International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Wind Turbine Damage Detection through Convolutional Neuronal Network

Jose Eduardo Castillo Morales[1], Perfecto Malaquías Quintero Flores[2], José Crispín Hernández Hernández[3], Alberto Reyes Ballesteros[4], Edmundo Bonilla Huerta[5]

[1, 2, 3, 5] Tecnológico Nacional de México. Instituto Tecnológico de Apizaco
División de Estudios de Posgrado e Investigación, Avenida Instituto Tecnológico s/n C.P. 90300, Apizaco, Tlaxcala, México.

[4] Instituto Nacional de Electricidad y Energías Limpias (INEEL), Gerencia de Control, Electrónica y Comunicaciones de la División de Tecnologías Habilitadas Reforma 113, Palmira, 62490 Cuernavaca, Morelos.

Abstract: ***The wind turbines now a days are alternative means to obtain renewable energy, nonetheless, due to the geometry and the environment they are usually affected by different factors such as corrosion caused by the contact with the saltines of the liquids or fractures caused by torsion generated by the wind. Because of this, they require constant monitoring to avoid damaging the integrity of a whole park. This article presents the implementation of a convolutional neuronal network to detect and classify the damages in the external structure of a wind turbine, using different algorithms like image processing filters in-between the layers, MaxPooling, Kernel, SoftMax y Backpropagation.***

Keywords: convolutional neuronal network, wind turbines, image processing, damages

1. Introduction

The renewable energy is being used in most countries because of the benefits that it carries, the wind energy is a sample of this kind of energy and it is obtained by using wind turbines which turn the kinetic energy of the wind into electricity, this is done through the rotation of three blades that have a sophisticated design and are built to work for about 20 years.

However, the steady use provokes the constant development of imperfections caused by factor such as the deterioration, the collision of particles carried by the wind, wind torsion and tension, edge erosion, surface cracks, damaged lighting rods, vortex generators damaged, as so on, which can be seen from the outside since the early stages of their development. Besides some of this damages such as the surface cracking may even indicate severe internal structure damage. Nevertheless, the internal damages such as delamination disunity or internal cracks cannot be detected.

This kinds of damages can cause the blades to not function in an efficient way, causing a series of flaws; a damage in a blade can cause vibration that impacts the rotation of the rotor (spindle) causing from small malfunction to the complete loss of the wind turbine. All of that can cause the feasibility of the park to be affected and also generates economical expenses for several millions of dollars.

Nowadays there is a large potential market that can cover the wind domain, such as design, infrastructure, safety and maintenance, consequently the cost reduction has become a critical factor for this kind of projects to be financially justified and competitive. For all of this the preventive maintenance for this structures must be done frequently, at the same time the minimum high of 80 meters cause this kind of activity to be done using cranes or hydraulic platforms to climb and safety ropes to descend, this and other kinds of methods are used to perform the inspection, this methods beside being highly expensive are also time consuming and put at risk the life of the personnel who does it, and even though most of this are related to wind turbines on land, the marine wind turbines present more difficulty and high costs.

In his article, the implementation of a convolutional neuronal network (CNN) is presented to detect the damages in the external structure of the wind turbines. The CNN classifies the images in different levels of severity, which are obtained by an external device (drone, telescope. Robot etcetera) helping the technician in the wind turbine state analysis and optimizing the preventive maintenance. According to the CNN characteristics it can be considered as a valid method to implement in the damage detection in wind turbines. Due that, the images obtained from this structures that are found in an unstable environment where the illumination varies constantly, and also have color, information and shape fluctuation.

Beside the grate amount of images that are obtained, the manual analysis of each one of them would be an extremely tedious task. The CNN implementation for the processing and sorting of an image can provide better results in the approach and detection of a failure by automatically providing suggestions of experts about the location of the highly probable damages.

2. Related Projects

Despite wind´s energy position among the best ways to obtain electricity. The investigation projects to detect external damages in wind turbines is very scarce. With the grate progress in the current technology the use of remote devices has become more common in this kind of tasks; this is why a drone implementation, a flying unmanned device approach, allows a low cost and frequent inspection, high-resolution images and minimal human interaction [1].

The damage analysis in the structure has been solved by the implementation of neuronal networks unfortunately, the implementation of such required major human supervision and expertise during this process [2]. On the other hand, the condition and damage detection in a wind turbine has been monitored base on SCADA data analysis [3] in this article the internal damages are dealt with this tool but setting aside the external analysis.

