

# PACTE

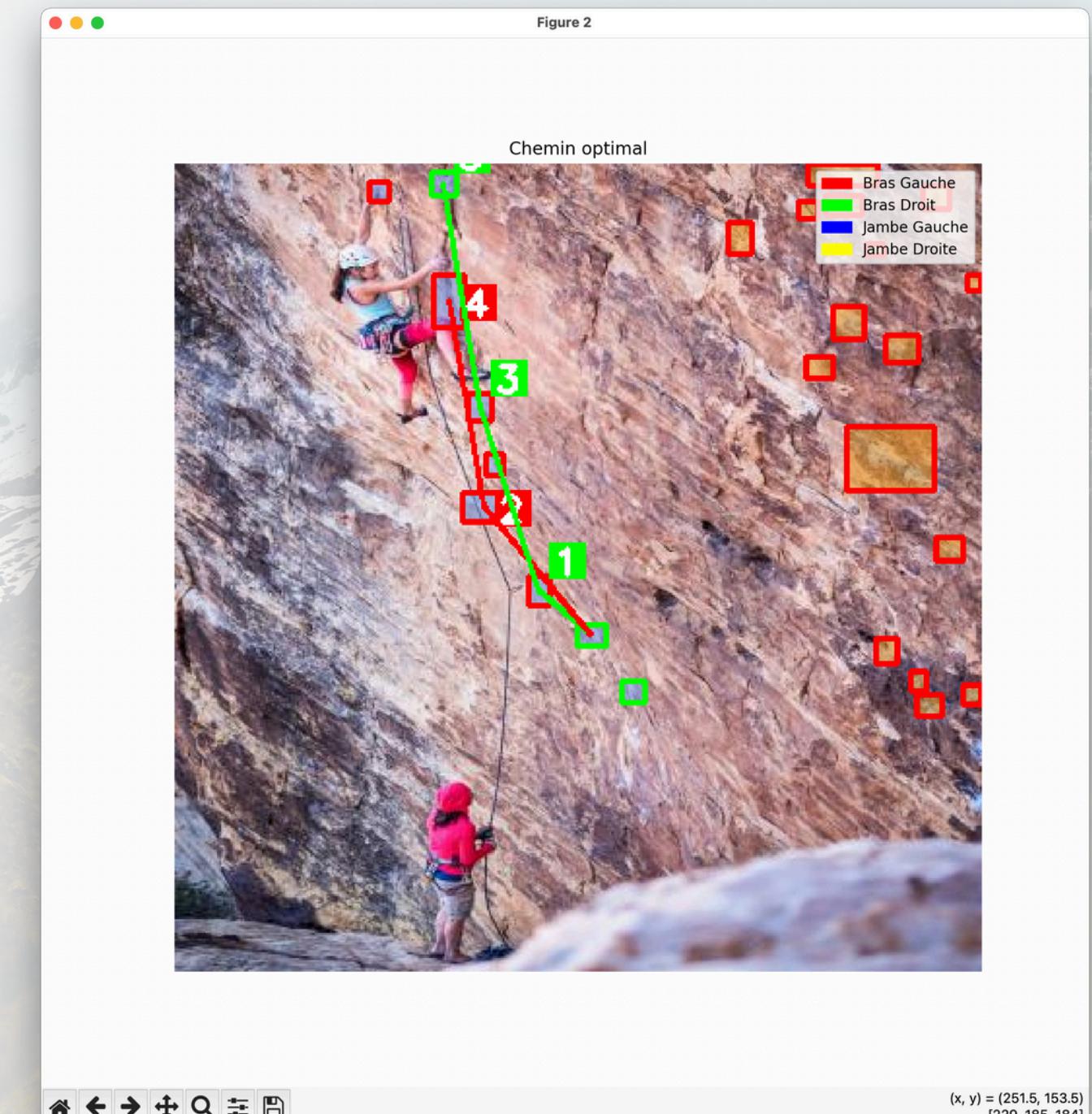
## Prise Analysis and Climbing Trajectory Estimation

