Detection of Defective and Non-Defective Fuse Configurations of the Fuse Boxes used in Wiring Harnesses in Automobiles with Deep Learning

[1]Avaneesh Khandekar, [2]Arnav Rokde, [3]Tanmay Gokhale, [4]Siddharth Yawatkar, [5]Swati Shekapure

[1,2,3,4]Student, [5]Assistant Professor
[1,2,3,4,5]Computer Engineering,
[1,2,3,4,5]Marathwada Mitra Mandal's College of Engineering, Pune, India.

Abstract : The fast and accurate automated quality visual inspection is increasingly gaining importance in manufacturing and product quality control for product efficiency. To effectively detect faults or defects in products, main methods focus on hand-crafted optical features. An important part in all automobiles is the fuse box which provides protection to all electronic devices in a vehicle such as headlights, AC, radio, indicators, etc. The fuse box contains on several fuses placed in their position with the help of color coding and ampere rating. The current practice is to conduct manual inspection, which leads to high production cost and other quality issues. Our approach focuses on use of a Convolutional Neural Network to extract powerful features with less prior knowledge about the images for defect detection. This can improve the quality of the fuse boxes shipped for final assembly of the vehicle and reduce its cost of manufacturing and testing. In this report, we suggest a method for Implementation of Convolutional Neural Network to classify the fuse box configurations into correct and wrong classes based on color coding of fuses which can be extended in a similar way to other manufacturing quality control problems.

IndexTerms **- Convolutional Neural Network,** ***Feature map, Fuse, Regressor, Softmax, quality inspection.***

I. INTRODUCTION

Computer vision has become one of the critical fields of study, and the industrial applications that are driven by the use of computer vision methodologies are becoming a huge part of industry. The precision and speed at which images taken from cameras are processed and classified have evolved over decades. Deep learning plays a major role as a computer vision device as the well-known boy in town.

With the advent of deep learning image processing technology we found a deep learning approach with some advantages over conventional computer vision techniques. The problem with a traditional image classification approach to extraction of features is that you have to choose which features to look for in each image. The principal difference in computer vision's deep learning approach is the idea of end-to-end computing. Defining the features and doing feature engineering is no longer required. The neural is doing the same for you. Our approach focuses on using a Convolution Neural Network to extract powerful features for identification of defects, with less prior knowledge of images.

Production quality control is a crucial field for evaluating the value of a product. Fuse boxes are designed for shielding the car's electrical circuits from exposure to the elements to prevent damage and short circuits. Fuses are made to control and safeguard electrical currents which flow into electrical components via wires. When fuses are blown, drivers may experience problems with the radio, dome lights, and other electrical components inside the vehicle. The fuse box provides color coding and ampere rating on different fuses put in their location. Inside the fuse case, fuses come in several different shapes, colors and sizes. They are often used to stabilize the electrical current which flows through wires, protecting devices from damage due to electrical overload. Most fuses in vehicles today are either rectangular or cylinder shaped. Rectangular fuses come with two push in-connectors which are connected by a fuse wiring covered by a plastic cover that blows when overloaded. Fuses are the protectors of the electrical devices in your car. Relays within the fuse box help safeguard passengers from the high voltage that the battery and alternator produce. Therefore, it is important that no mistakes are made when mounting the fuses in their frames.

The standard method is manual testing, which leads to high cost of production and other quality problems. Both the fuse boxes cannot be exhaustively checked in manual inspection. There is a reduction of upstream labor, consumables, plant capacity as well as revenue if a defective component is found at the end of the manufacturing line. At the other hand, if an undetected bad component gets into the final product, there will be both customer impact as well as market reaction. It could actually do irreparable harm to the reputation of the organization. We suggest an automated system for the testing phase before fuse boxes are assembled in vehicles to save time and increasing the manufacturer's productivity.

