Augmenting geospatial search with micro-terrain detail

Kent O'Sullivan
osullik@umd.edu
University of Maryland
USA

Nicole Schneider
nsch@umd.edu
University of Maryland
USA

Hanan Samet
hjs@cs.umd.edu
University of Maryland
USA

ABSTRACT

Geographic information systems (GIS) provide users with a means to efficiently search over spatial data given certain key pieces of information, like the coordinates or exact name of a location of interest. However, current GIS capabilities do not enable users to easily search for locations about which they have imperfect or incomplete information. In these cases, GIS tools may help with narrowing down to the general region of interest, but a manual last-mile search must then be performed by the user to find the exact location of interest within that region, which typically involves the visual inspection of remote sensing imagery data or street-view images to identify distinct landmarks or terrain features that match the partial information known about the location. This step of the search process is a bottleneck, as it encumbers the user with the burden of sifting through many possible candidate locations until the correct one is visually identified. Taking inspiration from the way humans recall and search for information, we present *the **G**eospatially **E**nhanced **S**earch with **T**errain **A**ugmented **L**ocation **T**argeting (**GESTALT**)*, an end-to-end pipeline for extracting geospatial data, transforming it into coherent object-location relations, storing those relations, and searching over them. We address the geospatial object ownership assignment inference task under uncertainty constraints and contribute a new gold standard Swan Valley Wineries dataset and a proof of concept implementation of the proposed architecture.

ACM Reference Format:
Kent O'Sullivan, Nicole Schneider, and Hanan Samet. 2023. Augmenting geospatial search with micro-terrain detail . In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Geographic information systems (GIS) provide users with a means to efficiently search over spatial data given certain key pieces of information, like the coordinates or exact name of a location of interest. However, current GIS capabilities do not enable users to easily search for locations about which they have imperfect or incomplete information. From psychology and neuroscience research about how humans develop cognitive maps of terrain for navigation and route planning [7, 12, 27**?**], we know that people tend to anchor memories of a location around its visible objects and landmarks, likely doing so hierarchically, or separately relating global and local features [27]. For example, a user may remember a location by a series of visual features encountered near it, like a large building, a bus stop, and a brightly colored sign, but fail to recall its exact address or physical coordinates. In these cases, GIS tools may help with narrowing down to the general region of interest, but a manual *last-mile* search must then be performed by the user to find the exact location of interest within that region. This last-mile search typically involves the visual inspection of remote sensing imagery data or street-view images to identify distinct landmarks or terrain features that match the partial information known about the location. This step of the search process is a bottleneck, as it encumbers the user with the burden of sifting through many possible candidate locations until the correct one is visually identified.

Taking inspiration from the way humans recall and search for information [6, 17, 27**?**], we present *GESTALT*, an end-to-end pipeline for extracting geospatial data, transforming it into coherent object-location relations, storing those relations, and searching over them. We address the geospatial object ownership assignment inference task using through probabilistic assignment of objects to locations and contribute a new gold standard *Swan Valley Wineries* dataset and a proof of concept implementation of the proposed architecture.

The rest of this paper is organized as follows. In section **??** we describe the process used to generate the gold standard wineries dataset. Next, in section 3, we define the *GESTALT* architecture and discuss how each subsystem contributes to our human-centric approach to automating the last-mile search. Then, we present implementation details in section **??** and a detailed comparison of object ownership assignment methods in section **??**. Finally, we summarize related work in section 8 and conclude by identifying future research directions in section 9.

2 PROBLEM DEFINITION

Describe the problem here......

2.1 Preliminaries

Define region, location, object..... *Objects* are any physical entity. For example, a *tree, building, lake, bridge, gate* or *sign* could be objects. *Objects* can also have attributes that provide amplifying information about the object, including things like *color, material, size, species etc.*.

Locations are physical entities that *do* something. They are the meaningful grouping of objects determined by ownership, proximity or utility. They have some purpose other than being an object. They could be a *business, attraction, property etc.*.

Define last-mile search, object ownership task, concept mapping, progressive search.....

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2.2 Datasets

GESTALT requires a list of all *objects* with their coordinates and a list of all *locations* with their coordinates. These objects and locations can be obtained in a variety of ways, which we briefly outline below.

2.2.1 Objects.

Open Street Maps (OSM). *Open Street Maps (OSM)* is a knowledge collective that contains open-source Geodata [4], which can be easily extended via their online editing interface. The primary concern with OSM is the accuracy and validity of crowd-sourced data [26]. However, a combination of human review and an application of machine learning for detecting anomalous behavior in OSM edits [15] goes a long way toward addressing these concerns.

OSM records objects' coordinates as a mixture of point coordinates and bounding polygons. OSM has a defined and curated ontology that defines the labeling scheme, maximizing interoperability. The OSM objects are crowd-sourced and of varying granularity and completeness. Incompleteness is part of the initial scope of OSM, with the founder noting that it's typically only what people want to add that gets added [4]. In general, the completeness of OSM is unassured, so scaling *GESTALT* beyond the trivial requires an automated method for object detection and resolution. Because of the lack of completeness and to enable evaluation with ground-truth labels, the Author used Google Earth to annotate objects present at six wineries within the Swan Valley Region of Western Australia. The wineries are separated in space and a semi-rural environment. The tags consist of an object name, its latitude & longitude, and any descriptive markings written as key:value pairs. The objects are stored by their ground truth parent location in a KML file. Additional dataset creation for the small New South Wales town of Buladelah aims to fill a 'suburban' setting and of the Darby St Restaurant Strip in Newcastle, New South Wales, for an 'urban' environment.

Object Detection. Explain here......

2.2.2 Locations. Google Maps[1] maintains *locations* as coordinate points with associated metadata. The locations are generally current and complete. Google Maps does not support bounding polygons at the location level; it appears to extend to as granular as ZIP Codes or suburb boundaries and no further. OSM Supports locations, but is less complete than Google Maps (at least for Australian Wine Regions.) In general, for *GESTALT* to function optimally, the input locations should be the union of Google Maps and OSM. However, given the limitations of Google Maps API usage, a dataset was manually curated in OSM using publicly available information and the Author's world knowledge. Creating the Swan Valley Winery dataset for this project has the benefit of yielding 31 additional nodes and associated metadata for the OSM project.

