ConverSAT: Efficient Satelite Imagery Querying.

Ridha Alkhabaz
University of Illinois Urbana-Champaign
Urbana, Illinois, USA
ridhama2@illinois.edu

Hari Umesh
University of Illinois Urbana-Champaign
Urbana, Illinois, USA
humesh2@illinois.edu

Vibha Nanda
University of Illinois Urbana-Champaign
Urbana, Illinois, USA
vibhan2@illinois.edu

Khushi Sidana
University of Illinois Urbana-Champaign
Urbana, Illinois, USA
ksidana2@illinois.edu

ABSTRACT

Current Geographic Information Systems (GIS) has high labor costs for its analytics operations. Furthermore, each GIS requires a special workflow and regularization to answer specific questions. Here, we are proposing an approach to automate the process of creating these workflows by utilizing the latest developments in Machine Learning, Database Systems, and Video Analytics [5, 8, 9, 14, 17]. Similar to the work done by Kang et al., we want to use considerably cheap neural networks to estimate selection and aggregation queries on satellite imagery [8]. Given the different nature of spatial imagery, we propose different kinds of indexing. We show a **twenty percent** speed-up and retention of a desired class for our proposed methods compared to naive applications of successful application of video analytics methods, like NoScope [9]. We also utilize specialized neural network and baseline models that exhibit **3x** speed-up because of our novel indexing and filtering.

ACM Reference Format:
Ridha Alkhabaz, Hari Umesh, Vibha Nanda, and Khushi Sidana. 2023. ConverSAT: Efficient Satelite Imagery Querying. . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Present Geographic Information Systems (GIS) provide great utility for various sectors of the economy, such as urban planning [15]. However, GIS systems usually require expensive manual labor from domain experts and computer scientists. Beyond manual labor, the data domain imposes many restrictions on the generalizability of current GIS. Thus, with the recent advances in areas like Video Analytics, we are proposing machine learning methods that provide significantly cheaper estimates [8, 9].

Motivation: Imagine an analyst trying to estimate the number of buildings in Bloomington, Illinois. The analyst would reach out to a GIS operator. GIS would retrieve relevant imagery from a private or public repository, like NASA LandSAT gallery [1]. Then, the operator would create a certain workflow to regularize and project the data. Afterward, they would impose certain heuristics to transform the data into a desired representation. Finally, they would use the manufactured representation to answer the analyst's question. As you may have noted, this paradigm was very common in video analytics literature before the recent advances in machine learning. Hence, we are motivated to utilize the current literature on video analytics and machine learning to decrease manual labor and expedite information retrieval in querying geospatial imagery.

Problem overview: Unlike Video frames, our images have seven channels. These channels include four depth channels and three color channels. Additionally, our images have a local dependency on each other compared to video frames exhibiting only sequential dependency. Hence, our solution should account for these key differences.

Given the size and magnitude of this problem, we take inspiration from different areas of computer science to answer selection and aggregation queries over geospatial imagery. Given indices of images and their coordinates, we build a Hilbert k-d tree to partition our dataset into local neighborhoods [4, 21]. Then, given the preserved neighborhood, we take a representative sample to see if a desired predicate relies within this neighborhood. We apply binary neural and regression-based classifiers to filter these neighborhoods. Finally, we gather neighborhoods where a desired predicate potentially resides and apply specialized neural networks and baseline models to answer a given query similar to the work done by Kang et al. [8, 9]. Using our novel filtering approach, we show **3x** speedup in detection. Moreover, we show that our filtering retains at least twenty percent more of the desired predicate images and achieves up to twenty percent speedup. Please note Figure 1 for an overview of our solution.

Our work will demonstrate the viability of machine learning methods to ease access to GIS information for different domain experts. It will allow distantly related expertise to have more enriched collaboration by removing the need for a computer scientist in-loop for such collaborations.

2 SYSTEM OVERVIEW

Our work has three main components: indexing, filtering and localization, and detection. In the following subsections, we will showcase each component extensively. Please note Figure 1 for an overview of our solution.