II. PROPOSED APPROACH

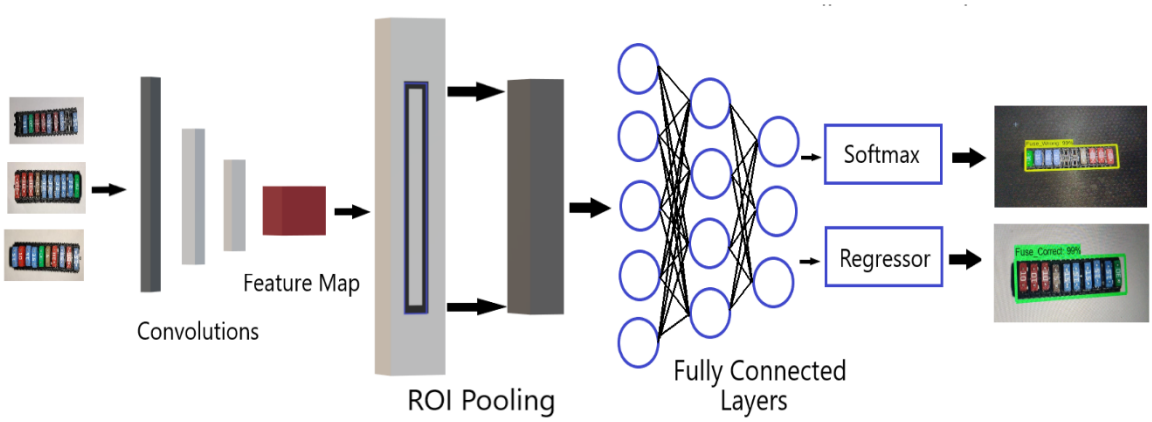

Figure 1:System Architecture

A. Convolutional Layer

A Convolutional Neural Network is a Layer Neural Networks which takes in an input image, assign features to various objects in the image differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in traditional methods filters are hand-crafted, with enough training, CNNs have the ability to learn these characteristics. Convolution refers to the mathematical combination of two functions to produce a third function. Hence, it merges two sets of information. The objective of the Convolution Operation is to extract the high-level features such as edges, color activations from the input image. Multiple such convolutional layers are added into the system to extract high level features with different set of kernels (filters).

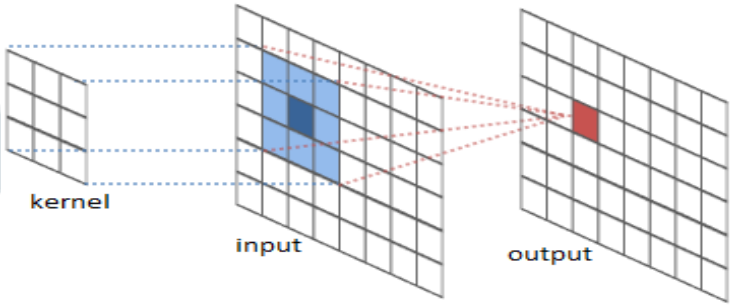

Figure 2:Convolutions

B. Feature Maps

The feature map is the output of a kernel applied to the previous layer in a CNN. A given filter is drawn across the entire previous layer, moved one pixel at a time. Each position results in an activation of the neuron and the output is collected in the feature map. In a convolutional neural network units within a hidden layer are segmented into "feature maps" where the units within a feature map share the weights, or in simple terms look for the same feature. In a feature map, the hidden units are special in that they are linked to different units in the lower layer. Thus units inside a feature map are bound to specific regions of the input image for the first hidden layer. The Network's hidden layers perform feature extraction from the image. This process produces a vectorized form that contains activations for the image. The image is divided into grids of size 1 pixel and weights or some values are generated for each particular pixel. The pixels contributing to activation have a higher value compared to the ones which don't activate any neuron.

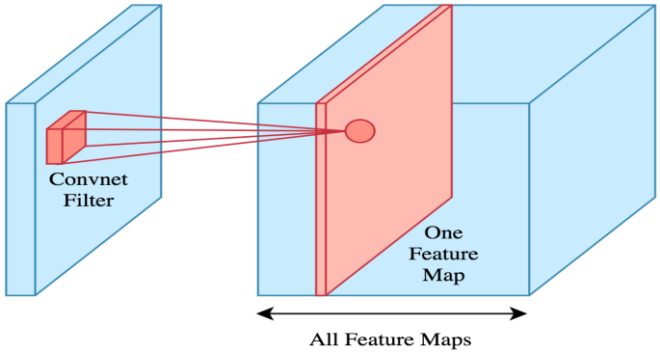

Figure 3: Feature Maps

C. Fully Connected Layer

This layer serves as a classifier on top of the extracted features which were detected during convolutions and pooling operations. This layer only accepts one dimensional data, as a result flattening the input is necessary. Neurons in this layer have full connections to all the activations in the previous layer. Fully connected layers are an essential component of Convolutional Neural Networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision. Fully connected layers in a CNN should not be confused with fully connected neural networks-the traditional neural network model in which all neurons in the next layer connect to all neurons from the previous layers.

