The Power Line Inspection Software (PoLIS): A Versatile System for Automating Power Line Inspection

Carol Martinez[1,2], Carlos Sampedro[2], Aneesh Chauhan[2], Jean François Collumeau[2], and Pascual Campoy[2] [1] [2]

[1]*Faculty of Engineering, Industrial Engineering Department, Pontificia Universidad Javeriana, Bogotá Colombia. carolmartinez@javeriana.edu.co*
[2]*Computer Vision and Aerial Robotics Group (www.vision4uav.com), Centro de Automática y Robótica (CAR) UPM-CSIC, Universidad Politécnica de Madrid, Spain. pascual.campoy@upm.es*

Abstract

A large amount of data, provided in the form of video data, is acquired during manned inspections flights of electric power lines. This data is analyzed by expert human inspectors to detect faults in the power lines infrastructure and prepare the inspection reports. This process is extremely time consuming, very expensive and prone to human error. In this paper, we present PoLIS: the Power Line Inspection Software, which has been developed with the objective of assisting the analysis of the data acquired during inspection flights. PoLIS is based on the cooperation between computer vision and machine learning techniques to automatically process video sequences acquired during inspection flights, resulting in a set of representative images per electric tower which we call Key Frames. These representative images can then be used for inspection purposes, leading to a drastic reduction of the human operators' workload. At the core of the strategy lies an electric tower detector, which is in charge of estimating the location of the towers within the images based on the combination of a sliding window search technique and a supervised classifier. The location of the tower is then tracked using a tracking-by-registration algorithm based on direct methods, estimating the position of the tower in different images. Finally, different criteria are applied for defining whether the image corresponds to a Key Frame image or not. Extensive evaluation of the proposed strategy is conducted using videos acquired during manned helicopter inspections. The videos constituting this database contain several thousand frames representing both medium and high voltage power transmission lines in the infra-red (IR) and visible spectra. The obtained results show that the proposed strategy can reduce the large amount of data present in the inspection videos to a few Key Frames for each tower. It is also demonstrated that the learning-based approach proposed in PoLIS is appropriate for detecting electric towers, a process which is made faster and more robust by coupling it with a tower tracking algorithm. A Graphical User Interface allowing the application of PoLIS to user-provided videos is also presented in this paper, illustrating the whole process and the automated generation of an inspection report.

Keywords:
Power Line Inspection, Machine Learning, Visual Tracking, Computer Vision

1. Introduction

Nowadays society relies heavily on electric power to satisfy many vital necessities and amenities, this is why uninterrupted electrical power supply is absolutely crucial. Thus the inspection of transmission lines with highly demanding requirements, including accuracy, frequency and cost is required. In order to provide safe, steady and reliable electricity supplies to its consumers, electric power companies invest significant resources in inspection and pre-emptive maintenance of these infrastructures. The most common inspection strategies consist of scheduling regular manned helicopter flights over the power lines, while recording multi-spectral data, typically of visual, infrared, and ultra-violet types; in addition to Lidar and/or radar data.

This data is recorded, tagged with global positioning coordinates and commented whenever necessary by the helicopter's crew over thousands of kilometers. This data is then handed over to ground-based operators in order to identify faults in the power line infrastructure and generate the corresponding reports. Two different types of inspection are conducted: intensive and non-intensive. Intensive inspections provide close views of the electric towers and their components. Non-intensive inspections are faster and safer for the helicopter crew at the cost of a lower level of detail. These inspections have two major drawbacks. First, the flights are very dangerous for the crew while performing intensive inspection because it requires flying close to the power lines. Second, the global inspection process is extremely costly both in hardware and personnel expenses. Hence companies in charge of the power network are willing to automate both data gathering and data processing stages.

Multiple solutions have been investigated in the past decades to answer this demand. Some of the most interesting ones are the use of Unmanned Aerial Vehicles (UAVs), Rolling On Wire (ROW) robots or hybrid approaches for replacing the helicopter-based data acquisition with a safer and cheaper alternative. This paper is focused on the software aspects of the solution, we will not expand on the hardware part since the proposed Power Line Inspection Software (PoLIS) applies both to manned and unmanned inspections.

Computer vision techniques have played a key role in the automatic identification of power line elements such as electric towers, insulators or cables. They target the automation of the data analysis for finding faults in the inspected power lines in a more cost-effective manner. However little research has been done using multi-spectral data, especially with synchronized frames. In addition, few researchers target the inspection of multiple components, multiple defect types and/or inspection type; instead focusing on only one combination component/defect for either intensive or non-intensive inspections. Indeed, detecting multiple components or defects is a challenging task (different types of towers, insulators, meters, etc.), since scales (tower scale: dozens of meters, insulator plate scale: dozens of centimeters) and defects (e.g. rust, contamination, current leak, flashover damage, etc) vary widely. The range at which the inspection is conducted also conditions the potentially detectable defects.

In previous papers [1] and [2], we have presented a machine learning-based detection algorithm for finding electric towers inside video footage and a tracking algorithm capable of keeping track of the detected towers. We present in this paper a global strategy applied to the inspection of power lines using multi-spectral synchronized frames assuming the *a priori* knowledge of the voltage category of the power line, i.e. medium or high voltage. The output of this process consists of the most suitable images for inspection, that is to say images that allow to check the state of the tower and its components. These images, which we call Key Frames, are the ones used by the operator to determine the state of the electric tower and its components in order to generate the inspection report. The scope of this paper is limited to non-intensive inspection although the presented inspection software has a much more general scope and is applicable to intensive inspections as well.

This paper is organized as follows: Section 2 will first present a state of the art of power line inspection using computer vision; Section 3 will then introduce PoLIS; Section 4 will present performance evaluation and optimization of the tower detector presented in [2], its integration with the tower tracking strategy presented in [1] and the Key Frame selection strategy we propose. Section 5 will present our conclusions and introduce future evolutions of PoLIS.

2. State of the art

Pagnano et al [3] highlighted some of the most important general challenges for UAV based power line inspection:

- *visual servoing*, extended by information from other sensors in order to ensure power line tracking and autonomous navigation;
- *obstacle detection and avoidance* since an UAV should not crash into the power line equipment;
- *robust control* for providing high stability and positioning, hence allowing close-up and comprehensive inspections.

In resolving these general challenges, computer vision is expected to play a key role. A UAV platform with on-board visual sensing and processing equipment can greatly facilitate the autonomous inspection task. Some of the main problems that need to be solved are related to autonomously:

1. detect and localize the tower, when it appears in the field of view (FOV);
2. track the tower in subsequent frames;
3. steer the camera to bring and maintain the tower in the center of FOV;
4. once the tower is in the FOV, depending on the kind of inspection, maneuver the UAV and the camera, in order to focus on the tower components to be inspected.

Much of the state of the art has focussed on the first two problems (primarily the first one). Several researchers have applied computer vision techniques for electric tower detection and segmentation in aerial images [4, 5, 6, 7, 8]. Since towers are usually linear structures, most of these approaches are based on detecting lines in an image. The detected lines are post-processed by applying user-defined heuristics, in order to keep only the lines belonging to the tower. Various image segmentation methods are then applied to extract the tower from the image, e.g. direct template matching is used in [4], watershed segmentation in [6], graph-cut in [7]. Golightly and Jones [5], instead of using lines, used corner features to detect a tower in the image. The rest of this section reports in more detail the current state of the art in tower detection and tracking in videos captured from aerial inspections.

Whitworth et al [4] defined an abstract electric tower model as having two straight, near-vertical edges close to each other. This simple model describes towers used for 11 and 33 kV overhead lines which typically consist of two or three bare conductors supported on ceramic insulators mounted on a steel cross-arm at the top of a wood pole [5]. A template matching is performed between the abstract model and the features in the image for locating the tower candidates. Other features, such as two straight edges of the cross-arm and three equi-spaced pin insulators, are used to refine the tower detection and segmentation. This template matching approach was designed for segmenting simple "T-shaped" towers from the video sequences. The template based approach is then recursively applied for tracking as well. The reported results showed good performance in varying quality video sequences, albeit only on a single type of tower.

Golightly and Jones [5] used a modified corner detector [9] to detect and track the tower tops. The original corner detector is modified so that the proximal pixels are clustered together such that only a single point is chosen as the representative of a particular corner. The detected corners exemplify a tower top in an image. The results were reported only for a single type of tower which supports medium voltage (11-33 kV) lines. (similar to [4]). Additionally, a corner matching criteria was proposed to find the correspondence between consecutive frames (beyond the scope of the paper).

Tilawat et al [8] reported a three step approach to tower detection. The first stage applies the "optimal line detector" for detecting the straight lines in an image, giving a set of possible tower candidates. Another linear transformation is then applied to the filtered image. The transformed image is divided into a set of non-overlapping windows, and each window is weighted based on the number of lines passing through each window. The windows with the highest number of lines are considered to be a tower. The proposed approach is simplistic, and as the authors suggest, presence of other linear objects in the image (roads, buildings etc.) will render the approach less useful.

Cheng and Song [7] defined the shape and appearance of a tower by a general rule set such as the tower and the cross-arm are straight; the main tower and the cross-arms intersect at a right angle; the cross-arm is shorter and narrower than the pole. The line segments detected in the pre-processing stage are filtered to remove the segments which, based on the rule-set, are detected as not belonging to the tower. It is assumed that the remaining segments roughly describe the tower. Given the filtered image, a couple of regions are located around the predicted tower. One region completely encompasses the tower, while the other contains at least some internal part of the tower. Taking the internal region as the seed, the graph-cut [10] algorithm is applied to segment the tower structure from the complete region. Authors report accurate segmentation in a variety of conditions. However, the results are reported for just one type of tower.

Sun et al [6] used the images from a stereo camera to segment the electric towers. Given a pair of corresponding 2D images, their approach involves either finding the intersection of the cross-arm with the tower body or the measure of the top of the pole (in cases where the pole may not have a cross-arm). Using the saturation channel of the HSV (hue, saturation and value) color space, a simple threshold is applied to segment the background, based on the assumption that the towers are gray and light in color. Several linear (horizontal and vertical) filters are applied to detect the tower body and the cross-arm candidates. Further heuristics are applied to the stereo image pair, based on the amount of overlap, angular separation and distance between the centroids of the tower candidates, to segment the tower body. Similar heuristics are applied to find the cross arms that intersect the detected tower body. Amongst the set of possible tower candidates, the best candidate is chosen using a simple comparison of the height and the number of cross-arms. In case no cross-arms were found, the watershed segmentation algorithm [11] was used to segment the tower body candidates and especially to locate the correct top of the tower. Authors use the segmentation result to locate the position on the tower where the power line is connected, in order to model the power lines between consecutive towers. This information is eventually used to detect the vegetation which might be trespassing the power line corridor. Like other state of the art works, the results on tower detection were reported only on a single tower type.

Recent advances in power line inspection make use of Machine Learning techniques in order to identify electric towers. Varghese et al [12] applied Deep Learning techniques for addressing the detection of several power line components, such as wires, pylons and insulators. The method is tested using 150 images, and authors report an F-score of 88% when detecting towers. In [13], a boost classifier was used for detecting electric towers.

A big restriction of the above mentioned approaches is that they have been reported on a single type of electric tower, and therefore the authors have made several assumptions which relate to the shape of the tower. However, as shown in Figure 7 and 8, electric towers come in a wide variety of shapes and sizes. Thus, the current approaches can not be considered suitable to solve the tower detection and tracking problem, since these approaches can not be generalized to different types of towers.

Moreover, a complete solution is also missing in the state of the art. Whether the visual inspection is carried out offline (on videos captured from aerial inspections), or online (e.g. on-board a UAV), it is expected that an autonomous inspection should be able to detect and track towers as soon as they appear. In a single aerial mission, one can expect a flight of several hours wherein several hundred towers will be covered. During such missions, the image/video data usually changes drastically due to changes in the background, light conditions and camera movements. A complete visual inspection solution must be robust against such changes and be able to perform detection and tracking during the complete mission.

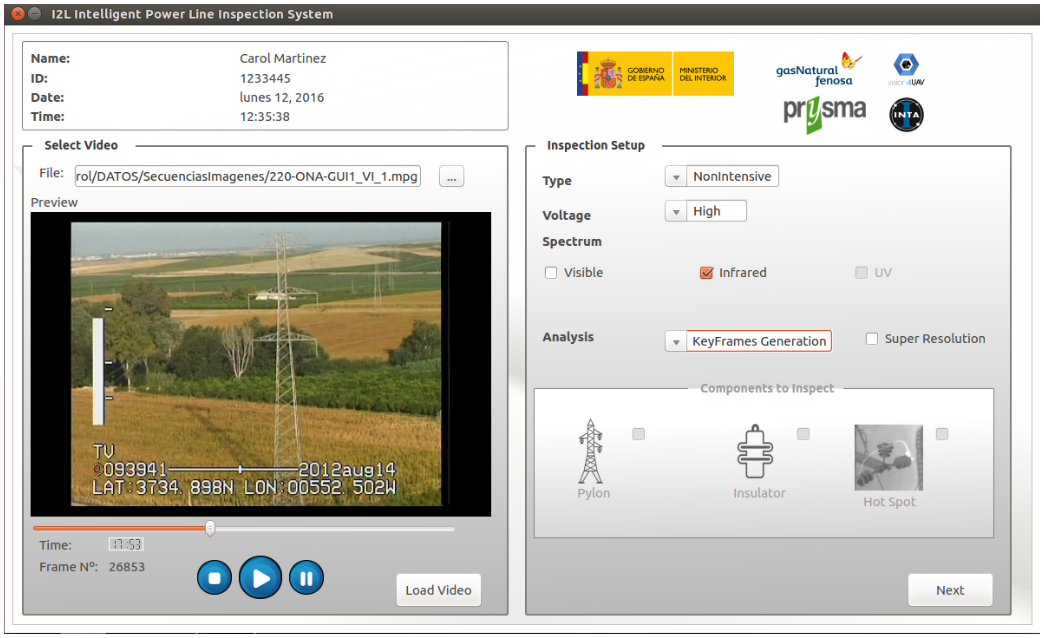
Finally, once the towers are detected, the eventual objective is to inspect the critical tower components (insulators, conductors, clamps etc.). Not all the frames, where the tower is detected, can be considered suitable for locating and inspecting tower components. Therefore, it is necessary to detect the most suitable frames containing the tower, which we call Key Frames, which are the best candidates for localizing and inspecting the tower components. Such Key Frames selection is also lacking in the current state of the art. In fact, some recent works have focused on component detection and inspection, which implicitly assume that the Key Frames are already available [14, 15, 16, 17]. This is the case of [18, 19] where Support Vector Machines (SVMs) classifiers were used for insulator identification and analysis; and in [20] were Convolutional Neural Networks (CNN) were used for electric wire detection.

Having a strategy for autonomous Key Frames selection is essential for filling the current gap between the research on tower detection and the research on the tower component detection and further inspection. Therefore, the present paper extends the state of the art by presenting a complete strategy that covers the three key aspects highlighted in this section. This strategy allows autonomous tower detection, localization and tracking in different spectrums (visible and IR), which is robust against motion, tower appearance changes, light condition changes, as well as the background noise. Furthermore, as part of the existing strategy, a methodology is also proposed and developed to autonomously extract the Key Frames which are the best candidates for tower components detection and inspection. The next section describes the complete strategy in detail.

