Prepared by: Shafique AKBARE

Strategy to detect components of Powerline

I. State of the Art

YOLO family models (YOLOv1 to YOLOv11) were pre-trained on the COCO dataset containing 80 annotated classes (person, car, dog, cat, bicycle, etc.) but without anything about powerline infrastructure like pylon, conductor, insulator, or any type of defects of electric towers.

Detectron2 family models (Faster R-CNN, Mask R-CNN, RetinaNet, Cascade R-CNN) were also trained on the COCO dataset containing the same 80 classes annotations and therefore without any powerline-specific classes.

Some sources do mention some models that can detect the basic elements. For example, the model Improved YOLOv5s from a research paper from Cornell University published recently in February 2025 at the link https://arxiv.org/abs/2502.06127 can detect pylons, conductors and insulators but no defects. The model itself, unfortunately, is not provided by the researchers, which would have been good to test for this technical use case.

Moreover, ongoing research at the University of Zurich, where results have not been published yet, mentions at the link https://rre.ethz.ch/research/research-pillars/power-systems/fault-detections-diagnostics-and-prognostics/image-recognition.html that they have a model that can detect pylon, insulator, conductor and a healthy/faulty classification which could be understood as "defects", but again, the model itself is not provided by researchers to test for this technical use case.

Therefore, this use case is unique and there is no other option than training a new model from scratch. The training uses 650 images that were provided.

II. Methodology

To answer this use case, the approach is to train a model with provided images based on a YOLO pre-trained model. The latest is YOLOv11, available since 2024, which has the best performances. As we do not have any annotations provided, it is first necessary to annotate manually some images. The choice is to annotate 50 images out of the 650 images provided (~7% of the total number).

The choice is to use YOLO format for objects detected based on the following classification extracted, which is also compatible with the requirement to have as output the class, score and bbox for each predicted result:

Basic elements:

0: pylon

1: conductor

2: insulator

Defect types:

3: pylon_fissure

Choices made are as follows for this conception:

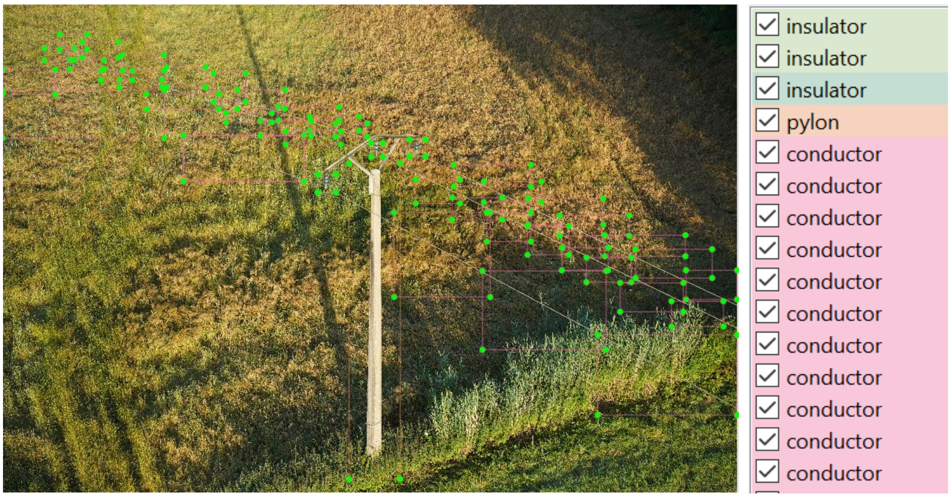
- Pylon definition was not provided in the ERDF classification file; therefore, the pylon is considered to be the main vertical structure of the powerline.
- A confusion exists: the ERDF classification names "isolators" but the use case description mentions it as "insulator". "Insulator" is considered as the correct term.
- The hardest part is labeling conductors as they are diagonal and therefore bounding boxes are harder to draw. The choice has been made to chain multiple bounding boxes for each conductor.
- Exception: in "Copy of B IMG03836.JPG", "Copy of B IMG03914.JPG", "Copy of B IMG03937.JPG", "Copy of B IMG03956.JPG", "Copy of B IMG03975.JPG", "Copy of B IMG04052.JPG", "Copy of B IMG04081.JPG", "Copy of IMG00002.JPG" where two or more conductors are very close, in that case a chained bounding box is done including them. Sometimes it is even the shadow of the conductor which looks like two conductors, like in "Copy of B IMG04029.JPG".
- The choice to predict the defect of fissure on a pylon was made after seeing this defect among the first 50 images: "Copy of B IMG04052.JPG", "Copy of B IMG04125.JPG", "Copy of B IMG04143.JPG", "Copy of B- IMG03805.JPG".
- Exception: "Copy of C IMG03461.JPG", "Copy of C IMG03619.JPG", "Copy of C IMG03620.JPG", "Copy of IMG00001.JPG" containing no powerline.
- Exception: "Copy of IMG00003.JPG" which is a blurred image.