However, the continuous monitoring of the turbine´s well-being using damage´s early detection methods can improve the reliability of the turbines and lower the maintenance costs before reaching a catastrophic stage. To achieve the detection and analysis of damages in the wind turbine the article [4] proposes a Deep learning method based in a Deep auto-encoder (DAE) using the operative supervisory control and data acquisition (SCADA) of the wind turbine data. First of all, a model of component network was built DAE using several restricted Boltzmann machine (RBM). Normal wind turbine SCADA data previously collected is use to form this (wise-layer) that is a multi-layer network model to extrapolate the connections between the SCADA attributes.

A prelaminar analysis of the data is done to prove that the SCADA current characteristics are not able to present irregular patterns before age occurrences BREAK- blade. A deep auto-encoder (DA) method is presented to lead an indicator of imminent blade brakes the reconstruction error (RE), from SCADA data. The DA model is a hidden multi-layer neuronal network symmetrically organized. In the training of the DA models the restricted Boltzmann machine is applied to start the weights and biases. The backpropagation method is used later to optimize the structured network even more trough the SCADA data analysis the RE tendency can be observed to be displaced trough the broken blade to detect the ER changes efficiently by monitoring on line, an exponentially movable control chart is deployed. The effectiveness of the follow-up approach proposed is validated by the brake blade in cases gathered in china. The results prove the capability of the approach assessment suggested for the identification of imminent brake blades [5].

The monitoring of the well-being of the wind turbines targets the emerging damages in an early stage to improve the maintenance. The artificial neuronal networks are a tool of machine learning that is commonly used for this purpose; the deep learning is a paradigm for automatic learning based on deep neuronal network that has shown great success in several applications during the last few years [6].

3. Convolutional Neuronal Networks

An artificial simple neuron is an elemental processor in which a vector is processed X (X1, X2, …, Xn) as an input and it produces a response or unique output. The neuronal networks are computational models based in a large set of simple neuronal units; this design is inspired on the structure and functioning of the nervous system in which a neuron is the fundamental element. [7].

Based on the fundaments and constitution, the artificial neuronal network presents a large number of characteristics similar to the ones of the brain. For instance, they are capable of learning from experience; generalize from previous cases to new cases, abstract essential characteristics from seemingly irrelevant inputs, etcetera. This allows them to offer several advantages and also allows this kind of technology to be used in multiple areas [8].

During the las few years Deep learning has been used in image sorting, object tracking, poses the estimation, detection and text recognition, detection of visual prominence, the action recognition and the scene tagging [9]. The applications that involve object recognition and artificial vision (facial recognition) are based on CNNs. The CNNs use for Deep learning has become more popular because it removes the need of manual extraction of the characteristics and learn directly by themselves generating excellent recognition results. Besides, it can be retrained to perform new recognition tasks taking the previous as a base [10].

The CNN is a kind of artificial neuronal network with supervised learning that processes it´s layers in similarity to the visual cortex of the human eye to identify different characteristics in the input that makes it possible to identify objects "see" [11]. The network is formed by multiple hidden and specialized layers and ranked. At the beginning the characteristic extraction phase is found composed by convolutional and reduction networks this means that the first layers can identify lines and curves. As the network advances the dimensions are decreased activating characteristics more complex every time until recognizing an animal or a face. At the end simple neurons are found to perform the classification [12]. These learn using filters and applying them to images. The algorithm takes a small square (or window) and applies it over the image, each filter allows the CCN to identify certain patterns in the image, the CNN looks for parts of the image where a filter matches the content of the image [13].

A famous case of a CNN application was detailed by a research team in Stanford in which the classification of skin lesions was shown using a single CNN [14]. The neuronal network was trained with images using just pixels and illness tags as an input.

3.1 Architecture

The general architecture for a CNN consist input layer(s) hidden layer(s) and an output layer this are several kinds of layer for example:

- Convolutional.
- Activation.
- Pooling.
- Aggrupation.
- Dropout.
- Dense.
- SoftMax.

This layer performs operations that modify the data to learn the specific characteristics of that data. Three of the most common layers are convolution, activation (or ReLU) and pooling [15].

