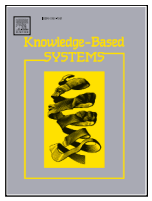
Knowledge-Based Systems 336 (2026) 115355

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



An annotation assistant for monitoring the electrical grid using aerial images

Cristina Benlliure-Jimenez [a,b], Adrian Penate-Sanchez [a], Javier Lorenzo-Navarro [a], Modesto Castrillón-Santana [a], Francisco Mario Hernandez-Tejera [a]

[a] University Institute SIANI, University of Las Palmas de Gran Canaria, Edificio Central del Parque Científico y Tecnológico, Campus Universitario de Tafira, Las Palmas de Gran Canaria, 35017, Las Palmas, Spain
[b] Department of Computer Science, Aerolaser System S.L., C. el Cincel, 2, Agüimes, 35118, Las Palmas, Spain

ARTICLE INFO

Keywords:
Electrical grid inspection
Aerial images
Efficiency improvement
Deep learning techniques
Robotic systems
Large-scale data
Annotation suggestions

ABSTRACT

Monitoring the electrical grid is essential to ensure reliable service and prevent accidents. This supervision is performed by aerial vehicles for image collection; later, these collected images are processed and analyzed by expert annotators. Due to the high costs of manually handling such as large datasets, we present a novel hybrid methodology that leverages deep learning to reduce and optimize annotation workload. The approach uses annotator-provided labels to train a neural network that makes annotation suggestions and gradually reduces the manual workload. Our work is closely related to active learning, but with a key difference: all data must be labeled and verified to guarantee correctness. Therefore, our methodology focuses on reducing the annotation time rather maximizing model performance. Our hybrid method assists annotators by suggesting annotations on high-confidence images that only need verification instead of being created from scratch. Using the proposed approach, annotators can complete their task at least 2.67x faster than with the previous fully manual labeling procedure.

1. Introduction

Ensuring the proper functioning of the electrical grid is a critical task for any modern country. Robotic systems have the potential to improve inspection efficiency and reduce costs, but security and legal restrictions currently prevent the use of fully autonomous drones. Instead, aerial inspections are performed with helicopters or manually operated drones, producing thousands of high-resolution images that must be annotated to generate reports for each tower. This annotation step is highly time-consuming and remains a bottleneck in the inspection pipeline.

The main objective of this work is to reduce the time required for manual inspection and annotation of power line images. To this end, we propose an Artificial Intelligence (AI) assistant that integrates human-in-the-loop verification and annotation into the supervision pipeline. Our approach combines automatic predictions with structured human feedback, so that bounding-box proposals are either verified or corrected by expert annotators. The proposed method, Weighted Image Sampling from Calculated Scores (WISCAS), is a human-in-the-loop annotation strategy designed to optimize the distribution of human effort in safety-critical aerial inspection tasks. Fig. 1 provides an overview of the proposed algorithm.

The key idea is that verification is significantly faster than annotation from scratch, so efficiency can be gained by guiding the annotator to only annotate a subset of images while quickly verifying the rest. The method operates as a data engine that follows an iterative loop in which predictions are generated, a subset of images is selected for verification or annotation, and the model is retrained with the corrected labels to progressively improve its suggestions. Inspired by previous work on combining annotation and verification [1], our approach guarantees that every image is eventually reviewed by an annotator while still reducing overall annotation time.

This principle is particularly important in safety-critical domains such as electrical infrastructure inspection. Overlooking anomalies could result in severe failures or fires. To quantify the efficiency gains of our approach, we analyze annotation times both through direct measurements and simulations, which allow us to estimate the cost savings under different settings.

Our main contributions can be summarized as follows:

E-mail addresses: maria.benlliure101@alu.ulpgc.es (C. Benlliure-Jimenez), adrian.penate@ulpgc.es (A. Penate-Sanchez), javier.lorenzo@ulpgc.es (J. Lorenzo-Navarro), mcastrillon@iusiani.ulpgc.es (M. Castrillón-Santana), mhernandez@iusiani.ulpgc.es (F.M. Hernandez-Tejera).

https://doi.org/10.1016/j.knosys.2026.115355
Received 4 July 2025; Received in revised form 30 December 2025; Accepted 14 January 2026
Available online 16 January 2026

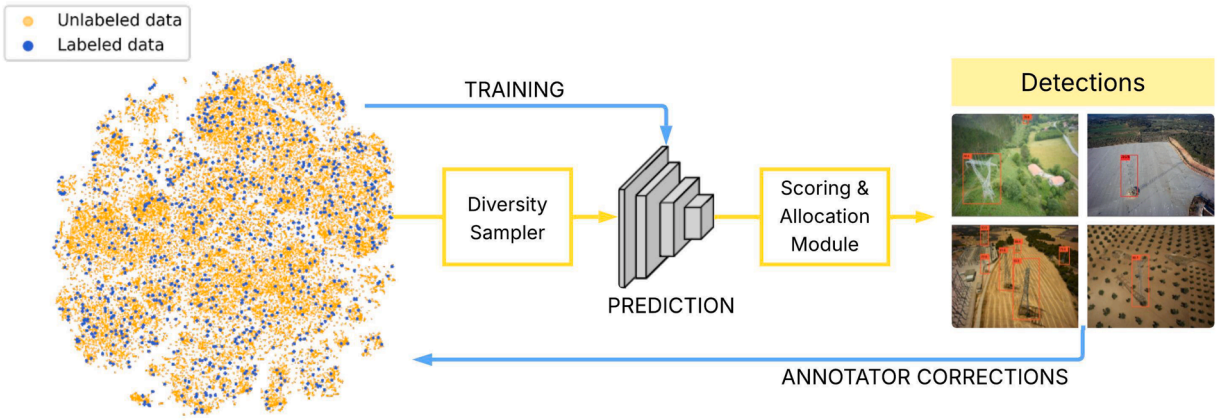


Fig. 1. Overview of the proposed algorithm, which combines automatic processing with annotator interaction. The automatic pipeline includes a Diversity Sampler that pre-selects a subset of unlabeled images, and a Scoring & Allocation Module that refines this subset based on model detections and calculated scores. The annotator then only needs to correct the wrong predictions of the neural network predictor, which greatly reduces time, and afterwards the newly annotated images are added back to the training set for the next loop.

- A hybrid method to save processing time on electric tower annotations while still guaranteeing every image annotation to be correct as expert annotators interact with every single sample.
- A novel method to select the best candidates and to estimate whether to verify more or less electrical towers leads to an even further reduction of annotation time compared to the vanilla version of our approach.
- A thorough validation of the system over a real scenario that includes 530 real electrical lines, 17,517 km of supervised electrical grid, and 60,000 images to validate the performance of the annotator assistant, all this over the whole expanse of a country with great environmental variability.
- The proposed application is already being commercially used in the field. We believe that showing results of industrial systems being used commercially is very important, as our experience and eventualities can be useful to other researchers trying to bring similar applications into production stages.

2. Related work

The monitoring of electrical infrastructures increasingly relies on large-scale image analysis, where robustness and reliability are crucial due to the safety-critical nature of the domain. Aerial imagery collected for inspection captures not only the geometry of electrical components but also diverse textures and environmental conditions, making the construction of representative datasets a demanding task.

In this context, two main research directions are relevant to our work: (i) methods for acquiring and processing aerial imagery of infrastructures, often supported by robotic or aerial platforms, and (ii) machine learning approaches that reduce annotation cost and improve reliability through human-AI collaboration. Within the latter, recent progress in object detection in aerial imagery, active learning strategies, and human-in-the-loop frameworks is particularly related to our contribution.

Recent advances in detection and multimodal fusion architectures [2–5] underscore the need for large annotated datasets, which motivates our focus on efficient annotation methodologies.

2.1. Robotic platforms for electrical grid inspection

Different robotic solutions have been explored to automate the acquisition of inspection data from electrical infrastructures. Aerial platforms such as Unmanned Aerial Vehicles (UAVs) have been equipped with cameras or Light Detection and Ranging (LiDAR) sensors to capture images of towers and power lines, often incorporating vision-based navigation and localization methods [6–10]. Other approaches rely on cable-climbing robots that latch onto power lines to perform close-range inspection [11–14], or ground-based vehicles that mount cameras to circumvent the limitations of UAV endurance and flight regulations [15]. Comprehensive surveys are available in Yang et al. [16], Foudeh et al. [17], Gonçalves and Carvalho [18].

While these systems demonstrate diverse strategies for data acquisition, their connection to our work is limited: we assume that inspection images are already available, and focus instead on methods to support efficient annotation and verification of such imagery through machine learning and human-AI collaboration. In our case, the data were acquired by an industrial partner using a dedicated aerial platform, which we describe in Section 4.1

2.2. Object detection methods in aerial imagery

Object detection in aerial images presents challenges such as scale variation, cluttered backgrounds, and small elongated objects like cables. Convolutional detectors including Faster R-CNN [19], SSD [20], and early You Only Look Once (YOLO) versions have been widely used in inspection tasks. More recent models, such as YOLOv5 or YOLOv8 [21,22], EfficientDet [23], and transformer-based detectors like DEtection TRansformer (DETR) [24], improve robustness in these scenarios.

Complementarily, advances such as Islam et al. [25] show that convolutional neural network (CNN) architectures retain positional cues after global pooling, reinforcing the feasibility of adopting lightweight detectors (e.g., YOLOv5) as reliable drivers of this human-AI annotation cycle.

Recent advances in detection and multimodal fusion architectures have shown substantial gains. Zhang et al. [26] mitigate low-data regimes through language-vision pre-training, whereas Zhang et al. [27] and Guo et al. [28] rely on complex relational modules and distillation to enhance large-scale pipelines. Despite their differences, these approaches all depend on reliable human-labeled data—whether to validate pseudo-labels, train relational modules, or supervise teacher-student transfer. This motivates our complementary focus on *efficient annotation methodologies*, where WISCAS optimizes human effort under the constraint that all images must be verified in safety-critical inspection.

In the context of power-line inspection, these methods have been adapted to detect towers and cables under varying conditions, showing their suitability for pre-annotation in industrial workflows. Our framework is compatible with any modern detector, and YOLOv5 was adopted as a representative baseline in our experiments.

2.3. Public datasets for aerial power-line inspection

Several datasets have been proposed to support object detection research in aerial imagery. Benchmarks captured from drones or low-altitude aerial platforms, such as VisDrone [29] and UAVDT [30], provide annotated images over urban and transportation scenarios. While useful for evaluating general detection methods, these datasets do not include electrical components such as towers or cables, limiting their applicability to our problem domain.

Developing deep learning methods for power line inspection relies heavily on the availability of annotated datasets. Collecting and labeling images of power line components is challenging and costly, often requiring UAVs or helicopters and expert annotators, especially since many components are small and defects are subtle. Additionally, privacy and data protection regulations, such as the General Data Protection Regulation (GDPR) in Europe, limit the public release of data from utility companies.

A few specialized datasets that include towers are publicly available. For instance, TTPLA [31] contains 1234 RGB images with annotations for towers, conductors, and insulators, primarily designed for segmentation tasks, with images captured from heterogeneous viewpoints such as ground-level and drones. Similarly, STN PLAD [32] comprises 2409 high-resolution UAV images annotated for towers, insulators, and dampers, also with a segmentation focus. More recently, the Aerial power infrastructure detection dataset [33] includes 3956 RGB aerial images centered on towers, providing additional segmentation annotations.

In contrast, our dataset consists of over 60,000 high-resolution aerial images captured from helicopters following real inspection protocols, exclusively targeting high-voltage towers across thousands of kilometers. This scale and homogeneity enable realistic simulation of industrial inspection scenarios. While we rely on proprietary data acquisition, our methodology remains dataset-agnostic and could be applied to public datasets or integrated with new detection architectures.

2.4. Machine learning to aid annotation

Reducing annotation effort is a long-standing challenge in computer vision. Classical active learning selects the most informative samples from large unlabeled pools to avoid labeling an entire dataset [34–36]. Recent works refine this idea with more advanced criteria: for instance, Liu et al. [37] select unlabeled samples estimated to exert the highest influence on model performance based on expected gradient calculations. [38] prioritize samples with high misclassification probability in class-imbalanced settings, while Choi et al. [39] estimate separate informativeness scores for classification and localization using dual network heads to better rank unlabeled images.

