ViT. The introduction of dropping-based dropout has a small effect on MM-ViT performance, and that zeroing-based dropout has a more significant effect on both ViT and MM-ViT. In particular, the addition of dropout pushes MM-ViT ahead of the baseline on most metrics.

6.3.5 Application to Image-Localization

Next, we apply the proposed MM-ViT model to image localization. Following the other experiments in this chapter, we treat each channel of the reference satellite imagery as a separate source of tokens. To do so, we train a separate resnet18 tokenization network for each. Similarly, the transformer decoder step of the MetaLoc approach is replaced with a MM-ViT model. This new model is trained following the same training procedure as used in the other localization experiments.

When evaluated on subsets of the channels available to reference image database, we observe that localization performance degrades gracefully as fewer and fewer channels are provided (Fig. 6.4). Interestingly, much of the performance is retained when only two

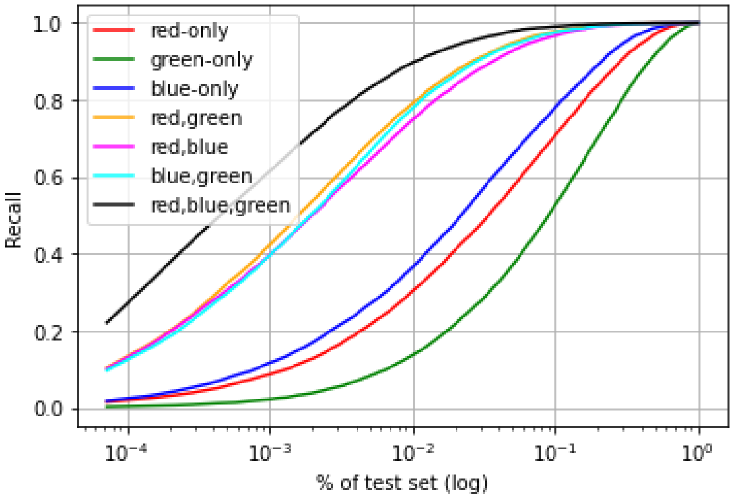


Figure 6.4: **Localization recall as reference channel-modalities are withheld.**

channels are provided, regardless of which two. A similar pattern is seen when the model is presented a single channel at test time.

6.4 Discussion

In this chapter we introduced a multi-modal vision transformer approach that is designed to be agnostic to the number of input modalities at both training and inference time. We extend existing approaches to positional encoding to account for the mode the token was drawn from, describe an approach to applying transformer encoders to token sequences with multiple dimensions to the problem of semantic segmentation, and evaluate the proposed approach on a multi-spectral satellite change detection dataset.

Limitations Our proposed MM-ViT model has a couple of limitations. The most obvious is that it is significantly more expensive than a comparable ViT model. In Tab. 6.2 we can see that in the 13 channel setting, a MM-ViT model requires ~4x GMACs during inference than an equivalent ViT. In the future, we plan to explore methods for reducing the memory overhead.

Chapter 7 Discussion

The aim of this dissertation is to unify under a single framework the various methods for global image-geolocalization. Overall, this dissertation proposes a new approach to image-localization under this framework and evaluates a number of such of methods for their ability to generalize to unseen areas and applicability under a variety of reference image settings.

In Chapter 3, we constructed a new framework within which existing image localization methods can be composed and compared. This framework supported our introduction of a new image-localization method which introduces a cross-attention mechanism to refine the representations of the reference image database by conditioning them on the features visible in the query image. We performed a detailed ablation study of the methods presented in this chapter, including varying the network responsible for computing token representations from each image, the hyperparameters of the introduced transformer decoder network, and more.

In Chapter 4, we evaluated the ability of several image-localization approaches to generalize to unseen areas. To support this we provided a more in-depth look at positional encodings and how geographic information can be included in them. We explored the impact of the choice in positional encoding. Next, we evaluated the generalization of these methods on both zero-data and low-data settings by introducing a simple strategy for transferring into the training set controlled amounts of data from held out regions.

In Chapter 5, we investigated the impact of reference imagery scale on localization accuracy. To do so we presented details on multiple resolutions of reference imagery and how they relate to the dataset used throughout this text.

Finally, inn Chapter 6, we presented a new approach to learning from multi-modal data with transformers. We applied this new model to localizing images against multi-modal reference databases; to do so we introduce a variant of our proposed hybrid approach which incorporates this multi-modal transformer variant. Next, we evaluate the proposed approach for global-image localization under a controlled setting intended to be a proxy for more realistic situations where multiple modalities of reference imagery are known to be available. Further, we also evaluated this approach on a multi-spectral semantic segmentation problem.

Bibliography

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 14

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 13, 45

[3] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau. Urban change detection for multispectral earth observation using convolutional neural networks. In *IGARSS*, 2018. 43, 48, 49, 50

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 17

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 16

[6] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 16

[7] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 2012. 1

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 13, 22, 32, 34, 45

[9] Connor Greenwell, Scott Spurlock, Richard Souvenir, and Nathan Jacobs. GeoFace-Explorer: Exploring the Geo-Dependence of Facial Attributes. In *ACM SIGSPATAL International Workshop on Crowdsourced and Volunteered Geographic Information (GEOCROWD)*, 2014. 1

[10] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 44

[11] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 1, 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 19, 25

[13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 16

[14] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2015. 16

[15] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 4

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 14

[17] Mohammad T Islam, Connor Greenwell, Richard Souvenir, and Nathan Jacobs. Large-Scale Geo-Facial Image Analysis. *EURASIP Journal on Image and Video Processing*, 2015. 1

[18] Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dm2c: Deep mixed-modal clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 44

[19] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE international conference on Robotics and Automation (ICRA)*, 2016. 1