3 ARCHITECTURE

The architecture of *GESTALT* is in Figure **??**. It decomposes into four essential functions: data acquisition, ownership assignment, concept mapping and search.

[1]Google Maps

3.1 Data Acquisition

The data acquisition component of *GESTALT* begins with a visual encoding of the world. The visual encoding is primarily remote sensing imagery providing a top-down view of the earth's surface, but it also includes street-view imagery and other photographs. The system collects two types of information from this imagery, *locations* and *objects*. These two information types are discussed in section **??**. Briefly recapping, objects are any physical thing in the world, and locations are the specific uses of a place that usually contains collections of objects.

The vision for a mature data collection system enables the autonomous collection of location and object data. The collection system will source location data from open-access systems like OSM and relies on crowd-sourced information. Businesses, attractions and other higher-level 'locations' are likely to be annotated by the open-source community or business owners themselves. Objects, the core of *GESTALT*, are much less likely to be annotated. Few people have the patience to manually tag the geolocation of apparently inconsequential things like trees, statues, fountains and telegraph poles. An automated solution aims to leverage publicly available remote sensing imagery data (Bing Maps Satellite data, for example) and public streetview and photo contributions to automatically identify objects, geo-locate them and add those tags to a database.

The design for this subsystem breaks maps into small geographically-bounded chunks (approximately the size of a 'location'). It will use remote-sensing imagery to create a grid of objects / not-object. It will retrieve ground-level imagery within and adjacent to that box. The first challenge is classifying an image as 'indoors' (where no objects will be visible from RSI and the closest building will 'own' it) or 'outdoors', where objects can map to the RSI. Numerous approaches exist to the indoor/outdoor scene classification [25]. Each object in an outdoor photo's distance from the camera geolocation will be estimated using a myriad of depth estimation techniques [10, 13]. Where multiple images cover the same area from different perspectives, the composite of these images will be used to estimate the positions of objects, as has been shown in prior work like IM2GPS from Carnegie Mellon University [5] and numerous other efforts over internet-available images [23]. As discussed in the following sub-section, some errors are permissible here and being 'close enough' is good enough as a starting point for the following systems.

The data extraction, cleaning and loading are implemented in Python in two parts, the *KML parser* for object extraction and the *Open Street Maps* query interface for location retrieval. The KML parser leverages the *fastKML*[2] and ingests a KML file divided by region (where each region is a bounding box covering an arbitrary number of locations). Within each region (for this test dataset), each location is separated, with its objects stored as its children. Attributes of the objects (e.g. colour, size, material) are recorded in the comments field as key:value pairs. The KML Parser extracts the objects into dictionaries organised by location before exporting the files as JSON for future analysis.

[2]Fast KML PyPI Repo

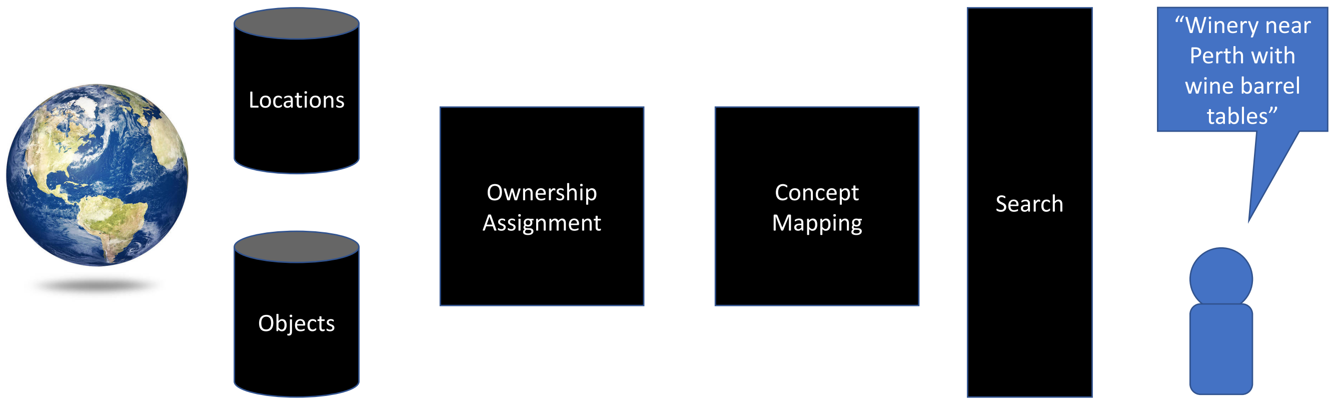

Figure 1: The architecture of *GESTALT* consists of the data collection subsystem, the ownership assignment process, the concept mapping process and the search subsystem.

The OSM query interface leverages the *OSMPythonTools*[3]. It passes a bounding box to the OSM Overpass-Turbo API[4] and requests the relevant location nodes in the area. The results are arranged into a dictionary and exported as JSON for further analysis.

3.2 Ownership Assignment

Ownership assignment is the unsupervised process through which objects are associated with locations. Objects need to be associated with locations in *GESTALT* because for the *concept mapping* process and *search* subsystem to work, they need to know which objects belong to each location. For example, assuming two adjacent wineries, a fountain between them would be west of one but east of the other. The mapping will be incorrect unless it is clear which winery it belongs to. Similarly, for search, the underlying idea for *GESTALT* is that people will remember particular objects at locations and use them as clues to find them again. Without an accurate object-to-location assignment, the search functionality will not work. The ownership assignment process needs to be unsupervised to enable scaling. Aspirationally, *GESTALT* will index the world's objects to allow searchers to find any location with *GESTALT*. Processing a world's worth of data necessitates an unsupervised approach.

