Investigating into Deep Neural Networks for Applicable Hard-hat Wearing Detection in Substations

Tong Li¹, Qipeng Chen², Jinrui Gan* and Peng Wu²

¹ State Grid Liaoning Electric Power Research Institute, Liaoning, 110006, China
² Artificial Intelligence on Electric Power System State Grid Corporation Joint Laboratory (GEIRI), Global Energy Interconnection Research Institute, Beijing 102209, China
Email: ganjinrui@geiri.sgcc.com.cn

Abstract. All staffs are strictly requested to wear hard-hats when working in substations. Various object detection algorithms, especially those based on deep learning, thus have been proposed for the corresponding purpose. A deep learning based-object detection algorithm commonly involves a fundamental neural network which dominates detection performances, so this paper investigates different types of networks’ applicability when utilizing them with a typical object detection algorithm for the monitoring of hard-hat wearing in substations. This is conducted from various perspectives concerned by real-world implementation that includes time consumption, computation speed, precision and more. As a consequence, this study provides a guideline to the selection of the most appropriate deep neural network architectures for the specific monitoring scenario.

1. Introduction
A substation is the main enabler of long-distance electric power transmission from generators to end consumers. It is fulfilled with devices (e.g., transformers, switches and more) of different scales where a device can have tens of components. Every single component has a chance to get defect which may cause equipment failures and rise chances to catastrophic system-level issues, so inspections and maintenance are highly necessary in all substations and have to be proceed frequently, comprehensively and manually at this moment. Though substation robotics have been deployed, for their limited functionalities and lack of intelligence they can only accomplish a relatively small subset of regular substation works, e.g., meter reading. It is foreseen that manual accessing to substations is not avoidable in the next few years. Because devices in a substation are mostly suspended installed, so earthquake and other accidents could cause falling down of these devices. Therefore, all persons are strictly requested to wear hard-hats to prevent them from being hurt. At present, almost all substations are installed with fixed cameras, via which one can see if anyone is wearing a hard-hat or not. However, appointing specific persons to keep eyes on all staffs all the time is impossible and labor-force wasting. As a result, a proper monitoring mechanism is thus absolutely necessary to guarantee that all staffs in a substation wearing hard-hats properly and continuously. Actually, with the rapid development of computer vision techniques in recent years, relative works have been seen published. In [1] a three-step detection framework is proposed which shows the availability to use computer vision techniques for hard-hat wearing detection. First, feature extraction is enabled by Speeded-Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK) and other techniques; second, candidate pixel regions of hard-hats in an image are then proposed by a template matching
mechanism; third, a cascade classifier which was trained with both positive and negative instances beforehand is finally applied to classify whether a region has a hard-hat or not. To make hard-hats more distinguishable, the authors propose to add a customized sticker to all hardhats, while this is not always applicable in a real-world scenario. In [2] a four step detection process that makes no change of a hard-hat itself is proposed. The process follows feature extraction, Support Vector Machine (SVM)-based classification, threshold-based color segment detection and Circle Hough Transform. In [3] a similar process of segmentation, pedestrian classification and helmet detection is proposed for the specific substation hard-hat wearing monitoring scenario. For all these three works, a set of features that can interpret the object of interest are selected manually and carefully beforehand, while selecting features by experience may not be sufficient to precisely interpret an object’ real characteristics. Nevertheless, deep learning-based detection algorithms could significantly improve feature extraction. Deep learning has become well-known to public in 2012 in the ImageNet Large Scale Visual Recognition Competition for its top performance on image classification. A deep neural network extracts the most interpretable features from an image then classifies it accordingly. Since then, more deep advanced learning techniques of higher performances have been proposed where a portion of them solves a more complicated problem, i.e., object detection, which both identifies objects of interests in images and determines their categories. The hard-hat wearing detection perfectly suits the object detection problem.