[20] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[21] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

[22] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 17, 21

[23] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017. 16

[24] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015. 16

[25] Dongwook Lee, Junyoung Kim, Won-Jin Moon, and Jong Chul Ye. Collagan: Collaborative gan for missing image data imputation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 43, 44

[26] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 43, 44

[27] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4

[28] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 43

[29] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5, 11

[30] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 16

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 45

[32] Gengchen Mai, Yao Xuan, Wenyun Zuo, Yutong He, Stefano Ermon, Jiaming Song, Krzysztof Janowicz, and Ni Lao. Sphere2vec: Self-supervised location representation learning on spherical surfaces. 2021. 32

[33] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 16

[34] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 17

[35] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[36] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision (ECCV)*, 2020. 16

[37] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 16

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021. 45

[39] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 19

[40] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. Vigan: Missing view imputation with generative adversarial networks. In *IEEE International Conference on Big Data (Big Data)*, 2017. 43, 44

[41] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. *arXiv preprint arXiv:2005.03860*, 2020. 2, 4, 5

[42] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, 2019. 25

[43] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021. 13, 45

[44] Abby Stylianou, Joseph D O'Sullivan, Austin Abrams, and Robert Pless. Images don't forget: Online photogrammetry to find lost graves. In *IEEE Applied Imagery and Pattern Recognition (AIPR)*, 2014. 1

[45] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 16

[46] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016. 5

[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision (ECCV)*, 2020. 44

[48] Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. Missing modalities imputation via cascaded residual autoencoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 43, 44

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 14

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 13, 23, 32

[51] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008. 16

[52] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 4, 16, 20

[53] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3, 19, 27

[54] Scott Workman and Nathan Jacobs. On the location dependence of convolutional neural network features. In *IEEE/ISPRS Workshop: EARTHVISION: Looking From Above: When Earth Observation Meets Vision*, 2015. 5, 6

[55] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 4, 5, 6, 16, 40

[56] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 13, 45

[57] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning (ICML)*, 2018. 43, 44

[58] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 25

[59] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. Signal-to-noise ratio: A robust distance metric for deep metric learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 16

[60] Yuhui Yuan, Xiaokang Chen, Xilin Chen, and Jingdong Wang. Segmentation transformer: Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2021. 13, 45

[61] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 44

[62] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6

[63] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 13

[64] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 13

[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 44

[66] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5, 11, 30

Vita

Connor Greenwell

Gardenside, Lexington, Ky

Education

B.S., Computer Science & Mathematics 2011–2016
Dept. of Computer Science
University of Kentucky; Lexington, KY

Appointments

Graduate Research Assistant 2016–2022
Multimodal Vision Research Laboratory
Dept. of Computer Science
University of Kentucky; Lexington, KY

Research and Development Intern Summer 2021
Computer Vision Team
Kitware, Inc.; Clifton Park, NY

ASTRO Graduate Student Researcher at ORNL Summer 2019
National Security Emerging Technologies Division
Oak Ridge Institute for Science and Education (ORISE); Oak Ridge, TN
Advanced Short-Term Research Opportunity (ASTRO) Program

Undergraduate Research Assistant 2014–2016
Dept. of Computer Science
University of Kentucky; Lexington, KY

Visiting Undergraduate Research Assistant Summer 2014
Dept. of Computer Science
University of North Carolina at Charlotte; Charlotte, NC
NSF Research Experience for Undergraduates Program

Publications

[1] Gongbo Liang, Connor Greenwell, Yu Zhang, Xin Xing, Xiaoqin Wang, Ramakanth Kavuluru, and Nathan Jacobs. Contrastive cross-modal pre-training: A general

strategy for small sample medical imaging. *IEEE Journal of Biomedical and Health Informatics*, 2021.

[2] Benjamin Brodie, Subash Khanal, Muhammad Usman Rafique, Connor Greenwell, and Nathan Jacobs. Hierarchical probabilistic embeddings for multi-view image classification. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021.

[3] Scott Workman, M Usman Rafique, Hunter Blanton, Connor Greenwell, and Nathan Jacobs. Single image cloud detection via multi-image fusion. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2020.

[4] Gongbo Liang, Yu Zhang Connor Greenwell, Xiaoqin Wang, Ramakanth Kavuluru, and Nathan Jacobs. Weakly-supervised feature learning via text and image matching. *arXiv preprint arXiv:2010.03060*, 2020.

[5] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[6] Tawfiq Salem, Connor Greenwell, Hunter Blanton, and Nathan Jacobs. Learning to map nearly anything. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019.

[7] Connor Greenwell, Scott Workman, and Nathan Jacobs. Implicit land use mapping using social media imagery. In *2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2019.

[8] Menghua Zhai, Tawfiq Salem, Connor Greenwell, Scott Workman, Robert Pless, and Nathan Jacobs. Learning geo-temporal image features. 2018.

[9] Connor Greenwell, Scott Workman, and Nathan Jacobs. What goes where: Predicting object distributions from above. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

[10] Ryan Baltenberger, Menghua Zhai, Connor Greenwell, Scott Workman, and Nathan Jacobs. A fast method for estimating transient scene attributes. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[11] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. Deepfocal: A method for direct focal length estimation. In *IEEE International Conference on Image Processing (ICIP)*, 2015.

[12] Mohammad T Islam, Connor Greenwell, Richard Souvenir, and Nathan Jacobs. Large-scale geo-facial image analysis. *EURASIP Journal on Image and Video Processing*, 2015.

[13] Connor Greenwell, Scott Spurlock, Richard Souvenir, and Nathan Jacobs. Geoface-explorer: Exploring the geo-dependence of facial attributes. In *ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, 2014.