Seiden: Query Processing in Video Database Systems

Project Report submitted by

Daneti Prasanth (421131)
Dushetti Manideep (421138)
Dondapati Syam Suhith (421136)

Under the supervision of

Dr. Nagesh bhattu S
Assistant Professor

Department of Computer Science and Engineering,
National Institute of Technology, Andhra Pradesh

06-05-24

DECLARATION

I hereby certify that the thoughts articulated in this written material are entirely mine unless otherwise acknowledged. In addition I pledge that I have not contravened any academic honesty and integrity protocol by fabricating or misrepresenting any information, data or source in this document. Should I violate any of these rules, I understand that the institute will take punitive measures against me which may include legal action against parties whose copyrights were infringed upon or from whom consent was not sought where necessary.

Daneti Prasanth	Dushetti Manideep	Dondapati Syam Suhith
421131	421138	421136
Date:__________	Date:__________	Date:__________

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH

Certificate

This is to certify the thesis entitled "Seiden: Revisiting Query Processing in Video Database Systems" submitted by Daneti Prasanth (421131), Dushetti Manideep (421138), Dondapati Syam Suhith (421136) to National Institute of Technology, Tadepalligudam in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide research work carried out by them under my supervision and guidance. This work has not been submitted elsewhere for the award of any degree

Dr. Nagesh bhattu S
Assistant Professor
Dept. of CSE

Place: Tadepalligudam
Date:

September 11, 2024

Abstract

Recent updates in video database systems have really changed the game, especially with the new, lighter Oracle models like YOLOv5s. Take "Seiden," for instance, a top-notch system that really makes the most of these improvements to step up both the speed and accuracy of processing video queries. Unlike the older systems that used proxy models, Seiden goes straight for the Oracle model, applying it directly to a select bunch of frames to create a base that don't depend on the query. When it is time to execute the given query, Seiden uses a clever technique based on the multi-arm bandit algorithm, which smartly chooses between looking into new parts of the video and focusing on areas that are likely to give valuable results. This method has made things up to 6.6 times faster than the old ways and has made the results more reliable across different queries and datasets. By ditching proxy models for direct use of Oracle models and using these sharp sampling strategies, Seiden is setting new standards in how we handle video queries, making it a real leader in the field of video analytics.

Table of Contents

1 Introduction **5**

2 Literature Review **6**
2.1 Oracle Models 6
2.2 Proxy Models 6
2.3 SEIDEN 6

3 Methodology **7**
3.1 Query Processing Overview 8
3.2 Model Architecture 8
3.2.1 Sampling Ratio 9
3.2.2 Index Construction 10
3.2.3 Query Execution 10
3.2.4 Label Propagation 12

4 Experimental Results and Analysis **13**
4.1 Datasets 13
4.2 Evaluation metrics 13
4.3 Evaluation on proxy system and seiden 14

5 Future Scope and Conclusion **15**
5.1 Conclusion 15
5.2 Future Scope 15
5.3 Acknowledgement 15

List of Figures

3.1 Proxy System 7
3.2 Index construction 9
3.3 Query Execution 9
3.4 Index construction algorithm 10
3.5 Reward 11

List of Tables

4.1 Some examples for F1-Scores of tasti-pt and Seiden with reuse 14

Chapter 1

Introduction

In today's digital era, where video content dominates data traffic, efficiently managing and querying large-scale video databases is more crucial than ever. Recognizing this need, the Seiden system emerges as a groundbreaking solution, designed to handle complex video queries swiftly and effectively. This project introduces an innovative approach to video database management by leveraging the latest advancements in computer vision and machine learning.

Seiden distinguishes itself by employing a method known as Approximate Query Processing (AQP), which intelligently balances speed and accuracy. Instead of analyzing every single frame within a video, Seiden smartly samples keyframes and utilizes powerful predictive models to estimate the answers to queries. This strategy allows Seiden to provide quick, approximate answers that can be refined progressively, offering a practical solution for scenarios where timely responses are more valuable than exact precision.