Alessandro ARENSBERG AND Guillaume RABEAU

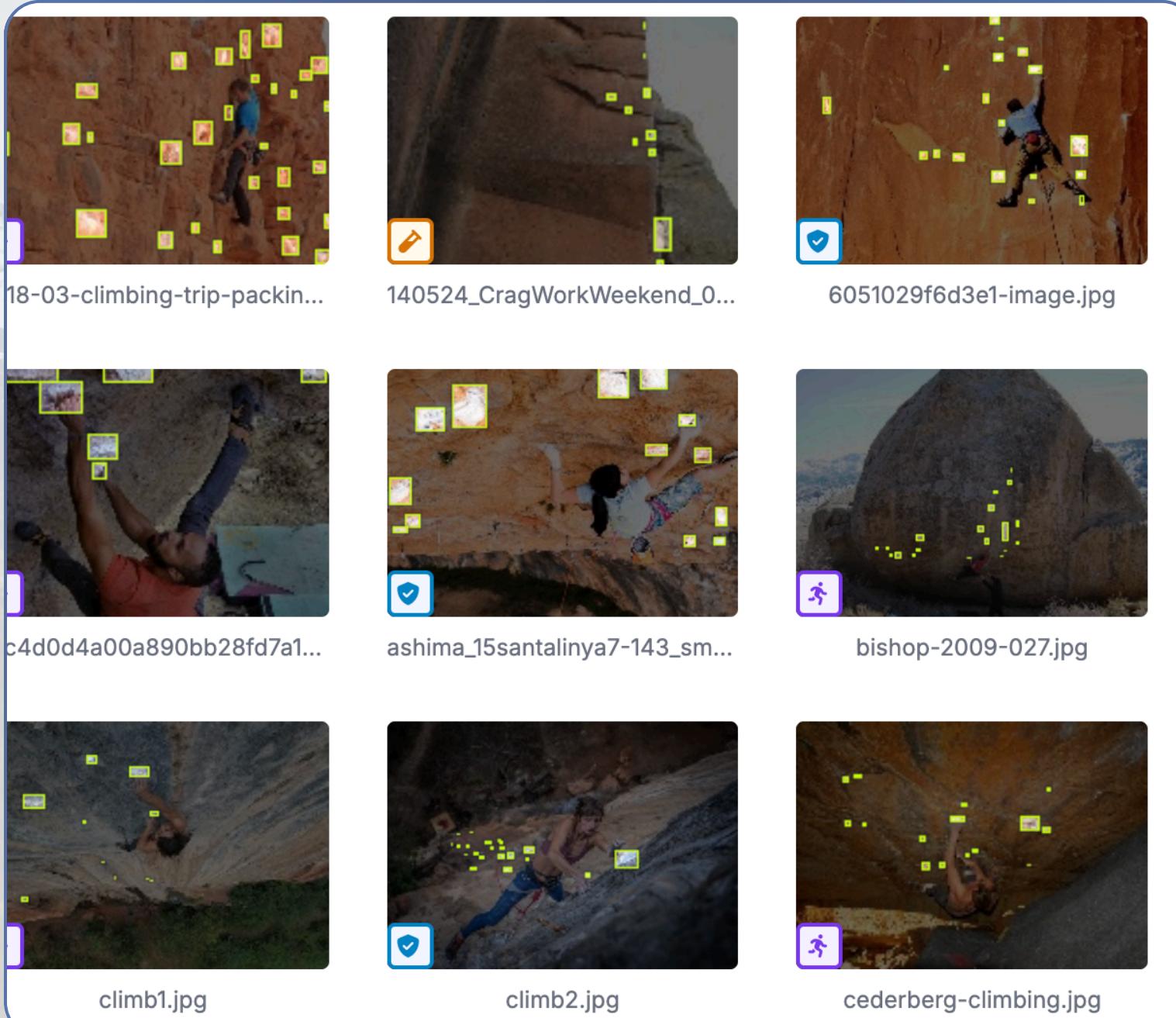


How easy can we  
find a way to  
climb a wall?

