A REPORT

**ON**

# Leather Defect Classification using CSP DenseNet

BY


| | |
|---|---|
| Kadam Hrushikesh Sanjay | 2020B4A71093G |
| Mane Atharv Mahesh | 2020A7PS0153G |


AT

CENTRAL ELECTRONICS ENGINEERING RESEARCH INSTITUTE (CEERI),

CHENNAI CENTRE

A Practice School-I Station of

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**(July, 2022)**

**A REPORT**

**ON**

# Leather Defect Classification using CSP DenseNet

BY


| | | |
|---|---|---|
| Kadam Hrushikesh Sanjay | 2020B4A71093G | Mathematics and Computer Science |
| Mane Atharv Mahesh | 2020A7PS0153G | Computer Science |


Prepared in fulfilment of the

Practice School-I Course Nos.

BITS C221/BITS C231/ BITS C241

AT

CENTRAL ELECTRONICS ENGINEERING RESEARCH INSTITUTE (CEERI),

CHENNAI CENTRE

A Practice School-I Station of

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**(July, 2022)**

# Acknowledgements

We have been immeasurably enriched by working under the supervision of Mr. Suriya Prakash, scientist at CEERI Chennai, and our group mentor, who has an excellent level of knowledge and who has an art of encouraging, correcting and directing us in every situation possible, which has enabled us to complete the project successfully. We would also like to acknowledge Mr. Amalin A. Prince, our PS Instructor. Lastly, we would like to acknowledge the role of BITS Pilani University in giving us this exposure through the Practice School course.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

(RAJASTHAN)

Practice School Division

**Station:** Central Electronics Engineering Research Institute (CEERI) , Pilani

**Centre:** Chennai **Duration:** 7 weeks **Date of Start:** 30/05/2022

**Date of Submission:** 19/07/2022

**Title of the project:** Leather Defect Classification using CSP DenseNet

| **STUDENT NAME** | **ID NUMBER** | **DISCIPLINE** |
|---|---|---|
| Kadam Hrushikesh Sanjay | 2020B4A71093G | Mathematics and Computer Science |
| Mane Atharv Mahesh | 2020A7PS0153G | Computer Science |

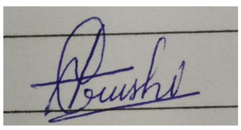
**Name and Designation of the Expert :** Mr. Suriya Prakash, Scientist, CEERI

**Name of the PS Faculty :** Mr. Amalin Prince

**Key Words :** Image Processing, Texture Classification, Deep Learning

**Project Areas :** Leather Quality Analysis, Applications of Image Processing

**Abstract :** Leather is a very essential raw material for manufacturing industries. Prior to using this raw leather for manufacturing, a visual inspection is done to confirm the quality of the leather used. Here we have attempted to automate the inspection process. To achieve better productivity of defect classification in industrial leather, we decided to implement a CSP-DenseNet-based Leather defect classification algorithm. A dataset of 3600 images was used to train the Deep Learning model. Cross Entropy Loss was used for the loss function, which was back-propagated on the Adam Optimizer. The model consists of 3 CSP-Dense-Blocks of 6,12, and 24 dense layers. It classifies leather into six categories namely; folding marks, grain off, growth marks, loose grain, non-defective and pinhole.

**Signature of Students:** **Signature of PS Faculty:**

**Date:** 19/07/2022 **Date:**

# Table of Contents

# 1. Introduction

Leather is a fabric made from animal hide and is treated with chemicals to make them suitable for manufacturing products like handbags, clothing, footwear, furniture, sports equipment, etc. The global leather goods market size was estimated at USD 394.12 billion in 2020. Because of the importance of leather in manufacturing industries, it is vital to use good quality leather to ensure a good quality product. Manual defect detection is one of the most time-consuming steps in the manufacturing process. Trained inspectors are required to use various angles and distances to examine the same section of leather multiple times. The defective areas are then marked manually, and then grade the leather, based on the level of defect. (i.e., minor, major, severe, etc.) and then classify defects.

