

# Convolutional Neural Network and OpenCV Based Mobile Application to Detect Wear out in Car Tyres

Allipilli Harshitha<sup>1\*</sup>, Samyuktha Samala<sup>2</sup>

<sup>1,2</sup>Department of Computer Science and Engineering,  
Stanley College of Engineering and Technology for Women, Hyderabad, India.

Corresponding author: \*[allipilliharshita@gmail.com](mailto:allipilliharshita@gmail.com)

Received Date: 1 February 2021

Accepted Date: 22 March 2021

Published Date: 30 March 2021

---

---

## HIGHLIGHTS

- The wear out and not wear out car tyres were collected from stores around the streets and the data is augmented.
- Used OpenCV for preprocessing the image i.e., removing noise and distortion from the images.
- Deep Learning was used to develop the classifier model in order to classify between the wear out and not wear out tires.

---

---

## ABSTRACT

This work proposes a technique for detecting wear out of car tyres. Tyre is the only part of the vehicle which is in contact with road. Hence tyre condition should be monitored timely in order to have a safe drive. Tyre wear out occurs because of the parameters such as rubber degradation, when there are around 4 to 5 punctures, bulged tyre. We consider some of the above parameters to assess the wear of tyre using the computer vision techniques such as OpenCV and convolutional neural networks. OpenCV and convolutional neural networks are most used in object detection and image classification. We used these techniques and obtained an accuracy of 90.95%, with which we can predict the wear of tyre to avoid dangerous accidents.

**Keywords:** *OpenCV, convolutional neural networks (CNN), tyre wear out.*

## INTRODUCTION

Tyre is a ring-shaped component that surrounds the wheel's rim to transfer a vehicle's load from the axle through the wheel to the ground and to provide traction on the surface travelled over. Tyres are designed in such a way that it can be able to bear the load and the pressure of the vehicle. Tyre is made up of materials like synthetic rubber, natural rubber, fabric, wire, carbon black and other chemical components. Tyre contains tread and body. Tread provides traction and body provides containment for quantity of compressed air.

Tyre is the only part of the vehicle which is in contact with the road. The tyre performance should be monitored timely in order to maintain safety. A layman might not know when a tyre is going to wear out. If the tyre is not replaced in time then there might be chances of fire, heavy damage and might



cause accidents. To avoid such fatal situations, a device or a mobile application is to be developed with, which it can be known when a tyre is wear out.

Tyre contains tread which is one of the essential elements to be considered in order to check whether the tyre is wear out or not. Tread wear occurs through normal contact with road or terrains. There are several types of abnormal tread wear that were caused due to poor alignment of wheel, gravel roads, rocky terrains, rough terrains, over-inflation, under-inflation, unbalanced wheels, rubber degradation and other parameters.

A 4-digit number is present on the tyre which is the month and year when the tyre is manufactured, tyre should be replaced within 6 years from the date of manufacture or else the rubber degrades and results in accidents. If the tyre is punctured for 4 or 5 times, then it is considered as wear out and should be replaced. Tyre should be properly fitted, inflated and balanced on to the car. If the tyre is not parked properly or if rubber degrades then cracks or bulges occur which causes tyre burst and leads to fatal accidents.

### **What if we do not change the tyre after wearing out?**

There are various disadvantages that occur if the tyre is not changed on time after wear out such as brakes may fail, the control over the vehicle is lost, cannot handle the vehicle, tyres might lose grip and friction on certain road conditions like wet and icy roads, vehicle loses grip even on the dry roads, causes the car to skid during sudden stops, mileage of car reduces, cannot go for long drives, if so then it leads to burst of tyre causing fatal accidents.

## **LITERATURE REVIEW**

In order to know about the groundwork behind this tyre wear out detection, we need to go through few of the works published by other authors. As we go through below, we see that groundwork on tyre wear out is very minute and the work is more focused in mechanical terms. The methodology which we used is computer science tools, hence we get to know more about the methodology we use to solve this kind of problem.

Praful Darekar et.al (2018) researched on detection of tyre wear out using colour coding technique with the use of sensors. Different colours of colour coding are added to tyre. Those colours can be visualized by humans with which the change of colour of tyre can be detected easily to ensure that the tyre is wear out.