# Project objectives



# Dataset



**Train**  
45 images

**Test**  
13 images

**Valid**  
6 images

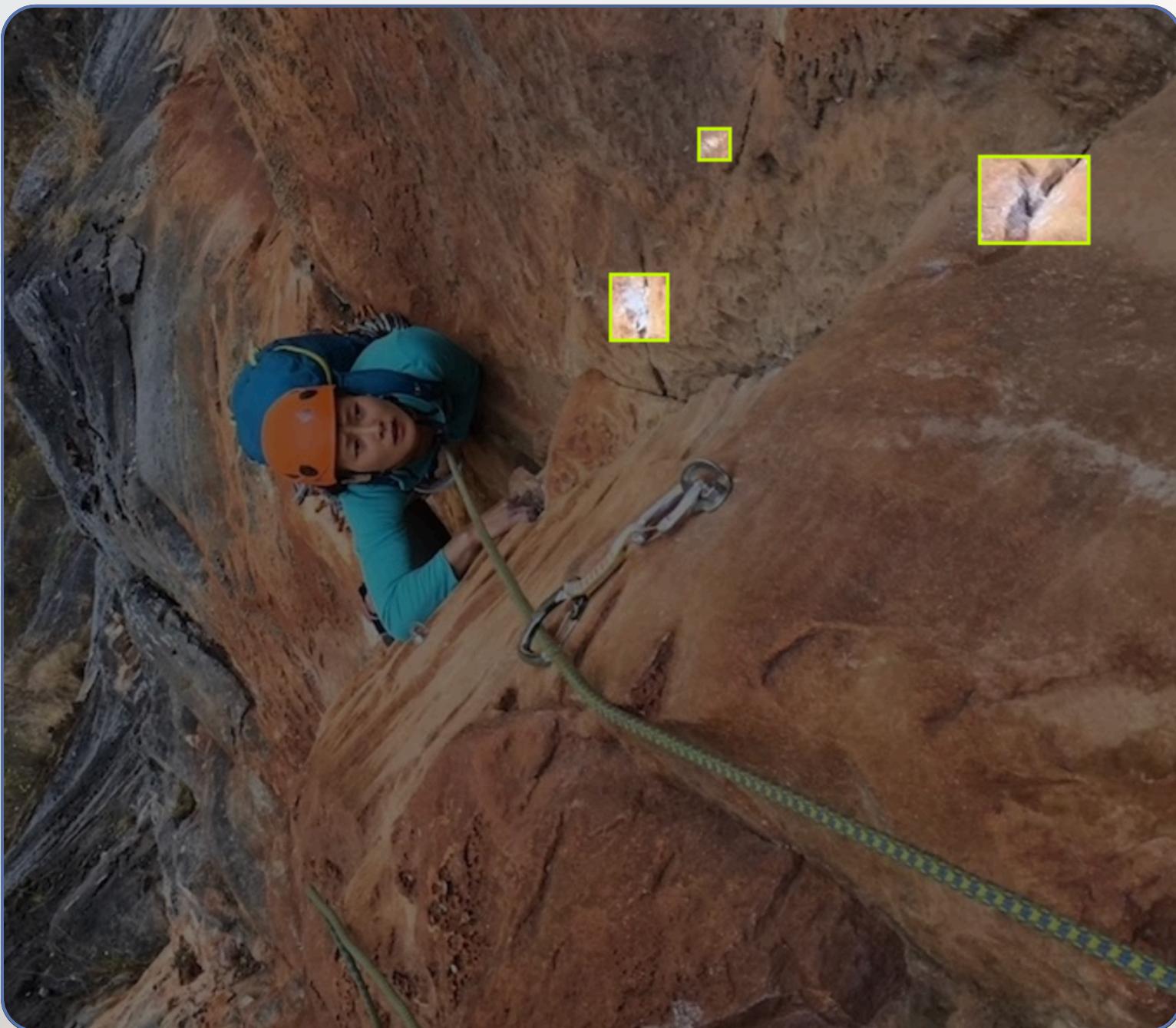
x22

**Train**  
990 images

**Images** : 1009  
**Source** : Universe  
Roboflow



# Dataset



**Train**  
45 images

**Test**  
13 images

**Valid**  
6 images

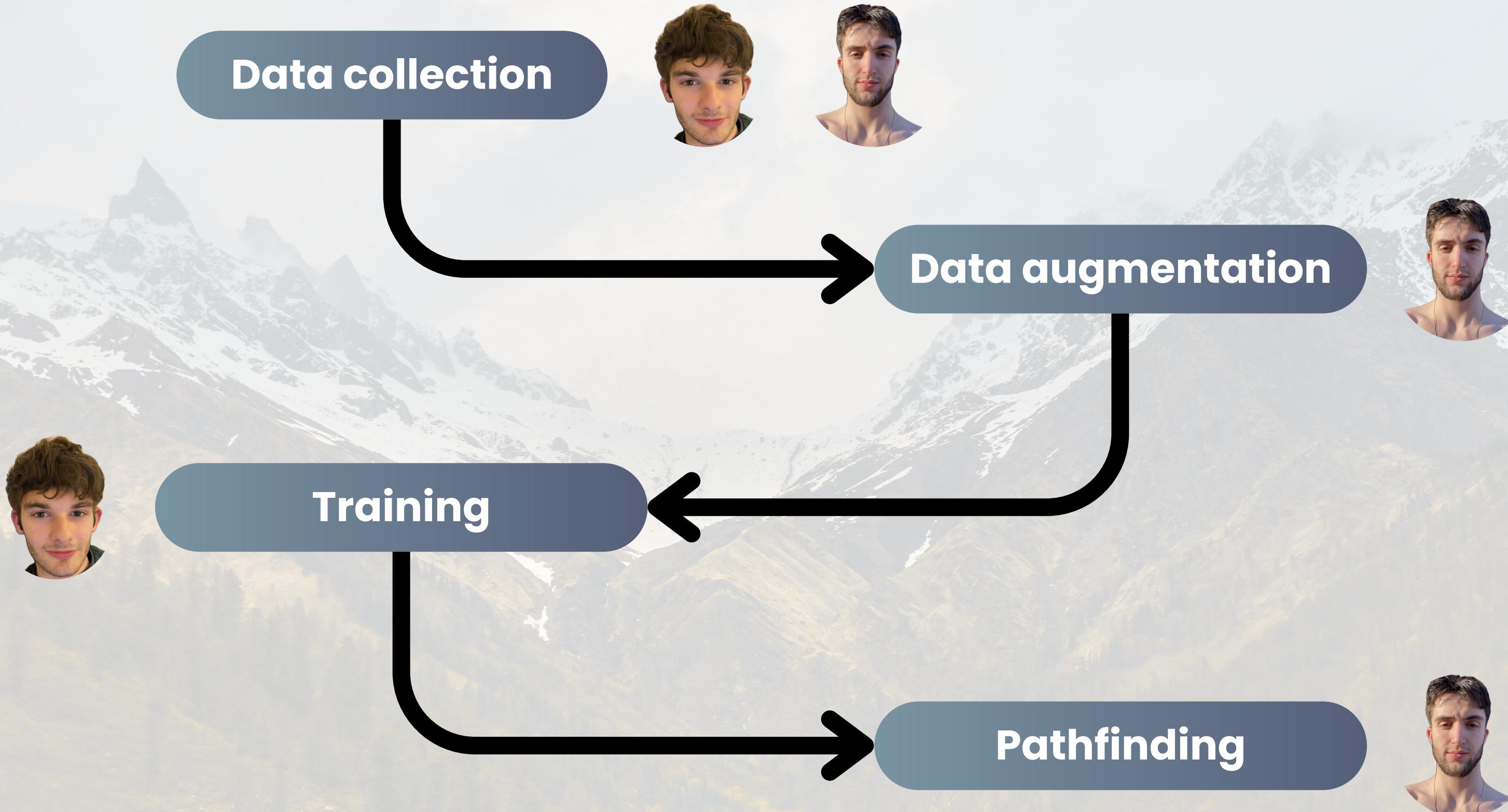
x22

**Train**  
990 images

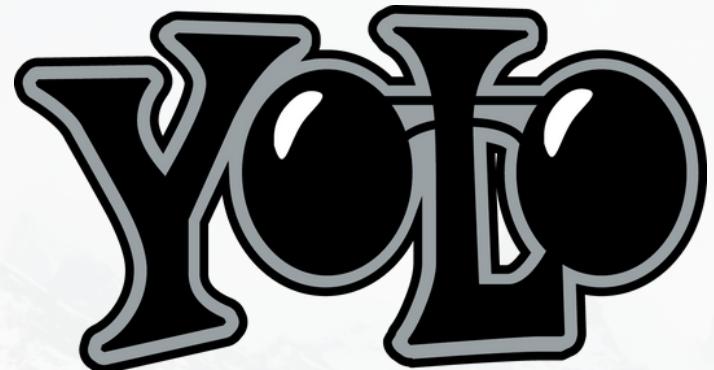
**Images** : 1009  
**Source** : Universe  
Roboflow



# Steps



# Methodology



$$L_{total} = \lambda_{cls} \cdot L_{cls} + \lambda_{loc} \cdot L_{loc} + \lambda_{obj} \cdot L_{obj}$$

## 1. Classification Loss (`cls_loss`)

$$\lambda_{cls} = 0.5, \lambda_{loc} = 2.5, \lambda_{obj} = 1.0$$

Measures whether the model correctly predicts the class of each object. Uses focal loss to handle class imbalance.

## 2. Localization Loss (`loc_loss`)

Evaluates how well the predicted boxes match the ground truth. It combines:

- CIoU loss: considers the distance, size, and aspect ratio between predicted and target boxes.
- Edge alignment loss: encourages boxes to align with image edges.
- Box ratio loss : penalizes unrealistic width/height ratios.

## 3. Objectness Loss (`obj_loss`)

Measures whether the model correctly detects the presence of an object. Also uses focal loss.

# Pathfinding

1

**Box clustering** : DBSCAN for grouping boxes into clusters based on spatial proximity

2

## Constructing a graph

- **Nodes** represent sockets
- **Edges** represent links between nodes if the maximum distance is respected
- **Weights** are adjusted to favor movement within the same cluster

