Tackling the Skies with UAV Swarm Detection

Sound Antidote

Capstone Project Report

Northeastern University

Team Members:

Abdur Rahman Danish
Daniel Varghese
Divi Joshi
Ishan Lokhande
Shivani Avasarala

Date of Submission: Dec. 14th, 2023

Executive Summary:

The aerospace defense and agriculture industry come with their own unique challenges that call for applications in object detection, capable of real-time processing and accuracy. Sound Antidote, a Toronto based start-up, is leading the development of algorithms, using the latest advances in computer vision to help with these challenges. In this paper, we explore the latest developments in object detection and distance estimation, namely YOLOv5, YOLOv8 and YOLO-NAS.

YOLO-NAS stands out from the others due to its quantization-friendly architecture, making it conducive to environments with limited computational resources. It boasts high accuracy, especially in detecting small objects, and demonstrates lower latency, which is crucial for real-time applications. One aspect that makes YOLO-NAS unique is post-training optimization, which helps select the best model size to improve speed while minimizing loss in performance.

YOLOv8, characterized by its transformer-based architecture, does not support post-training quantization, which can be a constraint when computational power is limited. However, it benefits from pseudo-labeling and knowledge distillation during its training process.

The deployment of these models is not without challenges. Training these models on image datasets is resource-intensive, requiring not only significant computational power but also operational and technical expertise. Factors such as varying lighting conditions and cluttered backgrounds further compound the difficulty of achieving high accuracy in object detection. Our hands-on approach, supplemented by insights from the latest academic research, has led to valuable learning experiences in training YOLO models with Sound Antidote's image datasets.

Table of Contents

1. Introduction

The evolution of Unmanned Aerial Vehicles (UAVs), or drones, has been remarkable, marked by rapid technological advancements and an increasing presence on our skies for a variety of applications.

- **Early Development and Military Use**: The origins of UAVs can be traced back to the early 20th century, primarily for military purposes. From early models used for target practice and reconnaissance, technological progress has led to sophisticated UAVs with remote capabilities, surveillance, and even armed functions.
- **Commercial and Civilian Use**: Moving beyond military usage, UAVs have found applications in agriculture, environmental monitoring, disaster management, delivery services, and aerial photography. Their affordability and accessibility have fueled widespread adoption across industries.

As the number of UAVs grows, contemporary projects focus on enhancing detection systems. These technologies aim to monitor UAVs, especially in sensitive areas, identifying unauthorized or potentially hazardous activities. The integration of AI and machine learning enhances UAV detection capabilities.

So, all in all, the increasing presence of UAVs in our skies brings challenges in detection and regulation, which are critical areas of focus for ongoing and future projects. We have moved ahead with a variety of methodologies to assess and evaluate each one against one another in varying environments and scenarios.

2. Problem Statement

Sound Antidote reached out to Northeastern to seek guidance from graduate students to optimize their current application to detect UAV swarms. Due to the complexity of the application, and technical challenges, they needed to understand trade-offs between modern object detection models and their limitations.

By training and testing image datasets provided by Sound Antidote, along with research into recent publications on computer vision, and object detection, our team was able make recommendations. This paper sets out to navigate the complexities between these models, namely YOLOv5, YOLOv5 and YOLO-NAS; along with factors that affect accuracy, performance and adaptability.

3. Literature Review

3.1. Selecting the Right Library for Drone Detection

- **Accuracy and Efficiency**: Prioritize a library with high accuracy in drone detection across diverse environments. The speed and precision of object

detection libraries significantly impact real-time applications. Libraries like Faster R-CNN and SSD exhibit varying effectiveness, demanding thorough evaluation for drone detection tasks.

- **Library Evaluation**: Assess object detection libraries, including YOLO, Faster R-CNN, and SSD, focusing on their performance in drone detection. We opted for YOLO's variations in implementing Sound Antidote due to their compatibility with existing frameworks, particularly with past versions like YOLOv5 and YOLOv8. Our primary focus is on comparing the applications of YOLO-NAS and YOLOv8 within the context of Sound Antidote.

3.2. Evaluating Computational Load

- **Resource Utilization**: Evaluate the computational demands of each library, considering factors such as processing speed, memory usage, and power consumption. Huang et al. (2017) highlight the trade-offs between accuracy and speed, underscoring the need for resource optimization in practical applications.
- **Hardware Compatibility**: Ensure the selected library is compatible with available hardware, operating efficiently without causing significant delays or excessive processing power. As suggested by Garcia-Garcia et al. (2017), the library's compatibility with existing hardware is pivotal for effective and efficient deployment, especially in real-time scenarios.

3.3. Tech Specs for Long-Range Detection

- **High-Resolution Cameras**: Utilize high-resolution cameras that can capture detailed images over vast distances. This is crucial for detecting small objects like drones. We emphasize the importance of camera resolution in enhancing the detection capabilities of vision systems
- **Advanced Sensors**: Suggest advanced sensors, such as infrared or thermal sensors, to enhance detection capabilities, especially under challenging conditions like low light or obscured visibility. Gonzalez et al. (2016) highlight how different sensor types can significantly improve detection capabilities in various environmental conditions
- **Signal Processing Techniques**: Incorporate sophisticated signal processing techniques to analyze the data from these sensors effectively, enabling the detection of drones at greater distances as noted by, once again, Gonzalez et al. (2016)

3.4. Applications and Integration

- **Security and Surveillance**: Integrate the chosen library into security systems for monitoring sensitive areas, such as airports, military bases, or critical

infrastructure. This requires careful consideration of the technology's capabilities and limitations in the context of using AI in surveillance

- **Traffic Management**: Use in airspace management, especially in urban areas where drone activity is expected to increase. Here, the adaptability and scalability of the chosen technology are critical.
- **Environmental Monitoring**: Implementing computer vision in environmental conservation, particularly for wildlife monitoring and detecting illegal activities, is crucial. The accuracy and reliability of detection systems are paramount, as emphasized by Anderson et al. (2013) in their exploration of technology applications in wildlife monitoring.
- **Disaster Response**: For disaster response scenarios, the ability of the system to provide quick and accurate assessments is crucial, as highlighted by Meier (2015) in his exploration of drones in humanitarian settings

3.5. Juxtaposing YOLOv8 and YOLO-NAS[1]

- YOLOv8 represents a revolutionary leap in object detection. With its transformer-based architecture, YOLOv8 introduces knowledge distillation and pseudo-labeling, creating a model with unprecedented accuracy and performance.
- Enter YOLO-NAS, a trailblazer in object detection, inheriting its legacy from YOLOv6 and YOLOv8. What sets YOLO-NAS apart is its introduction of a quantization-friendly basic block, a masterstroke for object detection models. This innovation, combined with advanced training schemes, AutoNac optimization, and pre-training on top datasets, propels YOLO-NAS to new heights.