More recently, global optimization strategies based on meta-heuristic methods have been proposed to address similar selection and parameter tuning problems in complex learning systems. In particular, immune-inspired approaches such as the Immune Plasma Algorithm (IPA) and its extensions formulate data selection and weighting as global optimization problems, enabling adaptive exploration of large and non-convex search spaces. Recent studies have demonstrated the effectiveness of IPA-based methods in optimizing competing objectives and accelerating convergence in high-dimensional engineering problems, including feature selection and path planning, through variants such as multi-population and hybrid immune-plasma formulations [40–42].

Weak supervision offers a complementary perspective, where annotators (i.e., human verifiers in that work)can refine or confirm predictions with minimal input [43], and other approaches incorporate localization-aware scoring to consider both classification and bounding box uncertainty [44]. Large-scale frameworks such as Haussmann et al. [45] extend active learning to object detection in autonomous driving: they focus on selecting informative subsets of unlabeled data, but once selected, all images in the subset must receive exhaustive (full) annotation.

While such meta-heuristic approaches offer strong adaptability, their use often prioritizes fully automated optimization, which may limit transparency and direct human control in safety-critical annotation pipelines. In contrast, WISCAS assumes that *all* images must eventually be human-reviewed due to the safety-critical nature of inspection. The novelty lies in structuring the process into two complementary sets: (i) an *annotation set*, where samples are selected not only based on their informativeness or error likelihood, but also based on their expected annotation cost, balanced across training stages; and (ii) a *verification set*, where samples predicted as correct can be quickly validated without exhaustive labeling. This design ensures complete dataset coverage, while achieving efficiency gains by explicitly optimizing the *distribution* of human effort, rather than merely reducing the number of labeled samples.

These strategies illustrate how machine learning methods aim to reduce annotation costs. In the following subsection, we turn to human-in-the-loop approaches that more directly embed annotator interaction into the annotation process.

2.5. Humans in the loop

As discussed in the previous subsection, many active learning methods aim to improve model performance while reducing the amount of labeled data. In our context, however, *all* samples must eventually be annotated and all annotations must be correct. Thus, the critical metric is not the number of labeled samples, but the time required for humans to generate accurate annotations. Accordingly, the most relevant works to our setting are those that explicitly integrate humans into the loop and focus on accelerating annotation rather than avoiding it.

From this perspective, recent meta-heuristic optimization frameworks can be seen as complementary to human-in-the-loop systems rather than direct replacements. For instance, recent works on global and immune-inspired meta-heuristic planning strategies demonstrate how global optimization can dynamically balance competing objectives and accelerate convergence in complex decision-making problems [46,47]. Integrating such global optimization strategies into human-centered annotation workflows could enable adaptive adjustment of selection criteria over time, while preserving expert oversight, representing a promising direction for future extensions of human-in-the-loop inspection frameworks.

Chen et al. [48] propose to replace the manual drawing of bounding boxes with single-click supervision, reducing effort while retaining annotation quality. Liao et al. [49] provide a systematic ablation study that highlights practical limitations from both the machine's and the annotator's perspectives when scaling dataset creation.

Several works address semantic segmentation annotation by incorporating interactive boundary refinement: Acuna et al. [50] and Ling et al. [51] introduce learning-based boundary suggestions into the annotation interface, while Wang et al. [52] improve on these approaches using energy-based functions that yield higher-quality interactive masks. Shen et al. [53] show that scribble-based interaction can substantially accelerate 3D data annotation, illustrating the benefits of lightweight human input for complex data modalities.

Most relevant to our problem is the methodology of Yi et al. [1], who proposed an annotation strategy for 3D model parts where confident predictions required only verification, while uncertain samples were annotated from scratch. This line of work illustrates the importance of adapting the level of human effort per sample—a principle that directly motivates the design choices presented in the following section.

3. Method

In this section, we introduce in detail the overall procedure that we use to reduce manual annotation time. We will especially focus on the

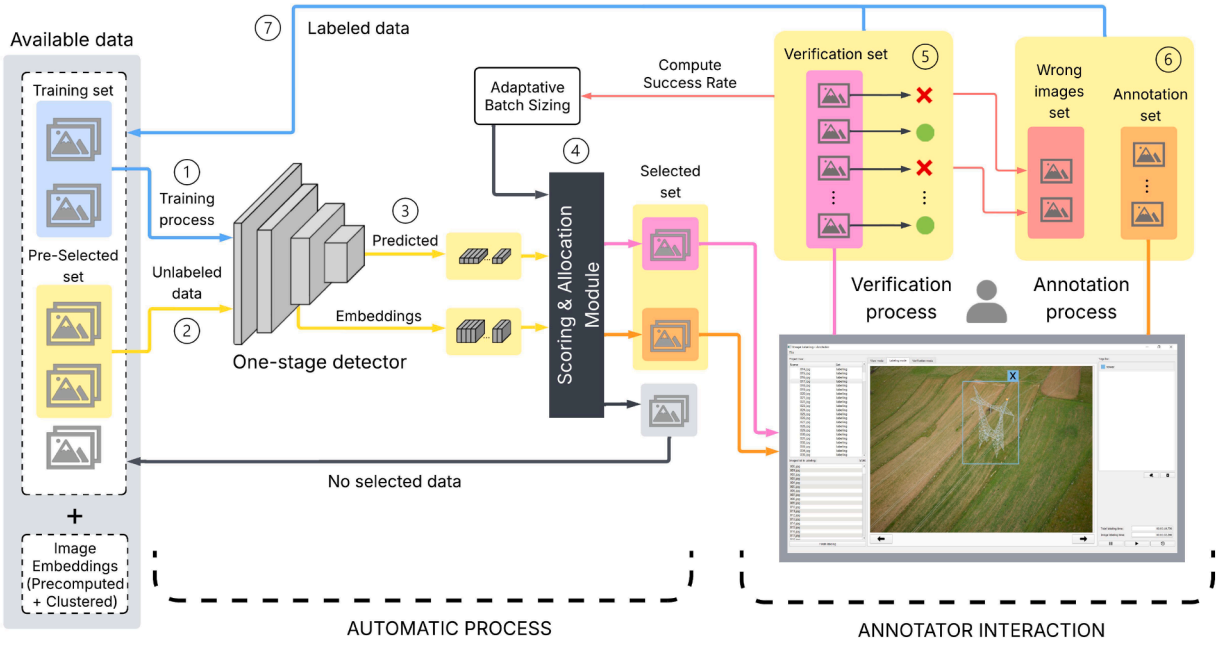


Fig. 2. Overview of the proposed algorithm, which combines automatic processing with annotator interaction. The automatic pipeline consists of: (1) training the model with the labeled data; (2) pre-selecting a subset of the unlabeled data; (3) estimating detections; and (4) scoring and allocating the pre-selected subset by calculated scores to obtain the selected set, while the non-selected images are returned to the unlabeled pool. The annotator then interacts with the system by (5) verifying images in the selected set through Yes/No validation and (6) correcting images in the annotation process. Finally, (7) the newly annotated data are added to the training set for the next loop.

contributions we introduce to what could be considered a fairly standard approach.

The proposed approach is currently being used within the pipeline of our electrical grid surveillance, and real results are shown in this paper over large areas. Our method has been applied to over 530 electrical lines, representing 60,000 aerial images of power lines captured from a helicopter from the Spanish Electrical Grid. Bounding boxes are used to delimit each individual tower in the data. The power lines used in this work extend over 17,517 km and cover a wide range of environments (Section 4.2). In Fig. 6, we can also see the actual spread of all 530 electrical lines that are used in this document over which we flew our helicopter.

3.1. Overall approach

We will first define the data engine workflow designed to perform the annotation of each of the monitored electrical towers. The general idea is that we will first annotate some images to bootstrap a tower detector, and then a detection model will be trained. Once a detector is trained, we can begin to produce suggestions for the human doing the work. The idea is to reduce the amount of time required to annotate each tower by having our model produce bounding box proposals in each new image. Most of these bounding boxes will be correct and contain an electrical tower, and the annotator will only need to validate the proposal instead of producing the annotation from scratch. This iterative process keeps going on until all images are annotated; each loop takes care of 1000 images. By having humans in the loop, we can guarantee that all annotations are correct, which is the fundamental difference between a proposal system like ours and an active learning system. The general steps of the method are shown in Fig. 2 and are summarized as follows:

1. Train a detector with the labeled data.
2. Select a subset of unlabeled images.
3. Predict the pre-selected images with the trained detector model.
4. Score and filter the images with predictions to select and allocate a subset for verification and annotation.
5. Inspect the verification set indicating which image is correct or not.
6. Visualize and edit, if necessary, the images from the annotation set and the images from the verification set with bad predictions.
7. Add the new labeled images to the training set and retrain the model to repeat the process until there are no data left.

We propose using a set of easy images for verification in each loop, along with a set of images to annotate from scratch, to enrich the training dataset of the detection model. Our approach assumes that the detector will easily predict some images, so we aim to process them as quickly as possible. However, it will struggle with others due to underrepresented areas in the current training set. Identifying these challenging images early allows us to enhance the training set, increasing the likelihood that they will be correctly detected in the next iteration. The methodology is fairly straightforward, but several key aspects must be considered to achieve optimal performance.

1. How do we select the images for the next loop? (Section 3.4.1)
2. How many images should we verify and how many should we annotate in each loop? (Section 3.4.2)
3. How do we select which images are good for verification? (Section 3.4.3)
4. How do we find the hard images early to enrich the training set? (Section 3.4.4)

The first question requires a method that enables both the detector and the overall algorithm to succeed. We propose considering image similarity instead of using only random sampling. The answer to the second question will directly impact the number of failed predictions. If we verify too many images, each failure will require human intervention twice: first to reject the incorrect proposal, and then to annotate the image manually. Our goal is to minimize such cases where human involvement is needed twice for a single failed prediction.

The third question is also important because we need a way to estimate whether the detector is likely to produce correct suggestions. This

is challenging, since it involves evaluating all detections within a single image. Finally, the fourth question is critical: we must identify the most valuable images to annotate in order to enrich the training dataset and reduce the number of incorrect proposals in future iterations of the process. The complete workflow including preprocessing and human interaction is summarized in Algorithm 1.

Algorithm 1 Overall iterative annotation process with pre-processing and annotator interaction.

Require: Unlabeled dataset D, initial labeled set L_0, batch size n, detector initialization
Ensure: Fully annotated dataset L

$L \leftarrow L_0$ ▷ Current labeled set with initial annotations
Preprocess all images in D (feature extraction and clustering)
while $D \neq \emptyset$ **do**
  Train detector model M with labeled set L
  Select pre-selected subset $P \subset D$ (e.g., 10,000 images) using clustering/similarity
  Predict bounding boxes with M for all images in P
  Compute sizes of verification n_v and annotation n_a sets ▷ Size of annotation set, ensuring $n_v + n_a = n$
  Select n_v images from P for verification set V (using scoring function)
  Select n_a images from remaining P for annotation set A (using scoring function)
  Remaining images in P are kept for future iterations
  Annotation interaction:
  for each image $i \in V$ **do**
    Annotator marks image as `correct` or `incorrect`
    if `correct` **then**
      $V^+ \leftarrow V^+ \cup \{i\}$
    else
      Move image i to annotation set A
    end if
  end for
  Compute verification success rate: $p^+ \leftarrow |V^+|/|V|$
  for each image $i \in A$ **do**
    Annotator adds/edits bounding boxes (full correction)
  end for
  Update labeled set: $L \leftarrow L \cup V^+ \cup A$
  Remove processed images from D
end while
return Fully annotated dataset L

3.2. Deep learning detector

Our framework is detector-agnostic: any modern object detector can be integrated to generate tower proposals that are later verified by expert annotators. In practice, a single-stage detector is preferable due to the need for real-time interaction during annotation. For our experiments, we adopted YOLOv5 [21] as a representative baseline, given its balance of accuracy and inference speed. A detailed description of the architecture and training setup is provided in Section 5.