One of the first well-known deep learning-based object detection algorithms in use is Regions with CNN (RCNN) [4] which follows three steps in sequence: a region proposal step which uses a selective search mechanism to propose pixel regions in an image, a feature extraction step which extracts the feature vector out from each region proposal based on a fundamental deep neural network; and a final classification step which uses SVM to classify a feature vector into the correct category. RCNN is not computational efficiently or the repeating computation of features for each region proposal, so in [5] a more advanced version called Fast-RCNN is proposed to handle this issue. Instead of extracting a feature vector for each region, it directly acquires the feature map for the whole image then obtains the feature vector for each region on need. As seen, both traditional and deep-learning based algorithms above go through separate steps to detect objects. For higher efficiency and precision, a more advanced algorithm called Faster-RCNN makes a significant change by integrating the aforementioned separate (i.e, region proposal, feature extraction and classification) steps into one single end-to-end process with a whole neural network [6]. Faster-RCNN thus has then been applied to different object detection application scenarios. Though more and more novel algorithms of better precision performances have also been proposed afterwards, most of them are updated versions of Faster-RCNN of certain modifications, e.g., Region-based Fully Convolutional Networks (RFCN) [7].

Relevant works that use deep learning-based object detection algorithms for hard-hat wearing detection have been reported. In [8], we first use adaptive background subtraction to detect and obtain moving objects then uses two separate deep neural networks to identify motorcyclist and hard-hat wearing respectively. In [9] a Faster-RCNN hard-hat wearing detector is implemented to identify if a construction worker is wearing hard-hat.

As discussed, a dominator object detection algorithms is the utilized deep neural network which determines the detection precision, speed, memory consumption and performances from more perspectives. In a substation, video sources, fixed cameras and robotics, connect to platforms of different hardware specifications of different preferences, e.g., low requirements of computation sources, high precision, or high speed. Therefore, in this paper, different networks’ availability for hard-hat wearing detection is evaluated. Besides of the chosen neural network, deep learning based object detection algorithms performances’ are also affected by various utilized mechanisms. In order to highlight the neural networks’ impacts, this paper takes Faster-RCNN as the object detection algorithm rather than the other more recently proposed novel algorithms that are basically modified versions of Faster-RCNN.
This paper is structured as follows. Section two formulates the problem. Section three shows the main methodologies. Section four shows the experimental results. This paper is concluded in section five.

2. Problem Formulation
The goal is to identify whether a person shown in an image is wearing a hard-hat or not and to further determine where exactly this person (or the hardhat) is locating. As seen in figure 1, an image of substation shows one of four situations, namely no person appeared, person with hard-hat worn, person wearing no hat and person wearing other hat.

![Sample images of different situations. A) no person; B) person wearing hard-hat; C) person wearing nothing; D) person wearing other hat than hard-hat.](image)

**Figure 1.** Sample images of different situations. A) no person; B) person wearing hard-hat; C) person wearing nothing; D) person wearing other hat than hard-hat.

Logically, given an image, the process first determines whether the scene contains a person then determines whether this person is wearing a hard-hat, other type of hat or no hat, and this is a typical object detection task and available to be solved with deep learning. The general step-by-step framework of object detection could be described by figure 2. Given a set of images, one needs to first use specific labeler tools to manually enclose the pixel regions of the target object in each image and to assign this region a proper label. Both the original images and the region information are then used for model training. Provided a new image, the well trained model can identify the object of interests.
For the hard-hat case, the simplest starting operation is to prepare a set of images containing persons without wearing hard-hats with object location information provided. By the end, a model that can recognize persons without wearing hard-hats may be yield. However, this model would highly likely misrecognize a person wearing hard-hat as a target, because these two categories of objects share a great number of common visual features. Therefore, it is highly recommended to also provide negative image samples, i.e., images of persons wearing hard-hats, for model training, so training algorithms could find distinguishable features between these two categories of objects. Similarly, negative samples regarding persons wearing other hats are also required so the final model could distinguish between hard-hats and other hats. Therefore, the final training set should contain images of ‘persons wearing hard-hats’ with positive class labels and images of ‘persons wearing nothing’ and ‘persons wearing other hats’ with negative class labels. Furthermore, it is also found that by additionally annotate hard-hats and taken them as the third class label can significantly improve the detection precision.

