UCL: Unsupervised Curriculum Learning for Utility Pole Detection from Aerial Imagery

Nosheen Abid*, György Kovács*, Jacob Wedin*, Nuria Agues Paszkowsky§, Faisal Shafait†‡, Marcus Liwicki*
*EISLAB Machine Learning, Luleå Tekniska Universitet, Sweden
§Research Institutes of Sweden, Sweden
‡Deep Learning Lab, National Center of Artificial Intelligence, National University of Sciences and Technology, Pakistan
†School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Pakistan

2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA) | 978-1-6654-5642-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/DICTA56598.2022.10034610

***Abstract*—This paper introduces a machine learning-based approach for detecting electric poles, an essential part of power grid maintenance. With the increasing popularity of deep learning, several such approaches have been proposed for electric pole detection. However, most of these approaches are supervised, requiring a large amount of labeled data, which is time-consuming and labor-intensive. Unsupervised deep learning approaches have the potential to overcome the need for huge amounts of training data. This paper presents an unsupervised deep learning framework for utility pole detection. The framework combines Convolutional Neural Network (CNN) and clustering algorithms with a selection operation. The CNN architecture for extracting meaningful features from aerial imagery, a clustering algorithm for generating pseudo labels for the resulting features, and a selection operation to filter out reliable samples to fine-tune the CNN architecture further. The fine-tuned version then replaces the initial CNN model, thus improving the framework, and we iteratively repeat this process so that the model learns the prominent patterns in the data progressively. The presented framework is trained and tested on a small dataset of utility poles provided by "Mention Fuvex" (a Spanish company utilizing long-range drones for power line inspection). Our extensive experimentation demonstrates the progressive learning behavior of the proposed method and results in promising classification scores with significance test having $p - value < 0.00005$ on the utility pole dataset.**

Keywords*—Aerial Imagery, Electric Poles, Computer Vision, Deep Learning, Unsupervised Learning

I. Introduction

Having an uninterrupted electric power supply has become a necessity for the efficient functioning of modern-day society, leading to the prevalence of electricity towers. These towers, however, are immensely vulnerable to natural hazards, e.g., extreme weather conditions [1], corrosion of overhead power lines [2], road accidents [3], short circuits, forest fires, and entanglement by trees or other tall vegetation near the utility towers [4]. Such hazards would not only lead to the deterioration of transmission and distribution of electric power, but could also increase the fragility of utility poles. Fragile utility poles can be dangerous for pedestrians and can damage nearby property and vehicles. These factors all contribute to the importance of the inspection and maintenance of electricity lines, including electricity towers.

The detection of electricity towers is necessary for maintenance, planning, and operations, as well as for risk management and rapid damage assessment after calamities. Mapping an electrical pylon is challenging, and not having exact pylon locations is fairly common [5]. Determining the exact location of power pylons is laborious and time-consuming, and the process includes human interpretation for high spatial resolution imaging from Unmanned Aerial Vehicle (UAV)/aircraft and ground field studies [6]. The high level of human involvement makes finding utility poles in a large area a daunting task. Therefore, there is a need to find more cost-effective methods for mapping assets, such as utility poles.

Remote sensing (RS) offers a promising solution for the automatic detection and mapping of electrical pylons. In fact, many different sensors have been examined for the task, including Synthetic Aperture Radars (SAR) [7], [8], and Light Detection and Ranging (LiDAR) [9], [10], [11]. Cetin and Bikdash [6] mapped electricity poles using the shadow information in aerial images, and Sun et al. [12] mapped power poles using stereo images. Wang et al. [11] proposed a semi-automated approach to classify power lines using LiDAR data of urban regions with precision and recall of about 98%. Another widely researched approach is the use of optical sensors from satellites and UAVs [13], [14], [15], [16]. Due to the small size of utility poles, it is difficult to detect them efficiently from the low resolution of free access or low price satellite imagery [17]. Hence utility companies increasingly use UAVs with high spatial resolution to survey their networks. Utility poles can be efficiently detected from aerial imagery, when the spatial resolution is 30 cm or higher. However, different lighting conditions, background noise, and other factors can still affect the detection of electric poles. In this work, we have used UAV based high spatial resolution gray scale imagery for utility poles detection.

Deep learning methods are proving to be very effective for computer vision tasks. Recent Deep Neural Networks (DNN) based approaches have achieved human-level accuracy in many visual representation learning tasks, like analysis and classification of natural images [18], art images [19], and medical images [20]. More specifically, DNNs have been deployed for effective mapping of a variety of objects from high-resolution RS imagery such as roads [21], buildings [22], and solar arrays [23]. In the last decade, solutions based on Deep Convolutional Neural Networks (DCNN) have also been designed to effectively map utility poles and power grids from aerial imagery. Recently, Huang et al. [24] developed and released a large labeled dataset ($263km^2$) for power grids and reported baseline results for utility pole detection and

power line interconnection. Zhang et al. [25] used RetinaNet and modified brute-force-based line-of-bearing to estimate the locations of detected roadside utility poles with crossarms from Google Street View (GSV) images.

Although the use of deep learning models is providing state-of-the-art results, it also carries some serious limitations.

1) These models are mostly supervised and require large amount of labeled data to train the deep architecture. Labeling the data is labour-intesive and time-consuming task. In most of the cases it requires expert domain knowledge making it more expensive.
2) The resulting models are domain specific. Meaning that once they are trained on one dataset, their performance in (terms of accuracy) significantly decreases when deployed on another dataset of the same problem domain.