This current process requires a lot of human effort and hence demands more time and capital. In addition, humans are prone to lack consistency and get fatigued which may lead to increased error. Hence, an automated vision-based mechanism is required to reduce human interference and improve consistency. Image processing and deep learning methods are being implemented and researched to achieve the same.

In this report, we have reviewed six research papers written by researchers working in the field All of them contain deep learning techniques. Guohua Liu and Xiangtong Zheng (2021) involves frequency domain analysis, Chen et. al(2021) is based on hyperspectral imaging and sixth one is a statistical approach. Rest are solely deep learning based. Luo et. al (2021), the RBD-Net model was chosen to implement. The algorithm classifies leather into six categories namely; folding marks, grain off, growth marks, loose grain, non-defective and pinhole.

# 2. Literature Review

## 2.1 Detection and localisation of defects on natural leather surfaces:

This paper by Gan et. al (2021) explores a neural network-based method to detect the presence of defects in texture and then localise and classify them. The paper first implements an AlexNet feature extractor and then applies a classifier for the detection of the presence of defects. The paper has tried both Support Vector Machine(SVM) and Softmax functions for this purpose. Then the paper implements a localisation and classification algorithm to classify the defects. The paper has tried both (You Only Look Once)YOLOv2 and Fast Region-based Convolutional Neural Network(RCNN) for this purpose. Even though the algorithm implemented in this paper's experimentation could classify the defects into lesser categories, it gave impressive results in terms of detection and localisation of the defects. It has achieved 100% maximum accuracy for the task with a maximum IoU of 74% with Fast RCNN and 64% with YOLOv2.

## 2.2 Fabric defect detection based on information entropy and frequency domain saliency:

This paper by Guohua Liu and Xiangtong Zheng (2021), introduces a two-dimensional entropy matrix, which reflects one-dimensional entropy information of the image spatially, according to the relationship between information entropy and image texture. The image is reconstructed into a quaternion matrix by combining two-dimensional entropy and three feature maps that characterise the opponent colour space representation of the input image. The paper then uses a hypercomplex Fourier transform to transform the quaternion image matrix into the frequency domain. The paper proposes a new method for local tuning of amplitude spectrum, thereby suppressing the background pattern while retaining the defect region. Finally, the inverse transform is performed to obtain a saliency map. One can then threshold this saliency map and localise the defect accurately. The experiments on implementation of this paper were done over a dataset of 336 images of fabrics of various designs (stars, dots and boxes), obtaining impressive accuracies of 98.38%,95.18%, and 97.74% in each design, respectively.

## 2.3 Texture defect classification with multiple pooling and filter ensembles based on a deep neural network:

This model introduced by Uzen et. al (2021) consists of three basic stages: preprocessing, feature extraction, and classification. In the preprocessing stage, the texture images were first divided into $n \times n$ equal parts. Then, median filtering and pooling processes were applied to each piece prior to performing image merging. The proposed pre-treatment stage is aimed at clarifying fabric errors and increasing performance. For the feature extraction stage, deep features were extracted from the texture images using the pre-trained ResNet101 model based on the transfer learning approach. Finally, classification and testing procedures were conducted on the obtained deep-effective properties using the SVM method. The multiclass TILDA dataset was used in order to test the proposed model. In experimental work, the MPF-DNN model for all four classes achieved a significant overall accuracy score of 95.82%

## 2.4 RBD-Net: Robust Breakage Detection (RBD) algorithm for industrial leather:

This paper by Luo et. al (2021) introduces a new architecture, the RBD-Net. The model is a modification of the YOLO v5 model which uses Cross Stage Partial Dense Net (CSP-DenseNet) as its backbone, Bidirectional Feature Pyramid Network (BiFPN) as its neck and the Decision Network head. The backbone of the algorithms essentially is used for the extraction of features. The neck is BiFPN neck is used to assign weights to the features and thus decide the importance of each feature in the final prediction. The addition of the Decision Network allows the network to capture local as well as global shapes spanning a large area of the image for better identification of leather defects in the image. The algorithm is not only a defect classification algorithm but also a defect localization algorithm.