3.3. Annotator interaction process

As shown in Fig. 2, the proposed method has two main components. The first is automatic and involves training the detector and selecting images, some for verification and others for manual annotation. The second component involves human interaction: verifying the proposed bounding boxes and annotating a small subset of images. To support this, we developed a graphical tool (Section 5.2) that human operators use to perform these tasks. This tool is fully integrated into the training pipeline, enabling continuous learning of the detection model as new annotations are added.

The detector automatically proposes most bounding boxes to minimize human effort, while human annotation focuses on the most challenging images, facilitating automatic detection of similar images in subsequent iterations. Although WISCAS does not explicitly manage human annotation errors, this workflow reduces their likelihood by minimizing repetitive tasks and keeping annotators focused on critical validation.

In the interactive process, the human annotator will have to view two sets of images. The first one has “easy” images, and the annotator indicates if an image is correct or not (verification process); the second one has a small set of images, and the annotator makes changes to the wrong bounding boxes and annotates some new ones (annotation process). The verification/annotation process is done iteratively, as shown in Fig. 2.

Once all images have been checked by the expert annotators, the model is retrained, adding the new annotations to the training dataset, and repeating the process until all images have been processed.

3.3.1. Verification process

The first step for the annotator is to visually verify the images in the verification set. In this task, the annotator decides whether an image is correct or not (Yes/No). An image is considered correct only if all proposed detections are accurate. Otherwise, it is labeled as incorrect if it contains any false negative or false positive detections. Additionally, the annotator has the option to reject the proposal if any bounding box is deemed imperfect.

Yes Verification: The image is completely correct.

No Verification: The image requires modifications that will be made in the annotation process.

3.3.2. Annotation process

During the annotation process, the annotator reviews the annotation set, which consists of the images rejected during verification and those selected by our approach as informative for enriching the training dataset. The annotator can add, delete, or modify detections as needed. Once the annotation process is complete, the number of detections kept, added, modified, and deleted per image is recorded, along with the total time spent per image (see Section 3.5). This set can also contain images without detections; in fact, the scoring function explicitly considers these cases, as images with no detections are usually cheaper to annotate and are therefore often selected into the annotation set.

3.4. Weighted image sampling from calculated scores, WISCAS

The data engine component of our approach tries to reduce human annotation time by choosing a good subset of images that balances difficulty with diversity in labeling. In order to achieve this balance, a data set is pre-selected to be evaluated by our method to obtain scores per image. This subset of images will be sorted with different criteria to get the verification and annotation sets; the union of both is what we define as the selected set. The subset consists of a large set of images for which annotations can be produced reliably by our detector (the verification set) and another much smaller set of images that need to be very informative to enrich the training set of images and allow the detector to get better in each loop (the annotation set). Finally, the training dataset is composed of all previously processed images. Such subsets can be visually identified in Fig. 2 and are defined as follows:

Pre-selected set: is a set of 10,000 images that are chosen in each iteration of our method. These are images that have not been processed yet. How we select these images is one of the contributions of our paper, as explained in Section 3.4.1.

Selected set: This batch of 1000 images is selected from the pre-selected set. That will be split into those that will form the verification set and those that will go to the annotation set. The number of images that will be used in the verification set and the annotation set is explained in Section 3.4.2.

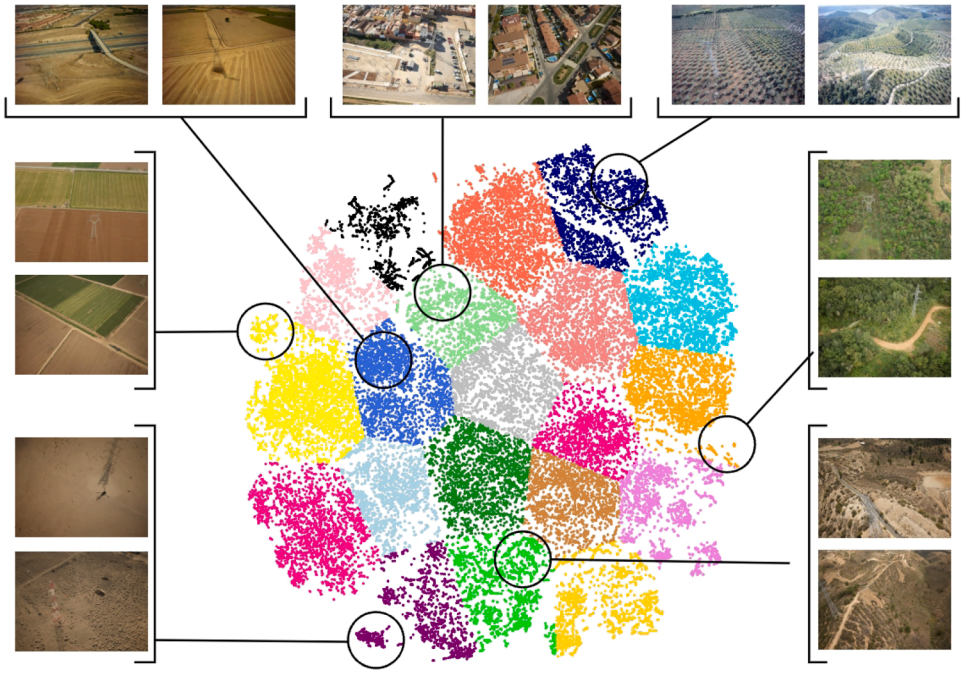


Fig. 3. Latent space representation of the dataset obtained using ResNet50 features and t-SNE dimensionality reduction. Each color indicates a different cluster identified in the embedding space. Example image thumbnails are overlaid in representative regions of the clusters to illustrate the visual content associated with those groups.

Verification set: In this subset of the selected set, the images that have a high probability of being correct are grouped. The verification process is defined as when an expert annotators indicate whether the image is correct or not. If an image has an undetected element or an erroneous detection, the entire image is labeled incorrect and will be introduced into the annotation subset to be corrected, Section 3.4.3.

Annotation set: We use different calculated scores to select images for the annotation set, Section 3.4.4. Additionally, the wrong images from the verification set are added to this subset to correct them. The expert annotator views all the images in this set; the annotator can delete, add a bounding box, or both.

Training set: The training set is the aggregation of all processed images up to the actual step of the annotation process. This data set is used to train the detector that produces annotation suggestions for humans.

3.4.1. *Pre-selecting the images for each loop*

In our algorithm, one of the initial challenges we encountered was determining how to select the images to be annotated in each loop. To this end, we first construct a pre-selected set of 10,000 candidates from the pool of unprocessed images. From this set, 1000 images are then drawn for the annotation process. Initially, we adopted a random sampling approach for unprocessed images, which is commonly used. However, we soon realized that our detector exhibited a bias towards recognizing electrical towers with specific backgrounds or bounding box sizes that were more prevalent in its training data. This bias is a natural outcome, since the detector is trained on a random sample from the entire dataset, leading to these tendencies. However, we believed that it was possible to mitigate this issue by employing a more refined sampling method. Our proposed solution involved incorporating a sampling probability that retained an element of randomness while also considering the diverse environmental contexts captured in each image.

To capture the environmental context of each image, we acquired an embedding using a pre-trained RESNET50 [54] model, trained on the ImageNet dataset [55]. Embeddings correspond to the output of the global average pooling layer (i.e., the layer immediately preceding the final fully connected classification layer), resulting in a 2048-dimensional feature vector. We perform a reduction in dimensionality on the embedding obtained using t-SNE (t-distributed Stochastic Neighbor Embedding) [56], Fig. 3. This reduction allowed us to map the embeddings to a two-dimensional space. Subsequently, we applied a Gaussian Mixture Model (GMM) clustering technique to these two-dimensional embeddings. This clustering method offers the advantage of modeling soft assignments and generating Gaussian distributions centered around each cluster's centroid.

In our implementation, the GMM was configured with 20 components (full covariance, tol = 10^{-12}, max_iter = 600). This choice reflects the moderate contextual diversity of the dataset: although all images depict the same primary object class (electrical towers), variability arises from background, terrain, illumination, and acquisition geometry. Using 20 components was sufficient to capture these contextual differences without over-segmenting visually similar scenarios. Since clustering is only used to guide probabilistic pre-selection rather than to define hard data partitions, the overall method is not sensitive to this parameter.

For each cluster, we define a probability $p(c)$ that combines a uniform component and a component proportional to the number of available images within that cluster. Specifically, we assign

$$p_u(c) = \frac{0.5}{|\text{clusters}_{\text{avail}}|}, \qquad p_n(c) = \frac{n_{\text{img}}(c)}{n_{\text{img}}(\text{total})} \times 0.5, \tag{1}$$

$$p(c) = p_u(c) + p_n(c). \tag{2}$$

Then, for each image i belonging to cluster c, the probability is distributed equally within the cluster:

$$p(i) = \frac{p(c)}{n_{\text{img}}(c)}. \tag{3}$$

The equal weighting between the uniform and proportional components (0.5/0.5) was selected as a design compromise to balance two competing objectives. The proportional term preserves the empirical data distribution, preventing excessive bias toward rare clusters, while the uniform term increases the probability of sampling underrepresented ones that typically correspond to less frequent environmental

conditions. This choice promotes diversity during the early stages of the annotation process, allowing rare scenarios to be incorporated into the training set sooner, which contributes to improved robustness in later iterations. Since this weighting only affects the probabilistic pre-selection stage and all images are ultimately reviewed by expert annotators, the overall WISCAS framework remains robust to moderate variations of this ratio.

What we achieve with this combined probability is that we guarantee that images from several clusters will have a high probability of being within each pre-selected set of 10,000 images, from which 1000 are finally chosen for annotation. The complete procedure is summarized in Algorithm 2.

Algorithm 2 Sampling 10,000 images (Pre-selected set) with cluster-based probabilities.

Input: Unlabeled dataset D, cluster assignments for all images.
Step 1: Define available images

$$D_{\text{avail}} = \{i \in D \mid i \text{ is unlabeled}\}$$

Step 2: Count images per cluster

$$n_{\text{img}}(c) = |\{i \in D_{\text{avail}} \mid c(i) = c\}|$$

$$n_{\text{img}}(\text{total}) = \sum_{c} n_{\text{img}}(c)$$

Step 3: Compute cluster probabilities (scale 0–1)

$$p_u(c) = \frac{0.5}{|\text{clusters}_{\text{avail}}|} \quad \text{(uniform per cluster)}$$

$$p_n(c) = \frac{n_{\text{img}}(c)}{n_{\text{img}}(\text{total})} \times 0.5 \quad \text{(proportional to cluster size)}$$

$$p(c) = p_u(c) + p_n(c)$$

Step 4: Compute image probabilities within cluster
for each image $i \in$ cluster c **do**

$$p(i) = \frac{p(c)}{n_{\text{img}}(c)}$$

end for
Step 5: Sample 10,000 images
Select up to 10,000 images from D_{avail} (without replacement) using probabilities $p(i)$
Output: Pre-selected set of 10,000 images.

3.4.2. Size of verification and annotation sets

We consider that in each iteration i, we incorporate unlabeled images n into the process ($n = n_v + n_a$). In all of our experiments $n = 1000$, it is also important to mention that verification images and annotation images are chosen separately with a unique performance maximization criterion. We need to determine how many of those will be used in verification and how many in annotation.

The size of both the verification set n_v and the annotation set n_a is based on the accuracy of the detector from the last iteration executed $i - 1$. The success rate of the previous verification set (p^+) is calculated as the percentage of images that were classified as correct ($n_{v^+}^{old}$) over the total size of the verification set (n_v^{old}) for the previous step. These values are then used to determine the sizes of the new verification n_v^{new} and annotation n_a^{new} sets:

$$p^+ = \frac{n_{v^+}^{old}}{n_v^{old}} \tag{4}$$

$$n_v^{new} = \lceil p^+ \cdot n \rceil \tag{5}$$

$$n_a^{new} = n - n_v^{new} \tag{6}$$

where $\lceil \cdot \rceil$ denotes rounding up to the nearest integer.

