Package theft is a problem that negatively impacts the public, especially during a pandemic situation where the demand for online shopping and package delivery increases. This project aims to build a machine learning-based software system which can identify the action of package theft and improve home security for families and companies. The main deliverables of this project are a large-scale package theft detection dataset and a machine learning system built upon that.

**Dataset Collection and Annotation**
- Balanced sampling mechanism is used to avoid dataset bias, such as package theft, normal delivery, normal pickup, and irrelevant normal videos.
- In our dataset, videos are downloaded from public Internet sources, as well as recorded by our team members.
- Videos are annotated in the format of: video name, class category, starting and ending frame of package theft action.
- Breakdown of the Package Theft Detection Dataset:

<table>
<thead>
<tr>
<th>Sub-category</th>
<th>Normal video</th>
<th>Package theft video</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Irrelevant</td>
<td>500</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td>Normal</td>
<td>85</td>
<td>1097</td>
<td>1182</td>
</tr>
<tr>
<td>Normal pick-up</td>
<td>500</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td>Normal delivery</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>2182</td>
<td></td>
<td>2182</td>
</tr>
</tbody>
</table>

**Human Pose Estimation Model**
- Human pose estimation is the problem of localizing human joints in images or videos. We used OpenPose,[2] one of the most widely used estimators, as our tool.
- In each picture, or each frame in the video, OpenPose will provide us the human joints information in the format of 18 key points, along with the x, y coordinates and its confidence. The image to the right-top shows the index and human joints relations.[2]
- The image to the right-bottom shows how the OpenPose works on our video. Overall the ratio of human pose detected is about 60%. The accuracy will increase when people are closer to camera. The code we used is a pytorch version of OpenPose from [3].
- After getting the human pose information, we concatenate them to the end of the C3D feature from the video, and feed the integrated data into the anomaly detection neural network.

**End-to-End Anomaly Detection Model**
- As our main system, the anomaly detection model takes videos as inputs and generates anomaly scores for video segments, ranging from 0 to 1 (0 = normal, 1 = abnormal).[1]
- This model serves as the end-to-end solution for package theft detection.
- This model can leverage weakly-labelled training videos by using MIL ranking loss function with sparsity and smoothness constraints.
- After training this model with package theft and normal videos, it can generate high anomaly scores for package theft scenes but low anomaly scores for normal scenes.
- The image below is our pipeline. It is modified from the pipeline of [1] paper.

**Machine Learning Models**
- Two deep learning systems are used to build our package theft detection system.
  - Anomaly detection system is used to detect the package theft activities.
  - Human pose estimation system is integrated with the anomaly detection system to improve detection accuracy.

**Future Work and References**

**Result and Evaluation**

**Result:**
- The x-axis in the pictures represents the frame number and the y-axis represents the anomaly score. Pictures below show that our model can generate high anomaly score for package theft scenes and low anomaly score for normal scenes.

**Evaluation:**
- The evaluation metrics used in this project are ROC curve and AUC:
  - ROC (Receiver Operating Characteristics Curve): The ROC curve shows the performance of a classification model at all classification thresholds.
  - AUC (Area under the ROC Curve): AUC tells how much the model is capable of distinguishing between different classes.
- The following figure shows the performance of our model:

**Future Work:**
1. Add object detection to the system to improve the performance.
2. Train the model with more data to improve the performance.