- The convolution makes the input images go through a set of convolutional filters, and each one of them triggers certain characteristics of the images.
- The rectified linear unit (ReLU) allows a faster and more effective training by assigning the negative values to cero and keeping the positive values. This is known in some cases as activation, because only y the activated characteristics follow the path to the next layer.
- The pooling simplifies the output by doing a nonlinear reduction of the simple rate which diminishes the number of parameters that are needed by the network to learn.

These operations are repeated trough tens or hundreds of layers, a process in which each layer learns to identify different characteristics. The convolution layer can be considered a layer in which the filter consists in a small square with a (fixed height and width) that spread out through the total deepness of the input volume [16], every time that it passes it moves in the height and width of the input volume (image).

This process results in a bi-dimensional activation map that provides the output of that filter in each spatial position [17], as is shown in figure 1.

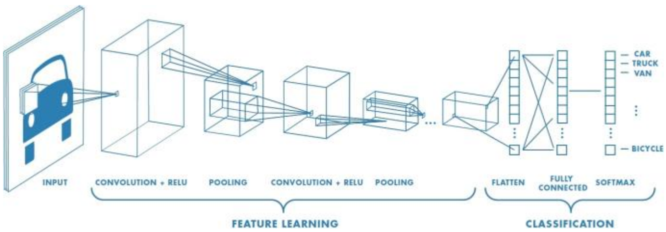


Figure 1: example of a network with several convolutional layers [18].

After learning the characteristics in the many layers the architecture of a CNN goes to the classification phase. The layers called from the proximal to the last one generate a vector of X dimensions, where X is the number of classes that the network is going to be able to predict. This vector contains the probabilities for each class of any image that is being classified. The final layer of the architecture of a CNN is a classification layer, such as Softmax, to generate the classification output [19].

4. CNN Training

For the implementation of a convolutional neuronal network, it is needed for this one to learn on its own to recognize a wide range of shapes, figures and curves for this it was used a great amount of images (100º or more per class) for the network to be able to get each one of the unique characteristics and allows to differentiate each class. Two image bases were used: training and validation each one with a folder of corresponding classes.

The network performs a series of processes where the first stage classifies the images that contain the characteristics, it means that the input images will be suitable for the following stages, for the evaluation of the images a scenery validation has to be performed to stablish each one of the parameters that the image has to have (plain, visual angle, movements and shots of the camera).

In the second stage, they are processed and classified in accordance to the presence of damage in the blades or the pole of the wind turbine.

Finally, if the image presents a damage, this is classified according to the applicable grade of damage (low, medium, high) this class are settled manually based on the criteria researched by the maintenance area of the wind park. Each stage is done by a different network which makes 3 neuronal networks interlinked forming a final network.

4.1 Image Processing

The measurements in pixels measure the total number of pixels in height and width of the image. The resolution is the precision of the images in a bitmap and it is measured in pixels per inch, better resolution.

Generally, the better the resolution the better quality at printing. As a beginning, the network takes as an input the pixels of an image. If we have an image with 1024x683 pixels of height and width, it is equal to 699, 392 neurons.

And if we only have 1 color (gray scale). If we had 3 colors we would need 3 channels Red, Green, Blue (RGB) and the we would use 1042x683x3 = 2, 098, 176 input neurons per image.

For the feeding of the network we would have to normalize the image because the colors of each pixel go from 0 to 255 according to the tone, that is why it is necessary to binarize and apply some filters on the image before putting them as an input in the training of the network.

This time the “Weighted method or luminosity method” was used to change the RGB image to a gray scale according to the next equation:

$$Y = ((0.3 * R) + (0.59 * G) + (0.11 * B)).$$

According to this equation the red color has formed 30% Green has formed 59% and it is the largest of the three colors and blue has formed 11%.

This values are according to the different light intensities in relation to the color that can be seen this is according to the eye response to the visible spectrum, this is why the measurement of the White and black equivalent (gray scale) must be done as a balanced measurement of the different color components for each pixel.

These measurements correspond to the numbers, 0, 2989 for red, 0, 5870 for green y 0, 1140 for blue. To enhance the main characteristics of the image that because of the capturing mechanism or by mistake can be a blurry image, the “prewitt” filter was implemented. In which the operators are based on the gradient estimation using a 3x3 mask. In the figure 2 the two operators are shown, in the *x*-axis direction and the *y*-axis where A is the given image.

$$Gx = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \bullet A$$

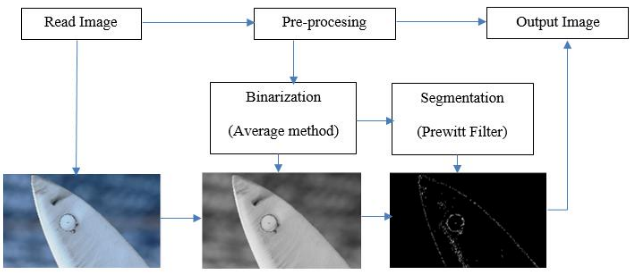
$$Gy = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \bullet A$$

Figure 2: Mask of the prewitt operators'.