At the heart of Seiden's efficiency is its use of an exploration-exploitation technique derived from the multi-armed bandit problem in statistics. This method helps Seiden decide which video segments to focus on during query processing, optimizing the use of computational resources by concentrating on segments that are likely to yield the most valuable information for a given query.

Through its intelligent sampling and adaptive processing strategies, Seiden not only accelerates query response times but also reduces computational overhead, making it a particularly appealing choice for large-scale applications. Whether it's monitoring traffic intersections, managing surveillance footage, or analyzing broadcast content, Seiden offers a robust framework that stands to transform how we interact with and extract value from video data in a rapidly evolving digital landscape.

This project not only highlights the capabilities of the Seiden system but also illustrates its potential impact on industries reliant on large volumes of video data, paving the way for more responsive, efficient, and intelligent video database management systems.

Chapter 2

Literature Review

2.1 Oracle Models

In the world of systems that manage and analyze video databases, oracle models are incredibly important for their accuracy and detail. These models use complex machine learning or deep learning techniques to carefully examine video data and pull out clear and detailed information. Unlike simpler models that don't dig as deep and use less computing power, oracle models go into great detail and provide very precise information about what's in the video. Although they need a lot of computer power, especially to work quickly in real-time, the detailed and accurate information they provide is very valuable in situations where being exact is critical.

2.2 Proxy Models

Proxy models are simpler forms of more complex systems, helping to quickly process large amounts of data. They give results that are close enough to accurate for many uses, reducing the need for heavy computer processing and making things faster. This is especially helpful in situations where quick decisions are necessary, like when watching video feeds to spot dangers immediately or studying how customers behave in stores. By using proxy models, systems can manage large data more easily, leading to quicker decisions and better efficiency without overwhelming the computer system.

2.3 SEIDEN

Seiden is a system built to manage and analyze big collections of video data quickly and accurately. It uses simpler models called lightweight oracle models to keep computer usage low while still getting accurate results. Seiden smartly picks specific video frames to analyze using advanced technology, which helps it respond to questions much faster than older methods. It's really good at dealing with different types of data queries, making it perfect for tasks like watching over traffic or keeping an eye on security. Seiden mixes the speed of real-time processing with the detailed analysis from deep learning, improving both the speed and accuracy of video analysis.

Chapter 3

Methodology

Our project took a direct and simple approach. We began by gathering videos, a query, and utilizing an oracle model as our starting inputs. Additionally, in proxy systems, we employed models like Resnet-18 to give each video frame a proxy score. In this setup, a frame's proxy score for retrieval queries can range from 0 to 1, which reflects how likely it is to fulfill the query's criteria. For aggregate queries, the proxy score gives an approximation of the total for that frame. Initially, the oracle model identifies I-frames within the video. As the process advances, more frames are chosen based on the continuing action in the video. These scores, whether from a proxy system or not, play a crucial role in powering an Approximate Query Processing (AQP) algorithm. This algorithm is designed to make sure the results align with the user's accuracy requirements, whether it involves selecting specific frames or summarizing data. The final output of this AQP algorithm is what we present as the query result.

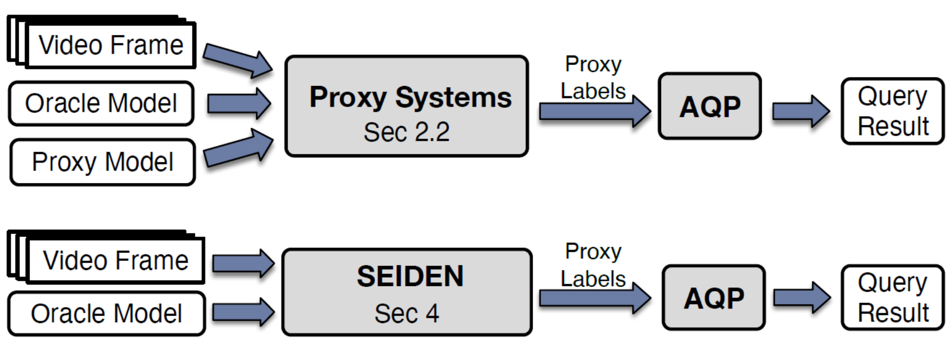

Figure 3.1: Proxy System

3.1 Query Processing Overview

Our objective is, we need to decrease the time consumed for processing the query. The query processing time (t_{QP}) contains index construction time (t_{IC}) and query execution time (t_{QE}) using the constructed index.

