



Fabric Defect Detection Using YOLOv2 and YOLO v3 Tiny

R. Sujee^(✉), D. Shanthosh, and L. Sudharsun

Department of Computer Science and Engineering, Amrita School of Engineering,
Amrita Vishwa Vidyapeetham, Amrita University, Coimbatore, India
r_sujee@cb.amrita.edu, shanthoshdevaraj@gmail.com, sudhas1728@gmail.com

Abstract. The paper aims to classify the defects in a fabric material using deep learning and neural network methodologies. For this paper, 6 classes of defects are considered, namely, Rust, Grease, Hole, Slough, Oil Stain, and, Broken Filament. This paper has implemented both the YOLOv2 model and the YOLOv3 Tiny model separately using the same fabric data set which was collected for this research, which consists of six types of defects, and uses the convolutional weights which were pre-trained on Imagenet dataset. Observed and documented the success rate of both the model in detecting the defects in the fabric material.

Keywords: YOLOv2 · YOLOv3 Tiny · Fabric defect detection

1 Introduction

In the growing trend of Artificial Intelligence, machine learning, and deep learning, many man-made tasks are automated and developed. But still, there are many industries to be revolutionized in these technologies. In that an important industry is textile. In textile industries, quality inspections are one of the major problems for fabric manufacturers as it requires lots of man-hours and wages. Currently, in textile industries, inspection is carried out manually, due to this there are possibilities of human errors with high inspection time. To overcome this problem, automating the process of finding defects is necessary.

In this paper, two object detection algorithms YOLOv2 and YOLOv3 Tiny are used to solve the problem of finding defects in fabric material with better accuracy and minimum loss with less computational time to suite the industrial rapidness. Six different defects:-Oil Stain: Discoloration of fabric due to oil; Holes: A small opening in fabric; Broken Filament: Individual filaments constituting the main yarn are broken; Rust: Presence of red or yellow color coating made of iron oxide formed by oxidation, especially in the presence of moisture in the fabric; Grease: A black thick oily substance in fabric; and Slough: A thick bundle of yarn is wrapped into the fabric in the direction of the weft due to the slip of the yarn coils from the pirn during weaving; are classified and localized.

Supported by Amrita School of Engineering, Coimbatore.

© IFIP International Federation for Information Processing 2020

Published by Springer Nature Switzerland AG 2020

A. Chandrabose et al. (Eds.): ICCIDS 2020, IFIP AICT 578, pp. 196–204, 2020.

https://doi.org/10.1007/978-3-030-63467-4_15

2 Literature Survey

Paper [1], proposed a new descriptor based on an mutual information, which is then used as input to neural networks. One and two hidden layer of neural networks were used. The result revealed that the recognition rates were 100% for training and 100% for generalization.

In paper [2] it was shown that image quality can be enhanced by improving contrast using a histogram matching and thus texture classification can be carried out in a better and efficient way.

Paper [3], suggested a method based on local homogeneity and mathematical morphology. The spatial homogeneity of each pixel was calculated to create a new homogeneity image. This image was placed in a series of morphological and Structuring Element operations to determine the feature of the existing fabric. The functionality of the system was evaluated extensively using various fabric samples, which varied in defect type, size and shape, text background, and image resolution.

In paper [4], it was proposed to provide a diagnostic procedure for detecting and classifying defects in warp and weft using a computer program developed by MATLAB that analyzed images of fabric samples obtained using a flat scanner. Three were determined to detect defects. Finally, a neural network was introduced to identify the class of defects present in the fabric.

Paper [5], described a system of pre-processing train images collected by a web crawler. The result showed before developing the system, two important factors have to be taken care of. First of all, the size of the training image and the acquisition image should be the same, second, the size of the object's location should be the same as the training image and the detection image. Six classes were used in the study and that satisfied the YOLO training value.

In paper [6], the 276 images associated with the textile fabrications were collected, pre-processed and labeled. After that, YOLO9000, YOLO-VOC, and Tiny YOLO were used to build models for fabric detection. Through a comparative study, YOLO-VOC was chosen for further development by optimizing the super-parameters of the deep neural network. In this paper, only 3 types of defects are identified, namely, Belt yarn, Knot tying, and Hole, with less number of defect images. In the proposed paper, we try to identify 6 defects as mentioned above, using YOLOv2 and YOLOv3.