Manfred (2014) conducted an experiment on the parameters that causes wear of tyre. Certain parameters like tyre design, model, material used, pressure applied, temperature and surface of road that results in the wear of tyre.

Tyre pressure monitoring system (TPMS) device is developed by Sivaraos et.al (2019) that monitors the temperature and pressure of tyre. Tyre pressure and temperature plays an important role in safety of tyres which should be monitored properly.

### **Convolutional Neural Networks**

Convolutional neural network is a deep learning technique which is used to classify the images. CNN takes an image and applies weight and bias to it to classify the images properly. CNN was introduced



to recognize handwritten zip code (LeCun et.al, 1989) and later it was used to classify the MNIST images which are the handwritten digits (LeCun et.al, 1998). Now-a-days CNN is used to classify objects (Szegedy et.al, 2014, Krizhevsky et.al, 2012), detect objects (Szegedy et.al, 2013, Girshick (2015), Erhan et.al (2014), Ren et.al, 2015), objects motion detection in video streaming (Ryoo et.al, 2014).

CNN contains 4 layers which are used to train the model effectively to classify the images properly. The functionality of each layer is detailed below.

### a. Convolution Layer

It is the topmost layer in the architecture of CNN. It contains certain filters such as Sobel and Scharr filters. These filters size must be less than the input image. A dot product is computed in between filter and the input image at each and every position by sliding around the input image. At last all the products are summed up to get a single number. These values are then sent to the next layer (Qihong Ke et.al, 2018).

### b. Activation Layer

Activation functions are used in this layer which decides whether a neuron should be active or to be dropped in the neural network. Activation functions that are broadly used are: Sigmoid, Tanh, Relu, leaky Relu.

Sigmoid: The sigmoid function is

$$F(x) = \frac{1}{1+e^{-x}} \quad (1)$$

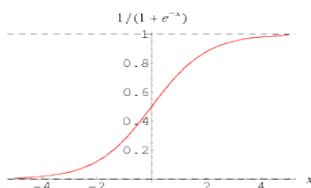


Figure 1: Sigmoid

Sigmoid function is S-shaped curve which returns output in a interval [0,1].

Tanh: The Tanh function is

$$F(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (2)$$

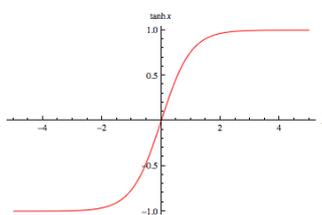


Figure 2: Tanh

The hyperbolic tangent (tanh) function returns output in the interval [-1,1].

The disadvantage of sigmoid and tanh functions is that they easily saturate to 0,1, -1 ranges which reduces the model training performance.



Relu (Rectified Linear Unit): The Relu function is

$$R(x) = \max(0, x) \quad (3)$$

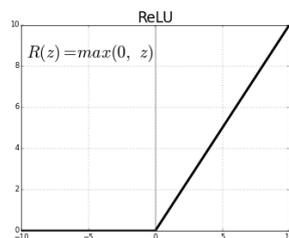


Figure 3: Relu

The advantage of Relu is that it does not saturate, and it gives fast and accurate output. Relu works better with large amount of data. The disadvantage of Relu is it causes neurons to fire or dead sometimes.

Leaky Relu: To overcome the cause of Relu, we use leaky Relu. The leaky Relu function is:

$$R(x) = \begin{cases} x & x > 0 \\ 0.01x & otherwise \end{cases} \quad (4)$$

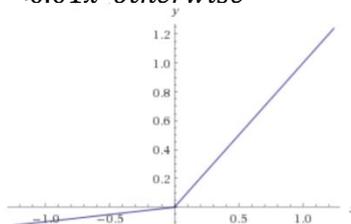


Figure 4: Leaky Relu

Leaky Relu allows non-zero values though the function is not active.

### c. Pooling Layer

Pooling layer down samples the features that are obtained from previous layers. Hence it is also known as down sampling layer. It reduces the noise and distortion by selecting only the important features from the previous layers that is it selects only the maximum value to retain maximum information (Jarrett et.al, 2009). There are 2 types of pooling layers. **Average pooling** which takes the average of all the pixel values from each window. **Max pooling** is used to find maximum value from each window to retain maximum information of the image.