The Ownership Assignment process accepts two inputs, a collection of locations with their coordinates and a collection of objects with their coordinates. The process works to assign each object to its parent location. The process ends when objects are mapped to their parent locations. Of note, because the human eye can see over property boundaries and other invisible lines on maps, we can accept a small margin of error where objects from neighboring locations might be mislabeled. For example, perhaps there is a large red shed at the back of a location's property that is not visible to the main part of the parent location but is clearly visible to the neighboring location. There will also exist some objects which plausibly could be seen and remembered by patrons of several locations, for example, a lake or a giant statue. This multiple-ownership situation is one of the driving requirements for implementing concept mapping to extract additional discriminatory information between locations based on the geospatial layout of objects.

3.3 Concept Mapping

Concept mapping is the process of determining the geospatial relations between objects. Much information is implicit in the relative positioning of objects within a location. For *GESTALT*, there are three types of relations. The first is *Static Cardinal Relations* which encodes whether an object is North, South, East or West of a location. Static Cardinal Relations support simple queries where the user knows that a location has a lake on its western side. The second is *Dynamic Cardinal Relations* which determines whether an object is North, South, East or West of another object within a location. These queries support cases where a searcher might remember standing at a lake northeast of a location and that there was a

[3]OSMPythonTools PyPI Repo
[4]Overpass-Turbo API

swingset to the immediate west of them but still in the northeast of the location overall. The third are *Positional Relations* which are applied to the two other types to enable reasoning about objects that are left, right, up, down, beside, behind etc., other objects. Positional relation use will be extensive because few humans think in cardinal directions, and most spatial reasoning is conducted from the person's perspective. Positional Relations enable users to query for locations where there is a letterbox on the left of the driveway as you enter the driveway while the house is in front of you.

Concept mapping needs to be unsupervised and support aggregation. Tracking every object's relative location to every other object quickly becomes intractable, so mechanisms to aggregate depending on the level of granularity need to be applied. Accordingly, the underlying data structure must support aggregation and relative position querying.

3.4 Search

The core function of *GESTALT* is searching. The search function assumes that the searcher only has partial information about a location. There are two elements of partial information. First is a general idea of the region in which the location occurs. Here region means the area surrounding a location. For example, in searching for a winery, it is assumed that the searcher knows that they are in the swan valley region of Western Australia. A region could be an administrative boundary like a city, suburb, or general geographic area. Either way, we assume that the searcher can prune their search space to the commencement of the *last-mile* search before using *GESTALT*. The second assumes that the searcher knows a subset of the objects associated with a location. They may or may not know any of the attributes of those objects (for example, material, color etc).

The search problem can then be framed in several ways. A *set membership problem* is the most straightforward and most efficient. Given a set of locations, each of which has a set of objects it 'owns' and a set of objects in the search term, which locations have complete coverage of the search set. Bloom Filters are the obvious choice of data structure to support this search method. A limitation of using bloom filters is that if the user has little information to discriminate locations, almost any location will satisfy a query. For example, searching "tree" would return every winery in the Swan Valley region. A second limitation is the lack of support for aggregation. Searching for 'tree' might yield nothing, but searching for '30 trees' would considerably prune the result.

The second approach to search incorporates the concept of mapping and becomes a spatial search with a specific method depending on the objects' underlying data structure. The general case is as follows: Given a set of locations with geospatial mappings of their child objects and a subset of those geospatial mappings of child objects which location does that subset match? If using a graph structure, each location's objects become a graph where the objects are nodes and the geospatial relations are edges encoding the spatial relations (e.g. west of, north of). It is a subgraph matching problem, which is, of course, NP-Complete. Alternately, representing each location's objects as a KD-Tree rooted on the centroid of the object cluster would allow for dynamic searching. For example, assuming an initial split on the longitude of objects, we could immediately tell that all objects in the left subtree would be west of that root. For either of these geospatial approaches, a translation layer from the positional relational to cardinal relational will need to occur.

Regardless of the formulation of the search problem, there is a clear requirement for semantic search across objects. For simple spelling variations (e.g. 'colour' in the King's Australian English versus 'color' in American English), a string distance metric like *Levenshtein* distance would suffice. But a richer semantic search is required for more pronounced linguistic variations like 'water fountain' versus 'drinking fountain' versus 'bubbler'. The first option to reduce the likelihood of inconsistently named objects is to enforce compliance with the Open Street Maps ontology, which is an extensive definition of locations, objects and their descriptions. While adherence to the ontology enforces internal consistency, it does not overcome the issue of a user searching with unknown terms. A straightforward option could be to use the vector embedding of a word as a starting point and use the k-nearest words in vector space as alternate search terms. It is unlikely that this will significantly impact the false positive rate once an appropriate similarity threshold is set but may increase the system's overall recall. A more complicated approach could leverage an external semantic data source like DBPedia or WikiData, or even WordNet to search for semantically similar terms to substitute in the search.

The search problem must balance precision and recall while not being computationally intractable. An effective search process will use bloom filters to prune the search space for the more complicated geospatial search. Semantic enrichment should be applied independently at each stage of the search in an attempt to improve the recall of *GESTALT*.

3.5 Sumamry of Architecture

The Architecture of *GESTALT* is designed to be lightweight and modular. The core requirement is to improve the ability to find locations of interest based on partial information. The search subsystem needs to balance precision in reducing the number of candidate locations with maximizing the recall of possible candidate locations. The recall is prioritized. The search space should be pruned with set membership checks based on the intuition that there is no point in running an expensive geospatial query over a location that doesn't contain the objects in question. Bloom filter checks are cheap; the human eye doesn't see invisible lines on a map. Accordingly, the ownership assignment subsystem can be inexact, and objects should be 'shared' between locations where appropriate. That location sharing in membership assignments improves the recall of the system. As a corollary, the subsequent spatial search process maintains the system's precision using the concept mappings of objects to extract implicit information about the location. Underlying the search problem requires collecting and processing objects and locations autonomously, at scale. Collecting and processing objects and locations is the first stage explored in section **??**, implementation.

4 OBJECT DETECTION

Objects can be fed into *GESTALT* in three ways, which we describe here.