3

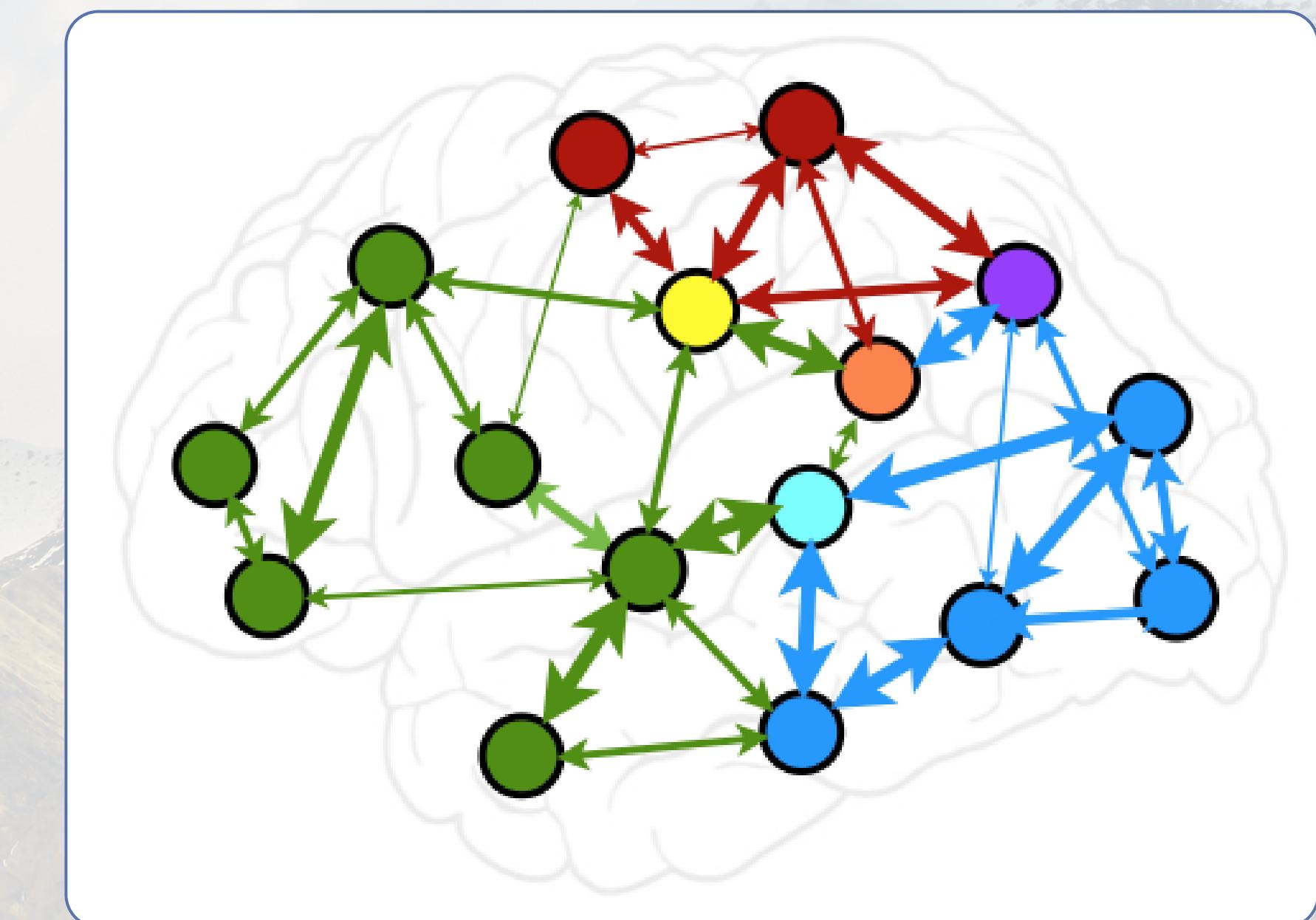
## Algorithm A\*

Explores possible states, prioritizing those that minimize total