To tackle these limitations, the concept of Curriculum Learning (CL) (proposed by Bengio et al. [26]) has been used by some researchers in the machine learning domain [27]. Recently, Abid et al. [28] used the CL concept and proposed Unsupervised Curriculum Learning (UCL) to deal with the limitations of supervised deep learning architectures. A similar unsupervised learning based approach [29] is used for detecting burnt regions of 2019-2020 wildfire happened in Australia.

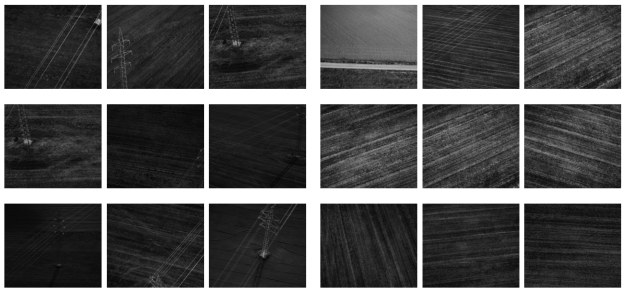
In this paper, we use an unsupervised deep learning framework to classify utility poles from high-resolution grayscale imagery. The proposed solution is an updated version/modification of UCL [28]. In the UCL framework we fine-tune a pre-trained computer vision deep learning model using pseudo-labels generated from the clustering of examples in the target domain. To prevent the model from learning noise, only "reliable samples" are used in this fine-tuning step. These "reliable samples" are provided by the selection operation, which aims to filter out (or exclude) samples far from the cluster centroids to avoid outliers. For this filtering step, a similarity index threshold is applied. A selection operation based on a similarity threshold alone, however, can generate imbalanced sample selection in different categories, leading to a problem of imbalanced class distribution among reliable samples. This situation requires a modification in the proposed selection operation of UCL method that filters the samples from clusters without creating an imbalanced class distribution problem.

In this paper, we introduce a new selection operation to avoid imbalanced class distribution problems in an unsupervised deep learning framework for electric pole classification. The main reason this is needed is that the provided dataset of utility poles has a small count (368) of utility pole images, compared to the count (3483) of images not containing utility poles. The detailed experimental validation shows that the proposed solution has the capability to learn the characteristics from electric poles from limited sample size.

The rest of the paper is structured as follows: Section II discusses the dataset used and the pre-processing we applied on it. Section III describes the used methodology and explains the proposed solution. In Section IV, a detailed experimental analysis is presented. Section V gives the critical analysis of the updated UCL. Lastly, Section VI gives a summary of the work and an outline of future directions.

II. Area of Interest and Data

The data consists of high-resolution (5328×4608 pixels, or 24.55 megapixels) grayscale aerial photographs of farmland intended to monitor installed electric towers. Some of these images contain an electric tower (also called an electricity pylon or a transmission tower), while others do not. In the images containing towers, the tower itself often occupies only a small percentage of the pixels. Some images only contain the shadow of a tower - these images are also labeled as tower images. Most of the images of towers also contain power lines. The non-tower images mostly contain farmlands, and a fraction of them also contain power lines. Examples of tower and non-tower images are shown in Figures 1a and 1b, respectively.



(a) Tower Images Samples (b) Not Tower Images Samples

Fig. 1: Preprocessed utility pole dataset image samples.

A. Cleaning and Balancing

The dataset is in a raw state, meaning that some images are junk images, depicting close-ups of rocks and similar objects taken in the beginning of take-off or at the end of landing - in other words, not aerial photographs. These images have been manually removed. A few images containing almost no electricity tower were also removed. This data cleaning was done to avoid the unsupervised framework from learning the noisy samples. Fig. 2 shows a few examples of removed images.

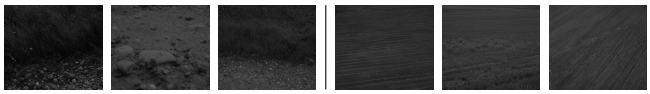


Fig. 2: Examples of Removed Images; left: from "Non_tower" category, right: from "tower" category.

The dataset provided has 3851 samples in total, out of which 368 contains tower images, while a large majority (3483) are non-tower images. After data cleaning, the total count of samples remaining is 3371, out of which 3045 are non-tower and 326 are tower images. Because of this largely unbalanced class distribution, the data has also been balanced.We did so by randomly sampling non-tower images so as the remaining set would contain the same amount of tower and non-tower images. After balancing, we got a count of 694 samples. These samples were divided into 0.8:0.2 for training and validation, and testing, respectively.

III. Updated UCL for Electric Pole Learning

The proposed updated UCL method formulates the electric poles detection from RS data in an unsupervised manner. The standard UCL framework is composed of three essential modules:

1) A CNN architecture that learns the robust and distinctive electric pole features from the data, and
2) The unsupervised clustering scheme that splits the images into groups on the basis of similar appearance properties.
3) The selection operation that provides a number of reliable samples for each cluster, which can be then used for fine-tuning the CNN architecture.

The core concept of the framework is to *iteratively* fine-tune the deep feature extraction and clustering mechanism in an unsupervised manner. The concept of UCL has also been explored in the computer vision [30], [31] and RS community [29], [28] which make use of transfer learning and latent space representation for cross-domain adaptation. The clustering results are treated as pseudo labels to fine-tune the deep model. The process of fine-tuning the deep model continues progressively with a growing count of training samples and respected pseudo labels until the model converges.