4.1 Ground Truth Hand Labeled Tags

The first method we use to obtain object tags is by hand labeling them. For benchmarking purposes we curated the Swan Valley Wineries dataset containing ground truth object tags and their associated locations stored in KML....... (define KML?)........ Attributes of the objects (e.g. color, size, material) are recorded in the comments field as key:value pairs. Each object from the hand-labeled dataset is assigned a confidence score of 1.0, since it is manually identified and tagged.

4.2 OSM Object Tags

The second method involves leveraging existing OSM object tags. While businesses, attractions and other 'locations' are commonly annotated by the OSM community, objects are more sparsely tagged, since few people have the patience to manually label apparently inconsequential things like trees, statues, fountains and telephone poles. *GESTALT* ingests what object tags do exist in OSM through ....describe the process here (API?)................ Each object ingested from OSM is assigned a confidence score of 1.0, since these tags are maintained by the open-source community and are subject to some degree of review and ....explain why we have some confidence in them.............

4.3 Noisy Image-based Tags

The third, and most important method by which *GESTALT* ingests objects, is through automatic object detection. Given a set of images and their associated geo data, ...what is it actually called?........... the Object Detection module uses YOLO [5] to identify objects in each image. Those objects are then tagged with the geolocation of the image, and stored...........how?........... For our experiments, we pull ....how many?............ images over ....timeframe.......... and use pre-trained YOLO v.8 to detect objects from 80 classes (based on the COCO dataset.........footnote........). Each object identified is assigned a probability score ....how does YOLO get that????..........

5 OWNERSHIP ASSIGNMENT

The results section discusses the empirical analysis of the clustering methods tested during the implementation of the Ownership Assignment process.

5.1 Dataset

The experiment design is simple and designed to provide an upper bound for performance on an optimal dataset with no noise (the Swan Valley wineries dataset.) The Swan Valley Wineries dataset consists of 31 wineries retrieved from OSM and $\tilde{1}50$ objects hand-labeled across 6 of those 31 locations with ground-truth location labels.

5.2 Method

Using 1/distance to nearest location as probability of assigning an object to that location. Achieves a fuzzy clustering of objects to locations, since objects can be assigned to multiple locations with some probability p.

[5]YOLO

5.3 Results

Examining the errors reveals the following insights about each clustering technique.

When K-Means clusters incorrectly, labels are still correct. When K exceeds the number of clusters, it fragments the actual clusters. However, in ideal conditions like the Wineries Dataset, where locations are separated, the centroids of these cluster fragments are still closest to the correct location. As a consequence, they are correctly labeled despite being incorrectly clustered. We expect the accuracy will drop when locations are more densely packed. However, when we aim to process all objects and locations in a region concurrently, the number of clusters will likely approach the number of locations, and the issue will be less pronounced.

RESULTS TABLE HERE.........

Location	Precision	Recall
K-Means	$K = 1$	1
	$K = 6$	6
	$K = 31$	31
	$K = 1$	1
	$K = 6$	6
	$K = 31$	31
	$K = 31$	31

Table 1: ....

5.4 OLD, INTEGRATE OR DELETE........

Ownership assignment is implemented in Python in two ways. The first is the trivial implementation, where the location returned from the OSM query is a bounding polygon. In this case, if the point lies within the minimal enclosing rectangle of the polygon, it is added as a 'member' of that location. The second, more common (and more challenging approach) formulates an unsupervised learning problem using clustering libraries from *scikit-learn*[6]. Given a collection of objects and a collection of locations within a bounding box region, clustering assigns each object to its 'parent' location. Under the assumption that the number of locations equals the number of clusters, K-Means clustering proved to be the most effective approach. After clustering the objects, we determine the centroid of the object cluster. Given a KD tree constructed from location point coordinates, a nearest neighbor search on the KD tree with a query parameter of the object centroid yields the nearest location and is assigned ownership of that cluster A detailed performance comparison is in section **??**.

Overall, initial proof-of-concept clustering uses K-Means and DBSCAN clustering. An optimal solution to the ownership assignment problem is an exciting and unusual clustering problem. Assuming that the collection of locations is complete and that the point coordinate of the location is central to the collection of objects that belong to that location, the clustering problem is the assignment of an arbitrary number of objects to any of a set of possible centroids. Not every centroid will have objects, and objects are not uniformly distributed. Initial investigation into the DVBSCAN algorithm Ram

[6]Scikit-Learn PyPI Repo

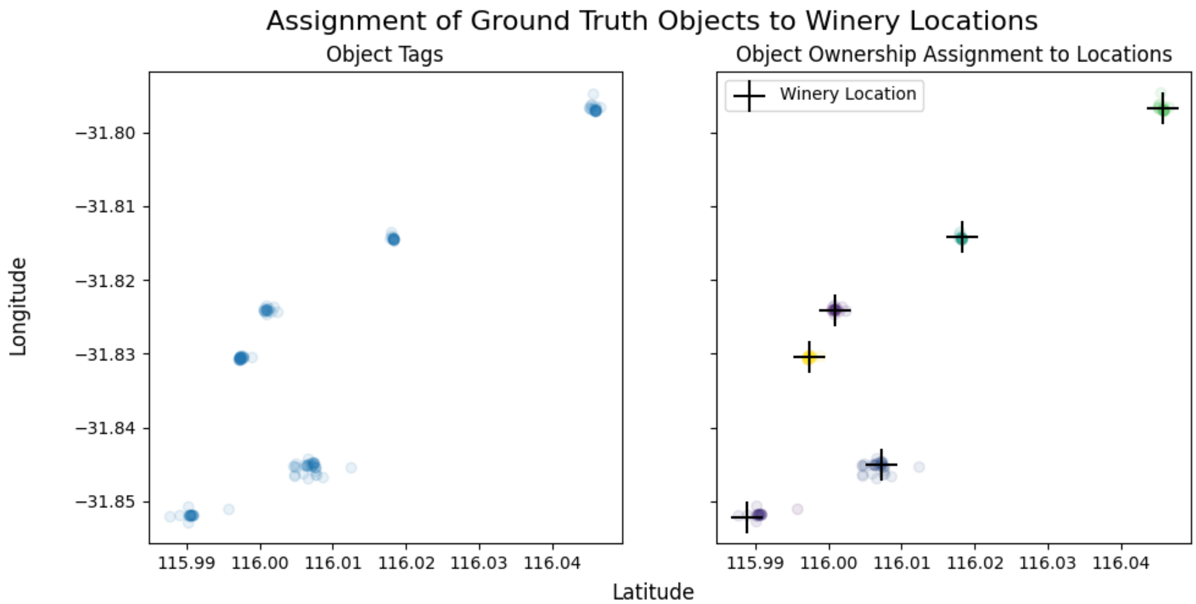

Figure 2: ....

et al. proposed in 2010 presents a promising direction to resolve this problem.[19]

Overall, the Ownership Assignment process needs to produce a data structure that will permit set membership checking for search and set the conditions for the concept mapping process. It is the most complete component of *GESTALT*. The choice of clustering algorithm needs refinement, and location-based bloom filters need to be implemented to support efficient search, but it is otherwise functional.

