FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 33, No 1, March 2020, pp. 133 - 153
https://doi.org/10.2298/FUEE2001133A

WIND TURBINE TOWER DETECTION USING FEATURE DESCRIPTORS AND DEEP LEARNING

Fereshteh Abedini[1], Mahdi Bahaghighat[2], Misak S'hoyan[3]

[1]Department of Electrical Engineering Amirkabir University of Technology Tehran, Iran
[2]Department of Electrical Engineering Raja University Qazvin, Iran
[3]Department of Information Security National Polytechnic University of Armenia
Yerevan, Armenia

Abstract. *Wind Turbine Towers (WTTs) are the main structures of wind farms. They are costly devices that must be thoroughly inspected according to maintenance plans. Today, existence of machine vision techniques along with unmanned aerial vehicles (UAVs) enable fast, easy, and intelligent visual inspection of the structures. Our work is aimed towards developing a vision-based system to perform Nondestructive tests (NDTs) for wind turbines using UAVs. In order to navigate the flying machine toward the wind turbine tower and reliably land on it, the exact position of the wind turbine and its tower must be detected. We employ several strong computer vision approaches such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Features from Accelerated Segment Test (FAST), Brute-Force, Fast Library for Approximate Nearest Neighbors (FLANN) to detect the WTT. Then, in order to increase the reliability of the system, we apply the ResNet, MobileNet, ShuffleNet, EffNet, and SqueezeNet pre-trained classifiers in order to verify whether a detected object is indeed a turbine tower or not. This intelligent monitoring system has auto navigation ability and can be used for future goals including intelligent fault diagnosis and maintenance purposes. The simulation results show the accuracy of the proposed model are 89.4% in WTT detection and 97.74% in verification (classification) problems.*

Key words: *Machine Vision, Object Detection, Vision Inspection, Wind Tur-bine, Deep Learning.*

Received August 27, 2019; received in revised form November 26, 2019
Corresponding author: Mahdi Bahaghighat
Department of Electrical Engineering Raja University Qazvin, Iran
(E-mail: m.bahaghighat@aut.ac.ir)

1 Introduction

Providing reliable and affordable electricity in order to face the increasing demand of energy in the near future is a worldwide concern. In this regard, developing renewable and clean energy sources such as wind turbine (WT) farms in smart grid (SG) infrastructures can play a crucial role in increasing the capacity of electricity production in many countries across the world. SG deploys widely information and communication technologies (ICT) [1] subsystems. There is almost unlimited number of possible applications of ICT subsystems within the smart grid. SGs with these infrastructures make it more possible to develop reliable systems through Artificial Intelligence (AI) [2–4].

On the other side, guaranteeing the reliability of wind turbines is of great importance. In case of failure and faulty operation, the grid will face interruptions in its service. Challenges and costly breakdowns such as mechanical deformations, surface defects, overheated components in rotor blades, nacelles, slip rings, yaw drives, bearings, gearbox, generators, and transformers are the ones which should be monitored to detect faults intelligently in a wind turbine farm [5–8].

Besides, WTs are costly devices that should have advanced maintenance systems [5]. In order to increase the lifetime of the WTs and reduce the maintenance cost, it is essential to improve the monitoring and maintenance approaches and reach solutions to avoid failure during in-service operation [9, 10]. Vision inspection paves the way toward generating reliable, efficient, and economical electrical energy in wind turbine farms. Image processing and machine learning methods have been widely employed to assist in monitoring and fault diagnostic solutions in energy systems [10–15]. The authors in [11] proposed a learning-based approach to inspect power line infrastructures. In [12], authors suggested a smart framework for system reliability, using machine learning algorithms, to predict failures for preventive maintenance of system components. The authors in [10], benefited from the image data and suggested a model to estimate the rotational velocity of the turbine blade. Estimating the velocity of blades helps to predict the amount of generated power by the WTs in the smart grid. This will ensure the grid to be a more reliable system.

In [13], the authors employed machine learning algorithms to manage the energy of loads and sources in smart grids. The problems of malicious activity prediction and intrusion detection have been analyzed using machine learning techniques in smart grid communication systems in [14, 15]. The authors in [14] detected malicious events and improved system reliability. In [15], a novel method was proposed to reliably warn and anticipate abnormalities and failures in distribution and communication systems.

Deep learning techniques in computer vision applications such as autonomous inspection and monitoring have had a tremendous impact in recent years [16]. Using convolutional neural networks (CNNs) have led computer vision to more advanced approaches. The main feature of a CNN is its deep architecture [17]. One of the common and effective approaches in deep learning is to use a pre-trained network. Several classification problems have been solved using pre-trained networks [17]. For instance, the authors in [16] used deep CNN architecture in the fault classification of power line insulators.

Precise monitoring and forecasting of emerging faults and failures of WTs are critical tasks and can be complex challenges. If the system problems are detected and notified accurately, they can be fixed as soon as possible to increase the reliability of the system. Intelligent vision inspection techniques can be employed to make these predictions and controls to be done automatically and reliably. Therefore, in this work, we propose an intelligent approach, which deploys computer vision techniques to detect wind turbine towers (WTTs). Detection of WTTs can ease the challenges in automatic fault detection and diagnostics process in wind farms through unmanned aerial vehicles (UAV) [18]. In [19], they used signal processing approaches and employed a combination of line and feature detection to locate the wind turbine towers. They started the wind turbine detection stage with Hough transform to detect lines but there are many objects in the background with line shapes, such as horizon, shadow, mountains, and power lines, which are not desired to be located. Detecting the lines and then removing the false detections can cause computational cost and decrease the overall accuracy.

