

Détection d'anomalies dans des vidéos acquises par drone pour la maintenance préventive de lignes électriques

Guillaume Fourret^{1,3} Christophe Fiorio¹ Gérard Subsol¹ Marc Chaumont^{1,2} Samuel Brau³

¹ Équipe ICAR, LIRMM, Univ. Montpellier, CNRS, Montpellier, France

² Univ. Nîmes, Nîmes, France

³ Drone Geofencing, Nîmes, France

{guillaume.fourret, fiorio, gerard.subsol, marc.chaumont}@lirmm.fr

samuel.brau@drone-geofencing.com

Résumé

Les lignes électriques sont constituées de plusieurs composants susceptibles de se détériorer au fur et à mesure de leurs utilisations. Pour détecter des possibles problèmes sur ces objets et ainsi prévenir des pannes coûteuses sur le réseau, les drones aériens sont de plus en plus utilisés car ils permettent d'inspecter rapidement de grandes distances, et d'avoir un bon angle de vue sur les différents composants et leurs défauts. Cependant, l'analyse des vidéos de vol par des experts est laborieuse. De plus, le nombre d'anomalies différentes pouvant provoquer une coupure du réseau est grand et faire une solution automatique dédiée pour chacune n'est pas envisageable, notamment à cause du faible nombre d'exemples pour chacune d'entre elles. De ce fait, nous nous sommes tournés vers une solution non supervisée et générique permettant d'attribuer un score d'anomalie aux objets sans nécessiter d'informations a priori sur l'apparence des défauts.

Mots clefs

Détection d'objets, Détection d'anomalies, Deep Learning, Drone aérien, Vidéo.

1 Introduction

Les coupures dues à des défauts des composants des lignes électriques coûtent très cher aux fournisseurs d'électricité. De nombreuses causes de dysfonctionnement sont possibles, allant de la végétation alentour jusqu'à la rupture des composants. La maintenance préventive est donc essentielle pour ces entreprises afin d'anticiper ces pannes. L'inspection de lignes électriques par drone est utilisée couramment mais elle nécessite une inspection visuelle par des techniciens et pourrait être automatisée [1, 2, 3]. Par exemple, l'analyse automatique d'un des composants principaux, l'isolateur, a été étudiée notamment dans le cas de l'absence d'un disque [4, 5]. Mais ces méthodes sont dédiées à l'analyse d'un objet connu à l'avance et surtout d'une anomalie bien identifiée. Or, une anomalie peut avoir un aspect différent en fonction de la condition

de prise de vue (distance, éclairage...). Des méthodes récentes de détection d'anomalies plus génériques utilisent des réseaux de neurones de classification entraînés sur des très grandes bases de données d'objets présentant ces anomalies [6]. Or, les anomalies sont, par définition, rares et la collecte d'une base de données permettant un entraînement devient une tâche compliquée, c'est pourquoi des solutions non-supervisées ont été proposées. Ces algorithmes s'intéressent à la classification de données extrêmes (aussi appelées "outliers") qui se démarquent par rapport à une distribution que suivent la majorité des données. Par exemple, dans [7], un descripteur local est calculé pour chaque disque d'une chaîne d'isolateurs, et la détection d'une anomalie se fait de manière non supervisée en utilisant le Local Outlier Factor (LOF) pour comparer les descripteurs entre eux.

2 Description de notre méthode

Nous proposons dans cet article un pipeline visant à proposer un score d'anomalie à tous les objets détectés d'une ligne, sans s'intéresser à une anomalie précise.

Pour notre méthode non-supervisée, nous posons l'hypothèse qu'une anomalie est par définition rare. La majorité des objets que nous allons détecter sur une ligne électrique sera donc en bon état, et les anomalies se démarqueront. Notre idée générale est donc de comparer après le vol tous les objets d'une même classe entre eux pour voir lesquels se distinguent. Une étape préalable de détection des objets est donc indispensable. Pour cela, nous utilisons un réseau détecteur d'objets afin d'obtenir les imagerie des objets durant le vol. Ensuite, nous utilisons un réseau CNN afin d'extraire un vecteur caractéristique de chaque imagerie, permettant de décrire l'objet de manière efficace. Enfin, nous comparons ces vecteurs pour trouver lesquels se démarquent.