### d. Fully Connected Layer

Fully connected layers take neurons from previous layers and connects to every activation unit of next layers. This layer is used to compile the values obtained from previous layers to give the final result. In this layer we use activation functions such as sigmoid and SoftMax to classify the images into classes.

### OpenCV

Open computer vision is open source computer vision library. OpenCV library has more than 2500 algorithms which are related to computer vision and machine learning algorithms. These algorithms



are used to detect, recognize and classify the objects. OpenCV is also used to increase the quality of image.

Meghna Raj Saxena et.al (2019) conducted an experiment on object detection using machine learning and OpenCV. Object detection is identifying objects in images and in video streaming. MATLAB tool is used to detect behaviour of humans to implement CC TV surveillance systems. Zainab et.al (2010) developed a CC TV surveillance system to place at the sideways of the road to capture the images and video solve the traffic and accident related issues.

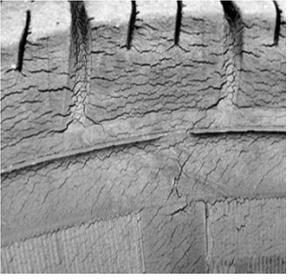
Nilesh et.al (2013) conducted an experiment to monitor the vehicles and traffic on the highways. The author used visual C++ with intel OpenCV video streaming system in order to capture the images and video to detect and monitor the movements of vehicles. Rahul et.al (2017) conducted a research to detect license plate number from video or images. The image is converted into text using OpenCV libraries and tesseract to recognize character patterns from the image.

## DATASET DISCUSSION

Many kinds of tyre images in different resolutions, noise characteristics and with variety of camera types are collected. The types of tyres that were collected are cracked, blast, punctured, new and old images. Images are taken from different angles, distances, different lighting conditions, some are blurred and covered with dirt. Some of the images are collected from internet and some are collected by going to the shops manually which is a time-consuming process.

All these images were segregated as wear out and not wear out in separate folders to determine the classes to be classified by the CNN. The wear out tyre images are around 300 and not wear out tyre images are around 290 representing a total training image of 590. The test images are separated same as wear out and not wear out images considering 210 images for testing.