## 2.5 Surface Defect Detection of Wet-Blue Leather Using Hyperspectral Imaging

This paper is by Chen et. al(2021). It introduces an inspection based on 5 defects including brand masks, rotten grain, rupture, insect bites, and scratches. Hyperspectral Leather Defect Detection Algorithm(HLDDA) is used. It includes Hyperspectral Target Detection(HTD) and Deep Learning(Deep Learning) techniques. In HTD, Weighted Background Suppression Constrained Energy Minimization (WBS-CEM) and WBS-Hierarchical CEM (WBS-hCEM) were developed by using weighting to suppress the background and enhance the contrast between the target and background. In Deep Learning, 1D-Convolutional Neural Network (CNN), 2D-Unet and 3D-UNet architectures were designed to segment defect areas. 1D-CNN emphasizes on defects with spectral features, 2D-Unet emphasizes on defects with spatial features, and 3D-Unet can simultaneously process both. The experimental results show that the algorithm could quantify and estimate the size of the defect effectively.

## 2.6 Automated leather defect inspection using statistical approach on image intensity.

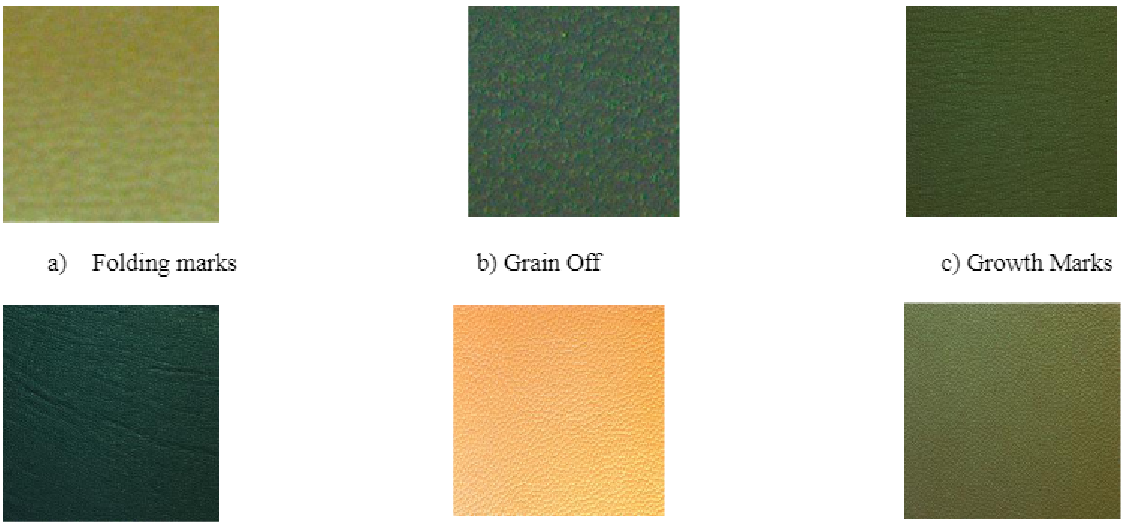
This paper by Gan et. al(2020) proposes a method to detect defective images using histogram analysis of grayscale pixel intensities. The first step is to convert the image into grayscale and then a Gaussian filter is applied. The purpose of the gaussian filter is to emphasize the defect as much as possible. Each image is then divided into smaller segments. While dividing the image it is important to consider that the size of each segment must be around the same size as the defect area. This is done so that the difference between defective and non-defective segments is maximized. The next step is to compute statistical features for each segment which include; Sensitivity, Specificity, Precision, F1-score, Error rate and  Accuracy. An algorithm based on Kolmogorov-Smirnov's test is then used for feature selection. Finally, features are classified by known classifiers, which are: SVM, k-Nearest Neighbors, decision tree, ensemble classifier and Naive Bayes. The evaluation metrics for the model are then determined using a 70/30 for the train/test split. Dataset I resulted in an average accuracy of 97.11% and F1-score of 97.39%. Dataset II resulted in an average accuracy of 74.02% and an average F1-score of 82.35%