In paper [7], a working model was developed for portable devices such as a laptop or mobile phone which lacks the Graphics Processing Unit (GPU). The model was first trained in the PASCAL VOC dataset then it was trained in the COCO database. It was seen that the YOLO-LITE speed was 3.8x faster than the SSD Mobilenetv1.

Paper [8] suggested several improvements to the YOLO adoption method, both novel and based on previous work. The improved model, YOLOv2, performed better while compared to Faster RCNN with Resnet and SSD. Also, YOLO9000 was trained in the COCO acquisition and ImageNet classification data and it was shown that YOLO9000 was able to predict more than 9000 different object classes.

3 Proposed System

3.1 Choice of YOLO

From the literature survey, it was clear that YOLO outperforms other object detection frameworks. The result of the comparative Performance of YOLO2 and other state-of-the-art detection systems from paper [8] is shown in the Table 1. As mAP and FPS is better for YOLOv2 we have chosen the YOLO model to detect defects in the fabric. For optimal FPS and mAP, we chose YOLOv2 with input size 416×416 .

Table 1. Performance comparison

Detection frameworks	Train	mAP	FPS
Faster R-CNN VGG-16	2007 +2012	73.2	7
Faster R-CNN ResNet	2007 + 2012	76.4	5
YOLOv2 288×288	2007 + 2012	69.0	91
YOLOv2 416×416	2007 + 2012	76.8	67
YOLOv2 544×544	2007 + 2012	78.6	40

Table 2. Sample annotation data

Class	x1	y1	x2	y2
1	0.24609	0.70361	0.41601	0.59277
1	0.77197	0.75976	0.12793	0.46679

3.2 Data Set Creation

First step in proposed system is to collect fabric material with and without defects used in real factories and capture the images of those fabric materials from different orientations and angles. Then the images should be pre-processed and made to meet the dimensions of $416 * 416$ pixels.

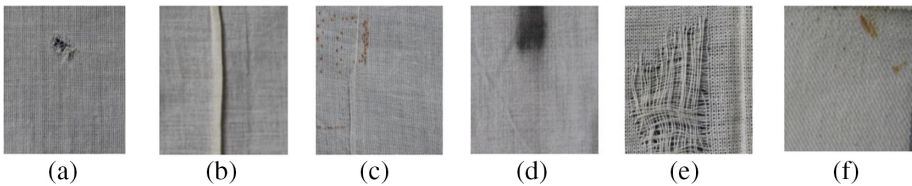


Fig. 1. Dataset (a) Hole (b) Slough (c) Rust (d) Grease (e) Broken Filament (f) Oil Stain

After preprocessing images, in order to feed them to YOLOv2 it is necessary to annotate the images with rectangle boxes specifying their defect area in the following format. Table 2 shows the format and sample values of center of defect, width and height of the defect with respect to the image. Figure 1 displays the sample defect images from the dataset used in training YOLOv2 and YOLOv3 Tiny.

Each of the fields in Table 2 represent following data (Table 3):

Table 3. Number of images used for each class

Class	Hole	Rust	Broken filament	Oil stain	Slough	Grease	Total
Defect images	80	160	100	85	75	100	600

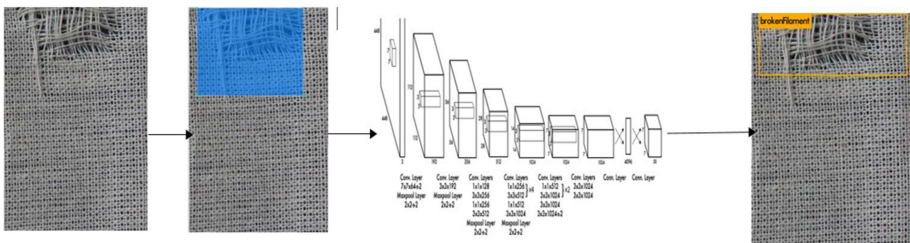
- class: Represents the type of defect present in the corresponding box
- x1, x2: Center of defect with respect to the image
- y1, y2: Width and Height of the defect with respect to the image coordinates.

3.3 Object Detection Algorithm

In this proposed system, it is not only necessary to classify the images as defects and not defects but it is also important to identify their place of occurrence for improving the manufacturing process in the future. In order to achieve this, paper uses YOLO object detection algorithm.