$$min(t_{QP} = \frac{t_{IC}}{N} + t_{QE})$$

Seiden always constructs indexes one time for a given dataset and the model. So, the index construction time is for all queries. The query execution stage happens to check whether the result meets the accuracy constraint desired by the user using the approximate query processing (AQP) technique. The time consumed by the AQP technique depends on the difference between the proxy labels (S_p) and the oracle labels (S_o). Thus, t_{QP} is given by:

$$t_{QP} \propto \alpha \cdot |S_p - S_o|^2 + (1-\alpha) \cdot \frac{t_{IC}}{N}$$

3.2 Model Architecture

In this report, we outline the architecture of the Seiden model, which is engineered to optimize the processing of video data and queries for efficiency. Seiden utilizes a distinctive method for both Index Construction and Query Execution.

Index Construction Phase: Unlike traditional video database management systems (VDBMSs) that employ proxy models, Seiden directly implements an oracle model, specifically Yolov5s, on a selected subset of video frames. This approach allows us to generate an index that directly incorporates the labels produced by the Oracle model, offering a more accurate baseline for further processing.

Query Execution Phase: In this stage, Seiden identifies additional frames that may hold important information relevant to the query and the video's content. It uses MAB Sampling to pinpoint frames that could enhance the data used in query responses. The Oracle model is then applied to these selected frames to further enrich the available information.

Following the completion of both phases, Seiden gathers and integrates the labels from all frames analyzed during the Index Construction and Query Execution phases. This aggregated labeling is then utilized to generate proxy labels for the remaining frames in the video that were not directly analyzed.

These proxy labels are essential for the operation of advanced Approximate Query Processing (AQP) algorithms. The primary role of these algorithms is to ensure the query results adhere to the predefined accuracy criteria. Depending on the nature of the query, different algorithms may be employed to ensure the response is specifically tailored to meet the unique demands of the query.