cost (number of moves + distance to goal)

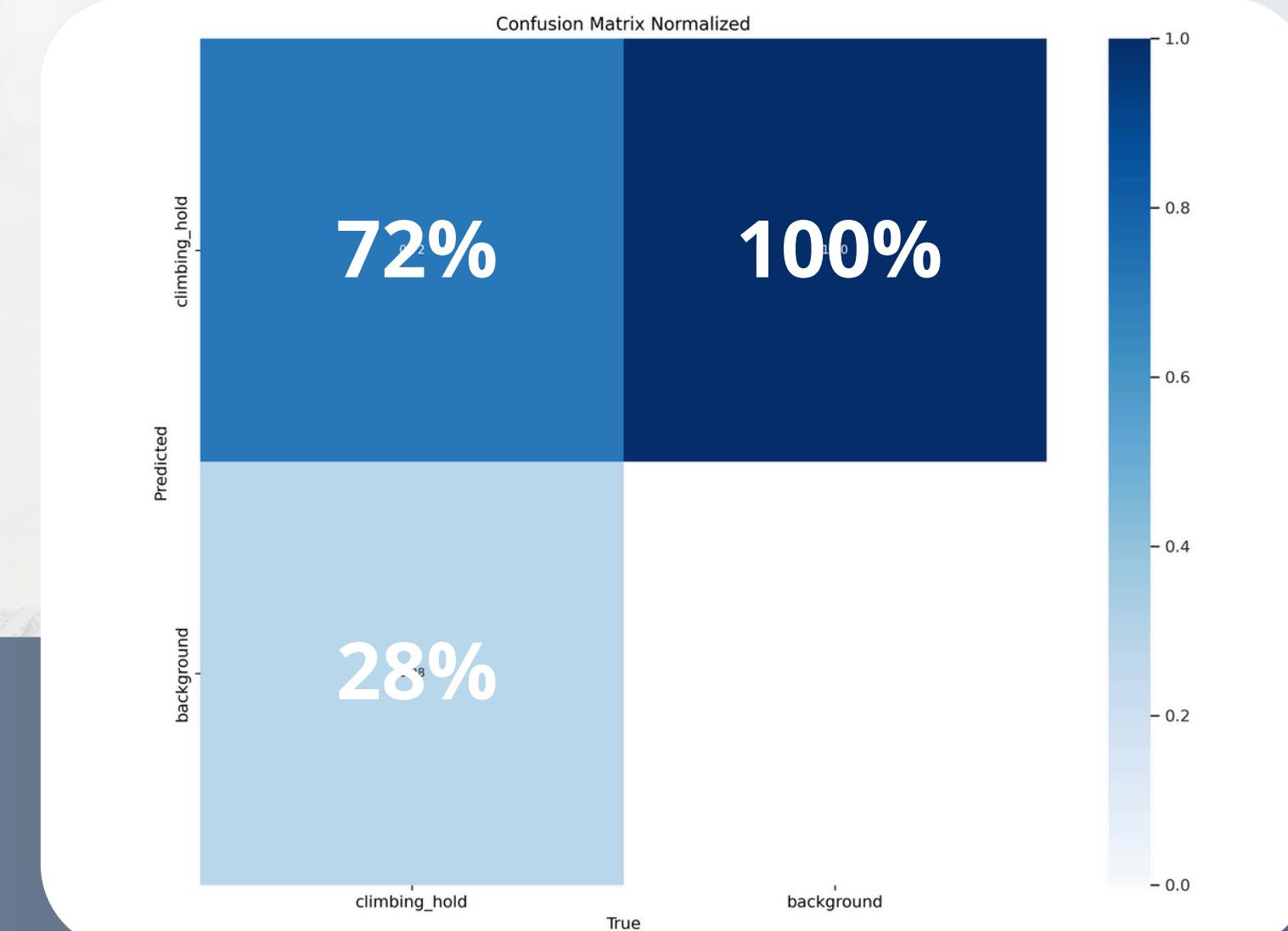
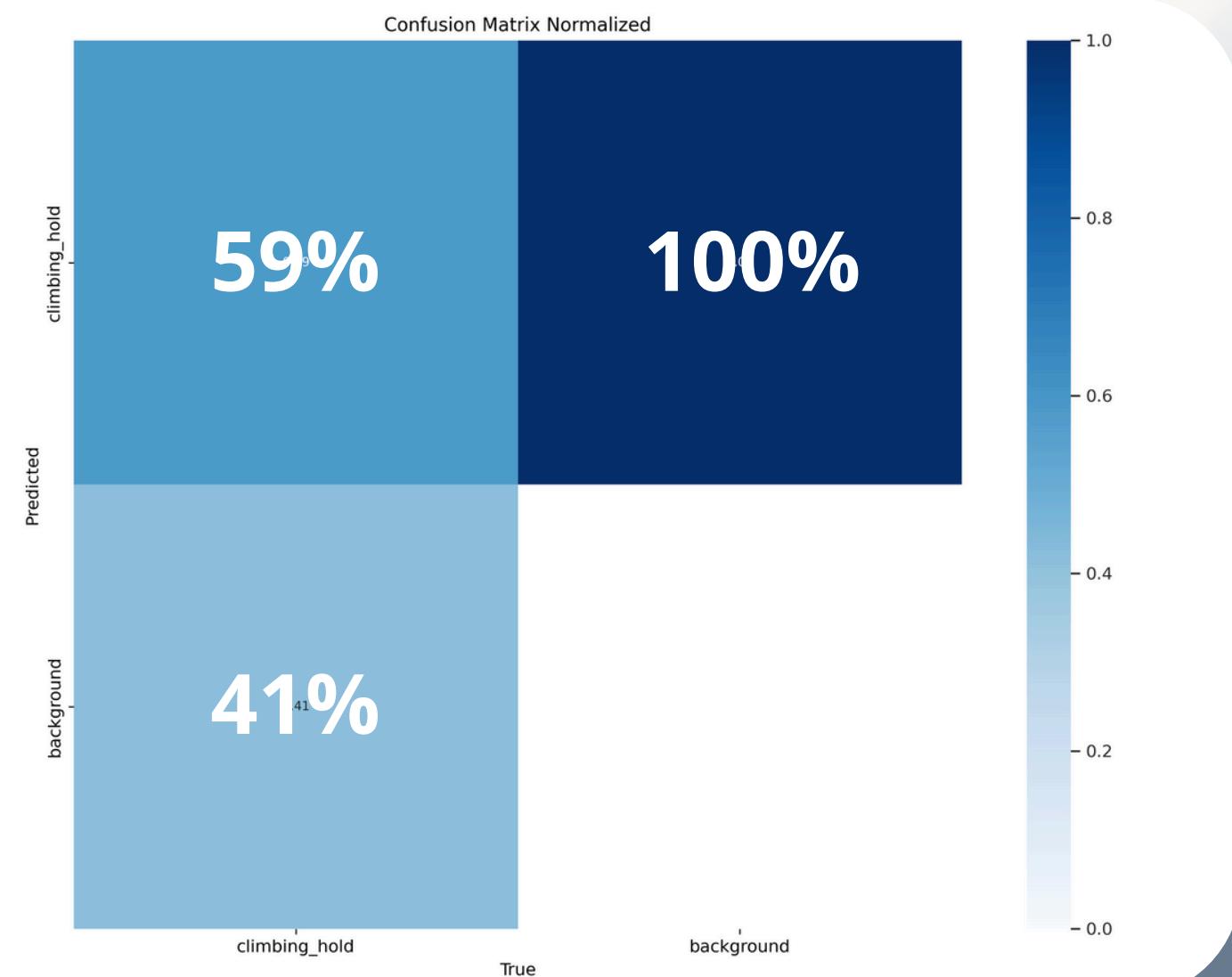
4