**Table 1:** Shows the wear out images of tyres

<b>Burst</b>	<b>Puncture</b>	<b>Crack</b>	<b>Bulge</b>
			



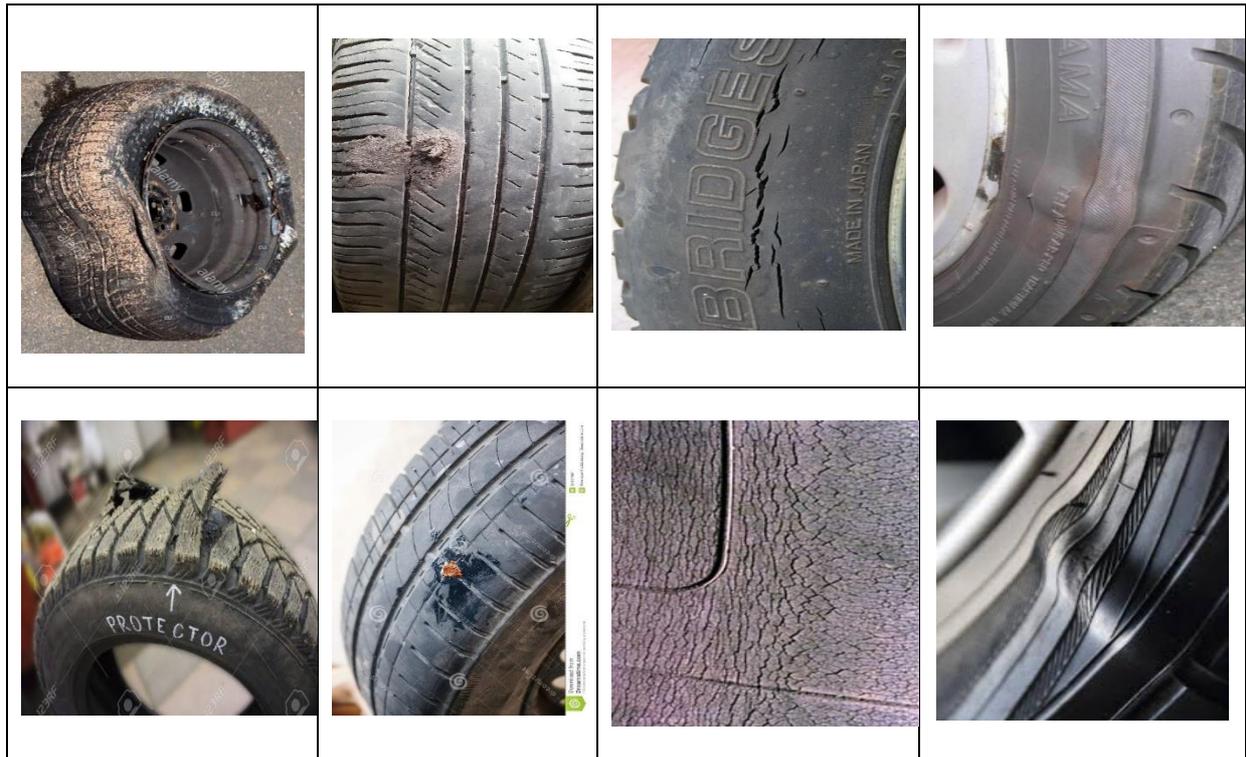


Table 1 shows the images of wear out tyres. Column 1 contains the images of tyres that are blasted or burst which are not to be used at any cost because they are completely wear out. Column 2 contains the images of tyres that are punctured, tyres once punctured might occur one more time also, if the punctures on tyre are more than 4 then the tyre need to be changed. Columns 3 contains the images of tyres that are cracked, tyres which are cracked need to be changed immediately because it leads to blast and results in accidents. Column 4 contains the images that are bulged, this type of tyres must be changed immediately since it leads to blast of tyre.

**Table 2:** shows the wear out images of tyres

Different Designs	Light Conditions	Stucked stones
		



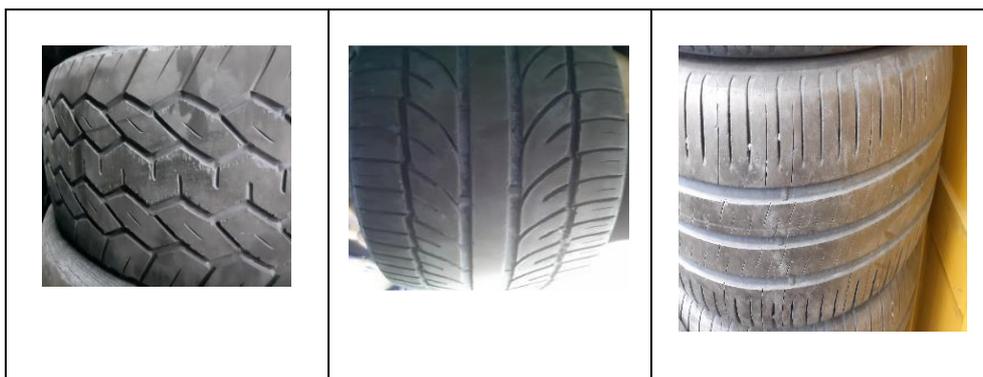
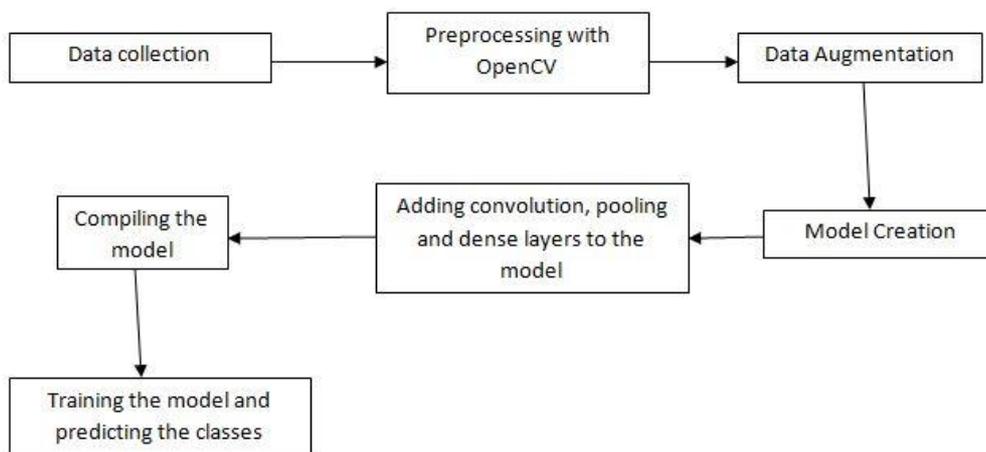