D. Softmax Classification

Softmax is an activation feature of a kind. Different activation functions are used, depending on the number of classifications needed and the data used. A softmax layer, allows a multi-class method to run on the neural network. In short, the neural network will now be able to calculate the likelihood of the dog being in the picture, as well as the likelihood of additional objects being in the picture. Softmax is applied just before the output layer, using a neural network layer. The Softmax layer must have the same number of nodes as the output layer.

E. Bounding Box Regressor

Bounding-box regression is a common technique used by recent object detection approaches to refine or predict localization boxes. Usually, bounding-box regressors are trained to return to nearby bounding boxes of a predefined target object classes from either area proposals or fixed anchor boxes. In object detection, we typically use a bounding box to identify the position of the target. The bounding box is a rectangular box that can be determined by the coordinates of the x and y axis in the upper-left corner, and the coordinates of the x and y axis in the lower-right corner.

F. Pooling Layer

Pooling Layer is commonly inserted in between successive convolution layers or between convolution and RELU layers. It reduces the number of parameters and computation in the network. It continuously reduces the dimensionality which results in shortened training times and controls overfitting. The aim is to perform maximum pooling on non-uniform size inputs in order to obtain feature maps of a fixed size. This stage's performance should be a list of bounding boxes of possible object positions. These are often called region proposals or regions of interest. Typically we need to generate a lot of regions of interest in the proposal process. If during the first stage (region proposal) an object is not identified, it cannot be properly identified in the second phase. That's why it's extremely important for the region proposals to have a high recall. And that's achieved by generating very large numbers of proposals (e.g., a few thousands per frame). In the second stage of the detection algorithm, most of them will be marked as background.

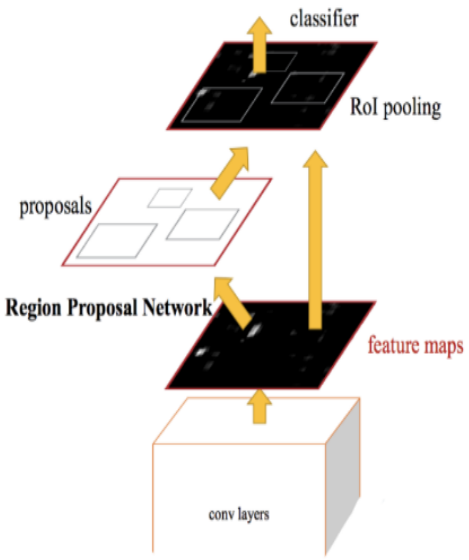

Figure 4: ROI Pooling

The following equations can be used to calculate the output size of each convolutional and pooling layer:

$$output\_size_{conv} = \left\lfloor \frac{input\_size + 2 * pad - [dilation * (kernel\_size - 1) + 1]}{stride} \right\rfloor + 1$$

$$output\_size_{pool} = \left\lceil \frac{input\_size + 2 * pad - kernel\_size}{stride} \right\rceil + 1,$$

Figure 5: Output Size

III. FUSE

Fuse can be classified into two types correct and wrong. Wrong fuse can be further categorized into two types which are misplaced and missing. A non-defective Fuse is show in Fig.6 where the configuration is in right order. Defective fuses are shown in Fig.7 and Fig.8 where the fuse configuration is not in the correct order or it is missing.

Figure 6: Non-Defective Fuse

Figure 7: Defective Fuse(Incorrect Order)

Figure 8: Defective Fuse(Missing fuse)

IV. METHODOLOGY

In this section, a system is proposed to identify defects in the fuse image. The design of the defect detection system is based on the Faster R-CNN architecture. As shown above there will be two categories for fuses to be classified which are correct and wrong.

A. Faster R-CNN

R-CNN was developed by R. Girshick in 2014 is the first move in faster R-CNN that uses a search selective to find regions of interests and passes it on to a ConvNet. By combining similar pixels and textures into multiple rectangular boxes, it tries to figure out which areas could be an entity. Faster R-CNN is making more progress than Fast R-CNN. Area Proposal Network (RPN) is replacing selective search mechanism. RPN is a network for proposing areas, as revealed by the name. For example, if the input image has 600x800x3 dimensions, the output feature map will be 37x50x256 dimensions, after having the output feature map from a pre-trained model (VGG-16).