3. PoLIS: Power Line Inspection Software

The main objective of PoLIS is to reduce the amount of data and time needed by human operators to conduct manual inspection of the power line infrastructures (visually), on video sequences acquired in inspections flights. PoLIS interacts with the human inspector through a GUI (Graphical User Interface) containing different menus, which allows to execute the proposed image analysis strategy on user-provided videos. In this GUI the inspector loads the video and defines the type of inspection to be conducted (spectrum to be analyzed; voltage level; type of analysis, which in the scope of this paper is Key Frames extraction; among other features), as shown in Figure 1.

Once the video is selected by the inspector, it is automatically processed by PoLIS. As a result, a set of representative frames per tower, which we call Key Frames, are identified automatically by PoLIS. These frames will be the only frames available to the inspector to conduct the inspection. PoLIS allows the inspector to visualize the Key Frames and to define the kind of fault(s) found per tower. In order to achieve this, PoLIS provides a set of menus, with typical faults as well as user-defined faults. Finally, after all Key Frames have been inspected, PoLIS generates an inspection report automatically. Figure 1 shows a representative image of PoLIS GUI.


I2L Intelligent Power Line Inspection System

Name: Carol Martinez
ID: 1233445
Date: lunes 12, 2016
Time: 12:35:38

Select Video

File: rol/DATOS/SecuenciasImagenes/220-ONA-GUI1_VI_1.mpg

Time: 17:53
Frame Nº: 26853

Inspection Setup

Type: NonIntensive
Voltage: High
Spectrum: □ Visible ☑ Infrared □ UV
Analysis: KeyFrames Generation □ Super Resolution

Components to Inspect: □ Pylon □ Insulator □ Hot Spot

Figure 1: Graphical user interface of PoLIS. Different menus allow the inspector to set-up the inspection (video to inspect, inspection type, etc) as well as inspect the Key Frames in order to determine faults in the towers and generate inspection reports.

3.1. *The Core of PoLIS*

PoLIS reduces the human operator's workload during inspection by automatically identifying and selecting representative Key Frames. The Key Frames selection process is the most important part of the proposed software, as it will considerably reduce the amount of data the inspector has to analyze.

Figure 2 shows the flowchart of PoLIS. Its core is the Tower Detection stage. The strategy used is based on computer vision and machine learning techniques. Machine learning techniques are used for detecting and estimating the location of electric towers within the images (Tower Detector); and computer vision techniques are used for improving the tower detection, and also for tracking the towers (Tower Tracker). The latter makes the tower detection process faster and more robust, by improving the estimation of the tower's position in the different frames (see Section 4).

The output of the Tower Detection stage is the position of the tower in the current frame, i.e., the Region of Interest (ROI) of the tower. This ROI is used in the Key Frame generation stage for analyzing if the current frame corresponds to a Key Frame or not, based on different criteria (explained later in Section 3.3). If it is considered a Key Frame, then the tower ID and the current image corresponding to the Key Frame are stored.

After all the frames have been processed, only Key Frames are shown to the inspector for conducting a manual inspection of the towers. With PoLIS, the inspector has the option of zooming in and out, in order to visualize carefully the components in the tower; and also the option of specifying the kind of fault found. Once all the Key Frames have been inspected, an automatic report is generated. In this report, information about the number of towers inspected and the types of faults found are presented.

3.2. Tower Detection Stage

For detecting the towers in the images, the Tower Detection Stage combines the estimations from a machine learning based Tower Detector and the estimations from a Tower Tracker, based on image registration techniques. The Tower Detector is the first algorithm to operate. It is used to find an electric tower within the current image. If a tower is detected, then the tracking algorithm gets initialized. When a new image is analyzed, since a tower has been already detected in the previous frame, the Tower Tracker algorithm is used to estimate the position of the tower in the current frame and in the following frames. The results of the Tower Detector and Tower Tracker are analyzed by different criteria, described in [21], in order to define if the found region contains a tower. These criteria are used to switch between the Tower Detector and Tower Tracker. If the criteria are not satisfied, the estimation of the tower's position, in the current frame, is considered

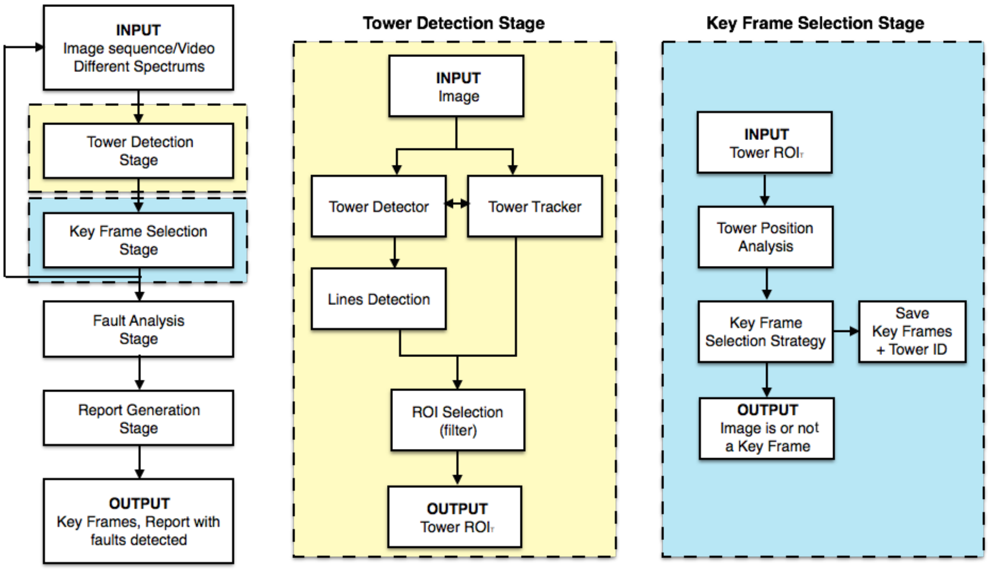


Figure 2: Power Line Inspection Software: PoLIS. The first diagram shows the general structure of PoLIS. The second diagram shows the strategy followed in the tower detection stage. The third diagram shows the structure for Key Frame selection.

unreliable, which ends the tracking and puts the Tower Detector back in charge of detecting a new tower.

Most of the time, the tracking stage operates in isolation. Nevertheless, the Tower Detection stage acts as a backup to detect the position of the tower whenever the tracking stage needs to be reinitialized.

3.2.1. Tower Detector

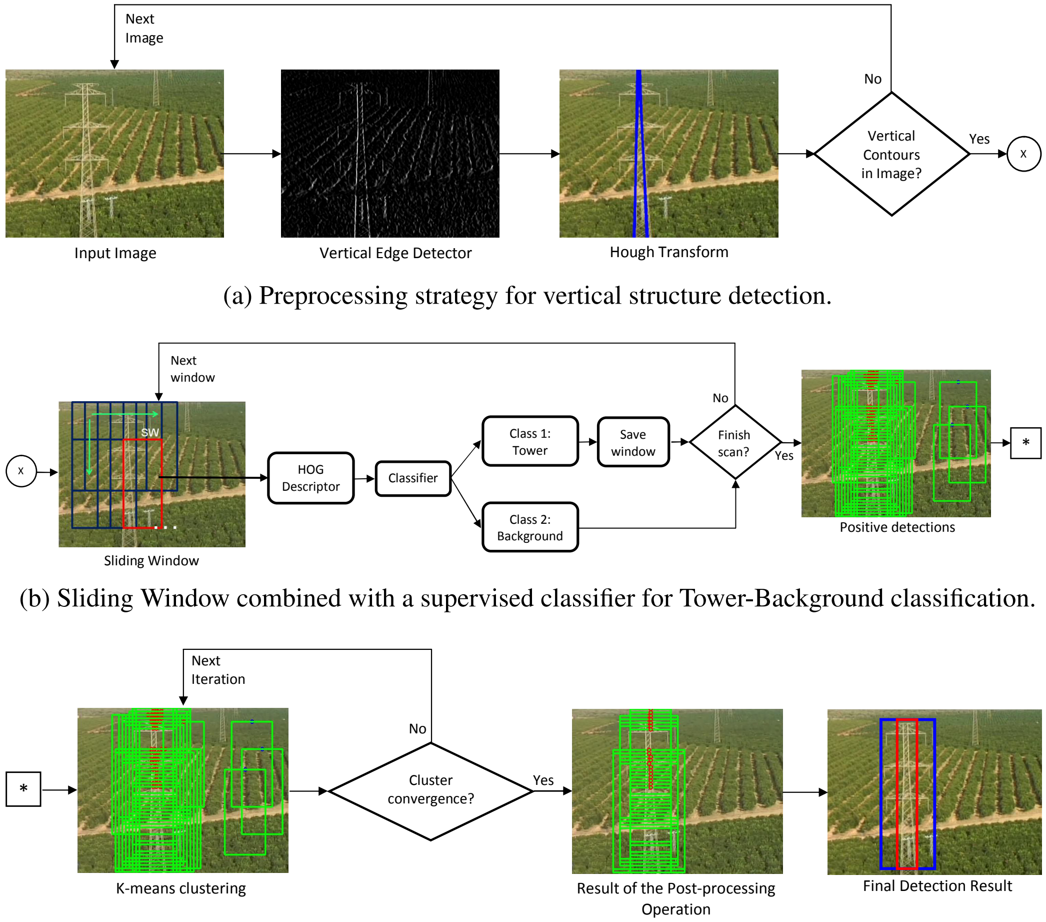
The objective of the Tower Detector module is to recognize and locate the electric towers within an image. For this purpose, the Tower Detector module is composed of three main blocks: a preprocessing block, which is designed to speed-up the detection process by detecting vertical structures in the image using Hough Transform; a core block, which is based on a supervised electric tower classifier applied to every region proposed by a Sliding Window algorithm; and a post-processing block which is in charge of removing false positives. In the next sections, each block of the Tower Detector is described in detail.

Preprocessing block (Figure 3a): This block is in charge of speeding-up the entire detection pipeline by discarding some frames which do not fit several requirements pertaining to the presence of valid vertical contours. For computing the vertical contours in the current frame, the following strategy is applied: the vertical Sobel mask is computed over the input image (converted to gray-scale) in order to extract vertical edges. Then, the resultant image is thresholded in order to retain the most prominent vertical edges and remove noise. This threshold's value has been found experimentally ($sThr = 40/255$) giving a good compromise between noise reduction and vertical edge detection. The last step performed in the Preprocessing stage is a Hough Transform operation applied over the thresholded image to detect vertical lines. The implemented Hough Transform applied here is based on the application of 2 filters within the Hough Space:

- Range of θ: The vertical lines have to be in the range: $[-10°, 10°]$
- Number of votes within the Hough space: a threshold in the Hough space ($hThr = 200$) is applied in order to keep the lines which have a higher length than $hThr$.

Core block (Figure 3b): In the Core block resides most of the intelligence of the Tower Detector. This block is composed of 3 sub-blocks: a Region Proposal algorithm based on a Sliding Window approach, a Feature Extractor module based on Histogram of Oriented Gradients (HOG) [22], and a Supervised Learning Classifier trained for Tower-Background classification. In the following lines the three mentioned submodules are described in detail:

- Region proposal: This submodule is in charge of selecting the ROIs within the image which can be potential candidates for belonging to the Tower class. In the strategy presented in this paper, the region proposal algorithm consists of a Sliding Window technique that uses three main window sizes for High Voltage ($\mathbf{SW}_1$:300×400 pixels; $\mathbf{SW}_2$:250×350; and $\mathbf{SW}_3$:150×350 pixels), while one main window size is used for Medium Voltage ($\mathbf{SW}_1$:120×250 pixels). These window sizes have been selected experimentally based on the average size of the cropped images used for training the classifiers.
- Feature Extractor: In each candidate ROI generated by the sliding window algorithm, HOG features are computed. The configuration of the HOG feature extractor is shown in Figure 4 and summarized here:
  - Window Size: 48×96 pixels.
  - Cell Size: 8×8 pixels.
  - Block Size: 16×16 pixels (2×2 cells).
  - Block Stride: 8 pixels (50% of block overlapping).
  - Histogram configuration: 9 bins, 20° each (unsigned gradient).



(a) Preprocessing strategy for vertical structure detection.

(b) Sliding Window combined with a supervised classifier for Tower-Background classification.

(c) Post-processing stage for detection refinement.

Figure 3: Tower Detection Strategy. (a) Given an input image, a pre-processing step is applied to detect vertical lines within the image. (b) If some vertical structures are detected, a Sliding Window algorithm is applied, using different window resolutions, combined with a supervised classifier for Tower-Background classification. (c) A post-processing step based on k-means clustering is computed for detection refinement. The ROI computed from the vertical structure detection is depicted in red color. The blue ROI shows the final detection result provided by the Tower Detector

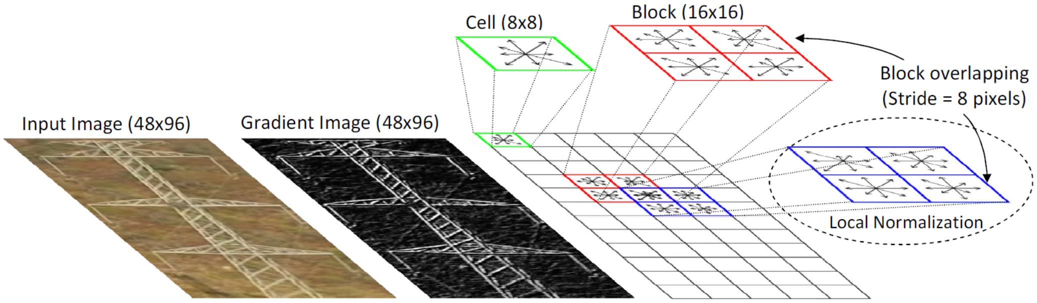


Figure 4: HOG Feature extractor configuration.

The resulting HOG descriptor vector of size 1980×1 is used as input to the corresponding classifier.

– Supervised Learning Classifier: In the work presented in this paper, four supervised learning classifiers have been evaluated: L_2 Regularized Logistic Regression, Support Vector Machines (SVM) (with Linear and RBF kernels), and Multi-Layer Perceptron (MLP). In the next lines, the theoretical foundation of each of the evaluated classifiers is presented.

- L_2 Regularized Logistic Regression. For the construction of this classifier, the implementation presented in [23] has been utilized. In this implementation, the loss function of the Regularized Logistic Regression is given as:

$$J(\omega) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l} log(1 + e^{-y_i\omega^T x_i}) \tag{1}$$

where ω are the parameters to be learned by the classifier. C is the regularization parameter, and (x_i, y_i) is the instance-label pair of the i_{th} training sample.

- SVM classifiers. For the construction of the SVM classifiers the implementation of [24] has been used. The loss function presented for this type of classifier is given as:

$$\begin{aligned} \min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i, \\ subject \quad to: y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i, \\ \xi_i \geq 0 \end{aligned} \tag{2}$$

where ω (weights), b (intercept term) and ξ_i (*slack variables*) are the parameters to be learned by the SVM classifier. C is the regularization parameter, (x_i, y_i) is the instance-label pair of the i_{th} training sample, and $\phi(x_i)$ is a *feature mapping* function.
In the formulation presented in Eq. 2, a kernel function can be defined as:

$$K(x_i, x_j) = \phi(x_i)^T\phi(x_j) \tag{3}$$

The kernel function can lead to different type of classifiers. In this paper, two different types of kernels have been considered:

◦ SVM with Linear kernel.

$$K(x_i, x_j) = x_i^T x_j \tag{4}$$

◦ SVM with RBF kernel.

$$K(x_i, x_j) = exp(-\gamma \left\| x_i - x_j \right\|^2), \gamma > 0 \tag{5}$$

As can be seen in Eq.2 and Eq.5, in this case C and γ are the parameters to be selected for the SVM with RBF kernel classifier.
In Eq. 3 to 5, $K(x_i, x_j)$ represents the kernel function, (x_i, x_j) are points in the input feature space, ϕ is a *feature mapping* function, and γ is the parameter which defines the width of the Gaussian kernel.