The sum of the results taking this mask as a base gives as a result the following estimate of the gradient module:

$$\nabla f \approx |(x7+x8+x9)-(x1+x2+x3)|+|(x3+x6+x9)-(x1+x4x7)|$$

The figure 3 shows the preprocessing stage where a) is taken from the image data base for the network training, b) is the transformed image using the "average method" filter and c) using the final "prewitt" filter.



a) b) c)

Figure 3: Architecture of the preprocessing of the input image.

For the network to learn directionality, it was stablished that one of the images of the training folder to have an angle and close up of 0.3 degrees.

This is for the algorithm to learn that the images are not going to be always in the same position or a complete structure. Figure 4, we can see the small changes in the images according to the new transformation parameters.

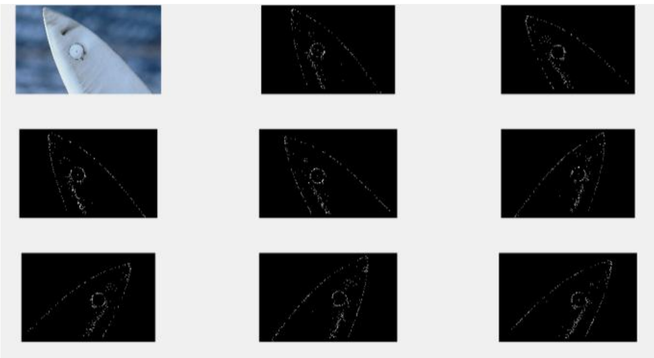


Figure 4: directionality application.

4.2 MaxPooling

The following process is the discretization based in the samples, reducing the dimensionality of the image and allowing the overlap of the contained characteristics in a grouped sub region. This is done on one side to help the excessive adjustment by providing an abstract shape of the representation.

Beside it reduces the computational cost by reducing the amount of parameters to learn and provides an invariability of the basic translation to the internal representation. It means Max Pooling is the application of a mobile window through an input space 2D where the maximum value inside that window is the output as is shown in figure 5.

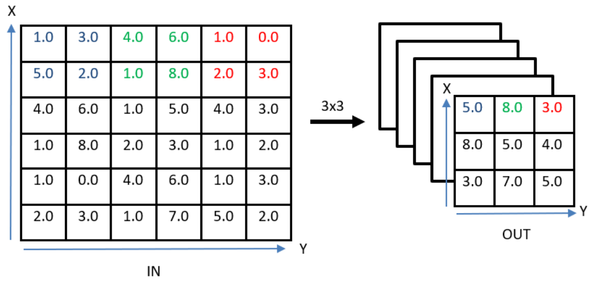


Figure 5: MaxPooling application.

4.3 Convolutions

The stage referring to the network are the convolutional neurons that are in charge of the extraction of characteristics, replacing the simple neurons by matrix processers that make an operation over the image data 2D that go through them, instead of a unique numeric value. The function to calculate the neuron´s output is:

$$Y_j = g\left(b_j + \sum_i K_{ij} \otimes Y_i\right)$$

Where the output Y_j of a neuron j is a matrix that calculates by using a lineal combination of the output Y_i of the previous layer each one of them managed by the convolutional corresponding nucleus Kij in that connection.
That amount is added to an influence b_j and then goes through a non-lineal activation function g (·). In other words, we take a close to the image pixel group and do the algebraic equation of two vectors of same length to give back just one vector (scalar product) against a small matrix that is called kernel.
Where the kernel goes through all the input neurons (from left to right from top to bottom) and generates a new output matrix that definitely is going to be our new hidden neurons layer.

The kernel takes at the beginning random values (1) and are going to be adjusted using backpropagation. (1) one improvement is to make it follow a normal distribution following the symmetry, but the values are random.