Step 1: Data Gathering and Augmentation

A mobile phone is used for taking pictures of correct and wrong fuses. The images are then resized in the dimension 800*600 pixels. 800 represents the length and 600 represents the breadth of the image. The total number of pictures taken is 300 where 150

are for correct fuse configurations and rest for wrong fuse configuration. For the data augmentation part the images are flipped and mirror so bringing up the dataset size to 1200. The images are then rotated to get vertical images. So the final dataset size comes to 2400 images in which there are 1200 correct fuse configuration images and 1200 wrong fuse configurations.

Step 2: Extensively Labeling Dataset

Label the bounding boxes with correct labels for the correct and wrong classes. LabelImg is used to label images and draw bounding boxes for each image. This generates XML files having the attributes: Xmin, Ymin, Xmax, Ymax, ClassName.
XML files are converted into CSVs for further processing. The CSV files are created to store information for the coordinates of the bounding boxes drawn on the images. These CSV files are then converted into TFRecords that serve as input data to the Tensorflow training model which in our case is the Faster R-CNN model.

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	flippedIMG_20200126_143846.jpg	800	600	Fuse_Correct	216	161	534	253
3	flippedIMG_20200126_143850.jpg	800	600	Fuse_Correct	196	184	523	278
4	flippedIMG_20200126_143902.jpg	800	600	Fuse_Correct	200	197	536	291
5	flippedIMG_20200126_143909.jpg	800	600	Fuse_Correct	180	146	546	241
6	flippedIMG_20200126_143913.jpg	800	600	Fuse_Correct	210	253	534	343

Figure 9: CSV Format

Step 3: Training the model

A label map is created after generating the TFRecords which tells the trainer what each object is by defining a mapping of class names to class ID numbers. A text editor is used to create a labelmap.pbtxt file which contains the number of categories of classes our dataset is divided into. Finally, you must configure the training pipeline for object detection. It specifies which model should be used and which parameters for the training. Next to begin training we specify where to store the logs, the path to the pipeline configuration and the model to be used. The max number of steps has been set to two lakh with an early stop feature by providing a keyboard interrupt. The progress of the model can be viewed using tensorboard. Tensorboard is a visualization tool provided by tensorflow to view various graphs for your model during training. Tensorboard is also used for viewing histograms of biases, weights or tensors as they change during the training of the model. Figure 6 shows the graph between the number of steps used to train the model and total loss during the training period.

Step 4: Saving model

After the model is trained successfully the final move is to produce the frozen inference graph. A frozen inference graph is a frozen graph that cannot be trained anymore, it defines the graphdef and is actually a serialized graph. To evaluate a model various methods can be used which include plotting ROC curve, calculating mAP (mean average precision). Precision is defined as the ratio between the true positive (TP) and total number of positive predictions which is TP+FP where FP stands for False positives. To calculate mean average precision the following formula is used where AveP is the average precision and Q is the number of queries. We find the average precision for any given query q.

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

Figure 10: Precision Calculation

The mean average precision is calculated using the above equation. It is calculated every ten minutes during training and is done up to eleven thousand steps. Eventually the model was trained until a hundred and forty thousand steps.

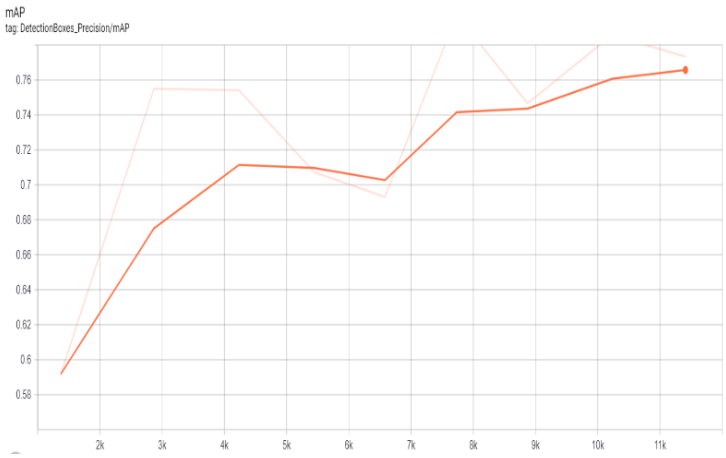

Figure 11: Mean Average Precision

The total loss is the classification loss over 2 classes (A Defective / Non-defective fuse is present or not).
This loss is calculated for the detection network which takes input from the Region Proposal Network (RPN) in the form of locations and bounding boxes.