Once the sizes of the new verification and annotation sets have been established, we present the strategies used to select the best images for verification and the best images for annotation. The complete procedure for selecting and splitting the 1,000-image batch into verification and annotation sets is summarized in Algorithm 3.

3.4.3. Selecting the verification set

First, we run our detector on all images, which requires no annotator interaction. Using the detector's performance information for each image, we decide which images belong to the verification set and which to the annotation set.

For the verification set, we employ a scoring system to identify the top-performing images. This score is computed as the geometric mean of the confidence values associated with the detected bounding boxes in each image (C_u).

$$C_u = \left(\prod_{i=1}^{d} c_i \right)^{\frac{1}{d}} = \sqrt[d]{c_1\, c_2\, c_3 \ldots c_d} \tag{7}$$

Here, d represents the number of detections estimated by the model in an image, and c denotes the confidence value assigned to each detection. Using this approach, we can identify images with the highest overall performance based on the quality of their detected bounding boxes.

By employing the geometric mean, we prioritize images where the detector is consistently confident across all detections. Since the geometric mean is sensitive to low values, a single detection with a low confidence score will heavily penalize the overall score of that image. This behavior is desirable because images with incorrect detections are redirected to the annotation set for correction, thereby reducing the time cost associated with requiring the annotator to perform two actions on those images.

3.4.4. Selecting the annotation set

Selecting the annotation set is a crucial step in our approach, as it determines which subset of data will be presented to the annotator for labeling.

This is done through a scoring process that assigns a relevance score to each image based on multiple factors related to prediction quality and annotation complexity. The result is a sorted list of images, from which the highest-scored n_a samples are selected for manual annotation. This strategy ensures that annotators focus on the most valuable images at each iteration, improving annotation efficiency while supporting progressive model refinement.

The three components used to compute the annotation score were selected to reflect complementary aspects of annotation relevance in a human-in-the-loop setting. The average entropy term (H) captures the detector's uncertainty and prioritizes images where predictions are ambiguous, which are typically more informative for improving the model. The maximum feature distance term (D_{max}) promotes representational diversity by favoring images whose detections span a broader region of the feature space, thereby reducing redundancy in the annotated data. Finally, the bounding box count term ($f(n_{bb})$) acts as a proxy for annotation cost, allowing the system to prioritize images that are informative while remaining manageable for human annotators. Together, these components balance informativeness, diversity, and human effort, which are the key objectives of the WISCAS annotation strategy.

This intuition is formalized through a weighted combination of the three components, yielding the annotation score used to rank images:

$$SCORE = H * \alpha_1 + D_{max} * \alpha_2 + f(n_{bb}) * \alpha_3 \tag{8}$$

The weighting parameters are determined through sensitivity analysis to preserve transparency and controllability in a safety-critical

human-in-the-loop annotation setting, rather than being derived from a fully automated global optimization process.

The weighting of each calculated metric is discussed in Section 6. The values of each α range between 0 and 1, and the sum of α_1, α_2, and α_3 must equal 1.

Average entropy per image (H) measures the uncertainty or randomness of the information detected in an image for our model. Higher average entropy values indicate greater ambiguity or complexity, suggesting that the image may require closer attention during annotation. This metric is computed by first calculating the entropy of each detected object (H_{bb}) using its confidence value (c) and its complement ($1-c$), and then averaging these values across all objects in the image.

$$\rho_c = \{c, 1-c\} \tag{9}$$

$$H_{bb} = -\sum_{j=1}^{2} \left(\rho_{c_j} \cdot \log_2(\rho_{c_j})\right) \tag{10}$$

$$H = \frac{\sum_{i=1}^{n_{bb}} H_{bb}}{n_{bb}} \tag{11}$$

Maximum distance between bounding box center-based features ($D_{\max}$) is another metric used to determine which images are included in the annotation set. During inference, we extract feature embeddings from the internal representation of the model by sampling the latent space at the center point of each predicted bounding box. These embeddings capture both semantic and spatial characteristics of the detected objects.

The feature vector associated with each bounding box is obtained by extracting the detector's intermediate convolutional feature representation at the spatial location corresponding to the bounding box center, yielding a fixed-dimensional embedding. Pairwise distances are computed using the Euclidean metric in this feature space, and the resulting $D_{\max}$ value is normalized by the maximum distance observed across all bounding-box features in the current iteration.

By computing the maximum pairwise distance between these center-based embeddings within a single image, we assess the diversity and dispersion of object representations, an indicator of image complexity. This metric also helps reduce redundancy by avoiding the selection of images with tightly clustered or highly similar detections. As illustrated in Fig. 4, objects with similar features tend to form compact clusters in latent space, enabling us to prioritize more informative samples. To ensure fair weighting, the calculated distance is normalized by the maximum observed distance in the latent space.

The logarithmic score based on the number of bounding boxes is an additional criterion used to manage the complexity of images selected for annotation. This score considers the total number of bounding boxes in an image, helping to prioritize samples with a manageable number of objects. The goal is to avoid overwhelming annotators with overly complex images that could hinder annotation efficiency. Conversely, images with too few objects are also penalized, as they may be more appropriate for verification in subsequent iterations rather than annotation.

The optimal number of bounding boxes per image depends on the complexity of the dataset. In our case, this ideal range lies between 2 and 4. To prioritize images within this range, a dynamic threshold ($n_{bb_{\max}}$) is used in the logarithmic scoring function (Eq. (12)), which favors images falling within the preferred interval. This threshold is recalculated at each iteration based on the subset obtained through weighted sampling. It is always less than μ, but it can be adjusted to fit the specific needs of the annotation task.

$$n_{bb_{max}} = \begin{cases} n_{bb}^{max} & \text{for} \quad n_{bb_{max}} \leq \mu \\ \frac{n_{bb}^{max}}{2} & \text{for} \quad n_{bb_{max}} > \mu \end{cases} \tag{12}$$

In addition to the threshold used in the logarithmic scoring function, we explicitly discard images with an excessively high number of bounding boxes by assigning them a score of zero. This prevents overly complex images from being selected for annotation, reducing cognitive load on annotators.

Conversely, images with no detected bounding boxes are highly valuable for annotation, as they offer a blank canvas where annotators can freely identify and label any objects of interest. These images are never included in the verification set, since no confidence values can be computed. Additionally, if such images indeed contain no objects, they are relatively quick to annotate, further improving annotation efficiency.

$$f(n_{bb}) = \begin{cases} 0.9 & \text{for} \quad n_{bb} = 0 \\ \frac{n_{bb}}{n_{bb}^{max}} * \log(\frac{n_{bb}}{n_{bb}^{max}}) & \text{for} \quad 0 < n_{bb} \leq n_{bb_{max}} \\ 0 & \text{for} \quad n_{bb} > n_{bb_{max}} \end{cases} \tag{13}$$

Algorithm 3 Selection and splitting of 1,000-image batch.

Require: Pre-selected images set, previous iteration verification success rate p^+, weights $\alpha_1, \alpha_2, \alpha_3$
Ensure: Verification set V, Annotation set A

$n \leftarrow 1000$ ▷ Total batch size for this iteration
$n_v \leftarrow \lceil p^+ \cdot n \rceil$ ▷ Size of verification set
$n_a \leftarrow n - n_v$ ▷ Size of annotation set
Run detector on all images in the pre-selected set
for each image i in pre-selected set **do**
    Extract bounding boxes and confidences
    Compute verification score C_u (geometric mean of confidences of image i bounding boxes)
    Compute annotation metrics:
    - H: average entropy of detections
    - $D_{\max}$: max distance between center-based embeddings
    - $f(n_{bb})$: logarithmic score of number of boxes

    Compute annotation score: $SCORE = \alpha_1 H + \alpha_2 D_{\max} + \alpha_3 f(n_{bb})$
end for
$V \leftarrow$ top n_v images ranked by C_u
$A \leftarrow A \cup$ top n_a images (not in V) ranked by $SCORE$
return V, A

Computational complexity

Algorithms 2 and 3 introduce limited computational and memory overhead within the WISCAS pipeline. The initial image-level feature embeddings used for clustering in Algorithm 2 are computed once offline as a preprocessing step. While this stage incurs a one-time cost proportional to the dataset size, it does not affect the iterative human-in-the-loop workflow. The subsequent sampling procedure operates linearly with respect to the number of available unlabeled images, and its memory requirements are also linear.

For Algorithm 3, the dominant computational cost corresponds to detector inference over the pre-selected images. The feature embeddings required to compute $D_{\max}$ are directly extracted from intermediate detector representations during inference and therefore do not require additional forward passes. The computation of annotation metrics scales with the number of detected objects per image, which remains small in the considered inspection scenario, making their contribution negligible compared to neural network inference. Finally, ranking operations scale as $\mathcal{O}(M \log M)$ for a batch of M images. Overall, WISCAS scales efficiently and introduces minimal additional computational and memory overhead beyond standard detector execution.

3.5. Accurate time measurement

To evaluate the impact of time savings on performance improvements, our proposed method must provide precise time estimations. Repeating the full human annotation process for every baseline or new

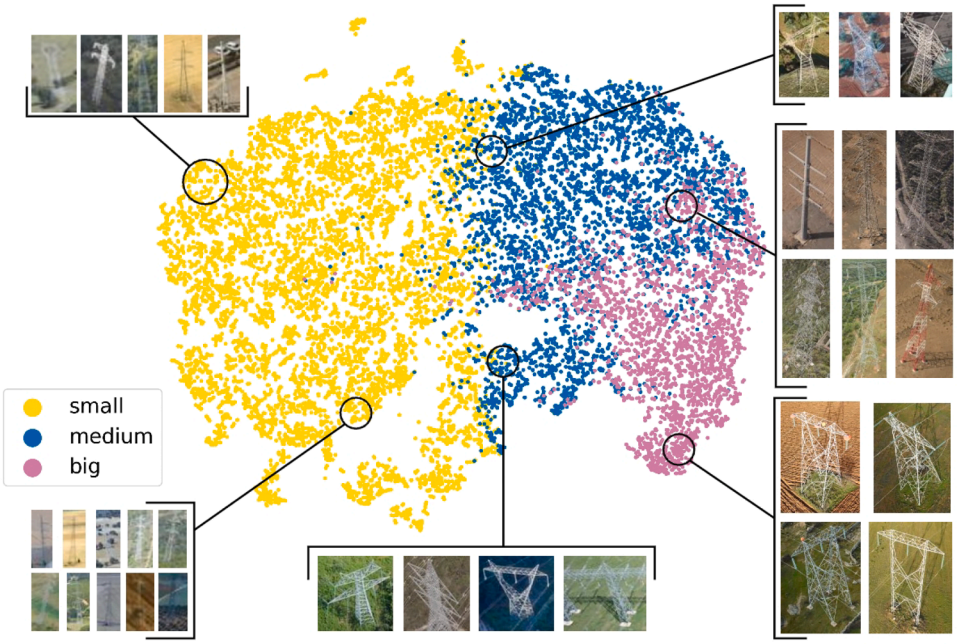


Fig. 4. Latent space of bounding-box center-based features extracted by the detector in a single iteration, using unlabeled images. Each point corresponds to one detected bounding box, and colors indicate the bounding-box size. Example thumbnails of transmission towers are overlaid in representative regions to illustrate the visual content associated with those parts of the feature space. This representation illustrates how the detector organizes object proposals in the feature space according to their spatial scale.

methodology is impractical. Instead, we measure the time required for each type of annotator interaction, successful or unsuccessful, and use these values to estimate the total time per baseline.

To support this, we integrated a timer into the graphical user interface of the annotation tool developed for our annotators, as detailed in Section 5.2. This timer records the duration of each task, allowing us to obtain accurate estimates of global annotation time. Additionally, we modeled the sub-task durations separately, rather than using a single average for annotation and another for verification. This separation is necessary to capture the differences in time requirements across baselines.

The application includes dedicated timers for the verification and annotation interfaces, ensuring that each task type is tracked independently. As expected, the average verification time is shorter than the annotation time, since verification only involves confirming or rejecting existing labels. In the following section, we detail how each sub-task time is calculated for both verification and annotation processes.

To estimate the verification time per image, we define two separate formulas depending on whether the image is correctly annotated or contains errors. These estimations are based on the number of bounding boxes in the image (n_{bb}^{v}), the average time required to verify each bounding box (t_{v}^{bb}), and a constant overhead.