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

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

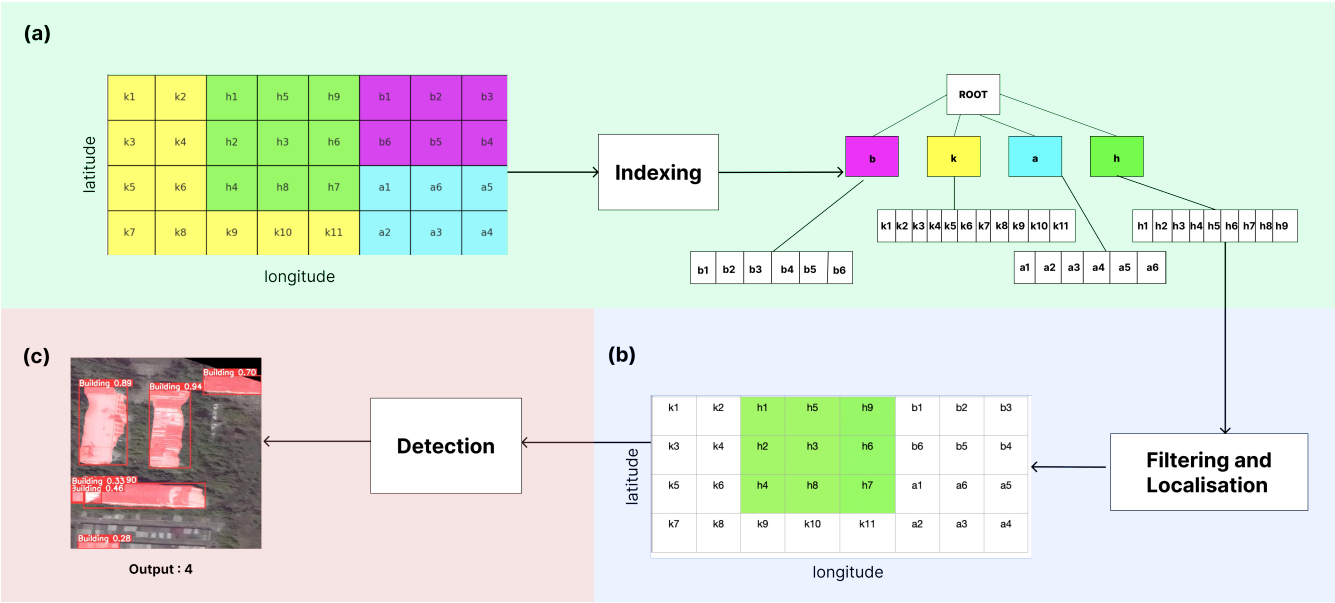

Figure 1: From top to bottom. Part (a) has an example of our partitioning and indexing. Part (b) shows our filtering approach. Part (c) illustrate our detection mechanism.

2.1 Indexing

We were highly inspired by Tsunami by Ding et al. [5]. Similar to Tsunami, we take into account data columns' correlations. Thus, we utilize the Hilbert curve Geohashing feature of preserving local neighborhoods to partition our records into regions. These regions are determined by the Hilbert curve Geohashing precision [7]. Please refer to Figure 6 to understand precision-induced hierarchy in Hilbert curves. Please note that a lower precision would decrease the number of regions and increase its size. On the other hand, this trade-off between the number of regions and their sizes would give us a huge speed-up in filtering on a potential cost of accuracy. With a lower number of regions, we need a lower number of samples, but these samples would not be as representative. This lack of representation is due to the diversity of features induced by the regions' size, i.e., larger plots of land might have more diverse features. After experimenting with many precisions, we created our current partition that divides our data into fifty-one regions with an approximate size of twenty-four square kilometers per region. We then create a K-d tree with unique keys for every region [4]. Please refer to the Figure 1 part (**a**) for an illustration.

2.2 Filtering & Localization

We were highly inspired by NoScope by Kang et al. [9]. As mentioned, our data domain imposes restrictions that prevent us from naively applying NoScope to our data. In Figure 3, we aimed to do a ten frames GAP sampling strategy for our data. Also, in Figure 3, we show the result of our sampling strategy. Our sampling strategy assumes geospatial features are uniformly distributed within relatively compact regions. Thus, instead of iterating over each sample in a region, we could generate a random representative set of samples. By our prior assumption, this sample set would be representative. Furthermore, we could apply our filtering models on less number of frames. Additionally, Figure 3 shows that our sampling strategy has more breadth in its dataset coverage.