## Parameters

- **Initial state** : start and finish holds
- **Status check** (members too far apart; number of engaged members; at least one foot on the ground)



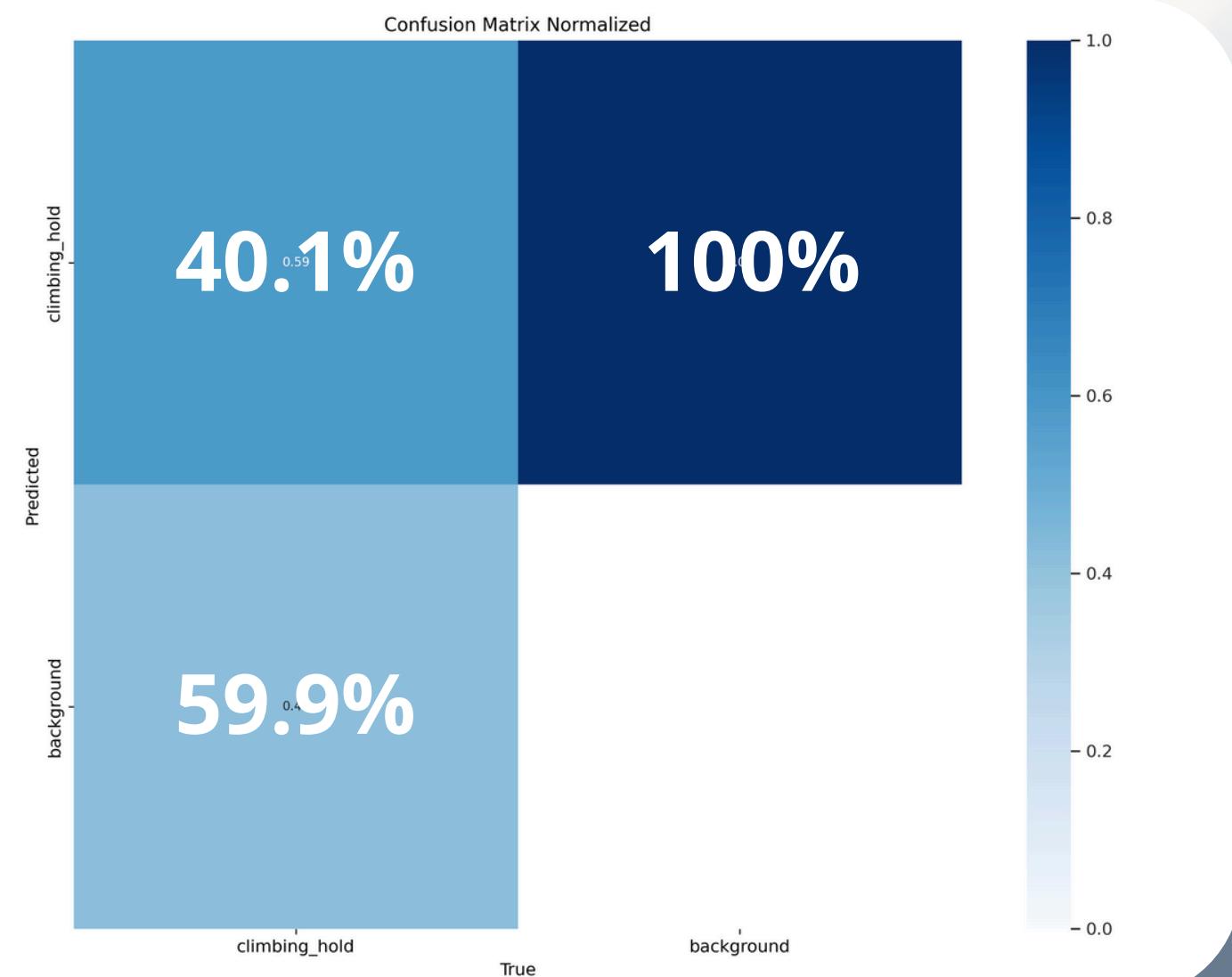
# Prediction (train)