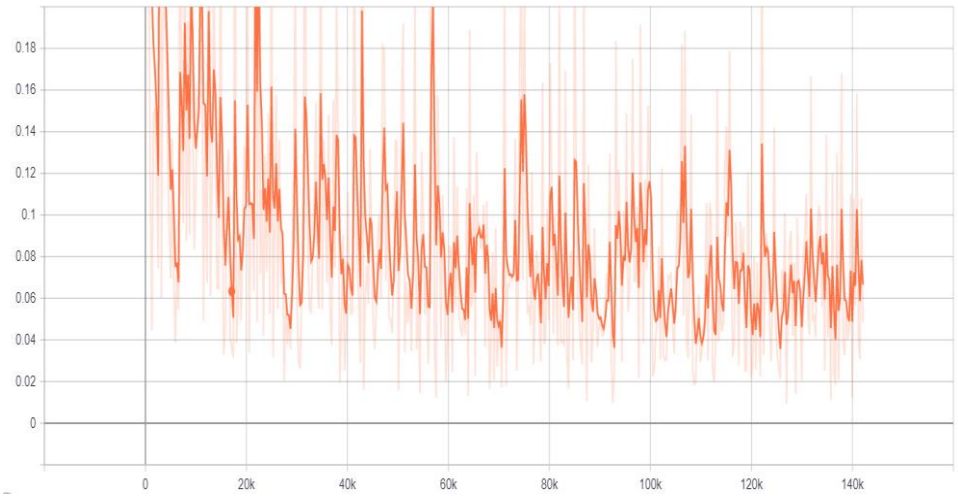

Figure 12: Total Loss

The localization loss denotes the model's ability to correctly localize the fuse in a given image.

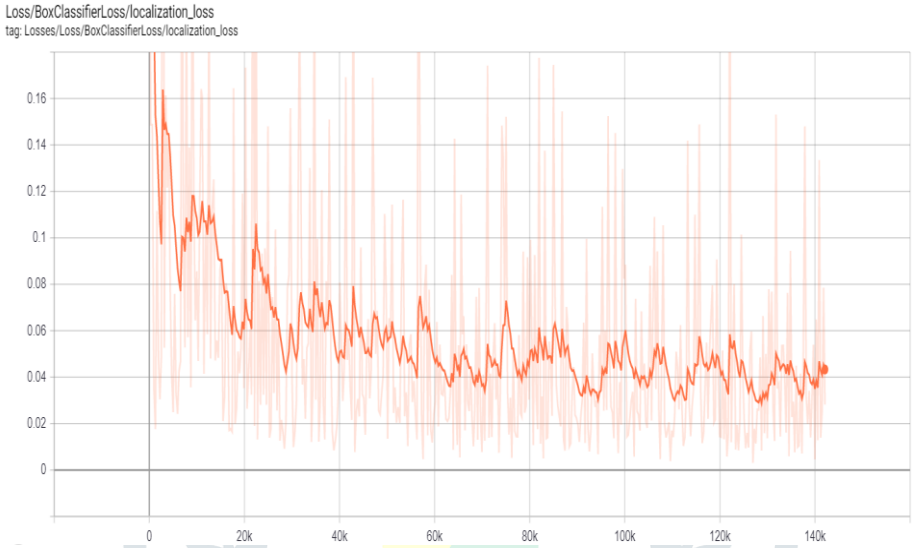

Figure 13: Box Classifier Localization Loss

The RPN Loss denotes the model's ability to check which locations contain the fuse and the corresponding locations and bounding boxes will pass on to the detection network.