In this research, we developed a new vision-based model to detect WTTs and verify it. The proposed verification step which is implemented using a deep learning classification method, is an extra phase to improve the reliability of the navigation system. This classifier decides between OK and NG (Not Good) detection results. Here, OK means a WTT is detected correctly and the UAV should update its navigation information while NG means a false detection has occurred and the UAV should keep its previous knowledge. In our future works, the UAV will be embedded with thermal vision cameras for advanced visual Nondestructive Tests (NDT).

The remainder of this paper is organized as follows. Section II presents the methodology of the proposed model. The experiments and results are reported in Section III. Finally, in Section IV, we conclude the paper.

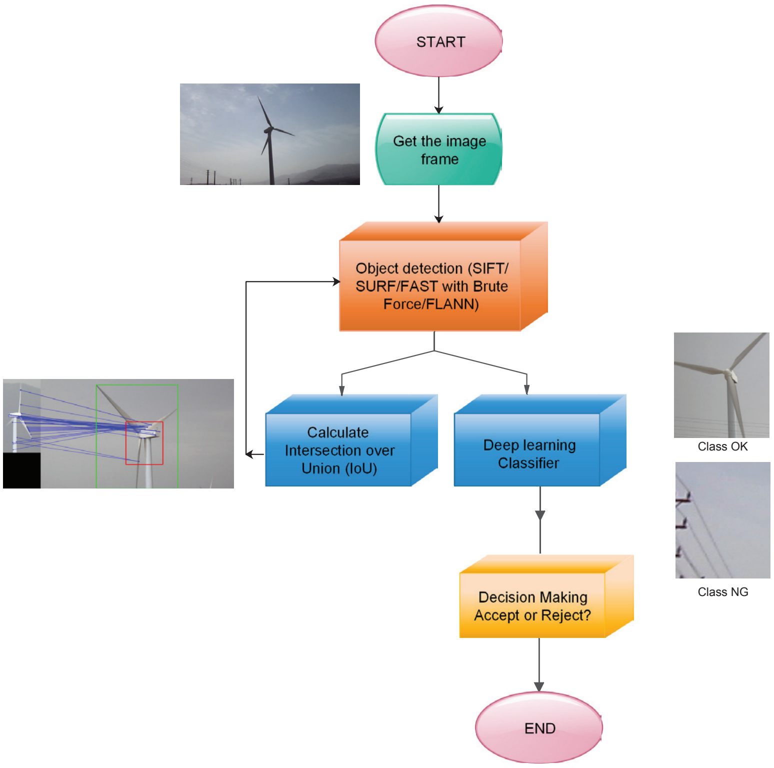

START
Get the image frame
Object detection (SIFT/ SURF/FAST with Brute Force/FLANN)
Calculate Intersection over Union (IoU)
Deep learning Classifier
Class OK
Class NG
Decision Making Accept or Reject?
END

Fig. 1: Flowchart of the proposed model.

2 Methodology

In this work, we target developing an appropriate infrastructure to perform vision-based Nondestructive tests (NDT) for wind turbines using UAVs for future works. In order to have a precise navigation and guide the flying machine toward the wind turbine tower and reliably land on it, the position of the wind turbine and its tower must be estimated. To tackle this issue, we have proposed our model based on the flowchart of Fig. 1. There are two main steps in our model, the first one probing a WTT location in an input image and the second one checking whether it is a real WTT or not using a deep learning classifier. This classifier decides between OK and NG detection results. The UAV updates the position information when the classifier output is OK.

2.1 Dataset

In our research, DB1 includes about 1500 images which have been captured by us in different angles, distances, and backgrounds in a real wind turbine farm. In this wind turbine farm more than 300 WTTs exist and in average 5 images/WTT are available in DB1. This dataset is used for object detection problem. In Fig. 2, some selected samples from DB1 are depicted. Beside this dataset, about 2000 images consisting of two different classes, *Not Good (NG)* as Class 1 and *OK* as Class 2 with equal distribution (1000 images for OK class and 1000 samples for NG), have been collected as DB2 to evaluate the performance of the proposed algorithm in the verification stage (classification problem). Fig. 3 illustrates several examples of two mentioned classes in DB2.

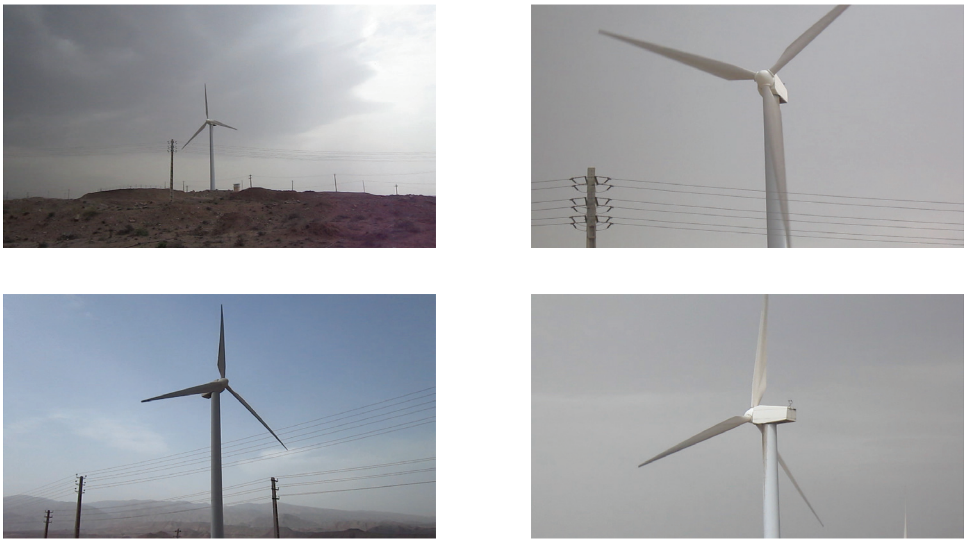


Fig. 2: Some selected image samples in our dataset (DB1)