- MLP: The Feed-Forward Neural Network utilized in the experiments presented in this paper is a 3-layered MLP, implemented using [25]. This Neural Network configuration was utilized in [2], without optimal parameter selection. In this paper, we extend the work presented in [2], selecting the optimal number of hidden units. Thus, the parameters to be selected in this case will be the number of neurons in the hidden layer.
A Feed-Forward 3-layered MLP can be modeled as:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x)) \tag{6}$$

where the vector $h(x) = s(b^{(1)} + W^{(1)}x)$ represents the hidden layer, $W^{(1)}$ is the weight matrix connecting the input vector to the hidden layer, $b^{(1)}$ is the bias term of the hidden layer, $W^{(2)}$ is the weight matrix connecting the hidden layer to the output layer, $b^{(2)}$ is the bias term of the output layer, and s and G represent the activation function of the hidden and the output layer respectively.

In this paper, the selected activation function for the neurons in the proposed MLP is the $tanh$, which expression is given by Eq. 7.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

Post-processing block (Figure 3c): This module is responsible for removing the false positive windows. After the Core block has been executed, several windows in the image are classified as positive, i.e. are supposed to contain electric tower. With the aim of preserving the most representative windows (those which are more susceptible to have an electric tower inside) and remove the windows that can be false positives, a clustering algorithm based on K-means is applied.

For the work presented in this paper, an evaluation of two post-processing approaches has been carried out:

- Based on the Image space: the proposed clustering algorithm takes as input the x coordinate of the center of the ROIs defined by the windows, and removes the clusters with less members in an iterative process. The result after the application of this post-processing approach is shown in Figure 3c.
- Based on the Hough space and Image space: in this case, additionally to the clusters computed in the Image space, the clusters in the Hough space are calculated. For this purpose, the ρ parameter is used to feed the K-means algorithm. Once both types of clusters are computed, the proposed approach runs in an iterative process keeping in each iteration the closest clusters in terms of image coordinates and removing the rest. The aim of this approach is that the clusters corresponding to the windows computed by the supervised learning classifier converge towards the clusters of vertical lines computed using the Hough Transform.

3.2.2. Tower Tracker

The Tower Tracker is in charge of estimating more efficiently the position of the tower in different frames. The strategy selected for conducting this task is the HMPMR-ICIA (Hierarchical Multi-Parametric and Multi-Resolution Inverse

Compositional Image Alignment Algorithm) proposed in [26]. This is a tracking-by-registration algorithm based on direct methods, which has shown to have fast and robust performance for tracking electric towers [21], and also for tracking objects from cameras on-board UAVs [1].

Direct methods [27] use all the pixels of the object to estimate the motion of the object from the previous frame (where the object's position is known), to the current frame (where the object's position is unknown). Because they do not extract specific features, direct methods are more generally applicable to different scenarios. This is of great importance for tracking electric towers due to their different kind of sizes and shapes.

The core of the HMPMR-ICIA is the Inverse Compositional Image Alignment algorithm [28]. This algorithm estimates the parameters of the motion model that defines the motion of the object from one frame to another. This is conducted by minimizing the Sum of Squared Differences (SSD) as shown in Eq. (8), using a gradient descent approach:

$$\sum_{\mathbf{x}}[T_{(0)}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) - I_{(F)}(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2 \tag{8}$$

Where $T_{(0)}$ is the template image found by the Tower Detector (see Section 3.2.1); $I_{(F)}$ is the current image; $\mathbf{x} = (x, y)^{\mathrm{T}}$ represents the pixel coordinates; $\Delta\mathbf{p}$ is the increment in the parameters of the motion model; and $\mathbf{W}(\mathbf{x};\mathbf{p})$ is the motion model that will be estimated, where $\mathbf{p}^j = (p_1, p_2.., p_i)^{\mathrm{T}}$ is the vector of parameters that describes the transformation from one frame to another.

This ICIA algorithm relies on the linearization of Eq. 8, which is only valid when the range of motion is small (so that the first-order approximation can be valid – i.e. close enough – to find a minimum). In the current application, this assumption is not always applicable as the constant vibrations of the camera can produce large and sudden motion from one image to the next. In addition, the effects of this motion in the image plane increase when the tower is closer to the camera (i.e. the closer the camera and the tower are, the greater the perceived motion in the image). For this reason, in this application the HMPMR-ICIA algorithm is used, i.e. the ICIA is extended with a HMPMR structure in order to cope with large frame-to-frame motions. More details of the advantages of extending the ICIA algorithm with the HMPMR structure can be found in [26].

In the HMPMR structure, the MR structure is created by repeatedly downsampling the images by a factor of 2 [29] (creating a pyramidal structure), according to the different levels (pL) defined for the MR structure. For this application, the number of levels has been defined as $pL = 3$ (i.e. levels: 0, 1, and 2) taking into account that the images have been taken at 30 FPS. Therefore, it is assumed that the objects of the scene move smoothly from frame to frame.

On the other hand, in the multi-parametric MP structure of the HMPMR strategy, different motion models are estimated in each pyramid level. The MP structure is created by defining different motion models to be estimated in each resolution of the image. The complexity of the motion model, should increase, as the resolution of the image increases [26] (the image with the higher resolution is used to estimate the most complex motion model).

For tracking electric towers, in the lowest level of the pyramid $\mathbf{W}^0$ (the one with the highest resolution image, where the superscript represents the level), the following motion model is estimated:

$$\mathbf{W}^0 = \begin{bmatrix} 1+p_1 & 0 & p_2 \\ 0 & 1+p_1 & p_3 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$\mathbf{x}'_{(F)} = \mathbf{W}^0\, \mathbf{x}_{(0)} = \mathbf{W}^0(\mathbf{x}_{(0)}; \mathbf{p})$$

The motion model $\mathbf{W}^0$ is a 3×3 matrix (Eq. 9) parameterized by the vector of parameters $\mathbf{p} = (p_1, p_2, p_3)^{\mathrm{T}}$, as follows: p_2 and p_3 represent the translation in the X and Y axes of the image coordinate frame (upper left corner in the image), and p_1 represents the scale factor. This motion model transforms the 2D pixel coordinates $\mathbf{x}$ (where $\mathbf{x} = (x, y, 1)^{\mathrm{T}}$) in image $\mathbf{T}_{(0)}$, i.e. in frame 0, into the 2D coordinates $\mathbf{x}' = (x', y', 1)^{\mathrm{T}}$ in image $\mathbf{I}_{(F)}$.

This motion model will be the most complex model estimated by the HMPMR-ICIA algorithm and has been selected taking into account that during the inspection, the camera is fixed in the vehicle, therefore small rotations around the different axes, due to the vehicle's movement, do not have a significant impact on the visual characteristics of the tower in the image plane. Thus, most of the image motion, in the image plane, between the vehicle and the electric tower can be represented only by changes in position and in scale.

Therefore, in level 0 of the pyramid, the full motion model shown in Eq. 9 is estimated, and in levels 1 and 2, only the translation parameters are computed (i.e. when $\mathbf{W}^1$ and $\mathbf{W}^2$ are calculated, $p_1 = 0$, and only p_2 and p_3 are estimated).

Following the HMPMR strategy, i.e. by estimating only a small number of parameters in the lowest resolution levels and smoothly increasing the complexity of the motion model through the MR structure, it is possible to increase the range of motion that the algorithm can tolerate, and therefore obtain a robust estimation of the motion model $\mathbf{W}$, which is crucial for this application.

3.2.3. Detector and Tracker Interaction

Once an electric tower is detected by the Tower Detector, detection criteria defined in [21] are then applied in order to define if the found region contains a tower. If the detection criteria are satisfied (i.e. a tower is detected), the vertical structure of the detected tower is used by the tracker to estimate the position of the tower in the next frame. This new region will be the template image $\mathbf{T}_{(0)}$ used by the HMPMR-ICIA algorithm described in Section 3.2.2.

With this information, the tracking algorithm is now initialized. Different components of the tracking-by-registration strategy, such as the pyramidal structure (the two hierarchical structures of the HMPMR strategy), the Hessian matrix, etc., are created and calculated (see [26]). The tracking initialization stage is carried out every time the template image is updated (this occurs when the tower leaves the Field of View (FOV) of the camera, or when the estimation of the tracker is not considered reliable).

When a new frame is analyzed, the tracking algorithm is in charge of estimating the position of the tower in the new frame. The result of the tracking algorithm is checked by different criteria which analyze either the performance of the tracking algorithm or the position of the tower in the image plane (e.g. if the tower is too close to the camera, the algorithm assumes the tower will leave the FOV of the camera, in order to search the next electric tower).

These tracking criteria are used to switch between the Tower Detector and Tower Tracker. If some of those criteria are not satisfied, the Tower Detector will operate until a new electric tower is found. Conversely, if the criteria are satisfied, then the position of the tower in the current frame is obtained by the Tower Tracker.

An example of the general idea of the HMPMR-ICIA algorithm for tracking towers can be seen in Figure 5. The reference image ($\mathbf{T}_{(0)}$) is defined in the first frame (Frame 0, upper left image). This reference image corresponds to a subimage or $\mathbf{ROI}_{(0)}$, that is found by the Tower Detector.

When a new frame is analyzed, e.g. $\mathbf{I}_{(1)}$ (Frame 1, upper right image), the motion $\mathbf{W}_{(1)}$ between $\mathbf{I}_{(0)}$ and $\mathbf{I}_{(1)}$ (Frame 1, green solid arrow) is found by the HMPMR-ICIA, assuming that an initial estimation of the motion $\mathbf{W}_{\text{init}}$ is known (Frame 1, yellow/dashed arrow). Thus iteratively estimating the incremental motion model $\mathbf{\Delta W}$. When an initial estimation is not known, it can be assumed as the identity matrix when the frame-to-frame motion is small. Therefore, the motion $\mathbf{W}_{(1)}$ is estimated, and as a consequence of this, $\mathbf{ROI}_{(1)}$ is found, i.e. the

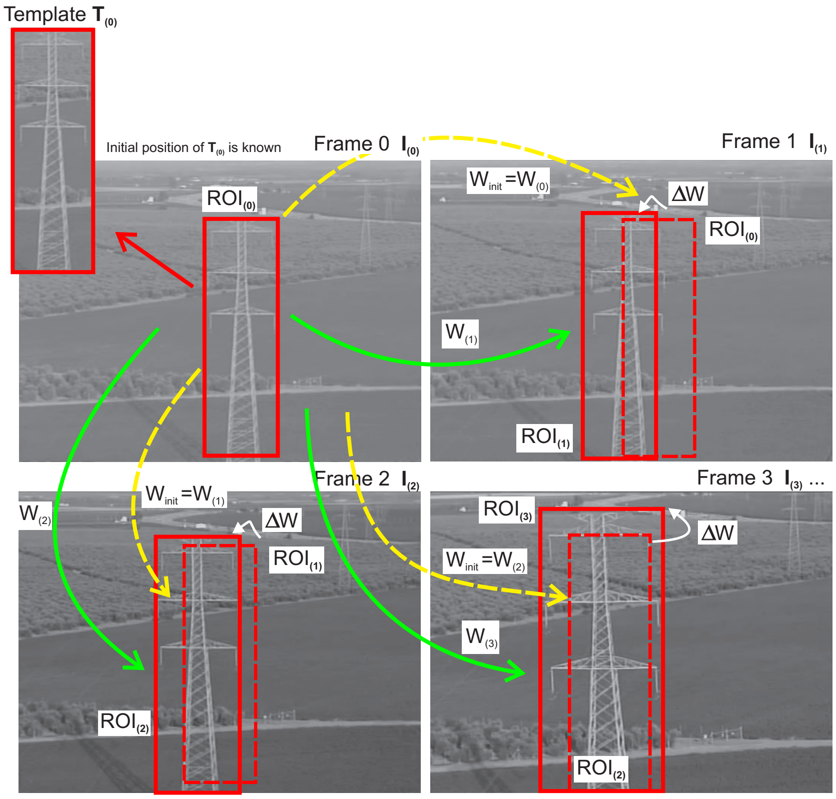


Figure 5: An example of the tracking-by-registration strategy. $\mathbf{ROI}_{(0)}$ is defined in $\mathbf{I}_{(0)}$ by the tower detection algorithm. When Frame 1 (upper right image) is analyzed, the motion $\mathbf{W}_{(1)}$ between $\mathbf{I}_{(0)}$ and $\mathbf{I}_{(1)}$ (green/solid arrow) is found by an image registration technique assuming that $\mathbf{W}_{\text{init}}$ is known (yellow/dashed arrow). Thus iteratively estimating the incremental motion model $\mathbf{\Delta W}$. Using $\mathbf{W}_{(1)}$ the position of the tower $\mathbf{ROI}_{(1)}$ is found. Then, $\mathbf{W}_{(1)}$ is propagated to the next frame, as an initial estimation of the motion $\mathbf{W}_{\text{init}} = \mathbf{W}_{(1)}$ (Frame 2, bottom left image). The process is repeated in each frame and therefore the tower is tracked.

position of the tower in the current frame (e.g. $\mathbf{I}_{(1)}$).

The estimated motion $\mathbf{W}_{(1)}$ (Figure 5, Frame 1, green/solid arrow) is propagated to the next frame, as an initial estimation of the motion $\mathbf{W}_{\text{init}} = \mathbf{W}_{(1)}$ (yellow/dashed arrow, Frame 2, bottom left image). The process is repeated with each frame of the sequence, and therefore, the tower is tracked throughout the sequence.

3.3. Key Frame Selection Stage

For selecting Key Frames, the system needs to know the frames belonging to a span, (i.e. the frames that belong to a specific tower), in order to determine which frames are the most appropriate for conducting visual inspection. This information is automatically generated by PoLIS when processing the video, taking into account when a tower is detected for the first time and when this tower leaves the FOV of the camera (when the tower detector does not find a tower in a defined number of frames)

It is important to mention that currently, this estimation is based only on image information, if additional data is available (e.g. GPS coordinates of each tower), this information could be used to improve the Key Frames selection strategy. The way PoLIS estimates the number of towers is also used to determine when the span starts and ends. All the frames belonging to a specific span are the ones analyzed to determine the Key Frames of each tower.

PoLIS saves the images' IDs of the frames belonging to a span (i.e. to a specific tower). After all the video is inspected, PoLIS uses the images' IDs to select the Key Frames. Different criteria to select Key Frames have been explored:

- Select as Key Frames the last 20 frames of the span. This option can be used when it is desirable to obtain a summary video for each tower.