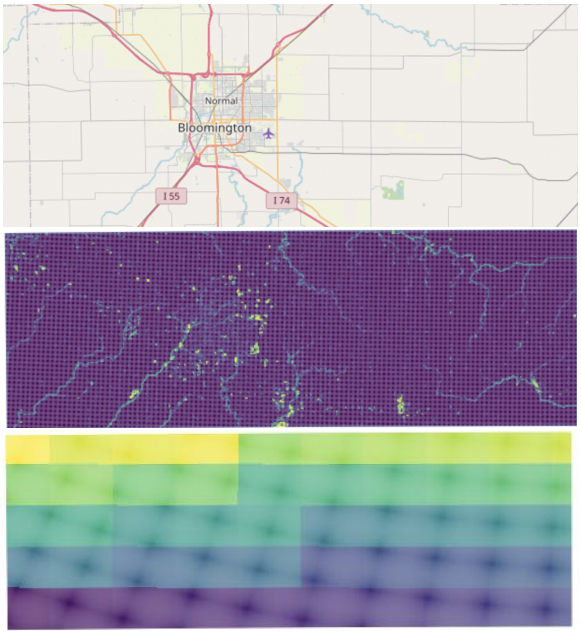

Figure 2: From top to bottom. Snapshot of our dataset using OpenStreetview [2]. Followed by a depth map produced by plotting the average of a depth measure per sample. Then, The partition of our samples as discussed in Section 2.1

We apply two models to test whether a predicate exists within a region. We used convolutional neural networks (**CNN**) trained on ten percent of data to classify samples, similar to Kang et al. [9]. Our convolutional networks had about three hundred thousand parameters. Another filtering method we utilized was to create a texture representation for each sample in our dataset. Our representation vector for each sample consisted of the mean and standard deviation for four depth channels. Then, we created logistic regression (**Log**) on top of a shallow neural network using these vector representations of samples to predict whether our frames had the desired predicate. Our logistic regression-based filtering model had about forty parameters. Given an arbitrarily chosen ratio, if the predicate existed more than the desired ratio, we passed the regions' data to the detection phase.

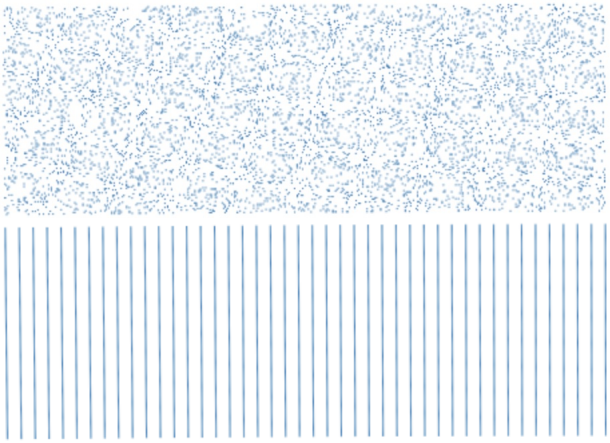

Figure 3: Filtering coverage using our filtering approach (UP) and NoScope naive implementation (Down).

2.3 Detection

We would have several samples at this stage that may contain our data demand. Given the recent advances in machine learning, we chose to utilize two types of detection methods. We were inspired by the incorporation of specialized neural networks in BlazeIt by Kang et al. [8]. Given their cheap training cost and versatility, we trained them using ten percent of our dataset. Our specialized neural networks were a variation of `ResNet` provided by `Pytorch`, a deep learning python library [6, 18]. We took a pre-trained version of `ResNet-34` and fine-tuned it for a desired task. We created two kinds of **Taget-Model annotated sets** as outlined by BlazeIt. We will discuss the training datasets and fine-tuning process further in section 4. We also looked into utilizing You Only Look Once (YOLO) by Redmon et al. [19] as a baseline model. We found a version of YOLO with fewer parameters and considerable speed-up compared to YOLOv4 used by Kang et al. in BlazeIt [8]. We chose the eighth version as a detection method for the previous purposes [10].

3 EVALUATION DETAILS

3.1 Data Overview

Our data is provided by the National Center for Supercomputing Applications (NCSA). We are collaborating with Dr. Aiman Soliman and research assistant Priyam Mazumdar at NCSA [20]. Our collaboration is mainly to supply us with Landsat imagery for the state of Illinois stored at NCSA archives. Our collaboration alleviated some pre-processing burdens on our project. We were supplied with masks for certain objects, such as buildings and roads. Moreover, we were provided ground truth labels for each sample for the number of buildings and their estimated area. These labels were generated by assuming buildings are at least six feet high. Using this assumption, they zeroed out pixels that did not match these criteria and generated the masks and labels. As a team, we did check a subset of the dataset using a **numIslands** approach on the generated masks to ensure its quality.