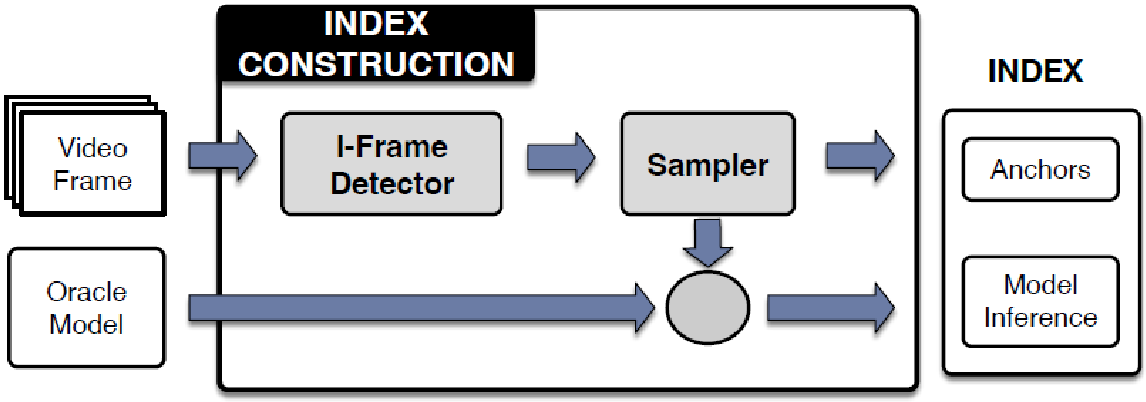

Figure 3.2: Index construction

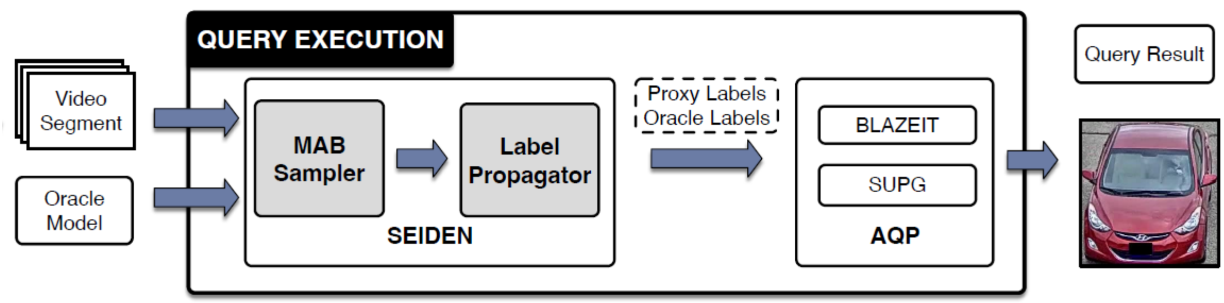

Figure 3.3: Query Execution

3.2.1 Sampling Ratio

The Index Construction step does not depend on specific queries. Seiden chooses a subset of I-frames from the video randomly, they are called anchor frames and organize the video into different segments based on the generated anchors. During the Query Execution phase, Seiden employs adaptive sampling to pick additional frames, focusing on segments that are most likely to contain relative information for the given query. Seiden operates with a predefined sampling budget, N, and uses a parameter, the anchor sampling ratio α, to allocate this budget across the 2 phases. Exactly, αN anchor frames are selected during the Index Construction phase, and the remaining $(1-\alpha)N$ samples are chosen through adaptive sampling during the Query Execution phase. This method helps Seiden maintain a balance between broad coverage and targeted focus.

3.2.2 Index Construction

Seiden initiates the process by taking the video X_s, a sampling budget N, the Oracle model M, and the anchor sampling ratio α as inputs. The anchor sampling ratio, α, indicates the portion of the total sampling budget αN allocated for the Index Construction phase. Initially, Seiden use random sampling to select αN frames from the video's I-frames. If αN is more than the available amount of I-frames in the video, Seiden takes all the I-frames and carries them over the unused portion of the budget to the Query Execution phase. Following this, the Oracle model M will run on all the chosen I-frames to generate inference results.

Algorithm 1: Index Construction

Input : Video X_s, Sampling budget N, Oracle model M,
Anchor sampling ratio α

Output: Anchor indices I_A, Anchor labels L_A

Function INDEXCONSTRUCTION (X_s, N, M, α):
    I_A = getIFrames(X_s) ▹ Get I-Frames from Video
    if $\alpha N < len(I_A)$ **then**
        I_A = randomSample(I_A, αN) ▹ Sample subset of I-Frames
    else
        $I_A = I_A$ ▹ Select all I-Frames
    end
    $L_A = \mathrm{M}(X_s, I_A)$ ▹ Get oracle labels of all selected anchors
    return I_A, L_A

Figure 3.4: Index construction algorithm

3.2.3 Query Execution

The Query Execution in Seiden comprises two primary stages:
(1) MAB Sampling
(2) Label Propagation

MAB Sampling: During this step, Seiden responds to the specific requirements of the query, whether it's a retrieval predicate or an aggregate expression. It adaptively selects additional frames from segments likely to yield relevant information, aiming for a more accurate query response. This selection process is treated as a multi-armed bandit problem.

Multi-Armed Bandit Background: The multi-armed bandit problem is an issue of optimal resource allocation among competing options, where each option offers a variable reward. A player faces multiple arms (options), each providing a stochastic reward, r_k. The objective is to maximize the total rewards received over a series of choices, illustrating the classic dilemma between exploring new arms and exploiting known ones.

