Your mission is to build a **reusable, modular and robust pipeline** for detecting **early forest disturbances**, with a focus on **bark beetle outbreaks**, by reusing and improving the existing fordead

package.

You'll (1) productionize and parameterize the fordead pipeline, (2) use the PhiDown package to ingest Sentinel-2 data, (3) apply change-point detection with ruptures to detect

disturbance onset more precisely, and (4) assess your detection using the <u>European Forest</u>

<u>Disturbance Map</u> and **ERA5 wind speed** data to distinguish between **wind damage and bark**beetle infestations.

© Core Features

1. Modular Inference Pipeline

- Reuse the fordead_package GitLab code and wrap it in a modular, CLI-based inference pipeline.
- Parameterize the pipeline to allow users to easily input:
 - A Region of Interest (ROI) (GeoJSON or GeoPackage),
 - A date range (e.g., 2018–2025),
 - Output paths.

2. Sentinel-2 Ingestion with PhiDown

- Use PhiDown to download Sentinel-2 Level 2A imagery over the ROI and date range.
- Select the most relevant bands for forest monitoring (e.g., B4, B8, B11, NDVI).
- Build cleaned, temporal stacks ready for inference.

3. Precise Disturbance Detection with ruptures

- For pixels or segments with detected disturbances, apply <u>ruptures</u> (or if you know a more relevant method, please use it and document why your choice) to **refine the** precise moment of change within the year.
- Goal: transform "this year had a disturbance" into "the change happened around May 23rd".

4. Wind vs Bark Beetle Differentiation (via ERA5)

- Use the European Forest Disturbance Map as reference data.
- Focus on the "Wind or Bark Beetle" labeled events.
- For each disturbance polygon:
 - Use ERA5 wind data to compute annual wind maxima at the time of detection.
- Apply a heuristic or threshold to infer the likely origin:
 - o High wind speed → Wind
 - o Low wind speed → Bark beetle
- Compare your pipeline's disturbance detection with these reference events:
 - Compute detection lead time,
 - o Precision / Recall for bark beetle-likely zones.

Particular Season Problems Bonus Features (Optional but appreciated)

♦ Bonus A – Bulk Evaluation on All Bark Beetle Events

- Run your pipeline over all bark beetle-likely zones (filtered from the map using ERA5based separation).
- Evaluate your detection rate and average lead time.
- Generate plots: ROC, PR curve, lead time distribution.

Bonus B – Multi-ROI or Batch Inference

• Extend your CLI or scripts to support batch inference across multiple ROIs (e.g., loop over communes).

Required Datasets

Sentinel-2 Data

• Downloaded using PhiDown package

Disturbance Events

• From: <u>European Forest Disturbance Map (web app)</u> and the raw data (<u>Zenodo</u>) Export as polygons with at least: date, type_raw, geometry

In GEE, you can access the data like this:

 $const\ disturbance Agent Data = ee. Image ("projects/ee-albaviana/assets/disturbance_agent_v211"). clip (combined Geometry);$ $const\ disturbance Number Data = ee. Image ("projects/ee-albaviana/assets/number_disturbances_v211"). clip (combined Geometry);$ $const\ disturbance Year Data = ee. Image ("projects/ee-albaviana/assets/latest_disturbance_v211"). clip (combined Geometry);$

Wind Speed

• Use ERA5 via Google Earth Engine for retrieve **maximum wind speed** around disturbance date.

Deliverables

Please share the following in a Git repository (GitHub, GitLab or private link). We let you define your tech choices etc... but we would like to have:

README File

Must include:

- Setup instructions if needed,
- Summary of results (metrics + plots),
- Time spent (estimation appreciated),
- Assumptions / limitations / open questions.

Notebook(s)

- A notebook that demonstrates:
 - o Running inference on the ROI,
 - Visualizing disturbance predictions,
 - o Estimating disturbance timing with ruptures,
 - o Using ERA5 wind speed to separate wind vs beetle damage,
 - o Evaluating precision/recall/lead time.

Time Estimate

- Core tasks: \sim 2 full working days (\sim 16–20h)
- Bonus: 0.5–1 additional day

If you're short on time, focus on reproducibility and pipeline clarity rather than exhaustive evaluation.

Questions?

Feel free to reach out if something is unclear or blocked. We value:

- Clean, modular code,
- Documented assumptions,
- Clear methodology and visuals,
- Reusability over novelty.

We're looking forward to seeing how you tackle this challenge! •