In UCL, any CNN architecture can be deployed. The selection of CNN model varies from the nature of problem and its complexity. Like standard UCL, we use VGG-16 model (pre-trained on ImageNet [32]) as the CNN model tasked with learning the electric pole features. This pre-trained VGG-16 is first used to extract features from the images in electric pole dataset. The output of the last convolutional layer is then extracted to get feature maps. These feature maps are flattened to generate feature vectors for each image. These feature vectors are clustered using a well-known clustering algorithm, k-means. The size of the input layer of VGG-16 is adjusted according to the image patch size of the electric pole dataset. In the more formal description, samples of the dataset are represented by $\{x_i\}_{i=1}^{N}$. VGG-16 is used to extract features maps $\{f_{Map_i}\}_{i=1}^{N}$ which are later flattened $\{f_i\}_{i=1}^{N}$ to get feature vectors. These feature vectors are clustered into K clusters $\{c_k\}_{k=1}^{2}$ using the k-means objective function $\{y_i\}_{i=1}^{N} \leftarrow \min \sum_{i=1}^{N} \sum_{k=1}^{2} |f_i - c_k|$ where each feature vector is assigned a cluster label $\{y_i\}_{i=1}^{N}$ on the basis of it minimum distance from the respected centroid $\hat{c}_k$, where c is the centroid of the kth cluster. In the current configuration, the K parameter of k-means clustering is set to 2 to cluster the images into two groups of either be an electric pole or not an electric pole image patch.

As the pre-trained VGG-16 model used here is trained on a dataset from a different domain, the acquired clusters for RS data will be rather noisy initially. Hence, these clusters can not be used for fine-tuning the deep model straight away for electric pole detection. A selection operation is applied to prune the clusters by removing the noisy samples and filtering out the relevant features to tackle this situation. This is done by first calculating a similarity score between each sample, and their respective cluster centroid. Then, the a specific count of samples having a greater similarity scores are selected and are designated as "reliable samples". This specific count increases over fine-tuning iterations. With this pruning, only samples that are close to the centroid (or in other words, samples that are sufficiently similar to the centroid) are selected. This filtering mechanism helps VGG-16 to focus on learning the prominent features of the cluster, avoiding the noise and outliers. The "reliable samples" chosen by the selection operation are then used for fine-tuning the deep model, using the cluster-IDs as pseudo labels. The model fine-tuned in this manner is used for feature extraction in the next iteration. As the fine-tuning is carried-out on in-domain data, the resulting model is now better adapted to working with RS imagery, thus clusters generated in the next iteration would be comparatively better (that is, less noisy). With every iteration, the model learns the prominent features using pseudo labels generated by clustering.

A. Reliable Sample Selection

In their study, Abid et al. [28] use a λ threshold parameter for the similarity score to determine which samples to include in the reliable set. Any sample with a similarity score above the threshold is considered reliable and thus is included in the reliable set. The value of the λ parameter is decided through empirical testing and will vary depending on the dataset used.

One potential issue with using a set threshold value for similarity is that one of the clusters can have much more homogeneous features, giving it higher similarity scores. That cluster would then contribute many more reliable samples than the other(s), causing an imbalance in the set of reliable samples.

As a solution to this problem, in this work, we propose an alternative method to determine which samples are reliable, by focusing on the number of reliable samples rather than the threshold value. In effect, for each iteration, we set the algorithm to select a preset number of the most reliable samples from each cluster. The exact algorithm is presented below.

The number of reliable samples (n) from each cluster is set to the minimum of the following values:

- 75% *of the number of samples in the cluster*
  This limit is motivated by the assumption that the least similar 25% of the samples are likely to be noise, so far removed from the "best" samples that they are unlikely to contribute much when training the network.

- 3% *of the total number of samples in all clusters, multiplied by the iteration number*
  This makes the number of reliable samples start low (choosing only the most reliable samples), and increase gradually with each iteration.

This means that the number automatically increases for each iteration and that both clusters contribute an equal number of samples to the "most reliable samples" dataset used for fine-tuning. The reliable sample count increases by 3%, until it reaches the 75% cap, and once it reached this cap, it does not increase any longer. The 3% figure has been shown to work well on the electric pole dataset used in this work, but a lower value may yield better results for larger datasets.

B. Implementation Details

For fine-tuning the deep model, the Stochastic Gradient Descent optimizer was used with a learning rate of 0.0001 and momentum of 0.9, and the categorical cross-entropy loss function. Experiments are conducted on a Windows 11 computer with 32GB of RAM and a GeForce 1660 Super GPU with 6GB of RAM. The code was written using Python 3.10.4, TensorFlow 2.8.0, scikit-learn 1.0.2, and Pandas 1.4.2 (as well as various other packages). Training times naturally varied depending on the number of reliable samples, but an entire 10-iteration run typically took less than 30 minutes due to the small dataset.

IV. Experimental Validation

The proposed updated UCL is analyzed by conducting extensive experimentation in different configurations. All experiments are performed at least five times on the electric tower dataset, thus the reported results are the average scores attained from multiple experiments with respective standard deviation (σ). Multiple experimentation allowed us to perform significance test on obtained results, i.e., t-test. The discussion of experiments and results are divided into four parts, namely 1) Direct testing of electric pole dataset on ImageNet weights, 2) Supervised and unsupervised VGG-16 fine-tuning, 3) Cluster analysis, and 4) Error analysis.

A. Direct Testing on ImageNet Weights

Initially, the performance of pre-trained (ImageNet) VGG-16 is computed with an electric tower dataset. VGG-16 with ImageNet weights is directly tested on the test set of the electric tower dataset. In this direct testing, we have considered two settings. First, we extract the feature maps of the last convolution layer of VGG-16 and apply k-means clustering to them. Second, we train only the last classification layer of VGG-16 for two epochs with cluster labels generated by the k-means algorithm (in the previous setting). For both settings, we compute the performance of VGG-16 with ImageNet weights on the electric pole dataset. The experiments are conducted in a 5-fold setting. The results of VGG-16 with ImageNet weights are reported in the first and second row of Table I. The first and second row in the table shows the mean and standard deviation (σ) of Precision, Recall, and F1-Score. It can be seen that clustering extracted feature maps from pre-trained VGG-16 by k-means gave an F1-Score of 75.68%. Only training the last layer with pseudo-labels generated by the k-means clustering algorithm improved the F1-Score by 3%, i.e., 78.63% with 2.21% of σ on the electric pole dataset.