For correctly annotated images, the verification time (t_{v}^{+}) is calculated solely based on n_{bb}^{v} and t_{v}^{bb}. For incorrectly annotated images, an additional term (t_{v}^{p}) is included, representing the average time required for the annotator to detect errors in the bounding boxes. The total verification time in this case is denoted as t_{v}^{-}. Table 1 summarizes the values used for each of these components.

$$t_{v}^{+} = t_{v}^{i} + t_{v}^{bb} * n_{bb}^{v} \tag{14}$$

$$t_{v}^{-} = t_{v}^{i} + t_{v}^{bb} * n_{bb}^{v} + t_{v}^{p} \tag{15}$$

The temporal estimation of the annotation process (t_a) is based on a comprehensive formula formula that considers annotator actions, such as detecting, annotating, deleting, or modifying detections (which involves both deletion and re-annotation). The number of bounding boxes added (n_{bb}^{a}), modified (n_{bb}^{m}), and erased (n_{bb}^{e}) is estimated using the Intersection Over Union (IoU) between the predicted detections and the final annotations provided by the annotator. This allows us to approximate the type and number of interactions required for each image. Table 2 summarizes the values used for each of these components.

Table 1
Variables and measured times used to estimate the verification time per image. The table reports the notation and average duration (in seconds) for: (1) The global verification time per image (t_{v}^{i}), (2) The bounding-box verification time (t_{v}^{bb}), and (3) the penalty time when a correction is required (t_{v}^{p}).

Variables	Notation	Seconds
image global time (verification)	t_{v}^{i}	0.27
bounding box verification time	t_{v}^{bb}	0.14
verification penalty time	t_{v}^{p}	0.61

Table 2
Variables and measured times used to estimate the annotation time per image. The table reports the notation and average duration (in seconds) for: (1) the global annotation time per image (t_{a}^{i}), (2) the bounding-box annotation time (t_{a}^{bb}), (3) the visualization time (t_{a}^{v}), and (4) the annotation erasing time (t_{a}^{e}).

Variables	Notation	Seconds
image global time (annotation)	t_{a}^{i}	0.5
bounding box annotation time	t_{a}^{bb}	1.63
visualization time	t_{a}^{v}	0.34
annotation erasing time	t_{a}^{e}	0.89

$$t_a = t_{a}^{i} + (t_{a}^{bb} * n_{bb}^{a}) + (t_{a}^{v} * n_{bb}^{m}) + (t_{a}^{e} * n_{bb}^{e}) \tag{16}$$

The estimation formulas for the verification and annotation processes are fundamental for predicting the time required to accurately label objects within an image dataset. These estimations allow us to simulate and compare the performance of different labeling strategies, facilitating the selection of the most time-efficient approach for a given

Fig. 5. Hardware setup of the helicopter used for data acquisition and the configuration of sensors and cameras for electrical grid inspection. On the left, the helicopter platform is shown. On the right, the sensor systems are illustrated: (a) sensors mounted on a gimbal system beneath the helicopter, and (b) sensors mounted in a fixed (static) configuration.

dataset. For each baseline, the total annotation time is computed by aggregating the previously defined sub-task durations, resulting in more accurate and reliable time predictions.

4. Robotic platform and data acquisition

In our surveillance pipeline, we work with aerial images of electrical towers over thousands of kilometers. Due to the high costs of our data acquisition situation, there are no public datasets that could be useful for a first approach to artificial intelligence applied to tower detection. For this reason, a pipeline that can transform our daily work into usable data for machine learning solutions was the first requirement that we faced. We will now give further details of the robotic platform that we use and the aerial images that we captured and were used in this work.

4.1. Robotic platform

The standard electrical tower inspection pipeline involves the use of an aircraft, helicopter, or drone to fly over each of the lines that need to be inspected. The aircraft carries a set of mounted sensors that need to be manually pointed toward each electric tower by an operator to be able to generate reports for each electrical tower afterwards; see Fig. 5 for a detailed definition of all sensors. The equipment used for the inspections consists of a LiDAR sensor, an Inertial Measurement Unit (IMU), an infrared (IR) camera, a video camera, and a high-resolution camera of up to 100 megapixels. Other surveillance is also performed in parallel, but in our work, we are focusing on the image reports that are done for each tower.

In our case, data acquisition flights are conducted by a certified helicopter operator authorized by the Spanish Aviation Safety Agency (AESA) for specialized operations (SPO) of power line inspection, which are classified as high-risk operations under Regulation (EU) 965/2012. Each mission follows a risk-assessed and AESA-approved Standard Operating Procedure (SOP). Within this framework, the aircraft typically maintains at least 20 m of separation above and laterally from the power lines as specified in the approved SOP. Final flight parameters are always under the responsibility of the licensed pilot in command, who ensures compliance with safety and regulatory requirements for each area of operation.

Table 3
Distribution of the dataset by geographical region in Spain. For each region, the table reports the total kilometers of power lines inspected, the number of aerial images acquired, and the number of distinct power lines covered. The last row (TOTAL) indicates the overall dataset size: 17,517 km, 60,000 images, and 530 lines.

Region	Kilometers	Images	Lines
Andalucía	2925	5845	62
Aragón	1550	3,941	47
Canarias	538	9573	35
Cantabria	323	716	11
C. La Mancha	3745	11,877	70
Castilla León	1337	9185	25
Catalunya	287	411	6
Extremadura	2296	4606	35
La Rioja	150	438	7
Madrid	1645	6747	108
Murcia	474	1325	12
Navarra	354	830	15
País Vasco	661	1624	41
C. Valenciana	1232	2882	56
TOTAL	17,517	60,000	530

4.2. Electrical tower image dataset

Our aerial images dataset has 60,000 images distributed in 530 different power lines over 17,517 km located in Spain. Table 3 indicates the number of kilometers, total images, and power lines by geographical region. All Spanish regions have data in our dataset, except Asturias, Galicia, Islas Baleares, Ceuta, and Melilla.

As the dataset originates from real inspection flights, incidental captures of private property were possible. In compliance with the EU General Data Protection Regulation (GDPR) and the Spanish Organic Law 3/2018 on Personal Data Protection and Guarantee of Digital Rights (LOPDGDD), all imagery was anonymized before dataset construction, ensuring that the final dataset contains no identifiable personal data.

The dataset was not fully annotated from the beginning; instead, it was initially built by randomly annotating a subset of images to estimate annotation costs and create a first training set. This process was later refined through several iterations that ultimately converged into the WISCAS methodology.

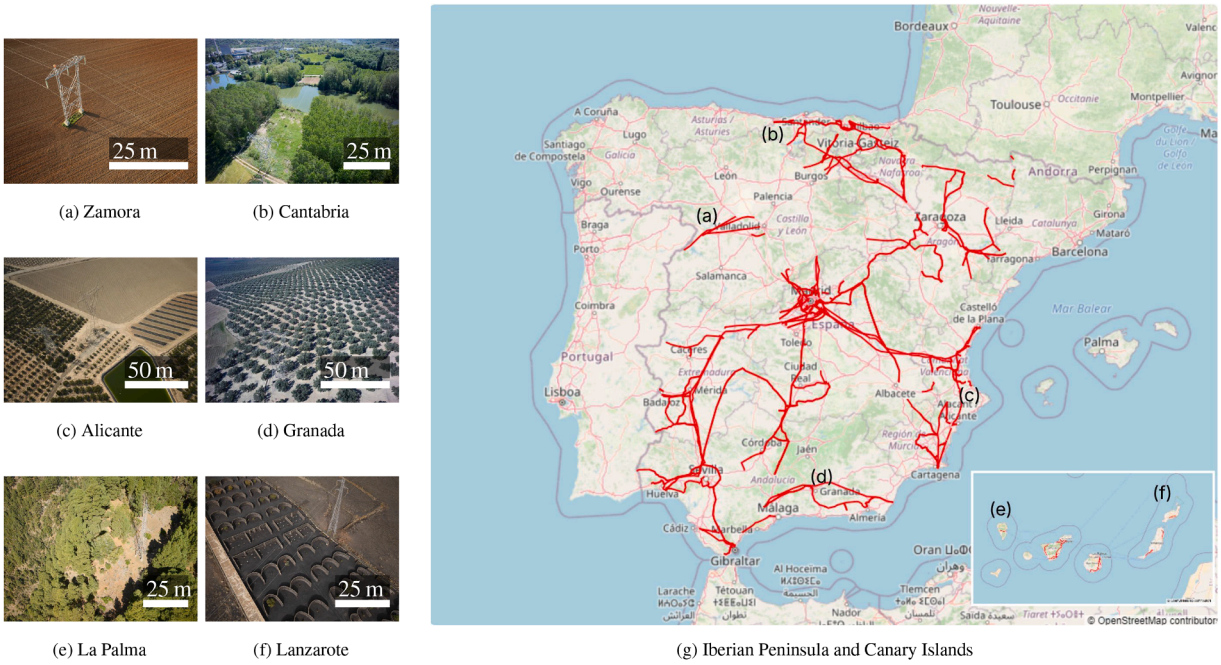


Fig. 6. Examples of aerial images and geographic distribution of the dataset. (a–f) Sample images from different regions of Spain: (a) Zamora, (b) Cantabria, (c) Alicante, (d) Granada, (e) La Palma, and (f) Lanzarote. Scale bars are included in each image for reference: 25 m in (a), (b), (e), and (f), and 50 m in (c) and (d). The scale is approximate due to image perspective and was calculated using LiDAR data from the corresponding circuits. (g) Map of Spain showing in red the power line circuits included in the dataset.

Fig. 6 shows a map of Spain with the power lines to which the towers in our dataset belong. On the map, you can see that we have a large part of the data concentrated in the capital, Madrid, unlike other areas such as Galicia, where we do not have any data. This is because based on the available data, lines have been selected that offer us a small representation of the available data, but there are regions from which the collaborating company has not collected data.

In the captured images, the electrical tower is present in a small portion of the full image. Given the security measures necessary to fly near the line and the structure of the tower itself, which reveals the background of the image, most pixels will belong to the background in such images. Therefore, the influence of landscape patterns, occlusions, and other common factors on image acquisition greatly impacts tower detection. This requires significant effort in modeling electrical towers, considering the structure of the electrical grid and the substantial variability of the background. To ensure accurate detection, we invest considerable effort in training our tower detector with a diverse dataset that covers a wide range of environments. By incorporating these various scenarios, we equip our detector to effectively handle the challenges posed by different landscapes, occlusions, and other typical conditions encountered during image acquisition. In Fig. 6, we can see the high variability that we have been talking about, and in Table 3, the details for each region are shown.

In addition, it must be considered that a greater distance between the selected lines does not necessarily imply a greater diversity between the visual characteristics. For example, in the Canary Islands, we find very close regions but with very different ecosystems that enrich the diversity of our data Fig. 6(e) and (f).

The grid structure of the electrical towers, combined with the diversity of surrounding environments, presents a challenge for accurate reporting of each individual tower. To address this, it is crucial to ensure strong representation in all environmental conditions. By incorporating a broad and diverse set of environments into the training process, we can improve both the accuracy and robustness of the detection system. To this end, we propose using a generic embedding of each image to guide the selection process through clustering. Once clusters of images with similar environmental characteristics are identified, we can ensure that the detector is consistently trained on a representative sample of all relevant scenarios.

5. Implementation details

This section explains specific implementation details to allow us to replicate our work and the details of the user interface developed for the annotation and verification process.

5.1. Detector

For our experiments, we used a YOLOv5 tower detector from the YOLO [57] family of single-stage CNN-based object detectors. Its architecture combines a Cross Stage Partial (CSP) backbone [58], a Spatial Pyramid Pooling (SPP) [59] block with Path Aggregation Network (PANet) [60], and a YOLOv3 detection head with Generalized Intersection over Union loss (GIoU-loss). This configuration provides a good balance between training efficiency and detection accuracy in fine-grained scenarios, making it suitable for large-scale annotation workflows.