4. Methodology

The project is multifaceted, focusing on three interconnected topics: Object detection, distance and velocity calculation.

4.1. Data collection and understanding the data

We were provided 400 images and 400 annotations files of three types of drones for object detection, involving the testing of object detection models.
The three types of drones[2] were:

- DJI Mavic
- E-Flite Opterra
- DJI Phantom

[1] Refer to Appendix A for a trade-off table between YOLOv8 and YOLO-NAS.
[2] Refer to Appendix B to see images of the types of drones used.

Each.txt file of annotations contains an array of 4 defining metrics[3] i.e. [class, X-coordinate, Y-coordinate, width, height]

4.2. Analysis Techniques

- Object detection involved the implementation and evaluation of YOLOV8 and YOLO-NAS algorithms on the collected image datasets.
- Distance estimation analysis included a comprehensive review of different methods, focusing on both monocular and stereo vision while considering strengths, limitations, and practical applications.
- Investigations into velocity calculation methods, specifically centroid tracking, involved a review of existing literature and relevant studies.

4.3. Tools

- Google Colab provided a collaborative environment for executing and analyzing object detection algorithms.
- YOLOv8 and YOLO-NAS served as the primary models for object detection, while literature review and research were fundamental tools for understanding distance estimation and velocity calculation methods.

5. Analysis and Findings

5.1. Object detection working

- The image is divided into a grid. Each cell in this grid takes on the responsibility of trying to find objects within it.
- Within every cell, the system tries to predict if there are objects present. It does this by looking for features and patterns within that cell.
- These cells can predict multiple bounding boxes and give a likelihood score for each class of object they think is present.
- The bounding boxes that have a high confidence score and match the ground truth closely are the ones that are chosen as the final predictions.

5.2. The YOLO algorithms

Traditional object detection methods work in two parts. First, they identify regions of interest in an image and then classify those regions. YOLO, on the other hand, does this in one single step.

[3] Refer to Appendix C to see the content of the annotations file of images used.

5.2.1. YOLOv8:

The YOLOv8 architecture comprises two main components: the modified CSPDarknet53 backbone and the head, featuring multiple convolutional and fully connected layers for predicting bounding boxes, objectness scores, and class probabilities. Notably, YOLOv8 incorporates a self-attention mechanism in the head, enabling the model to focus on different image parts based on relevance. It supports diverse backbones, such as EfficientNet, ResNet, and CSPDarknet, offering users flexibility. Designed for speed and efficiency, YOLOv8 provides advanced data augmentation techniques and pre-trained models for easy use and transfer learning on various datasets

How to work with YOLOv8?

- Install required packages
- Define paths for input and output
- Perform train/val/test split in a ratio of 3:1:1
- Train YOLOv8 on Custom data
- Select best model after training
- Predict on test data
- Plot the results

We got the following results[4]:

- Phantom - Precision: 71%, Recall: 69%
- Mavic Mini - Precision: 69%, Recall: 67%
- E-flite - Precision: 75%, Recall: 77%

5.2.2. YOLO-NAS:

NAS (Network Architecture Search) is a subfield within AutoML. The recent automation introduced in YOLO-NAS represents a significant breakthrough, allowing researchers and developers to shift their attention from intricate architecture design to more impactful pursuits such as exploring innovative applications and addressing broader challenges.

YOLO-NAS is a new object detection model that comes in three different sizes: S, M, and L. The size of the model determines its accuracy and efficiency[5].

How to work with YOLO-NAS?

- Start by installing pip packages. The YOLO-NAS model itself is distributed using super-gradients package.
- Define input and output paths.

[4] Refer to Appendix E for YOLOv8 results on mavic-mini dataset

[5] Refer to Appendix D for the Mean Average Precision and Latency of the different versions of NAS on a pre-trained model within NAS.

- Split images and their respective annotations to train, validation, and test subsets in a ratio of 3:1:1.
- Set Global parameters for epochs, batch size and workers.
- Train the NAS model (small, medium, or large) on the train dataset.
- Hyper-parameter tuning and selection of the best model
- Apply the selected model to predict on the test dataset, assessing its generalization performance on unseen data.
- Visualize and interpret the results through plotting.

We got the following results[6]:

- YOLO-NAS Medium on Phantom dataset-
  - Precision: 39.93%
  - Recall: 75.25%
  - MaP: 72.62%
  - F1: 51.34%
- YOLO-NAS Large on Phantom dataset-
  - Precision: 19.19%
  - Recall: 76.75%
  - MaP: 74.94%
  - F1: 30.70%

5.3. Monocular Vision

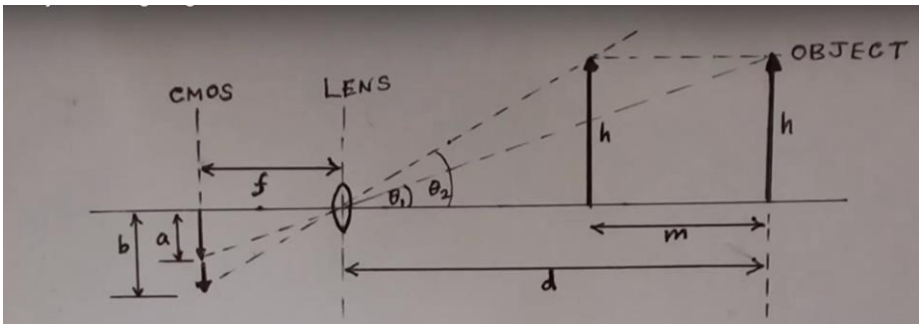
In recent years, multiple approaches have emerged to ascertain an object's size or the distance between the camera and the object. The prevalence of cameras in everyday life has significantly increased, especially with the widespread availability of these devices in modern mobile phones.

Determining the distance between a camera and a specific object relies on the fundamental principles governing camera operation. If the actual size of the object is known, calculating the distance becomes a straightforward process. This simplicity stems from a clear understanding of the mechanics underlying camera functionality.

Monocular vision refers to the capability to gauge depth and distance using a single eye or camera. This method depends on a range of cues, including perspective, shading, texture, motion, and occlusion, to deduce the three-dimensional layout of the environment.