You Look At Once (YOLO), is a real-time object detection system, and is a neural network that can see what is in the image and where content is, just in one passing. It offers bounding boxes around discoveries and can detect multiple objects at a time, providing a huge advantage when comparing it to other neural network models.

To train neural networks, a framework called Darknet is used, it is open source and serves as the basis for YOLO written in C/CUDA. The original repository, by J Redmon. Darknet is used as the framework for training YOLO, i.e it sets the architecture of the network. Figure 2 shows the flow of the proposed method for training starting from capturing the defect, annotating the defect image, training the model using the annotated image obtained from the above step. During test phase when image is given to the trained model it gives the output as show in the last step of flow diagram.

**Fig. 2.** Flow diagram

YOLOv2 Model was mainly aimed at improving recall and localization while maintaining the accuracy of the classification. For better performance, techniques such as batch normalization, high resolution classification, convolution with anchor boxes were incorporated. The architecture of Darknet-19 is the backbone of YOLOv2.

YOLOv3 Tiny Model. In YOLOv2 there are No residual blocks, no skip connections, and no up-sampling. YOLO v3 incorporates all of the above features. YOLOv3 use a new network for performing feature extraction. The latest network is a hybrid approach of YOLOv3, and residual network, so it has some shortcut connections. The Darknet-53 architecture is the backbone of YOLOv3. Tiny-yolov3 is a simplified version of YOLOv3, which contains a smaller number of convolution layers than YOLOv3, in turn the neural network doesn't need to take up much of the memory, reducing the need for hardware. And it also greatly speeds up detection, as a result it is well suited to use in a resource-constrained and rapid computing environment.

3.4 System Specification

For the current implementation of this paper, the camera and system with the following specification are used to capture and test the textile material. Table 4 specifies system specification.

Table 4. System specification

Component	Specification
Camera	DSLR
RAM	8 GB
Processor	Intel Core i7-7700HQ
Graphic card	NVIDIA GeForce GTX 1050Ti 4 GB

With this system environment, we were able to implement YOLOv2 for fabric defect detection. Due to the limitation of the RAM and Graphics card, where YOLOv3 requires 16 GB and NVIDIA GeForce GTX 1080Ti 8 GB GPU, we implemented YOLOv3 Tiny in the available system environment.

4 Data Preprocessing

The dataset created had a total of 1000 images of both defect and non-defect images put together. The images were resized to $416 \times 416 \times 3$ to feed into the neural networks. The defect data images were annotated with a GUI tool called labelImg. The neural network is fed with annotated defect images to train the model. After running K means clustering separately for YOLOv2 and YOLOv3 Tiny on the manually labelled boxes, it turns out that most of the manually labelled boxes which is referred as actual bounding box have certain height-width ratios. So instead of directly predicting a rectangle to localize the defects, YOLOv2 and YOLOv3 predict off-sets from a pre-established set of boxes with particular height-width ratios to overcome the difficulty of detecting the objects having midpoints in the same grid we call these as anchors which are set before training the neural network using the actual bounding box. In YOLO v2, 5

anchor boxes are used and in YOLOv3 Tiny, tiny 6 anchor boxes are used. While training, the model uses this anchor box to localize the defect accurately especially when there are multiple defects in a single image and while predicting the defects in the test dataset bounding boxes are predicted relative to these predefined anchors.

5 Implementation

The dataset was split as 80-20. The YOLOv2 and YOLOv3 tiny model was trained separately using the 80% of the images from the fabric defect dataset and tested on the randomly picked 20% images of fabric defect dataset.

Table 5. Layer information - YOLOv2

Number	Layer	Input	Output
0	conv	$416 \times 416 \times 3$	$416 \times 416 \times 32$
1	max	$416 \times 416 \times 32$	$208 \times 208 \times 32$
2	conv	$208 \times 208 \times 32$	$208 \times 208 \times 64$
3	max	$208 \times 208 \times 64$	$104 \times 104 \times 64$
4-6	conv	$104 \times 104 \times 64$	$104 \times 104 \times 128$
7	max	$104 \times 104 \times 128$	$52 \times 52 \times 128$
8-10	conv	$52 \times 52 \times 128$	$52 \times 52 \times 256$
11	max	$52 \times 52 \times 256$	$26 \times 26 \times 256$
12-16	conv	$26 \times 26 \times 256$	$26 \times 26 \times 512$
17	max	$26 \times 26 \times 512$	$13 \times 13 \times 512$
18-24	conv	$13 \times 13 \times 512$	$13 \times 13 \times 1024$
25	route		
26	conv	$26 \times 26 \times 512$	$26 \times 26 \times 64$
27	reorg	$26 \times 26 \times 64$	$13 \times 13 \times 256$
28	route		
29-30	conv	$13 \times 13 \times 1280$	$13 \times 13 \times 30$