3. Methodologies
This section first introduces Faster-RCNN as a whole and then describes a set of well-known deep neural networks which could be taken as fundamentals of Faster-RCNN.

3.1. Faster-RCNN
A Faster-RCNN framework could be divided into four parts which act sequentially to detect target objects out from images. They are a backbone deep neural network, Region Proposal Network (RPN), a Region of Interest (ROI) pooling layer, and a classification layer. At the first stage, an image could be converted into a set of feature maps by the backbone deep neural network, as a result of which the former unstructured image data could be transferred into a vector of structured data which is more proper for computing purposes. The responsibility of a RPN is to first propose several candidate regions out from a given image then computes the feature vector for each of these regions on the basis of the previously derived feature maps. Another responsibility of RPN is to convert all region proposals’ feature maps that are in different sizes into formalized sized images. Finally, Faster R-CNN uses the classification layer to predict a region’s label and location. See [8] for more about Faster R-CNN.
3.2. Alexnet
As seen, Alexnet has eight layers in total with five convolutional and three fully connected layers, in addition to which Alexnet has utilized certain modifications for higher capability than the most original convolutional neural networks like LeNet. Alexnet is the first that proposes to use Rectified Linear Unit (RLU) to replace Sigmoid as the activation function. RLU is expressed as $f(x) = \max(0, x)$, and the sigmoid function is expressed as $\sigma(x) = 1/(1 + e^{-x})$, where $x$ is actually the summation of weighted inputs. The use of RLU simplifies the derivation process and significantly improves the speed to converge. Alexnet also proposes to use Local Response Normalization (LRN) after the activation and pooling functions in order to activate those neurons with greater responses. Furthermore, in order to avoid over-fitting, Alexnet takes use of both data augmentation and Dropout.

3.3. VGG
The relation between the depth and capability of neural network is explored, via which the author proposes a deeper and more precise network framework called VGG. A VGG network has the same general structure as Alexnet that it also has five convolutional layers and three fully connected ones. However, a convolutional layer in VGG actually contains two to three mini convolutional layers. Compared with alexnet, the convolution filter in VGG is much smaller which is only 3*3. Actually, the connection of two 3*3 filters could perform the same as one single 5*5 filters, and the connection of three could be the same as a 7*7 filter. Therefore, VGG uses less parameters to obtain a same level of performance. Similarly, VGG also uses smaller max-pooling layers which on one hand reduces the total number of parameters and on the other hand can capture more detailed features.

3.4. Squeezenet
Squeezenet has the similar network structure as Alexnet and VGG of both convolutional layers and fully connected layers, while its convolutional layers are more special that is called fire module. A fire module has two kinds of layers, namely a squeeze layer and an expand layer, where the first is a layer of 1*1 filters and the second is a layer of both 1*1 and 3*3 filters. The great use of 1*1 filters significantly reduced the parameter amount of 3*3 filters by nine times. To further limit the parameter amount into a proper level, the author also recommend to limit the number of filters between squeeze and expand layers such that the number of filters in the former should be no more than that of the latter. The ultimate motivation of Squeezenet is to keep the level of the precision of the network architecture and at the same time compress the size of the network as much as possible. As a result, more efficient distributed computing are enabled, and the model could be embedded into and used in the client end hardware more efficiently.

3.5. ResNet
The success of VGG implies that deeper networks can more precisely approximate the complex relation between inputs and outputs. However, as more evaluation works are carried out, deeper neural networks are usually found to be no longer effective due to the truth of gradient vanishing. In order to both utilize the better approximation performance of deep neural networks and at the same time solve the gradient vanishing issue, Residual Neural Network (RexNet) is proposed. ResNet uses the highway network module to modify the output. Assume that X is the input to this module, F(X) represents the first stage mapping which corresponds the two layers, and the output equals $H(X) = F(X) + X$, so the original $H(X) = F(X)$ is replaced. This operation greatly enlarge the backpropogation derivation effect so solves the gradient vanishing issue. The ResNet network thus construct a deep network by combing multiple such modules that can eventually reach 152 convolutional layers, i.e., ResNet-152.