5.3.1. Explanation of how distance is calculated with Monocular vision

[6] Refer to Appendix F for a visual on how YOLO-NAS trains on Phantom dataset



- h: Height(radius) of the object
- m: Distance object is moved.
- f: Focal length
- a: Reflection height 1
- b: New reflection height 2
- **d: The unknown parameter distance**

The unknown parameter d, i.e., the distance between the camera and the object in its original position can be calculated by the following algorithm:

Formula

$$\frac{a}{f} = tan\theta_1 = \frac{h}{d} \quad \text{-----1}$$

$$\frac{b}{f} = tan\theta_2 = \frac{h}{d-m} \quad \text{-----2}$$

DIVIDE EQUATION 1 & 2

$$\frac{a}{b} = \frac{h}{d} \times \frac{(d-m)}{h}$$

$$\frac{a}{b} = \frac{(d-m)}{d} = 1 - \frac{m}{d}$$

$$\frac{m}{d} = 1 - \frac{a}{b}$$

5.3.2. Advantages of Monocular Vision:

- **Simplicity of Implementation and Deployment:**
  - Utilizes only one camera, reducing complexity and making deployment straightforward.
  - Requires fewer computational resources for efficient operation.
- **Robustness to Environmental Changes:**
  - More resilient to variations in lighting, background, and occlusion compared to stereo vision.
  - Not reliant on the consistency of stereo pairs, enhancing adaptability.
- **Compatibility with Existing Images and Videos:**
  - Works seamlessly with images and videos from various sources, such as social media posts, and surveillance footage.
  - Efficiently processes data typically captured with a single camera.

5.3.3. Drawbacks of Monocular Vision:

- **Reliability and Accuracy:**

  Less reliable and accurate compared to stereo vision, as it relies on assumptions and heuristics.

- **Sensitivity to Noise and Distortion**:

  Greater sensitivity to noise and distortion, impacting the quality and consistency of both the image and the feature vector.

5.4. Stereo vision

Stereo distance measurement in image processing is based on the principles of stereopsis, which is the ability of the brain to interpret the different perspectives provided by the two eyes to perceive depth. In image processing, this concept is applied using stereo vision, where a pair of images is captured from slightly different viewpoints, simulating the perspective of the left and right eyes.

5.4.1. Basic overview of how stereo distance measurement works

- **Stereo Camera Setup & calibration:**
  - Selecting suitable stereo cameras is crucial for accurate distance measurements. High-resolution cameras with synchronized image capture enhance precision. Computational power is equally vital, ensuring real-time processing for efficient distance calculations
  - Two cameras are set up to capture images of the same scene from slightly different positions, simulating the left and right eyes.

- The two cameras are set up horizontally[7] at the same level and vertically displaced by a predefined distance called a 'baseline'.
- The pictures need to be captured at the same time with both cameras.

- **Image Rectification:**

  The captured images are rectified to ensure that corresponding points in the left and right images lie along the same horizontal scan lines. This simplifies the subsequent processing. This is done to solve the correspondence problem.

- **Feature Matching:**
  - Feature matching[8] involves identifying corresponding local features, such as key points or corners, in both left & right images.
  - SIFT (Scale-Invariant Feature Transform)[9] and SURF (Speeded-Up Robust Features) algorithms are used for detecting and describing these local features.
  - Detected features are then matched between left and right images to establish correspondence.
- **Disparity Calculation[10]:**
  - Disparity refers to the apparent shift or difference in the position of an object between the left and right images.
  - The calculation of disparity involves finding the pixel-wise differences between the corresponding features in the left and right images.
  - This difference is used to estimate the depth or distance of objects in the scene.
  - The collection of disparity values for each pixel in the image forms a disparity map. The disparity map is calculated by finding the horizontal offset between corresponding features in the left and right images. This map provides a visual representation of the disparities across the stereo images
- **Post-Processing:**

  Various techniques, such as filtering and smoothing, may be applied to the depth map to improve accuracy and reduce noise.

- **Triangulation:**
  - Using the disparity information and the known baseline, the depth or distance to each feature in the scene can be calculated through triangulation.

[7] Refer to Appendix G on how the final camera apparatus looks like.
[8] Refer to Appendix H for a visual of Feature Matching working.
[9] Refer to Appendix I for a trade-off table between SIFT and SURF algorithms.
[10] Refer to Appendix J for a visual on Disparity calculation.

- The triangulation process involves using the disparity, baseline, and the focal length of the cameras to determine the distance of each point in the scene.

- **Distance Estimation:**

  Once the depth map is obtained, the actual distance from the cameras to the objects in the scene can be estimated based on the triangulation results.

5.4.2. Stereo matching

Stereo matching, akin to human vision's magic, fuses two perspectives to unlock 3D depth in 2D images. Using two slightly offset cameras, stereo matching analyzes horizontal shifts, or disparities, between corresponding points, encoding depth fingerprints. Creating a disparity map involves comparing these disparities across the image pair. This grayscale "depth map" assigns brightness levels to pixels, revealing closer or farther objects. Complex matching techniques, like block matching, meticulously compare small image patches, determining the hidden disparity. To handle real-world complexities, global optimization techniques ensure smooth transitions in the disparity map. The grand finale transforms the disparity map into a 3D point cloud, a digital replica teeming with in-depth information.

Stereo matching[11] continually evolves, pushing accuracy and efficiency boundaries. As research advances in algorithms and harnesses computing power, we anticipate more stunning 3D reconstructions and groundbreaking applications.[12]

5.4.3. Stereo matching using correlation

The paper (Zaarane et al., 2019) proposes a novel approach to vehicle detection using a single camera. Rather than employing time-consuming stereo matching algorithms on images from both cameras, the method detects vehicles in one camera's images and matches them with the corresponding positions in the other camera's images using cross-correlation. The best match is determined when the cross-correlation result surpasses a predefined threshold, indicating optimal correlation. This innovative technique allows for efficient vehicle detection without the need for extensive stereo processing and is particularly effective outside the overlapping field of views of both cameras.

5.4.4. Stereo vision algorithms

- **Local Stereo Vision Algorithm:**

  This algorithm computes local disparities using nearby pixels, employing methods like box filtering within specific pixel neighborhoods. While computationally efficient, it struggles with accurate determinations in texture-

[11] Refer to Appendix K for more information on types of stereo matching.
[12] Refer to Appendix L for a flowchart on Stereo Matching's working.

less or repetitive areas and has limitations in handling occlusions, especially in scenarios with occluded objects.