2.1 Détection d'objets

Nous nous sommes tournés vers Yolov5 [8] car c'est un détecteur assez précis, léger et rapide en inférence pour tourner potentiellement sur des cartes embarquées dans les

drones.

2.2 Extraction des vecteurs caractéristiques

Nous calculons pour toutes les imagerie résultantes de l'étape de détection un vecteur caractéristique servant à décrire l'apparence de l'objet. Celui-ci est obtenu en utilisant la dernière couche convolutionnel d'un réseau de neurones VGG16 [9], pré-entraîné sur ImageNet. Les réseaux ResNet [10] et EfficientNet [11] ont également été testés comme extracteur de caractéristiques, mais VGG16 a été retenu car il nous donne expérimentalement les meilleurs résultats de discrimination des anomalies.

2.3 Score d'anomalie

À partir de la séquence de vol, nous obtenons une série de vecteurs caractéristiques \mathbf{X}_i extraits des imagerie venant de Yolo. Certains de ces vecteurs représentent potentiellement un objet présentant une anomalie. Nous utilisons le Local Outlier Factor [12] (LOF) comme algorithme de détection d'anomalies non supervisé.

Pour cela, on utilise la notion de "densité local atteignable" (local reachability density) par rapport à ses k plus proches voisins définie comme suit (1) :

$$lrd_k(\mathbf{X}_i) = 1 / \left(\frac{\sum_{\mathbf{X}_j \in N_k(\mathbf{X}_i)} d_k(\mathbf{X}_i, \mathbf{X}_j)}{|N_k(\mathbf{X}_i)|} \right) \quad (1)$$

$d_k(\mathbf{X}_i, \mathbf{X}_j)$ est la reachability-distance qui représente le maximum entre la distance euclidienne de \mathbf{X}_i et \mathbf{X}_j , et la distance euclidienne entre \mathbf{X}_j et son k -ème plus proche voisin. $N_k(\mathbf{X}_i)$ représente l'ensemble des k plus proches voisins de \mathbf{X}_i .

A partir de cette densité local atteignable, nous pouvons calculer le Local Outlier Factor d'un vecteur (2) qui correspond au ratio entre la densité local atteignable de \mathbf{X}_i et celle de ses voisins dans $N_k(\mathbf{X}_i)$.

$$LOF_k(\mathbf{X}_i) = \frac{\sum_{\mathbf{X}_j \in N_k(\mathbf{X}_i)} lrd_k(\mathbf{X}_j)}{|N_k(\mathbf{X}_i)| \times lrd_k(\mathbf{X}_i)} \quad (2)$$

Cet algorithme sert à donner un score d'anomalie en fonction de la densité local d'un vecteur. Plus un vecteur se retrouve isolé (peu de voisins et/ou voisins éloignés), plus ce score LOF sera éloigné des autres vecteurs.

3 Expérimentation et résultats

Pour la 1^{re} étape de détection, nous avons sélectionné le modèle Yolov5m6, que nous avons entraîné sur 143 images d'isolateurs basse tension en utilisant une carte GPU Nvidia RTX3070 Laptop. Ces images de résolution 4096*2160 proviennent de 2 vidéos de missions de maintenance préventive par drone prises, en conditions réelles.

Les hyperparamètres et les poids utilisés sont ceux du pré-entraînement sur ImageNet réalisé par Ultralytics¹. Nous avons fixé un batch size de 4, une résolution d'image de 1280*720, et l'entraînement dure 85 époques. La figure 1

présente des résultats de ces détections sur une image de test.



FIGURE 1 – Exemple de détection d'isolateurs par Yolov5m6.