Moreover, our imagery has seven channels, consisting of the laser mapping channel and four more corresponding to specific features like texture or depth. Our laser channel is very similar to grey-scaled images in datasets like **MNIST** [13]; we are regularizing the shape of our imagery to be squares of 244 pixels in width and height. Furthermore, we plan to prototype our system over the city of Bloomington and its neighboring farmland. Then, we aim to expand our scope to account for the whole state of Illinois. We chose the city of Bloomington as our case study to detect early issues in our system based on the advice from our collaborators. Their advice noted the City of Bloomington's variety of urban and farmland landscapes. This makes Bloomington a good model for the state of Illinois. Please refer to Figure 2 for an overview of our data[1].

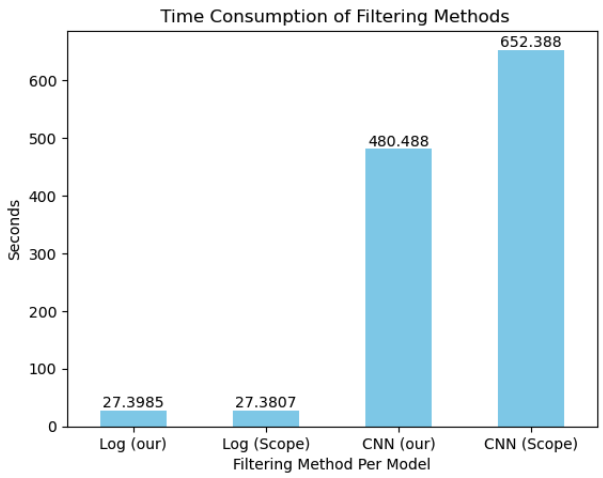

Figure 4: Execution time for different different combination filtering approaches and models. Please note (our) refer to our filtering approach detailed in sections 2.1 and 2.2. Also, (Scope) refer to applying a naive version of NoScope.

[1]You can find a copy of our data here.

Model (Filtering Method)	Time	Estimate	Ground Truth (Filtered set)
-	Seconds	Num of Buildings	True number (in our filtered set)
YOLOv8 (Naive)	> 11520	NA	66685 (53079)
YOLOv8 (Our filtering)	≈ 2143	62438	66685 (53079)
Specialized NN (Naive)	≈ 2509	62843	66685 (53079)
Specialized NN (Our filtering)	≈ 779	12895	66685 (53079)

Table 1: Here are the results for answering (Q1). We used Log filtering model with sampling using 10% percent per region.

3.2 Experimental details and Queries

We divided our evaluation experiments into two folds[2]. First, we investigated the speed-up we achieved due to our clever indexing and filtering in finding our desired samples/frames. Please see our results in Figures 4 and 5. We will discuss our results further in Section 4. Second, we will test how far from the ground truth given an aggregation query, such as **Q**1. We investigated the trade-off between an expensive and relatively cheap detection method. We applied a version of YOLO trained on publically available data and specialized neural networks used in BlazeIt [3, 8, 19]. Please refer to Table 1 for further details.

Listing 1: (Q1) our testing query.

```
SELECT fCOUNT(*)
FROM BLOOMINGTON
WHERE Obeject == 'Building'
At Confidence 67%
```

3.3 Hardware Details

We have been pleased by NCSA's support in this project. We are using the Hardware-Accelerated Learning (HAL) cluster's resources. These resources consist of graphical processing units (GPU) and one terabyte worth of storage in a shared directory for our project [11]. For replication purposes, HAL GPUs are NVIDIA V100 16 GB HBM 2. We send daily requests to use these GPUs' for a user-specified period of time. For small experiments and code debugging, we utilize our personal laptops. Our personal devices are mainly from the Apple Silicon line-up.

4 DISCUSSION

4.1 Indexing & Filtering

We noticed considerable gains in Figures 5 and 4 because of our filtering approach. One is the minimum additional retention of twenty percent of images with a desired class, like images with buildings. Moreover, this retention is not attributed to randomness since our results were obtained by running our experiments four times and taking the mean of the results. We also notice a significant speed-up when using a **CNN** filtering model. We argue that due to the shallow nature of our regression-based model (**Log**), we do not see any speed-up in the filtering part of our solution. However, at the inference/detection stage, we find more than **3x** speed-up because of our filtering approach compared to naively approaching our detection models.

[2]You can find our code here.