Table 6. Layer information - YOLOv3 Tiny

Number	Layer	Input	Output
0	conv	$416 \times 416 \times 3$	$416 \times 416 \times 16$
1	max	$416 \times 416 \times 16$	$208 \times 208 \times 16$
2	conv	$208 \times 208 \times 16$	$208 \times 208 \times 32$
3	max	$208 \times 208 \times 32$	$104 \times 104 \times 32$
4	conv	$104 \times 104 \times 32$	$104 \times 104 \times 64$
5	max	$104 \times 104 \times 64$	$52 \times 52 \times 64$
6	conv	$52 \times 52 \times 64$	$52 \times 52 \times 128$
7	max	$52 \times 52 \times 128$	$26 \times 26 \times 128$
8	conv	$26 \times 26 \times 128$	$26 \times 26 \times 256$
9	max	$26 \times 26 \times 256$	$13 \times 13 \times 256$
10	conv	$13 \times 13 \times 256$	$13 \times 13 \times 512$
11	conv	$13 \times 13 \times 512$	$13 \times 13 \times 512$
12-15	conv	$13 \times 13 \times 512$	$13 \times 13 \times 33$

YOLOv2 model consists of 23 Convolution Layers, 5 Max Pooling Layers, 2 Route Layers, and 1 Reorg layer. Before training the YOLOv2 model for fabric dataset the followings parameters were tuned; the number of classes was set as 6, the batch parameter and subdivisions were fixed as 64 and 16 respectively, the maximum batches was set as 12000 using the formula - number of classes * 2000, steps were set as $0.8 * \text{maximum batches}$, $0.9 * \text{maximum batches}$ which equals to 9600 and 10800. The number of filters used was 33 calculated using the formula - $(\text{Number of Classes} + 5) * 3$. The five anchor box values used are $[[1.3221, 1.73145], [3.19275, 4.00944], [5.05587, 8.09892], [9.47112, 4.84053], [11.2364, 10.0071]]$. The Table 5 explains YOLOv2 architecture and its layers in detail.

YOLOv3 model consists of 10 Convolution Layers, and 6 Max Pooling Layers. In YOLOv3, the six anchor values used are $[[10, 14], [23, 27], [37, 58], [81, 82], [135, 169], [344, 319]]$ and rest of the parameters used are same as YOLOv2. The Table 6 explains YOLOv3 Tiny architecture and its layers in detail. The proposed model is implemented using both YOLOv2 and YOLOv3 Tiny separately with the architecture mentioned in the Table 5 and Table 6 respectively.

6 Results and Discussion

In defect detection, the actual bounding boxes are the manually labeled box which specify where the defect is in the image and the predicted bounding is the box the model gives as an output. IOU is used for the evaluation of the model which is computed by Area of Intersection divided over Area of Union of those two boxes. For detection to be accurate this paper use Non Max Suppression (NMS). In NMS all the boxes having probabilities less than or equal to a pre-defined threshold are discarded. For the remaining boxes, box with the highest probability is chosen and taken as the output prediction and boxes having IoU greater than the threshold with the output box from the above step are discarded. Mean Average Precision is used as one of the factor to check the performance of the model. mAP is calculated by taking the mean of average precision over all classes and/or over all Intersection over Union thresholds and Average Precision score calculated by taking the average value of the precision across all recall values. The formula for IOU, Precision and recall are mentioned in the Fig. 3.

The field with greater risk should have the IoU value high. But, in the fabric industry, the defects produced doesn't cause any risk to human lives and our primary focus is to know the type of defect and the prime area where the defect occurred rather than the exact point, therefore IOU is set to 0.5, reducing further will make the model to point totally a different area relative to the defect which will not solve our purpose.