B. VGG-16 Fine-tuning

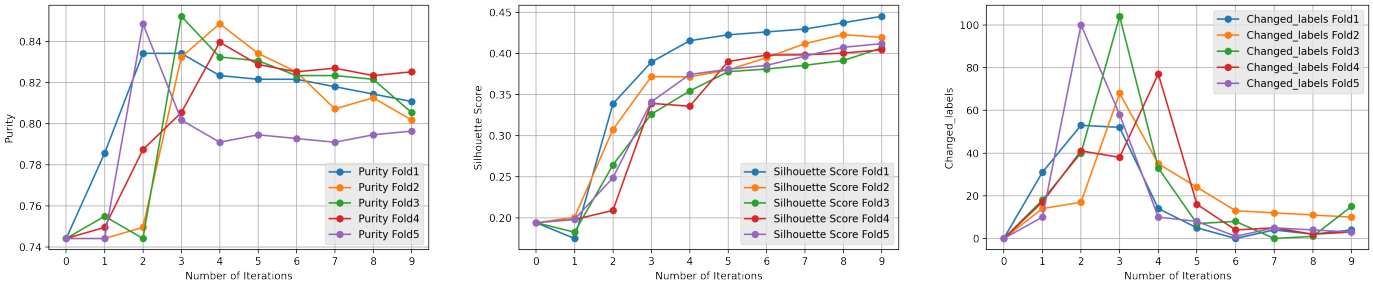
1) Supervised fine-tuning: In general, supervised models perform better than unsupervised ones on a specific data as they are trained using the true labels, whereas unsupervised ones try to learn based on prominent features in the datasets. The performance of the unsupervised framework has been evaluated, considering the supervised model as the benchmark. VGG-16 is trained for 50 epochs in the supervised setting with early stopping criteria on validation loss. The last row in Table I reports the mean and σ of fine-tuned VGG-16 in a supervised manner. The supervised model gave an average F1-Score of 95.58% with 2.35% σ on the electric pole dataset. F1-Score obtained from the supervised fine-tuned model is the highest possible and will be used as this study's benchmark for unsupervised fine-tuning.

2) Unsupervised Fine-tuning: We have analyzed the progressive learning behavior of an unsupervised framework that learns the variations in the dataset with the assumption that the ground truth is unavailable. A clustering algorithm generates the pseudo-labels to train the deep model at every fine-tuning iteration. Initially, UCL is used without updating the selection operation and using fix threshold value [28] for extracting the reliable samples present near the centroids. The UCL framework [28] gave an average F1-Score of 63.59% with 1.22% σ with a small electric pole image dataset(see Table I 3rd row). This F1-Score is worse than the one obtained direct inference on ImageNet weights. The 1st iteration of fine-tuning the model is inclined towards extracting only 1 sample from a cluster and the other with a few hundred images. This sample selection leads to the poor fine-tuning of the deep model.

In this work, we have updated the selection operation, which selects the equal count of samples from each cluster to avoid the imbalanced class distribution problem. We have considered two types of experimentation settings; i) partially freeze the deep model for fine-tuning and ii) fine-tune the entire deep model. In the first setting, VGG-16 is fine-tuned in an unsupervised manner with its initial 12 layers frozen. As it can be seen in the Table I forth row, the UCL with updated selection operation reported an average F1-Score 87.32% with 2.63% σ. The obtained F1-Score shows that the updated UCL is significantly better than standard UCL ($p - value < 0.000005$), and only VGG-16's classification layer trained with pseudo-labels. We then fine-tuned the entire

	Precision		Recall		Macro F1-Score	
	Mean	σ	Mean	σ	Mean	σ
K-means Clustering	0.8524	0.0000	0.7615	0.0000	0.7568	0.0000
VGG-16's last layer trained with Pseudo Labels	0.8066	0.0356	0.7892	0.0215	0.7863	0.0221
UCL with fixed lambda [28]	0.8152	0.0031	0.6661	0.0088	0.6359	0.0122
UCL with flexible lambda early layers frozen (Our)	0.8791	0.0215	0.8716	0.0270	0.8732	0.0263
UCL with flexible lambda (Our)	0.9023	0.0127	0.8999	0.0035	**0.8998**	**0.0032**
Supervised	0.9556	0.0192	0.95672	0.0000	0.9558	0.0236

TABLE I: Shows results conducted in different configurations to show the learning capability of UCL in comparison with clustering and supervised training.



(a) Purity graph of generated clusters during fine-tuning iterations UCL with updated selection operation.

(b) Silhouette Score graph of generated clusters during fine-tuning iterations of the UCL with updated selection operation.

(c) Change in the cluster labels of samples over fine-tuning iterations of the UCL with updated selection operation.

Fig. 3: Cluster analysis during the UCL with updated selection operation fine-tuning iterations.