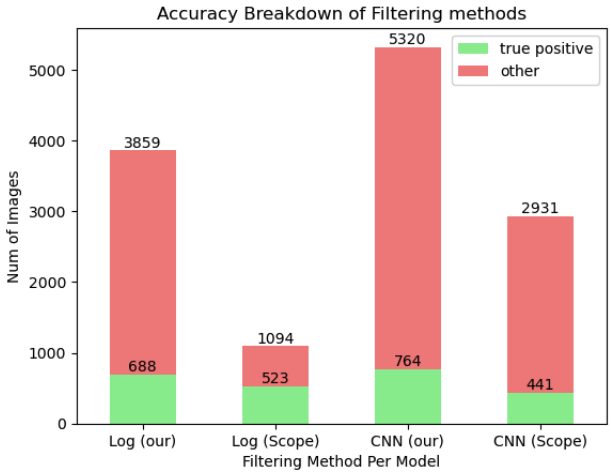

Figure 5: Filtering outcomes for different different combination filtering approaches and models. Please note (our) refers to our filtering approach detailed in sections 2.1 and 2.2. Also, (Scope) refers to applying a naive version of NoScope.

4.2 Detection: Specialized Neural Networks

Inspired by Kang et al., we used a pre-trained version of `ResNet-34` to specialize for a certain task. Our task was to classify each image by the number of buildings in it. Here, we run into a couple of problems. One of which is class imbalance. Since Bloomington is surrounded by farmland, ninety percent of our images contained no buildings. Moreover, few images had a high number of buildings. For instance, two images in our data had fifty-nine buildings. Thus, we had to incorporate some transformation measures to increase the number of examples of some classes. For instance, for any classes with fewer than ten examples, we applied random rotation and added Gaussian noise to add more examples. Furthermore, we noticed we lacked examples for two classes: images with fifty-seven and fifty-eight buildings. So, we fine-tuned a version of `ResNet-34` with missing classes and another version to classify images with several buildings up to fifty-six buildings. These versions had two different **Target-model annotated sets**. We saw a considerable improvement with the later version of `ResNet-34`. However, as you may note in Table 1, our best-specialized model still significantly underestimated the number of buildings.

5 PREVIOUS WORK

Video Analytics: We drew inspiration from two prominent systems, BlazeIt and NoScope, to gain insights into the dynamics of the latest advancements in video analytics technology [8, 9]. **BlazeIt** is a system that optimizes queries of spatiotemporal information of objects in video. BlazeIt accepts queries via FRAMEQL, a declarative extension of SQL for video analytics that enables video-specific query optimization. It introduces two new query optimization techniques: using neural networks as control variates to quickly answer approximate aggregation queries with error bounds and a novel search algorithm for cardinality-limited video queries. BlazeIt can deliver up to 83x speedups over the recent literature on video processing. BlazeIt has a few limitations. First, it is limited to queries that can be expressed in FrameQL. Second, it may not be as efficient as imperative code for some queries. Third, it does not support all types of queries, such as joins or subqueries. Fourth, it is still under development and may have bugs or performance issues. **NoScope**: Noscope introduces a system that analyzes large-scale video datasets efficiently. The authors address the challenge of applying neural networks to video data by proposing a novel indexing structure that optimizes queries for specific spatiotemporal patterns. NoScope achieves significant query processing speed by filtering irrelevant frames early in the pipeline, reducing the computational burden on the neural network. The system demonstrates improved performance over traditional approaches in terms of both accuracy and efficiency when applied to real-world video datasets. NoScope has limitations such as potential false negatives due to its frame-level processing, dependency on a specific neural network architecture, and challenges with diverse video resolutions. The need for manual parameter tuning also hinders its out-of-the-box applicability. Despite these constraints, the paper contributes to efficient video analysis at scale, providing insights for future developments in the field.

Geographic Information Systems: We also applied tools and techniques from the vast research on autonomous Geographic Information Systems (GIS). We mainly drew references from two sources, Autonomous GIS and Spatial databases. Li and Ning propose a new concept called Autonomous GIS, which is an AI-powered GIS that leverages large language models (LLMs) to address spatial problems with minimal to no human intervention automatically [14]. It outlines five autonomous goals for Autonomous GIS: self-generating, self-organizing, self-verifying, self-executing, and self-growing. The authors developed a prototype system called LLM-Geo to demonstrate the feasibility of Autonomous GIS. LLM-Geo can handle natural language instructions, collect spatial data, perform spatial analysis, and generate visualizations without human input.

