Vision AI: Improving Stadium and Venue Operations Efficiency with Artificial Intelligence and Computer Vision

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kaihil Patel

Fall, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Rosanne Vrugtman, Department of Computer Science

Vision AI: Improving Stadium and Venue Operations Efficiency with Artificial Intelligence and Computer Vision

CS4991 Capstone Report, 2023

Kaihil Patel
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
knp6zvz@virginia.edu

ABSTRACT

Stadiums and venues across the world experience operating inefficiency issues at each event they hold, causing revenue loss and unfavorable attendee experiences. The use of Artificial Intelligence in the vision field is a modern and effective solution to these issues. I built a prototype using the new, state-of-the-art computer vision model developed by Ultralytics, YOLOV8. By integrating YOLOV8 with Roboflow's Supervision in Python, I designed a model to track guests' movements across a venue using live data. Although major findings are still being discovered from the model, I expect to identify the primary movement patterns of people along with corresponding peak times and events that could influence both. Understanding the demand and flow of people is critical in providing the optimal resource supply and solutions to increase revenue and improve attendee experience. Advanced models are needed to target specific issues within the venue. In addition, collecting more data enables the models to learn and generalize from a wider range of visual patterns and variations, leading to improved accuracy and performance overall.

1. INTRODUCTION

As an avid sports fan, I love going to stadiums to catch a game in my free time. During the summer of 2023, I had the opportunity to attend a few games and analyze the fan experience. Here is a brief walkthrough of that journey.

As for most fans, my experience at the ballpark was fun, exciting, and memorable. Although the overall experience was enjoyable, there were some glaring issues ranging from disorienting gate entry lines and seat chart confusion to confusing concessions procedures and uncleanly concourses. As the crowd eagerly gathers, the subtle undercurrent of operational inefficiencies at the stadium becomes palpable, hinting at a need for streamlined processes to enhance the overall fan experience. Issues such as those described, and many others, negatively impact not only the event attendees but the venue hosts, as well.

Modern problems require modern solutions. Rather than appointing humans to direct traffic or constantly look for operational inefficiencies, a data-driven, computer modeling system should be implemented. Computer Vision and Vision Artificial Intelligence are the driving technologies behind this solution.

Although future steps would include analyses and actionable outcomes, my internship scope focused on the minimum viable product (MVP) for this technology by creating the classification and tracking models using the YOLO V8 (by Ultralytics) and Roboflow Python packages.

2. RELATED WORKS

Many experts in urban planning and related fields have suggested the use of live computer vision to optimize traffic flow in cities. Research conducted by Umair, et al (2021) recommends the use of object detection and tracking techniques to estimate vehicle queue length in urban traffic scenarios. This eliminates the need for additional sensors. The method achieved an average accuracy of 93% for queue length estimation, demonstrating its efficiency and robustness. My adaptation would provide an estimate for wait times but still avoid the drawbacks of not accounting for other transportation methods as foot traffic is the only action occurring.

Another example would be how Talaat et al (2023) discusses the application of neural networks detecting state change to detect visual signs of wildfires. Early detection of wildfires can have immense advantages by tackling the problem before it causes harm to the surrounding environment. This approach aims to enhance accuracy, reduce false alarms, and be cost-effective compared to traditional methods. However, my system is less computationally intensive, requires fewer sensors and cameras, and is on a smaller scale which means false positives are not as harmful and the system is easier to maintain.

3. PROJECT DESIGN

My project relied on a basic architecture. Because it was a proof-of-concept project to present to the client, my group was not worried about connecting the inputs and results to a front-end page. The theoretical final product would include connections to the client's live front-end site for their internal use.

3.1 Client Needs and Features

When the project scope was introduced to my team, it included an outline of the problem, the client, and the basic needs. The client was looking for a way to improve the attendee experience ratings at their venue using modern technology solutions. My team was provided with data collected from various events at the venue in the form of survey responses that asked questions relating to the experience from entry to exit. My group analyzed the data in charts, graphs, and manual read-throughs to identify the common issues attendees were enduring. After research and discussion, entry gate lines, concession lines, and cleanliness were identified as the primary focus of what we would work on for the project.

3.2 System Requirements and Overview

After the project scope was finalized, system requirements were outlined in collaboration with the client and company. Due to the project being in its early phases and the idea being a prototype, we had the freedom to interpret the requirements as we viewed reasonable. The model needed to take in a video input and use it to produce an output. The program had to be easy to use and the outputs had to be simple enough to read and understand. It was also important to create the outputs in a manner that allowed insights to be easily extracted from them. This led us to the belief that the results should be in graphical format in addition to a video output. Speed was also a limitation that we had to consider along the way, although it was not the highest priority.

3.3 Machine Learning Model Selection

Python was regarded as the standard programming language for this project as it was a familiar language to all group members, in addition to it being an industry standard. However, research needed to be done on how to identify and track people in a video.

The YOLO Python package by Ultralytics was introduced to us as a common computer vision model. YOLO stands for You Only Look Once, which means that the model only needs to process an image once to find all the objects in it. We decided to use the most recent version, YOLO v8, in integration with another package, Roboflow's Supervision, which was recommended by the company's computer vision experts.