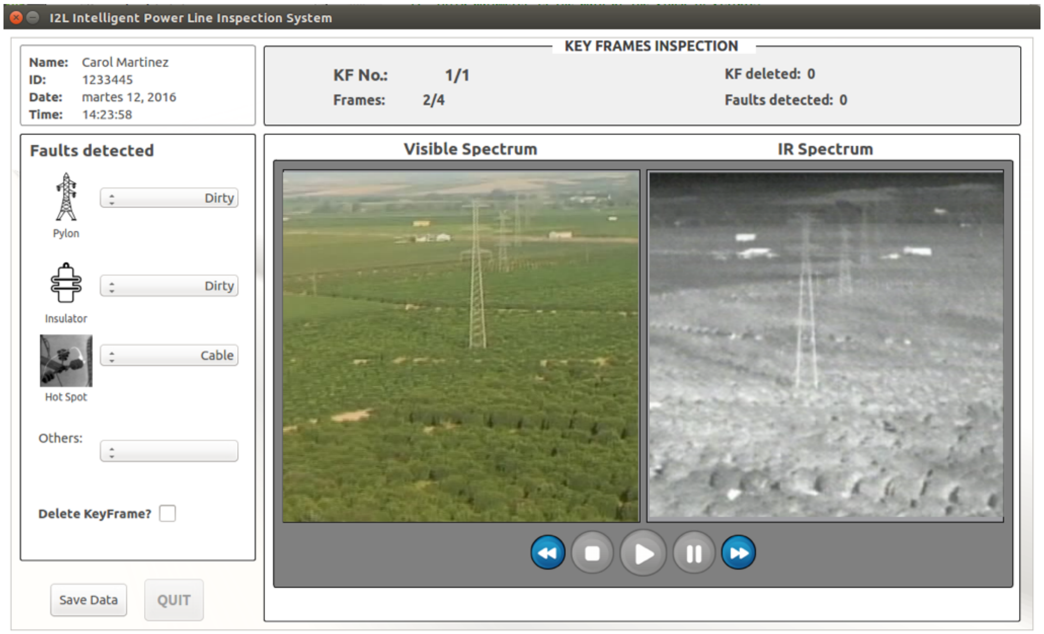
- Select as Key Frames 4 frames from the last 40 frames. This option can be used when the inspection is focused on analyzing, mainly the insulators. Thus, taking into consideration the trajectory of the flight, it could be ensured that in the last frames, the tower is closer to the camera but it has not left the FOV of the camera yet.

- Select as Key Frames 4 well distributed frames from the span, where the tower can be seen at different scales. This option can be used when the inspection aims not only at analyzing the insulators, but also at checking the structure of the electric tower.

The current strategy used by PoLIs is the last strategy: select as Key Frames four well distributed frames from the span (when the tower is far, is not that far, and two when the tower is closer to the camera). This strategy has been adopted taking into account that the towers in the videos can have different trajectories, so it is difficult to ensure that, by using only the last frames, the information required for inspection is visible at least in one Key Frame. Additionally, this strategy makes it possible to analyze for possible problems when vegetation is covering the tower (when the tower is far from the camera), and for problems related to the state of the insulators and other components of the tower (when the tower is closer to the camera).

Once PoLIS has selected Key Frames per tower, the inspector can visualize these frames, inspect them (at different resolutions), and decide the type of faults found, which could be based on pre-defined faults or new faults that can be introduced to the software. Figure 6, shows the options available for inspecting the Key Frames. In the right side of the GUI, the inspector can visualize the Key Frames. Zoom in and zoom out features are available to provide a better view the state of the tower. Additionally, on the left side of the GUI, the inspector can specify the type of fault detected from drop-down menus that contain basic faults. If the detected fault is not available in the menus, the inspector has the possibility of creating a new fault.

Finally, after inspecting all the Key Frames, PoLIS allows the automatic generation of a report, where all the information related to the inspection and faults found, will be summarized.



I2L Intelligent Power Line Inspection System

Name: Carol Martinez
ID: 1233445
Date: martes 12, 2016
Time: 14:23:58

KEY FRAMES INSPECTION

KF No.: 1/1
Frames: 2/4
KF deleted: 0
Faults detected: 0

Faults detected

Pylon: Dirty
Insulator: Dirty
Hot Spot: Cable
Others:

Delete KeyFrame? □

Figure 6: Key Frame Inspection window in PoLIS. This window allows the inspector to visualize the generated Key Frames (right side of the GUI). Zoom in and zoom out features are available to determine the faults present in the tower. In the left side of the GUI, drop-down menus contain basic faults. Additionally, new faults can be inserted.

4. Experimental evaluation and results

4.1. Ground Truth for Tower Detection

In order to evaluate the performance of PoLIS, ground truth data was collected from several real aerial inspection videos. The objective of the collected data is to assess the performance of PoLIS against manual inspectors at the task of detecting the presence of towers in the videos. A strategy for generating the ground truth data was devised to manually label the frames inside a video to mark the presence or absence of an electric tower. The strategy was divided into two key steps: data collection and data cleaning. The following sub-sections describe each of the steps in more depth.

4.1.1. Data Collection

Data: The data used for collecting the ground truth comes from the aerial inspection videos provided by our industrial partners. A total of 14 videos were labeled: 8 videos containing High Voltage towers, and 6 containing Medium Voltage towers. The videos are extremely diverse, that is, each video contains a huge variety of towers (see Figure 7 for some example images). Furthermore, the videos are of very low visual quality, captured in a variety of illumination conditions and contain a huge variety of backgrounds (cities, villages, forests, deserts, etc.). Overall, 477612 frames were labeled.

Labeling Strategy: A labeling software was developed which allows a human user to go through each frame of a video and accordingly label the frames for the presence or absence of a tower. Labeling, here, is very repetitive in nature and can easily lead to human/labeler induced error. Therefore, there will surely be some concerns related to the quality of the labeled data. For example, within a video sequence, the exact frame where a new tower appears (or leaves) is quite subjective, and different people might label different frames where they first observe the appearance (or leaving) of a tower in a sequence of frames. Furthermore, errors in labeling due to fatigue (forgetting to label a region, or forgetting to press a key, etc.) will also lead to erroneous labels. In order to solve this problem, each video was labeled independently by three people.

Table 1 summarizes the key features of the labeled videos and, in the final column, reports the agreement in the labels given by the three labelers per video. As can be noticed, the labelers do not reach a complete consensus on approximately 15% (approximately 71000) of the frames. This inconsistency in the labels is due to the reasons mentioned earlier.

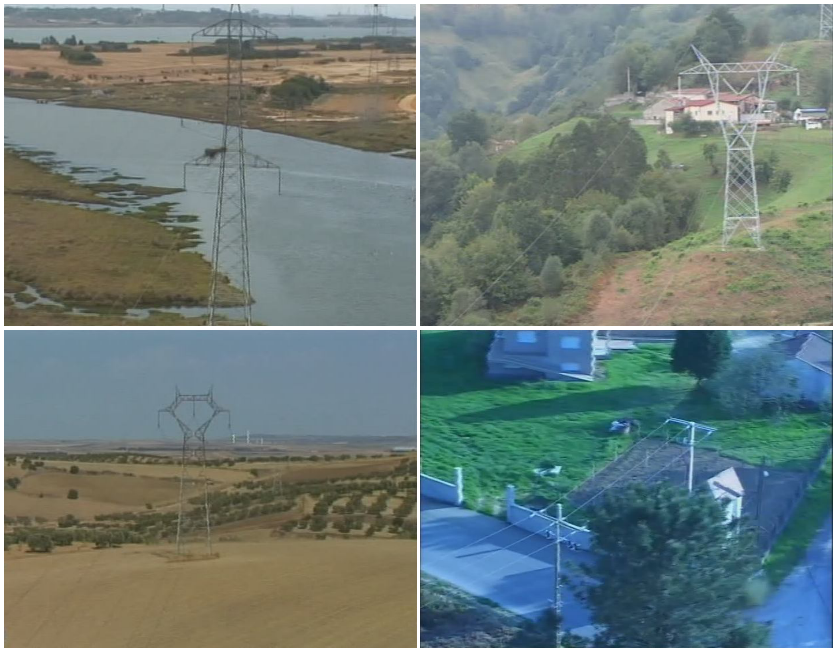


Figure 7: Ground truth data. The videos selected are diverse, with a large variety of towers, backgrounds (cities, villages, forests, deserts, etc.). Note should be taken of the videos' low quality and heterogeneity.

Data cleaning: Given the three labels per image, one possible approach to get the final label can be the majority decision. However, it has been argued that the majority vote might not be suitable in scenarios such as ours [30, 31]. Whitehill et al [31] present a probabilistic inference approach called GLAD (Generative model of Labels, Abilities, and Difficulties), which, given the labels from multiple labelers, computes the probability of a label being correct for each image. Additionally, this method also provides two additional qualitative parameters - the quality of the labelers; and the level of difficulty of an image. The GLAD tool (a software implementation of system described in [31]) was used to combine the labels from the three labelers, leading to a probabilistic label (probability of having an electric tower within the current frame) for each frame. Additionally, for each frame, a qualitative value expressing its difficulty is obtained (occurs especially with the frames where the tower is first appearing or the ones where the tower is leaving). Finally, for each video, an estimate of the labelers expertise is also obtained.

In order to evaluate the tower detection strategy, only the probabilistic labels are used. For the experiments reported here, the probabilistic label was converted

Table 1: Characteristics of the labeled data: summary of the videos used for ground truth collection as well as the percentage of frames, per DVD, for which the ground truth labels from all three labelers are same.

DVD	# Frames	Tower Type	Labeling Consensus (%)
DVD-01	16638	HV	85.92
DVD-02	52301	HV	83.75
DVD-03	46200	HV	83.88
DVD-04	24299	HV	85.50
DVD-05	31726	HV	85.82
DVD-06	52800	HV	88.89
DVD-07	30263	HV	84.13
DVD-08	13452	HV	88.37
DVD-09	12336	MV	88.85
DVD-10	95886	MV	87.51
DVD-11	24477	MV	83.77
DVD-12	23697	MV	89.35
DVD-13	29700	MV	83.25
DVD-14	23837	MV	81.98

to a deterministic one (if probability is greater than 0.95, then the label is 1, a Tower; otherwise it is 0, Background). However, the extra information coming from the measures of image difficulty as well as labeler quality might be exploited in future.

4.2. Evaluation of Classifiers for Tower Detection

The purpose of this section is to evaluate, compare and select the most suitable classifier for the task of tower detection in both types of power transmission lines, High Voltage (HV) and Medium Voltage (MV). At the end of the evaluation process, two different classifiers were selected, one for HV and one for MV.

In this paper, four different classifiers have been studied and analyzed for both types of power transmission lines: L2 Regularized Logistic Regression, SVM with Linear kernel, SVM with RBF kernel, and Multi-layer Perceptron (MLP). For obtaining a reliable comparison of the different classifiers, the following methodology has been applied:

- Optimal parameter selection. For each classifier, a K-fold cross-validation procedure over the parameters of the corresponding model is performed. Once the cross-validation procedure is applied, the final parameter selected is the one that minimizes the cross-validation error.

- Train and test the different classifiers using the same training and testing datasets. Once the optimal parameters of each classifier have been selected, several training and evaluation phases are performed using the selected model for each classifier. In each of these evaluations, the Train and Test sets are selected randomly.

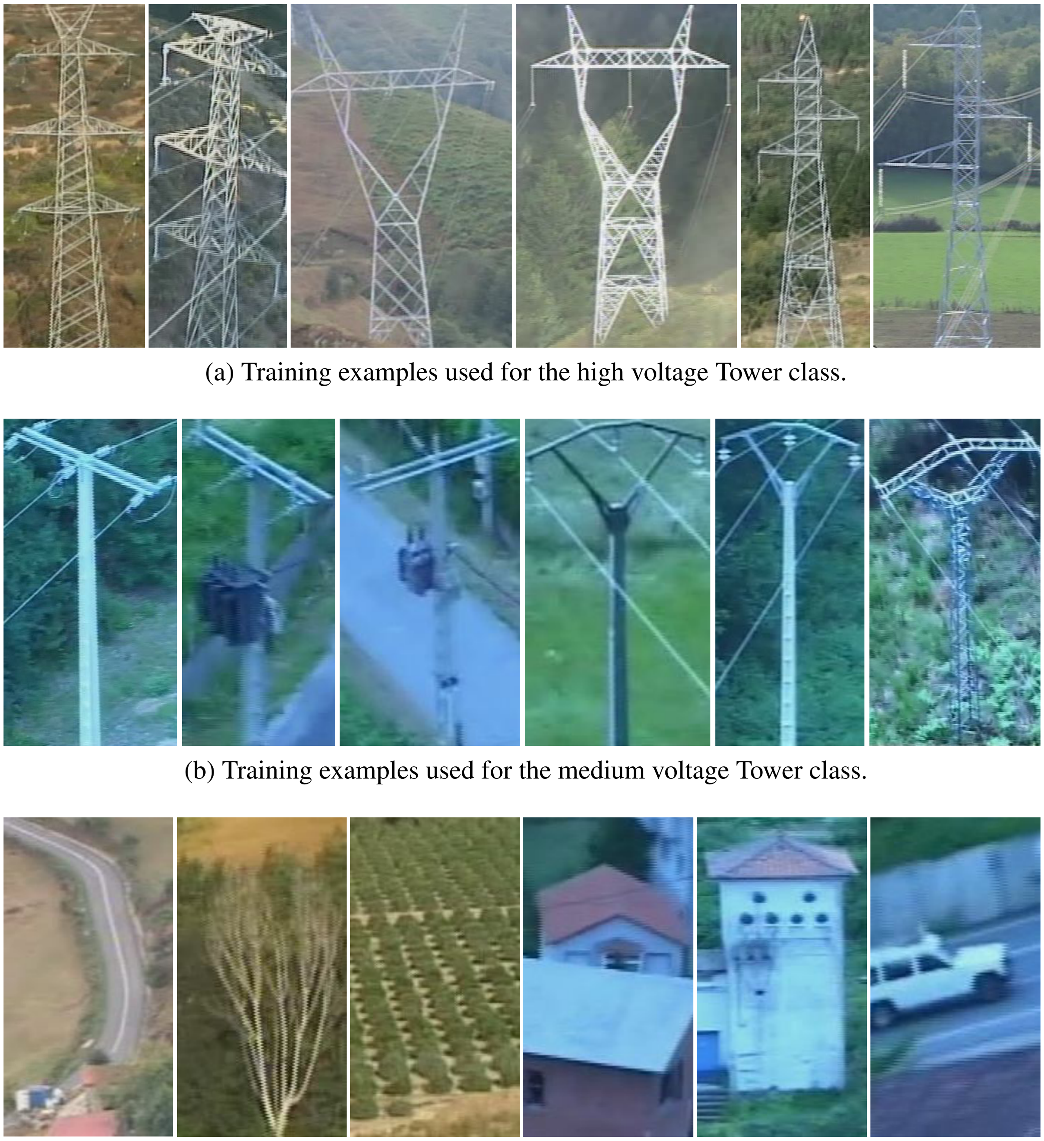
4.2.1. Data collection for the Evaluation of the Classifiers

Due to the lack of public datasets specialized in the components of power distribution lines, a dataset of images for training and evaluating the classifiers was created from aerial inspection data provided by an electrical company. The data supplied by the company consists of several videos of real aerial inspections performed by a manned helicopter. The total amount of videos used for creating the dataset for High Voltage was composed of 12 videos of non-intensive inspections of 720×576 pixels resolution, and of 6 videos of intensive inspections of 720×576 or 1920×1080 pixels resolution. The data collected for Medium Voltage was composed of 8 videos of 720×576 pixels resolution.

From these videos, a supervised dataset of cropped images was created where each of those images was either labeled as Tower or Background. Several examples of the cropped images collected and labeled are depicted in Figure 8. As can be noticed in Figure 8a, the structure of the high voltage towers is very heterogeneous, having structures with symmetric arms, non-symmetric arms, etc. In addition, Figure 8c shows the enormous variety of backgrounds that compose the dataset.