During the first convolution 64 filter are applied, which gives us 64 output matrices this is known as "feature mapping"), each one of the 1024x683x1 totaling 44, 761, 088 neurons to our first neuron hidden layer. As we move the kernel, we get a new image filtered by the kernel.

This first 64 filter convolution is similar to get 64 newly filtered images. By applying ReLU for a faster and more effective training. By assigning the negative values a cero value and keeping the positive ones. The function to be used consist on:

$$F(x) = Max\,(0, x)$$

This is sometime known as activation, because just some of the characteristics that had been activated keep going to the next layer.

This new images are drawing some characteristics of the original image, this helps to discriminate the characteristics of the image that we need, this means, if the damage that we have is a crack, or a brake, or even just dirt. We can see how this convolutional process works using the kernel filter in figure 6.

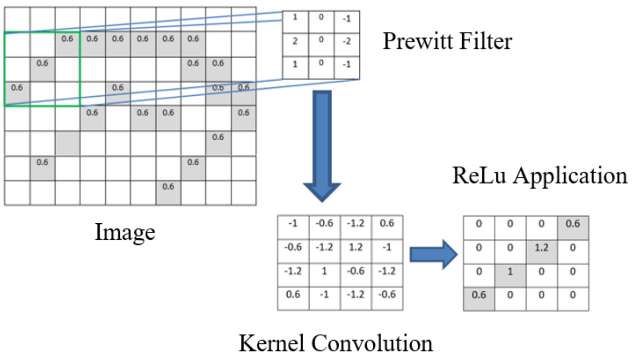


Figure 6: Convolution process with the kernel filter.

As a consequence, the amount of neurons is reduced before the next convolution, because after the first convolution a hidden layer of 44, 761, 088 neurons has been obtained, that in reality are our 64 characteristics maps.

This is why if a new convolution is performed without doing a reduction the layers would be sky high, what would imply more processing, to be able to reduce the size of the next neuronal layer a subsampling process is done in which we reduce the size of the filtered images, but the most important characteristics detected by every filter should prevail.

There are several kinds of sub stamping but the one that is going to be used will be the previously mentioned MaxPooling.

Using this method, the 64 images of previously obtained characteristics but rather than taking 1 pixel we will take 2x2 (2 height per 2 width = 4 pixels) and we will keep the highest value.

The resulting image is reduced to half 512x342 = 22, 380, 544 pixels. And even being less pixels it maintains the most important information to detect the desired characteristics. In chart 1 we compare the 2 convolution and the output results.

Convolution	Input Image	Kernel Application	Mapping	Maxpooling	Output Convolution
1	1024x683x1	64 3x3	1024x683x64	2x2	512x342x32
2	512x342x32	32 3x3	512x342x32	2x2	256x171x32

Chart 1: results of the convolution filters.

4.4 Softmax

To this, his new hidden layer “traditional”, a new function called Softmax was applied that connects with the final output layer that is going to have amount of neurons corresponding to the classes that are being classified.

This is why we have 3 final output neurons: high, medium, low. These outputs at the time of training will have a format known as “one-hot-encoding”. A representation of absolute variables such as binary vectors establishing the next format: high [1, 0, 0]; medium [0, 1, 0] y low [0, 0, 1].

The Softmax function is the one that is in charge of passing the probability (between 0 y 1) to the output neurons. [0, 3 0, 6 0, 1], being an output, this indicates that there is a 30% chance for the damage to be high, 60% to be medium and 10% to be low.

4.5 BackPropagation

Once all the characteristics had been extracted by the previous layers the learning process is implemented using the backpropagation algorithm, this is similar to the traditional network in which we have an input and an expected output (supervised learning) and using Backpropagation the value of the interconnection weights is improved in between the neuronal layers and as we repeat those weights are adjusted to the optimal conditions.

Nonetheless, the value of the weights of the different kernels because is a great advantage at the learning process, as we saw every kernel consist of a reduced size.

The complete network structure we can see it in a graphical way in figure 6. Where each one of the stages with each one of the applied filters used for the final result is shown.

5. CNN Prediction

Once the network has been trained, a series of tests were generated to test the functionality of the network and if it was capable of learning in an autonomous and correct way. The test by stages were also done: if it is appropriate, if it has a flaw and the severity.

During the first stage the input consisted on 50 images (25 adequate and 25 inadequate) we shall remember than the inadequate ones are based on the parameters for image taking. During the second stage the same input was presented 50 images 25 where damage was present and 25 that had no damage.