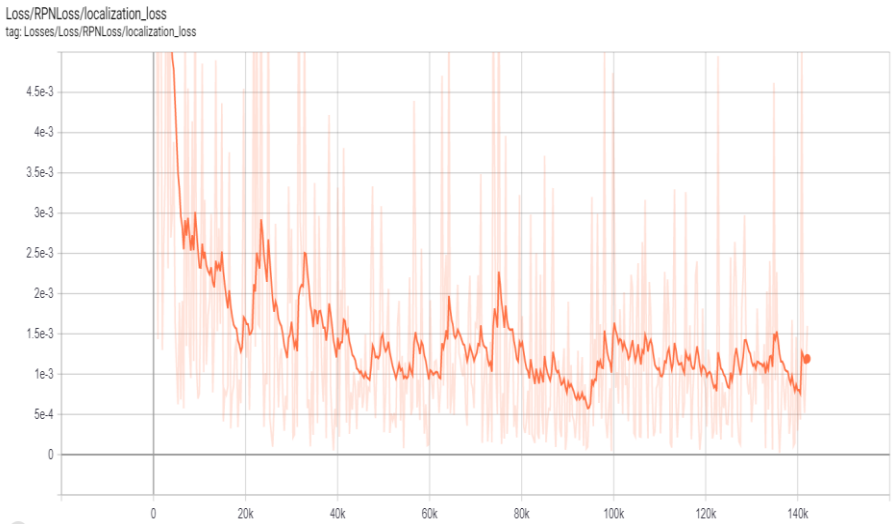

Figure 14: Region Proposal Network Loss

Step 5: Testing

Test on validation set by giving unseen data as input to the saved model (inference graph). The algorithm predicts the Regions of Interest (ROI) in unseen images and does classification. Predicting ROIs is very accurate with this although classification into correct and wrong is not up to the mark. We believe with more data augmentation it will give better results.

V. RESULTS

With the help of Deep Learning the defects in the fuse box can be identified. The captured images can be seen by the user. A report will be generated for the user, based on the defects found. The error rate will be generated by the report and can be seen by the user. The defective images will also be available for the user to see. The user can see the defect rate per hour and defect rate per day with the help of intuitive graphs. Following correct fuse configuration images were tested and predicted correctly

Figure 15: Detection (Non-defective)

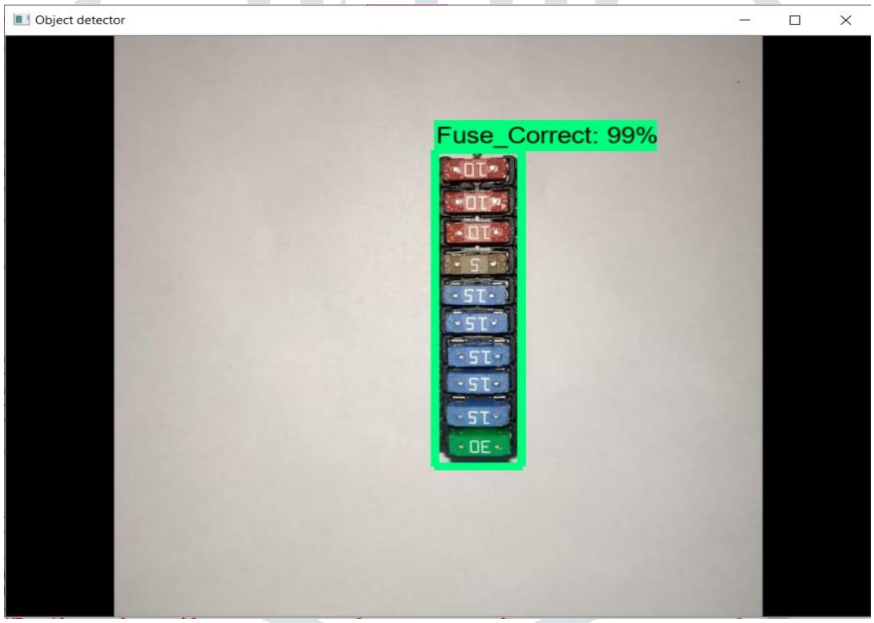

Figure 16: Detection (Non-Defective)

The other images used for testing which have wrong configuration were also predicted correctly by the model.