6 CONCEPT MAPPING

Concept mapping has been partially implemented in Python leveraging the *Scipy* library[7]. Two different approaches have been trialled. The first is simple dynamic arithmetic on the coordinates stored in a Pandas data frame. If one set of coordinates is above, below, left or right of another, it is north, south, east or west, respectively. While these calculations are in constant time for straightforward comparisons of known objects (e.g. "is the pond west of the bridge"), the time complexity rapidly increases as soon as aggregations are employed. Queries of "Give me everything west of the duck pond" would execute in $O(N)$ time as each element has to be examined. Worst-case queries would run in $O(N \text{ sup } 2)$ time, where every object is checked for its position relative to every other object.

The second (better) approach (only partially implemented) instantiates the objects within a location into a KD-Tree. Assuming that the object centroid is the root, we can quickly complete queries like "Give me everything west of the duck pond" by leveraging the structure of the subtrees to return the requested set. Similarly, getting the relative positions of two objects searches for a common ancestor. It uses the path between the children and their ancestor node to infer their spatial relation to each other.

A third approach, designed to leverage the *Neo4J Python Library*[8] to connect to a *Neo4J Graph Database*[9] but not implemented frames concept mapping as a graph traversal problem. In this formulation, each object is a node on a graph. Weighted, labelled edges exist between each node within a given proximity threshold to the node. The edge labels describe the neighbouring node's cardinal direction and the distance weights. After constructing the object graph, queries for 'give me everything west of the duck pond' would freely explore nodes connected by west, north and south edges. It can only traverse along an east edge so long as the total cost of travelling east would be less than the cumulative value of the 'west' travel up to this point.

Overall, concept mapping aims to enable geographic search over objects by explicitly representing their geospatial relationships to each other. The author implemented a very basic approach using coordinate arithmetic was quickly determined to be infeasible for the extensive data sets that *GESTALT* anticipates processing. KD-Trees for the objects in each location have been implemented, as have the KD-Trees for the locations themselves. This conceptual KD-Tree of KD-Trees approach performs a natural aggregation function which, provided that regions are created consistently, will allow for relative spatial queries at different levels of granularity. Empirical evaluation of the performance of the arithmetic, KD-Tree and Graph-based approaches is yet to be completed.

[7]SciPy PyPI Repo

[8]Neo4J PyPI Repo

[9]Neo4J Website

7 SEARCH

The search function has been implemented using the Python *Pandas*[10]Pandas PyPI Repo library. This approach assumes a single data frame of objects and their determined locations because of the number of possible attributes an object can have and the relatively few that they possess, this is a sparse data structure. The sparseness does indicate the discriminatory power of remembering attributes. For example, a 'door' is not informative, but a 'blue door' on your favorite seaside restaurant is more likely to prune the search space. Because *GESTALT* is designed only for the last-mile search and assumes a small starting region, it may remain feasible to use a simple data structure like a Pandas data frame containing all the objects for all the locations for the query region. More work with the aggregation functions is required to determine if it can support all the necessary aggregation queries comparing object collections.

Semantic search has not been implemented. However, the Levenshtein string distance metric (with $threshold = 0.8$) checks for small spelling discrepancies in input words. The priority weights towards retrieving all possible objects, so we accept the increased risk of mistakenly including an object to move the recall closer to 100%. The next component to be implemented is a nearest-neighbor retrieval mechanism using word embeddings. Prior work indicates that developing databases of embeddings is trivial[16], but using existing datasets tools like word2vec, GloVe and fasttext can generate embeddings over large, publicly available corpora that can be recreated.

As discussed in the subsection on Ownership Assignment implementation, bloom filters are a much more efficient operator for set membership testing. The KD-Tree is more suited for geospatial queries, so the Pandas Dataframe currently supports the gaps between the two in supporting aggregation queries. More work is required to integrate these data structures into a coherent search pipeline that maximizes recall while actively pruning the search space at every step so that the searcher can find their locations of interest. Natural language querying is an active area of research yet to present a solution capable of effectively translating natural language queries and their SQL solutions. Given the relatively constrained domain of this problem set, it is a good candidate for implementation as a low priority for improvement.

7.1 Scalability

The worst case complexity is for a search of N object terms where every object category is mapped to every location L. In that case we have to do N lookups to pull N sets of size L and intersect them. Realistically this would not happen and some object classes would be very discriminative and we could intersect those first to be more efficient.

8 RELATED WORK

signpost here

[10]https://pypi.org/project/pandas/

8.1 OSM Search Tools

On 08 May 2023 the Open Source Intelligence platform *Bellingcat*[11] released *Bellingcat OSM Search*[12][28], a tool also aimed at pruning search space using tagged objects. Using drop-downs and sliding bars, a user can specify a query for OSM tagged objects that appear within an adjustable distance from each other inside a region of interest. Although aiemd at addressing the same *last-mile* search problem as *GESTALT*, there are several? .... key limitations to the Bellingcat OSM Search Tool that *GESTALT* overcomes.