A total amount of 11635 images were labeled either as Background or Tower class, differentiating in the latter case between high and medium voltage. In Table 2 a summary of the number of images utilized for each class is presented. The 4995 images of Background class were utilized in both types of power transmission lines, High and Medium voltage.

Taking into account the number of labeled images shown in Table 2, a data augmentation method has been applied with the aim of increasing the total number of images used for training, preventing possible overfitting problems. The data augmentation procedure consists of a horizontal flip of the images belonging to Tower class, obtaining the mirrored image of each original one. With this



(a) Training examples used for the high voltage Tower class.

(b) Training examples used for the medium voltage Tower class.

(c) Training examples used for the Background class.

Figure 8: Examples of images labeled for training and evaluating the considered classifiers. (a) Examples of images belonging to the Tower class (High Voltage). (b) Examples of images belonging to the Tower class (Medium Voltage). (c) Examples of images belonging to the Background class.

Table 2: Total amount of cropped images collected and labeled.

Voltage	**Tower**	**Background**
High	2325	4995
Medium	4315	4995

approach, the total number of tower images is: 4650 of High Voltage towers and 8630 of Medium Voltage towers.

4.2.2. Selection of the Optimal parameters of each classifier

In this section, the process for selecting the optimal parameters of each classifier is presented. This procedure is needed not only for conducting a coherent comparison between the different classifiers that have been evaluated in this paper, but also to prevent overfitting problems by applying L_2 regularization and cross-validation methods.

For the purpose of selecting the optimal parameter of each classifier, a 5-fold cross-validation procedure over the training set has been conducted, where the cross-validation error has been measured according to Eq. 10.

$$J_{CV} = \frac{1}{N} \sum_{i=1}^{N_f} J_v^{(i)} \tag{10}$$

Where N is the total number of folds, and $J_v^{(i)}$ is the validation error in fold i.

The range of values for each parameter of the corresponding classifier is presented in the following lines:

- L2 Regularized Logistic Regression: In this case the range of values of the Regularization parameter C is:

  C = [0.0001, 0.0003, 0.0005, 0.0008, 0.001, 0.003, 0.005, ... , 10, 30, 50, 80, 100]

- SVM Linear: In this case the range of values of the Regularization parameter C is:

  C = [0.0001, 0.0003, 0.0005, 0.0008, 0.001, 0.003, ... , 1, 3, 5, 8]

- SVM RBF: In the case of this classifier, and as can be noticed in Eq. 5, the parameters to be optimized are the regularization parameter (C) and the parameter related to the width of the gaussian (γ). The ranges of values that have been utilized for the cross-validation procedure are:

C = [0.008, 0.01, 0.03, 0.05, 0.08, 0.1, 0.3, 0.5, 0.8, 1]

$\gamma = [0.008, 0.01, 0.03, 0.05, 0.08, 0.1, 0.3, 0.5, 0.8, 1]$

- MLP: The parameters to be evaluated in the case of the MLP are the number of hidden units that configure the hidden layer of the neural network. The range of values for the number of hidden units is:

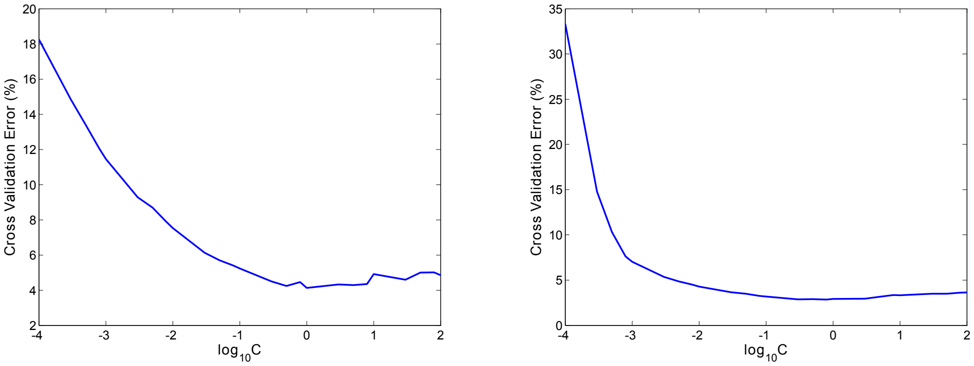
  $\#$ of hidden neurons = [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

After conducting the 5-fold cross-validation procedure to each of the values of the aforementioned parameters (or combination of parameters in case of SVM with RBF kernel), the results obtained are depicted in Figures 9, 10 and 11. As shown in Figures 9a, 9b, 10a, 10b, the minimum value of the cross-validation error can be easily derived, which provides the optimal L_2 regularization parameter value (see Eq. 1 and 2 in section 3.2.1). In Figures 10c and 10d for the SVM RBF configurations, a combination of the parameters C and γ has been applied over the range explained above, which leads to a number of $2^{10} * 5 = 5120$ trainings (taking into account the 5 folds performed using the cross-validation procedure). As can be appreciated in Figures 11a and 11b for the MLP configurations, the minimum cross-validation error obtained in the High Voltage case is 4%, whereas in the Medium voltage case it is reduced to 2.85%, revealing the high influence of the amount of data utilized for training the models in the accuracy of the MLP classifier.

4.2.3. Comparison of the Classifiers

In this subsection, a comparison between the different classifiers has been conducted, with the purpose of selecting the one that has the lowest error on the test set. For the comparison procedure, 10 evaluations have been made, where the Train and Test set have been randomly picked in each evaluation. In each evaluation, 80 % of the data has been used for training and the rest 20 % for testing. The total amount of images used in the comparison of the classifiers is summarized in Table 3 for High Voltage and Table 4 for Medium Voltage.

Final results of the comparison between the different classifiers are depicted in Figure 12. According to these results, the classifiers with the highest accuracy in the Test set are the classifiers based on SVM, with the RBF kernel based SVM giving the highest accuracy results, both in High and Medium voltage images. From the analysis of Figure 12 it can be derived that the behavior of the MLP



(a) Parameter selection for Logistic Regression classifier in HV ($C_{opt} = 1$).

(b) Parameter selection for Logistic Regression classifier in MV ($C_{opt} = 0.8$).

Figure 9: Results of the 5-fold cross-validation procedure for selecting the optimal parameters of the L2 Regularized Logistic Regression classifier, for the High Voltage (HV) and Medium Voltage (MV) datasets.

Table 3: Total amount of samples used in the comparison of the classifiers for HV images.

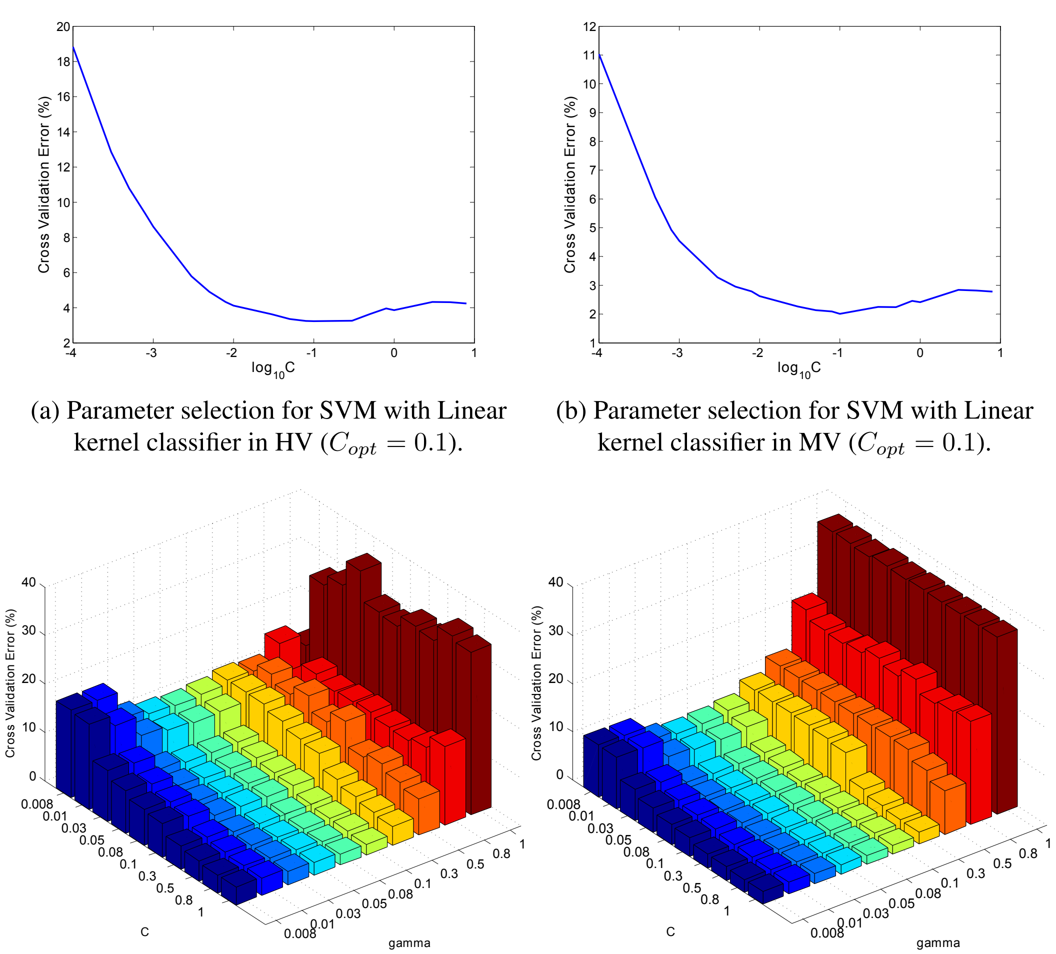
Set	**Tower**	**Background**
Train	3720	3996
Test	930	999

Table 4: Total amount of samples used in the comparison of the classifiers for MV images.

Set	**Tower**	**Background**
Train	6904	3996
Test	1726	999

and Logistic Regression Classifiers is very similar in terms of test accuracy, being the MLP classifier the one with highest generalization error (difference between Train and Test error), as can be appreciated in Figures 12a and 12c, where the percentage of accuracy in the Train set obtained with the MLP classifier is very high, both in High and Medium voltage. This can be explained due to the fact that the MLP classifier is the one that has more hyper-parameters within its model, and therefore more prone to overfitting problems.

Another important result derived from the analysis of Figures 12b and 12d



(a) Parameter selection for SVM with Linear kernel classifier in HV ($C_{opt} = 0.1$).

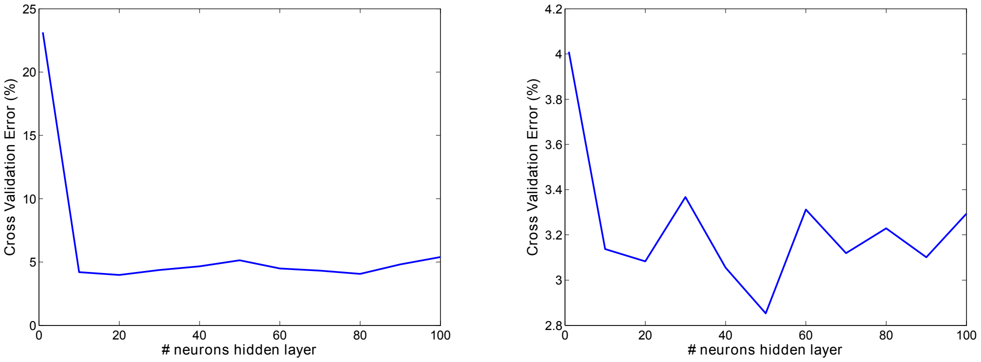
(b) Parameter selection for SVM with Linear kernel classifier in MV ($C_{opt} = 0.1$).

(c) Parameter selection for SVM with RBF kernel classifier in HV ($C_{opt} = 1; \gamma_{opt} = 0.1$).

(d) Parameter selection for SVM with RBF kernel classifier in MV ($C_{opt} = 1; \gamma_{opt} = 0.08$).

Figure 10: Results of the 5-fold cross-validation procedure for selecting the optimal parameters of the SVMs classifiers, for the High Voltage (HV) and Medium Voltage (MV) datasets.

is the difference in percentage of accuracy between the High Voltage and the Medium Voltage classifiers, which in average, is about $1\% - 1.5\%$ in most of the tests. This result can be easily argued due to the number of images obtained for each dataset (see Tables 3 and 4), where the number of images of Medium Voltage towers is almost twice the amount of images of High Voltage towers. This



(a) Parameter selection for MLP classifier in HV ($\#neurons_{opt} = 20$).

(b) Parameter selection for MLP classifier in MV ($\#neurons_{opt} = 50$).

Figure 11: Results of the 5-fold cross-validation procedure for selecting the optimal parameters of the MLP classifier, for the High Voltage (HV) and Medium Voltage (MV) datasets.

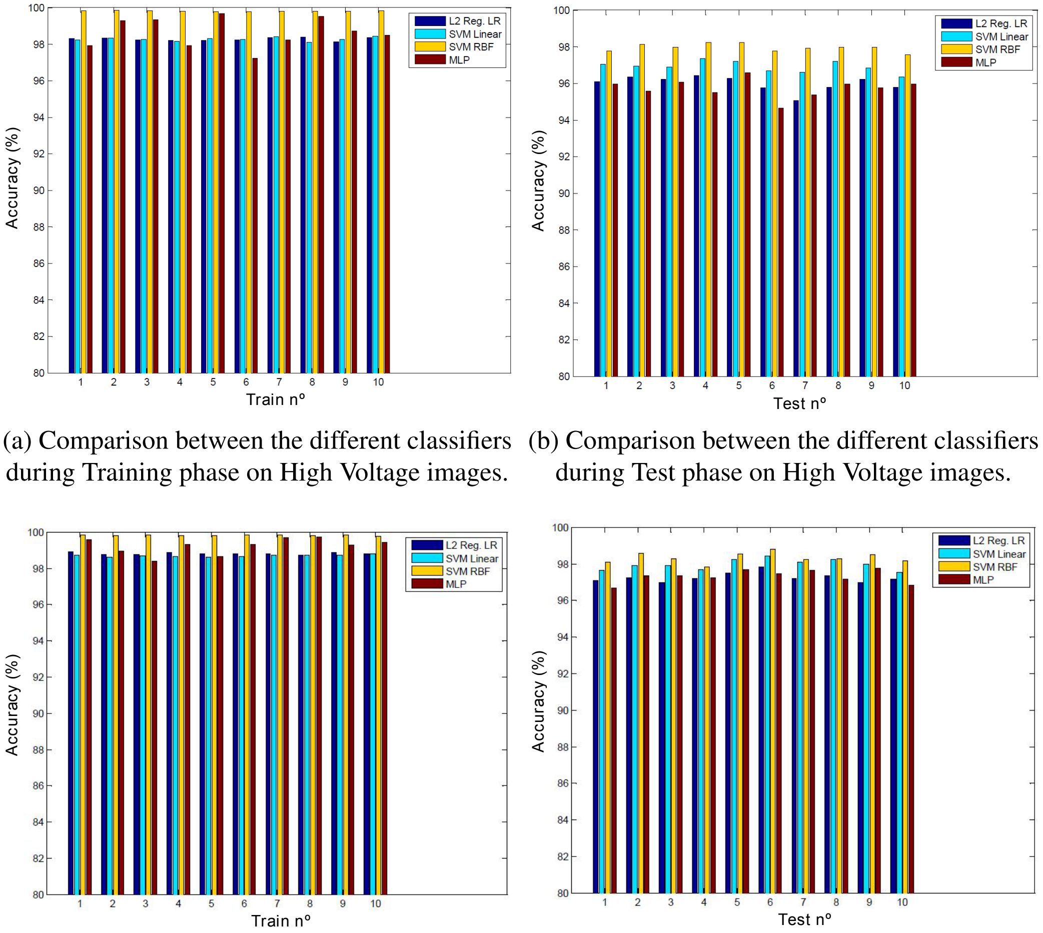
fact leads to a lower generalization error for all the classifiers in Medium Voltage, as can be noticed by comparing Figure 12c with Figure 12d.