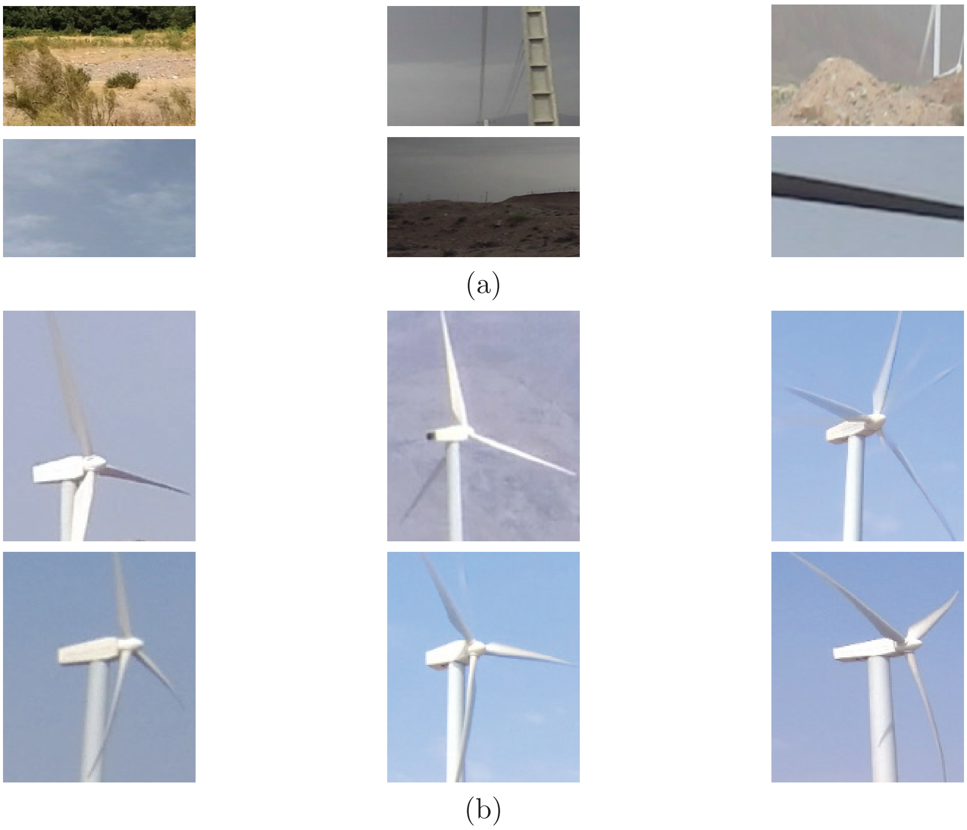


Fig. 3: Several samples from our database (DB2) for verification problem including two classes: NG (Not Good) and OK, (a) NG (b) OK.

2.2 The WTT object detection

Object detection is a fundamental field of study in computer vision and image processing applications [20–27]. Recently, various algorithms have been suggested for object detection purposes [28–31]. These algorithms extract local interest features (key points) and describe them to identify the objects [28–31]. In [28], a well-known algorithm, Scale-Invariant Feature Transform (SIFT) was presented as a scheme for extracting highly distinctive invariant features, which can be used to match different views of objects. The advantage of SIFT is its invariance to scaling, rotation, and translation. The SIFT key point detectors and descriptors have reported to be remarkably effective in different applications [28].

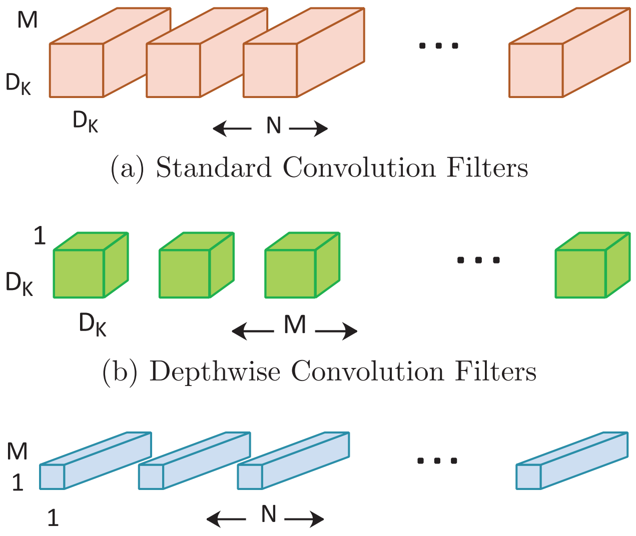
SIFT is computationally expensive, especially for real-time systems. This has led into thorough research toward an alternative algorithm with lower computational cost such as Speeded Up Robust Features (SURF) and Features from Accelerated Segment Test (FAST) [28–31]. In [29], the authors implemented a new detector and descriptor called SURF which is invariant to scaling and rotation. It is competitive and often superior in terms of repeatability, distinctiveness, and robustness to SIFT, and can be calculated and compared much faster.

SIFT and SURF are both based on detectors and descriptors. Once key points are extracted, a template-matching algorithm must be applied to describe the features. Here, we adopt Brute-Force [28] and FLANN (Fast Library for Approximate Nearest Neighbors) based matcher [32] to test the similarity between the descriptors for the training and test images. The Brute-Force matcher is simple; it takes the descriptor in the first set and compares it to all other descriptors in the second set using a distance calculation. Then the closest ones are returned as the best matches. FLANN contains a collection of optimized algorithms for fast nearest neighbor searches in large datasets and for features which are high dimensional.