(1) They can do progressive search by executing multiple queries sequentially – ok
(2) They can't do concept mapping type queries – depends if we implement it or not
(3) They return generic points meeting criteria rather than tagged locations – ok
(4) They don't use objects not tagged in OSM, which are global objects people are certain enough about to tag (optimizing for precision and majorly missing recall) – this is key
(5) They don't rank results – tied to the point above, also key – Instead we pull many many tags via object detection on images geo-ed for locations, and associate them with the location(s) near to the geotag, which is noisy but captures much more of the possible things people might remember (optimizing for recall). Note we would not want to add these noisy tags to OSM (which would allow Belingcat to search on them), since they are inherently noisy. We return ranked results to handle the uncertainty w.r.t assigning objects to multiple locations and just having rogue object tags that aren't really there – the best locations matching the query will cover most of the search terms. This allows flexibility beyond what OSM object tags can cover (anything a detection model or human annotator wants to put as the tag).
(6) They return 0 results if a search term fails to hit in the window of interest. – we should mention this – People can search anything they remember, and even if some don't hit (weren't tagged in OSM or by the object detection step) the results won't be 0 like Belingcat does, instead will be based on the other search terms. This is much more natural to how humans do search...
(7) By relying on OSM API queries, their tool can do any region as long as its sufficiently small, at the expense of being slow (API calls are slow). – can we compare the same queries on swan valley in our tool and theirs? – We need to address our scalability issues

8.2 Psychology of Geospatial Reasoning

Several ideas from criminology, psychology and neuroscience drive the underlying notion of *GESTALT* that people remember objects and anchor on those 'things' that they see while experiencing a location to find it again.

GeoGuessr. A popular online game called *Geoguessr*[13] demonstrates that for many people, figuring out where they are in the world can be a source of much joy. A 2023 journal article analyzing

[11]Bellingcat Website
[12]Bellingcat OSM Search Tool
[13]Geoguessr Website

the strategies employed by a top player identifies several successful strategies that are used by a leading player [**?**]. The Geoguessr problem is a distinct and, in many ways, reciprocal problem to the *last-mile* search problem *GESTALT* seeks to solve. While the last mile search assumes a general region is known, the geoguessr frequently has no idea where in the world they are and needs to use clues from the interface (powered by google street view) to determine the country, state, county etc., that they are in. While not directly relevant, Geoguessr demonstrates the importance of searching for landmarks, and often even benign objects like bus stops, in locating where in the world an image is.

Winthropping. The Winthrop Method is a geospatial search method developed by Captain Winthrop of the Royal Engineers for use in Northern Ireland in the 1970s [7] to detect clandestine weapons caches and concealed improvised explosive devices. The underlying logic is that to find something, a human has to have some method of navigating to it and that the objects in our environment help to form mental models of the terrain, we can use to navigate by. A popular example is the closing scene of the film *The Shawshank Redemption* where the protagonist Andy Dufrense provides instructions to his fellow prisoner Ellis Redding on how to find a gift left for him *"It's got a long rock wall with a big oak tree at the north end... find that spot. At the base of that wall, you'll find a ...piece of black, volcanic glass. There's some thing buried under it I want you to have."* The two reasons that Winthropping works are *Affordance* and *Satisficing*. Affordance refers to the interaction of an agent with its environment and, in simple terms, means that particular objects will have a more significant impact on us and remain in our memory than others. When combined with the idea of satisficing, in which an agent makes a satisfactory or sufficient but possibly sub-optimal decision, it illustrates the value of object-based search. We cannot remember every detail of a location, so our brains will record only a few key objects or experiences for us to leverage. A 2013 Neuroscience study shows that when we revisit those locations, the objects we remember serve as keys to other memories of that location [12], *GESTALT* aims to exploit these geospatial aspects of memory for helping searchers to find the locations they are looking for.

8.3 Remote Sensing Imagery

Relevant to the data collection subsystem if *GESTALT*, 2018 efforts to improve the state of the art in object detection from remote sensing imagery focused on developing datasets for training and evaluating models. The xView project supports the detection of 60 classes of objects [8] using horizontal bounding boxes. The DOTA project supports a much more modest 20 classes [29]. Both focus towards shipping and industrial applications and so will not generalize well. The 2021 update to the DOTA project highlighted that remote sensing object detection continues to suffer from the arbitrary rotation of objects and the vast disparities in the clustering of objects [1]. DOTA version 2 tries to address these issues by employing orientation bounding boxes but is still very constrained in the classes of objects it supports. A separate effort by Li et al. in 2020 can detect objects less constrained to heavy industry but only accounts for 20 or so objects [9]. Overall current work on remote-sensing object detection indicates that it is still an emerging field incapable of supporting the labeling of micro-terrain features required for the proposed system. An area growing parallel with remote sensing object detection is image captioning and visual question answering. In addition to implementing previously discussed object detection techniques, they employ alternate data sources to augment their ability to provide answers to natural language questions about remote sensing imagery. In 2017 Shi and Zou demonstrated that it is possible to automate caption generation for remote sensing imagery. However, their experimentation showed it to be ineffective at tasks like counting objects [22], an essential requirement of micro-terrain analysis.

8.4 Visual Question Answering

The paper that initiated the domain of Remote Sensing Visual Question Answering (RSVQA) was published in 2020 by Lobry et al. and showed an excellent ability to answer direct questions about a given remote sensing image, including area estimates, object counts and determining the relative locations of objects [11]. These capabilities are all helpful in the concept mapping component of *GESTALT*. Importantly they also incorporated geospatial information from OpenStreetMap into their system. A fundamental limitation they identified is that the lack of information in OSM about specific micro-terrain features and the inability of object detection models to provide it presents a significant gap holding back the advancement of the RSVQA field. My work may contribute towards closing that gap. Later work by Zheng et al. and Yuan et al. highlight that RSVQA is very much in its infancy, and they focus on improving the underlying models used in RSVQA systems [30, 31]. In addition to the gap identified by Lobry et al., one limitation is that the RSVQA approach focuses on answering questions about the things the user is already looking at. In my partial information use case, the user doesn't know exactly where to look, so using the RSVQA approach would render no improvement to performance over the visual inspection itself. The real challenge is identifying where to look, not what they are already looking at.