4.2.4. Comparison of the classifiers using an image sequence

From previous results, it has been shown that the SVM RBF classifier is the one that obtained the lowest test error both in High and Medium Voltage. In this test, PoLIS is run using three different classifiers, the SVM RBF, the SVM with linear kernel (SVM L), and logistic regression (LR). The MLP classifier is not considered in this test due to its high generalization error obtained in the previous test (see Figure 12).

In this test, DVD-02 file was chosen (see Table 1). This sequence contains High Voltage towers and a variety of backgrounds. PoLIS was commanded to run until detecting 28 Towers (20.0000 frames). Figure 13 shows the results of this test, where the “Detection Status” variable is plotted. For visualization purposes, we have given different values to this variable. The red line represents the ground truth (GT) data. Every time a tower is present in the image, the detection status variable of the GT data is equal to 3. The yellow, green and blue lines correspond to the status of the detection variables when using PoLIS with the LR, the SVM L, and SVM RBF classifiers, respectively.

In Figure 13, it can be seen that the performance of both classifiers (LR yellow



(a) Comparison between the different classifiers during Training phase on High Voltage images.

(b) Comparison between the different classifiers during Test phase on High Voltage images.

(c) Comparison between the different classifiers during Training phase on Medium Voltage images.

(d) Comparison between the different classifiers during Test phase on Medium Voltage images.

Figure 12: Results of the Comparison between the different classifiers considered.

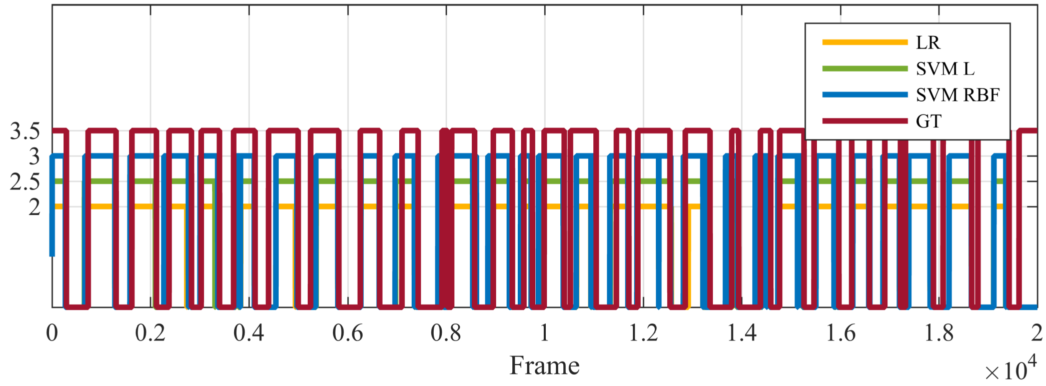


Figure 13: Performance of PoLIS when using SVM L, SVM RBF, or LR classifiers. The red line corresponds to the ground truth (GT) data that represents the presence/absence of electric tower in the frame. Every time the presence of a tower in the image is detected by PoLIS, the detection status variable is set to a specific value (which is chosen manly for visualization purposes) depending on the classifier that is being used: LR (yellow line = 2), SVM L (green line = 2.5), and SVM RBF (blue line = 3).

line and SVM RBF blue line) is very similar, every time there was a tower in the image, both classifiers were able to detect it. All the towers in the sequence were successfully detected by the classifiers. In some cases, the SVM-RBF detected the tower earlier than the LR, but there is no situation where any of the classifiers missed the tower. This is a very important result because if a tower is missed, then this tower can not be inspected.

From these tests we can conclude that SVM-based classifiers and LR classifier behave similarly in terms of detection accuracy. However, when analyzing the processing time, the LR classifier is much faster than SVM. For the previous test, when processing 4 spans, the mean processing time was 0.86s per frame for LR, 1.21s per frame for SVM L, and 7.65s per frame for SVM RBF. Therefore, considering a compromise between accuracy and processing time, the LR classifier has been selected as the one to be used by PoLIS.

4.3. Evaluation of PoLIS

Different tests were conducted to analyze the performance of PoLIS (in terms of detection of electric towers and of extraction of Key Frames). For these tests, we selected some sequences from the available GT data: 5 containing High Voltage (HV) towers, and 3 containing Medium Voltage towers (MV). The sequences selected are shown in Table 5. These sequences contain different types of towers and different backgrounds, all with variable quality as can be seen in the images shown in Figures 7 and 8.

Table 5: Image sequences used for the evaluation of PoLIS.

Sequence	# Frames	Tower Type
DVD-01	16638	HV
DVD-02	52301	HV
DVD-04	24299	HV
DVD-05	31726	HV
DVD-08	13452	HV
DVD-11	24477	MV
DVD-13	29700	MV
DVD-14	23837	MV

4.3.1. Tower Detection Stage

In a previous work [26], the detection of towers using a strategy based only on the Tower Detector (TD) was compared with a strategy that combines a tower detector and a tower tracking algorithms (TD+TT). In this section, we extend that evaluation using more videos with both MV and HV towers. Table 6 summarizes the results obtained in this test.

Table 6: Comparison of the detection of towers when using TD and TD+TT strategies.

Image Sequence	Tower Type	# Frames	# Frames with Towers GT	# Frames with Towers		Processing Time (hr)	
				TD	TD+TT	TD	TD+TT
DVD-01	HV	16638	11035	4661	5647	2.6	0.73
DVD-02	HV	52301	31318	23660	25233	5.04	4.47
DVD-04	HV	24299	13336	4911	5392	3.78	1.35
DVD-05	HV	31726	19996	8261	9674	3.64	1.71
DVD-08	HV	13452	7910	4357	6403	3.65	1.66
DVD-11	MV	24477	18229	13084	15548	15.55	8.21
DVD-13	MV	29700	23324	18078	19958	17.42	8.83
DVD-14	MV	23837	20528	11372	14202	15.84	9.1

In Table 6, it can be seen that by combining tower detection and tower tracking

(TD+TT), the number of frames containing electric towers is greater than when using only the Tower Detector (TD) (compare the Frames with Towers column). Additionally, the processing time reduces significantly (see Processing Time column). When using only the Tower Detector algorithm (TD), the search is conducted in the complete image; whereas when using TD+TT, after a tower is detected, the tracking algorithm estimates the position of the tower taking into account its previous position. This is why the computational cost when combining TD+TT is lower.

Figure 14 shows some of the results obtained in two different sequences (DVD-02 and DVD-08). The green lines represent the ground truth data, the blue lines represent the results when using only the tower detector (TD), and the red lines show the results when using the tower detector and tower tracker (TD+TT). For visualization purposes, we have given different values to represent the presence or absence of electric towers. When there is a tower in the image, the green line reaches the value of 2, and when the algorithms found a tower in the image both red and blue lines reach 1, otherwise the value is 0.

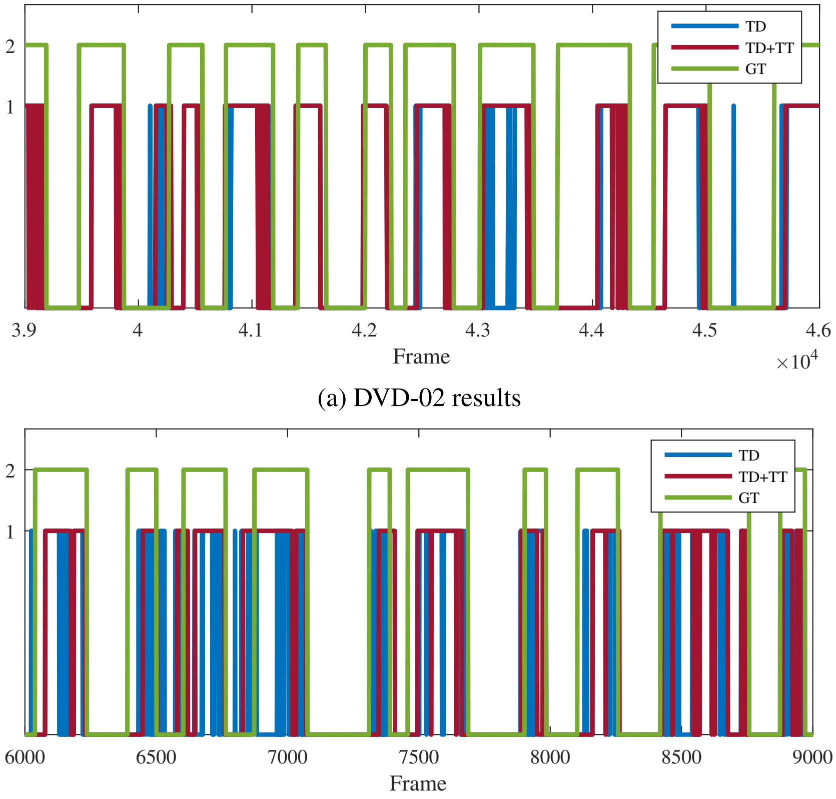
In Figure 14, it can be observed a more stable behavior of the detection of towers when the tower detector and tower tracker are used in combination (TD+TT), this is due to the fact that once a tower is found by the TD, the TT estimates the position of the tower in the following image. Therefore, coping with the instabilities of Tower Detector (TD). This is why for PoLIS we have decided to use TD+TT as the strategy for detecting towers.

4.3.2. Key Frame Selection Stage

In this section the Key Frame generation part of the Power Line Inspection Software (PoLIS) is tested. This is the key component of PoLIS as it allows a significant reduction of the time required to perform the inspection task. The tests have been conducted using 8 different image sequences from the ground truth data: five image sequences containing High Voltage towers (HV), and three containing Medium Voltage towers (MV), see Table 5.

- **High Voltage**

Figure 15 shows a collection of representative images of the selected sequences for High Voltage towers. The green rectangle shows the result of the Tower Detection stage of PoLIS. From those images it can be seen that the sequences have different type of towers, backgrounds, and different resolutions, making it difficult sometimes, to detect a tower. See for example, the lower right image of Figure 15, where it is difficult for a human eye to recognize the whole tower.



(a) DVD-02 results

(b) DVD-08 results

Figure 14: Tower detection test. Comparison of TD and TD+TT strategies. The green lines represent the ground truth data (reaches the value of 2 when there is a tower in the image), blue lines represent the results when using only the tower detector TD, and the red lines show the results when using the tower detector and tower tracker (when there is a tower in the image both red and blue lines reach 1, otherwise the value is 0). A more stable behavior of the detection of towers can be perceived when TD and TT are integrated.

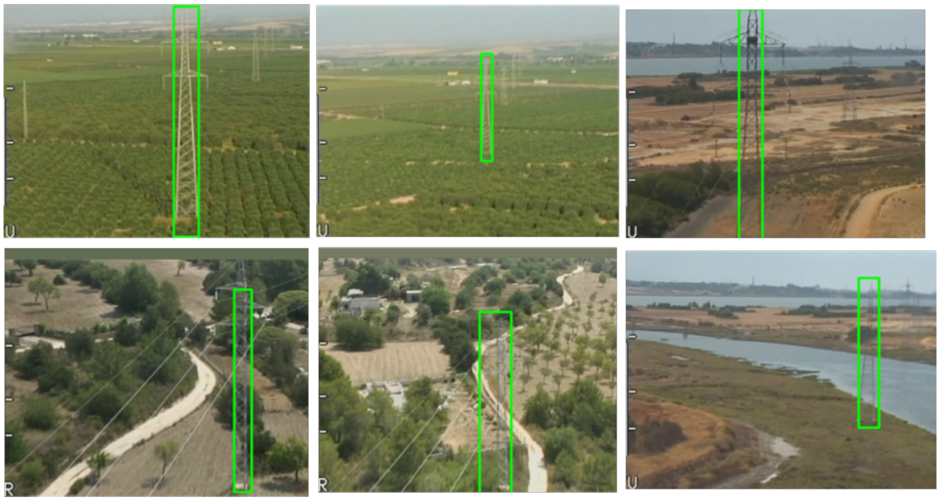


Figure 15: Tower Detection results in representative images from the HV sequences used for the evaluation of PoLIS. Green rectangles represent the results of the Tower Detection stage, which is the core of PoLIS.

Table 7 presents the results obtained when testing PoLIS with the five sequences containing HV towers. If columns 2 and 7 are compared, it is possible to see the reduction of time that is achieved by the proposed software for power line inspection. When PoLIS is used for power line inspection, the inspector instead of checking for example, in DVD-01, 11035 frames, he will focus on analyzing 137 representative frames. Therefore, with the 137 frames, the inspector will define the type of faults each tower has, using the PoLIS GUI and generate a report with the inspection results.

Table 7: Results of PoLIS with different image sequences for HV towers

Image Sequence	# Frames	Video Lenght (min)	# Frames with towers GT	# Frames with towers PoLIS	Processing Time (hr)	# KeyFrames
DVD-01	16638	11.1	11035	5647	0.73	137
DVD-02	52301	35	31318	25233	4.47	307
DVD-04	24299	16.2	13336	5392	1.35	219
DVD-05	31726	21.1	19996	9674	1.71	229
DVD-08	13452	9	7910	6403	1.66	203

In Figure 16, it is possible to analyze in more detail the behavior of PoLIS.

In that Figure, due to space limitations, only the results for DVDs 1 and 2, are presented. In the plots, the green line represents the GT data, which shows in which frames there are towers in the images (green line reaches a value of 2). The red line represents in which frames towers were detected by PoLIS (red line reaches a value of 1), and the blue lines represent the frames selected by PoLIS as Key Frames. From these plots, it is possible to see that in most of the frames containing towers, PoLIS detected towers. Additionally, following the blue lines, it can be seen that from those frames containing electric towers, PoLIS selected representative frames.

