15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

A new deep learning architecture for detection of long linear infrastructure

Jayavardhana Gubbi, Ashley Varghese, Balamuralidhar P
TCS Research and Innovation, Bangalore, India
j.gubbi@tcs.com, ashley.varghese@tcs.com, balamurali.p@tcs.com

Abstract

The use of drones in infrastructure monitoring aims at decreasing the human effort and in achieving consistency. Accurate aerial image analysis is the key block to achieve the same. Reliable detection and integrity checking of power line conductors in a diverse background are the most challenging in drone based automatic infrastructure monitoring. Most techniques in literature use first principle approach that tries to represent the image as features of interest. This paper proposes a machine learning approach for power line detection. A new deep learning architecture is proposed with very good results and is compared with GoogleNet pre-trained model. The proposed architecture uses Histogram of Gradient features as the input instead of the image itself to ensure capture of accurate line features. The system is tested on aerial image collected using drone. A healthy F-score of 84.6% is obtained using the proposed architecture as against 81% using GoogleNet model.

1 Introduction

Infrastructure inspection is a tedious manual task that is undertaken because of current industry necessities and to prolong life of infrastructure. The industrial needs for insurance, maintenance and Quality of Service (QoS) purposes ensures application of newer technologies in order to increase consistency and reduction of manual labour. Specifically, if the assets are distributed in wide geographies and hard to reach places, the need for industrial automation becomes key. In such scenarios, emerging rotor based UAVs will play critical role in gathering the much needed data from wide perspectives. However, they come with many challenges including navigation, data acquisition, processing and decision making. For instance, monitoring power lines is a very big challenge. They span across hundreds and thousands of kilometers. This challenge exponentially increases when there is a change in geography and terrain. Downtime is undesirable for a power line grid. It is inherent that all pre-emptive maintenance and repair needs to be done on these infrastructure before it actually fails. Inspection using aerial imagery is the most feasible method for accomplishing the task. With vast improvements in hardware and embedded systems aided by better understanding of aerodynamics, unmanned aerial vehicles - particularly rotor based drones (quadcopter, hex copter and octocopters) are becoming increasingly common.

Processing huge amounts of aerial images or video data accurately will create many unforeseen challenges. Manual assessment and inspection of thousands of images will be cumbersome and prone to human error. The aerial images captured using Drones will contain not just the images of the power lines, but also captures highly variable background like vegetation, roads, different texture of soil etc. This makes the aerial image processing very challenging because of the background heterogeneity. The region occupied by line in such images will be very less compared to the background thereby creating a highly biased datasets towards negative samples. With all this challenges posed, and ten and thousands of images have to be analysed, it becomes inherent that we use an automated approach. Traditional approaches either from first principals or from shallow machine learning methods have been attempted in the past with success [1] but adapting and extending them to larger range of applications will have its own system challenges.

In this work, the main aim is to develop a new algorithm based on emerging deep learning that has shown a lot of promise in object detection scenarios both in terms of accuracy as well as in terms of reproducibility. We first focus on the most important and challenging class of power line conductor detection due to its biased nature of the dataset and the background that it is likely to inherit at the recognition stage. The success of Deep Learning can be attributed to the new greedy layer-wise learning proposed in addition to rectified linear unit (ReLU) as activation function and the use of GPUs that makes learning faster and efficient [2, 3]. There are many deep learning architectures proposed recently in literature for image classification. In this paper, the more classical approach of fine tuning the existing models as well as a new architecture is proposed and compared.

2 Related work

For power line monitoring, fewer groups are focused and most of them use drones as a medium to capture the data. Sharma *et al.* [1] segments edges using point pair as seeding point and grow the contour along the linear feature boundary, which appears to be the most sensible way. Ceron *et al.* [4] propose a circle based search for detecting line segments using canny and steerable filters. Ramesh *et al.* [5] use pixel intensity based k-means clustering followed by morphological operations. All these methods are based on first principles and use the characteristics of the power line for detecting them. To the best of our knowledge, this is the first work that attempts to use deep learning for power line detection. This is particularly relevant as machine learning gives flexibility of extending it to multi object classification.