3.6. Inception
The very first version of inception network uses inception modules that are series of 1*1, 3*3 and 5*5 small convolutional filters to replace bigger filters. The use of different filters is to capture features of
different sizes. By using such inception modules, the total number of parameters in the deep neural network are reduced significantly. The second version of inception network on one hand adds a batch normalization layer for the over-fitting issue and on the other hand modifies the 5*5 filters of the aforementioned modules into stacked 3*3 filters. Inception V3 further replaced all n*n filters in the medium layers into stacked 1*n and n*1 filters. Inception V4 further adapt the aforementioned highway network module of resnet to improve the computing efficiency, while we evaluates inception v3 rather than v4 to better distinguish inception network from resnet.

In conclusion, since the success of Alexnet, more advanced deep neural networks are proposed from different point of views, i.e., network compressing, solving gradient vanishing issue and

4. Experiments
This section shows the different performances of hard-hat wearing detection in substation when using different deep neural networks as the backbone network of the object detection algorithm faster-rcnn. As discussed, five different neural networks are concered here, namely Alexnet, VGG, Squeezenet, Resnet101 and InceptionV3, where Alexnet is taken as the basis for comparision and the other four networks are chosen for their different features. Because the ultimate goal is to provide guidelines for choosing the most proper neural network for the hard-hat detection scenario, this paper concerns about different networks’ storage requirements, computation efficiency and precision.

The total number of images used are 1707, where images containing ‘people wearing hard-hat’ equal 747, and ‘people not wearing hard-hat’ equal 977, i.e., some images contain both labels. All these images are divided into three potions with the ratio of 8:1:1 for training, validation and testing respectively.

In order to obtain more convincible results, all networks are tested with the same and simplest possible parameter setting. For each of them, their performances are tested repeatedly with learning rates of 0.0001, 0.0002, 0.0003, 0.0004 and 0.0005 respectively. The learning rate are set to constant variable during the training period. For precision evaluation, Average Precision (AP) is taken as the criteria. The following shows the time consumption when using either GPU or CPU for analyzing one single image, in addition to which the storage consumption of each network relative Alexnet are also shown.

The following table shows the computation time for each selected network and learning rate setting, in addition to which the storage consumption of each network relative to Alexnet is also given. As seen in table 1, Squeezenet significantly reduces the storage consumption to only 2.3% of Alexnet. The network of the biggest size is VGG19 which is as big as 247.1% of Alexnet. Each element in the table shows the time consumption of image analysis by one neural network with either CPU or GPU. As seen, for each network using CPU rather than GPU, the computation period could increases by 5-20 times. Among all of these networks, AlexNet spends the shortest time. Though Squeezenet has the smallest size, the computation time of it is even longer than Alexnet. InceptionV3 spends the longest time that the average image analysis time with GPU and CPU are 1.37 and 26.41 seconds respectively.