2.3 The verification step based on deep learning classifiers

We used SIFT, SURF, FAST algorithms to extract features and detect WTs. Since the object detection stage plays a crucial role in reliability of navigating toward the correct target, we must verify our detection results. If the towers are not detected reliably, UAVs and thermal cameras may hit the blades or land on wrong objects. As a result, the UAVs and cameras may be damaged or the inspection and estimation data may be erroneous.

There are a lot of classical approaches for classification problems such as Random Forest (RF), AdaBoost, k-Nearest Neighbor (kNN), and Support Vector Machine (SVM) [33–38] but to verify the object detection output accuratelly, we propose use of a pre-trained CNN as the classifier. The authors in [17] presented MobileNet as a class of more efficient models for mobile and embedded vision applications. As indicated in Fig. 4, MobileNets are based on an architecture that uses depth wise separable convolutions to build lightweight deep neural networks [17, 39–41]. In [17], the authors introduced two simple global hyper-parameters that effectively compensate for latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application according to the limitations of the problem. They have presented many experiments on resource and accuracy trade-offs and have demonstrated better performance compared to other common models on ImageNet classification [17].



(a) Standard Convolution Filters

(b) Depthwise Convolution Filters

(c) 1×1 Convolution Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Fig. 4: The standard convolution filters in (a) are replaced by two layers: depth-wise convolution in (b) and point-wise convolution in (c) to build depth-wise separable filters [17].

In this work, we applied MobileNet [17], ShuffleNet (an extremely efficient convolutional neural network for mobile devices) [42], EffNet (an efficient structure for convolutional neural network) [43], SqueezeNet [44], and ResNet [45, 46] pre-trained classifiers with two added fully connected layers *Dense 1* and *Dense 2*. Dense 2 is the output layer and fixed for our binary classifier while Dense 1's parameters are targeted as optimization parameters.

ShuffleNet is a practical CNN architecture with high computational efficiency. It provides more feature map channels to encode more information. This is an important point for the performance of very small networks. ShuffleNet is well designed and developed for embedded devices such as mobile phones with very low computing power [42]. SqueezeNet tries to reduce time cost and parameters noticeably while holding on the accuracy [44]. Residual Neural Network so-called ResNet utilizes the bottleneck architecture efficiently to obtain impressive performance [45, 46]. In this model an innovative structure with skip connections and features heavy batch normalization was introduced. Such skip connections are also known as gated units or gated recurrent units and have a strong similarity to recent successful elements applied in RNNs. ResNet has proven to be powerful in a lot of applications but one major disadvantage is that the deeper networks usually need several weeks for training, making it impractical in real-world applications. In addition it has large size for most embedded devices. In comparison to ResNet, ShuffleNet has the lower complexity with the same settings. MobileNet and ShuffleNet are favourite models for embedded and mobile systems but EffNet is the optimized model that can be replaced with them in the same applications. EffNet deploys spatial separable convolution, which is simply a depthwise convolution splitted along the x and y axis with a separable pooling between them. It has been shown that it has the same capacity even when applied to narrow and shallow architectures [43]. EffNet block is developed to guarantee the safe replacement of the vanilla convolution layers in mobile hardware applications. Therefore, it has two main advantages, first is the quicker inference and second the application of a larger, deeper network becoming possible [43]. A comparison of MobileNet and ShuffleNet with EffNet is shown in Fig. 5. In this figure, *dw* means depthwise convolution, *mp* means max-pooling, *ch* is for the number of output channels and *gc* is for group convolutions [43].

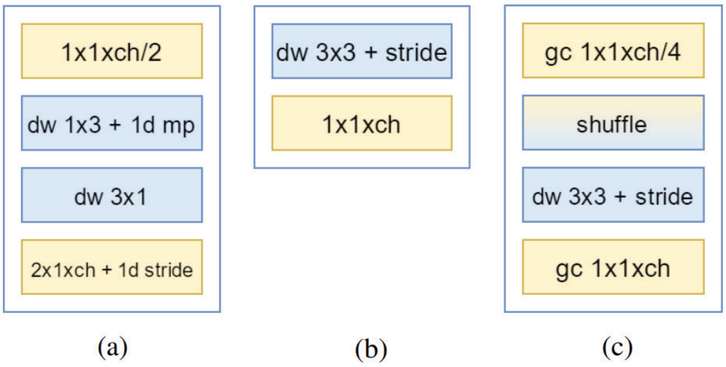


Fig. 5: A comparison of MobileNet and ShuffleNet with EffNet [43]
(a) An EffNet block (b) A MobileNet block (c) A ShuffleNet block

In the next section, we elaborate experimental results in our research.

3 Experiments and Results

In our work, all implementations and simulations have been done using Python programming language and TensorFlow by a Core i7 CPU and

NVIDIA GTX 1050 GPU with 16 GB DDR4 RAM memory. As mentioned before in Section 2.1, We carried out our experiments and simulations using images which were captured by us in a real wind farm.

3.1 Using SIFT, SURF, and FAST for WTT object detection

In order to detect the wind turbine towers, firstly, the features and descriptors are extracted using SIFT, SURF, and FAST schemes and then, by applying Brute-Force and FLANN template matching algorithms, a bounding box is predicted for the wind towers. To evaluate the performance of our proposed object detection method, we used the Intersection over Union (IoU) [47]. IoU can be calculated having the ground-truth (GT) bounding box and predicted bounding box of the model. Fig. 6 illustrates different examples of the simulation results in our dataset. The features key points are drawn in blue, predicted bounding box is in red and the ground-truth bounding box is in green.

