**Question 1: Open Edge**

* For the more robust and generalized model, we will need the reference area that contains the edge (or fence) which will be hard coded by user like the below illustration.

A construction site with workers

Description automatically generated

* The green area is input or drawn by user which will contain the edge (or fence). Therefore, the position of fence will be hard coded to be more robust and generalized for all scenarios.

A blurry image of a bench

Description automatically generated

* We could crop the green area to get just the fence area and remove all remaining background area for reducing noise which will make the Classification model confused.
* After we get the green area, we could use this area as input data for **Classification model** to classify whether the edge (or fence) is **open** or **close**. Before input the image to model, we should Encode the image first, we could use:
  + **Vision Transformer** which will encode the image into image representation, so we could use this image representation for classify task.
  + Or we could use **Vision Transformer from CLIP** model, after Vision Transformer being trained within CLIP model, ViT could understand the context of the image more deeply. Therefore, it could be a good choice for extracting the image representation.
* Note: Before using ViT, we must fine-tune it on the dataset about **opened fence** and **closed fence**. Therefore, the model could understand the context of edge (or fence).

A blurry image of a person sitting on a bench

Description automatically generated

Closed Fence

A green line in a corner

Description automatically generated

Opened Fence

* In the training phase of Classification model, we could use the Data Augmentation Technique to enrich the training dataset. Which could help us to cover all the scenarios and make the model more generalized and robust, and it also resolves the issue of small training dataset.

**Question 2: Lifting Load Danger**

* Before running the system, user is prompted to input 2 values:
  + The angle of mounted camera with respect to the ground.
  + And the reference area represents the ground and serves as a basis for the bird’s eye view transformation.
* Assuming that we have detected the object (loads) already using the Object Detection model (e.g Yolov5), now we need to project the object to the ground area to find the **falling zone** of an object.

A red line on a metal surface

Description automatically generated

* To find the projection of the object to the ground area, we could use the Bird’s eye Transformation in Computer Vision to transform the image into the Top-down perspective, to use this transformation we need the Angle and Reference area to transform.
* After we get the transformed image, we now could find the projection of the bounding box of loads into the ground is the **transformed bounding box of loads**. Then we store the location of transformed bounding box as the falling zone of loads when we inverse the transformation process.

**Question 3: Class hierarchy**

1. The solution is stored in the folder “question3”.
2. A close-up of a fruit

   Description automatically generated

* To resolve the problem of Softmax loss function, we could change from single class classification to multi-label classification by using Hamming loss function. In hamming loss, it measures the percentage of incorrectly predicted labels compared to the true labels. Unlike Softmax Loss directly optimizes the number of label errors rather than probabilities estimates.
* We could change to use the Focal Loss to addresses the problem of label suppression by assigning dynamically scaled weights to each class based on their loss contribution. With this loss, we could downweight the loss assigned to the dominating class and emphasize the loss for other relevant classes.

1. The implementation for this problem is stored in folder “question4”.