8.5 Geospatial Question Answering

Towards identifying where the user could be looking, the field of geospatial question answering, related to geospatial information retrieval, offers promising directions. In 2018 Punjani et al. sought to determine whether geospatial information could be incorporated into a question-answering system [18]. They used the established Frankenstein variant of the Qanary approach to developing question-answering pipelines to develop GeoQA. GeoQA can answer questions in several useful geospatial categories, including point, range, and property-based queries. They built their system on linked data collected from OpenStreetMaps and WikiData. More recently, the efforts to develop WorldKG, a geospatial knowledge graph of the world, extend the linked-data approach and allow users to generate SPARQL queries to answer complex geospatial questions across the fused knowledge of OpenStreetMaps, DBPedia and WikiData [2]. These linked data approaches to geospatial question answering are valuable advancements but do not include enough micro-terrain detail to satisfy the requirement of my project to allow a user to find a location based on partial information about the micro-terrain of the location.

8.6 Pictoral Querying

Prior work in pictoral querying shows the utility of identifying locations from maps based on knowledge of where a subset of locations on a given map are [24]. Their approach focuses on matching locations from a user-defined pictorial input to an underlying database of maps. In the 27 years since Soffer and Samet first specified the Pictoral Query Language, considerable advances in digital storage and access to geospatial data have addressed some of the initial scale limitations that the conversion of maps to digital pictorial representations presents. For example, the role of a system like MAGELLAN [21] in *GESTALT* is replaced by sourcing the locations from OSM, and the requirement for a separate system to efficiently index and query map tiles like MARCO [20] is now handled by OSM in its entirety. Additional work in pictorial querying notes the problem of searching when there are multiple instances of the objects. The problem is NP-Complete when formulated as a subgraph matching problem [3]. *GESTALT* addresses the complexity issues by limiting the size of the initial search space to the *last-mile* search over a region and by pruning results at each step, only checking locations that pass the set-membership test of the bloom filter, for example. The query interface is the most appealing part of the work in Pictoral Querying. Being able to specify queries pictorially, as a user's internal concept map, to *GESTALT*, is likely to improve user experience and leverage the benefits of human geospatial memory and will be considered for future work.

8.7 Human Spatial Cognition

9 CONCLUSION

9.1 Future Work

Throughout this paper, we identify many avenues for future work. Sections **??** and 3 explain the requirement for large-scale pictorial to geospatial scene mapping to enable the large-scale identification of objects to fuel *GESTALT*'s search. Section **??** highlights the need to develop datasets in dense suburban and urban locations to enable robust testing of the ownership assignment process. Sections 3 and **??** identify the requirement to trial the DVBSCAN algorithm to improve the ownership assignment process and the need to test the concept mapping proposed using KD-Trees robustly. Sections **??** and 8 emphasize the need for a user query interface. Work on pictoral querying offers an exciting direction that enables abstracted user querying and leverages the cognitive advantages of geospatially constructing their query. The search component of *GESTALT* needs to be tested at scale. While section **??** notes that the assumption of only a *last-mile* search allows us to assume small datasets, the performance of the belling tool discussed in section 8 shows how quickly performance degrades if not managed well. Finally, and most importantly, though the psychology literature indicates that the *GESTALT* approach should be helpful to a searcher, there is no work evaluating this theory and measuring the extent to which it is functional. A user study should be prioritized for a fully functional *GESTALT* prototype before it expands to full scale.

9.2 Conclusion

The *GESTALT* project aims to reduce the time a searcher spends on the *last-mile* of searching for a location. It assumes that the *last-mile* is in a constrained geographical region and allows users to search for objects they are likely to remember from candidate locations. *GESTALT* is designed to collect geospatial information about locations and objects within geographic regions from open-source geospatial and pictorial data. It infers the associations between objects visible to searchers in the real world and the location that they belong to and stores them using bloom filters and KD-Trees for efficient representation and search. *GESTALT* implements concept mapping to allow a user to query the implicit geospatial relations between objects in candidate locations, leveraging the inherent ordering of the multidimensional KD-Tree data structure for efficient search. *GESTALT* demonstrates at a trim level, on the Swan Valley wineries dataset, that the approach is feasible and identifies future work in scaling it.

FUTURE: online clustering: [14]

REFERENCES

[1] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, et al. 2021. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE transactions on pattern analysis and machine intelligence* 44, 11 (2021), 7778–7796. https://ieeexplore.ieee.org/iel7/34/4359286/09560031.pdf

[2] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. 2021. Worldkg: A world-scale geographic knowledge graph. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4475–4484. https://dl.acm.org/doi/pdf/10.1145/3459637.3482023

[3] Andre Folkers, Hanan Samet, and Aya Soffer. 2000. Processing pictorial queries with multiple instances using isomorphic subgraphs. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 4. IEEE, 51–54. https://ieeexplore.ieee.org/iel5/7237/19521/00902863.pdf

[4] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive computing* 7, 4 (2008), 12–18. https://ieeexplore.ieee.org/iel5/7756/4653458/04653466.pdf

[5] James Hays and Alexei A Efros. 2008. IM2GPS: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition*. IEEE, 1–8. https://ieeexplore.ieee.org/iel5/4558014/4587335/04587784.pdf

[6] Jason Helbing, Dejan Draschkow, and Melissa L.-H. Võ. 2020. Search superiority: Goal-directed attentional allocation creates more reliable incidental identity and location memory than explicit encoding in naturalistic virtual environments. *Cognition* 196 (2020), 104147. https://doi.org/10.1016/j.cognition.2019.104147

[7] David Keatley, Chris O'Donnell, Brendan Chapman, and David D Clarke. 2021. The psycho-criminology of burial sites: developing the winthropping method for locating clandestine burial sites. *Journal of Police and Criminal Psychology* (2021), 1–10.