The goal was to compute the intersection over union of detected bounding box and ground-truth box based on Equation (1).

$$IoU = \frac{Area \quad of \quad Overlap}{Area \quad of \quad Union} \tag{1}$$

We considered the IoU as a scoring factor and will decide on the performance of the suggested model based on Equation (2).

$$Prediction = \begin{cases} 0 & \text{IoU} \leq \Lambda \\ 1 & \text{IoU} \geq \Lambda \end{cases} \tag{2}$$

If $Prediction = 0$, it actually means a detected object is not acceptable. Fig. 6.(a) demonstrates a poor detection with $IoU = 0.0198$, while Fig. 6.(b) is an example of a less good prediction with $IoU = 0.3129$, Fig. 6.(c) is a good detection with $IoU = 0.5734$, and Fig. 6.(d) demonstrates an excellent bounding box, $IoU = 0.9708$.

The detection rate of the WTT, dr_{WTT}, can be calculated as introduced in Equation (3). In this equation, N is the total number of images in the experiment.

$$dr_{WTT} = \frac{\sum_{i=1}^{N} Prediction}{N} \times 100 \tag{3}$$

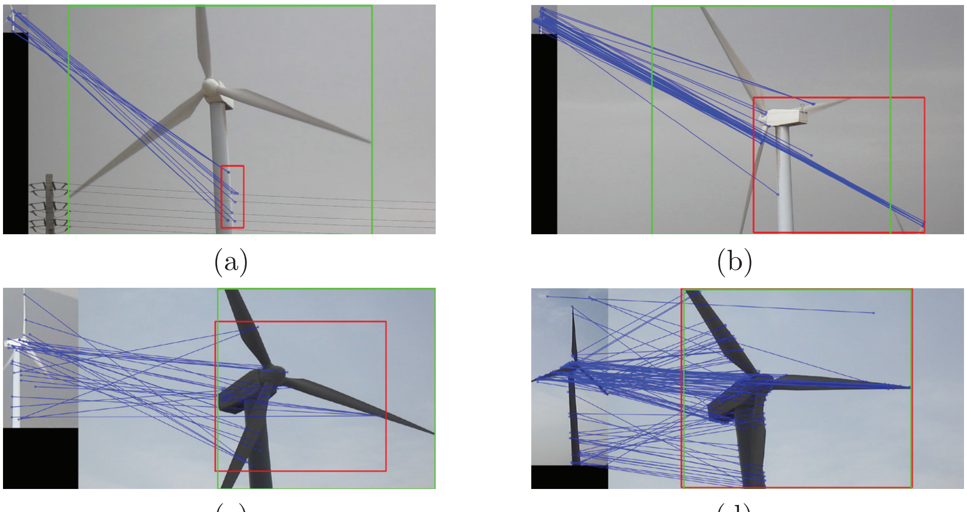


Fig. 6: Detecting wind tower in different images. (a): IoU=0.0198, poor detection using FAST, SIFT, and FLANN. (b): IoU=0.3129, less good detection using FAST, SURF, and FLANN. (c): IoU=0.5334, good detection using SURF and Brute-Force. (d): IoU=0.9708, excellent detection using FAST, SIFT, and Brute-Force.

To evaluate the performance of the proposed method, we measured the accuracy using Equation (4) and the following metrics: TP (True Positive) indicates the number of correctly classified samples in the OK class, FN (False Negative) indicates the number of samples for which the classification is NG class, but misclassified, TN (True Negative) indicates the number of samples that properly classified as not belonging to the OK class, and FP (False Positive) indicates the number of samples belongs to the class OK but misclassified [48]

$$Accurracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

The results of applying SIFT, SURF, and FAST feature extractors with Brute-Force and FLANN template matching algorithms are summarized in Table 1. According to this table, FAST outperforms other feature extractors in term of accuracy. It extracts more features than SIFT and SURF, that makes FAST more powerful in the detection of wind towers. We also observe that SURF is more accurate in extracting and describing features in comparison to SIFT. As it is mentioned in Table 1, FLANN template matching works slightly better than Brute-Force.

Table 1: Result of the WTT object detection

Method			Accuracy(%)		
Feature extractor & descriptor	Matcher	Runtime (s)	IoU=0.5	0.3	0.25
SIFT	Brute-Force	0.3955	45.4	74.8	79.8
SURF	Brute-Force	0.3021	46.6	76.9	82.0
FAST & SIFT	Brute-Force	0.0728	51.1	81.8	84.1
FAST & SURF	Brute-Force	0.0623	52.7	83.6	87.3
SIFT	FLANN	0.3666	47.0	78.5	83.2
SURF	FLANN	0.2309	49.5	82.7	84.9
FAST & SIFT	FLANN	0.0821	52.6	82.8	87.3
FAST & SURF	FLANN	0.0542	54.8	83.8	89.4

3.2 Applying deep learning classifiers to verify object detection results

To verify the object detection performance, firstly we used the pre-trained MobileNet classifier with MobileNets Body Architecture [17] that is presented in Table 2. In addition, we deployed the following parameter setting:

The optimizer used in our work is named Adam [49]. We set learning rate to be 3×10^{-7} for the optimizer. At two last layers, we used Fully Connected (FC) networks. In these layers, each neuron reads the neurons output in the previous layer and processes the information it needs, and produces the outputs for the next layer [50–52]. The general formula is the following, where b is the BIAS; weights of connections are w_i, f is a nonlinear activation function.

$$f(W^t x) = f(\sum_{i=1}^{3} W_i x_i + b) \tag{5}$$

The most common activation functions are Sigmoid function, hyperbolic tangent function (Tanh), and rectified linear function (ReLU). Their formulas are as follows ([50–52]):

$$f(W^t x) = Sigmoid(W^t x) = \frac{1}{1 + exp(-W^t x)} \tag{6}$$

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

$$f(W^t x) = Relu(W^t x) = max(0, W^t x) \tag{8}$$

The Sigmoid function receives a value range between 0 and 1, and a real-valued number as the firing rate of a neuron: 0 for not firing or 1 for firing. The hyperbolic tangent functions as a zero-centered output range and uses $[-1, 1]$ instead of $[0, 1]$. For Relu function, if the input is less than 0, its activation will be thresholded at zero. The Softmax function also can be used as the output neuron function and is a logistic function. The function definition is as follows ([50–52]):

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1^{e^{x_k}}}^{K} X_i} \qquad for \quad j = 1, ..., K \tag{9}$$

In our work, Softmax is used for the final classification at the final layer of the NN.