The Upper Confidence Bound (UCB) algorithm addresses this by calculating the $UCB_{k,t}$ for each arm k at each step t:

$$UCB_{k,t} = r_{k,t-1} + c\sqrt{\frac{2lnN}{N_{k,t-1}}}$$

Here, $r_{k,t-1}$ represents the expected reward from arm k until step t1, $N_{k,t-1}$ is the count of selections from arm k up to t1, and N is the total number of selections across all arms. Parameter c helps balance exploration and exploitation, with higher values favouring exploration.

Problem Formulation: Seiden compares the frame-choosing challenge to a multi-armed bandit scenario.

Arms: After the Index Construction, Seiden divides the video into segments based on the anchor frames. Each segment is viewed as an arm.

Rewards: Rewards in this context depend on the type of query, with each video segment's reward being determined by its relevance and information content relative to the query. Seiden employs the UCB algorithm to decide which segment to sample next, aiming to optimally balance the exploration of less sampled segments against the exploitation of more rewarding segments.

$X_{k,t-1}$ represents the i^{th} sample taken fro arm k. For retrieval queries, which involve constraints like precision or recall, this random variable $X_{k,t-1}$ is set to 1 if the sample satisfies the predicate else 0. For aggregate queries, $X_{k,t-1}$ is equal to $f(x_{k,t-1})$ where f is the aggregation function that is specified in the query and the value is a real number (R).

After t1 samples from video segment k, our model calculates its reward based on the empirical standard deviation $\sigma_{k,t-1}$ which is calculated for the random variable $X_{k,t-1}$. This measure of variability provides a basis for the reward structure in the multi-armed bandit framework, reflecting the consistency or variability of the samples from each segment.

$$\hat{r}_{k,t-1} = \hat{\sigma} X_{k,t-1} \quad \text{where} \tag{4a}$$

$$\textbf{Retrieval queries:} \quad X_{k,t-1} = \begin{cases} 1, & \text{if sample satisfies predicate} \\ 0, & \text{otherwise} \end{cases} \tag{4b}$$

$$\textbf{Aggregate queries:} \quad X_{k,t-1} = f(x_{k,t-1}) \tag{4c}$$

Figure 3.5: Reward

3.2.4 Label Propagation

Label Propagation Step: Following the MAB Sampling process, Seiden utilizes a label propagation technique to extend labels from frames that are sampled to the rest of the frames in the video. To accurately assign these labels, Seiden employs a distance metric, d, which identifies the closest sampled frames to any given non-sampled frame q. The chosen metric is the temporal distance between frames. This metric is selected due to two main advantages: it leverages the inherent sequential nature of video data and is cost-effective to compute.

Example: Let us take an example where two sampled frames are identified by their frame IDs 10 and 20, which yield aggregate outputs for 5 and 10 from the Oracle model, respectively. For a non-sampled frame with a frame ID of 14, the aggregate result is linearly interpolated to 7. This method of interpolation is similarly applied to retrieval queries; for example, if the results for the sampled frames were 0.3 and 0.4, then the interpolated result for a frame with ID 14 would be roughly 0.34.

After this process, each frame that is not sampled receives an approximate proxy label. For aggregate queries, this involves computing an estimated aggregate value for all non-sampled frames. For retrieval queries, particularly those involving precision or recall requirements, an estimated probability is determined. These proxy labels, in combination with the oracle labels from sampled frames, are subsequently refined through Approximate Query Processing (AQP) modules for enhanced query response accuracy.

Chapter 4

Experimental Results and Analysis

4.1 Datasets

We assessed our baselines using four distinct datasets: Cherry, UA-DETRAC, Dashcam, and Jackson.

Cherry: The Cherry dataset consists of a one-hour video captured at a Seattle intersection during the morning. It includes 100,000 frames, with 36% featuring cars and 2.5% featuring buses. This dataset has a resolution of 960x540.