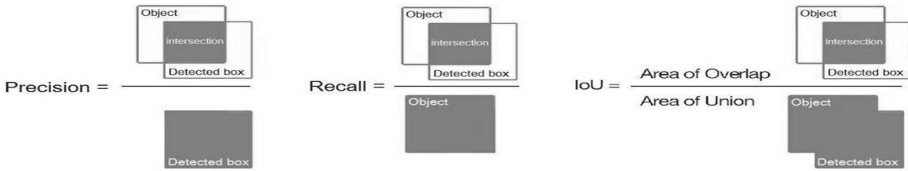


Fig. 3. Formula for calculating Precision, Recall and IOU

The Performance Metrics of YOLOv2 and YOLOv3 Tiny is shown in Table 7. The IoU threshold used for both the models was 50%, used Area-Under-Curve for each unique (Fig. 4 and Table 8).

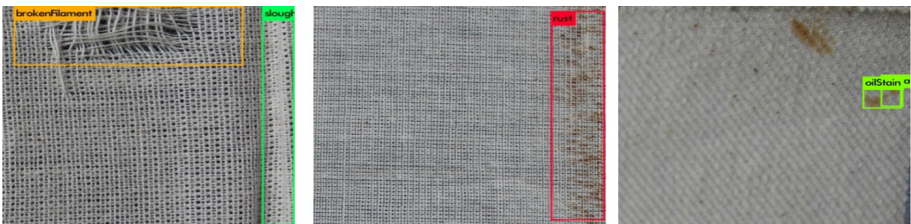


Fig. 4. Experimental test results of YOLOv2

Table 7. Performance metrics comparison

Model	Precision	Recall	F1-score	Accuracy	mAP
YOLOv2	0.77	0.66	0.71	83.12	65.25
YOLOv3	0.58	0.60	0.50	72	60.13

Table 8. Class-wise accuracy of YOLOv2

Class	Hole	Rust	Broken filament	Oil stain	Slough	Grease	Average accuracy
Accuracy	87.5	78.125	85	76.5	86.6	85	83.12

7 Conclusion and Future Work

YOLOv3 Tiny is lighter and faster than YOLOv2, by sacrificing some accuracy for it. Initially, with a small dataset, YOLOv2 and YOLOv3 tiny gave similar same accuracy. But, with an increase in dataset resulted in the greater accuracy for YOLOv2. But in case of YOLOv3 tiny, as the number of layers was less, the model was not able to learn enough about the defects after a certain extent from the available dataset. This paper faces challenges due to evolving patterns and styles in the fabric material as even a hole in certain areas can be due to a new style. And the mixture of colors in the material can also decrease the accuracy of the model as the trained images are in a similar background.

As this model will not address all the defects identified in a cloth. The model can be trained with other type of defects. Due to lacking variety in data set the identification of the defects may be affected by various factors like evolving colours and patterns. A well developed data set can be created and trained. As we were short of dataset, we produced the dataset and processed it in a short period of time. Further improvements to the paper in the upcoming years can be made by including same background, different background, single color fabric, multicolored fabric image datasets. In the future, methods to be discussed to implement the proposed system in a cost-efficient manner in the industries.

References

1. Abdel-Azim, G., Nasri, S.: Textile defects identification based on neural networks and mutual information. In: 2013 International Conference on Computer Applications Technology (ICCAT) (2013)
2. Sujee, R., Padmavathi, S.: Image enhancement through pyramid histogram matching. In: 2017 International Conference on Computer Communication and Informatics (ICCCI)
3. Rebhi, A., Abid, S., Fnaiech, F.: Fabric defect detection using local homogeneity and morphological image processing. In: 2016 International Image Processing, Applications and Systems (IPAS), Hammamet (2016)
4. Jmali, M., Zitouni, B., Sakli, F.: Fabrics defects detecting using image processing and neural networks. In: 2014 Information and Communication Technologies Innovation and Application (2014)

5. Jeong, H., Park, K., Ha, Y.: Image preprocessing for efficient training of YOLO deep learning networks. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai (2018)
6. Zhang, H., Zhang, L., Li, P., Gu, D.: Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks. In: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi (2018)
7. Huang, R., Pedoeem, J., Chen, C.: YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers (2018)
8. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger (2017)