In the verification problem, the detected objects should be classified into two categories: OK and NG. As a result, the number of neurons in FC output layer (Dense 2) would be equal to two. The classification results are presented in Table 3. Based on the information given in this table, we could achieve the accuracy of 96.01% in classifying the objects in the 5^{th} experiment ($E5$) as the best result of MobileNet with runtime around 0.034857 (s). Then we conducted our new experiments ($E7$ to $E12$) using ResNet50 rather than MobileNet. According to $E11$, it could result in the higher accuracy about 98.92% in 0.061942(s). Although ResNet50 achieved the better result in term of the average accuracy, the runtime was doubled. Afterwards, $E13$ to $E18$ were implemented for EffNet, ShuffleNet, and SqueezeNet. The obtained results can be evident that EffNet with 1000 iterations and 100 neurons before the output layer ($E14$) can lead to 97.74% average accuracy in 0.039351(s) as the best scenario. The validation accuracy and the validation loss for $E14$ are depicted in Fig. 7. In Fig. 7.(a), the network has been trained for 1000 epochs and we have obtained a validation accuracy of 97.74% and as it is apparent in Fig. 7.(b) the validation loss follows the training loss which is very low.

Table 2: MobileNets Body Architecture [17].

Type/Stride	Filter Shape	Input Size
Conv/s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw/s1	$3 \times 3 \times 32$dw	$112 \times 112 \times 32$
Conv/s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw/s2	$3 \times 3 \times 64$dw	$112 \times 112 \times 64$
Conv/s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw/s1	$3 \times 3 \times 128$dw	$56 \times 56 \times 128$
Conv/s1	$1 \times 1 \times 128 \times 128$	$56 \times \times 56 \times 128$
Conv dw/s2	$3 \times 3 \times 128$dw	$56 \times 56 \times 128$
Conv/s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw/s1	$3 \times 3 \times 256$dw	$28 \times 28 \times 256$
Conv/s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw/s2	$3 \times 3 \times 256$dw	$28 \times 28 \times 256$
Conv/s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw/s1 Conv/s1	$3 \times 3 \times 512$dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw/s2	$3 \times 3 \times 512$dw	$14 \times 14 \times 512$
Conv/s1	$1 \times 1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw/s2	$3 \times 3 \times 1024$dw	$7 \times 7 \times 1024$
Conv/s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool/s1	Pool 7×7	$7 \times 7 \times 1024$
FC/s1	1024×1000	$1 \times 1 \times 1024$
Softmax/s1	Classifier	$1 \times 1 \times 1000$

3.3 Decision making and updating the positions

Remembering the Fig. 1, the last stage of our model is the decision making. In order to update the location of the WTT, Equations (10) and (11) are used for X-axis and Y-axis, respectively.

$$X_i^* = \begin{cases} \mathrm{X}_{i-1}, & \mathrm{D}_i = NG \\ \mathrm{X}_i^P, & \mathrm{D}_i = OK \end{cases} \tag{10}$$

$$Y_i^* = \begin{cases} \mathrm{Y}_{i-1}, & \mathrm{D}_i = NG \\ \mathrm{Y}_i^P, & \mathrm{D}_i = OK \end{cases} \tag{11}$$

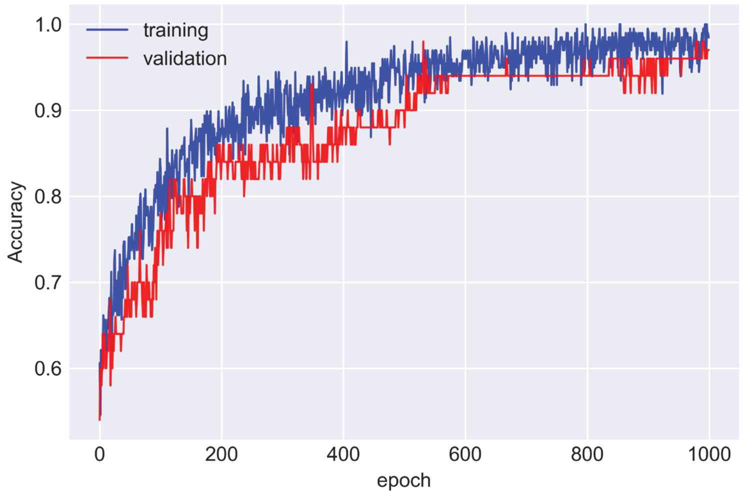
where in these equations, D_i denotes the status of the detected object in current image frame, i. Here, X_{i-1}, X_i^P and X_i^* are the X positions for previous frame $(i-1)$, predicted X position for current frame (i), and the new value of X position for the current frame, respectively (for $i = 1, 2$). The same rule applies for Y_{i-1}, Y_i^P and Y_i^*, in Y positions (for $i = 1, 2$). According to the result of the classifier and these equations, the UAV can decide on updating its information about the location of the WTT.

