**Problem Statement / Objective**

- Time-consuming, expensive, and subjective modern PV anomaly detection would benefit from an anomaly detection algorithm [1-4]
  - Traditionally, checking for anomalies:
    - Requires turning off all PV panels
    - May be dangerous for operators, as PV panels are often at a significant height or steep angles

**Objectives**

- Develop a lightweight detection and classification method for anomalies using Deep Learning models with input IR images in real-time
- Implement onto an edge device to simplify, automate, and overall reduce the budget of PV panel anomaly detection

**Dataset & Data Preparation**

- Datasets are from public, online collections of IR images with an aerial view of PV panels
- Images went through the following preprocessing:
  - Resized to 640x640px
  - Rotate -10 to 10 degrees
  - Flipped vertically or horizontally
- Images containing no PV panels, “background images,” are around 10% of the training dataset

**Requirements**

- Software Requirements for running models includes Yolov5s and EfficientNet:
  - Operating System: Ubuntu 18.04
  - Language: Python >= 3.6.9
- Hardware Requirements:
  - Jetson Nano: NVIDIA Tegra X1, 3964 MiB
  - Language: Python >= 3.6.9
  - Operating System: Ubuntu 18.04

**Methods & Algorithms**

- **Yolov5s**
  - Open-source object detection model developed by Ultralytics for multi-class classification
  - Used in 1-Stage Approach to both detect and classify anomalies
  - Used in 2-Stage Approach to detect defect panels
  - Fast and efficient when doing detection with real-time inference
  - Yolov5s selected for its accuracy with minimal resources, best suited for the Jetson Nano, which has smallest model size

- **EfficientNet-B0**
  - Convolutional Neural Network developed by AutoML MNAS with comparatively better performance for our application
  - Used in 2-Stage Approach for Classification of all individual PV panels
  - Selected network B0 due to minimal resource use to implement on the Jetson Nano

**String Detector**

- String anomaly was most difficult for the above classification models to detect accurately
- Team-built CV method classifies abnormally bright panels as String anomalies using statistical analysis
- Threshold for String anomalies is 1.5 * mean brightness and 1.5 * median brightness
- Was used in both 1-Stage and 2-Stage Approach as the final decision-maker for anomalies to be classified as a String, but was found to have decreased performance of detection algorithm and is therefore not included in final implementation

**Results**

<table>
<thead>
<tr>
<th></th>
<th>1-Stage Approach</th>
<th>2-Stage Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP w/ String Detector</td>
<td>0.792</td>
<td>0.671</td>
</tr>
<tr>
<td>mAP w/o String Detector</td>
<td>0.872</td>
<td>0.708</td>
</tr>
<tr>
<td>Speed Test on Jetson Nano</td>
<td>1.016 sec/image</td>
<td>0.982 sec/image</td>
</tr>
<tr>
<td>Speed Test on w/o String Detector</td>
<td>0.473 sec/image</td>
<td>0.59 sec/image</td>
</tr>
</tbody>
</table>

**Conclusion**

- Our objective was to develop an anomaly detection algorithm as an alternative to traditional inspection techniques used by solar farm inspectors
  - Two solutions are proposed: 1-stage and 2-stage approaches, both of which offer more efficient and consistent predictions of solar panel anomalies than the conventional techniques
  - These anomaly detection algorithms can significantly improve the efficiency of inspections
  - Inspectors can rely on automated data analysis and machine learning to identify anomalies, enabling them to prioritize their efforts on specific panels that require attention. This streamlines the inspection process, saving valuable time and resources
  - In conclusion, our findings indicate that the 1-stage approach without a string detector exhibits the best performance, with high accuracy and real-time capabilities, making it an optimal solution for anomaly detection.

**Future Work**

- Implement into a drone equipped with an IR camera
- Real-time detection and classification within drone

**References**