Pour avoir une meilleur idée de la qualité de la détection de Yolov5m6, nous avons créé une interface graphique permettant de voir facilement les résultats sur de nouvelles vidéos de tests, comme le montre la figure 2. Le graphe affiché montre le nombre d'isolateurs détectés à chaque image de la vidéo. On peut donc voir sur une vidéo où sont les moments d'intérêts (ici le drone a survolé 3 poteaux qui supportaient respectivement 3,7 et 3 isolateurs, et perdait de vue 2 isolateurs pendant quelques secondes sur le 2ème poteau).

Nous avons utilisé des algorithmes de suivi comme Deepsort [13] et Strongsort [14]. Ces algorithmes permettent de suivre et d'assigner un id à chaque objet détecté dans une séquence d'images, en se basant sur une estimation de leur mouvement par filtre de Kalman et sur la ressemblance de texture grâce à une corrélation entre des vecteurs caractéristiques calculés sur les imagerie. Cela nous a permis d'améliorer la détection d'isolateur en baissant le seuil de confiance à partir duquel nous considérons les détections de Yolo comme valides. Nous filtrons ensuite les potentiels faux positifs créés, en prenant en compte uniquement les objets suivis apparaissant sur plus de 15 images d'affilés (voir figure 3).

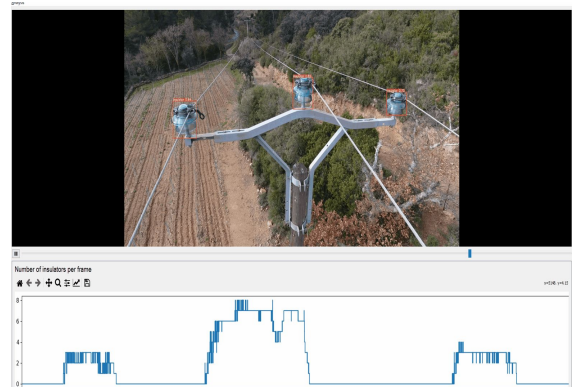


FIGURE 2 – GUI affichant les résultats de l'analyse d'un drone survolant 3 poteaux, supportant respectivement 3,7, et 3 isolateurs. Le graphe montre en abscisse la timeline de la vidéo et en ordonnée le nombre d'objets détectés.

1. <https://github.com/ultralytics/yolov5>



FIGURE 3 – Graphe de détection filtré par tracking. Les "tracklets" (ensemble des bounding box d'un objet identifié) apparaissant moins de 15 frames sont considérées comme des faux positifs et ne sont donc pas pris en compte. On voit que par rapport à la Figure 2, le nombre d'isolateurs détecté est plus stable lors du survol des poteaux.

Nous effectuons ensuite l'extraction de vecteurs caractéristiques par VGG16 et le calcul du LOF.

Nous avons lancé cette chaîne de traitement sur une vidéo de vol fournie par Enedis, où un isolateur présente une fissure. Nous avons représenté sur le graphe de la figure 4 en rouge les scores d'anomalies correspondant aux images de cet isolateur défectueux. On remarque donc que les scores des images de cet isolateur font partie des plus distincts et se démarquent. A noter que les valeurs négatives de LOF sur les graphes viennent de l'implémentation par sklearn qui donne l'opposé de la valeur de LOF. De plus, les scores sont triés par ordre décroissant (et non pas par ordre d'apparition des objets dans la vidéo) afin de repérer plus facilement les objets se démarquant.

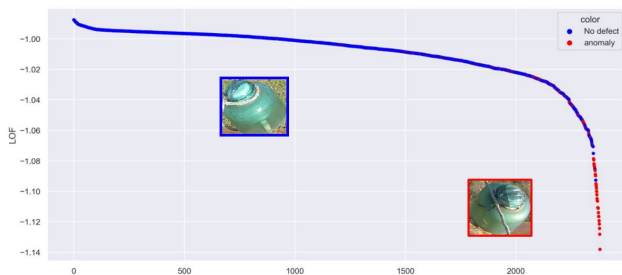


FIGURE 4 – Score LOF ($k=400$) associé aux imagettes des isolateurs détectés par YOLO pendant un vol. Les imagettes sont triées par ordre décroissant de leur score d'anomalie.