UA-DETRAC: The UA-DETRAC dataset is made up of several short videos from traffic cameras, each lasting one minute. These videos were merged into a longer video for analysis. The scenes typically show heavy traffic, leading to a high object presence rate of 94%. The videos are all in 960x540 resolution.

Dashcam: This dataset includes footage from a dashcam over 52 minutes. Unlike stationary camera datasets, both the background and foreground of the Dashcam video are in motion. The dataset predominantly captures cars, appearing in 60% of the frames, and buses in 5.6% of the frames. The resolution here is also 960x540.

Jackson: Used primarily to test state-of-the-art video database management systems, the Jackson dataset comprises footage from a static camera over approximately 2 hours and 46 minutes, resulting in 300,000 frames at a resolution of 300x300. Unique to this dataset is its nighttime setting, with cars appearing in 4 % of the frames.

4.2 Evaluation metrics

The F1 score is the chosen metric for evaluating our model's performance. This score calculates the average overlap between the predicted and the actual ground truth answers at the token level. It serves as a balanced metric, being the harmonic mean of precision and recall. The model's performance is considered strong only if both precision and recall are high, making the F1 score a comprehensive measure that reflects the combined effectiveness of these two aspects.

$$F1score = \frac{2(Precision * Recall)}{(Precision + Recall)}$$

The F1 score ranges from 0 to 1, where 1 is the max value and 0 is the least.

4.3 Evaluation on proxy system and seiden

In this experiment, we evaluate Seiden's capacity to leverage previous oracle model results for subsequent queries, in a manner akin to tasti-pt. We start with two precision queries focused on "CAR" and "BUS." When addressing the second "BUS" query, Seiden utilizes all the oracle labels previously collected from the first "CAR" query during both the Index Construction and Query Execution phases. Instead of initiating a new Index Construction phase, Seiden directly samples additional frames relevant to the "BUS" predicate, allocating the entire frame budget to the Query Execution phase. This approach highlights a significant accuracy enhancement, as the reuse of oracle labels from the first query informs and refines the frame sampling for the second query.

Table 4.1: Some examples for F1-Scores of tasti-pt and Seiden with reuse

Dataset Query	**Cherry Q2**	**Cherry Q2**	**Dashcam Q2**	**Dashcam Q2**
Object	**Car**	**Bus**	**Car**	**Bus**
TASTI-PT	0.92	0.63	0.91	0.48
SEIDEN	0.95	0.68	0.93	0.62

Chapter 5

Future Scope and Conclusion

5.1 Conclusion

Seiden is an advanced system designed to handle and analyze large amounts of video data efficiently and accurately. It uses special models and smart technology to quickly answer questions about video content, making it very useful for tasks like monitoring traffic or ensuring security. As technology improves, Seiden could get even better at analyzing videos and become easier for everyone to use. This system is particularly important as more data is captured in video form, helping us to make better decisions and understand our surroundings more effectively. Seiden is proving to be an essential tool for working with video data in today's digital world.

5.2 Future Scope

In the future, Seiden has a lot of potential to get even better. We can make it smarter by adding more advanced artificial intelligence, which would help it understand videos more deeply. Another cool idea is using edge computing, which means processing video data right where it's collected, making everything faster and saving data space. We might also teach Seiden to learn and adapt on its own to new situations, which would make it even more useful. Plus, making Seiden easier for everyone to use could help more people in different fields, like schools or shops, to analyze videos without needing a lot of technical knowledge.

5.3 Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our gratitude for all of them. We owe our sincere gratitude to our project guide Dr. Nagesh Bhattu S, Department of Computer Science and Engineering, National Institute of Technology Andhra Pradesh, who took keen interest and guided us all along, till the completion of our project work by providing all the necessary information. We avail ourselves of this proud privilege to express our gratitude to all the faculty of the department of Computer Science and Engineering at NIT Andhra Pradesh for emphasizing and providing us with all the necessary facilities throughout the work. We offer our sincere thanks to all our friends, fellow mates and other persons who knowingly or unknowingly helped us to complete this project.