- **Global Stereo Vision Algorithm:**

  Global stereo vision algorithms consider nonlocal constraints to achieve better accuracy. Dynamic programming structures, like those used in global approaches, optimize disparity globally with a focus on overall image consistency. This method enhances accuracy by addressing challenges in occlusion and uniform texture complexity. However, it comes with higher computational complexity, making real-time processing potentially slower.

- **Semi-Global Stereo Vision Algorithm:**

  Striking a balance between local and global factors, semi-global stereo vision algorithms, such as Hirschmüller's semi-global algorithm, optimize disparity globally while incorporating local constraints. This results in moderate computational complexity, offering a compromise between accuracy and processing speed. Semi-global algorithms effectively handle challenges related to occlusion and sub-pixel accuracy, making them suitable for real-time processing while maintaining a good level of accuracy.

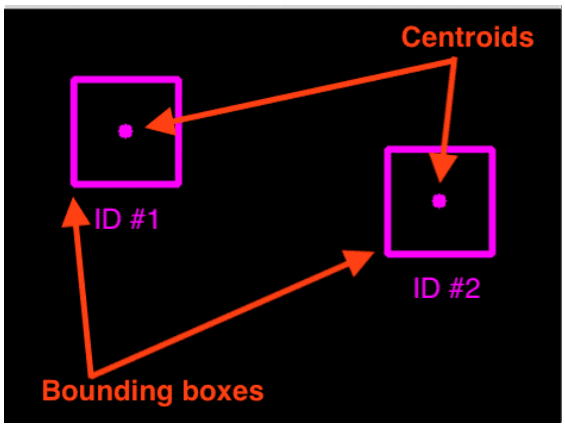
5.5. Velocity

Velocity calculation is a crucial aspect of object tracking, providing insights into the speed of moving objects within a video stream. One effective method for tracking objects is the centroid tracking algorithm. This algorithm involves several key steps, including the computation of centroids and the subsequent calculation of velocity based on the distances between centroids in consecutive frames.

5.5.1. How to calculate velocity using centroid tracking method:

Step 1: Centroid Tracking Algorithm Steps:

1: Accept bounding box coordinates and compute centroids:

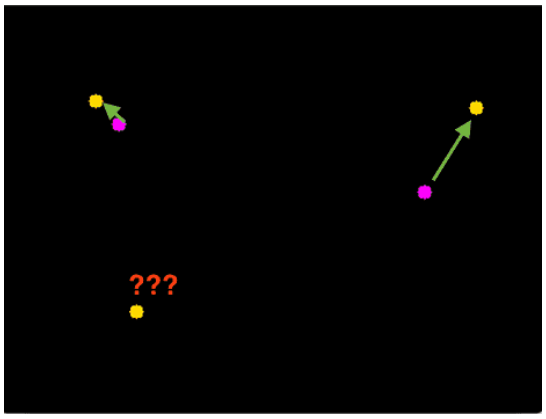


- Object bounding box coordinates are provided for each frame, generated by any object detection method.
- Centroids, representing the center (x, y)-coordinates of the bounding box, are computed for each detected object.
- Unique IDs are assigned to the initial set of bounding boxes.

2: Compute Euclidean distance between new and existing objects:

- In subsequent frames, object centroids are computed again.
- Euclidean distances between each new and existing object pair are calculated.
- These distances determine potential associations between the new and existing centroids.

3: Update (x, y)-coordinates of existing objects:



- The algorithm associates' centroids by minimizing Euclidean distances between subsequent frames.
- Objects with minimum distances are matched, updating their (x, y)-coordinates.
- This step forms the basis of the object tracker.

4: Register new objects:

- In cases where more objects are detected than currently being tracked, new objects are registered.
- Registration involves assigning a new object ID and storing the centroid of the bounding box coordinates for the new object.
- The process then returns to Step #2 for subsequent frames.

5: Deregister old objects:

- To handle object loss, disappearance, or leaving the field of view, old objects are deregistered.
- If an object cannot be matched to any existing objects for a specified number of subsequent frames (N), it is considered lost.
- The deregistration ensures the algorithm adapts to changing scenarios and updates the list of tracked objects accordingly.

Step 2: Velocity Calculation

Velocity is determined by multiplying the distance of centroids between the previous frame and the current frame by the frame rate of the video. Additionally, this value is scaled based on the unit of measurement, typically in meters per pixel.

$$Velocity = Distance\ of\ Centroids \times Frame\ Rate \times Scale$$

6. Discussion

Stereo vision, while innovative for UAV swarm detection, grapples with distance-related accuracy drops and sensitivity to environmental factors like inconsistent lighting and reflections. The crux of deploying such a system lies in the complex calibration of camera pairs, where precision is paramount, and errors can significantly undermine depth mapping.

The environment plays a pivotal role; sparse visual features can thwart the technology's ability to accurately gauge depth, crucial for velocity analysis. In hot climates, managing excess heat through cooling systems becomes necessary due to the computational load.

To mitigate stereo vision's limitations, it is often paired with other sensing technologies. Radar, with its moderate power needs, adeptly pinpoints position and movement. Lidar offers unmatched precision despite fluctuating power demands, while ultrasonic sensors provide energy-efficient but restricted detection.

Monocular vision presents its own challenges, mainly pertaining to depth perception, challenges in tracking multiple objects in dynamic environments, limited field of view, environmental sensitivity and texture reliance. Although there is some overlap in complexities compared with stereo vision, most issues relevant to monocular vision are due to limitations caused by field of view, which is reduced.

7. Recommendations

Stereo vision, a potent tool in computer vision, provides insights into the three-dimensional structure of objects. Its versatility spans applications from robotics to aerial surveys, understanding spatial arrangements comprehensively. Algorithmic approaches balance speed, accuracy, and complexity, adapting to real-time processing and diverse environmental conditions. Ongoing research refines stereo vision algorithms for enhanced performance. Sound Antidote's recommendations emphasize scenario-based algorithm selection, high camera frame rates, integrating complementary technologies for correspondence challenges, and investing in heat-dissipating technologies for optimal performance.

These strategic steps align with key takeaways, forming a comprehensive framework for Sound Antidote's stereo vision system advancement.

- **Stereo Vision Fundamentals:**

  Stereo vision serves as a powerful tool by capturing in-depth information, enabling a three-dimensional understanding of objects and scenes. Its versatility finds applications in diverse industries, showcasing its efficacy in providing spatial insights.

- **Algorithmic Approaches:**

  Algorithms in stereo vision balance speed, accuracy, and computational complexity, allowing for customization based on specific task requirements. YOLO NAS balances speed with accuracy and is a good fit provided computational power is not an issue. Environmental adaptability ensures the effectiveness of stereo vision algorithms in various scenarios.