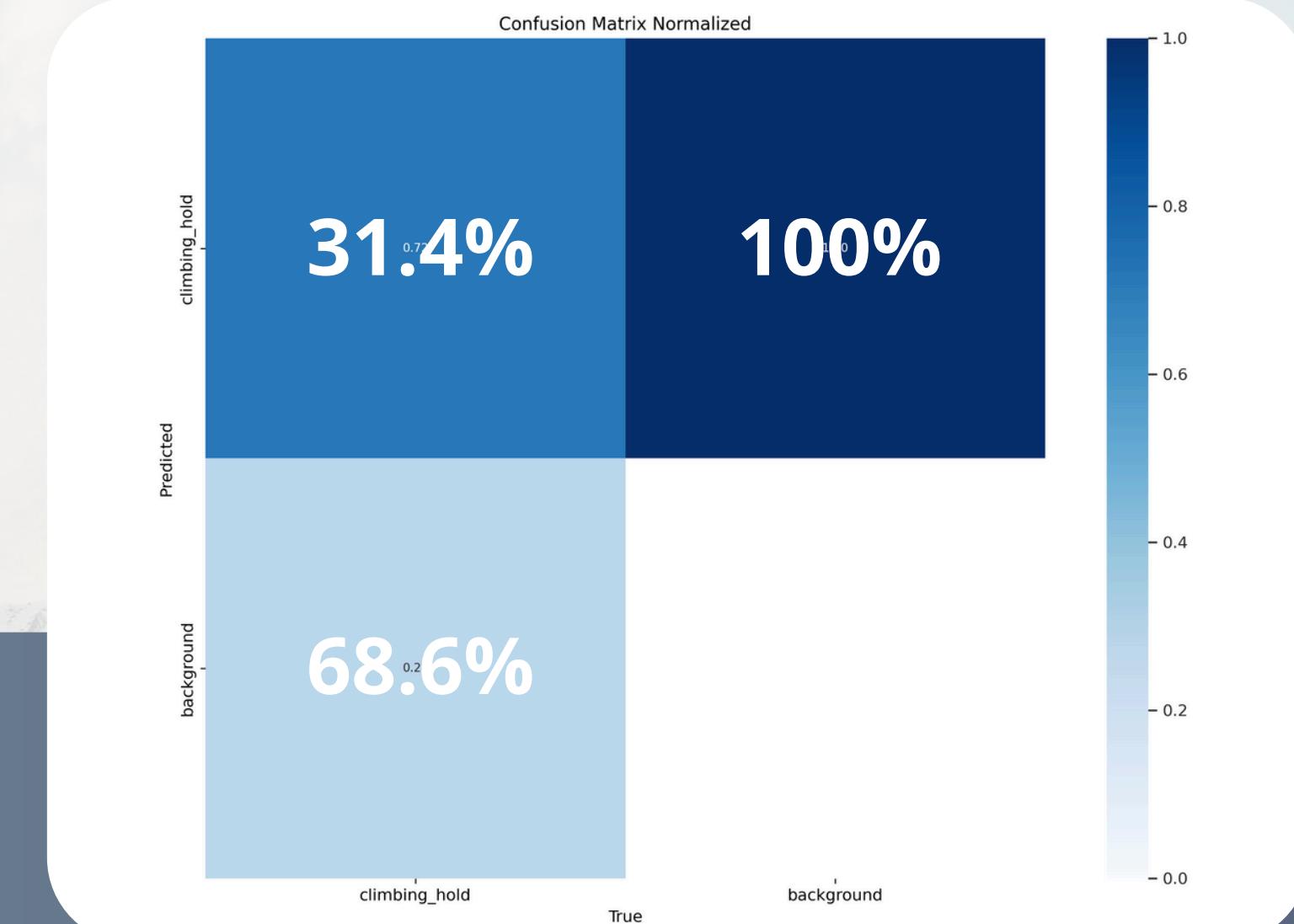
PACTEvFinale

PACTEv3

# Prediction (test)

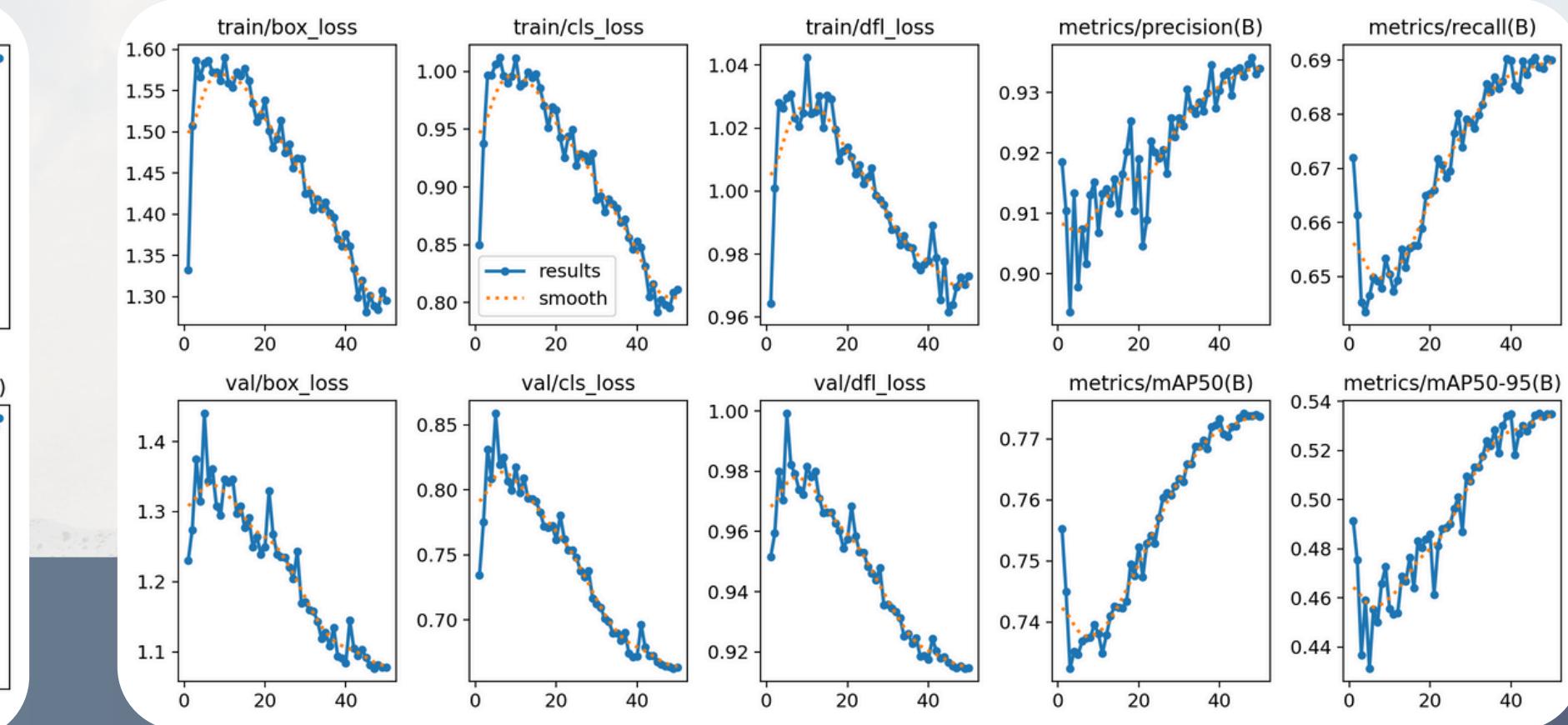
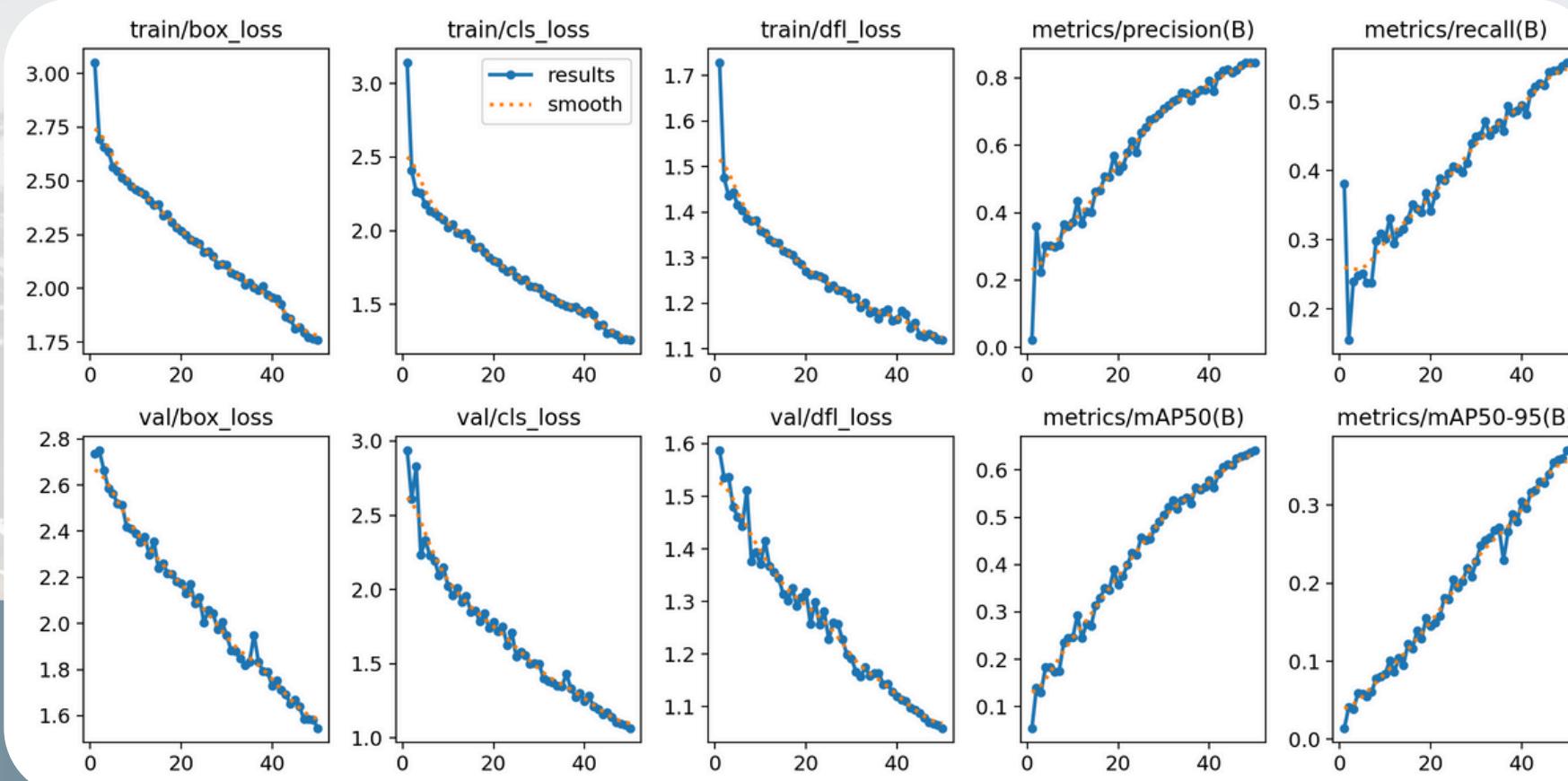