Table 2 shows the images of not wear out tyres. Column 1 contains the images of tyre with a new design which looks like as if it is wear out but it is actually the shape, hence we have included such kind of different designed tyre images into the data set so as to make the model train with all kinds of tyres and classify it properly. Column 2 contains the images of tyre that are taken in different light conditions. Column 3 contains the images of tyre that is filled with small stones, but this tyre can be used as it is not considered as wear out since the stones are small and does not cause any affect to the tyre.

## PROPOSED METHODOLOGY

The main purpose of this project is to detect and classify the tyre whether it is wear out or not to reduce the effect of accidents. Considering all the parameters that were discussed in the Introduction section such as how to identify the wear and what are the causes if tyre is not changed on time. Hence, we have implemented the model using OpenCV and CNN to classify the wear effectively. The methodology/architecture that was followed is discussed below.



**Figure 5:** A graphical representation of process of proposed methodology

The graphical representation of process of methodology proposed is efficient and appropriate for the detection of wear of tyre through images.

The first step is collection of data of various kinds of tyres like punctured, blast, cracked, new and old tyres (which we discussed in dataset discussion section). These data are collected with different noise



and distortion and at different lighting conditions and from different angles and distance to train the model with all the types of images.

The second step is pre-processing with OpenCV techniques such as converting the BGR image to RGB image since when an image is read it reads as BGR i.e., blue has more priority than other two, so we convert it to RGB making red to have high priority and applied bilateral filter which is used to blur the side ways to reduce the noise and distortion in order to make the image look clear without much disturbance.

The third step of pre-processing is data augmentation. We applied techniques such as rotation, rescaling, width shift, height shift, shear range, zoom range, and horizontal flip to make the image to be viewed in every angle and from any distance on training and testing data sets.

After augmenting the data, we create our model in a sequential manner. Then the layers such as convolution layer along with Relu, pooling layer, and dense layer are added on to the model. The first layer is the convolution layer in which the input image is taken and a 3x3 filter such as Scharr filter or Sobel filter or user-defined filter is applied on to it. These filters are used to identify and highlight gradient edges or features of the image and these filters size must be less than input image size. In our model we have used Sobel filter as it can detect all the vertical and horizontal edges of the image.

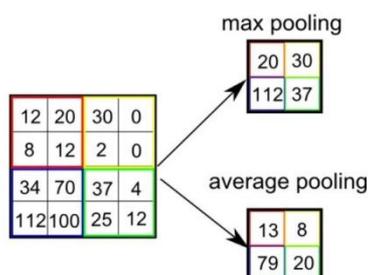


**Figure 6:** Sobel and Scharr filters

Using the Sobel filter, the dot product of height and width is computed between input image and the filter. The filter slides out through the image and computes dot product and finally all the values are summed up to get a single value. If the image size is 6x6 and the filter size is 3x3 then the output of sum of dot products is 4x4 which is computed with the formula  $[N - F + 1]$ , where N is the input image size and the F is the filter size. By doing the above the input image loses some amount of information since the image is shrinking and losing data of edges on applying a filter. In order to overcome this, we do padding i.e., if the size of the image is 6x6 then we give as 8x8 by padding and apply a filter then we get the exact image size even after applying a filter. The formula to do padding is  $[N + 2P - F + 1]$ , where P is padding size. The output from convolution layer is taken as input to Relu which is an activation function that converts the negative values present in the data to positive values. Hence Convolution and Relu together are used to make image as single and positive.

The next layer is pooling layer. Pooling layer is also known as down sampling layer since it reduces the matrix size of the image and retains maximum information from the image. The working of pooling layer is discussed below.