Autonomous GIS is a novel concept with promising potential for revolutionizing GIS technology. However, it is still in its early stages of development and faces several limitations. First, it relies heavily on the capabilities of LLMs, which may not always be able to interpret and execute complex spatial queries accurately. Second, Autonomous GIS may struggle to handle large and complex datasets, potentially leading to performance slowdowns or inaccurate results. Third, the lack of a standardized interface or framework for Autonomous GIS could hinder its adoption and integration with existing GIS workflows.

Laurini presents a new approach to **spatial multi-database** querying that ensures the topological continuity of fragmented objects [12]. This approach is based on constructing an r-tree-like structure that organizes global indices. The authors also present special tools for ensuring the continuity of fragmented objects semantically, topologically, and geometrically. These tools can be used to accelerate queries against fragmented objects. The authors also describe how to use adjacency tables to accelerate queries further.

However, the proposed approach is limited to spatial multi-database queries that involve fragmented objects. The approach may not be as efficient as traditional spatial database querying approaches for queries that do not involve fragmented objects. Additionally, the approach relies on constructing an r-tree-like structure, which can be computationally expensive for large datasets. As a result, the approach may not be scalable to large datasets. Finally, the authors do not provide a detailed implementation of their approach, so assessing its performance in real-world scenarios is difficult. As a result, it is difficult to say whether the approach is practical for real-world applications.

Hilbert curve: Meng et al. propose enhancements to the Hilbert Curve [16], a space-filling curve used for spatial data indexing. The authors introduce improvements aimed at achieving more efficient parallel spatial data partitioning. The research focuses on optimizing the Hilbert Curve to enhance its performance in parallel computing, specifically for spatial data applications.

The paper acknowledges limitations in achieving optimal load balancing for irregularly distributed spatial data using the enhanced Hilbert Curve proposed for parallel processing. Uneven workloads among parallel processors and sensitivity to varying system architectures are recognized challenges. Nonetheless, the research provides valuable insights into optimizing spatial data partitioning in parallel computing, highlighting the need for future work to address irregular data distributions.

6 LIMITATIONS

Due to resource and time allocation constraints, our current system is designed to address specific query types. We envision substantial potential for future research and development by expanding the system's capabilities to encompass LIMIT and JOIN queries. This expansion would enhance the system's adaptability and utility, providing a more comprehensive solution for a broader array of queries and data retrieval needs.

Our system's geographical coverage is focused solely on the Bloomington, Indiana area as of the present phase. This localized scope was determined due to limitations imposed by our available resources. Nonetheless, there exists an opportunity to extend this system's geographical reach to encompass other regions. This extension would amplify the system's applicability and foster a valuable resource for Geographic Information System (GIS) researchers. The integration of other regions would benefit researchers and serve as a catalyst for advancing spatial analysis and decision-making processes across diverse geographical landscapes. Expanding our scope would also force us to challenge our uniform predicate distribution assumption.

Given the rich nature of our data, we focused on specific aspects of the data. For instance, we only used our depth channels' mean and standard deviation and greyscaled our images for faster inference execution. Thus, it is fair to assume utilizing more data might help us reduce our errors.

7 FUTURE WORK

There are many things to be addressed in our present work. First, given the limited time and resources, we couldn't fine-tune specialized neural networks beyond sixty-seven percent validation accuracy. Second, our assumption of predicate uniform distribution needs to be revisited. It might be more beneficial to have a texture-based partition in the future. Third, our solution only attempts selection and aggregation queries. Extending our system to more query types, such as `JOIN`, would be beneficial. Additionaly, our evaluation scheme only contained an aggreagtation query. We would like to include more selection and aggregation queries in our future work.

8 CONCLUSION

In conclusion, this study demonstrates the significant potential of specialized models in enhancing the efficiency of Geographic Information Systems (GIS). By leveraging advancements in geospatial image filtering and specialized models, we have successfully addressed the limitations of traditional GIS, which typically rely on expensive and labor-intensive processes conducted by domain experts and computer scientists. Our approach, which involves a novel combination of Hilbert k-d trees for data partitioning and advanced neural network models for efficient querying and analysis, has shown promising results in speeding up the information retrieval process and reducing manual labor.