We used a PyTorch implementation of the detector to predict bounding boxes and extract features from the images by creating a hook to compute detection center-based features. We selected layer 17 for feature extraction, although we also performed experiments with layers 20 and 23.

As shown in Section 4.1, we use a 100-megapixel camera as our main sensor. Due to hardware limitation issues when training neural networks with large images, we resized our dataset to 640 x 480. Detections in this image size can be directly transformed into the original 100 mp images when further detail is needed. Due to the textures that we find in the aerial images, electrical towers tend to "camouflage" with the environment, making their detection difficult, Section 4.2. To improve the performance of our detector, we have made use of data augmentation

techniques when training our models (shifts, flips, rotations, brightness, zoom, mixup, and mosaic).

5.2. Annotation tool

To facilitate annotator interaction, we developed a Python application with a graphical user interface that enables human annotators to carry out the required tasks. This tool integrates the YOLOv5 detector and manages both the training and inference processes for all images in the dataset. In its current version, the application schedules annotator tasks in fixed daily cycles of 1000 images for the sake of consistency and simplicity in the experimental setup. However, future versions will support a variable number of images per cycle, providing greater flexibility. The training of the updated detector is performed overnight to avoid interrupting annotator workflows.

Verification interface The interface of the created verification process allows you to visualise the current image and indicate with keys or through the graphical interface if it is the correct verification or not. When the annotator processes the image, the next image will automatically be displayed to speed up the process.

Annotation interface The labeling interface allows the annotator to add bounding boxes and erase detections, Fig. 2. Since an image can have more than one annotation, to change the image, it is necessary to use keys or buttons.

6. Experimental results

In this section, we will describe in detail the different tests that were performed using the method presented in this paper. We will establish a set of baselines to assess the effectiveness of our approach in comparison to other alternatives. We have also created a way of simulating the annotation process to allow for comparison between baselines and to be able to develop our approach without the need to re-perform the verification and annotation steps.

6.1. Baselines

The simulated methods are divided into two categories. The first category includes methods that do not use a fixed verification or annotation subset; instead, they divide the n predicted unlabeled samples using a confidence threshold. The second category consists of methods that use a fixed verification and annotation subset size ($n_v + n_a = n$), selecting n images from a large pool of unlabeled predictions based on scoring metrics.

The baselines presented here represent natural approaches to implementing the machine learning-assisted annotation pipeline proposed in this work. The first baseline involves fully manual annotation, where reports for each tower are created entirely by hand. From there, we progressively introduce each of the proposed contributions to demonstrate the incremental improvements achieved through our approach. In the following sections, we provide a detailed description of each baseline.

Manual Annotation: Manual annotation was the original approach used to generate reports for electrical towers in our workflow. The annotators manually created bounding boxes, from which the corresponding reports for each tower were derived. As shown in our results (Fig. 8), this process was highly time-consuming and incurred significant labor costs for the company.

Random Sampling: The first baseline we introduced leverages a trained detector to generate bounding box proposals for images where the model is highly confident in its predictions. In this approach, 1000 unlabeled images are randomly selected from the pool of images pending annotation. For each image, the average confidence is computed based on the confidence scores of the model's detection predictions. If an image's score exceeds a predefined threshold, it is assigned to the verification set; otherwise, it is placed in the annotation set, either due to a lower score or the absence of detections.

Uncertainty Sampling - Full Annotation: To assess the behavior of a classical active learning strategy under the requirement that the full dataset must eventually be reviewed, we implement an uncertainty sampling baseline. At each iteration, all unprocessed images are ranked according to an image-level uncertainty score, and the top $n = 1000$ samples are selected for *full annotation*. The uncertainty score for an image i is defined as $U(i) = 1 - \bar{c}_i$, where $\bar{c}_i$ denotes the mean confidence of all detections in image i. For images where the detector produces no bounding box predictions, we set $U(i) = 1$, corresponding to maximal uncertainty. Unlike the proposed WISCAS approach, this baseline does not include a verification step: all selected images are directly sent to full annotation. The detector is retrained iteratively after each annotation cycle until the unlabeled pool is exhausted.

This baseline follows the standard active learning paradigm and intentionally ignores annotation cost asymmetry and verification mechanisms, allowing us to assess the limitations of uncertainty-only selection in safety-critical annotation settings where full human review is required.

Weighted Sampling: In the weighted sampling baseline, we improve upon the Random Sampling of 1000 unlabeled images by incorporating embedding-based clustering, as described in Section 3.4.1. This approach ensures greater background variability in the training dataset, which helps the detector generalize better across different environments. The classification of images into the verification or annotation set remains the same as in the **Random Sampling** baseline, based on the average confidence per image. As shown in Fig. 8, allowing underrepresented clusters to contribute earlier to the training process leads to a consistent improvement in performance.

6.2. WISCAS

Finally, we fully incorporate our proposed contributions into a final baseline, which we call **WISCAS** (Weighted Image Sampling from Calculated Scores), to facilitate interpretation of the results. Unlike the previously described methods, this approach focuses on selecting higher-quality images for the annotation set. To achieve this, we first apply weighted sampling to select 10,000 images, from which 1000 will be chosen based on calculated scores. We also introduce specific selection strategies for both the verification and annotation subsets, aiming to better address the weaknesses of the detector.

The selection strategy for the verification set remains the same, using confidence scores as a ranking criterion to identify the most reliable detections. In contrast, the annotation set is constructed using three terms, described in Section 3.4.4: a score based on the maximum distance between center-based features of bounding boxes ($D_{\max}$), the average entropy of detections within each image (H), and a logarithmic score based on the number of bounding boxes ($f(n_{bb})$). These components are combined using a weighted scheme and guide the sampling process, as defined in Eq. (8). To illustrate the effect of each individual contribution, we present two additional baselines that isolate and test their impact on the final results.

In **WISCAS v1**, the three metrics used for annotation set selection are assigned equal weights when computing the final image score. In contrast,**WISCAS v2** employs a weighted scheme in which

the average entropy (H) contributes 0.25 (α_1), the maximum distance between bounding box centre-based features ($D_{\max}$) contributes 0.15 (α_2), and the logarithmic score ($f(n_{bb})$) contributes 0.60 (α_3) to the final score.

The weights $\alpha_1, \alpha_2, \alpha_3$ were explored through a sensitivity analysis, with several representative combinations evaluated under the same simulation protocol. Equal weights (0.33/0.33/0.33, WISCAS v1) required 10.72 h of simulated annotation time; (0.25/0.60/0.15) required 10.71 h; (0.25/0.15/0.60) required 10.82 h; and the selected configuration (0.60/0.15/0.25, WISCAS v2) achieved the best result with 10.24 h. The variation across settings was small (≤0.58 h, ∼5%), indicating that WISCAS is not overly sensitive to moderate weight changes, while prioritizing the bounding-box count term yielded the most efficient outcome.

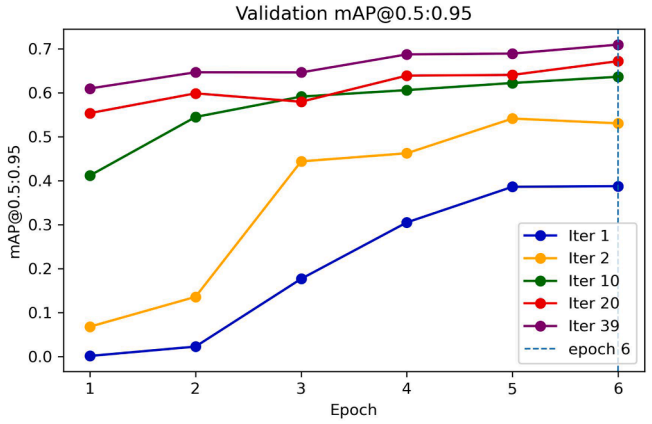


Fig. 7. Validation learning curves (mAP@0.5:0.95) for representative iterations of the process. The y-axis shows the mean Average Precision (mAP), while the x-axis indicates the training epochs. Each curve corresponds to one iteration of the annotation loop (examples shown: Iter 1, 2, 10, 20, and 39). Model performance increases rapidly during the first epochs and stabilizes by epoch 5–6, as indicated by the dashed vertical line. This confirms that limiting training to six epochs per iteration is sufficient for convergence while reducing computational cost.

Although detection accuracy remains virtually identical between WISCAS v1 and v2—as expected since all images are eventually annotated—v2 demonstrates a statistically significant reduction in annotation cost (see Section 6.4).

6.3. Simulated process

Once the dataset was fully annotated, simulations were performed to compare the different annotation approaches explored during dataset construction (see Section 4.2 for details on dataset creation). The fully annotated dataset was divided into three subsets for the simulations: 10,000 images for validation, 10,000 for testing, and 40,000 images to simulate the annotation process. These 40,000 images represent the unlabeled data targeted by our annotation strategies. At the end of the simulation, the model is trained on these annotated images to evaluate final detector performance.

In the first iteration, the sizes of the verification and annotation subsets were fixed to $n_v = 500$ and $n_a = 500$. From the second iteration onwards, n_v and n_a were computed dynamically based on the detector's accuracy in the previous cycle, as described in Section 3.4.2.

For each iteration of the process, we trained the detector for six epochs. Fig. 7 shows the validation mAP@0.5:0.95 across epochs for representative iterations of the process. In all cases, performance rises steeply during the first epochs and stabilizes by epoch 5–6. Training for more epochs yields only marginal improvements ($< \Delta$ mAP), confirming that six epochs suffice for convergence at each iteration while reducing computational cost. Nevertheless, depending on the dataset and task characteristics, the number of epochs required for convergence may need to be re-evaluated.

The simulations automate the unsupervised steps of the pipeline, including model training, predicted label generation, subset selection, and filtering of selected images, while emulating annotator interactions during verification and annotation. Detections are compared with the ground truth to compute Intersection over Union (IoU) values. Each detection is classified as correct ($\alpha \geq 0.6$), requiring modification ($0.3 < \alpha < 0.6$), or a false positive ($\alpha \leq 0.3$), and these classifications are used to estimate annotation time for each approach, as described in Section 3.5.

To assess annotation reliability, a hold-out set of 1000 images was fully annotated independently by two expert annotators. Inter-annotator agreement, summarized in Table 4, includes total boxes per annotator, the percentage of matching boxes (IoU ≥ 0.5), mean IoU of matches, and Cohen's κ (0.874), indicating very high agreement.

Table 4
Inter-annotator agreement statistics for the hold-out set of 1000 images. The table reports: the total number of bounding boxes annotated independently by Annotator A and Annotator B; the number of matching boxes (IoU ≥ 0.5) and the corresponding match percentages with respect to each annotator; the mean IoU of the matching boxes; and Cohen's κ coefficient measuring overall agreement.

Metric	Value
Total boxes Annotator A	1266
Total boxes Annotator B	1333
Total matching boxes (IoU ≥ 0.5)	1133
Percentage of matches (Annotator A)	89.5%
Percentage of matches (Annotator B)	85.0%
Mean IoU of matching boxes	0.758
Cohen's κ	0.874

The simulation outcomes may be inherently biased, as the dataset was constructed iteratively: an initial random subset was labeled to estimate costs, and subsequent refinements were informed by intermediate results that guided the evolution towards the WISCAS methodology. Consequently, simulated annotation times and efficiency gains may overestimate improvements achievable in fully independent scenarios.

WISCAS is intended to optimize the annotation and verification process rather than replace expert annotator supervision; all images must still be reviewed by an annotator, while YOLOv5 prioritizes images to reduce annotation cost without compromising human-level accuracy.

6.4. Annotation efficiency by strategy

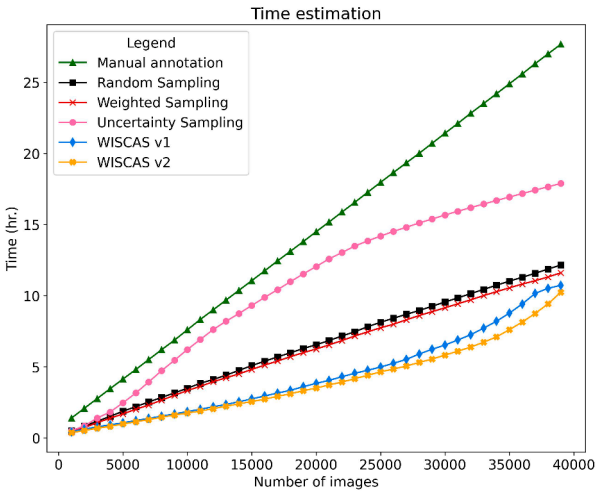
As previously described, the dataset for this project was initially annotated using the **Random Sampling** method. These annotations serve as the ground truth throughout all evaluations presented in this paper. The key metrics for assessing the effectiveness of our annotation strategies are: (1) How much time is required for a human annotator to produce a fixed number of annotations? and (2) How many annotations can a human produce within a fixed time frame?