During the last stage 150 images were used with different damages with a different level of severity where 50 images were used for each severity level (high, medium, low).

An example of the test that was done in each stage is presented in chart 2. The final test of the severity stage containing rights and wrongs in the prediction of each one of the images is presented.

Test: 3 Stage: Gravity								
Images	Prediction		images	Prediction		images	Prediction	
High	Right	Wrong	Medium	Right	Wrong	Low	Right	Wrong
IA1	1	0	IM1	1	0	IB1	1	0
IA2	1	0	IM2	1	0	IB2	1	0
IA3	1	0	IM3	1	0	IB3	1	0
IA4	1	0	IM4	0	1	IB4	1	0
IA5	1	0	IM5	1	0	IB5	1	0
IA6	1	0	IM6	1	0	IB6	1	0
IA7	1	0	IM7	1	0	IB7	1	0
IA8	1	0	IM8	1	0	IB8	1	0
IA9	1	0	IM9	1	0	IB9	1	0
IA10	1	0	IM10	1	0	IB10	1	0
IA11	1	0	IM11	1	0	IB11	0	1
IA12	1	0	IM12	1	0	IB12	0	1
IA13	1	0	IM13	1	0	IB13	1	0
IA14	1	0	IM14	1	0	IB14	1	0
IA15	1	0	IM15	0	1	IB15	1	0
IA16	1	0	IM16	0	1	IB16	1	0
IA17	1	0	IM17	0	1	IB17	1	0
IA18	1	0	IM18	1	0	IB18	1	0
IA19	1	0	IM19	1	0	IB19	0	1
IA20	1	0	IM20	1	0	IB20	1	0
IA21	1	0	IM21	1	0	IB21	1	0
IA22	1	0	IM22	1	0	IB22	1	0
IA23	1	0	IM23	1	0	IB23	1	0
IA24	1	0	IM24	1	0	IB24	1	0
IA25	1	0	IM25	1	0	IB25	1	0
IA26	1	0	IM26	1	0	IB26	1	0
IA27	1	0	IM27	1	0	IB27	1	0
IA28	1	0	IM28	1	0	IB28	0	1
IA29	1	0	IM29	1	0	IB29	1	0
IA30	1	0	IM30	1	0	IB30	0	1
IA31	1	0	IM31	1	0	IB31	1	0
IA32	1	0	IM32	1	0	IB32	1	0
IA33	1	0	IM33	1	0	IB33	1	0
IA34	1	0	IM34	1	0	IB34	0	1
IA35	1	0	IM35	1	0	IB35	0	1
IA36	1	0	IM36	1	0	IB36	1	0
IA37	1	0	IM37	1	0	IB37	1	0
IA38	1	0	IM38	1	0	IB38	1	0
IA39	1	0	IM39	1	0	IB39	1	0
IA40	1	0	IM40	1	0	IB40	1	0
IA41	1	0	IM41	1	0	IB41	1	0
IA42	1	0	IM42	1	0	IB42	1	0
IA43	1	0	IM43	1	0	IB43	1	0
IA44	1	0	IM44	1	0	IB44	1	0
IA45	1	0	IM45	1	0	IB45	1	0
IA46	1	0	IM46	1	0	IB46	1	0
IA47	1	0	IM47	1	0	IB47	1	0
IA48	1	0	IM48	1	0	IB48	1	0
IA49	1	0	IM49	1	0	IB49	1	0
IA50	1	0	IM50	1	0	IB50	1	0
Total	50	0	Total	46	4	Total	43	7

Chart 2: test: 3, stage: Gravity

In chart 3, the next chart the results of the rights of each test and the percentage of the reduction of the wrongs in each one of them is presented.

Stage 3	Test 1		Test 2		Test3		
Gravity	Success	Mistake	Success	Mistake	Success	Mistake	Error Reduction
High	40	10	47	3	50	0	
Medium	36	14	41	9	46	4	
Low	24	26	33	17	43	7	
Percentage	66.66%	44.44%	80.66%	19.44%	92.66%	6.44%	38%

Stage 1	Test 1		Test 2		Test3		
Adequate	Success	Mistake	Success	Mistake	Success	Mistake	Error Reduction
Yes	46	4	49	1	50	0	
No	45	5	47	3	49	1	
Percentage	91.00%	9.00%	96.00%	4.00%	99.00%	1.00%	8%

Stage 2	Test 1		Test 2		Test3		
Faults	Success	Mistake	Success	Mistake	Success	Mistake	Error Reduction
Yes	39	11	44	6	48	2	
No	44	6	47	3	50	0	
Percentage	83.00%	17.00%	91.00%	9.00%	98.00%	2.00%	15%

Chart 3: assessment of the comparative results of the CNN prediction

The error reduction allows us to have a better prediction of the images; this is why the final percentage is satisfying enough to fulfill the task of damage detection. As a final result the images that present damage will be classified in the folder that contains their belonging class.