Bibliography

[1] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica, *BlinkDB: queries with bounded errors and bounded response times on very large data*, *Proceedings of the 8th ACM European Conference on Computer Systems*, Pages 29-42, Year 2013.

[2] N. S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression*, *The American Statistician*, Volume 46, no. 3, Pages 175-185, Year 1992.

[3] M. R. Anderson, M. Cafarella, G. Ros, and T. F. Wenisch, *Physical representation-based predicate optimization for a visual analytics database*, *ICDE*, Publisher IEEE, Pages 1466-1477, Year 2019.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, *Neural codes for image retrieval*, *European conference on computer vision*, Publisher Springer, Pages 584-599, Year 2014.

[5] V. Braverman and R. Ostrovsky, *Generalizing the layering method of indyk and woodruff: Recursive sketches for frequency-based vectors on streams*, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Publisher Springer, Pages 58-70, Year 2013.

[6] S. Bubeck, N. Cesa-Bianchi, et al., *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*, *Foundations and Trends® in Machine Learning*, Volume 5, no. 1, Pages 1-122, Year 2012.

[7] M. D. Buhmann, *Radial basis functions*, *Acta numerica*, Volume 9, Pages 1-38, Year 2000.

[8] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, *API design for machine learning software: experiences from the scikit-learn project*, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, Pages 108-122, Year 2013.

[9] J. Cao, K. Sarkar, R. Hadidi, J. Arulraj, and H. Kim, *FiGO: Fine-Grained Query Optimization in Video Analytics*, *Proceedings of the 2022 International Conference on Management of Data*, Pages 559-572, Year 2022.

[10] P. Chunduri, J. Bang, Y. Lu, and J. Arulraj, *Zeus: Efficiently Localizing Actions in Videos Using Reinforcement Learning*, *SIGMOD*, Pages 545-558, Year 2022.

[11] M. Daum, B. Haynes, D. He, A. Mazumdar, M. Balazinska, and A. Cheung, *TASM: A Tile-Based Storage Manager for Video Analytics*, arXiv preprint arXiv:2006.02958, Year 2020.

[12] J. Deng, A. C. Berg, and L. Fei-Fei, *Hierarchical semantic indexing for large scale image retrieval*, *CVPR 2011*, Publisher IEEE, Pages 785-792, Year 2011.

[13] S. Garg, S. Mitra, T. Yu, Y. Gadhia, and A. Kashettiwar, *Reinforced Approximate Exploratory Data Analysis*, *Proceedings of the AAAI Conference on Artificial Intelligence*, Year 2023.

[14] T. F. Gonzalez, *Clustering to minimize the maximum intercluster distance*, *Theoretical computer science*, Volume 38, Pages 293-306, Year 1985.

[15] S. Guha and B. Harb, *Wavelet synopsis for data streams: minimizing non-euclidean error*, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, Pages 88-97, Year 2005.

[16] B. Haynes, A. Mazumdar, M. Balazinska, L. Ceze, and A. Cheung, *Lightdb: A dbms for virtual reality video*, *VLDB*, Volume 11, no. 10, Year 2018.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask R-CNN*, *Proceedings of the IEEE international conference on computer vision*, Pages 2961-2969, Year 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Pages 770-778, Year 2016.

[19] W. He, M. R. Anderson, M. Strome, and M. Cafarella, *A method for optimizing opaque filter queries*, *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, Pages 1257-1272, Year 2020.

[20] J. M. Hellerstein, P. J. Haas, and H. J. Wang, *Online aggregation*, *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, Pages 171-182, Year 1997.

[21] K. Hsieh, G. Ananthanarayanan, P. Bodik, S. Venkataraman, P. Bahl, M. Philipose, P. B. Gibbons, and O. Mutlu, *Focus: Querying large video datasets with low latency and low cost*, *USENIX*, Year 2018.