Pour tester l'aspect générique de la méthode, les figures 5 et 6 montrent les résultats de tests sur des poteaux en bois dont certains étaient rongés par les termites, et sur des poteaux en acier dont le couvercle avait disparu². On remarque que ces objets en défauts se retrouvent également avec des scores LOF se démarquant des objets sans défauts.

2. N'ayant pas de base de données pour entraîner le réseau à les détecter, les imagettes utilisées ont été découpées manuellement depuis les vidéos de vol.

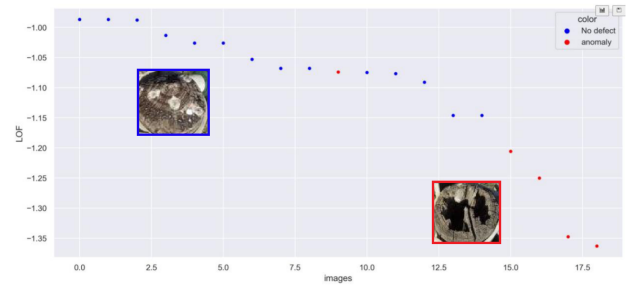


FIGURE 5 – Score LOF ($k=2$) pour des poteaux en bois, trié par ordre décroissant. Les poteaux rongés ayant une texture différente, ils obtiennent globalement un score d'anomalie différent.

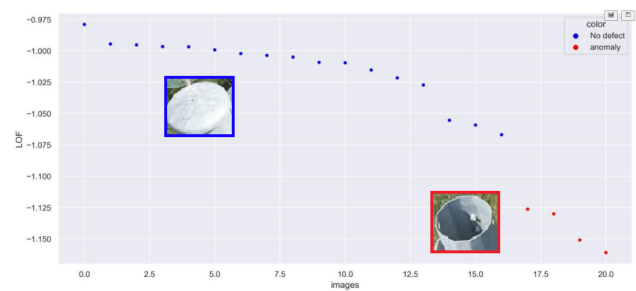


FIGURE 6 – Score LOF ($k=2$) pour des poteaux en acier, trié par ordre décroissant. L'absence de couvercle sur certains poteaux affectant beaucoup l'aspect visuel, leurs scores d'anomalies se démarquent beaucoup.

Nous avons implémenté cette fonctionnalité dans notre interface graphique afin de faciliter l'analyse des résultats (figure 7). L'interface affiche pour chaque id créé par le tracking les frames où l'objet identifié apparait durant la vidéo. La couleur (du vert au rouge) dépend de la valeur du score d'anomalie calculée par le Local Outlier Factor.



FIGURE 7 – En abscisse on présente la "timeline" de la vidéo. Chaque objet détecté est représenté par une barre colorée en fonction de sa normalité (vert=normal, rouge=anormal). L'utilisateur peut alors cliquer sur la barre pour visualiser l'image représentative de la tracklet considérée afin de l'étudier.

4 Conclusion et perspectives

Nous avons proposé une méthode d'analyse de défauts visuels générique. Nous collaborons actuellement avec Enedis afin de pouvoir récolter de plus amples exemples d'anomalies dans le but de pouvoir évaluer correctement l'efficacité de notre méthode, cependant de par la rareté des anomalies, cette base est difficile à constituer. De plus, cela nous permettrait de quantifier l'impact du choix de la valeur de l'hyperparamètre k du LOF. L'introduction du Self-Supervised Learning (SSL) dans le domaine de la détection d'anomalie est également une piste prometteuse [15].