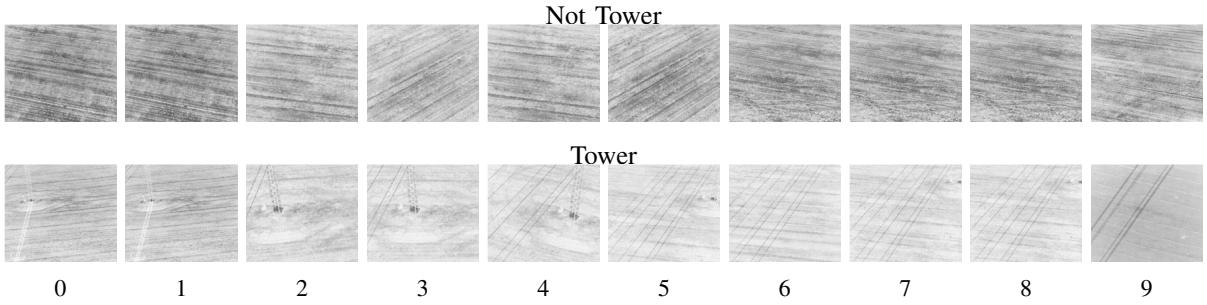


Fig. 4: Samples closest to the centroids of "Not Tower" and "Tower" clusters generated using k-means over 10 fine-tuning iterations of UCL with updated selection operation. The centroids shown here are obtained from Fold1.

deep model in updated UCL. The performance in terms of accuracy improved further by 2%, by reporting an average F1-Score 89.98% with 0.32 σ, which is also significantly ($p - value < 0.000005$) better than the standard UCL and direct inference on ImageNet weights. The proposed updated UCL removed the imbalanced class distribution problem and can learn the prominent features from the electric pole dataset with limited samples without needing expert labeling.

C. Clustering Analysis

The generated clusters using k-means clustering in an unsupervised framework are further analyzed to monitor the training process. The progressive fine-tuning of the unsupervised framework is observed for ten iterations. In the beginning, the clusters for the electric tower dataset are created with the features extracted from pre-trained VGG-16 on ImageNet dataset, a domain different from our target domain that is UAV imagery. As a result, the generated clusters of remotely sensed electric tower images are not compact. Two measures are used to analyze the quality of the clusters: Purity and Silhouette scores. Purity, on the one hand, is a measure that uses the true labels to calculate the correctly classified labels over the total number of samples in the cluster. On the other hand, the Silhouette score calculates the compactness of the clusters based on the distance between the samples within the cluster and the neighboring cluster(s).

Figure 3 shows the graph of Purity, Silhouette score, and change in the cluster labels over fine-tuning iterations of fully trained UCL with updated selection operation. It can be seen in Graph 3a that the purity tends to increase over starting three fine-tuning iterations from 0.74 to 0.85 for 5-Fold experiments. This increase is observed from 1 to 4 fine-tuning iterations. Later it remains between 0.79 to 0.83 for the rest of the fine-tuning iterations. The Silhouette scores in Graph 3b show a slight decrease over the first fine-tuning iteration and then a sudden increase over the second and third iterations of fine-tuning. This increase indicates that the clusters are better separable over fine-tuning iterations. After the third iteration, the increase in the Silhouette scores is slow. Though an increase can be observed in the graph, its maximum goes to 0.45 value in 10 fine-tuning iterations. Based on the Silhouette score, the models fined-tuned at second, third, and fourth iterations are most likely to be the reasonable models to retain. In graph 3c, the maximum change in the cluster samples' labels is observed at the second, third, and fourth iterations for respected folds of fine-tuning. Their Silhouette score is also showing a remarkable increase at respective iterations. All the models have shown the drastic change in labels at either the second or third fine-tuning iteration except Fold4. Fold4 with a red curve has shown drastic change in cluster labels at two iterations; 2 and 4. In parallel, its Silhouette score increases at iteration 2 but decreases at iteration 4. Hence, considering iteration 2 will be the better choice for Fold4.

Based on these observations, the fine-tuned models of second or third iterations of the respected folds of UCL with updated selection operation seems a reasonable choice to retain. That is the second fine-tuning iteration for folds 1, 4, and 5, and the third fine-tuning iteration for folds 2 and 3, respectively.

The centroids generated in the training process of UCL updated with selection operation in Fold1 are shown in Figure 4. In the early iterations of fine-tuning, the model successfully learns the electric pole features from the data and generates clusters around these characteristics. After some iterations of fine-tuning, the model deviates to other prominent features present in the dataset, like electric wires. The dataset used in this study is complicated because it has similar prominent features in both categories of the tower and not the tower, like background and electric wires. These similar characteristics raise the difficulty for the unsupervised model to only focus on the specific features of electric poles (mostly cropped in the image or occluded with electric wires) by ignoring the other prominent characteristics in the data. Adding to it, the sample count for fine-tuning the UCL is considerably small (556 samples). With these limitations, the model can still learn the desired tower category in the first five fine-tuning iterations by creating the centroid around respected features of the tower and not tower images.

D. UCL Fine-tuning Analysis

UCL with updated selection operation has been fine-tuned for 10 iterations leading to 10 fine-tuned deep models. So far, the clustering analysis has shown that either second or third iteration of fine-tuning the respected fold is most suitable to deploy. To ensure, we have tested all the fine-tuned models over ten iterations with the test dataset, see Graph 5. It can be seen that all the five folds reported F1-Score below 50% on ImageNet weights when no training has been initiated. This is because the classification layer is not trained so far. After the first fine-tuning, the F1-Score increased to almost 75% for folds 2 to 5 and 83% for fold 1. In the next couple of fine-tuning iterations, the F1-Score increased to 90%. This increase in the F1-Score indicates that the model is learning the prominent features of utility poles. For the later iterations of fine-tuning UCL, the F1-Scores remain between 81% to 85%. Like clustering analysis, the results on test data also indicate that the second fine-tuning iteration model is most suitable for folds 1, 4, and 5, and the third fine-tuning iteration model for folds 2 and 3, respectively.