Table 3: Classification results for the verification problem based on MobileNet, ShuffleNet, EffNet, SqueezeNet, and ResNet

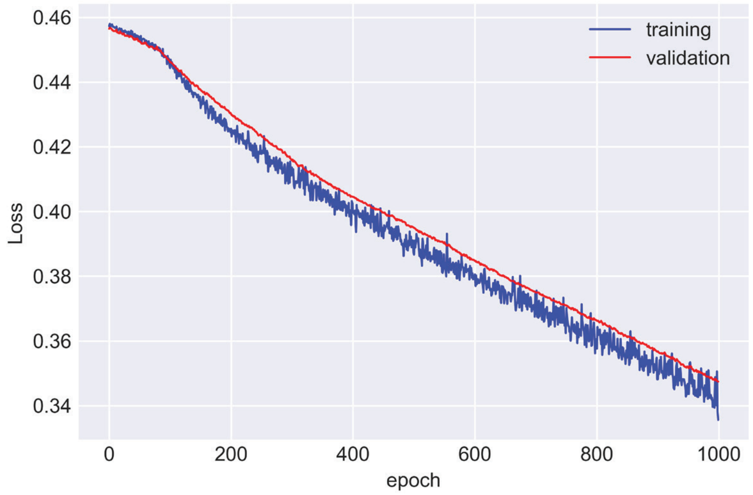
Experiment	Base model	Epochs	Dense 1	Run time (s)	Accuracy (%)
E1	MobileNet	20	40	0.027058	51.11
E2	MobileNet	50	40	0.030192	54.43
E3	MobileNet	100	100	0.032696	76.69
E4	MobileNet	300	100	0.032752	86.22
E5	MobileNet	1000	100	0.034857	96.01
E6	MobileNet	1000	40	0.033198	85.00
E7	ResNet50	20	40	0.055964	56.01
E8	ResNet50	50	40	0.054390	56.13
E9	ResNet50	100	100	0.059588	88.21
E10	ResNet50	300	100	0.060267	95.78
E11	ResNet50	1000	100	0.061942	98.92
E12	ResNet50	1000	40	0.060906	89.46
E13	EffNet	100	100	0.036546	79.12
E14	EffNet	1000	100	0.039351	97.74
E15	ShuffleNet	100	100	0.033497	77.43
E16	ShuffleNet	1000	100	0.035001	95.89
E17	Squeezenet	100	100	0.040951	73.41
E18	Squeezenet	1000	100	0.041828	93.36

4 Conclusion

Wind turbine tower (WTT) as a main component in a farm is a mechanical structure where its components are formed and constructed using carbon



(a)



(b)

Fig. 7: The comparison between the training and validation results of EffNet pre-trained classifier for the verification problem
(a) Accuracy (b) Loss

fiber reinforced plastic (CFRP) [9]. In order to have intelligent and proactive maintenance services for the farm, it is essential to develop monitoring infrastructure based on Vision Inspection (VI) technologies. This can increase the lifetime of the WT farm and reduce the maintenance cost, provided that accurate faults and failures predictions are available via advanced Nondestructive tests approaches such as intelligent VI and thermal visions [6–10].

In this paper, We suggested a scheme to detect the wind turbine towers to facilitate monitoring, controlling, and maintenance tasks in smart grids. We deployed machine learning techniques with vision inspection proposes to navigate a flying machine in a wind turbine farm precisely. We used SIFT, SURF, and FAST as feature extractors and Brute-Force and FLANN as matchers to detect wind turbines. Our simulation results have shown that FAST as the feature extractor with SURF as the descriptor along with FLANN matcher outperforms in object detection task with the 89.4% accuracy. Besides, in order to improve navigation reliability, an additional binary classification step was considered based on ResNet, MobileNet, ShuffleNet, EffNet, and SqueezeNet. Among all mentioned classifiers, ResNet50 obtained the highest average accuracy about 98.92% in $0.061942(s)$. Its runtime was almost two times compared to EffNet with average accuracy around 97.74%. Therefore, from practical points of view such as computational-efficiency and memory restrictions for an embedded device, our well-tuned pre-trained EffNet was considered as the best classifier among all mentioned models in our research.

References

[1] S. A. Motamedi, “Psnr enhancement in image streaming over cognitive radio sensor networks,” *ETRI Journal*, vol. 39, no. 5, pp. 683–694, 2017.

[2] M. A. Dimitrijević, M. Andrejević-Stošović, J. Milojković, and V. Litovski, “Implementation of artificial neural networks based ai concepts to the smart grid,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 27, no. 3, pp. 411–424, 2014.

[3] A. Janjic, S. Savic, G. Janackovic, M. Stankovic, and L. Z. Velimirovic, “Multicriteria assesment of the smart grid efficiency using the fuzzy analitical hyerarchy process,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 29, no. 4, pp. 631–646, 2016.

[4] R. Martać, N. Milivojević, V. Milivojević, V. Ćirović, and D. Barać, “Using internet of things in monitoring and management of dams in serbia,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 29, no. 3, pp. 419–435, 2015.

[5] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. Tameghe, and G. Ekemb, “Wind turbine condition monitoring: State-of-the-art review,

new trends, and future challenges," *Energies*, vol. 7, no. 4, pp. 2595–2630, 2014.

[6] H. Sanati, D. Wood, and Q. Sun, "Condition monitoring of wind turbine blades using active and passive thermography," *Applied Sciences*, vol. 8, no. 10, p. 2004, 2018.

[7] F. P. García Márquez, I. Segovia Ramírez, A. Pliego Marugán, Á. Huerta Herráiz, and M. Papaelias, "A novel walking robot based system for non-destructive testing in wind turbines," 2019.

[8] X.-l. Li, J. Sun, N. Tao, L. Feng, J.-l. Shen, Y. He, C. Zhang, and Y.-j. Zhao, "An effective method to inspect adhesive quality of wind turbine blades using transmission thermography," *Journal of Nondestructive Evaluation*, vol. 37, no. 2, p. 19, 2018.