La première étape de détection d'objets par Yolo n'a été entraîné que sur une centaine d'exemples. Afin de rendre toute la méthode le plus générique possible, nous voulons explorer la piste du Few-Shot Learning. Le Few-Shot dans le cadre de la classification est assez répandu, mais beaucoup moins pour de la détection [16] à cause de la difficulté supplémentaire de devoir localiser l'objet parmi beaucoup de zones potentielles. Enfin cet outil reste une aide de diagnostic pour les techniciens qui doivent valider ou non les résultats de la chaîne de traitement. Nous voulons donc aussi nous pencher sur l'active/incremental learning afin de prendre en compte les retours des experts pour améliorer continuellement notre modèle [17].

Remerciements

Nous remercions l'Association Nationale de la Recherche et de la Technologie ainsi que Drone Geofencing pour le financement de la thèse CIFRE. Nous remercions également Enedis pour ses vidéos de vol.

Références

- [1] Van Nhan Nguyen, Robert Jenssen, et Davide Roverso. Automatic autonomous vision-based power line inspection : A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*, 99 :107–120, 2018.
- [2] Xinyu Liu, Xiren Miao, et al. Data analysis in visual power line inspection : An in-depth review of deep learning for component detection and fault diagnosis. *Annual Reviews in Control*, 50 :253–277, 2020.
- [3] Diana Sadykova, Damira Pernebayeva, Mehdi Bagheri, et Alex James. IN-YOLO : Real-time detection of outdoor high voltage insulators using UAV imaging. *IEEE Transactions on Power Delivery*, 35(3) :1599–1601, 2019.
- [4] Xian Tao, Dapeng Zhang, Zihao Wang, et al. Detection of Power Line Insulator Defects Using Aerial Images Analyzed With Convolutional Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, 2018.
- [5] Xuefeng Li, Hansong Su, et Gaohua Liu. Insulator defect recognition based on global detection and local segmentation. *IEEE Access*, 8 :59934–59946, 2020.
- [6] Van Nhan Nguyen, Robert Jenssen, et Davide Roverso. Intelligent Monitoring and Inspection of Power Line Components Powered by UAVs and Deep Learning. *IEEE Power and Energy Technology Systems Journal*, 6(1) :11–21, 2019.
- [7] Markus Oberweger, Andreas Wendel, et Horst Bischof. Visual recognition and fault detection for power line insulators. Dans *Proc. 19th Comput. Vis. Winter Workshop*, pages 1–8, 2014.
- [8] Glenn Jocher, Ayush Chaurasia, et Alex Stoken et al. ultralytics/yolov5 : v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, Février 2022.
- [9] K Simonyan et A Zisserman. Very deep convolutional networks for large-scale image recognition. pages 1–14. Computational and Biological Learning Society, 2015.
- [10] Kaiming He, Xiangyu Zhang, et al. Deep Residual Learning for Image Recognition. Dans *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [11] Mingxing Tan et Quoc Le. EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. Dans Kamalika Chaudhuri et Ruslan Salakhutdinov, éditeurs, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.
- [12] Markus M. Breunig, Hans-Peter Kriegel, et al. LOF : Identifying Density-Based Local Outliers. *SIGMOD Rec.*, 29(2) :93–104, may 2000.
- [13] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, et Ben Upcroft. Simple online and realtime tracking. Dans *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, Sep. 2016.
- [14] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, et Hongying Meng. StrongSORT : Make DeepSORT Great Again. *IEEE Transactions on Multimedia*, pages 1–14, 2023.
- [15] Loïc Jézéquel, Ngoc-Son Vu, Jean Beaudet, et Ayméric Histace. Anomaly Detection via Learnable Pretext Task. Dans *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1178–1185, 2022.
- [16] Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen, et al. Few-shot object detection via baby learning. *Image and Vision Computing*, 120 :104398, 2022.
- [17] Daochen Zha, Kwei-Herng Lai, Mingyang Wan, et al. Meta-AAD : Active Anomaly Detection with Deep Reinforcement Learning. Dans Claudia Plant, Haixun Wang, Alfredo Cuzzocrea, Carlo Zaniolo, et Xindong Wu, éditeurs, *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 771–780. IEEE, 2020.