PACTEvFinale



PACTEv3

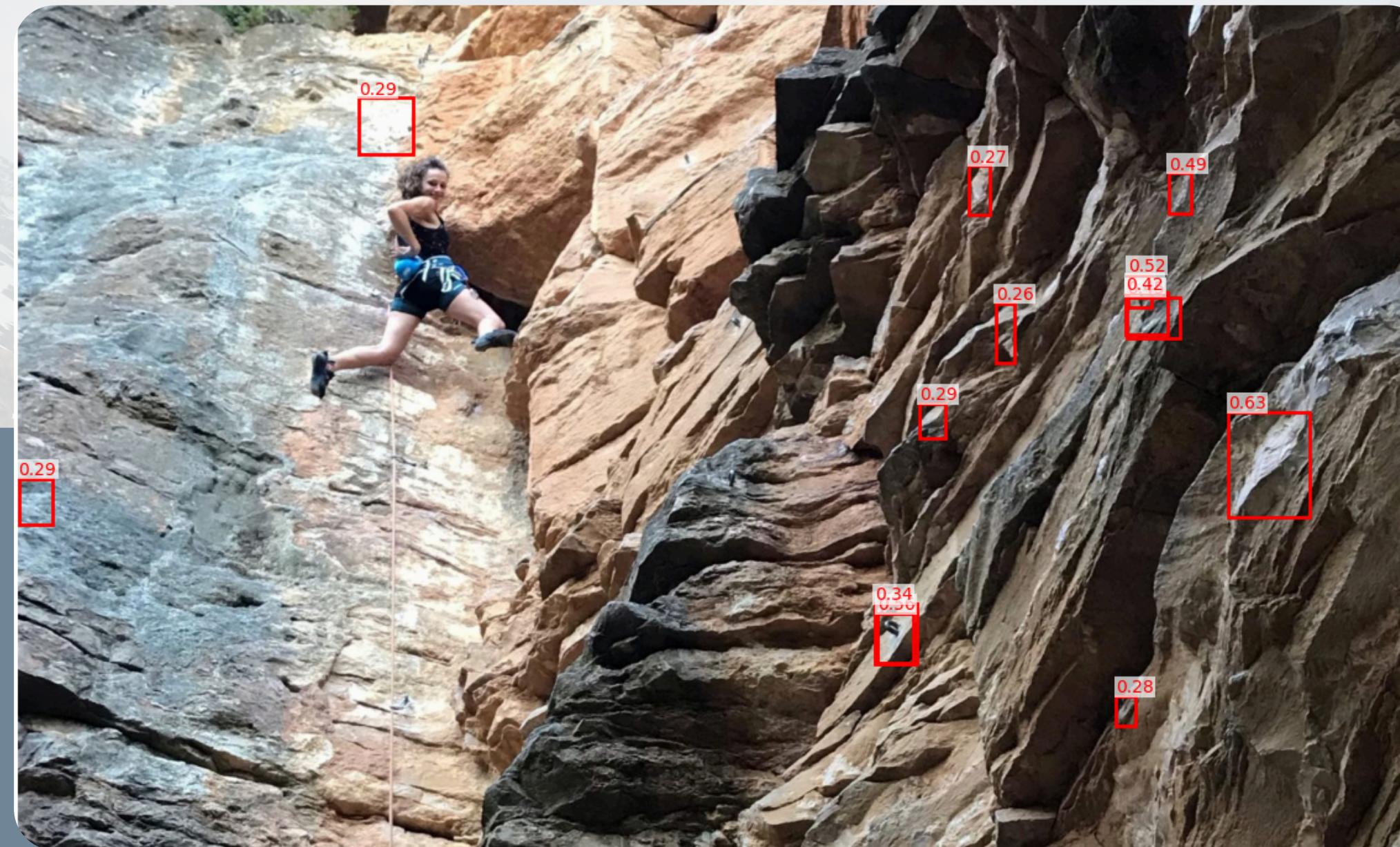
# Prediction



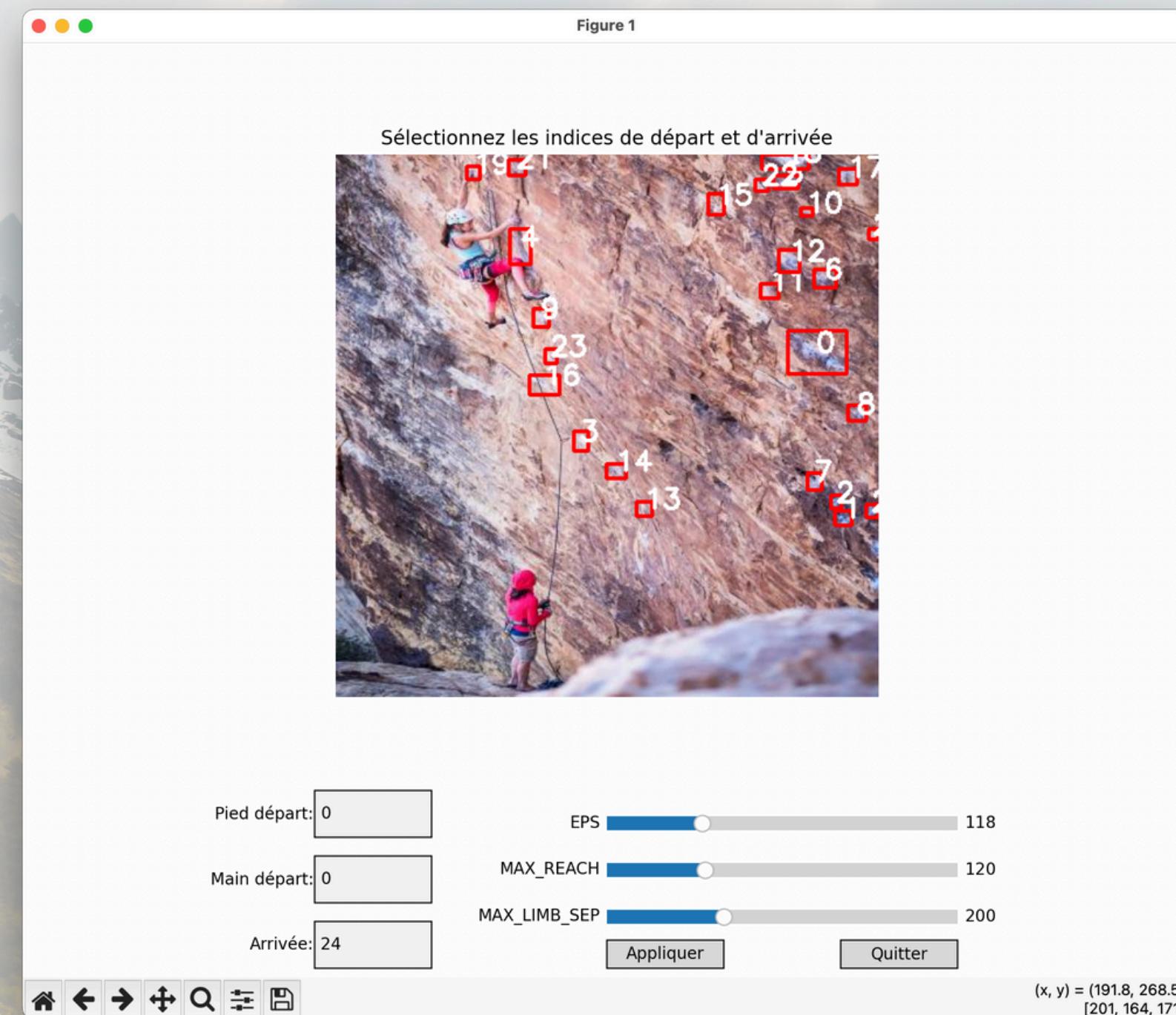
PACTEvFinale

PACTEv3

# Prediction



# APPLICATION





# Thank You

“ Pierre qui roule n'amasse pas mousse. ”