Figure 17: Detection (Defective)

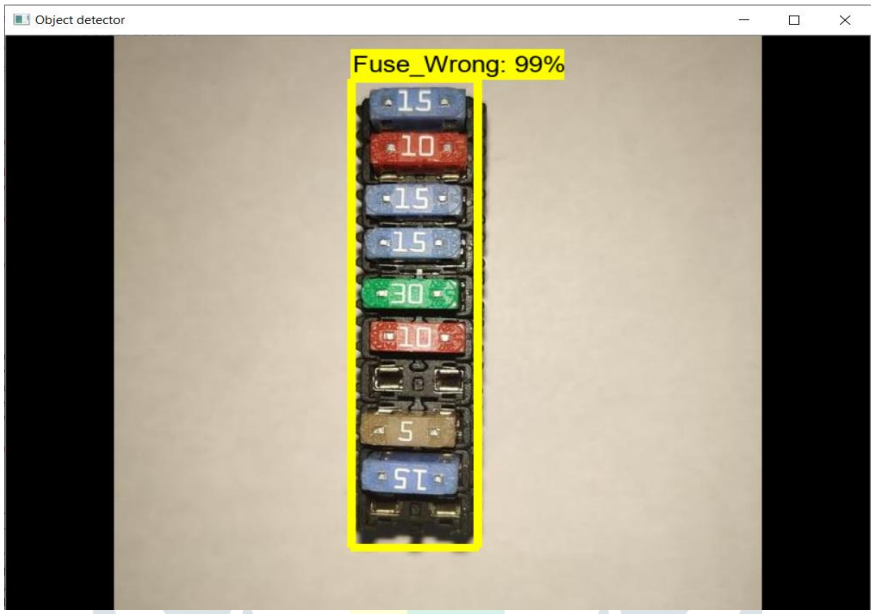

Figure 18: Detection (Defective)

VI. CONCLUSION AND FUTURE WORK

In this work, we test the ability of state-of-the art Deep Learning technologies like Faster R-CNN, CNN & ResNet to automatically extract information from images. The results show that using deep-learning technology can save a tremendous amount of time for manufacturers. Automating the quality control process can thus dramatically reduce the cost to upstream labor and reconfiguration. A productivity boost is achieved by the system in the manufacturing process. The model is able to classify single images within a duration of 5 seconds because of the model loading overhead. The model takes about 3 seconds per fuse for detections when supplied with multiple images. Hence, with an efficient of 100 fuses / 5 minutes. A manual system would take about 100 minutes per 100 fuses and lead to errors due to operator fatigue (100 minutes per 100 fuses estimate is based on expert opinion). When used in an automated conveyorized system, the model runs with an efficiency around 1 second per fuse. The misclassification rate is very low as depicted in the loss graphs above and hence, the system can be trusted to be used solely for the inspection work.

Automatically generating intuitive reports will help manufacturers in fast decision making and increase their productivity. The proposed methodology can be extended to any manufacturing plant with any type of objects which requires real time and fast defect segmentation. This work will give a chance to manufacturers to employ new techniques for quality management with minimum human interaction. This method combined with the technologies of IOT can result into a completely automated system for catching manufacturing defects.

REFERENCES

[1] Erhan Akın, Ilhan Aydın, Canan Tastimur & Mehmet Karakose, 2016. Rail defect detection with real time image processing. IEEE 14th International Conference on Industrial Informatics (INDIN).

[2] Alaa Hefnawy & Mohamed wafaa Elewa Hussam Elbehiery, 2005. Surface defects detection for ceramic tiles using image processing and morphological techniques. The Third World Enformatika Conference.

[3] Weng kin LAI JiWei OOI, Lee Choo Tay, 2019. Bottom hat filtering for defect detection with cnn classification on wiper arms. IEEE 15th International Colloqium on Signal Processing & its Application.

[4] Kincho H. Law Max Ferguson, Yung-Tsun Tina Lee, 2018. Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. arXiv Cornell University.

[5] Alisan Sarimaden & Erhan Akin Mehmet Baygin, Mehmet Karakose, 2017. Machine vision based defect detection approach using image processing. International Artificial Intelligence and Data Processing Symposium (IDAP).

[6] Zhaoqing Pan Oumayma Essid, 2018. Automatic detection and classification of manufacturing defects in metal boxes using deep neural networks. Plos One.

[7] Rohit Mittal Parth Deka, 2019. Quality inspection in manufacturing using deep learning based computer vision. Towards Data Science.

[8] Guojun Wen Shuang Mei, Yudan Wang, 2018. Automatic fabric defect detection with a multiscale convolutional denoising autoencoder network model. Sensors, Molecular Diversity Preservation International and Multidisciplinary Digital Publishing Institute.