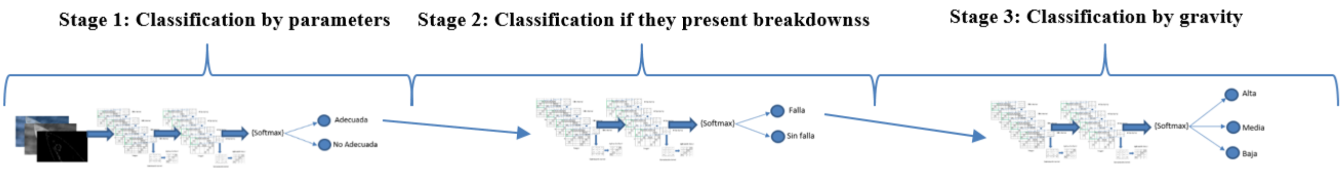
6. Conclusions and Future Work

In this article an image classifier with CNN implementation was presented for damage detection in a wind turbine. This was done through a use of a series of filters to reduce the processing an analysis of an image in a CNN, the binarization method and MaxPooling were used to transform the RGB image to black and white but without losing the important characteristics. In the convolutional stage we used the kernel filter to transform the image in 64 neurons with the respective characteristics applying the Backpropagation technique to correct the weight in between the neurons in the activation function ReLU was used to normalize all of the negative values to 0 and leave the positive results helping to select the important characteristics.

For the second convolution it was necessary to reduce the size of the images to avoid the processing to be too high, for this the new filter MaxPooling was used to make the important characteristics prevail in each filter, in this way the same filter was applied in the second convolution but instead of using 64 filters (neurons), this time just 32 were used. A Softmax algorithm was used for the network connection; this connects the output layer that contains the classifying classes with the rest of the neurons.

The tests that were done proved the feasibility and the speed to select an image that show some damage helping the technician in charge with the tedious task of manually analyzing each one of the images that taken from the large infrastructure that is a wind turbine. The percentage of the learning was noticeable, greatly reducing the error level prediction failure after the testing. Based on the environment and the great amount of information that is presented in the images we can say that this classifying method using CNN provides satisfactory results.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426



Input Image	1 Convolution	2 Convolution	Output Layer: Classification
	Filters	Filters	One-Hot-Encoded
	ReLU	ReLU	
	Subsampling	Subsampling	

Input Image	1 Convolution	2 Convolution	Output Layer: Classification
	Filters	Filters	One-Hot-Encoded
	ReLU	ReLU	
	Subsampling	Subsampling	

Input Image	1 Convolution	2 Convolution	Output Layer: Classification
	Filters	Filters	One-Hot-Encoded
	ReLU	ReLU	
	Subsampling	Subsampling	

Figure 7: Graphic representation of the CNN architecture.

In figure 7 we can see the final architecture of the network, having, as first stage, a convolutional network to classify the parameters, in the following stage a classification is made if damage is present or not using the second network, in this way the final stage can determine the severity.

6.1 Future Work

It is intended to update and improve the CNN to be able to obtain during the prediction satisfactory results at a 100% and leaving the local bases for the image analysis in real time speeding up the process for the immediate detection of damage in the wind turbine structure.