V. Discussion and Critical Analysis

When using supervised learning, the results were quite good, with an average F1-Score of 95.58%. Having access to ground truth labels when training the deep model enables it to learn the relevant features effectively. We do not need labeled data for an unsupervised approach, like UCL. With updates in the selection operation of UCL, we achieved an average F1-Score of 90%, but there is a small trade-off of 5% in F1-Score performance compared to supervised models. Let us deeper analyze the proposed UCL with an updated selection operation.

The proposed UCL with updated selection operation is further monitored by observing the reliable sets and similarity

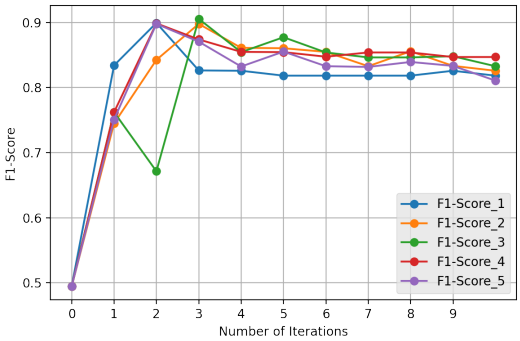


Fig. 5: The F1-Scores graph of fine-tuning iterations UCL with updated selection operation on test dataset of utility poles.

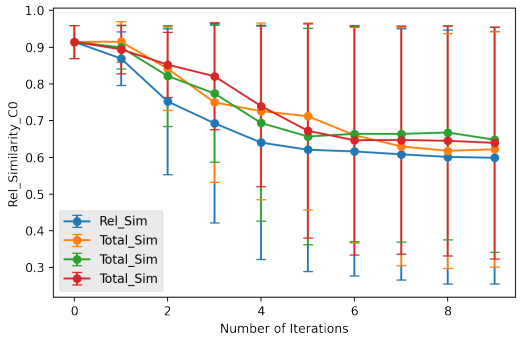


Fig. 6: Mean and standard deviation of similarity scores of reliable samples in cluster 0 over 10 fine-tuning iterations of UCL with updated selection operation.

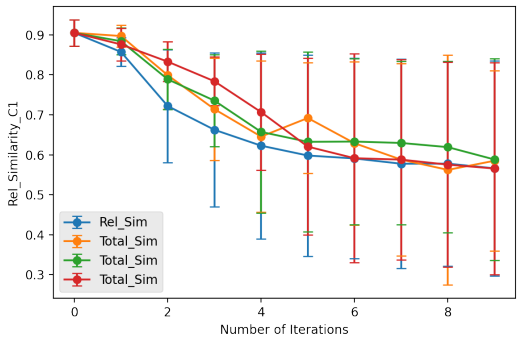


Fig. 7: Mean and standard deviation of similarity scores of reliable samples in cluster 1 over 10 fine-tuning iterations of UCL with updated selection operation.

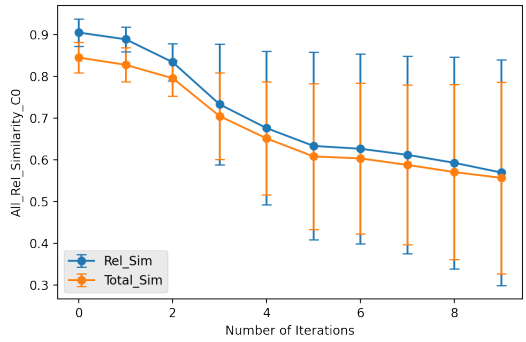


Fig. 8: Mean and standard deviation of similarity scores of reliable samples in cluster 1 over 10 fine-tuning iterations of UCL with updated selection operation.

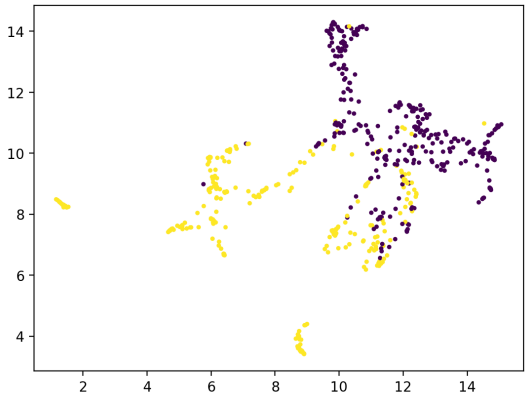


Fig. 9: UMAP with true labels on best iteration of UCL with updated selection operation.

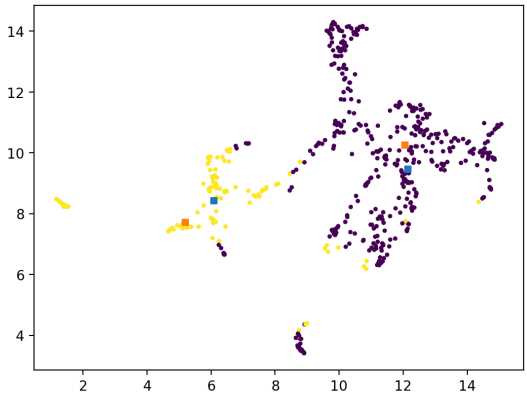


Fig. 10: UMAP with pseudo-labels of k-means clustering on best iteration of UCL with updated selection operation.