To do so, the free LabelImg package was used.

Examples of visual labeling with the 1st, 2nd and 3rd images provided out of the 650 images:

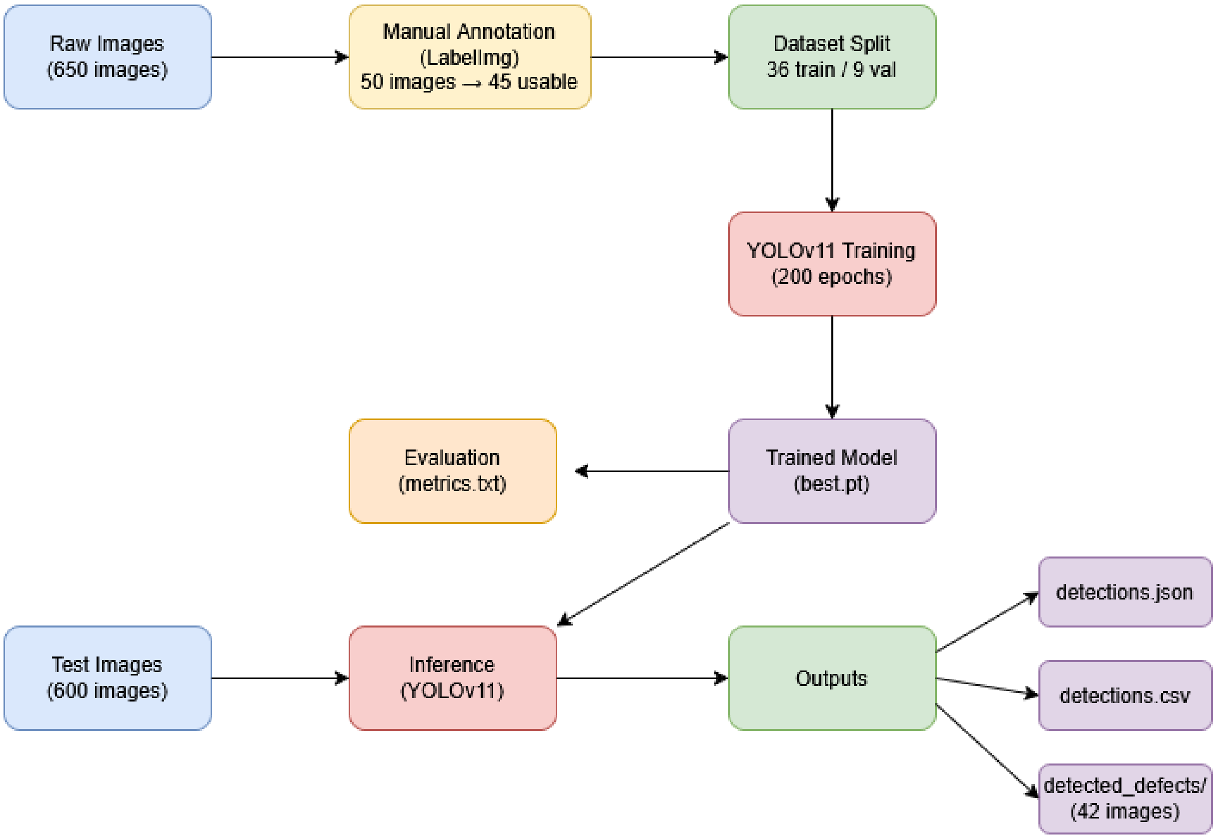
- "Copy of B IMG03400.JPG"
- "Copy of B IMG03411.JPG"