# 3. Approach

It was noted from the literature reviewed above that Deep Learning Models achieved excellent results in the classification and localisation of the marked defects in leather and an overall texture. Thus we chose to continue on these lines with the most state-of-the-art Deep Neural Networks (DNN). Due to the lack of an appropriately annotated dataset, training a DNN to localise the leather defects as in RBD-Net by Luo et al. (2021) was ruled out. Thus we decided to implement a CSP-DenseNet-based Leather defect classification algorithm.

# 4. Dataset Distribution

The dataset of **3600** images made by Moganam (2020) was used for training the model. The dataset has a total of six different classes for classifying different defects, including the class which has pictures of non-defective leather. Fig1. displays one sample from each of these classes. The data was pre-processed by resizing the image down to 227*227 pixels. The dataset was split in training, testing and validation subsets in 76%,12%,12% format each. The data was trained using the free version of Google Colab which offered us an NVIDIA Tesla K80 GPU to train, validate and test the model.



a) Folding marks b) Grain Off c) Growth Marks

d) Loose Grain e) Non Defective f) Pinhole

**Fig1. :** Samples from the dataset
*Source: Moganam, Kaggle dataset (2022)*

# 5. Architecture of the model

The model consists of **3 CSP Dense Blocks** of **6,12, and 24** Dense Layers. For the dense layer, the PyTorch pre-made implementation was used. The input to the network is an image. The network outputs the class to which the image belongs.

### 5.1.1. CSP DenseNet

Inspired by Wang et al. (2019), we implemented a CSP DenseNet Backbone for the classification model. The CSP DenseNet consists of “Blocks”, which in turn consist of “Dense Layers”. The input features to these blocks are first divided into two groups. One of these layers is passed through the Dense Layer while the other is simply copied and concatenated with the processed layer. **Fig2.** shows the overall representation of a block of architecture. The block ends with a “transition layer”, which acts as a downsampling mechanism to reduce the number of features for subsequent blocks, thus making the number of features not increase excessively and making the task computationally impossible. There are four different ways in which the transition layers can be connected to the last dense layer. **Fig3.** shows all the ways in which the transition blocks can be attached. In this report, we have chosen the complete CSP DenseNet approach.