The performance of object classification task has improved considerably in the last few years because of Deep Convolution Neural Network (CNN) [3]. In 2012, Krizhevsky *et al.* [2] developed a new CNN architecture for a 1000 class classification problem using

data from ImageNet [6]. Later many architectures have been proposed using this core idea. Google proposed a deep inception network called GoogleNet [7] for improved classification and detection performance. This model is referred to as pre-trained model in this paper that can be used in other applications as will be explained below.

3 Deep Learning for aerial image analysis

In this paper, we propose a deep learning architecture for power line detection from aerial images. We use two approaches a) using a pre-trained model as feature extractor; and b) develop our own architecture that gives flexibility in terms of training time and real-time operation.

3.1 Pre-trained models

In this approach, an existing pre-trained CNN model is taken and the last few layers are retrained with new set of object categories. Here, the existing model has been trained for classifying a set of object categories. Further, model has a set of values derived to make a decision for distinguishing between existing trained classes. That is, it has enough feature information to distinguish between all the classes, therefore the same information can be reused for the new set of object categories. Re-training the whole network again for a new set of data needs large data set and more time. By doing so, the knowledge gathered from a larger more generic problem can be *transferred* to the new domain. The benefit is the reduction in training time as well as the amount of data required for deriving correct representation. Again the decision is based on the closeness of the new data to existing data models. If it is entirely different from the existing class and if we have enough data set, then the whole network has to be trained again. In case of power line identification, existing training data has images with ropes and strings although an exclusive class is not available for rope detection. This implies that the model already has information and the final layer retraining is required to tap the information for the new classification.

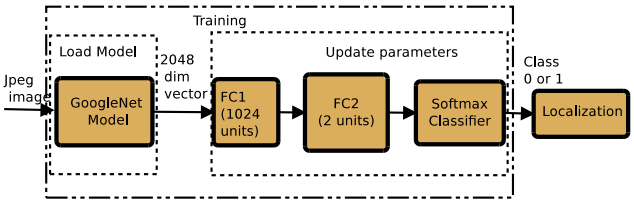


Figure 1: Architecture for pre-trained models

GoogleNet, a 22 layer architecture, is used as pre-trained model in this work. The way the pre-trained model architecture is adapted in this work is shown in Fig. 1. Fully connected layers are added at the end of the GoogleNet model and retrained on power line data set. The filter parameters of the other previous layers are kept untouched and reloaded at training time. The input to this architecture is the raw image pixels and output is the binary classified output. In this architecture, GoogleNet pre-trained model act as a feature extractor from the new class of images. Further, the layer just before the final classification layer is having an output of 2048 dimensional vector. It is the set of values given by the pre-trained model for each input image. These values are the input for newly added fully connected layer with hidden units of 1024 neurons. ReLU is used at the end of the layer as activation function. The final layer is a binary classifier which classifies whether any power line is present in the image or not. A *softmax* classifier is used at the final layer for classification.

3.2 New deep architecture for power line detection

We propose a deep CNN architecture with fully connected layers for power line detection. CNNs is known for its architecture that infers spatial structure of the image and it is widely adapted for image classification tasks. The proposed architecture has four CNN layers and two fully connected layers followed by a Softmax classifier. The network is trained with the popular Histogram of Gradient (HoG) [8] features. The architecture diagram of the network is shown in figure 2. Generally, CNN learns the representation itself from raw image and extracts the feature based on the optimization of loss function. It has been shown to make the hand-crafted features obsolete. However, in our application, power line is very sparse compared to the complex background. Training with HoG feature makes the classification task effective since the model would be trained with the feature that we want to learn for line detection. HoG is the histogram of the gradient vector that represents edge orientation and is illumination invariant.

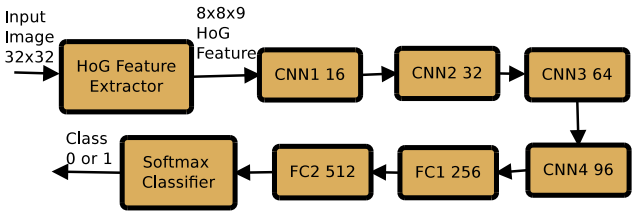


Figure 2: Architecture diagram of proposed deep network

During training, input to the network is 8×8 size HoG features with 9 channels. Generally, first layers captures the low level orientation details from HoG features. As the layer goes higher, it captures the high level features that is the result of one or more features at earlier stages. Network is trained for different gradient orientations. The network configuration details of the proposed network is given in the Table 1. The input HoG features is passed through CNN and fully connected layers for training. The first layer uses the filter with receptive field of 5×5 and the depth of 16. It covers the small local region and captures the gradients of all the orientation. The stride is fixed as 1 for all the layers. A ReLU is provided at the end of all the convolutional layers as an activate function. Max pooling is not used in this architecture since the HoG image size is small compared to the original image size 32×32. The second layer and third layer have the filter with kernel size of 3×3 and the depth of 32 and 64 channels respectively. Further, it has a stack