Our final model is trained using all 40,000 images from the experimental set, achieving a precision of 0.87 and a recall of 0.96, as shown in Table 5. Since the final set of annotated data is the same across all methods, the resulting model performance remains unchanged. However, the total annotation time required to obtain this dataset varies significantly depending on the approach used: 27.68 h for **Manual Annotation**, 12.15 h for **Random Sampling**, 17.89 h for **Uncertainty Sampling**, 11.60 h for **Weighted Sampling**, 10.72 h for **WISCAS v1**, and 10.24 h for **WISCAS v2**.

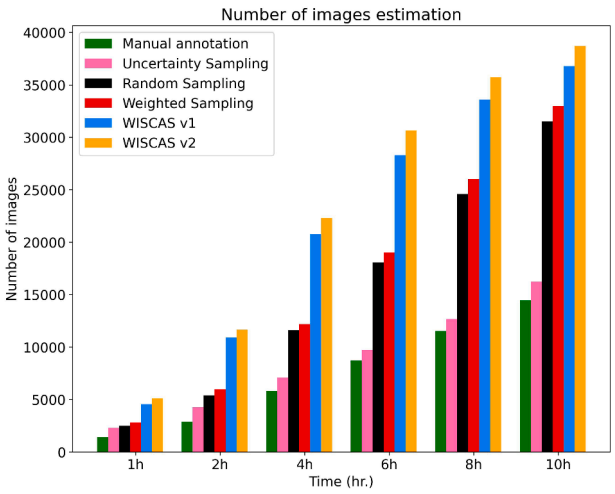
To confirm that the observed reduction in annotation time between WISCAS v1 and v2 is not due to random variation, we performed a paired t-test. The results indicate a t-statistic of 11.31 and a p-value of 1.33×10^{-29}, confirming that the difference of 0.48 h is statistically significant ($p < 0.05$).

These results clearly demonstrate the substantial time savings achieved by integrating AI-assisted tools into the annotation process. Furthermore, the introduction of this tool into the production pipeline has enabled the acquisition of high-quality annotations that can be leveraged for future projects beyond this one.

We now examine in detail the two key metrics identified as relevant to our work, as shown in Fig. 8(a). This figure presents a comparison of all baselines across both metrics. The left plot illustrates the time required to annotate N images, while Fig. 8(b) shows how many images can be annotated within X hours. In both cases, it is evident that all automated approaches significantly outperform manual annotation. Moreover, the WISCAS method, with its intelligent image selection strategy, yields further improvements in annotation efficiency.



(a) Relationship between human interaction time (y-axis, in hours) and the number of annotated images (x-axis). This plot shows how much interaction time is required to annotate a given number of images under different methods.



(b) Number of annotated images (y-axis) obtained for a fixed amount of human interaction time (x-axis, in hours). This bar chart compares the efficiency of the different methods.

Fig. 8. Comparison of annotation strategies in terms of human interaction time and annotation output. Subfigure (a) illustrates the time required to annotate a given number of images, while subfigure (b) shows the number of images annotated within a fixed interaction time. Results are reported for the different methods described in Section 6.1.

Table 5
Comparison of annotation strategies in terms of detection performance and required human interaction time. For each method, the table reports precision, recall, and F1-score, together with the estimated interaction time (in hours) needed to annotate the complete dataset. Manual annotation corresponds to the baseline, while Random Sampling, Uncertainty Sampling, Weighted Sampling, and WISCAS (versions v1 and v2) represent different strategies. Reported times are derived from simulation based on model errors and corrections, and account only for human interaction (excluding automatic computation).

Method	Precision	Recall	F1-Score	Time (h)
Manual Annotation	0.870	0.975	0.919	27.68
Random Sampling	0.867	0.970	0.915	12.15
Uncertainty Sampling	0.869	0.941	0.903	17.89
Weighted Sampling	0.875	0.969	0.919	11.60
WISCAS v1	0.878	0.964	0.919	10.72
WISCAS v2	0.878	0.963	0.918	10.24

When evaluating the time needed to annotate a fixed number of images (Fig. 8(a)), both versions of WISCAS consistently outperform the **Random Sampling**, **Uncertainty Sampling** and **Weighted Sampling** baselines throughout most of the process. However, this performance gap narrows towards the end, as WISCAS intentionally defers more complex images to later stages, leveraging the robustness gained from a stronger training dataset. This strategy results in a final annotation time reduction of approximately one hour compared to the best-performing non-WISCAS baseline, as shown in Table 5.

The final detector exhibits very high recall (0.96) at the expense of a lower precision (0.87), a trade-off intentionally adopted due to the safety-critical nature of electrical grid inspection, where missing a tower is considerably more harmful than detecting additional structures. From the annotator's perspective, this configuration is well aligned with the cost structure of the annotation process: removing false positive detections is significantly faster than creating bounding boxes from scratch, as reflected in the measured interaction times reported in Section 3.5.

Within WISCAS, the impact of reduced precision on verification workload is further mitigated by the selection strategy. Images affected by false positives tend to receive lower verification scores or higher ambiguity measures and are therefore redirected to the annotation set, avoiding repeated verification failures. As a result, annotation efficiency remains high despite the conservative detector configuration.

Table 6
Detection performance by geographical region and bounding-box size. For each region, the table reports Average Precision at IoU = 0.5 (AP@0.5) and mean Average Precision across thresholds from 0.5 to 0.95 (mAP@[.5:.95]). Results are given separately for small, medium, and large tower bounding boxes. Bold values indicate the best result within each column.

Region	AP@0.5			mAP@[.5:.95]		
	Small	Medium	Big	Small	Medium	Big
Madrid	0.493	0.791	0.822	0.238	0.530	0.604
Castilla La Mancha	0.530	0.763	0.740	0.278	0.549	0.520
Valencia	0.270	0.850	0.723	0.113	0.560	0.548
Navarra	0.590	0.857	0.868	0.313	0.638	0.660
Andalucía	0.636	0.701	0.505	0.260	0.459	0.406
Catalunya	0.578	0.865	0.509	0.284	0.562	0.324
Castilla León	0.631	0.840	**0.907**	0.379	0.624	0.653
Cantabria	0.697	0.853	0.829	0.397	0.637	0.533
Extremadura	0.570	0.641	0.645	0.280	0.300	0.542
La Rioja	0.633	0.836	0.763	0.363	0.610	0.542
Aragón	0.510	0.903	0.721	0.237	0.633	0.431
Murcia	0.532	**0.923**	0.698	0.282	0.629	0.557
País Vasco	0.611	0.856	0.833	0.347	0.639	0.579
Canarias	**0.741**	0.906	0.817	**0.570**	**0.766**	**0.716**

When assessing how many images can be annotated within a given time budget (Fig. 8(b)), the advantage of WISCAS becomes particularly evident at early stages of the process. Within the first few hours, both WISCAS variants enable the annotation of substantially larger subsets of the dataset compared to all baselines, allowing a large fraction of the final dataset to be obtained well before the total annotation time required by manual approaches.

6.5. Detection performance by region and box size

While Table 5 reports global metrics for detectors trained with different annotation strategies, performance differences are negligible since all methods converge to the same fully annotated dataset. For the remainder of this section, we analyze the robustness of the final detector (trained on all 40,000 images) across regions and tower box sizes.

Table 6 reports per-class AP (IoU = 0.5) and mAP@[.5:.95], considering bounding box size (small, medium, large) as classes and breaking results down by region. The model shows consistent performance for medium and large towers (AP@0.5 ≈ 0.8–0.9), while small towers remain more challenging (AP@0.5 ≈ 0.5 on average). These results support the claim that the detector generalizes well across regions, with limitations mainly due to object scale.

7. Conclusion

We have presented a novel deep learning-assisted pipeline for the annotation of electrical towers, aimed at reducing the human labour required to generate inspection reports for electrical providers. Our approach integrates an AI system in the loop to support, but not replace, human decision-making—ensuring that all final annotations remain under expert annotator supervision and maintain ground truth quality. Experimental results demonstrate a time reduction of up to 267% compared to fully manual annotation.

This work contributes the following: (i) a practical, interactive annotation method that significantly reduces human effort in electrical grid inspection tasks, (ii) tailored sampling strategies that optimize the contribution of AI-based detectors in production workflows, (iii) a fully developed tool now integrated into the company's production pipeline, improving efficiency in the most labour-intensive stages of grid surveillance, and (iv) the creation of a large, high-quality annotated dataset that can support future developments in autonomous drone-based inspection or partially automated helicopter data processing.

Importantly, the proposed system has undergone extensive validation in real-world conditions, spanning thousands of kilometers of infrastructure. While such large-scale field testing entails substantial costs, it provides strong evidence of the practical viability of robotic and AI-assisted systems in commercial settings. Unfortunately, many companies with similar capabilities choose not to publish their work, limiting progress in the field. We advocate for greater openness and collaboration between industry and academia, as this is essential for the accelerated development and deployment of reliable robotic systems in critical infrastructure domains.

Future work will focus on extending the proposed framework along several complementary directions. First, the current sensitivity-based strategy used to determine the weighting parameters of the scoring function could be enhanced through automatic optimization mechanisms, enabling a more adaptive and globally optimal selection of informative samples. In this context, meta-heuristic optimization methods, including immune-inspired algorithms, represent a promising avenue to dynamically balance uncertainty, diversity, and visual complexity in human-in-the-loop inspection workflows. Second, scalability aspects will be further explored to support larger inspection campaigns, additional sensing modalities, and higher data acquisition rates. Finally, while this work targets electrical grid monitoring, the proposed data engine is generic and could be readily extended to other safety-critical inspection domains, such as railway infrastructure, pipelines, or industrial facilities.

CRediT authorship contribution statement

Cristina Benlliure-Jimenez: Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization; **Adrian Penate-Sanchez:** Writing – original draft, Supervision, Methodology, Investigation, Conceptualization; **Javier Lorenzo-Navarro:** Writing – review & editing, Supervision, Investigation; **Modesto Castrillón-Santana:** Writing – review & editing, Supervision, Resources; **Francisco Mario Hernandez-Tejera:** Writing – review & editing, Supervision.

Data availability

The data that has been used is confidential.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Cristina Benlliure-Jimenez reports that equipment and research resources were provided by Aerolaser System S.L. Cristina Benlliure-Jimenez reports a relationship with Aerolaser System S.L. that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to acknowledge and express our sincere appreciation to Aerolaser System S.L. for their valuable contributions to this research, enabling us to conduct experiments and access the necessary resources. Their investment in our project demonstrated their belief in the importance of academic research and its potential impact on the field.