**Figure 7:** The working mechanism of pooling layer

The data from convolution layer is taken and it converts into 2x2 matrix using max pooling or average pooling. In max pooling it divides the matrix into 2x2 matrix each and takes max value from each to retain the maximum information from the image and to shrinks the image so as to consume less space in the memory. In average pooling layer it takes average of all the values, this pooling function is used rarely. Most often we use only max pooling function since it stores the maximum information from the image.

The final layer is dense layer or fully connected layer. Dense layer is used to sum up all the neurons from previous layers and makes a final classification result. In our model we used single Conv2D and MaxPooling layers, two dense layers are added in which one dense layer is with 256 neurons and the other dense layer is the final layer with single neuron which is used to obtain the classification result.

After all the layers are created for the model, we compile the model. Compiling the model is necessary in order to optimize the algorithm and increase the accuracy and reducing the loss. The main aim of compilation is to minimize the loss. A detailed summary of layers applied in the model are explained in the figure below.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 62, 62, 64)         1792
-----
max_pooling2d (MaxPooling2D) (None, 31, 31, 64)         0
-----
dropout (Dropout)           (None, 31, 31, 64)         0
-----
flatten (Flatten)           (None, 61504)              0
-----
dense (Dense)                (None, 256)                15745280
-----
dense_1 (Dense)             (None, 2)                  514
-----
Total params: 15,747,586
Trainable params: 15,747,586
Non-trainable params: 0
    