scores. In standard UCL, a similarity threshold is set to extract reliable samples from clusters, where samples selected from one cluster have a high tendency to be must larger or lesser in count than the samples extracted from another cluster. To avoid this situation of imbalanced class distribution, we made the threshold flexible by selecting the exact count of samples from each cluster based on similarity. Figure 6 and 7 shows mean and standard deviation of similarity score of reliable samples in cluster 0 and 1, respectively. In the initial iteration of fine-tuning, the most reliable samples that are 90±5% similar to the centroids are selected. In the following iterations, the σ increases in graphs of both clusters. One reason is that the sample count is increasing linearly with every fine-tuning iteration. The updated UCL is increasing the reliable samples linearly over fine-tuning iterations. An interesting observation can be seen in Figure 7 of cluster 1. In the later iterations, the mean and standard deviation gradually deviate from the maximum possible similarity, i.e., 1.0. One of the possible reasons is that the generated clusters are not separable based on Euclidean distance. Also, we are selecting samples based on the similarity with the centroid. In other words, the selection is made based on the Euclidean distance from the centroid. This limitation of Euclidean distance raises the demand for two alternative solutions; i) A clustering method that is independent of Euclidean distance and ii) A selection operation that is not dependent on Euclidean distances of samples from their respective centroids.

In Figure 8 we have visualized the similarity score of reliable samples and all samples in cluster 1 for the third fine-tuned model of fold 1. In the early iterations of fine-tuning, the reliable sample set has a comparatively higher average similarity than all samples in the cluster. With every subsequent iteration, the gap in the average similarities of both sets is decreasing. This observation is because more and more samples are becoming part of the reliable set. In the last iterations, majority of the cluster samples are considered in the reliable set.

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimensionality reduction technique that can be used for high dimensional data visualization. We have used UMAP to visualize high-dimensional extracted features of 4608 dimensions in 2D with true-labels and pseudo-labels, see Figure 9 and 10. For this visualization, we have considered the best fine-tuning iteration of UCL updated with selection operation in fold 1. In Figure 9, it can be seen that samples based on true labels, yellow and purple, are somewhat separable, but there is a small chunk of data that is overlapping in both categories. The purple cluster seems to be comparatively more compact than the yellow. The yellow cluster can be seen as roughly three chunks in one cluster. First small chunk of the yellow cluster is near the left edge of the graph, second chunk in the middle and third chunk in the lower left. The first chuck seems to have no outlier. The second chunk has a few purple samples. Whereas, the last chunk has quite some examples of purple cluster, showing a big overlap. This seems to be an region where both classes have common prominent features.

Figure 10 shows the sample distribution in k-means clustering, one cluster represented with purple and the other with yellow. The blue squares are the centroids of clusters, and orange are the closest samples to their respective centroids. Both clusters show a similar distribution as seen in the true labels (Figure 9). The model learned the first and second chunks of the yellow cluster. But the third chunk which has quite few examples of purple cluster in true labels (Figure 9) seems hard for the model to learn. The model learned some examples from the thrid chunk as yellow class but most of

these samples it learned as purple class. This misclassification is because this area is mostly overlapping with the purple category.

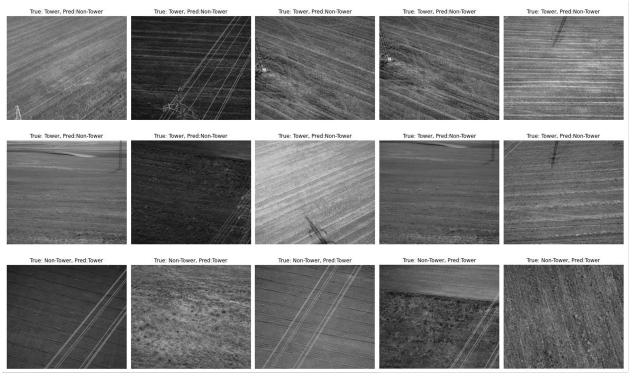


Fig. 11: Model 3 Misclassified Images, Output Layer Prediction

Some of the misclassified samples of updated UCL are visualized in Figure 11 where the first two rows represent failure cases of Tower images and third-row shows misclassified Non-tower images. It can be seen in the figure that most of the images misclassified as non-tower have very little representation of tower in them, in several cases only a shadow of a tower is visible. It is because images with just the shadow are labeled as tower images in the dataset. It can also be noted that some images having a cropped tower is occluded by electricity wires. The images misclassified as tower images, on the other hand, usually (but not always) have quite prominent wires. The electricity wires are quite prominent and are present in both categories; tower and not a tower. This may confuse the model to classify these images.

VI. Conclusion

The supervised deep models need a massive corpus of labeled data to train the deep model, which is time and labor-intensive and demands domain knowledge. This paper proposes an unsupervised deep learning framework for electric pole detection from grayscale high spatial resolution imagery. This architecture removes the requirement of data labeling. The unsupervised framework learns the characteristics of electric poles using the pseudo-labels generated from clustering. Such frameworks mostly select the samples from the clusters on a threshold basis, often leading to imbalanced class distribution problems. We have proposed a new selection operation technique that avoids imbalanced class distribution problems in filtering the samples from the clusters. These filtered samples are used for fine-tuning the CNN model with cluster IDs. A dataset of eclectic poles provided by Mention Fuvex (a Spanish startup utilizing long range drones for power line inspection) is used to prove the hypothesis. The experimental validation on provided utility pole dataset shows that the proposed solution can learn the prominent features of utility poles with generated cluster labels compared with direct clustering. The paper shows a statistically significant improvement of about 12% in comparison with direct inference on ImageNet weights by reporting F1-Score of 89.98% (see Table I second and fifth rows. However, a big room for future direction is still open. The selection operation of UCL can be explored by combining the fixed threshold method used in Abid et al. [29], [28] and proposed selection operation of equal selection reliable samples. Further, we have worked with a few hundred samples. One way to improve performance is to increase the number of samples. The dataset used has high spatial resolution images of quite a big size. They are resized to make them suitable for the deep model. Instead of resizing, the images can be divided into smaller patches to yield fruitful results.