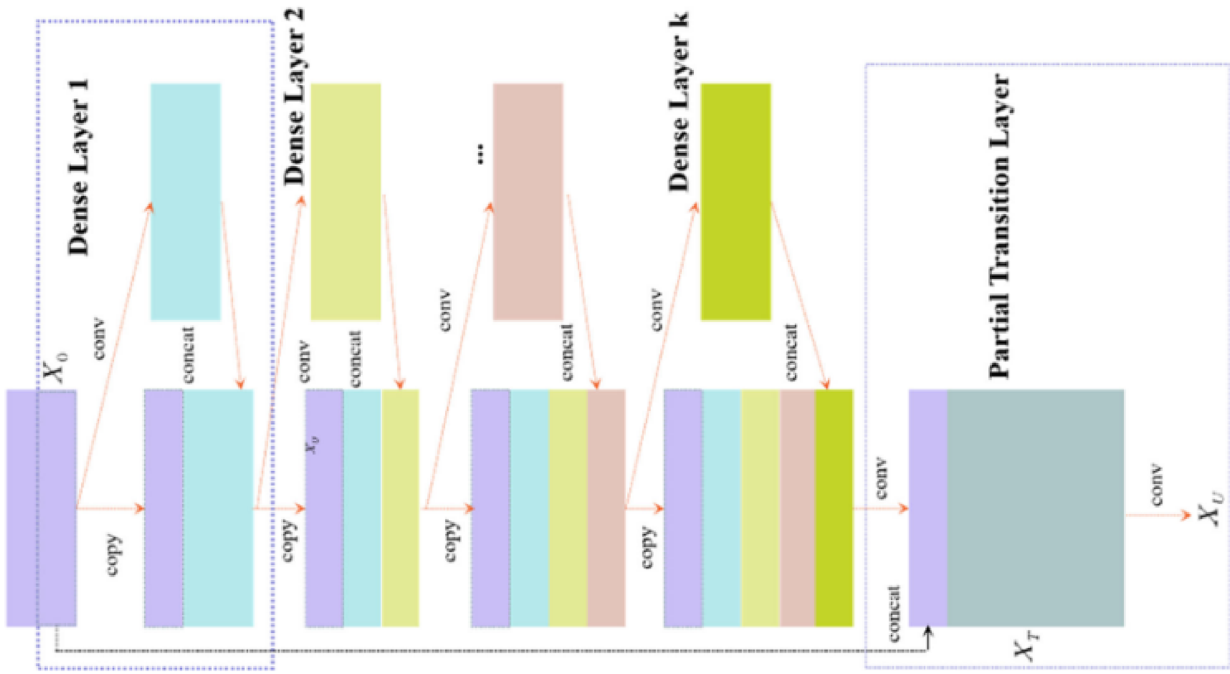



**Fig2.:** CSP- Dense Block Model
*Source: Wang et al. (2019)*

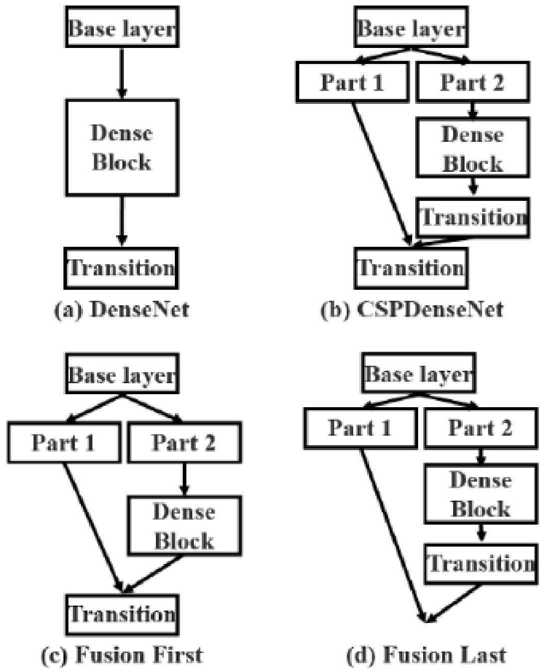



**Fig3.:** Ways of putting transition blocks in the CSP DenseNet implementation.
*Source: Wang et al. (2019)*

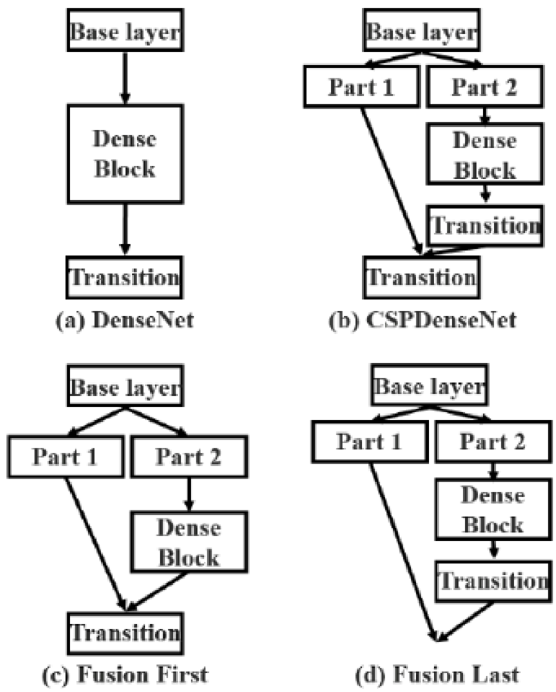



**Fig3.:** Ways of putting transition blocks in the CSP DenseNet implementation.
*Source: Wang et al. (2019)*

## 5.1.2. Dense Layers

CSP DenseNet builds upon the Dense Layers. Dense layers are convolutional layers in a CSP dense block where for a given layer in the block, outputs of all the previous layers along with the original input features are concatenated with the output of the current layer.
**Fig4.** shows how each Dense Layer looks.