- "Copy of B IMG03415.JPG"



III. Tech Choice

Workflows:



Model type:

Base model: YOLOv11n (nano version)

Pre-trained weights: COCO dataset

Training epochs: 200

Image size: 640x640

Batch size: 16

Evaluation Metrics and Results:

Standard YOLO evaluation metrics (following COCO evaluation protocol):

Precision: Proportion of positive detections that are correct

Recall: Proportion of actual positives that were detected

mAP50: Mean Average Precision at Intersection over Union threshold 0.50

mAP50-95: Mean Average Precision averaged over Intersection over Union thresholds 0.50 to 0.95

Intersection over Union: Measures how well a predicted bounding box matches the ground truth box. Calculated as: (area of overlap) / (area of union). Range: 0 (no overlap) to 1 (perfect match). Intersection over Union 0.50 means the boxes must overlap by at least 50% to be considered a correct detection.

Best mAP50: 53.87% (epoch 131)

Final Precision: 45.69%

Final Recall: 51.77%

Final mAP50: 44.73%

Inference Configuration:

Confidence threshold:

5% (reduced from default 25% for YOLO type of model due to small training dataset)

Defect detection:

42 images with pylon_fissure were detected (out of 600 test images), but most are false positives because some pylons have different forms like holes and stackable structures, and the model was not trained on those.

Two real pylon defects were detected: "Copy of IMG03448.JPG" and "Copy of IMG03422.JPG".

IV Current Limitations:

- Small dataset: Only 45 annotated images (7% of available images)
- Ambiguous cases like "Copy of B IMG03683.JPG": Does it have a burnt cable or just a shadow?
- Ambiguous cases like "Copy of IMG03448.JPG", "Copy of IMG03422.JPG": Do they really have a visible fissure in the lower part of the pylon?
- Low confidence threshold: 5% threshold indicates model uncertainty ideally all 650 should have been annotated for this technical use case
- Subclassification: Add subclassification of fissure types which can be possible if more pool of images with fissures

- More defect types: Add other defect types from ERDF classification

V Conclusion

- The current implementation successfully detects basic powerline components (pylon, conductor, insulator) and one type of defect (pylon_fissure) using YOLOv11. However, to improve accuracy and reliability, it is recommended to annotate at least 650 images with all types of defects, including subclassification of fissure types.
- Maximum scores for the other classes reached for pylon 97.61%, for conductor 96.87% and for insulator 96.86%
- Average score for pylon reached 45.41%
- The model shows promising results with a best mAP50 of 53.87%, demonstrating the feasibility of the approach despite the limited training dataset.
- For the future Data Platform, I propose the following storage architecture: all camera images stored in S3 for raw and processed images, geometries of real life assets of pylons and conductors (and rest of ERDF list) in a Amazon RDS for PostgreSQL database with PostGIS extension to enable computing scripts to retrieve useful insights through spatial queries, time series of detected defects for all types tracked over time in a Amazon RDS for PostgreSQL database with extension TimescaleDB, and YOLO annotations stored in S3 alongside images in the same bucket structure for efficient dataset loading during training. Storage capacity will scale with the number of flights, requiring lifecycle policies and tiered storage strategies to manage costs as data volume increases.