- **Ongoing Development:**

  Stereo vision algorithms continue to evolve through research and development, contributing to enhanced performance and the ability to handle complex tasks.

All said, we have outlined the most crucial takeaways we feel will be the linchpin of this project:

- **Algorithm Selection:**

  Choose Semi-Global for high-velocity scenarios and Global for high-altitude objects for optimized depth perception.

- **Camera Frame Rate:**

  Emphasize high frame rates in cameras for enhanced precision, especially in dynamic environments.

- **Correspondence Challenge and Integrating Technologies:**

Address correspondence challenges by integrating sensor fusion and other complementary technologies for a robust solution.

- **Heat-Dissipating Technologies:**

  Invest in effective heat-dissipating technologies to manage computational demands and ensure consistent performance.

These strategic recommendations and key takeaways form a comprehensive framework for Sound Antidote, aligning its stereo vision system with cutting-edge advancements and industry best practices.

8. Conclusion

This paper delves into contemporary approaches for object detection and distance estimation. Our evaluation of YOLO-NAS and YOLOv8 highlights YOLO-NAS's efficiency and YOLOv8's precision, particularly in UAV detection. As UAV integration rises, the need for advanced detection systems is critical. We recommend Sound Antidote to implement the suggested strategies, emphasizing scenario-based algorithm selection, high camera frame rates, and the integration of complementary technologies to address correspondence challenges in stereo vision. Our conclusion urges Sound Antidote to leverage these insights, integrating proposed technological synergies through ongoing research and development, positioning the company as a leader in accurate object detection and velocity estimation applications.

9. References

a. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). https://ieeexplore.ieee.org/document/7780460
b. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in neural information processing systems (pp. 91-99). https://papers.nips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html
c. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European conference on computer vision (pp. 21-37). Springer, Cham. https://arxiv.org/abs/1512.02325
d. Rosebrock, A. (2020). Practical Python and OpenCV: An Introductory, Hands-On Guide to Image Processing and Computer Vision. PyImageSearch. https://pyimagesearch.com/practical-python-opencv/
e. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7310-7311). https://arxiv.org/abs/1611.10012
f. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857. https://arxiv.org/abs/1704.06857
g. Gonzalez, R., Woods, R., & Eddins, S. (2016). Digital Image Processing Using MATLAB. McGraw-Hill Education. https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf
h. Anderson, K., & Gaston, K. J. (2013). Lightweight unmanned aerial vehicles will revolutionize spatial ecology. Frontiers in Ecology and the Environment, 16(3), 138-146. https://www.researchgate.net/publication/259502892_Lightweight_unmanned_aerial_vehicles_will_revolutionize_spatial_ecology
i. Meier, P. (2015). Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response. CRC Press. https://www.researchgate.net/publication/320239565_Digital_Humanitarians_How_Big_Data_Is_Changing_the_Face_of_Humanitarian_Response_Patrick_Meier_2015_CRC_Press_Boca_Raton_FL_978-1-4822-4839-5_259_pp
j. Rath, S., & Rath, S. (2023, June 5). *Train YOLO-NAS on Custom Dataset*. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Examples and Tutorials. https://learnopencv.com/train-yolo-nas-on-custom-dataset/
k. Zaarane, A., Slimani, I., Okaishi, W. A., Atouf, I., & Hamdoun, A. (2020, January 8). *Distance measurement system for autonomous vehicles using Stereo*

Camera. Array. https://www.sciencedirect.com/science/article/pii/S2590005620300011

l. Hsu, T.-S., & Wang, T.-C. (n.d.). *An improvement stereo vision images processing for object distance measurement*. International Journal of Automation and Smart Technology. https://www.ausmt.org/index.php/AUSMT/article/view/460/0
m. Saito, T., Okubo, T., & Takahashi, N. (2020, December 1). *Robust and accurate object velocity detection by stereo camera for autonomous driving*. arXiv.org. https://arxiv.org/abs/2012.00353
n. McGuire, K., De Croon, G., De Wagter, C., Tuyls, K., & Kappen, H. J. (2017, April 1). *Efficient Optical Flow and Stereo Vision for Velocity Estimation and Obstacle Avoidance on an Autonomous Pocket Drone*. IEEE Robotics and Automation Letters. https://doi.org/10.1109/lra.2017.2658940
o. Ho, H. W., De Croon, G., & Chu, Q. (2017, March 28). *Distance and velocity estimation using optical flow from a monocular camera*. International Journal of Micro Air Vehicles. https://doi.org/10.1177/1756829317695566
p. Zaarane, A., Slimani, I., Hamdoun, A., & Atouf, I. (2019, January 9). *Real-Time Vehicle Detection Using Cross-Correlation and 2D-DWT for Feature Extraction*. Journal of Electrical and Computer Engineering. https://doi.org/10.1155/2019/6375176
q. *Medium*. (n.d.). Medium. https://richmondalake.medium.com/yolo_nas_uncovered_essential_insights_and_implementation_techniques_for_machine_learning_engineers_87ee266b37f6
r. Aswini, N., & Uma, S. (2019, October 1). *Obstacle avoidance and distance measurement for unmanned aerial vehicles using monocular vision*. International Journal of Power Electronics and Drive Systems. https://doi.org/10.11591/ijece.v9i5.pp3504-3511
s. *Correspondence problem*. (2022, December 10). Wikipedia. https://en.wikipedia.org/wiki/Correspondence_problem#:~:text=The%20correspondence%20problem%20refers%20to,of%20objects%20in%20the%20photos
t. Spurgeon, W. (n.d.). *Stereo Vision for 3D Machine Vision Applications*. https://www.clearview-imaging.com/en/blog/stereo-vision-for-3d-machine-vision-applications
u. Li, P. (n.d.). *Lecture 9 & 10: Stereo Vision*. Stanford Vision Lab http://vision.stanford.edu/teaching/cs131_fall1415/lectures/lecture9_10_stereo_cs131.pdf
v. Science, B. O. C., & Science, B. O. C. (2023, June 19). *Computer Vision: Determining the Distance From an Object in a Video | Baeldung on Computer Science*. Baeldung on Computer Science. https://www.baeldung.com/cs/cv-compute-distance-from-object-video
w. Vision, C. (2023, March 16). *What are the advantages and disadvantages of monocular vision for face recognition?* www.linkedin.com.