```

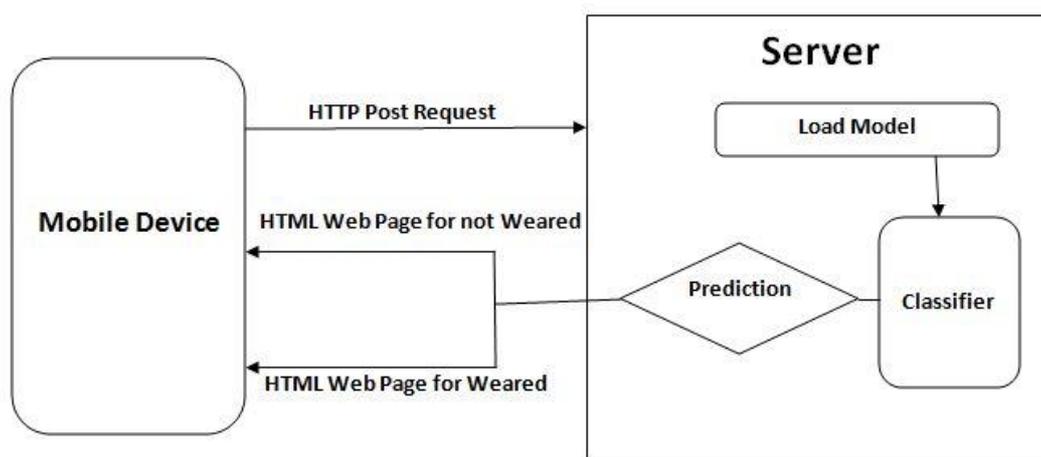
**Figure 8:** The summary of the model is detailed

After model creation and compilation, we train the model with training dataset and validation set. Early stopping monitor function is used in order not to over fit or under fit the data. Next a new set of images are given for predicting the classes, with which we ensured that our model could classify the tyres effectively.



## MOBILE IMPLEMENTATION

After designing the process in order to make CNN model effective we have come up with an idea of implementing on mobile devices. Users with the help of their tire pictures can find condition of their tires. Deploying CNN model require an environment that supports tensor flow runtime. Many cloud services like AWS and IBM can be used. We have opted Flask for deploying model, which is a Web Framework that is used to build web applications. The connection between server and application is obtained with WSGI and jinja2 template that is used for dynamic web pages. After design process we save the model in the .h5 file.



**Figure 9:** This figure shows the working of the mobile device. A user sends image of tyre to know the state of it by sending a request. Then the server loads the model and the server contains a classifier which runs on the uploaded image file and an HTML web page is generated to the user as a result.

For mobile application we initially need to load .h5 model, after initialization of app we need to specify route in order to redirect into different web pages based on the application. Html pages are included using render template in flask. In this application we have created html page with CSS and JavaScript. While html page is saved in template folder, style sheets and scripts are to be saved in static folder. Input is given to the server from html page. Using request object, we will be able to access incoming data, this data can be of any format, these images are decoded into raw format and the model is used for prediction of output. Different functions are created based on the functionalities to store and load model. We need to check whether the method is GET or POST. If it is a get request, we simply open same web page. If it is Post request, then we need to process the data. Based on the result the flask server we provide one among two html pages. In this application we have created safe and unsafe web pages if the condition is true it then redirects to safe web page indicating user that this tyre is in safe condition, if the condition is false then it redirects to unsafe web page which indicates that his tyre is unsafe and he need to replace the tyre. Debug mode is used to reload server continuously whenever changes are made this also acts as debugger whenever there is error. Flask implementations are best and easiest way as they can easily render to any webpage based on the result and can provide a simple and better interface to users.

## RESULTS AND DISCUSSION

The experimental results are presented here. The experiment was conducted with 590 training images and 210 test images. These training images are computed along several layers as discussed in the



above with several epochs, where epoch is defined as the ratio of product of number of iterations and batch size to the total number of training images. Here we have given batch size as 40 per epoch and number of iterations are 30 in which it takes 590 training images. Precisely one epoch is all data processed one time. Each epoch is validated with the validation data set so as to make sure that it does not over fit and the assumptions are correct. Validation data set is used to cross check whether the trained model is correctly predicting the value or not for the new data without being over fitting or under fitting. With this training and validating we have obtained 90.95% of accuracy with 0.26% of loss for 30 epochs.

**Table 3:** Confusion matrix for test data with n=210

	Actual not wear out	Actual wear out
Predicted not wear out	105	11
Predicted wear out	8	86

Table 3 shows the confusion matrix for the test data with 210 images. Confusion matrix is used to check error and accuracy percentage. Confusion matrix determines true positive, false positive, true negative and false negative rates. Where true positive rate is determined if actual and predicted values are same and are positive, true negative rate is determined if actual and predicted are same and negative, false positive and false negative are determined if actual and predicted values are different.

From the above table, we saw that the true positive rate i.e., the actual and predicted values for not wear out is 105, the true negative rate i.e., the actual wear out and predicted wear out is 86, false positive rate i.e., the actual not wear out but predicted wear out is 8, false negative rate i.e., the actual wear out and predicted not wear out is 11.

Accuracy is how often is our classification correct. The accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Samples}} \quad (5)$$

Using the above formula, the accuracy is calculated, and we obtained a desired value of 90.95%.

Misclassification rate or the loss rate is how often is our classification wrong. The loss rate is calculated as

$$\text{Loss rate} = \frac{\text{False Positive} + \text{False Negative}}{\text{Total Samples}} \quad (6)$$

Using the above formula, the loss rate is calculated, and we obtained a value of 0.09%.

**Table 4:** Classification report for the test data

	Precision	Recall	F1-Score
Not wear out	0.86	0.89	0.92



Wear out	0.90	0.80	0.89