According to Roboflow's official blog (Solawetz, 2023), Supervision combined with YOLO v8 performs the three primary tasks of object detection, image classification, and instance segmentation with high speed and accuracy, making it one of the best computer vision models available today. We opted to use pre-trained models since we were detecting humans and YOLO v8 is already highly trained in object classification of common objects, like humans. We used Roboflow as a supplement to count and track objects across the screen once YOLO detected and classified them with accuracy ratings. The program was built user-friendly in that the only changes that needed to be made for someone else to run it was a single line of code specifying the input video filename.

3.4 Building the Vision Models

The following subsections will delve into details about how we gathered data and what results came directly from it. This includes the data format and visuals from the inputs and outputs.

3.4.1 Inputs and Testing

The input the model needed to receive was a video, which we chose to be in a standard mp4 format. As we were creating the software, we tested it iteratively on video captured from our smartphones. Toward the final weeks of the internship, we captured data on smartphones from a stadium in Washington, D.C., to further test the models.

3.4.2 Outputs

There were three primary forms of output that we built our software to produce. The first was the output video itself. The display was divided into a grid of zones, which could be customized by the number of columns, number of rows, or custom zones. The zones counted the number of people currently in that zone and displayed that value. If the count became larger than a customizable threshold, the zone would flash red and send an email notification to the user. This was the second of the outputs. It was important for the user to be automatically and visually alerted if an area in the venue became overcrowded for them to inspect. The third model output was graphical. As the models were running, an additional data point being collected showed how long that object stayed in the frame. This data, along with the zone numbers and counts, was recorded in a CSV file. After the models finished running on the input videos, the CSV files were used to generate two plots: a bar graph of each unique object and how long they remained in the frame, and a multiple-line graph plotting the counts of people per zone over time.

3.5 Challenges and Solutions

Although the overall final product was successful, there were some road bumps along the way. One of the first issues was that the program ran slower than intended. This is because YOLO v8 and Supervision break the input video down into individual frames, and the algorithms are run on each frame individually. There was an exponentially increasing time function for the amount of time it took to process a video based on its length. To solve this, the group made the collective decision that each frame did not need to be analyzed. Since people physically move at a rate slower than frames can be captured, it was a valid solution to skip frames and only analyze every tenth frame.

Another problem we encountered was that the final graph outputs became messier and less interpretable as the input videos became longer and there were more zones to keep track of. In addition, it was hard for the program to build live manipulating plots as the video was being processed. The latter of the issues was due to the system we were running the program on and could not be solved locally, so the best solution was to generate the plots after the CSV files finished recording. The plot readability was improved by providing a facet option by zone and displaying a horizontal threshold line to better understand the peaks.

4. RESULTS

As the vision model improves over time, we hope to gain specific insights about what causes concession line backups and how people can effectively travel throughout a venue. However, due to the timeline of my internship, I was only able to run a fraction of the initial data collected on the models.

From the data collected from the center-field commons area, the indisputable result was that crowds tended to move toward the center-field lookout rather than to their seats or concessions. This could be to take pictures, for the nice view, or due to the snack and drink vendors placed there.

Actionable results were not produced by running the models on video data from the concession lines. This is due to a lack of data, unusual occurrences during data collection (i.e., rain delays), and an absence of additional information from the venue that we could not account for given our project scope (i.e., concession inventory and workflows).

5. CONCLUSION

Vision Artificial Intelligence models have vast potential in a stadium setting for increasing stadium revenue and improving the fan experience. Using modern techniques and software packages, this is possible, and a proof of concept was built by me and my internship team.

With the implementation of these advanced models, inefficiencies in the gate-entry and concessions lines can be pinpointed, and live alerts for cleanups and resource allocation can be employed. The vision models provide ways for inefficiencies to be analyzed for flaws and then optimized in real time. The technological possibilities to transform the modern stadium into the Stadium of the Future are endless.

6. FUTURE WORK

Due to the project being a proof of concept showcase to display the potential of computer vision in a stadium setting, possible future work is abundant. More robust models should be built to refine the data being extracted and stored so that a concrete output can be formed. In addition, a complete system would need to be created around this technology to make it an operational product. Some of those features include connectivity to the in-house databases and communication systems as well as providing clear, actionable items from the outputs.

Finally, as with all Machine Learning models, more data is the best way to improve them and bring more accurate results over time. More data enhances the model's ability to generalize and accurately recognize a wider range of objects and scenarios in real-world applications, reducing the risk of overfitting to specific training examples. Additionally, it allows the model to handle rare or edge cases more effectively.

REFERENCES

Solawetz, J. (11 Jan. 2023). *What Is YOLOv8? The Ultimate Guide*. Roboflow

Blog, https://blog.roboflow.com/whats-new-in-yolov8/.

Talaat, F. M., ZainEldin, H., (2023*). An improved fire detection approach based on YOLO-v8 for smart cities*. Neural Computing and Applications. 35(28), https://doi.org/10.1007/s00521-023-08809-1

Umair, M., Farooq, M. U., Raza, R. H., Chen, Q., Abdulhai, B. (2021). *Efficient Video-based Vehicle Queue Length Estimation using Computer Vision and Deep Learning for an Urban Traffic Scenario*. Processes, 9(10), 1786. https://doi.org/10.3390/pr9101786