https://www.linkedin.com/advice/3/what-advantages-disadvantages-monocular-vision#what-are-the-advantages-of-monocular-vision-for-face-recognition
x. L. (2023, June 5). *YOLO-NAS: Step by Step Guide To Custom Object Detection Training*. YouTube. https://www.youtube.com/watch?v=vfQYRJ1x4Qg
y. F. P. O. C. V. (2021, April 25). *Finding Correspondences | Uncalibrated Stereo*. YouTube. https://www.youtube.com/watch?v=erpiFudDBlg
z. T. M. L. (2021, April 30). *Computer Vision - Lecture 4.2 (Stereo Reconstruction: Block Matching)*. YouTube. https://www.youtube.com/watch?v=EVzEJQl8WFk
aa. T. M. L. (2021, April 30). *Computer Vision - Lecture 4.5 (Stereo Reconstruction: End-to-End Learning)*. YouTube. https://www.youtube.com/watch?v=9vrmwZ9Pl4o
bb. S. S. (2014, September 20). *Distance to objects using single vision camera.* YouTube. https://www.youtube.com/watch?v=Qm7vunJAtKY
cc. *Uncalibrated Stereo Image Rectification - MATLAB & Simulink*. (n.d.). https://www.mathworks.com/help/vision/ug/uncalibrated-stereo-image-rectification.html

10. Appendix

Appendix A: Trade-off between YOLOv8 and YOLO-NAS

Feature	Yolo-NAS	Yolo-v8
Architecture	Quantization friendly	Transformer-based
Pre-training Schemes	COCO, Object365, & Roboflow 100	Pseudo-labeling, Knowledge Distillation
Post-training Quantization	Yes (INT8)	No
AutoNac	Yes	No
mAP	Higher	Lower
Latency (ms)	Lower	Higher
Small Object Detection	More Accurate	Less Accurate
Real Time Applications	Faster	Slower