[9] C.-S. Tsai, C.-T. Hsieh, and S.-J. Huang, "Enhancement of damage-detection of wind turbine blades via cwt-based approaches," *IEEE Transactions on energy conversion*, vol. 21, no. 3, pp. 776–781, 2006.

[10] M. Bahaghighat and S. A. Motamedi, "Vision inspection and monitoring of wind turbine farms in emerging smart grids," *Facta Universitatis, Series: Electronics and Energetics*, vol. 31, no. 2, pp. 287–301, 2018.

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

[12] C. Rudin, D. Waltz, R. N. Anderson, A. Boulanger, A. Salleb-Aouissi, M. Chow, H. Dutta, P. N. Gross, B. Huang, S. Ierome *et al.*, "Machine learning for the new york city power grid," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 2, pp. 328–345, 2011.

[13] R. N. Anderson, A. Boulanger, W. B. Powell, and W. Scott, "Adaptive stochastic control for the smart grid," pp. 1098–1115, 2011.

[14] Z. M. Fadlullah, M. M. Fouda, N. Kato, X. Shen, and Y. Nozaki, "An early warning system against malicious activities for smart grid communications," *IEEE Network*, vol. 25, no. 5, pp. 50–55, 2011.

[15] Y. Zhang, L. Wang, W. Sun, R. C. Green II, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 796–808, 2011.

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

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[18] M. Babaie, M. E. Shiri, and M. Bahaghighat, "A new descriptor for uav images mapping by applying discrete local radon," in *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*. IEEE, 2018, pp. 52–56.

[19] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 998–1007.

[20] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern recognition letters*, vol. 27, no. 1, pp. 49–58, 2006.

[21] I. Laptev, "Improving object detection with boosted histograms," *Image and Vision Computing*, vol. 27, no. 5, pp. 535–544, 2009.

[22] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern analysis and applications*, vol. 8, no. 1-2, pp. 17–31, 2005.

[23] J. C. S. J. Junior, S. R. Musse, and C. R. Jung, "Crowd analysis using computer vision techniques," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 66–77, 2010.

[24] R. Akbari, M. K. Bahaghighat, and J. Mohammadi, "Legendre moments for face identification based on single image per person," in *2010 2nd International Conference on Signal Processing Systems*, vol. 1. IEEE, 2010, pp. V1–248.

[25] M. K. Bahaghighat, R. Akbari *et al.*, "Fingerprint image enhancement using gwt and dmf," in *2010 2nd International Conference on Signal Processing Systems*, vol. 1. IEEE, 2010, pp. V1–253.

[26] N. Karimimehr, A. A. B. Shirazi *et al.*, "Fingerprint image enhancement using gabor wavelet transform," in *2010 18th Iranian Conference on Electrical Engineering*. IEEE, 2010, pp. 316–320.

[27] M. Bahaghighat, M. Mirfattahi, L. Akbari, and M. Babaie, "Designing quality control system based on vision inspection in pharmaceutical product lines," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2018, pp. 1–4.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[30] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[31] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[32] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[33] M. Bahaghighat, L. Akbari, and Q. Xin, "A machine learning-based approach for counting blister cards within drug packages," *IEEE Access*, vol. 7, pp. 83 785–83 796, 2019.

[34] A. Esmaeili Kelishomi, A. Garmabaki, M. Bahaghighat, and J. Dong, "Mobile user indoor-outdoor detection through physical daily activities," *Sensors*, vol. 19, no. 3, p. 511, 2019.

[35] P. Thanh Noi and M. Kappas, "Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery," *Sensors*, vol. 18, no. 1, p. 18, 2018.

[36] H. Al-Shehri, A. Al-Qarni, L. Al-Saati, A. Batoaq, H. Badukhen, S. Alrashed, J. Alhiyafi, and S. O. Olatunji, "Student performance prediction using support vector machine and k-nearest neighbor," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE).* IEEE, 2017, pp. 1–4.

[37] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015.

[38] M. B. Stojanović, M. M. Božić, and M. M. Stanković, "Mid-term load forecasting using recursive time series prediction strategy with support vector machines," *Facta universitatis-series: Electronics and Energetics*, vol. 23, no. 3, pp. 287–298, 2010.

[39] W. Kim, W.-S. Jung, and H. K. Choi, "Lightweight driver monitoring system based on multi-task mobilenets," *Sensors*, vol. 19, no. 14, p. 3200, 2019.

[40] B. Siemiatkowska, M. Majewski *et al.*, "A system for weeds and crops identification–reaching over 10 fps on raspberry pi with the usage of mobilenets, densenet and custom modifications." 2019.

[41] W. Puarungroj and N. Boonsirisumpun, "Recognizing hand-woven fabric pattern designs based on deep learning," in *Advances in Computer Communication and Computational Sciences.* Springer, 2019, pp. 325–336.

[42] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[43] I. Freeman, L. Roese-Koerner, and A. Kummert, "Effnet: An efficient structure for convolutional neural networks," in *2018 25th IEEE International Conference on Image Processing (ICIP).* IEEE, 2018, pp. 6–10.

[44] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[46] T. Akiba, S. Suzuki, and K. Fukuda, "Extremely large minibatch sgd: training resnet-50 on imagenet in 15 minutes," *arXiv preprint arXiv:1711.04325*, 2017.

[47] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision.* Springer, 2014, pp. 391–405.

[48] Z. Zhong, J. Wen, B. Zhang, and Y. Xu, "A general moving detection method using dual-target nonparametric background model," *Knowledge-Based Systems*, vol. 164, pp. 85–95, 2019.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[51] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[52] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.