References

[1] Shihavuddin, A. S. M., Chen, X., Fedorov, V., Nymark Christensen, A., Andre Brogaard Riis, N., Branner, K., ... & Reinhold Paulsen, R. (2019). Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis. Energies, 12(4), 676.
[2] Sf Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., & Van Hoecke, S. (2016). Convolutional neural network based fault detection for rotating machinery. Journal of Sound and Vibration, 377, 331-345.
[3] G Dao, P. B., Staszewski, W. J., Barszcz, T., & Uhl, T. (2018). Condition monitoring and fault detection in wind turbines based on cointegration analysis of SCADA data. Renewable Energy, 116, 107-122.
[4] Zhao, H., Liu, H., Hu, W., & Yan, X. (2018). Anomaly detection and fault analysis of wind turbine components based on deep learning network. Renewable energy, 127, 825-834.
[5] Wang, L., Zhang, Z., Xu, J., & Liu, R. (2018). Wind turbine blade breakage monitoring with deep autoencoders. IEEE Transactions on Smart Grid, 9(4), 2824-2833.
[6] Helbing, G., & Ritter, M. (2018). Deep Learning for fault detection in wind turbines. Renewable and Sustainable Energy Reviews, 98, 189-198.
[7] Ketkar, N. (2017). Deep Learning with Python (pp. 159-194). Apress.
[8] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification, " 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), (Beijing, 2017, pp. 721-724.
[9] X. Wang, L. Liu, G. Li, X. Dong, P. Zhao and X. Feng, "Background Subtraction on Depth Videos with Convolutional Neural Networks, " 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-7.
[10] R. C. Çalik and M. F. Demirci, "Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems, " 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, 2018, pp. 1-2.
[11] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
[12] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network, " 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6.
[13] K. P. Korshunova, "A Convolutional Fuzzy Neural Network for Image Classification, " 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC), Vladivostok, 2018, pp. 1-4.
[14] Aurélien, G. (2017). Hands-On Machine Learning with Scikit-Learn & TensorFlow.
[15] S. Kido, Y. Hirano and N. Hashimoto, "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN), " 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, 2018, pp. 1-4.
[16] H. Yanagisawa, T. Yamashita and H. Watanabe, "A study on object detection method from manga images using CNN, " 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, 2018, pp. 1-4.
[17] M. E. Ruaro, P. Bonifazi and V. Torre, "Toward the neurocomputer: image Processing and pattern recognition with neuronal cultures, " in IEEE Transactions on Biomedical Engineering, vol. 52, no. 3, pp. 371-383, March 2005.
[18] Recuperado el 01 de Abril del 2019, de: https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html
[19] Lakhani, P., & Sundaram, B. (2017). Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology, 284(2), 574-582.

Author Profile

Jose Eduardo Castillo Morales computational systems engineer. Instituto Tecnológico de Villahermosa/Tabasco, México, 2016. He is currently a student of the Master's degree in computational systems in the research, development and implementation of intelligent technologies in the Instituto Tecnológico de Apizaco/Tlaxcala, México and his investigation is related to the implementation of a CNN to find damages in the external structure of a wind turbine.

Perfecto Malaquías Quintero Flores. Industrial engineer in production (manufacture) by the Instituto Tecnológico de Apizaco (ITA), Apizaco, Tlaxcala, México. Master's degree in computer science by the centro de graduados e investigación del Instituto Tecnológico de Toluca, Metepec, Edo. De México. PhD in computational sciences in l'Université Montpellier 2 (UM 2) in the laboratory for computational sciences, robotics and microelectronic of l'Université Montpellier 2 and the national center for scientific research of France (LIRMM-CNRS) in Languedoc- Roussillon, France. He obtained the certification in PSP, by Software Engineering Institute – Carnegie Mellon University, on June .2014.

José Crispín Hernández Hernández. PhD in computation by the Université d´ Angers, Campus Belle-Beille, France (2018). Areas of interest: data science. Data mining, fuzzy logic, evolutionary computing, artificial vision y neuronal networks.

Alberto Reyes Ballesteros. PhD in computer science by the ITESM CVA (2007), master´s degree in artificial intelligence (IA) mention of honor by the LANIA-Universidad Veracruzana (UV) in 1999, bachelor's degree in mechanic electrician engineer by the FIME-UV in 1989. postdoc in the Instituto Superior Técnico (IST) of the Universidad de Lisboa in the laboratorio de sistemas inteligentes. In 1990 he was admitted in the geotermoelectric "Los Humeros" CFE as a mechanic resident later he was admitted by the INEEL where he has developed national and international artificial intelligence projects for the electric sector. His areas of interest are: processing and data mining, automatic learning, and use of graphic probability models for prediction and decision making. he is currently a professor in the postdoc for the INEEL. He is also a member of the Sistema Nacional de Investigadores (SNI) level I.

Edmundo Bonilla Huerta. PhD. In computation by the Université d´ Angers, Campus Belle-Beille, France (2018). Areas of interest: data science. Data mining, fuzzy logic, evolutionary computing, artificial vision y neuronal networks.