Total/average	0.92	0.91	0.91

Table 4. details the classification report for the test data with 210 images. A classification report is used to measure the quality of predictions i.e., how many predictions are true and how many are false. This can be said through calculating precision, recall and f1-score.

Precision allows not to label an instance positive when it is actually negative. Precision is defined as ratio of number of correct positive responses to the number of all positive responses i.e., when it predicts correct classes, how often it is correct? Precision is calculated as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (7)$$

From Table 4, the precision for not wear out is 86% and wear out is 90% correct.

Recall allows to find all positive instances. Recall is defined as ratio of number of correct positive results to the number of all relevant samples i.e., for all instances that were actually positive, what percent was classified correctly? Recall is calculated as:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (8)$$

From Table 4, the recall for not wear out is 89% and wear out is 80% correct.

F1-score is a measure of a test's accuracy. F1-score considers precision and recall of the test data to compute the score. F1-score is calculated as:

$$\text{F1-score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \quad (9)$$

From Table 4, the F1-score for not wear out is 92% and wear out is 89% correct.

The model has obtained an accuracy of 90.95% for the test data. With which the model assures that it can be able to classify the images as wear out or not wear out. Hence the tyre can be replaced on time avoiding the dangers and ensures safety to the vehicles.

## CONCLUSION

Tyre is the only part of vehicle which is in contact with road. Hence, it is to be primarily considered. Since a layman does not know when the tyre wears out. To implement a model which can be able to detect and classify the wear out tyre images require tools like OpenCV and CNN. This paper presents a model that has used these two tools and techniques to classify the images and has given an accuracy of 90.95%, with which we can be able to classify it in order to play an important role in safety of vehicle users.



## REFERENCES

- Erhan, D., Szegedy, C., Toshev, A. & Anguelov, D. (2014). “Scalable object detection using deep neural networks”, *In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, Pages 2155-2162.IEEE, 2014.
- Girshick, R. “Fast r-cnn”, (2015) *In International Conference on Computer Vision (ICCV)*, 2015.
- Jarrett, K., Kavukcuoglu, K., Ranzato, M. & LeCun, Y. (2009). “What is the best multi-stage architecture for object recognition?”, *In Computer Vision, 2009 IEEE 12<sup>th</sup> International Conference on*, Pages 2146-2153.IEEE, 2009.
- Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). “Imagenet classification with deep convolutional neural networks”, *In Advances in Neural Information Processing Systems*,
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. & Jackel, L.D. (1989). “Backpropagation applied to handwritten zip code recognition”, *Neural Computation*, 1(4):541-551, 1989.
- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, 86(11):2278-2324, 1998.
- Manfred Klueppel. (2014). “Wear and abrasion of tires”, *In book: Encyclopedia of Polymeric Nanomaterials*, Pages 1-6, 2014.
- Meghna Raj Saxena, Akarsh Pathak, Aditya Pratap Singh, Ishika Shukla. (2019). “Real-time object detection using machine learning and opencv”, *International Journal of Information Sciences and Application (IJISA)*. ISSN 0974-225, Vol.11, No.1, 2019.
- Nilesh J. Uke, Ravindra C. Thool, (2013). “Moving vehicle detection for measuring traffic count using opencv”, *Journal of Automation and Control Engineering Vol.1, No.4*, 2013. Pages 1097-1105, 2012.
- Praful Darekar, Akash Saste, Shubham Bhandarkar, Nikita Kadam, B.R. Patil. (2018). “Detection of tyre wear using colour coding”, *International Research Journal of Engineering and Technology (IRJET)*, 2018.
- Qihong Ke, Jun Liu, Mohammed Bennamoun, Senjian An, Ferdous Sohel, Farid Boussaid, (2018) “Computer vision for assistive healthcare”, *In Computer Vision and Pattern Recognition*, 2018, Pages 127-145, 2018.
- Rahul R. Palekar, Sushant U. Parab, Dhrumil P Parikh, Vijaya N. Kamble, (2017) “Real-time license plate detection using opencv and tesseract”, *International Conference on Communication and Signal Processing (ICCSP), IEEE*, Pages 2111-2115, 2017.



- Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks", *In Advances in Neural Information Processing Systems*, Pages 91-99, 2015.
- Ryoo, M.S., Rothrock, B. & Matthies, L. (2014). "Pooled motion features for first-person videos", *arXiv Preprint arXiv: 1412.6505*, 2014.
- Sivaraos, M.J. Raguvaran, Aidy Ali, M.F. Abdullah, D. Sivakumar, M.A.M Ali, A. Hambali, Ms. Salleh, K. Umesh. (2019). "Investigation of tyre pressure drop phenomenon using specially designed real-time data mining and storage system", *International Journal of Recent Technology and Engineering(IJRTE)*, 2019.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). "Going deeper with convolutions", *arXiv Preprint arXiv: 1409.4842*, 2014.
- Szegedy, C., Toshev, A., & Erhan, D. (2013). "Deep neural networks for object detection", *In Advances in Neural Information Processing Systems*, Pages 2553-2561, 2013.
- Zainab Nazar Khalil Wafi, Badlishah Ahmad, R., & Paulraj, M.P. (2010). "Highways traffic surveillance systems (HTSS) using opencv", *2010 IEEE Control and System Graduate Research Colloquium (ICSGRC 2010)*, *IEEE*, Pages 44-48.