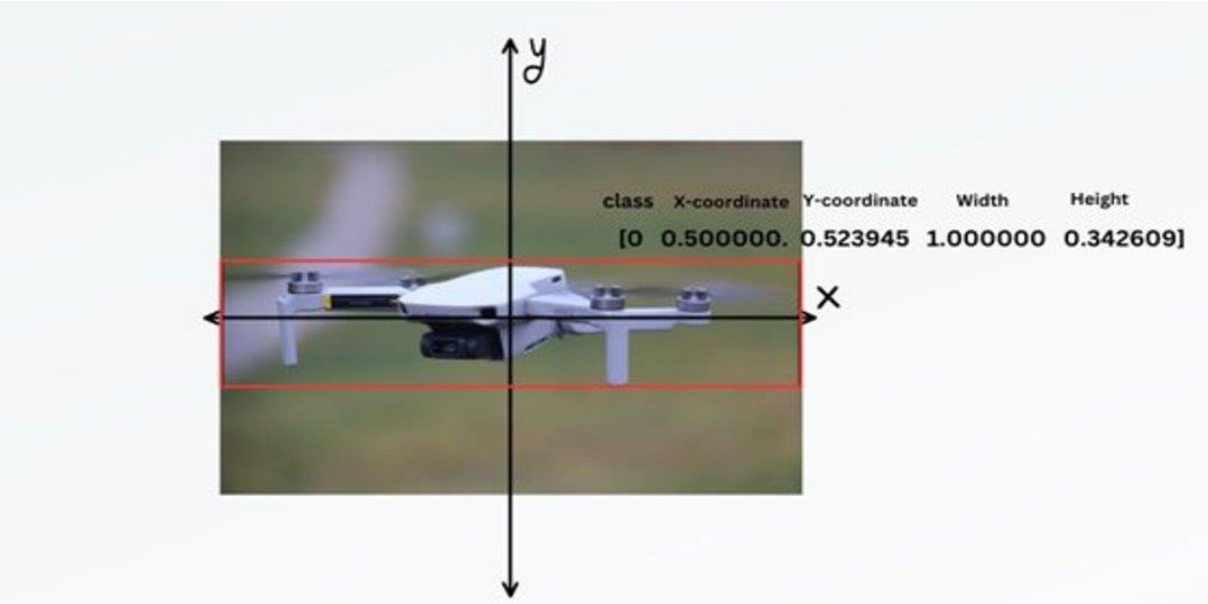
Appendix B: Types of drones

1. DJI Mavic

2. E-flite

3. DJI Phantom

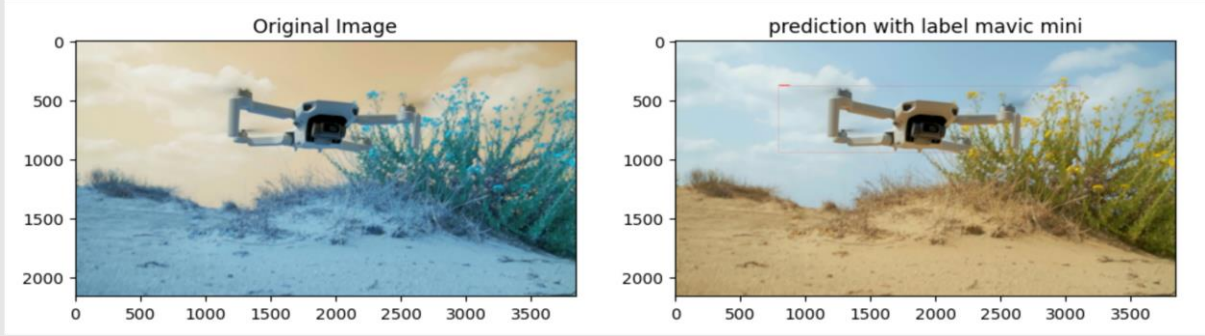
Appendix C: Annotations of images



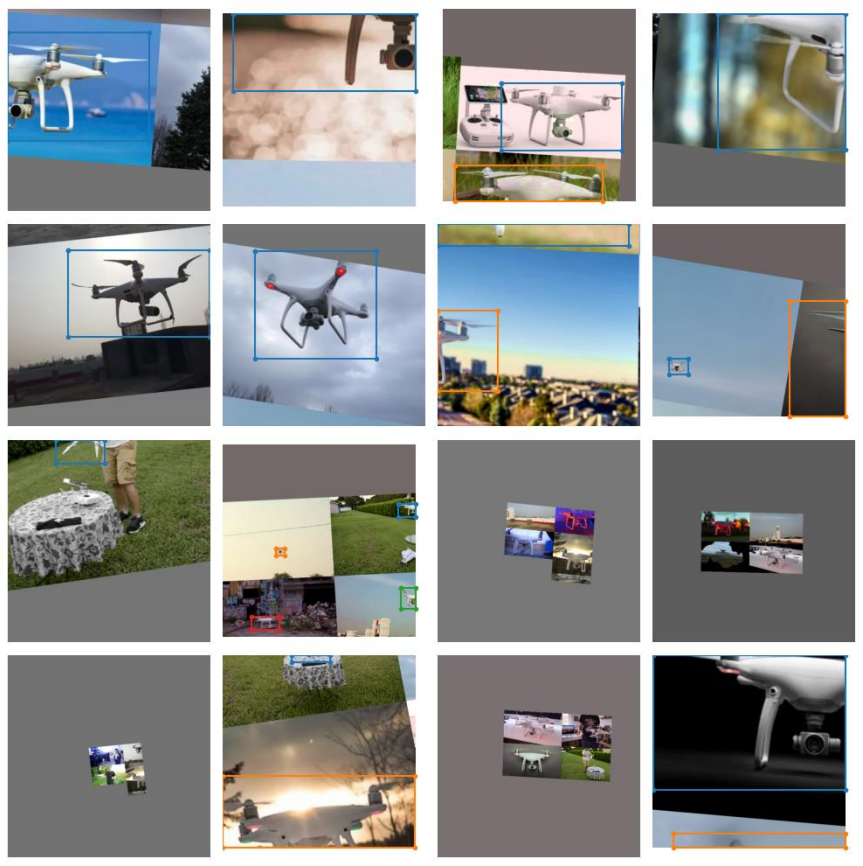
Appendix D: Mean Average Precision and Latency of the different versions of NAS on a pretrained model within NAS.

Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS M	51.55	5.85
YOLO-NAS L	52.22	7.87
YOLO-NAS S INT-8	47.03	2.36
YOLO-NAS M INT-8	51.0	3.78
YOLO-NAS L INT-8	52.1	4.78

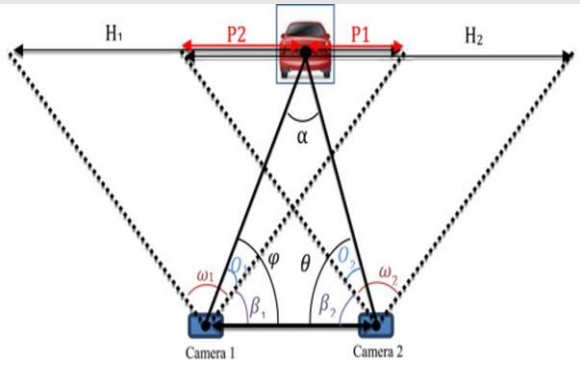
Appendix E: Result on Image dataset mavic-mini using Yolo V8

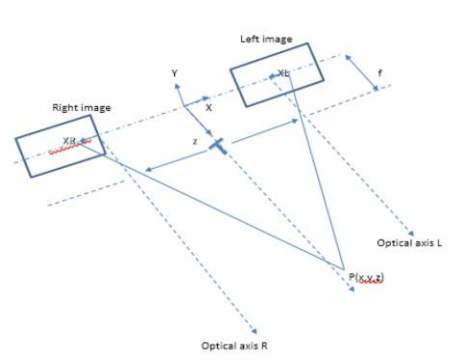


Appendix F: How Yolo NAS trains its model on phantom image dataset



Appendix G: Camera Setup & calibration





A is known. First find B and C then find θ and ϕ which will give α

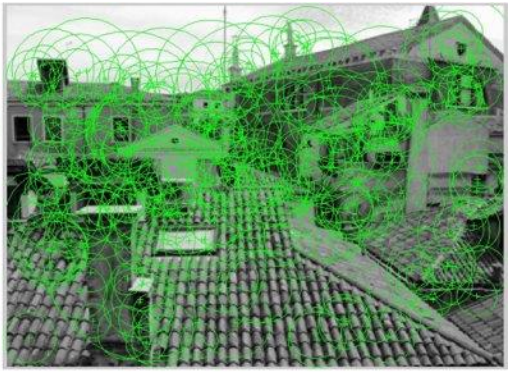
ω1, ω2: the view angles of the two cameras respectively.

H1, H2: the number of horizontal pixels of the two cameras respectively.

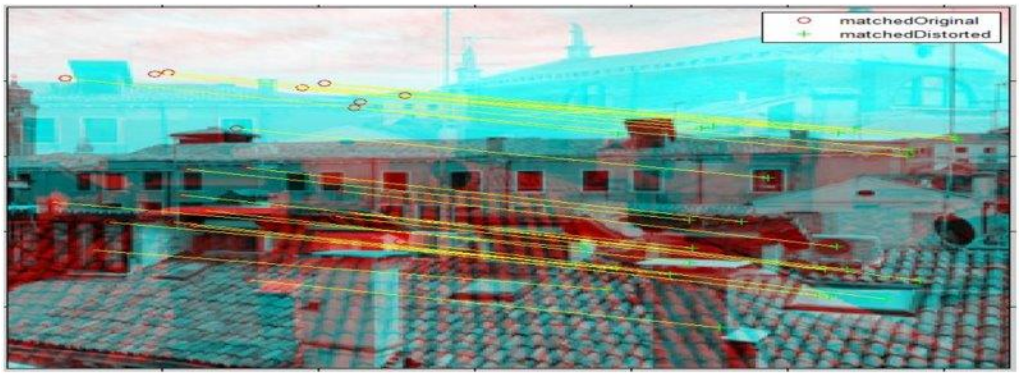
P1, P2: the position of the object in both cameras, where P1 is the distance in pixel between the centroid of the object and the end of the overlap area for the camera on the left. P2 is the distance in pixel between the centroid of the object and the beginning of the overlap area for the right camera.