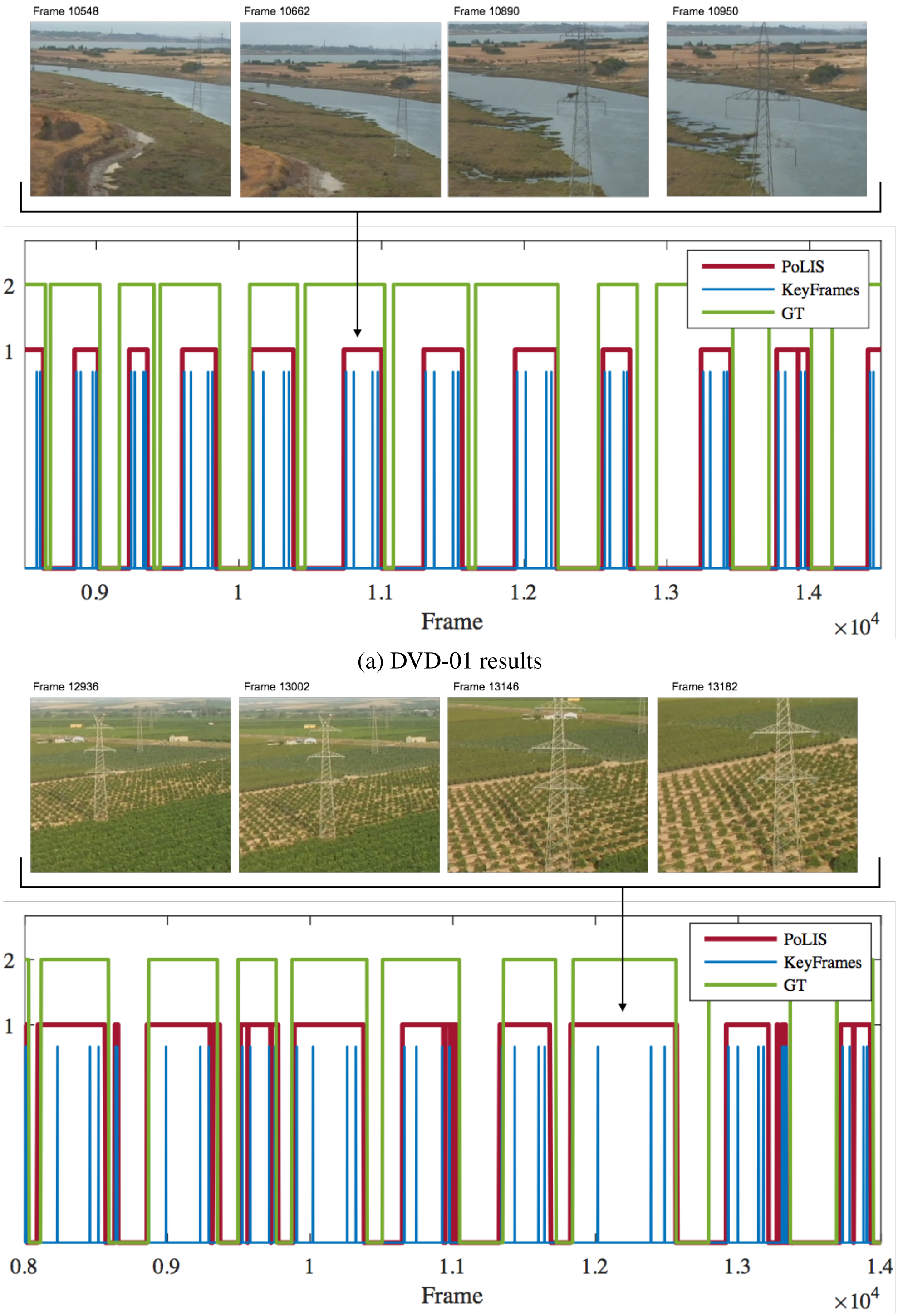
The four images that are located in the upper part of the plots shown in Figure 16 correspond to the Key Frames selected by PoLIS in the area indicated by the arrow. These are the frames that the inspector will use, when using PoLIS, for determining the state of the tower, instead of analyzing all the frames within the span (which can be approximately 1000 images). Therefore, the inspection time is reduced significatively.

- **Medium Voltage**

For the evaluation of PoLIS in image sequences containing Medium Voltage towers, three sequences from the GT data were selected: DVD 11, 13 and 14. Medium Voltage towers are more challenging for PoLIS in the sense that the structure of the towers is more simple (see Figure 17), allowing it to be more easily confused with background information; towers appear more frequently in the images (sometimes two or more towers in the same image); and the quality of the images in our dataset is not appropriate for inspection tasks (blurred images, low resolution, among others).

Figure 17 shows some examples of challenging images found when inspecting Medium Voltage towers. The green rectangle shows the result of the Tower Detection stage of PoLIS (the core of the system). In the images it is possible to see the different kind of towers used for the evaluation of PoLIS in image sequences containing MV towers, and how challenging it is to recognize them in the different type of backgrounds. See, for example, in the last two images from Figure 17, were it is difficult, even for a human, to identify a tower.

Table 8 presents the results obtained when evaluating PoLIS with videos containing MV towers. Analyzing the Key Frame generation component of PoLIS, it can be seen that when using PoLIS, the inspector instead of analyzing thousands of images, only has to analyze a few hundred of them, thereby reducing the time required for inspecting towers. For example in DVD-11 (see Table 8), instead of analyzing 24477 images, the inspector will analyze only 362.



(a) DVD-01 results

(b) DVD-02 results

Figure 16: Performance of PoLIS with videos containing HV towers. Green line corresponds to GT data. Red line represents the results of the Tower Detection stage of PoLIS, and the blue line represents the results of the Key Frame selection stage. The four images correspond to the Key Frames selected by PoLIS in the area indicated by the arrow.

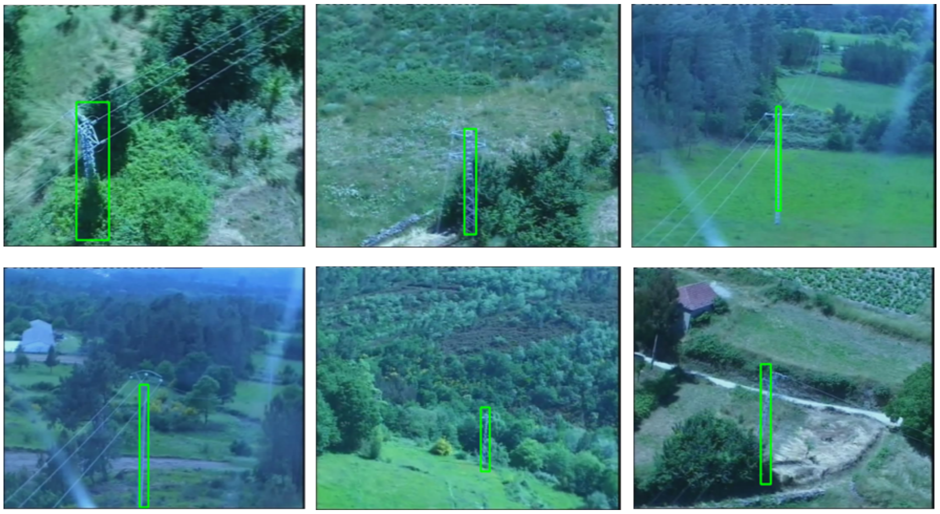


Figure 17: Tower Detection results using MV image sequences. Green rectangle represents the result of the Tower Detection stage of PoLIS.

Table 8: Results of PoLIS with different image sequences for MV towers

Image Sequence	# Frames	Video Lenght (min)	# Frames towers GT	# Frames with towers PoLIS	Processing Time (hr)	# KeyFrames
DVD-11	24477	16.3	18229	15548	8.21	362
DVD-13	29700	20	23324	19958	8.83	536
DVD-14	23837	16	20528	14202	9.1	329

On the other hand, Figure 18 shows the behavior of PoLIS when processing the image sequences of DVD-11 and DVD-13. For visualization purposes, we have given different values to represent the presence or absence of electric towers. In the plots, the green line represents the GT data, which shows in which frames there are towers in the images (green line reaches a value of 2). The red line represents in which frames towers were detected by PoLIS (red line reaches a value of 1), and the blue lines represent the frames selected by PoLIS as Key Frames. The four images that are located in the upper part of the plots, correspond to the Key Frames selected by PoLIS in the area indicated by the arrow.

The plots shown in Figure 18 reveal that PoLIS detected towers in the same frames that the GT data shows the presence of towers. Additionally, in the images located in the upper part of the plots, it is possible to see some of the challenging conditions that were mentioned before of the MV image sequences: towers appear more frequently, and the quality of the image sequence is not appropriate for inspection purposes. This is reflected in the plots, the red line (PoLIS result) is not as stable as the green line (GT data). This behavior was not observed with HV towers.

- **Infrared Images (IR)**

PoLIS was designed for processing the data acquired during power line inspection flights. Infrared images are usually captured when looking for hot-spots, especially in areas close to the tower. This is why, the strategy followed by PoLIS for extracting Key Frames could also prove useful when detecting hot-spots in infrared images.

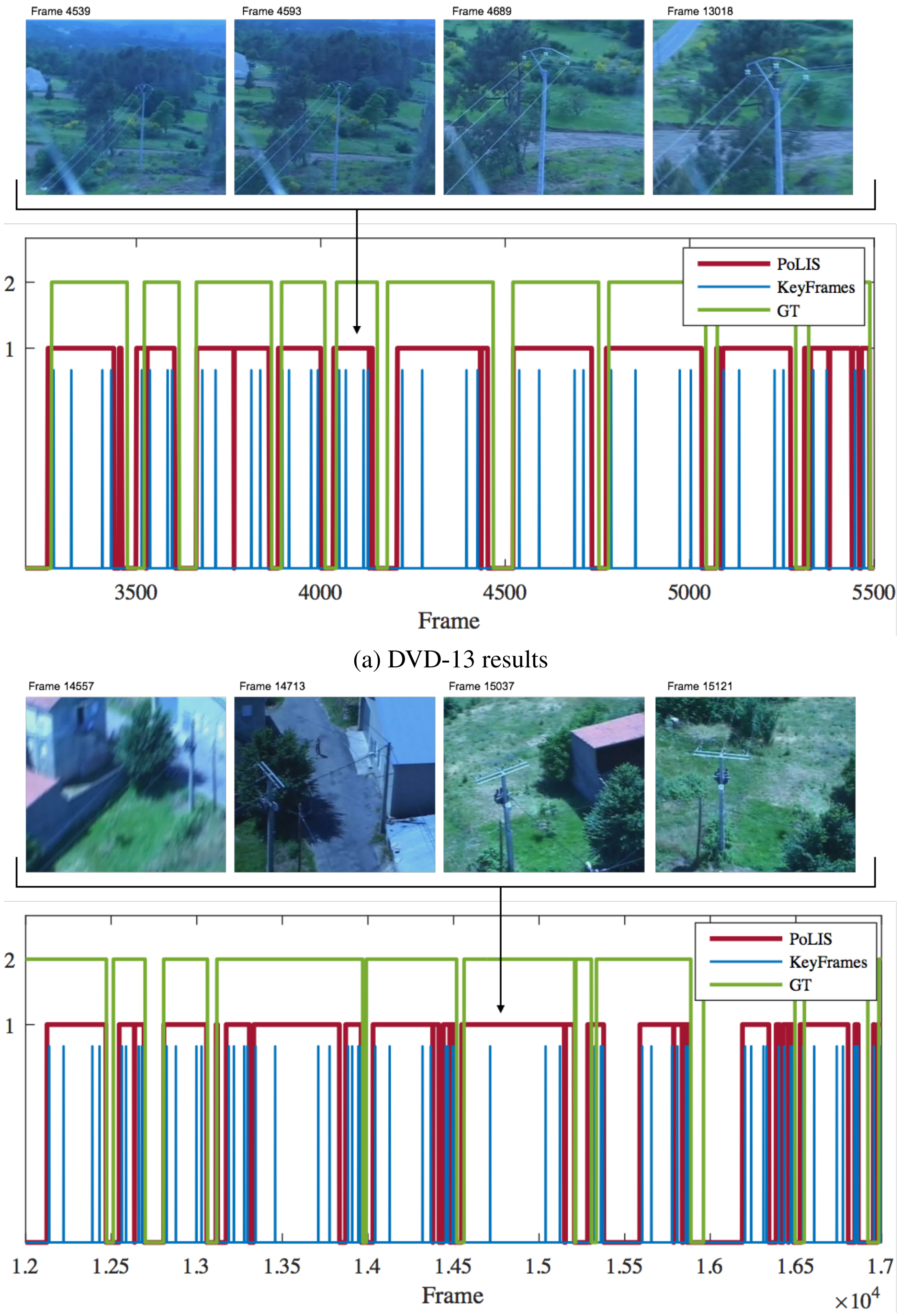
For showing the capabilities of PoLIS in this kind of images, an image sequence from an inspection flight, that contains both spectrums: infrared and visible images, was used. It corresponds to the same sequence of DVD-02. Figure 19 shows an example of the results obtained by the Tower Detection stage of PoLIS. When processing this image sequence, towers were detected in the different spans, obtaining therefore, similar results as the ones obtained when applying PoLIS in the images caputed in visible spectrum.

The reason PoLIS is able to detect and track electric towers in IR images is due to the strategies selected for the Tower Detection stage of PoLIS. The Tower Detector (TD) uses HOG features to detect towers, which are based on image gradients. On the other hand, the Tower Tracker (TT) is based on a direct method, which tracks the object based on the information of all the pixels of the object. These algorithms provide the Tower Detection with the ability to work with different image spectrums.

After processing all the images from the sequence, Key Frames were extracted. The Key Frame strategy that is currently active in PoLIS consisted on extracting four Key Frames per tower. Figure 20 shows three from the four Key Frames selected by PoLIS, for two different towers that appeared in the sequence (one tower per raw). When analyzing Key Frames, PoLIS offers the zoom in and zoom out options. Therefore, as can be seen in Figure 20, the Key Frames selected by PoLIS in the IR spectrum can be used for detecting hot-spots. Thus, these images show the potential use of PoLIS for inspecting also Infrared images.

- **Report generation**

After the inspection video has been automatically processed by PoLIS and the Key Frames have been selected, PoLIS presents those Key Frames to the inspector and allows him to conduct the inspection. With this information PoLIS



(a) DVD-13 results

(b) DVD-14 results

Figure 18: Performance of PoLIS with videos containing MV towers. Green line corresponds to GT data. Red line represents the results of the Tower Detection stage of PoLIS, and the blue line represents the results of the Key Frame selection stage. The four images correspond to the Key Frames selected by PoLIS in the area indicated by the arrow.

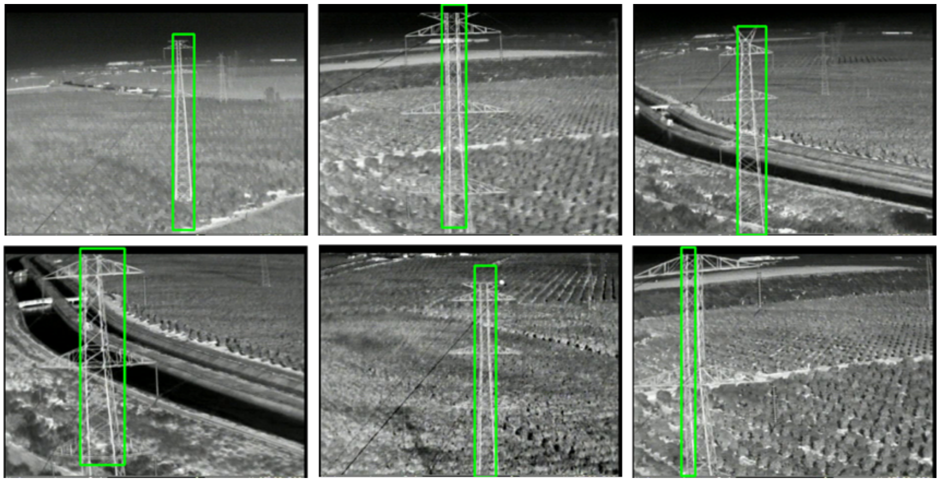


Figure 19: Tower Detection results using IR images. Green rectangle represents the result of the Tower Detection stage of PoLIS.