| Network (storage) | CPU/GPU | lr=0.0001 | lr=0.0002 | lr=0.0003 | lr=0.0004 | lr=0.0005 | Avg  |
|-------------------|---------|-----------|-----------|-----------|-----------|-----------|------|
| AlexNet (100%)    | CPU     | 1.30      | 0.95      | 0.95      | 1.12      | 1.02      | 1.07 |
|                   | GPU     | 0.23      | 0.21      | 0.21      | 0.23      | 0.25      | 0.23 |
| SqueezeNet (2.3%) | CPU     | 2.26      | 2.40      | 2.32      | 2.05      | 2.15      | 2.24 |
|                   | GPU     | 0.27      | 0.29      | 0.21      | 0.20      | 0.28      | 0.25 |
| VGG19 (247.1%)    | CPU     | 2.72      | 2.70      | 2.65      | 2.63      | 2.56      | 2.65 |
|                   | GPU     | 0.34      | 0.34      | 0.40      | 0.40      | 0.41      | 0.38 |
| Model     | CPU   | GPU   |
|-----------|-------|-------|
| ResNet101 | 14.98 | 0.86  |
|           | 14.84 | 0.83  |
|           | 14.37 | 0.82  |
|           | 14.61 | 0.82  |
|           | 15.30 | 0.96  |
|           | 14.82 | 0.86  |
| InceptionV3 | 27.87 | 1.48  |
|           | 28.54 | 1.30  |
|           | 25.69 | 1.46  |
|           | 25.08 | 1.33  |
|           | 24.87 | 1.31  |
|           | 26.41 | 1.37  |

Figure 3 shows the average precision of all five neural networks on different learning rate settings and labels. As seen, these five neural networks’ precision performances are significantly different from each other. Besides, a same network may behave different on different labels. In order to better distinguish these five networks’ precision performances, the overall average precision on all three labels are shown in the figure 3. ResNet101 performs significantly better than the others. Squeezenet performs similar with Alexnet, but with much smaller size and similar computation pressure. Because this application concerns most about the label of ‘people wearing no hard-hat’, so results shown in the third graph may make more senses. As seen, the results are similar as the overall result that Resnet101 still obtains the best score.

5. Conclusions
This paper investigates the different availabilities when applying different deep neural networks as the backbone of Faster-RCNN for hard-hat wearing detection in the substation scenario. Various aspects regarding these networks’ performances are evaluated where most of these are real-world implementation concerned criteria, including storage consumption, computation efficiency and average precision. For example, ResNet101 can obtain the highest detection precision but usually leads to relatively high computation and storage pressure. Though Squeezenet performs better on the storage side, its detection precision is only similar with Alexnet. This study provides a guideline to the selection of the most appropriate deep neural network architectures for the specific monitoring scenario.

Acknowledgments
The work was supported by State Grid Corporation of China Headquarters Project (5455HJ180015) which is 'Research regarding the Substation Robotic Intelligent Inspection Technique based on Actively Cooperated Intelligent Sensing Devices'.
References

[1] Mneymneh Bahaa Eddine, et al. 2017 Automated hardhat detection for construction safety applications Procedia Engineering 196: 895-902.

[2] Li K, Zhao X, Bian J, Tan M 2018 Automatic safety helmet wearing detection Proceeding of 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control and Intelligent Systems (CYBER) 617–622.

[3] Zhang H, Yan X, Li H, Jin R and Fu H 2019 Real-time alarning, monitoring, and locating for non-hard-hat use in construction J. Constr. Eng. Manag. 145 4019006, https://doi.org/10.1061/(asce)co.1943-7862.0001629.

[4] Girshick R, Donahue J, Darrell T and Malik J 2013 Rich feature hierarchies for accurate object detection and semantic segmentation Proceeding of CVPR IEEE.

[5] Girshick R 2015 Fast R-CNN Computer Science.

[6] Ren S, He K, Girshick R, et al. 2017 Faster R-CNN: towards real-time object detection with region proposal networks IEEE Transactions on Pattern Analysis & Machine Intelligence 39(6):1137-1149.

[7] Dai J, Li Y, He K, et al. 2016 R-FCN: Object Detection via Region-based Fully Convolutional Networks Curran Associates Inc.

[8] Zhu Z, Park M, Elsafty N 2015 Automated monitoring of hardhats wearing for onsite safety enhancement, Proceeding of the 5th International Construction Specialty Conference of the Canadian Society for Civil Engineering: 1–9.

[9] Mneymneh B E, Abbas M, Khoury H 2018 Vision-based framework for intelligent monitoring of hardhat wearing on construction sites J. Comput. Civ. Eng. 33: 04018066.