The experiments revealed a 3x speedup in detection times and a 20% improvement in retaining desired predicate images compared to traditional methods. These improvements are significant, as they not only enhance the efficiency of GIS but also broaden its accessibility to professionals from various fields who may not possess in-depth computer science expertise. The ability to quickly and accurately analyze vast amounts of geospatial imagery has far-reaching implications for urban planning, environmental monitoring, and other sectors that rely heavily on GIS data. As larger computing resources continue to evolve, it is crucial to continue exploring these avenues to advance the field further and expand the capabilities of GIS to meet the growing demands of our society.

REFERENCES

[1] [n. d.]. https://landsat.visibleearth.nasa.gov/

[2] [n. d.]. https://www.openstreetmap.org/export#map=12/40.5115/-88.9910

[3] A. Azizi, M. Yaghoobi, and S. R. Kamel. 2023. *Intelligent detection and assessment of damaged buildings using UAV imagery and yolov8* (2023). https://doi.org/10.21203/rs.3.rs-3143110/v1

[4] Jon Louis Bentley. 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* 18, 9 (sep 1975), 509–517. https://doi.org/10.1145/361002.361007

[5] Jialin Ding, Vikram Nathan, Mohammad Alizadeh, and Tim Kraska. 2020. Tsunami: A Learned Multi-Dimensional Index for Correlated Data and Skewed Workloads. *Proc. VLDB Endow.* 14, 2 (oct 2020), 74–86. https://doi.org/10.14778/3425879.3425880

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[7] Tammo Ippen. 2020. Geohash-Hilbert. https://pypi.org/project/geohash-hilbert/

[8] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. Blazeit. *Proceedings of the VLDB Endowment* 13, 4 (Nov 2019), 533–546. https://doi.org/10.14778/3372716.3372725

[9] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. Noscope. *Proceedings of the VLDB Endowment* 10, 11 (Aug 2017), 1586–1597. https://doi.org/10.14778/3137628.3137664

[10] Keremberke. 2023. Keremberke/Awesome-yolov8-models: Easy-to-use finetuned Yolov8 models. https://github.com/keremberke/awesome-yolov8-models

[11] Volodymyr Kindratenko, Dawei Mu, Yan Zhan, John Maloney, Sayed Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, and William Gropp. 2020. Hal: Computer system for scalable deep learning. *Practice and Experience in Advanced Research Computing* (2020). https://doi.org/10.1145/3311790.3396649

[12] Robert Laurini. 1998. Spatial multi-database topological continuity and indexing: A step towards seamless GIS data interoperability. *International Journal of Geographical Information Science* 12, 4 (1998), 373–402. https://doi.org/10.1080/136588198241842

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[14] Zhenlong Li and Huan Ning. 2023. Autonomous GIS: The next-generation AI-powered GIS. https://arxiv.org/abs/2305.06453

[15] Jacek Malczewski. 2004. GIS-based land-use suitability analysis: A critical overview. *Progress in Planning* 62, 1 (Jan 2004), 3–65. https://doi.org/10.1016/j.progress.2003.09.002

[16] Lingkui Meng, Changqing Huang, Chunyu Zhao, and Zhiyong Lin. 2007. An improved Hilbert curve for parallel spatial data partitioning. *Geo-spatial Information Science* 10, 4 (2007), 282–286. https://doi.org/10.1007/s11806-007-0107-z

[17] Vikram Nathan, Jialin Ding, Mohammad Alizadeh, and Tim Kraska. 2020. Learning Multi-Dimensional Indexes. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 985–1000. https://doi.org/10.1145/3318464.3380579

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640 [cs.CV]

[20] Aiman Soliman. 2020. Meet our Faculty: Aiman Soliman. https://urban.illinois.edu/people/profiles/aiman-soliman/

[21] Tibor Vukovic. 2016. Hilbert-Geohash - Hashing Geographical Point Data Using the Hilbert Space-Filling Curve. https://api.semanticscholar.org/CorpusID:85555593

9 APPENDIX

Please look at the following to understand the hierarchy induced by Hilbert curve geohashing methods.

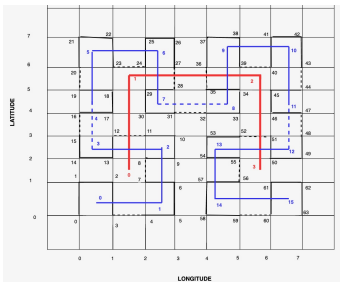

Figure 6: The different lines correspond to different Hilbert curves with vastly different specified precision. As you may note, from red to blue, to black, the precision increases.