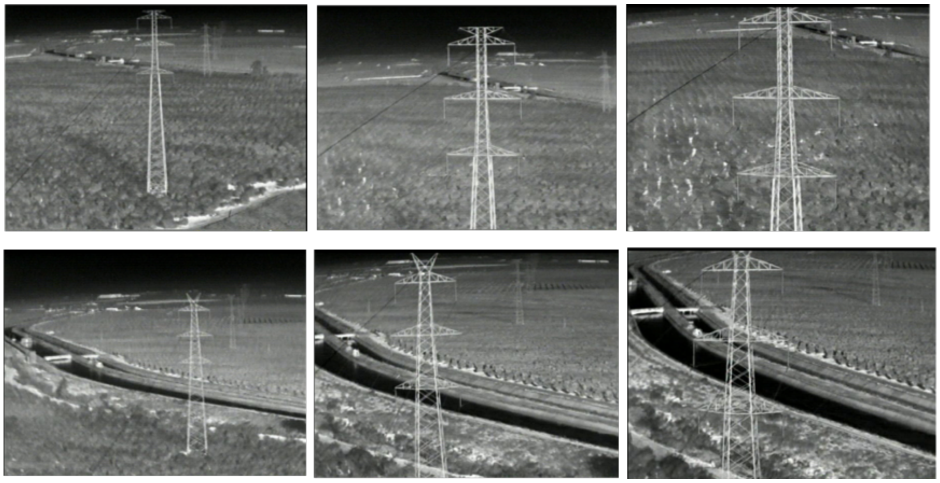


Figure 20: KeyFrame selection results from IR images. First and second rows show three from the four Key Frames selected by PoLIS, for two different towers.

automatically generates an inspection report, based on the faults specified by the inspector.

4.3.3. Result Analysis

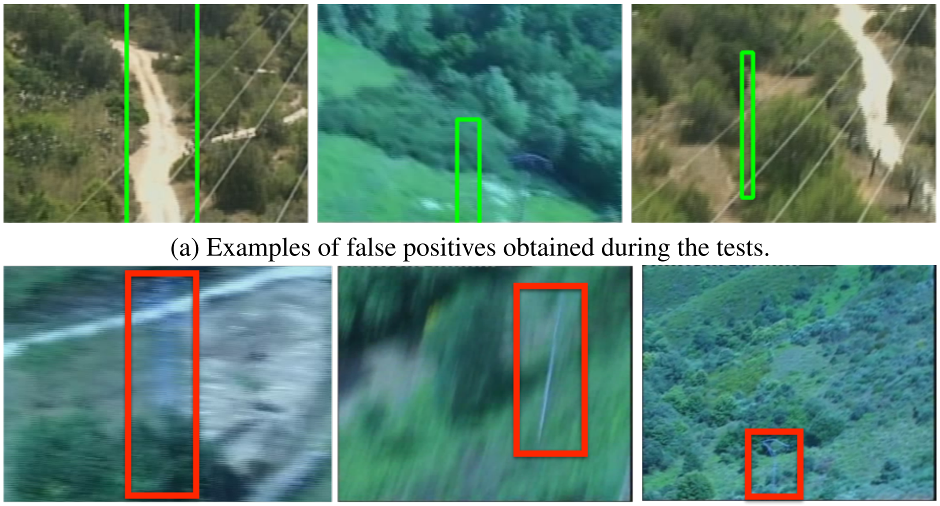
The results presented in this paper have shown the different features of PoLIS for power line inspection. PoLIS has been applied for processing image sequences containing Medium Voltage (MV) and High Voltage (HV) electric towers. In general terms, it has been demonstrated that PoLIS strategy allows to reduce the inspection time significatively. In terms of performance, given that the quality of the data from non-intensive flights is not the most appropriate for inspection tasks, it was demonstrated that PoLIS performance is promising for automating the demanding power line inspection process.

Table 9 shows the confusion matrices obtained in each test. These matrices show the performance of PoLIS, in terms of detection of towers, compared with the GT data that was available. The table shows that PoLIS can discriminate with, a good success rate, background information (true negative TN is high, $> 78\%$, in all the image sequences). Additionally, PoLIS success in detecting towers is high, this is why the obtained false positive (FP) percentage was low in all the tests (FP $< 22\%$).

Table 9: Confusion matrices per DVD. Each row corresponds to the individual matrix. Here, the True class is the Tower and the Negative class is the Background.

HV Sequence	**TP**	**FP**	**TN**	**FN**
DVD-01	45%	12%	88%	55%
DVD-02	78%	7%	93%	22%
DVD-04	41%	5%	95%	59%
DVD-05	49%	2%	98%	51%
DVD-08	71%	22%	78%	29%
MV Sequence				
DVD-11	85%	22%	78%	15%
DVD-13	71%	14%	86%	29%
DVD-14	85%	20%	80%	15%

Figure 21a shows some examples of the false positives found during the tests. Most of the false positives occurred when vertical patterns appeared in the images, for example roads (see first image from Figure 21a). This makes us think that the



(a) Examples of false positives obtained during the tests.

(b) Examples of challenging images encountered within the dataset.

Figure 21: Challenging images containing Medium Voltage towers used when evaluating PoLIS. Green rectangle depicts the PoLIS prediction. Red rectangle shows the position of the tower in the image, which has been manually selected.

main pattern the classifier is learning is the vertical structure of the tower, without including its arms. This is because the images used for training and testing contain both blurred and low resolution images (the arms are not as clearly visible). However, it is important to mention that for an inspection based on PoLIS the false positives are not relevant. This is because the final decision about the state of the tower is taken by the inspector. The software allows the inspector to delete Key Frames that do not contain towers.

On the other hand, in the tests, false negative (FN) rate was high, sometimes reaching the 59% percent. Digging into details of those frames, one of the reasons we have found is that in most of those frames it was in fact difficult for a human to recognize a tower. PoLIS takes advantage of the vertical structure of towers for detecting them. If this structure is not present, it is not possible for the classifier to detect a tower. Figure 21b shows examples of the challenging images that are found in the dataset used for testing PoLIS. The red rectangle (manually drawn) shows the position of the tower. It can be seen that by simply looking at the images, it is in fact difficult for a human eye to identify the tower. This happens especially in the image sequences containing MV towers.

The other reason of high FN rate is related to the way PoLIS detects towers. The tower detector TD is based on a sliding window approach with fix window sizes. If the towers of the span are smaller (at the beginning of the span) or bigger (at the end of the span) than those windows, then PoLIS will not detect those towers. Therefore, a high FN rate in this application can be interpreted as PoLIS being conservative when detecting towers. This is because one span contains thousands of images of the same tower, it is not a problem for PoLIS to skip some of them, and this is why the FN rate is not a concern for measuring the performance of PoLIS.

Table 10 compares the number of towers the GT data has, with the number of towers detected by PoLIS. It is important to mention that for the application proposed in this paper, PoLIS has to detect all the towers present in the GT data. If a tower is missing, this means the tower will not be inspected. As it is shown in Table 10, the percentage of towers detected by PoLIS is high ($> 85\%$) in all the tests, ensuring that most of the towers in the video will be inspected.

Table 10: Percentage of electric towers detected by PoLIS

HV Sequence	# Towers GT	# Towers PoLIS	% Towers Detected
DVD-01	34	32	94%
DVD-02	69	67	97%
DVD-04	55	47	85%
DVD-05	51	51	100%
DVD-08	42	42	100%
MV Sequence			
DVD-11	100	99	99%
DVD-13	145	139	96%
DVD-14	56	54	94%

Taking into account the results in Table 10, we consider them very promising, especially if both the data acquisition stage and PoLIS processing stage of the inspection task are conducted in a coordinated way to help each other. Would this happen, better resolution images, more appropriate for inspection, would become available and PoLIS results will improve significantly. Additional features could also be added to the software, such as automatic fault detection (for example for locating hot-spots).

5. Conclusions and future work

In this paper, a complete solution for power line inspection in multiple spectra imagery has been proposed: the Power Line Inspection Software (PoLIS). The proposed system aims to automate the power line inspection process by reducing the time required to generate inspection reports, which are created based on the analysis of the images captured in real inspection flights. The paper presented the different stages used by PoLIS to automatically extract the Key Frames (most suitable frames for inspection), and analyzed the performance of PoLIS conducting an extensive set of tests on a large dataset of real inspection videos which contains the ground truth data of visible and IR images, with highly varying backgrounds and electric towers.

An exhaustive evaluation of the different stages of PoLIS has been conducted using a dataset from real inspection flights composed of thousands of images, containing different types of electric towers in both, Medium and High Voltage (138416 frames of HV and 78014 frames of MV).

In general, the tests conducted reveal promising results of PoLIS. This will lead to a significan reduction in the workload of the inspector, simultaneously increasing the quality of the inspection by minimizing the scope for human error (e.g. tiredness of the inspector). It has also been shown that PoLIS can be used not only for processing images in the visible spectrum, but also in the IR spectrum, with promising results for improving the hot-spot detection procedure.

Additionally, results also show that the selection of HOG features, in combination with a supervised classifier, and the HMPMR-ICIA tracking algorithm provide PoLIS with the appropriate capabilities to automatically detect and track towers with different shapes and in different image spectra (visible and infrared). With the adopted strategy, robust estimations of the position of towers in the image were obtained in images with low resolution and containing heterogeneous backgrounds.

The different tests conducted have demonstrated that PoLIS is able to efficiently detect electric towers in most of the image sequences that were used. The percentage of detected towers was higher than 85% (reaching in some cases to 100%), when compared with the ground truth data. Results also showed that PoLIS can discriminate background information with an accuracy ranging between 78% and 98%.

During the tests, some false positive (FP$<$ 22%) and false negative (FN$<$ 60%) detections were found. For PoLIS, the false positives do not represent a problem for PoLIS (the inspector can ignore those frames). However, false negatives would imply that some towers were not inspected. In this sense, in order to have an even better performance out of PoLIS, it is important to ensure that the images used for inspection are captured with good quality (e.g. avoiding blurring) and with sufficient resolution to detect the targeted objects. Therefore, ensuring the detection of towers, at least in some of the frames that belong to the span.

This is why we believe that for automating the power line inspection process, both the acquisition of data from flights and the PoLIS processing stages should be coordinated. For example, some of the problems encountered in this paper can be avoided if the flight path and the image acquisition process are planned in order to ensure good quality images, thus avoiding unnecessary processing time of the algorithms.

Future work will focus on exploring strategies for the automatic detection of faults including Deep Learning techniques for electric tower detection, components detection such as insulators, and also for fault detection and analysis in order to help and guide the inspector towards the detection of specific type of faults.

6. Acknowledgements

The work was supported by the Spanish Ministry of Industry under the National R&D Program INNPACTO IPT-2012-0491-120000, the European funding FEDER, and also by the Spanish Ministry of Science under grant MICYT DPI2010-20751-C02-01. The authors also thank the Spanish companies Gas Natural Unión Fenosa and Prysma for the aerial inspection data supplied within mentioned R&D project. The LAL UPM and the MONCLOA Campus of International Excellence are also acknowledged for funding the predoctoral contract of one of the authors.

References

[1] C. Martínez, I. F. Mondragón, P. Campoy, J. L. Sánchez-López, and M. A. Olivares-Méndez, “A hierarchical tracking strategy for vision-based applications on-board uavs,” *Journal of Intelligent & Robotic Systems*, pp. 1–23, 2013.

[2] C. Sampedro, C. Martinez, A. Chauhan, and P. Campoy, “A supervised approach to electric tower detection and classification for power line inspection,” in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 1970–1977.

[3] A. Pagnano, M. Höpf, and R. Teti, "A roadmap for automated power line inspection. maintenance and repair," *Procedia CIRP*, vol. 12, pp. 234–239, 2013.

[4] C. Whitworth, A. Duller, D. Jones, and G. Earp, "Aerial video inspection of overhead power lines," *Power Engineering Journal*, vol. 15, no. 1, pp. 25–32, 2001.

[5] I. Golightly and D. Jones, "Corner detection and matching for visual tracking during power line inspection," *Image and Vision Computing*, vol. 21, no. 9, pp. 827–840, 2003.

[6] C. Sun, R. Jones, H. Talbot, X. Wu, K. Cheong, R. Beare, M. Buckley, and M. Berman, "Measuring the distance of vegetation from powerlines using stereo vision," *ISPRS journal of photogrammetry and remote sensing*, vol. 60, no. 4, pp. 269–283, 2006.

[7] W. Cheng and Z. Song, "Power pole detection based on graph cut," in *Image and Signal Processing, 2008. CISP'08. Congress on*, vol. 3. IEEE, 2008, pp. 720–724.

[8] J. Tilawat, N. Theera-Umpon, and S. Auephanwiriyakul, "Automatic detection of electricity pylons in aerial video sequences," in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, vol. 1. IEEE, 2010, pp. 342–346.

[9] J. Cooper, S. Venkatesh, and L. Kitchen, "Early jump-out corner detectors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 8, pp. 823–828, 1993.

[10] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1101–1113, 1993.

[11] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[12] A. Varghese, J. Gubbi, H. Sharma, and P. Balamuralidhar, "Power infrastructure monitoring and damage detection using drone captured images," in

Neural Networks (IJCNN), 2017 International Joint Conference on. IEEE, 2017, pp. 1681–1687.

[13] B. Han and X. Wang, "Learning for tower detection of power line inspection," *DEStech Transactions on Computer Science and Engineering*, no. iccae, 2016.

[14] I. Y. Gu, U. Sistiaga, S. Berlijn, and A. Fahlström, "Automatic surveillance and analysis of snow and ice coverage on electrical insulators of power transmission lines," in *Computer Vision and Graphics*, ser. Lecture Notes in Computer Science, L. Bolc, J. Kulikowski, and K. Wojciechowski, Eds. Springer Berlin Heidelberg, 2009, vol. 5337, pp. 368–379.

[15] W. Li, G. Ye, F. Huang, S. Wang, and W. Chang, "Recognition of insulator based on developed mpeg-7 texture feature," in *Image and Signal Processing (CISP), 2010 3rd International Congress on*, vol. 1, Oct 2010, pp. 265–268.

[16] S. Murari Mohan, I. Jan Jozef, and R. Eugeniusz, *Fault Location on Power Networks*. springer, 2010.

[17] M. Oberweger, A. Wendel, and H. Bischof, "Visual recognition and fault detection for power line insulators," in *19th Computer Vision Winter Workshop, Krtiny, Czech Republic*, 2014, pp. 1–8.

[18] X. Wang and Y. Zhang, "Insulator identification from aerial images using support vector machine with background suppression," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 892–897.

[19] Z. Zhao, G. Xu, Y. Qi, N. Liu, and T. Zhang, "Multi-patch deep features for power line insulator status classification from aerial images," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 3187–3194.

[20] J. Gubbi, A. Varghese, and P. Balamuralidhar, "A new deep learning architecture for detection of long linear infrastructure," in *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*. IEEE, 2017, pp. 207–210.

[21] C. Martinez, C. Sampedro, A. Chauhan, and P. Campoy, "Towards autonomous detection and tracking of electric towers for aerial power line inspection," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 284–295.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[25] S. Nissen, "Implementation of a fast artificial neural network library (fann)," Department of Computer Science University of Copenhagen (DIKU), Tech. Rep., 2003, http://fann.sf.net.

[26] C. Martínez, P. Campoy, I. F. Mondragón, J. L. Sánchez-Lopez, and M. A. Olivares-Méndez, "HMPMR strategy for real-time tracking in aerial images, using direct methods," *Machine Vision and Applications*, vol. 25, no. 5, pp. 1283–1308, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00138-014-0617-2

[27] M. Irani and P. Anandan, "About direct methods," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2000, vol. 1883, pp. 267–277.

[28] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1090 – 1097, December 2001.

[29] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," RCA Engineer, vol. 29, no. 6, pp. 33-41, Nov./Dec., 1984.

[30] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi, “Inferring ground truth from subjective labelling of venus images.” in *NIPS*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, 1994, pp. 1085–1092.

[31] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan, “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise.” in *NIPS*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2035–2043.