Table 1: Proposed network architecture

Layer	Filter	Channel	Output size
CNN1	5x5	16	8x8x16
CNN2	3x3	32	8x8x32
CNN3	3x3	64	8x8x64
CNN4	1x1	96	8x8x96

of two 3×3 filter which adds two non-linear activation function to the network. The addition of non-linearity makes the decision function more discriminative. The fourth CNN layer has the kernel size of 1×1 and the depth of 96. It changes the dimensionality of the feature maps. A *softmax* classifier is provided at the final layer for power line classification.

4 Experiments

An Ubuntu based workstation with configuration Intel core i7 @3.4Gx8, 32GB RAM and NVIDIA GM204GL [Quadro M4000] GPU card is used for training and testing purpose. Tensorflow, a deep learning library with python support is used for implementing deep learning network. Tensorflow [9] is an open source machine learning library from Google, which supports distribute computing among different GPUs.

Network is trained on the real data set collected by drone. Drone with camera at the bottom is flown over the high power transmission line for aerial images. The image contains the electric components like insulator, tower and line; also the background of trees, roads and building. The resolution of the image data is 1280×960. The collected images are annotated using LabelMe [10]. The ground truth binary mask is generated from annotated data and it is used for annotating the data set. For capturing the localization information, the images are divided into patches of size 32×32 and annotated with two classes "Line present" and "No line present". The labels are generated using first principles as explained in Sharma *et al.* [1]. The image patch approach ensures detection of power line as well as in localising them. The input image and ground truth image is shown in Fig. 3. It should be noted that the labels are given only to the power lines that have strong features and are closer to the camera. The power lines away from the camera are missed out and they are treated as noisy labels in our deep learning architecture. The whole dataset contains around 62400 image patches as both positive and negative samples together. The data set is partitioned into training, validation and testing data in the ratio of $7 : 1.5 : 1.5$. The same dataset and set up was used for both the training approaches.

Figure 3: Sample aerial images and its ground truth

As discussed earlier, two different approaches are used for training. The first is to fine tune the pre-trained model GoogleNet for the new data set. The trained models are available in protocol buffer (.pb) file format. During the building of computation graph, all the saved parameters are restored to the graph except the tensor node 'softmax:0', which is the classifier node for pre-trained model. The training image size is fixed to 32×32. The feature value of the each training image is extracted and stored at the tensor node $pool_3 : 0$, the layer just before the classification layer. It is a 2048 dimensional vector, which is fed as input to the newly added fully connected layer. These neurons are then connected to the 1048 neurons in the next layer. The *softmax* is used at the last layer for classification. During training, a mini-batch of 100 images are used at each training step of 4000 iterations.

In the second approach, training image is divided into image patches of size 32×32 similar to the GoogleNet input. During HoG feature extraction, 32×32 image is divided into 64 blocks and each block is having a cell of 4×4 pixel size. Gradient vector is computed for each cell and a histogram of 9 bins of gradient orientation is generated. This results in a feature vector of 576 ($8 \times 8 \times 9$) dimensional vector for each patch. During training, the HoG feature from all the training images are extracted and labeled with corresponding classes. In this approach, a mini-batch of 125 images are empirically chosen at each training step of 1000 iteration.

5 Results and Discussion

A total of 52 images are used for training and remaining image for testing. Using GoogleNet model, the patch size is fine tuned and the results are summarised in Table 2. Although the results for 64×64 appears better, the resolution of results appears very bad and is not suitable for further processing. Based on the result, 32×32 is chosen for all our experiments.