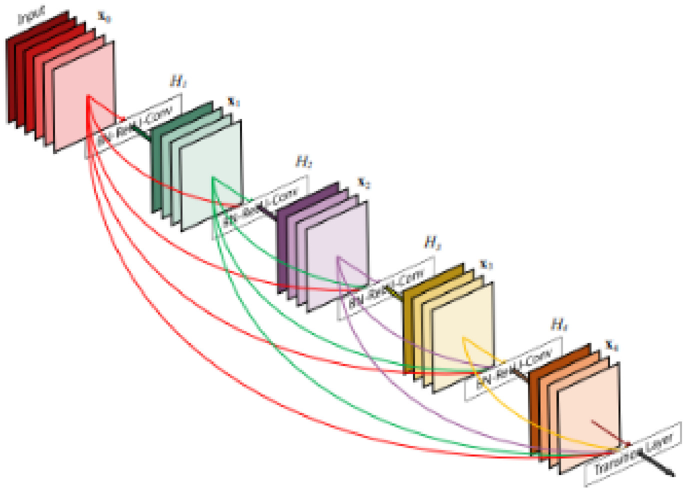


**Fig4.**: Connections between layers in Dense Block

## 5.1.3. Training Procedure

The CSP DenseNet implementation was trained on a dataset of 3600 images with due pre-processing as described in Section 4. with shuffling between the batches for each epoch. The shuffling was done to avoid overfitting. The image size of 227*227 was used to maintain maximum information in the image.
Considering the computational resources at hand, a batch size of 32 was used. Cross Entropy Loss was used for the loss function, which was backpropagated on the Adam Optimizer.
The model was trained in two sets of 20 Epochs considering the fact that because of long training times, the Google Colab server runtime might disconnect, and weights for each epoch were saved for further use. This approach also helped us to fine-tune our learning rate

### 5.1.4. Results

After completion of the training, the CSP DenseNet model achieved following results: Final average training accuracy = 91.2646% validation accuracy = 90.41667. After passing through the test dataset we found the accuracy to be 95.3704%. Fig5. shows the graph of the training accuracy function for each epoch while Fig6. shows the graph of how validation accuracy varies as a function of epoch.