Appendix H: Feature Matching

(f) Detected features using SURF in image1



(g) Detected features using SURF in image2

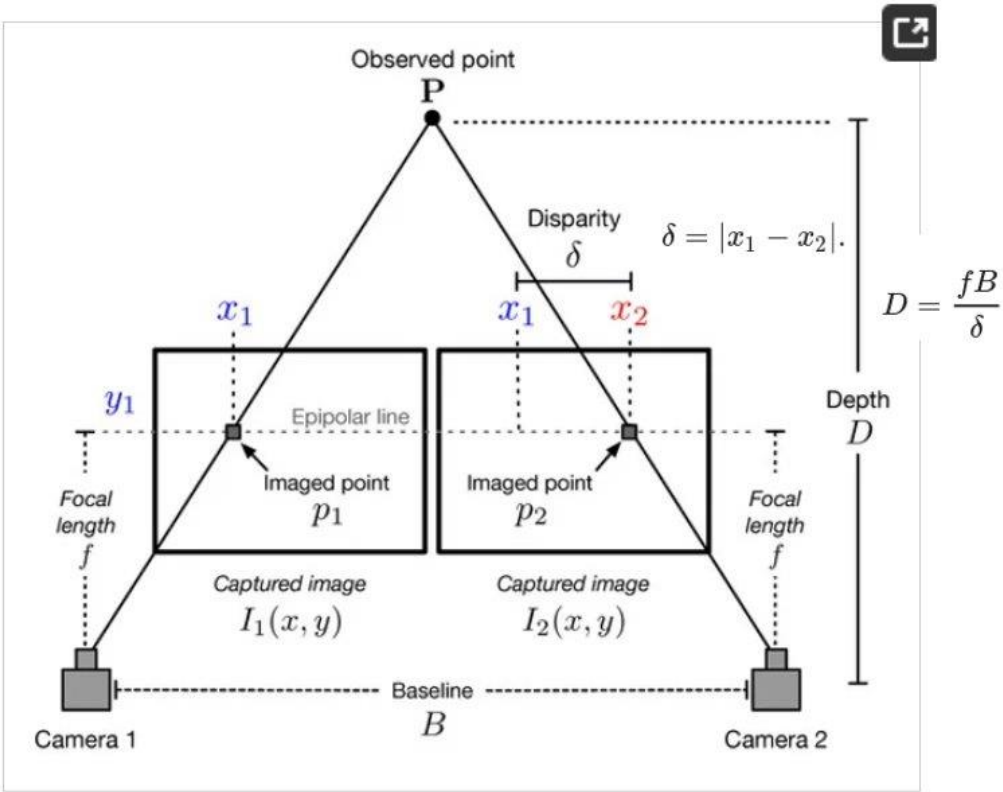


(h) Matching pairs identified between the image1 and image2

Appendix I: SIFT and SURF algorithms

ASPECTS	SIFT	SURF
Key Point Detection	Keypoints extracted by the SIFT detector.	Interest points detected using the determinant of the Hessian blob detector.
Descriptor Computation	Descriptors computed by the SIFT descriptor.	Feature descriptor based on the sum of the Haar wavelet response.
Invariance	Invariant against various transformations.	Not explicitly mentioned but designed to be robust against different image transformations.
Speed	Relatively slower compared to SURF.	Claimed to be several times faster than SIFT. Uses integer operations for efficiency.
Transformation Robustness	Designed to be invariant against transformations.	Claimed to be more robust against different image transformations.
Detection Technique	Keypoint detection by SIFT detector.	Interest point detection using the determinant of the Hessian blob detector.
Descriptor Method	SIFT descriptor.	Descriptor based on the sum of the Haar wavelet response.
Implementation Complexity	May have a higher computational cost.	Designed for efficiency and speed, potentially less computationally intensive.
Integral Image Usage	Not explicitly mentioned.	Uses a precomputed integral image for efficient computation of the determinant of the Hessian.
Cluttered Keypoints	May suffer from cluttered keypoints, affecting feature matching and stereo vision accuracy.	May suffer from cluttered keypoints, affecting feature matching and stereo vision accuracy.
Performance and Robustness	Performance may vary based on application and image characteristics.	Claimed to be faster and more robust, but reported cases where SIFT performs equally well or better in certain scenarios.
Speed and Efficiency	Speed and efficiency can vary based on implementation and use case.	Designed for speed, but actual performance may depend on the implementation and specific use case. Can impact suitability for real-time stereo vision applications.

Appendix J: Disparity Calculation

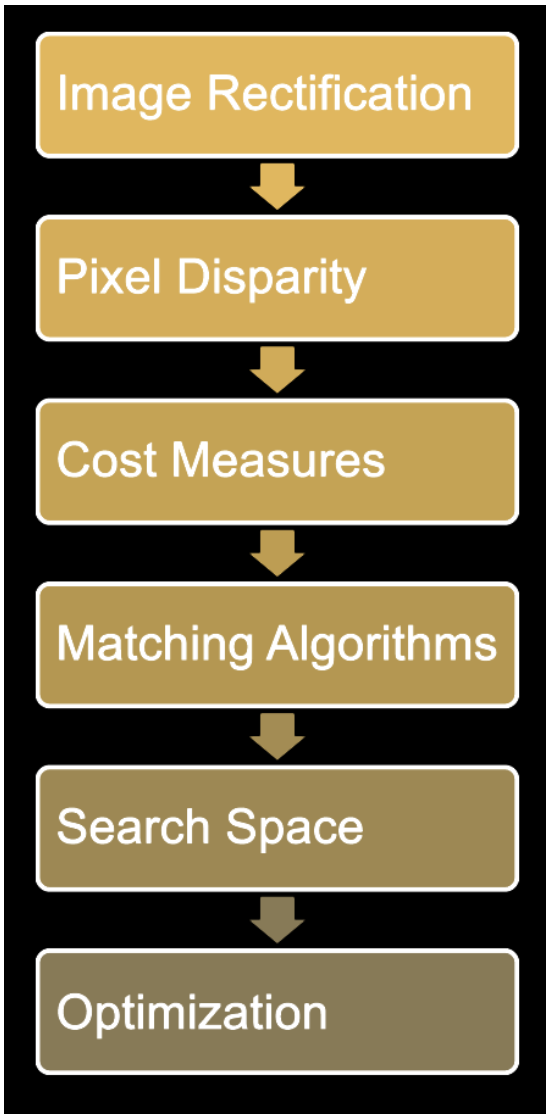


Relationship between Depth Estimation and Disparity in case of Stereo Image setup

Appendix K: Types of Stereo Matching

Aspect	Sparse Stereo Matching	Dense Stereo Matching
Focus	Recognized points (features) in the stereo pair.	Comprehensive, aims to match every pixel.
Computational Complexity	Low, historically favored for simplicity.	Modern methods address precision challenges.
Usage Trend	Declined due to challenges in pixel precision.	Rising popularity for a holistic solution.
Methodology	Feature-based using edge aspects, line segments.	Pixel-by-pixel, evaluates correspondence for all pixels.
Application	Suitable for specific points of interest.	Ideal for applications requiring detailed depth maps.

Appendix L: Stereo Matching



11. Authors Contact Information:

Abdur Rahman Danish – danish.a@northeastern.edu
Daniel Varghese - varghese.da@northeastern.edu
Divi Joshi - joshi.div@northeastern.edu
Ishan Lokhande - lokhande.i@northeastern.edu
Shivani Avasarala – avasarala.s@northeastern.edu