Acknowledgment

The authors would like to thank a ”Spanish company utilizing long-range drones for power line inspection” for providing the dataset of utility poles.

References

[1] M. M. Alam, Z. Zhu, B. Eren Tokgoz, J. Zhang, and S. Hwang, “Automatic assessment and prediction of the resilience of utility poles using unmanned aerial vehicles and computer vision techniques,” *International Journal of Disaster Risk Science*, vol. 11, no. 1, pp. 119–132, 2020.

[2] A. Joukoski, K. Portella, O. Baron, C. Garcia, G. Vergés, A. Sales, and J. De Paula, “The influence of cement type and admixture on life span of reinforced concrete utility poles subjected to the high salinity environment of Northeastern Brazil, studied by corrosion potential testing,” *Cerâmica*, vol. 50, pp. 12–20, 2004.

[3] S. Das, B. Storey, T. H. Shimu, S. Mitra, M. Theel, and B. Maraghehpour, “Severity analysis of tree and utility pole crashes: Applying fast and frugal heuristics,” *IATSS research*, vol. 44, no. 2, pp. 85–93, 2020.

[4] A. A. Oliveira, M. S. Buckeridge, and R. Hirata, “Detecting tree and wire entanglements with deep learning,” *Trees*, pp. 1–13, 2022.

[5] R. Jenssen, D. Roverso *et al.*, “Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning,” *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 107–120, 2018.

[6] B. Cetin, M. Bikdash, and M. McInerney, *Automated electric utility pole detection from aerial images*. IEEE Southeastcon, 2009.

[7] K. Sarabandi, L. Pierce, Y. Oh, and F. T. Ulaby, “Power lines: Radar measurements and detection algorithm for polarimetric SAR images,” *IEEE transactions on aerospace and electronic systems*, vol. 30, no. 2, pp. 632–643, 1994.

[8] K. Sarabandi and M. Park, “Extraction of power line maps from millimeter-wave polarimetric sar images,” *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 12, pp. 1802–1809, 2000.

[9] Y. Jwa and G. Sohn, “A piecewise catenary curve model growing for 3D power line reconstruction,” *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 12, pp. 1227–1240, 2012.

[10] E. Kim and G. Medioni, “Urban scene understanding from aerial and ground LIDAR data,” *Machine Vision and Applications*, vol. 22, no. 4, pp. 691–703, 2011.

[11] Y. Wang, Q. Chen, L. Liu, D. Zheng, C. Li, and K. Li, “Supervised classification of power lines from airborne LiDAR data in urban areas,” *Remote Sensing*, vol. 9, no. 8, p. 771, 2017.

[12] 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.

[13] R. Bernstein and V. Di Gesù, “A combined analysis to extract objects in remote sensing images,” *Pattern recognition letters*, vol. 20, no. 11-13, pp. 1407–1414, 1999.

[14] 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.

[15] A. H. Khawaja, Q. Huang, and Z. H. Khan, “Monitoring of overhead transmission lines: a review from the perspective of contactless technologies,” *Sensing and Imaging*, vol. 18, no. 1, pp. 1–18, 2017.

[16] G. Yan, C. Li, G. Zhou, W. Zhang, and X. Li, "Automatic extraction of power lines from aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 387–391, 2007.

[17] L. Matikainen, M. Lehtomäki, E. Ahokas, J. Hyyppä, M. Karjalainen, A. Jaakkola, A. Kukko, and T. Heinonen, "Remote sensing methods for power line corridor surveys," *ISPRS Journal of Photogrammetry and Remote sensing*, vol. 119, pp. 10–31, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[19] M. Sabatelli, M. Kestemont, W. Daelemans, and P. Geurts, "Deep transfer learning for art classification problems," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[20] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[21] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt, "Roadtracer: Automatic extraction of road networks from aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4720–4728.

[22] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 172–181.

[23] J. Yu, Z. Wang, A. Majumdar, and R. Rajagopal, "DeepSolar: A machine learning framework to efficiently construct a solar deployment database in the United States," *Joule*, vol. 2, no. 12, pp. 2605–2617, 2018.

[24] B. Huang, J. Yang, A. Streltsov, K. Bradbury, L. M. Collins, and J. M. Malof, "GridTracer: Automatic mapping of power grids using deep learning and overhead imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 4956–4970, 2021.

[25] W. Zhang, C. Witharana, W. Li, C. Zhang, X. Li, and J. Parent, "Using deep learning to identify utility poles with crossarms and estimate their locations from google street view images," *Sensors*, vol. 18, no. 8, p. 2484, 2018.

[26] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[27] A. Ul-Hasan, F. Shafaity, and M. Liwicki, "Curriculum learning for printed text line recognition of ligature-based scripts," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1001–1005.

[28] N. Abid, M. Shahzad, M. I. Malik, U. Schwanecke, A. Ulges, G. Kovács, and F. Shafait, "UCL: Unsupervised Curriculum Learning for water body classification from remote sensing imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102568, 2021.

[29] N. Abid, M. I. Malik, M. Shahzad, F. Shafait, H. Ali, M. M. Ghaffar, C. Weis, N. Wehn, and M. Liwicki, "Burnt Forest Estimation from Sentinel-2 Imagery of Australia using Unsupervised Deep Learning," in *2021 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2021, pp. 01–08.

[30] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[31] R. M. S. Bashir, M. Shahzad, and M. Fraz, "Vr-proud: Vehicle re-identification using progressive unsupervised deep architecture," *Pattern Recognition*, vol. 90, pp. 52–65, 2019.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.