Table 2: Results of GoogleNet model for different patch sizes

Patch Size	F-Score	Accuracy	Precision	Recall
16×16	48.3	86.02	33.98	83.24
32×32	80.99	90.41	73.57	90.07
64×64	83.5	89.66	75.23	94.04

A sliding non-overlaping window of size 32×32 is moved over the image and each patch is evaluated with the trained model. If any patch gets classified as line, it is marked with a rectangle for line localization. The two approaches are compared by measuring the F-score, accuracy, precision and recall as given in the Table 3.

Table 3: Results for both the approaches

Architecture	F-Score	Accuracy	Precision	Recall
HOG	84.6	92.9	83.0	86.3
GoogleNet	80.99	90.41	73.57	90.07

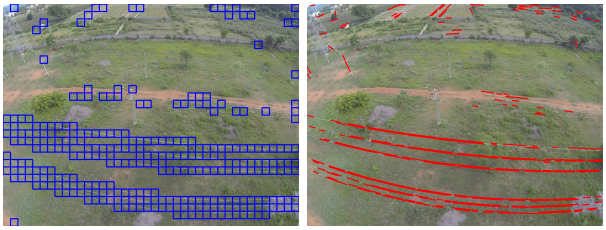
The time taken for training using the two approaches is summarised in Table 4. In addition to the better results obtained using the proposed architecture, the training time is far better than the pre-trained model.

The segmentation result using GoogleNet classifier is shown in Fig. 4a and the resulting line detection for

Table 4: Training time for both the approaches

Architecture	Patch Size	Training Time
HOG	32x32	0.16 min
GoogleNet	32x32	55.65 min

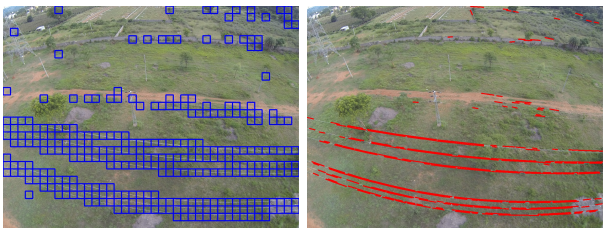
all positive patches is shown in Fig. 4b. The line detection that has been employed for final result is line segment detector (LSD) algorithm [11] that is highly optimised for this application. Similarly, the results using the proposed architecture is shown in Fig. 5.



(a) Line localization (b) After LSD is applied

Figure 4: Results for GoogleNet pre-trained model

The overall results are in favour of building dedicated



(a) Line localization (b) After LSD is applied

Figure 5: Results for architecture with HoG features

architecture if enough data is available and if high accuracies are anticipated. The pre-trained models work reasonably well and is suitable for feasibility analysis. The proposed architecture is not very deep and hence gives us easy implementation on drones and other embedded devices as well. In cases where the features of interest is in a different scale compared to the scene features extracted in case of pre-trained models, newer dedicated architectures may be required. Future work includes addressing affine transform and rotation that may be caused by drone orientation.

6 Conclusion

The main aim of this work is to assess whether the pre-trained models are suitable for all categories of applications as it enables transfer learning. The detection and localisation of the power conductor will enable business objectives such as line counting, detection of conductor snaps, sagging etc and it has been used as an application to test our hypothesis. Two approaches have been taken to detect the power line - using GoogleNet pre-trained model and using a new deep learning architecture. The proposed architecture consists of 4 CNN layers followed by 2 fully connected layer. Aerial images collected from drones are used for training and testing. A F-score of 84.6% is obtained using the proposed architecture against 81% obtained using GoogleNet. The better results using the proposed architecture indicates the need for the use of customised models in certain class of applications where the region of interest is very small compared to its background.

References

[1] H. Sharma, T. Dutta, V. Adithya, and P. Balamuralidhar, "A real-time framework for detection of long linear infrastructural objects in aerial imagery," in *International Conference Image Analysis and Recognition*, pp. 71–81, Springer, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Ceron, F. Prieto, *et al.*, "Power line detection using a circle based search with uav images," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 632–639, IEEE, 2014.

[5] K. Ramesh, A. S. Murthy, J. Senthilnath, and S. Omkar, "Automatic detection of powerlines in uav remote sensed images," in *Condition Assessment Techniques in Electrical Systems (CATCON), 2015 International Conference on*, pp. 17–21, IEEE, 2015.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[8] 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, pp. 886–893, IEEE, 2005.

[9] "TensorFlow: Large-scale machine learning on heterogeneous systems."

[10] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[11] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.