References

[1] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, A scalable active framework for region annotation in 3D shape collections, ACM Tran. Graphics (Proceedings of SIGGRAPH Asia) 35 (6) (2016) 210:1–210:12. https://doi.org/10.1145/2980179.2980238

[2] Y. Chen, R. Xia, K. Yang, K. Zou, Dual degradation image inpainting method via adaptive feature fusion and U-net network, Appl. Soft Comput. 174 (2025) 113010. https://doi.org/10.1016/j.asoc.2025.113010

[3] S. Zhang, Y. Chen, ATM-DEN: image inpainting via attention transfer module and decoder-encoder network, Signal Process. Image Commun. 133 (2025) 117268. https://doi.org/10.1016/j.image.2025.117268

[4] K. Yan, C. Wang, D. Zhou, Z. Zhou, RGBT Tracking via multi-stage matching guidance and context integration, Neural Process. Lett. 55 (8) (2023) 11073–11087. https://doi.org/10.1007/s11063-023-11365-3

[5] J. Zhang, J. Yang, Z. Liu, J. Wang, RGBT tracking via frequency-aware feature enhancement and unidirectional mixed attention, Neurocomputing 616 (2025) 128908. https://doi.org/10.1016/j.neucom.2024.128908

[6] N. Chebrolu, P. Lottes, T. Läbe, C. Stachniss, Robot localization based on aerial images for precision agriculture tasks in crop fields, in: Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), IEEE, Montreal, Canada, 2019, pp. 1787–1793. https://doi.org/10.1109/ICRA.2019.8794030

[7] J. Bian, X. Hui, X. Zhao, M. Tan, A novel monocular-based navigation approach for UAV autonomous transmission-line inspection, in: Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Madrid, Spain, 2018, pp. 1–7. https://doi.org/10.1109/IROS.2018.8593926

[8] V. Lippiello, J. Cacace, Robust visual localization of a UAV over a pipe-rack based on the lie group SE(3), IEEE Rob. Autom. Lett. 7 (1) (2022) 295–302. https://doi.org/10.1109/LRA.2021.3125039

[9] J.L. Paneque, V. Valseca, J.R. Martínez-de Dios, A. Ollero, Autonomous reactive LiDAR-based mapping for powerline inspection, in: Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Dubrovnik, Croatia, 2022, pp. 962–971. https://doi.org/10.1109/ICUAS54217.2022.9836213

[10] A. Savva, A. Zacharia, R. Makrigiorgis, A. Anastasiou, C. Kyrkou, P. Kolios, C. Panayiotou, T. Theocharides, ICARUS: automatic autonomous power infrastructure inspection with UAVs, in: Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Athens, Greece, 2021, pp. 918–926. https://doi.org/10.1109/ICUAS51884.2021.9476742

[11] Y. Gao, G. Song, S. Li, F. Zhen, D. Chen, A. Song, LineSpyX: a power line inspection robot based on digital radiography, IEEE Rob. Autom. Lett. 5 (3) (2020) 4759–4765. https://doi.org/10.1109/LRA.2020.3003772

[12] L. Wang, F. Liu, Z. Wang, S. Xu, S. Cheng, J. Zhang, Development of a practical power transmission line inspection robot based on a novel line-walking mechanism, in: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Taipei, Taiwan, 2010, pp. 222–227. https://doi.org/10.1109/IROS.2010.5648998

[13] N. Pouliot, P.-L. Richard, S. Montambault, LineScout technology opens the way to robotic inspection and maintenance of high-voltage power lines, IEEE Power Energy Technol. Syst. J. 2 (1) (2015) 1–11. https://doi.org/10.1109/JPETS.2015.2395388

[14] W. Chang, G. Yang, J. Yu, Z. Liang, L. Cheng, C. Zhou, Development of a power line inspection robot with hybrid operation modes, in: Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Vancouver, Canada, 2017, pp. 973–978. https://doi.org/10.1109/IROS.2017.8202263

[15] J. Park, U. Shin, G. Shim, K. Joo, F. Rameau, J. Kim, D.-G. Choi, I.S. Kweon, Vehicular multi-camera sensor system for automated visual inspection of electric power distribution equipment, in: Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Macau, China, 2019, pp. 281–288. https://doi.org/10.1109/IROS40897.2019.8968085

[16] L. Yang, J. Fan, Y. Liu, E. Li, J. Peng, Z. Liang, A review on state-of-the-art power line inspection techniques, IEEE Trans. Instrum. Meas. 69 (12) (2020) 9350–9365. https://doi.org/10.1109/TIM.2020.3031194

[17] H.A. Foudeh, P.C.-K. Luk, J.F. Whidborne, An advanced unmanned aerial vehicle (UAV) approach via learning-based control for overhead power line monitoring: a comprehensive review, IEEE Access 9 (2021) 130410–130433. https://doi.org/10.1109/ACCESS.2021.3110159

[18] R.S. Gonçalves, J.C.M. Carvalho, Review and latest trends in mobile robots used on power transmission lines, Int. J. Adv. Rob. Syst. 10 (12) (2013) 1–14. https://doi.org/10.5772/56791

[19] S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems 28, 2015, pp. 91–99.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 9905 of *Lecture Notes in Computer Science*, Springer, Cham, Switzerland, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[21] Ultralytics, YOLOv5, 2020, (https://github.com/ultralytics/yolov5). Accessed: August 27, 2025.

[22] Ultralytics, YOLOv8, 2023, (https://github.com/ultralytics/ultralytics). Accessed: August 27, 2025.

[23] M. Tan, R. Pang, Q.V. Le, EfficientDet: scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01078

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I, 12346 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2020, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[25] M.A. Islam, M. Kowal, S. Jia, K.G. Derpanis, N.D.B. Bruce, Global pooling, more than meets the eye: position information is encoded channel-wise in CNNs, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 8918–8927. https://doi.org/10.1109/ICCV48922.2021.00083

[26] D. Zhang, H. Li, D. He, N. Liu, L. Cheng, J. Wang, J. Han, Unsupervised pre-training with language-vision prompts for low-data instance segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2025). Online ahead of print; pages assigned as "PP". https://doi.org/10.1109/TPAMI.2025.3579469

[27] D. Zhang, L. Cheng, Y. Liu, X. Wang, J. Han, Mamba capsule routing towards part-whole relational camouflaged object detection, Int. J. Comput. Vis. (2025). Online first, published July 14, 2025. https://doi.org/10.1007/s11263-025-02530-3

[28] G. Guo, D. Zhang, L. Han, N. Liu, M.-M. Cheng, J. Han, Pixel distillation: cost-flexible distillation across image sizes and heterogeneous networks, IEEE Trans. Pattern Anal. Mach. Intell. (2024). Online ahead of print; published 1 December 2024. https://doi.org/10.1109/TPAMI.2024.3421277

[29] AISKYEYE Team, VisDrone2019: The Vision Meets Drone Object Detection in Image Challenge, 2019, (Dataset). Accessed: August 27, 2025, https://github.com/VisDrone/VisDrone-Dataset.

[30] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, Q. Tian, The unmanned aerial vehicle benchmark: object detection and tracking, in: Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part X, 11214 of Lecture Notes in Computer Science, Springer, Cham, Switzerland, 2018, pp. 375–391. https://doi.org/10.1007/978-3-030-01249-6_23

[31] R. Abdelfattah, X. Wang, S. Wang, TTPLA: an aerial-image dataset for detection and segmentation of transmission towers and power lines, in: Computer Vision – ACCV 2020, Lecture Notes in Computer Science, vol. 12627, Springer, Cham, Switzerland, 2021, pp. 601–618. https://doi.org/10.1007/978-3-030-69544-6_36

[32] A.L.B. Vieira-e Silva, H. de Castro Felix, T. de Menezes Chaves, F.P.M. Simões, V. Teichrieb, M.M. dos Santos, H. da Cunha Santiago, V.A.C. Sgotti, H.B. D. T.L. Neto, STN PLAD: a dataset for multi-size power line assets detection in high-resolution UAV images, in: Proceedings of the 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, Recife, Brazil, 2021, pp. 215–222. https://doi.org/10.1109/SIBGRAPI54419.2021.00037

[33] A. Savva, R. Makrigiorgis, P. Kolios, C. Kyrkou, Aerial Power Infrastructure Detection Dataset (Version 2.2), 2023, (Zenodo). Dataset. https://doi.org/10.5281/zenodo.7781388

[34] W. Luo, A. Schwing, R. Urtasun, Latent structured active learning, in: Advances in Neural Information Processing Systems 26, Curran Associates, Inc., Lake Tahoe, Nevada, USA, 2013, pp. 728–736.

[35] S. Roy, A. Unmesh, V.P. Namboodiri, Deep active learning for object detection, in: Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, Online, 2018, p. 91.

[36] D. Gudovskiy, A. Hodgkinson, T. Yamaguchi, S. Tsukizawa, Deep active learning for biased datasets via fisher kernel self-supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9041–9049. https://doi.org/10.1109/CVPR42600.2020.00906

[37] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, C. He, Influence selection for active learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9274–9283. https://doi.org/10.1109/ICCV48922.2021.00914

[38] J. Choi, K.M. Yi, J. Kim, J. Choo, B. Kim, J. Chang, Y. Gwon, H.J. Chang, VaB-AL: incorporating class imbalance and difficulty with variational bayes for active learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 6749–6758. https://doi.org/10.1109/CVPR46437.2021.00668

[39] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, J.M. Alvarez, Active learning for deep object detection via probabilistic modeling, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 10264–10273. https://doi.org/10.1109/ICCV48922.2021.01010

[40] S. Aslan, T. Erkin, An immune plasma algorithm based approach for UCAV path planning, J. King Saud Univ. Comput. Inf. Sci. 35 (1) (2023) 56–69. https://doi.org/10.1016/j.jksuci.2022.06.004

[41] S. Aslan, T. Erkin, A multi-population immune plasma algorithm for path planning of unmanned combat aerial vehicle, Adv. Eng. Inf. 55 (2023) 101829. https://doi.org/10.1016/j.aei.2022.101829

[42] S. Aslan, T. Erkin, DUALIPA: a new immune plasma algorithm for path planning of unmanned aerial vehicles, Cluster Comput. 28 (4) (2025) 259–282. https://doi.org/10.1007/s10586-024-04941-2

[43] D.P. Papadopoulos, J.R.R. Uijlings, F. Keller, V. Ferrari, We Don't need no bounding-boxes: training object class detectors using only human verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 854–863. https://doi.org/10.1109/CVPR.2016.99

[44] C.-C. Kao, T.-Y. Lee, S. Pradeep, M.-Y. Liu, Localization-aware active learning for object detection, in: Computer Vision – ACCV 2018, Lecture Notes in Computer Science, vol. 11214, Springer, Perth, Australia, 2018, pp. 506–522. https://doi.org/10.1007/978-3-030-01249-6_23

[45] E. Haussmann, T. Fischer, M. Schubert, J. Nieto, R. Siegwart, Scalable active learning for object detection, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1430–1435. https://doi.org/10.1109/IV47402.2020.9304793

[46] S. Aslan, T. Erkin, A multi-threaded back-and-forth algorithm for planning unmanned aerial vehicles, Aeronaut. J. 129 (1341) (2025) 3083–3108. https://doi.org/10.1017/aer.2025.10037

[47] S. Aslan, T. Erkin, A greedy initialiser based meta-heuristic approach for planning unmanned aerial vehicles, Aeronaut. J. 129 (1339) (2025) 2622–2647. https://doi.org/10.1017/aer.2025.10025

[48] L. Chen, T. Yang, X. Zhang, W. Zhang, J. Sun, Points as queries: weakly semi-supervised object detection by points, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 8823–8832. https://doi.org/10.1109/CVPR46437.2021.00871

[49] Y.-H. Liao, A. Kar, S. Fidler, Towards good practices for efficiently annotating large-scale image classification datasets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4350–4359.

[50] D. Acuna, H. Ling, A. Kar, S. Fidler, Efficient interactive annotation of segmentation datasets with polygon-RNN++, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 859–868. https://doi.org/10.1109/CVPR.2018.00096

[51] H. Ling, J. Gao, A. Kar, W. Chen, S. Fidler, Fast interactive object annotation with curve-GCN, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5257–5266. https://doi.org/10.1109/CVPR.2019.00540

[52] Z. Wang, D. Acuna, H. Ling, A. Kar, S. Fidler, Object instance annotation with deep extreme level set evolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7492–7500. https://doi.org/10.1109/CVPR.2019.00768

[53] T. Shen, J. Gao, A. Kar, S. Fidler, Interactive annotation of 3D object geometry using 2D scribbles, in: Computer Vision – ECCV 2020, Lecture Notes in Computer Science, vol. 12346, Springer, Glasgow, UK, 2020, pp. 751–767.

[54] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[56] L. van der Maaten, G.E. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

[57] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[58] C.-Y. Wang, H.-Y.M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, CSPNet: a new backbone that can enhance learning capability of CNN, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203

[59] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

[60] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8759–8768. https://doi.org/10.1109/CVPR.2018.00913