[8] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. 2018. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856* (2018). https://arxiv.org/pdf/1802.07856

[9] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. 2020. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing* 159 (2020), 296–307. https://reader.elsevier.com/reader/sd/pii/S0924271619302825?token=92C775A3C0FC0E9B054597B8CECD1F9DCA518AA4E715A2B5AAFD68F889D4B63D91C5053CB6D0 originRegion=us-east-1&originCreation=20230312185336

[10] Yang Liu, Jie Jiang, Jiahao Sun, Liang Bai, and Qi Wang. 2020. A survey of depth estimation based on computer vision. In *2020 IEEE Fifth international conference on data science in cyberspace (DSC)*. IEEE, 135–141. https://ieeexplore.ieee.org/iel7/9169739/9172417/09172861.pdf

[11] Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. 2020. RSVQA: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* 58, 12 (2020), 8555–8566. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9088993

[12] Jonathan F Miller, Markus Neufang, Alec Solway, Armin Brandt, Michael Trippel, Irina Mader, Stefan Hefft, Max Merkow, Sean M Polyn, Joshua Jacobs, et al. 2013. Neural activity in human hippocampal formation reveals the spatial context of retrieved memories. *Science* 342, 6162 (2013), 1111–1114. https://www.science.org/doi/pdf/10.1126/science.1244056

[13] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. 2021. Deep learning for monocular depth estimation: A review. *Neurocomputing* 438 (2021), 14–33. https://www.sciencedirect.com/science/article/pii/S0925231220320014

[14] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, et al. 2021. River: machine learning for streaming data in Python. (2021).

[15] Peter Mooney, Marco Minghini, et al. 2017. A review of OpenStreetMap data. *Mapping and the citizen sensor* (2017), 37–59. https://library.oapen.org/bitstream/handle/20.500.12657/31138/1/637890.pdf#page=46

[16] Mark-Christoph Müller and Michael Strube. 2012. Transparent, efficient, and robust word embedding access with WOMBAT. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. August 20-26, 2018, Santa Fe, New Mexico, USA*. Association for Computational Linguistics, 53–57. https://aclanthology.org/C18-2012.pdf

[17] Miguel Oliveira, Luís Seabra Lopes, Gi Hyun Lim, S. Hamidreza Kasaei, Ana Maria Tomé, and Aneesh Chauhan. 2016. 3D object perception and perceptual learning in the RACE project. *Robotics and Autonomous Systems* 75 (2016), 614–626. https://doi.org/10.1016/j.robot.2015.09.019

[18] Dharmen Punjani, Kuldeep Singh, Andreas Both, Manolis Koubarakis, Iosif Angelidis, Konstantina Bereta, Themis Beris, Dimitris Bilidas, Theofilos Ioannidis, Nikolaos Karalis, et al. 2018. Template-based question answering over linked geospatial data. In *Proceedings of the 12th workshop on geographic information retrieval*. 1–10. https://dl.acm.org/doi/pdf/10.1145/3281354.3281362

[19] Anant Ram, Sunita Jalal, Anand S Jalal, and Manoj Kumar. 2010. A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications* 3, 6 (2010), 1–4. https://www.researchgate.net/profile/Anant-Ram/publication/282334683_A_Density_based_Algorithm_for_Discovering_Density_Varied_Clusters_in_Large_Spatial_Databases/links/560d049d08aed543358d5e32/A-Density-based-Algorithm-for-Discovering-Density-Varied-Clusters-in-Large-Spatial-Databases.pdf

[20] Hanan Samet and Aya Soffer. 1996. Marco: Map retrieval by content. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 8 (1996), 783–798. https://ieeexplore.ieee.org/iel1/34/11231/00531799.pdf

[21] Hanan Samet and Aya Soffer. 1998. Magellan: Map acquisition of geographic labels by legend analysis. *International journal on document analysis and recognition* 1 (1998), 89–101. https://link.springer.com/content/pdf/10.1007/s100320050009.pdf

[22] Zhenwei Shi and Zhengxia Zou. 2017. Can a machine generate humanlike language descriptions for a remote sensing image? *IEEE Transactions on Geoscience and Remote Sensing* 55, 6 (2017), 3623–3634. https://ieeexplore.ieee.org/iel7/36/4358825/07891049.pdf

[23] Noah Snavely. 2011. Scene reconstruction and visualization from internet photo collections: A survey. *IPSJ Transactions on Computer Vision and Applications* 3 (2011), 44–66. https://www.jstage.jst.go.jp/article/ipsjtcva/3/0/3_0_44/_pdf

[24] Aya Soffer and Hanan Samet. 1997. Pictorial query specification for browsing through image databasess. In *Proceedings of the Second International Conference on Visual Information Systems*. 117–124. https://www.cs.umd.edu/~hjs/pubs/soffervlc.pdf

[25] Zhehang Tong, Dianxi Shi, Bingzheng Yan, and Jing Wei. 2017. A review of indoor-outdoor scene classification. In *2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*. Atlantis Press, 469–474. https://www.atlantis-press.com/article/25881214.pdf

[26] John E Vargas-Munoz, Shivangi Srivastava, Devis Tuia, and Alexandre X Falcao. 2020. OpenStreetMap: Challenges and opportunities in machine learning and remote sensing. *IEEE Geoscience and Remote Sensing Magazine* 9, 1 (2020), 184–199. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9119753

[27] Steven M Weisberg and Nora S Newcombe. 2016. How do (some) people make a cognitive map? Routes, places, and working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 42, 5 (2016), 768.

[28] Logan Williams. 2023. Finding Geolocation Leads with Bellingcat's OpenStreetMap Search Tool. *Bellingcat Blog* (May 2023). https://www.bellingcat.com/resources/how-tos/2023/05/08/finding-geolocation-leads-with-bellingcats-openstreetmap-search-tool/

[29] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. 2018. DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3974–3983. http://openaccess.thecvf.com/content_cvpr_2018/papers/Xia_DOTA_A_Large-Scale_CVPR_2018_paper.pdf

[30] Zhenghang Yuan, Lichao Mou, Qi Wang, and Xiao Xiang Zhu. 2022. From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–11. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9771224

[31] Xiangtao Zheng, Binqiang Wang, Xingqian Du, and Xiaoqiang Lu. 2021. Mutual attention inception network for remote sensing visual question answering. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–14. https://ieeexplore.ieee.org/iel7/36/4358825/09444570.pdf