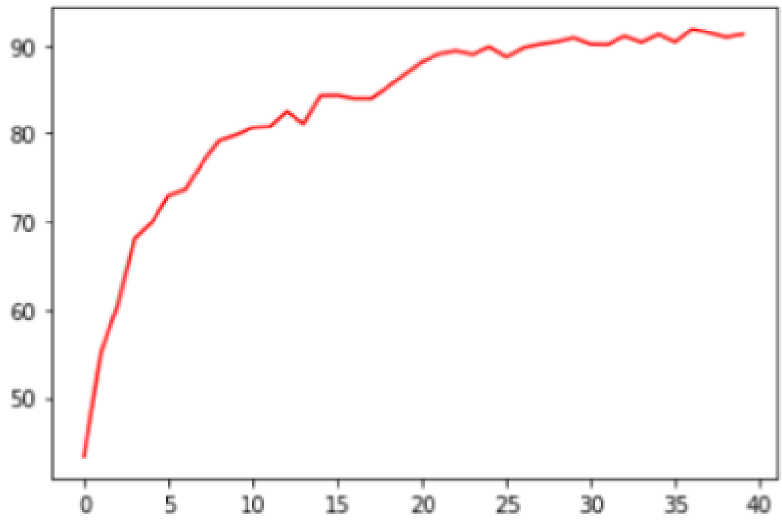



**Fig5:** Training Accuracy

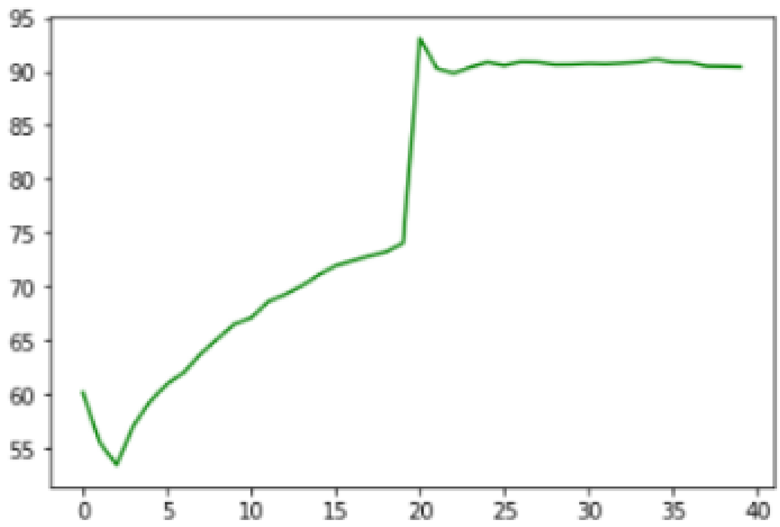



**Fig6.** Validation Accuracy Graph

# 6. Conclusion and Recommendation

To conclude this report, we have seen how important literature review is while solving an industry based problems related to Computer Vision and Machine Learning in general. We learnt how to read through a large amount of literature in small time and condense its matter for personal as well as implementational understanding, without losing out information. We also learnt how to implement Deep Learning architectures using PyTorch like CSP DenseNet.

We provide the following as recommendations for improvement over our model:

1. Use BiFPN for getting better feature outputs.
2. Getting a saliency map of the images using gradients and then using unsupervised learning techniques to segment the defects since the defects will be brighter amd highlighted from the background in the saliency map. In this way one can add a localization feature to the model without the need of a bounding box labeled dataset.

# References


**Gao Huang, Zhuang Liu, Laurens van der Maaten (2018) Densely Connected Convolutional Networks. IEEE** **https://ieeexplore.ieee.org/abstract/document/8721151?casa_token=uj7hy8kQbREAAAAA:aZj4LRzqsCVvfXoNgdQ2WzsYLl6QRiGg8LZUJn1vA6gPCXcUkQkmZFDTeNqwlld3w05pwV_eNPs**

**Y. S. Gan, Sze-Teng Liong, Danna Zheng, Yiyang Xia, Shuli Wu, Mengchen Lin, Yen-Chang Huang (2021) Detection and localization of defects on natural leather surfaces. Springer** **https://link.springer.com/article/10.1007/s12652-021-03396-2**

**Guohua Liu and Xiangtong Zheng (2021) Fabric defect detection based on information entropy and frequency domain saliency, Springer.** **https://link.springer.com/article/10.1007/s00371-020-01820-w**

**Huseyin Uzen, Muammer Turkoglu, Davut Hanbay (2021) Texture defect classification with multiple pooling and filter ensembles based on a deep neural network, Elsevier.** **https://www.sciencedirect.com/science/article/pii/S0957417421002797**

**Rong Luo, Ruihu Chen, Fengting Jia, Biru Lin, Jie Liu, Yafei Sun, Xinbo Yang, Weikuan Jia (2021) RBD-Net: Robust Breakage Detection (RBD) algorithm for industrial leather, Springer.** **https://link.springer.com/article/10.1007/s10845-022-01962-9**

**Y. S. Gan1, Sue-Sien Chee2, Yen-Chang Huang, Sze-Teng Liong, Wei-Chuen Yau(2020) Automated leather defect inspection using statistical approach on image intensity, Springer**
https://link.springer.com/content/pdf/10.1007/s12652-020-02631-6